♦ Mobility Management for SIP Sessions in a Heterogeneous Network Environment

Willem A. Romijn, Dirk-Jaap Plas, Dennis Bijwaard, Erik Meeuwissen, and Gijs van Ooijen

The next generation of communication networks is expected to create a heterogeneous network environment encompassing an ever-increasing number of different access networks and end-user terminals that will enable the introduction of and provide access to numerous feature-rich end-user services. It is essential that end users be able to roam from one access network to another if they are to enjoy a seamless roaming experience, which is especially important for multimedia applications such as voice, audio, and video. This paper describes how to make such a roaming experience possible in multimedia applications based on Session Initiation Protocol (SIP). It also describes several mobility management solutions and compares the suitability of SIP sessions for roaming across General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), and wireless local area network (WLAN) networks. The comparison is based on the implementation of a prototype. The advantages and disadvantages of each mobility management solution are discussed, as are the issues encountered during the implementation of the prototype.

Introduction

In the beyond-3G environment, users will have access to an increasing number of different access networks, ranging from traditional networks (e.g., Global System for Mobile Communications [GSM] and General Packet Radio Service [GPRS]) to the currently emerging network of Wi-Fi hotspots to future telecommunication access networks (e.g., Universal Mobile Telecommunications System [UMTS] terrestrial radio access network [UTRAN]), and possibly to other networks as well. To facilitate seamless continuation of services across these networks, users should be able to roam seamlessly from one access network to another. Mobility management, which is the technical prerequisite for such roaming behavior, involves controlling the network to which the user's terminal is connected—i.e, it involves discovering new access networks and switching from one access network to another [4]. The services that can be supported on an access network depend on the characteristics (e.g., the bandwidth restrictions) of the network; certain services may not be supported on certain networks. Therefore, it may be necessary to adapt ongoing service sessions to changes in the network environment. A typical example of such an adaptation is dropping video from

Bell Labs Technical Journal 9(3), 237–253 (2004) © 2004 Lucent Technologies Inc. Published by Wiley Periodicals, Inc. Published online in Wiley InterScience (www.interscience.wiley.com). • DOI:10.1002/bltj.20053

an audio-video session for a low-bandwidth access network.

One of the technologies commonly used for mobility management is Mobile Internet Protocol (MIP) [11]. MIP provides a transparent mobility management solution at the network layer for Internet Protocol (IP)-based applications. Following the approach of the IP Multimedia Subsystem (IMS) of the 3rd Generation Partnership Project (3GPP) [1], we consider multimedia services based on the Session Initiation Protocol (SIP) [8]. SIP is an application-layer protocol for setting up, maintaining, and terminating (multimedia) sessions. SIP has its own mechanisms for mobility management [13] for SIP-based applications as well as functionality for session adaptation.

There have been a number of studies comparing SIP-based and MIP-based mobility management. The comparisons of the performance of the two protocols in [14] and [3] demonstrate that, in general, application-layer mobility management protocols, such as SIP, perform worse than lower-layer protocols in terms of handoff delay, signaling overhead, and transparency. However, when suitability for deployment in next-generation networks is considered, it appears that SIP is a better mobility management solution, because it obviates the need for protocol stack and infrastructure changes [3]. A number of studies indicate that the suitability of a mobility management solution depends primarily on the type of application for which it is being considered. For long-lived Transmission Control Protocol (TCP) connections (e.g., FTP) and most standard Internet applications (e.g., Web browsing [i.e., HTTP] and chat), MIP offers a generic solution for roaming that seems to work well. However, for real-time applications, SIP is recommended [6, 14], because real-time applications (e.g., multimedia applications) have strict timing requirements that are not taken into account by MIP because it is a network-layer protocol. To optimize roaming behavior, applications should be able to influence or even control the mobility management process, as they can when SIP is used as the mobility management solution. An additional benefit of using SIP for application-layer mobility management is that it allows applications to adapt their service behavior, based on the mobility management strategy selected, to provide the best possible end user experience.

Panel 1. Abbreviations, Acronyms, and Terms

3GPP—3rd Generation Partnership Project
4GPLUS—4th Generation Platform Launching Ubiquitous Services
ACK—Acknowledgment
API—Application programming interface
CoA—Care-of-address
FTP—File Transfer Protocol
GPRS—General Packet Radio Service
GSM—Global System for Mobile Communications
HA—Home agent
HTTP—Hypertext Transfer Protocol
IMS—IP Multimedia Subsystem
IP—Internet Protocol
IPv4—IP version 4
JAIN*—Java* API for integrated networks
JMF—Java media framework
LAN—Local area network
MIP—Mobile IP
MIPv4—MIP version 4
MX—Multiple access
NIST—National Institute of Standards and Technology
P-CSCF—Proxy call session control function
RTP—Real-Time Transport Protocol
SDP—Session Description Protocol
SIP—Session Initiation Protocol
TCP—Transmission Control Protocol
UA—User agent
UDP—User Datagram Protocol
UMTS—Universal Mobile Telecommunications System
UTRAN—UMTS terrestrial radio access network
WLAN—Wireless local area network

In this paper we describe and compare several approaches to using MIP and SIP for mobility management and session adaptation of SIP-based multimedia applications. Within the scope of the Dutch research project 4th Generation Platform Launching Ubiquitous Services (4GPLUS) [2], we have designed and implemented a SIP client that combines mobility management—both MIP and SIP—and session adaptation [4, 5] and have used it to evaluate both technologies. The next section describes how MIP and SIP support mobility management and session adaptation. Then we describe the SIP client architecture and the prototype implementation in detail. Next, based on two scenarios, we describe the results of using MIP and SIP for mobility management and session adaptation of SIP-based applications. We conclude with recommendations and future work.

Mobility Management Technologies

End-user terminals are becoming more and more powerful. They can now have simultaneous access via multiple network interfaces to a variety of different types of networks. Each active network typically provides an IP address—either private or public—so it can address the terminal. This means that the terminal may have multiple, varying IP addresses when using different access technologies while switching between access networks. There are several protocols that help to manage these multiple changing IP addresses and to address the terminal consistently. MIP can be used to provide a fixed IP address to local applications and peers; the user can select one of the connected networks to send and receive traffic using this IP address. SIP can be used to initiate multimedia sessions with and receive multimedia sessions from other users. The first subsection below describes how MIP can be used to manage mobility. (Throughout the paper, the discussion of MIP is based on MIP version 4 [MIPv4].) The second subsection describes how SIP alone (i.e., without MIP) can be used to manage mobility.

Mobility Management with MIP

Figure 1 shows how the user of the terminal on the left, which has network interfaces to a local area network (LAN), a wireless LAN (WLAN), and GPRS, can set up a multimedia session with the other terminal.

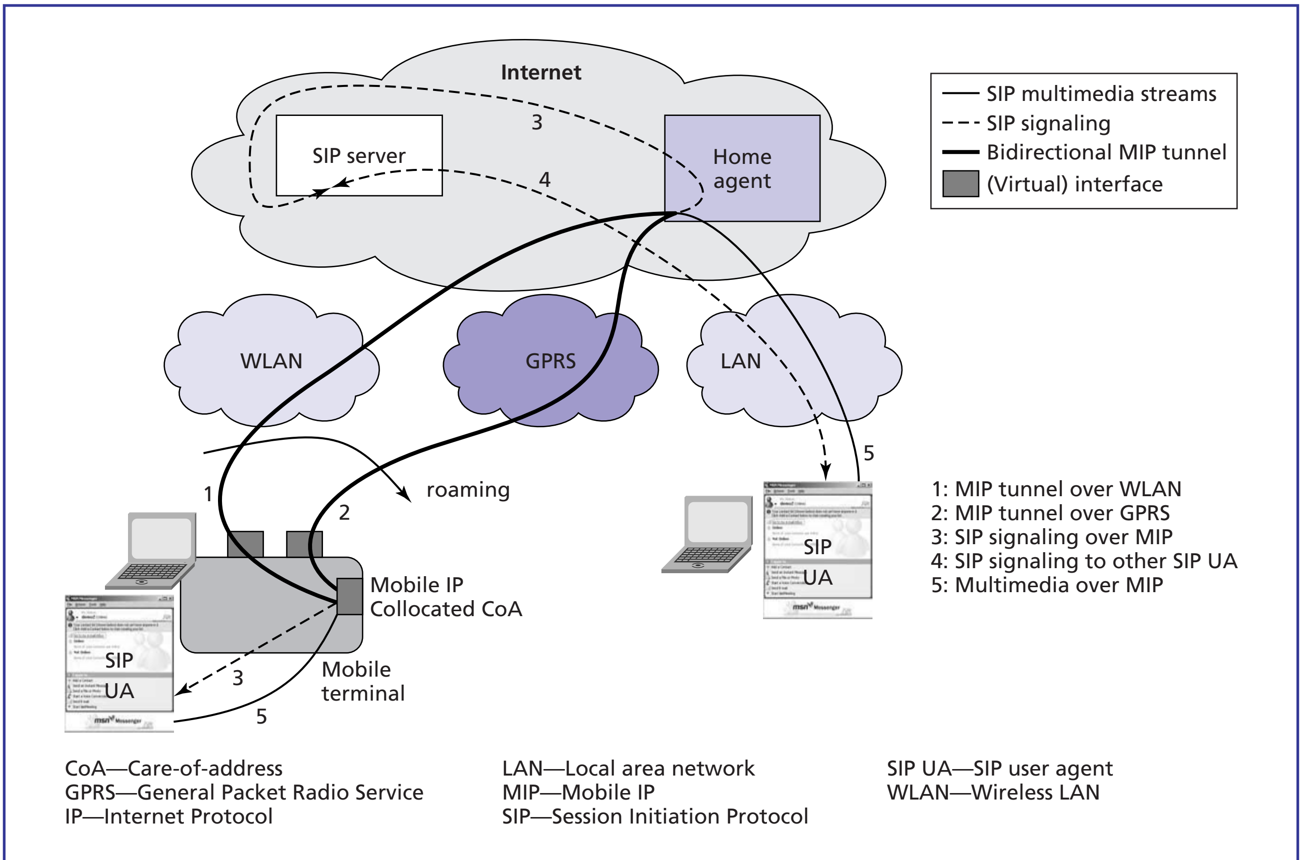


Figure 1.
Architecture supporting MIP.

MIP operates in the network layer of the TCP/IP protocol stack and handles the mobility of IP addresses in multiple access networks transparently for applications. MIP cannot hide network characteristics, which means that the behavior of applications can be unpredictable when the network characteristics required (e.g., a certain amount of bandwidth) cannot be met when switching over to a new network.

In MIP, a terminal has a fixed IP address on the MIP home agent (HA), which takes care of tunneling traffic destined for this fixed IP towards the terminal. Whenever the terminal changes its access network (and, therefore, the IP address of the physical interface), it sends a binding update to the HA to inform it of the new point of attachment.

MIP can use a special router—called a *foreign agent*—in a visited network that will forward all traffic that has the home IP address as source to the destination. (The normal rule for routers would be to drop such traffic.) Similarly, the HA will tunnel the traffic destined for the terminal to the foreign agent, which in turn will forward it to the terminal. In this case, MIP manages the tunneling, forwarding, routing, and mapping. Most MIP implementations can also decide which physical interface to use.

MIP also works without a foreign agent in the visited network by using a collocated care-of-address (CoA). With a collocated CoA, a bidirectional tunnel is used between the HA and the terminal. In this case, the terminal needs an IP address in the visited network, and MIP has its own virtual interface in the terminal. This interface is usually set as the default route for applications. One advantage of this approach is that triangular routing is avoided; the traffic between the mobile terminal and the corresponding host follows the same path. Another advantage is that the terminal also has a local IP address in the visited network. This enables certain applications to bypass MIP and handle mobility by themselves. Both solutions can be used for the mobility of SIP sessions. For our prototype and performance measurements, we have chosen MIP with a collocated CoA.

Figure 1 shows two alternative bidirectional MIP tunnels:

- The bidirectional MIP tunnel over a WLAN between the terminal and the HA, and
- The bidirectional MIP tunnel over GPRS between the terminal and the HA.

When only one of the access networks is available, or when one is explicitly chosen for MIP, the tunnel through that access network is used for the MIP traffic.

Mobility Management with SIP

Figure 2 shows how the user of the terminal on the left can set up a multimedia session with the other terminal in a situation in which MIP is not used.

SIP clients can use the re-REGISTER and re-INVITE messages [8] for mobility management. The re-REGISTER message can contain a new IP address at which a SIP client may be reached after it has roamed to a different network. The re-INVITE message can be used to adapt an existing SIP session (e.g., by adding video to or removing it from the session) after the SIP client has, for example, roamed to a different access network. These capabilities are illustrated in **Figure 3**.

As shown in Figure 3, a media session is first established between user agent 1 (UA1) and user agent 2 (UA2) in the standard way. Then, the terminal of UA1 roams to a different access network, from which it obtains a new IP address. Next, UA1 registers itself again with the new IP address by sending a re-REGISTER message. Finally, UA1 modifies the existing session by sending a re-INVITE message. (This message would allow, for example, a session with both audio and video on a WLAN network to be changed to a session with audio only over a GPRS network.) Note that after the final acknowledgment (ACK) of the re-INVITE message, the original media session is modified according to the new agreed-upon session description parameters, in accordance with RFC 3261 [8]. As a result, for a short period, the non-roaming terminal may transmit data to a non-reachable destination. A more serious problem is that, when the roaming terminal moves to a network with much less available bandwidth, the original media streams can result in flooding on the new network connection, preventing SIP signaling messages from reaching the SIP server.

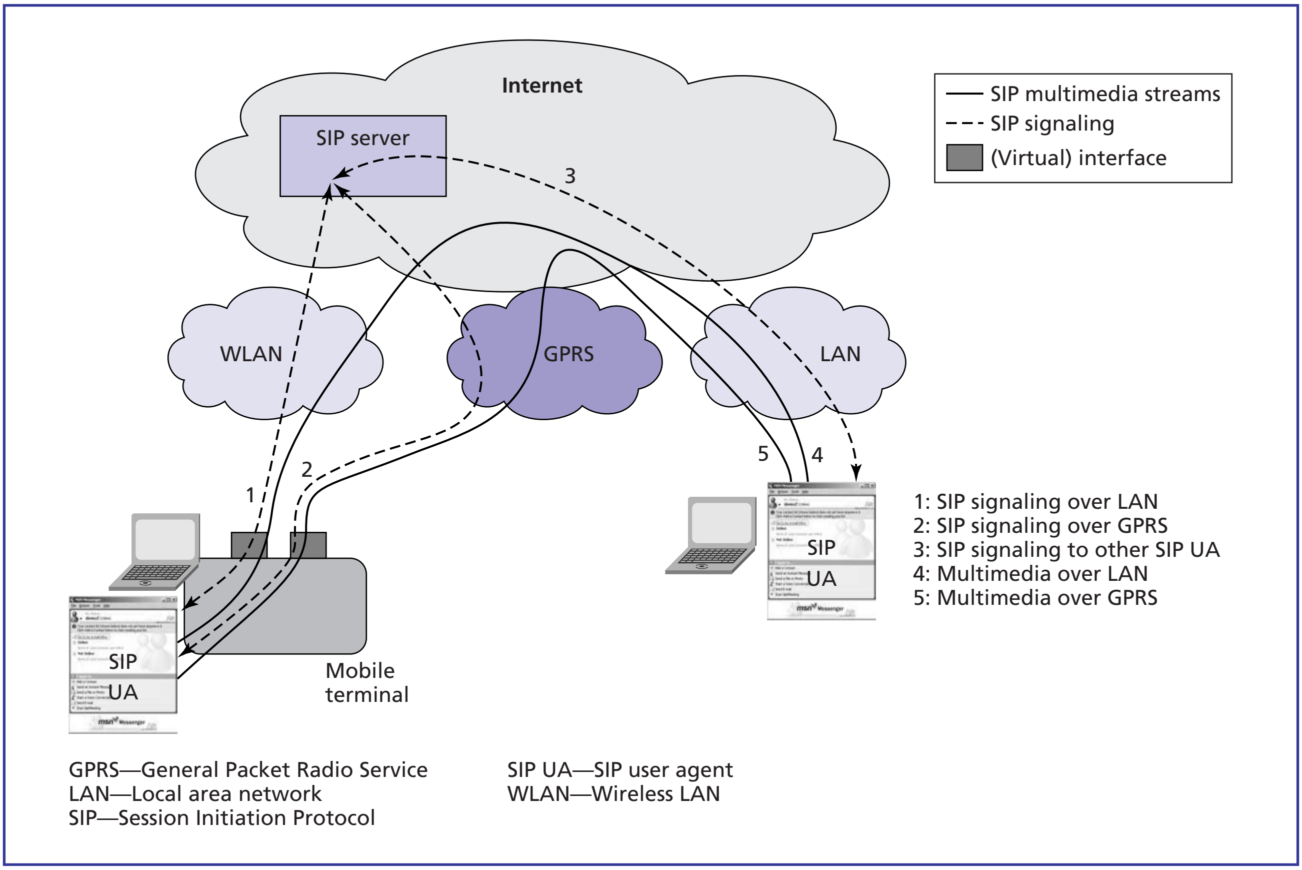


Figure 2.
Architecture supporting SIP.

A solution to this problem is proposed in the "Recommendations" section.

It is clear from the above that the terminal of the roaming user must have an intelligent entity that can be informed of any changes in the network environment of the terminal and can decide whether to trigger the re-REGISTER and/or the re-INVITE message. The next section discusses a complete SIP client architecture that contains such an entity.

SIP Client Architecture

The SIP client architecture to support mobility in a heterogeneous network environment is illustrated in **Figure 4**. The core of the multiple access (MX) SIP communicator is a standard non-mobility-aware SIP client using a standard SIP stack. To support mobility, we have added to the client the ability to receive information about network changes and to create Session Description Protocol (SDP) values that match the characteristics of the network. We have also added re-INVITE and re-REGISTER functionality that makes it possible to adapt existing sessions to a new network. A *decision maker* is in control of all mobility management within the SIP client.

Decision Maker

The responsibility of the decision maker is to make mobility management decisions within the application, taking into account the information available on the active networks and their characteristics (e.g., bandwidth and cost of usage). After the mobility manager—which is not part of the application but part of the terminal—initiates a handover to a new access network or detects changes in the network parameters, the decision maker is triggered. The decision maker then decides if it should trigger the SIP application to update the parameters of the SIP session with the new network properties. The decision maker obtains its information from the mobility client

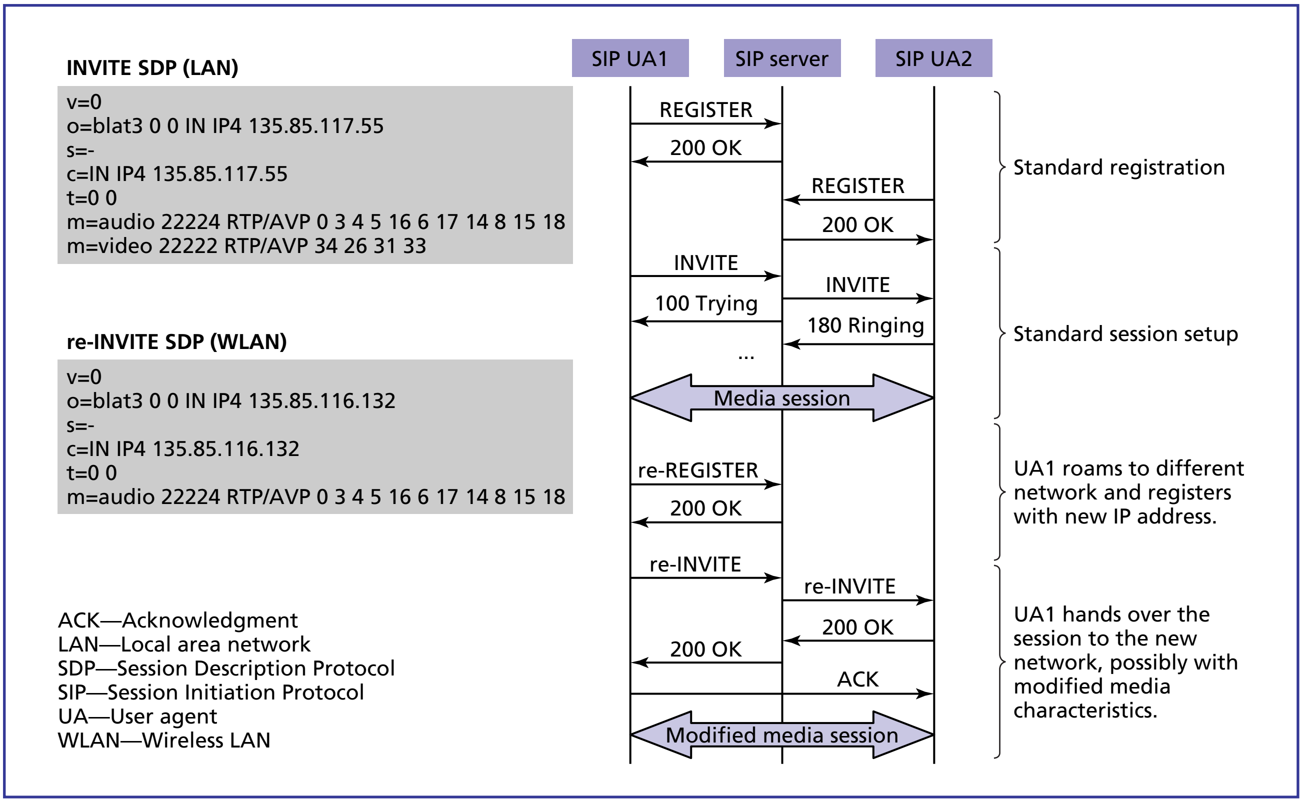


Figure 3.
SIP mobility management with re-REGISTER and re-INVITE messages.

component. The quality of the information provided by the mobility manager and by the policy functions inside the decision maker determines the quality of the decisions that can be made. Therefore, we have defined a number of network attributes to enhance roaming behavior.

Currently, two distinct cases of mobility are distinguished: one in which MIP is active and one in which it is not. When MIP is active, it takes care of terminal mobility, so network changes do not affect the IP address of the application. However, these network changes could affect the network parameters, such as the type of network used by MIP. Because SIP is needed to provide the session control functions required to adapt the session to new network parameters, the decision maker must issue re-INVITE messages for all active sessions in the application that are affected by the changed network parameters.

When MIP is not active, SIP is used to perform both the terminal mobility and the session control functions, and the decision maker must ensure that, when there is a network change, there is also a change in the IP address. When change occurs, the decision maker must re-REGISTER the application at the SIP server with the new IP address. Then, re-INVITE messages must be sent to all active sessions. **Figure 5** shows a simplified state diagram of the decision maker in the SIP application after a network change event has been received. The decision maker makes all the decisions in the diagram, the input signals are the triggers from the external mobility manager, and the tasks are operations in the MX SIP communicator. As can be seen in the diagram, not every change in the network will result in a change to the session; for example, if a network parameter changes but it has no impact on a session, no action is taken. In addition, when MIP is active and the network type does not change (which is often the case), no action is taken.

It is also possible to have the decision maker separate the data and control paths if there are multiple

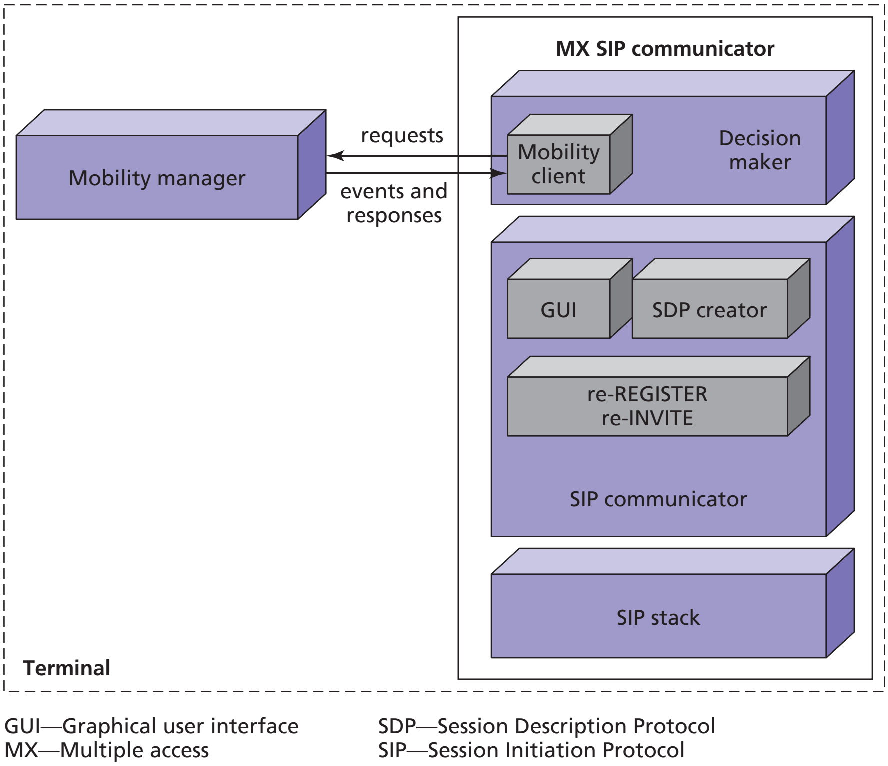


Figure 4.
SIP client architecture.

active network interfaces available. For example, control data could be sent over a GPRS network and multimedia data over a WLAN.

SDP Creator

When a non-mobile terminal is connected to a network, the SDP is determined by the requirements of the application, the capabilities of the terminal and the network, and user preferences, and it remains fixed. However, when a terminal roams between different networks, the SDP has to be modified to reflect such things as change of bandwidth and delay. The SDP creator is able to create an SDP profile based on the preferences of the user, the characteristics of the terminal, and the type of network to which the terminal is connected. This profile allows the application to adjust the amount of data that is sent and received by, for example, dropping the video part of a session on a low-bandwidth network. Examples of SDPs appear in Figure 3.

The re-INVITE and re-REGISTER Messages

Most applications implement only the registration and session-initiation parts of SIP and do not support mobility using the re-REGISTER and/or re-INVITE messages. The re-INVITE message has the same format as a standard INVITE message, but it can have different parameters and it is used during a session. Further, the response to a re-INVITE message differs from the response to an INVITE message, because there is already a session and the purpose of the re-INVITE message is to adapt the characteristics of that session. The decision maker triggers the re-INVITE message after a network change and sends it to the SIP stack.

The re-REGISTER message is the same as the standard REGISTER message; the REGISTER message already supports a retransmission mechanism as part of the registration expiration process. The application sends the re-REGISTER message, which contains the

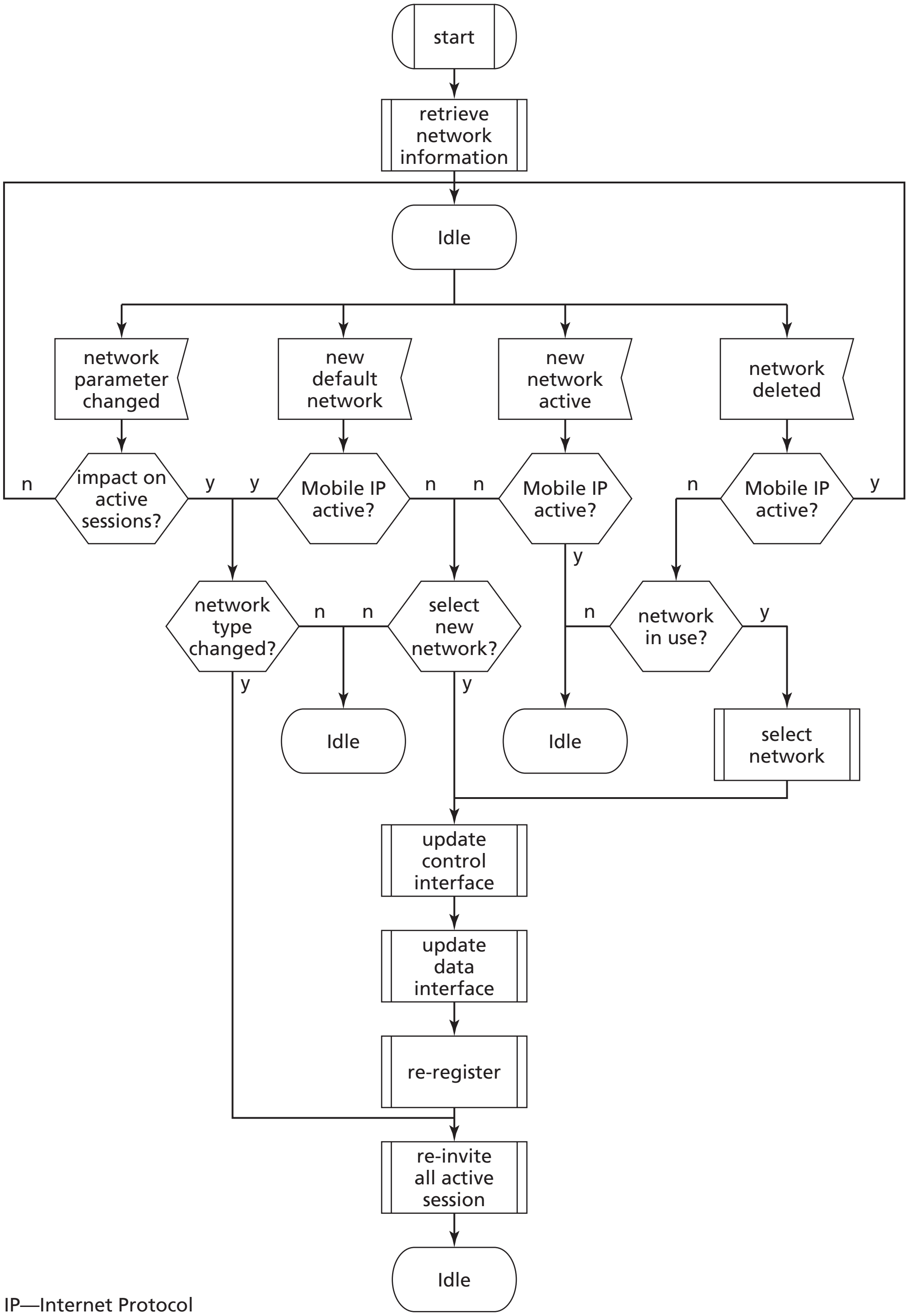


Figure 5.
State diagram of decision maker.

new network address, as soon as the mobility manager switches to a new network.

Mobility Manager

The mobility manager [7] is external to the SIP application, but it runs on the same terminal. It manages all network changes, such as network detection (i.e., the availability of new networks, the disappearance of networks) and automatic network authentication, and it also maintains the routing table. The mobility manager ensures that users always have the best possible connection that takes into account their preferences. The mobility manager keeps up-to-date information on all available networks and their characteristics and can switch networks when, for instance, a terminal goes out of reach of one network and comes within reach of another. MIP (when active) is also part of the mobility manager. Although the mobility manager is capable of maintaining network connectivity, it has no application-level knowledge; therefore, it is not aware of the impact of a network change on existing sessions. Such awareness is the domain of SIP and of the decision maker inside the SIP application.

Mobility Client

The mobility client communicates with the mobility manager and retrieves or receives network status information. (We have chosen to format network status information in XML messages.) There are two types of messages: event messages, which are sent toward the SIP client, and request or command messages, which originate at the client. The messages and events defined in **Table I** support mobility in an application and keep an application up-to-date on the status of the available networks.

Figure 6 illustrates the flow of messages that occurs when a new network is detected and the decision maker decides to switch over to it. The figure illustrates the most complex case, in which the IP addresses of both the data channel and the control channel must change and, subsequently, the multimedia streams must be rerouted to the new interface.

Prototype Implementation

The architecture described above has been implemented in a prototype. This prototype can be used to demonstrate a seamless roaming solution across

Table I. Messages between mobility manager and a mobile-aware application.

Name	Type	Purpose
Register for event†	Request	Express interest in a certain type of event from the mobility manager.
Set network default	Request	Initiate selection of another network as the default network.
Get info†	Request	Retrieve detailed network information from the mobility manager.
Network available	Event	Indication that a new network has been found. The network may be inactive and may not be connected yet.
New network active†	Event	Indication that a network has been connected (e.g., LAN or dialup) or is active (e.g., WLAN).
Network deleted†	Event	Indication that a network has been deleted.
Network parameter changed†	Event	Indication that a network parameter (e.g., status, bandwidth [WLAN], or cost of usage) has changed.
New default network†	Event	Indication that a new default network has been selected.
Network connectivity lost	Event	Indication that no network is available.

LAN—Local area network
MX—Multiple access
SIP—Session Initiation Protocol
WLAN—Wireless local area network

†Because this table presents a generic interface for all kinds of applications, not all requests and events are used by all applications. For example, the MX SIP communicator described in this paper only needs the requests and events marked with a dagger.

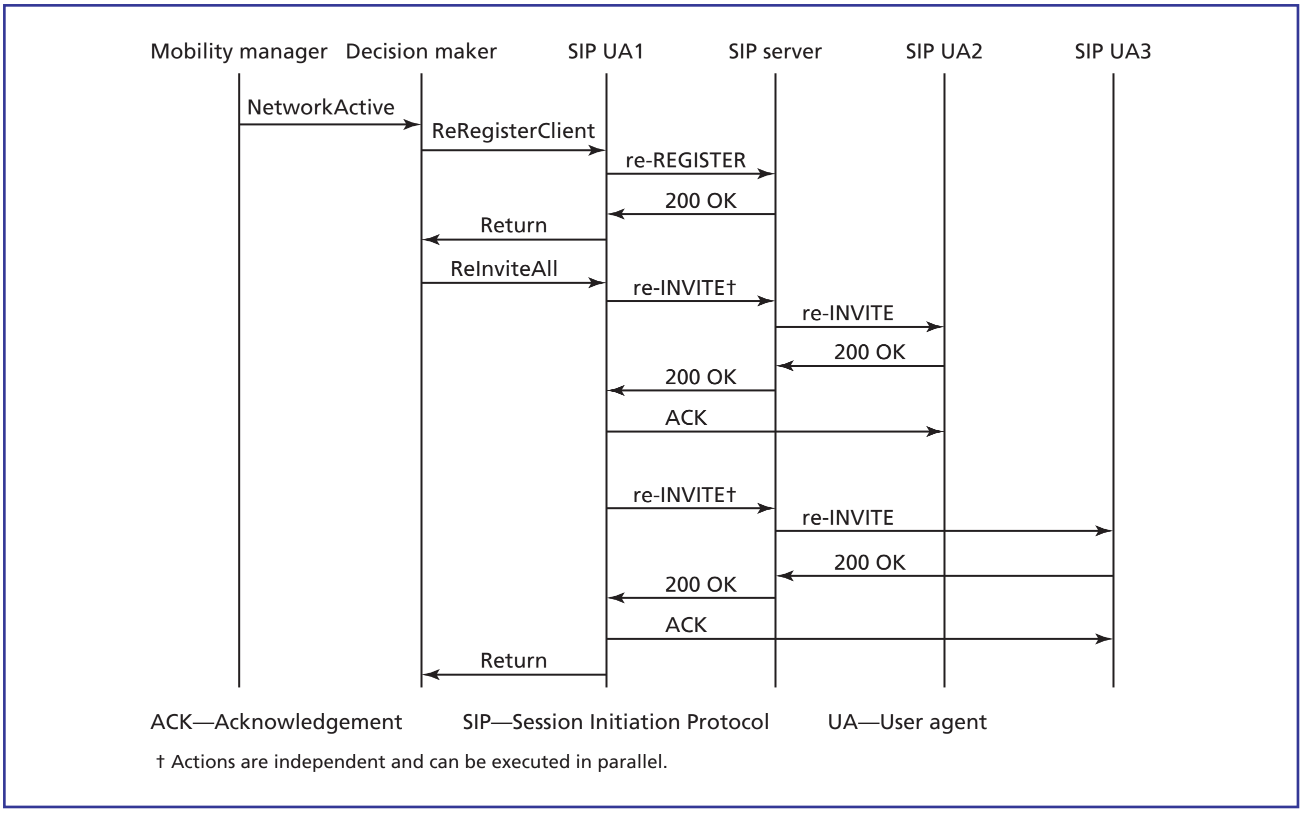


Figure 6.
Message-sequence diagram for SIP roaming.

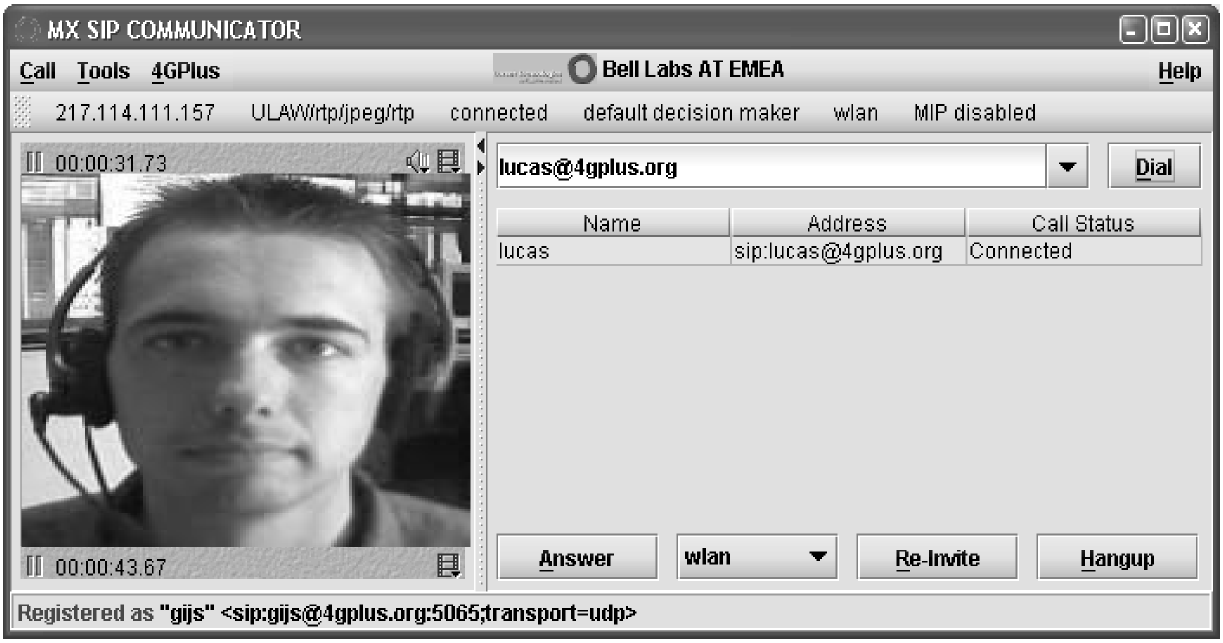
heterogeneous fixed and mobile network technologies and it adapts sessions based on changing network characteristics.

We have taken a Java*-based SIP client as the starting point for the prototype implementation. The SIP communicator is an open source SIP client application based on a public-domain Java application programming interface (API) for integrated networks (JAIN*) National Institute of Standards and Technology (NIST) SIP stack [12]. It runs on a Microsoft Windows* operating system and uses the Java media framework (JMF) for media streams. The JAIN APIs bring service portability, convergence, and secure network access to telephony and data networks and make rapid prototyping possible. We have extended the SIP client application by adding re-INVITE and re-REGISTER functionality and have combined it with a mobility manager to create the MX SIP communicator. **Figure 7** shows the MX SIP communicator with a registered client and an active session with audio and video.

The original SIP communicator already had a mechanism to perform registration and re-registration (after the expiration of a registration); the code was reused to implement the re-REGISTER. Re-registration is performed to communicate address changes to the SIP server. After a change in the network address, we initialize the SIP stack with the new properties.

For the implementation of the re-INVITE, we used standard INVITE functionality and added support for using information from the existing call and providing storage for the data that needs to be changed during a re-INVITE.

The decision maker was written from scratch and implements the state diagram in Figure 5. The decision maker caches the network information it retrieves from the mobility manager. The decision maker is policy driven in deciding when a network change will result in a re-REGISTER or re-INVITE. (For the prototype, the policy is to use both video and audio whenever possible.) This means that a re-INVITE may be initiated even



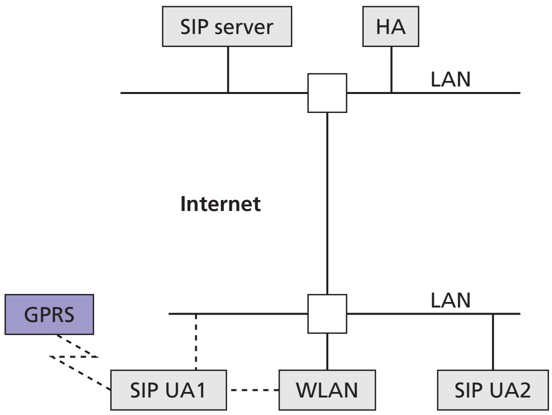
MX—Multiple access SIP—Session Initiation Protocol

Figure 7.
Snapshot of the MX SIP communicator.

when it is not strictly necessary. The decision maker is implemented in a modular fashion so that it can be enhanced with new functionality when needed.

The SDP creator constructs SDPs based on values in a configuration file and on the network type, which it receives from the decision maker. Depending on the SDP values, the MX SIP communicator modifies the session by adding or removing audio and/or video components.

For the mobility manager, we used the Lucent SmartClient [9]. This mobility manager supports the messages defined in Table I so that it can provide the information needed to implement the prototype.



GPRS—General Packet Radio Service
HA—Home agent
LAN—Local area network
SIP—Session Initiation Protocol
SIP UA—SIP user agent
WLAN—Wireless LAN

Figure 8.
Experimental test setup.

Roaming Scenarios

Figure 8 shows the experimental setup used to test the functionality of the SIP client and to measure its performance. This section describes several roaming scenarios for SIP sessions with and without MIP.

MIP for Interface Changes and SIP for Session Changes

In this scenario UA1 (on a WLAN) and UA2 (on a LAN) start a media session; then UA1 moves from a WLAN to GPRS. The session is handed over and the characteristics of the media stream are changed. In this scenario, MIP handles terminal mobility and SIP handles session control. MIP is active on the terminal of UA1 and it has obtained an IP address from the HA. Both WLAN and GPRS networks are available, and WLAN is selected. The scenario starts after both UAs are registered on the SIP server and an active audio session with an expensive (i.e., high-bandwidth and low-latency) codec has been established.

The first step in the scenario is the detection of a degraded WLAN signal by the mobility manager. The mobility manager selects GPRS as the new default network and sends a *Network parameter changed* event to the decision maker. The decision maker knows that a re-REGISTER message is not needed because MIP is active, so it triggers only a re-INVITE message for all active sessions. The scenario concludes with the following actions:

- UA1 sends a re-INVITE message over GPRS to the SIP server. (Note that the SDP contains the same MIP address and updated, less expensive [i.e., lower bandwidth] audio codecs.)
- UA2 accepts and sends an OK message to the SIP server, which forwards it to UA1.
- After receiving the OK message, UA1 sends an ACK message directly to UA2.
- After receiving the OK message, UA1 adapts Real-Time Transport Protocol (RTP) streams.
- After receiving the ACK message, UA2 adapts RTP streams.

As a result, the session is handed over with an interruption—not because of an IP stack change, but because different codecs are used—and with different media characteristics (i.e., audio only).

SIP for Interface and Session Changes

The main difference between this scenario and the previous one is that, in this scenario, MIP is not active on the terminal of UA1. In this scenario, two variations are possible.

In the first variation, UA1 (on a WLAN) and UA2 (on a LAN) start a media session; then UA1 moves from a WLAN to GPRS. WLAN and GPRS networks are both available, WLAN is selected, and the terminal of UA1 has two active network interfaces with different IP addresses. The scenario starts after UA1 and UA2 are both registered on the SIP server and an active multimedia session with both audio and video has been established. The trigger for a network handover could be the detection of a WLAN signal loss. The mobility manager reacts by selecting GPRS as the new default network and sending the *New default network* event to the decision maker. Then, the application sends re-REGISTER and re-INVITE messages as follows:

- The decision maker triggers a re-REGISTER message.
- UA1 sends a re-REGISTER message over GPRS to the SIP server, using the new IP address.
- The SIP server sends a 200 OK message to UA1.
- The decision maker triggers a re-INVITE message for all active sessions.
- UA1 sends a re-INVITE message over GPRS to the SIP server. (Note that the SDP contains a new public GPRS IP address and updated media characteristics.)
- UA2 accepts and sends an OK message to the SIP server, which forwards it to UA1.
- After receiving the OK message, UA1 sends an ACK message directly to UA2. (Note that there are no intermediate messages like 100 Trying or 180 Ringing.)
- After receiving the OK message, UA1 adapts RTP streams.
- After receiving an ACK message, UA2 adapts RTP streams.

As a result, the session is handed over with an interruption—because of the change to the IP stack and because other codecs are used—and with different media characteristics (i.e., audio only).

In the second variation, UA1 (on a WLAN) and UA2 (on a LAN) start a media session; then UA1 moves from WLAN to LAN. MIP is not active on the terminal of UA1. This scenario can only be executed if the terminal of UA1 has both a WLAN interface and a LAN interface and the LAN interface is not active. The scenario starts in the same way as the previous one—i.e., after UA1 and UA2 are both registered on the SIP server and an active multimedia session with both audio and video has been established. The detection of the availability of the LAN interface on the terminal of UA1 is the trigger that starts the network handover. The mobility manager now has a choice of two networks; it selects the best possible network for this scenario (i.e., a LAN) and sends the *New default network* event to the decision maker. The decision maker finishes the scenario with the same steps as in the previous scenario, except that the signaling now uses the LAN connection rather than the GPRS connection used in the previous scenario.

As a result, the session is handed over with an interruption—because of the change to the IP stack and because other codecs are used—and possibly with new media characteristics, using SIP messages only. The next section discusses the results of executing the above scenarios.

Experimental Results and Lessons Learned

The roaming scenarios described above have been used as the basis for performance measurements of the prototype implementation of the MX SIP communicator in combination with a SIP server. We selected a combination of network changes from the roaming scenarios above that we could repeat in a consistent way, both with and without MIP. In our test scenario, one mobile UA roamed from LAN to WLAN to GPRS to LAN and the other UA did not move. The clients were configured in such a way that only audio was available on the WLAN and the GPRS; both audio and video were available on the LAN. We executed the test scenario twice, once with and once without MIP. The SIP client used the User Datagram Protocol (UDP) for both data and signaling. During the execution of the scenario, the behavior of the client was observed. After the scenarios had been executed, the performance was calculated.

Performance Measurements

The session adaptation described in this paper is not transparent for the application, as is manifested by a short interruption of the session. The duration of this interruption consists of three components:

- The handover time (T_{ho}), which also includes the MIP handoff that is needed by the mobility manager to realize a network change. The handover time is completely independent of the SIP session control. During the handover time, the following steps are taken sequentially: network detected, network associated, network authenticated, IP address made available;
- The SIP-related delay (T_{sip}); and
- The time needed to adapt the RTP streams once the session has been adapted (T_{rtp}).

The total delay is the sum of the three parts (i.e., $T_{ho} + T_{sip} + T_{rtp}$). For this paper, we only measured the SIP-related delay; we did so by using the time stamps that are available in the MX SIP communicator. **Table II** shows the duration (T_{sip}) of the re-REGISTER and re-INVITE messages on the SIP message level. The re-REGISTER message time is measured from the sending of the first REGISTER message until the reception of the OK message. The re-INVITE message time is measured from the sending of the first INVITE message until the sending of the first ACK message.

The measurements were repeated a dozen times; they should be considered as experimental results for specific but equal environments set up to compare the effect of MIP on SIP session control. With a MIP client (with a collocated CoA and without a foreign agent), roaming is slower because of the additional layer (i.e., SIP and MIP). The measured values for

Table II. Measured duration of the re-REGISTER and re-INVITE messages.

	LAN (sec) re-REGISTER	LAN (sec) re-INVITE	WLAN (sec) re-REGISTER	WLAN (sec) re-INVITE	GPRS (sec) re-REGISTER	GPRS (sec) re-INVITE
Mobile IP/SIP	n/a	2.5...5.6	n/a	2.5...12.3	n/a	2.7...13.6
SIP only	0.3	0.9	0.3	1.0	1.9	2.2

GPRS—General Packet Radio Service
IP—Internet Protocol
LAN—Local area network
SIP—Session Initiation Protocol
WLAN—Wireless local area network

SIP-only are easily reproducible, but measurement results for SIP over MIP have such a large deviation that only the minimum and maximum values are given. One reason for the long delay (T_{sip}) is that the SIP control messages over UDP are retransmitted with a logarithmic backoff time if they are not acknowledged in time.

SIP Interoperability Issues

In the SIP specification, both optional and required behavior are described. When implementers choose different options for their SIP entities or when they overlook some requirements, interoperability problems can occur. Although this is a general protocol issue, we must bear it in mind whenever we integrate applications. An interoperability issue that we encountered in our testing is that a signaling port number is optional in a SIP request; when it is not supplied, the server assumes the default port (i.e., 5060). Suppose that a UA adds the default port number, but the SIP server does not forward the default port number to the destination UA, because it believes the port number is optional and it does not take into account the fact that proxies should retain the port number. Problems will then arise during the processing of the ACK message, because an ACK message with a port number is not considered the same as an ACK message without a port number. To reduce the number of interoperability issues, we recommend minimizing the number of implementation options in future standards.

SIP and MIP

During our experiments with roaming and session handover, the collaboration of MIP and SIP was studied in detail. Without the presence of SIP sessions, MIP provides a good mechanism for maintaining connectivity between a mobile terminal and the network to which it is connected. But when a SIP application begins interacting with the same network, communication becomes complicated because, while MIP tries to hide network changes from sessions, SIP tries to optimize sessions by re-inviting the other parties with session parameters that are adapted to the new network characteristics. Thus, if the mobile terminal moves to a network with a lower bandwidth, there is no guarantee that the SIP control messages will reach their destination, because the data flow that is already established may consume all the available bandwidth, leaving no room for SIP control messages. In such a case, the SIP re-INVITE will either fail or suffer a large delay. The loss of messages can be avoided by introducing a quality-of-service mechanism, but there will still be an additional delay, because the bandwidth must be guaranteed before the signaling can continue.

We experienced this problem in our testing with the GPRS network. The assigned bandwidth in the GPRS network varies, and it can be very low compared to that of a LAN or a WLAN. Because of the effect of this difference on signaling messages (e.g., SIP messages), switching from a WLAN that allowed high bandwidth to GPRS resulted more than once in message time-outs and terminated sessions.

Future Work

In this paper, we have described the role of SIP in mobility management and session control. However, not all aspects of this role are discussed in this paper; in particular, we have not taken into account the security implications of the SIP-only solution. This solution will not work in its present form in a fully compliant IMS network, because there is a tight security relationship between the UA and the proxy call session control function (P-CSCF) that will not survive a change in IP address. When such a change occurs, re-authentication is necessary during a re-INVITE. These issues are currently being studied in a 4GPLUS [2] follow-up project. However, it should be restated that, in pre-IMS architectures and non-IMS architectures, the SIP-only method is a valid method for session control.

Recommendations

After evaluating the results of our experiments, we propose a couple of changes to enhance the mobility of SIP-based applications. Enhancements can be made both to the interface between the mobility manager and the SIP client and to the SIP client itself.

Notify Network Changes in Advance

The introduction of a new *About to switch* event would circumvent the problems described above. The mobility manager would generate this event and would send it just before it switches to a new network. Then the following scenario could be implemented:

1. The mobility manager sends an *About to switch* event to the decision maker.
2. If the current bandwidth demand of the session exceeds the bandwidth available on the new network, the decision maker sends a re-INVITE message on the old network.
3. The mobility manager is notified that the application is ready to switch to a new network.
4. The mobility manager performs the switch.
5. The decision maker sends re-REGISTER and re-INVITE messages on the new network to update the contact address.

Separation of Data and Control

Our other recommendation, which is closely related to the previous one, is to separate the data flow from the control flow in those cases in which more than one network (e.g., GPRS and a WLAN) is available for the terminal. Control messages could be sent over the GPRS network and the data over the WLAN. This would prevent the loss of SIP control messages due to congestion. Separation of data and control can also be realized by using networks that support quality of service and can provide guaranteed bandwidth. To give priority to SIP control messages and MIP binding updates, these messages can be given preference in the traffic shaper that exists in modern operating systems (e.g., Linux,* FreeBSD, and Microsoft Windows 2000/XP). Whether or not this is possible depends on the MIP implementation, because not all implementations allow a secondary interface to be active.

Conclusions

In this paper, we have presented an architecture that combines mobility management and session control for SIP-based applications. We have discussed the implementation of a flexible prototype that uses MIP and SIP for mobility management and SIP for session control. We have evaluated both the collaboration of SIP with MIP and the SIP-only solution for mobility management and session control. Our experiments show that, although MIP outperforms SIP for mobility management [3, 14], the combination of MIP and SIP for mobility management and session control performs much worse than SIP-only mobility management. We therefore recommend using the SIP-only solution for mobility management and session control of multimedia applications.

During the process of implementing the prototype, we also encountered a number of other issues. One issue was an interoperability issue: although the message format and the exchange of SIP messages are RFC3261-compliant, the communication between two SIP endpoints is not guaranteed. Another issue was the flooding of the network that occurs when switching from a high-bandwidth to a low-bandwidth network using MIP. As a solution to this flooding, we propose sending the appropriate re-INVITE message before and after the handover takes place.

Acknowledgments

The work described in this paper is part of the research project called 4GPLUS [2]. This project is supported in part by the Dutch Freeband Impulse Programme on Telecommunication Applications [10] and is co-funded by the Dutch Ministry of Economic Affairs. The authors would like to thank all 4GPLUS members who contributed to this paper, either by providing input to it or by reviewing it.

References

[1] 3rd Generation Partnership Project, “IP Multimedia Subsystem (IMS); Stage 2 (Release 5),” TS 23.228, Sept. 2003, <http://www.3gpp.org>.

[2] N. Banerjee, W. Wu, S. K. Das, S. Dawkins, and J. Pathak, “Mobility Support in Wireless Internet,” IEEE Wireless Commun., 10:5 (2003), 54–61.

[3] M. S. Bargh, D. Bijwaard, H. Zandbelt, E. Meeuwissen, and A. Peddemors, “Mobility Management in Beyond 3G Environments,” Proc. of Wireless World Research Forum 9, WWRF9, (Zurich, Switz., 2003).

[4] M. S. Bargh, H. Zandbelt, and A. Peddemors, "Managing Mobility in 4G Environments with Federating Service Platforms (an Overview)," Proc. of EVOLUTE Workshop, Evolute '03, (Guildford, UK., 2003), A-6, 1–6.
[5] T. Dagiuklas, D. Gatzounas, D. Theofilatos, D. Sisalem, S. Rupp, R. Velentzas, R. Tafazolli, C. Politis, S. Grilli, V. Kollias, and A. Marinidis, "Seamless Multimedia Services Over All-IP Based Infrastructures: The EVOLUTE Approach," IST Mobile and Wireless Telecommun. Summit, (Thessaloniki, Gr., 2002), 494–498.
[6] R. van Eijk, J. Brok, J. van Bemmel, and B. Busropan, "Access Network Selection in a 4G Environment and the Roles of Terminal and Service Platform," Proc. of Wireless World Research Forum 10, WWRF 10, (New York, NY, 2003).
[7] Freeband Impulse Programme on Telecommunication Applications Website, <http://www.freeband.nl/>.
[8] JAIN API Specification, <http://java.sun.com/products/jain/api_specs.html>.
[9] T. T. Kwon, M. Gerla, and S. Das, "Mobility Management for VoIP Service: Mobile IP vs. SIP," IEEE Wireless Commun., 9:5 (2002), 66–75.
[10] C. Perkins (ed.), "IP Mobility Support for IPv4," IETF RFC 3344, Aug. 2002, <http://www.ietf.org/rfc/rfc3344.txt>.
[11] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002, <http://www.ietf.org/rfc/rfc3261.txt>.
[12] H. Schulzrinne and E. Wedlund, "Application-Layer Mobility Using SIP," Mobile Computing and Commun. Review, 4:3 (2000), 47–57.
[13] SmartClient product page, <http://www.lucent.nl/bell-labs/>.
[14] E. Wedlund and H. Schulzrinne, "Mobility Support Using SIP," Proc. ACM WoWMoM '99, (Seattle, WA, 1999), 76–82.

(Manuscript approved July 2004)

WILLEM A. ROMIJN is a senior system designer in the Bell Labs Advanced Technologies EMEA Department at Lucent Technologies in Hilversum, The Netherlands. He received his B.Sc. degree in information technology from the Hogere Technische School in Dordrecht, The Netherlands. He has worked on several European research projects. His fields of expertise include object-oriented software design, software component frameworks, network management, and multimedia signaling protocols. Currently, he is participating in the 4GPLUS research project for which he has contributed to the SIP client prototype and is working on integrating a SIP server with NavisRadius.

DIRK-JAAP PLAS is a member of technical staff in the Bell Labs Advanced Technologies EMEA Department at Lucent Technologies in Enschede, The Netherlands. He received an M.Sc. degree in computer science from the University of Groningen, The Netherlands, working on the management of CORBA-based systems. Based on this experience, he developed expertise in the area of open service architectures, such as the Open Service Access (OSA) specification defined by 3GPP. He is interested in next-generation service and network architectures and will expects to continue his work in this area. Currently, Mr. Plas is participating in the 4GPLUS research project, which focuses on mobility management and multimedia session control; he leads the work package on multimedia session control.

DENNIS BIJWAARD is a member of technical staff in the Bell Labs Advanced Technologies EMEA Department at Lucent Technologies in The Netherlands. He received his M.Sc. degree from the University of Twente, The Netherlands, working on language parsing and function approximation using artificial intelligence techniques. He now works on next-generation networks and the visualization of software and protocol dynamics. He received a central Bell Labs Teamwork Award for his work as a member of the 3G Services Platforms Team. His main interests are software reverse engineering, distributed computing, quality of service, and artificial intelligence.

ERIK MEEUWISSEN is a member of technical staff at Bell Labs in Hilversum, The Netherlands. He received M.S. and Ph.D. degrees in electrical engineering from Eindhoven University of Technology, The Netherlands. His professional interests include theoretical and applied aspects of communication networks and services. He is currently active in two Dutch collaborative research projects. In 4GPLUS, focusing on a service platform on top of heterogeneous networks, he is technically involved in multimedia session control and mobility management. In EQUANET, he serves as *overall project leader of a Dutch consortium working on end-to-end quality of service in next-generation networks. Dr. Meeuwissen is a member of the IEEE and the Dutch Electronics and Radio Society (NERG).*

GIJS VAN OOIJEN is a system engineer in the Bell Labs Advanced Technologies EMEA Department at Lucent Technologies in Hilversum, The Netherlands. He has a bachelor's degree in electrical engineering from the Hogere Technische School in Utrecht, The Netherlands. He has participated in projects funded by the European Commission for many years. His fields of expertise include embedded software, management systems, and network processors. He is currently involved in 4GPLUS, working on SIP-based roaming. ◆