Cross-Layer Mobility with SIP and MIPv6

RUI PRIOR[†], *SUSANA SARGENTO*[‡]
Institute of Telecommunications
[†]*University of Porto,* [‡]*University of Aveiro*
PORTUGAL
rprior@dcc.fc.up.pt, ssargento@det.ua.pt

Abstract

Terminal mobility may be handled at different layers. Though the MIPv6 protocol is the strongest candidate for handling mobility in the next generation networks, mobility management facilities are also provided by SIP, the most widely deployed and researched protocol for session control. When jointly used, this duplication of functions leads to inefficiencies in session setup signaling, particularly if coupled with end-to-end resource reservation for the media flows. This paper analyses these inefficiencies and proposes an integrated approach that minimizes the session setup delay. The gains of the proposal are demonstrated both by a delay analysis and by simulation results.

Keywords: SIP, MIPv6, mobility, cross-layer design

1. Introduction

Current research, standardization and market trends indicate that future telecommunication systems will be IP-based, representing a convergence of actual networks, services and applications onto a single infrastructure. This convergence requires the integrated support of different access technologies, mostly wireless. Moreover, users must be allowed to move freely without disruption of their ongoing sessions, even when the movement leads to a change in the access technology. While the use of IPv6 with mobility support [1] as a convergence layer greatly simplifies this process, service provisioning with the necessary quality and seamless mobility in such heterogeneous scenario is still a heavily researched topic.

The protocol that will most likely be used for the initiation and control of multimedia sessions is SIP (Session Initiation Protocol) [2][3], which has been adopted by the principal 3G standardization organizations and forums, like 3GPP and 3GPP2. Though SIP itself may be used for mobility management [4], this function is better handled at layer 3 by Mobile IPv6 (MIPv6) [1], even when SIP is used for session control, for several reasons: (1) applications need not worry about mid-session mobility unless serious changes in available resources force a session renegotiation, e.g., to a lower bitrate codec; (2) layer 3 mobility support must be in place to support non-SIP sessions (HTTP, FTP, etc.), and uniform mobility management is desirable for robustness and flexibility; and (3) seamless mobility may be achieved through the use of MIPv6 extensions like Fast Handovers [5].

The joint use of SIP and MIPv6, however, leads to some inefficiency issues in pre-session mobility, due to each protocol's unawareness of the other's mobility management capabilities. The issues are even worse when end-to-end resource reservation must be performed to ensure appropriate Quality of Service (QoS) to the session, requiring knowledge of the points of attachment of both terminals; however, end-to-end resource reservation is necessary to provide communication services with the same level of quality users have come to expect from the telephone network. This inefficiency may lead to a significant delay in session setup, especially in the presence of some packet loss (not uncommon in wireless links) and of large round-trip times (RTT). This paper proposes a scheme for the minimization of these delays based on simple procedures and cross-layer interactions, making SIP aware of the terminal's location, that is, the Care-of Address (CoA). The efficient joint use of SIP and MIPv6 will enable the seamless support of mobile multimedia applications.

The paper is organized as follows. Next section describes some previous work on the integration of SIP and MIP. Section 3 gives an overview of the target architecture for these optimizations. Sections 4 and 5 contain an analysis of the problem and the proposal of the solution, respectively. Section 6 describes the SIP registration procedures. Section 7 discusses the relation between the proposed optimizations and the dormancy/paging support for energy saving. An analytical comparison of the standard and optimized procedures is presented in section 8, and section 9 discusses simulation results of both. Finally, section 10 contains the main conclusions of the paper.

2. Related Work

Different degrees of integration of SIP and MIP (v4 or v6) have been proposed by several authors. Jung et al. [3] proposed the use of integrated mobility agents for SIP and MIP(v4). Some of the MIP functions (like binding refreshments) are transposed to SIP, and mobility is communicated to the correspondent nodes (CNs) via re-INVITE requests. This approach imposes different handover procedures for SIP and non-SIP sessions (UDP or TCP), and the security issues of establishing bindings with CNs via SIP were not addressed.

Politis et al. [6] proposed a hybrid SIP/MIP(v4) scheme for inter-domain mobility. Their approach avoids the IP-in-IP MIP encapsulation for SIP sessions, but not for non-SIP ones, for which the handovers are managed by MIP. Their work mostly concerns mid-session mobility which, in our case, is handled by a modified MIPv6 with Fast Handover extensions. Moreover, the encapsulation problem is mitigated in MIPv6 by the use of routing optimization.

Wang et al. [7][8] proposed an integrated SIP-MIP mobility management architecture, where MIP and SIP agents are broken down into functional blocks and then integrated, without duplication, into unified Home and Foreign Mobility Servers (HMS/FMS), thus avoiding redundancy. Their proposal mostly intends to solve the problems associated with different types of mid-session mobility that in our case are addressed at the layer 3. Moreover, their architecture is different from ours in that it requires one FMS per (access) network.

None of these proposals addresses the issues with the integration of end-to-end resource reservation with session signaling.

3. Architecture Overview

This section contains a simplified overview of the architecture considered for these optimizations. The routing infrastructure is based on IPv6, used as a convergence layer, and mobility support is provided at layer 3 by MIPv6 with Fast Handover (FHO) extensions [5].

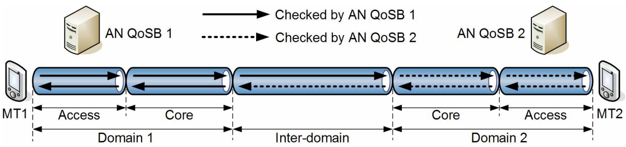


Figure 1. Admission control with terminals in different domains

The network is divided into administrative domains, each consisting on a number of access networks (ANs), possibly with different access technologies, interconnected by a core network (CN). One of the key components is the QoS Broker at the AN, responsible for controlling the admission of flows and the handovers, interacting with the Access Routers (ARs) in a PDP-PEP (Policy Decision / Enforcement Point) relationship. The QoS Brokers have information on available resources not only for the AN they control, but also in the core of their domain and the transmission direction of the inter-domain path segment. Therefore, the combined admission control performed at the caller and callee sides may ensure that enough resources are available along the end-to-end path to provide adequate QoS to the flow, as illustrated in figure 1.

In the CN there is a Service Provisioning Platform (SPP) containing the building blocks for creating services and applications. Among other components, the SPP contains a Multimedia Service Platform (MMSP), consisting of a broker and proxy servers, responsible for the provision and control of multimedia services. Though this platform is flexible enough to be adapted to different signaling protocols, the current implementation is based on SIP. Please refer to [9] for more information on the QoS subsystem of the architecture.

In order to establish a reservation for a flow ensuring that enough resources are available along the end-to-end path, admission control needs to take into account the available resources in the complete path, including the access, core and inter-domain path segments. To this end, each mobile terminal must be aware of its correspondent's physical location which, in IP terms, corresponds to its CoA. SIP's unawareness of pre-session MIPv6 mobility, as will be seen in the next section, is one of the sources of inefficiency in session initiation signaling. Mid-session mobility, on the other hand, is handled by MIPv6 with FHO, and does not require intervention of SIP. It is worth noting that the use of FHO allows for seamless handovers if mid-session mobility is performed at layer 3, which is not possible with SIP mobility — while seamless SIP mobility may be achieved with multi-homing, this approach would require two network interfaces, adding to the cost and energy consumption (therefore, to a lower battery life) of the mobile terminals.

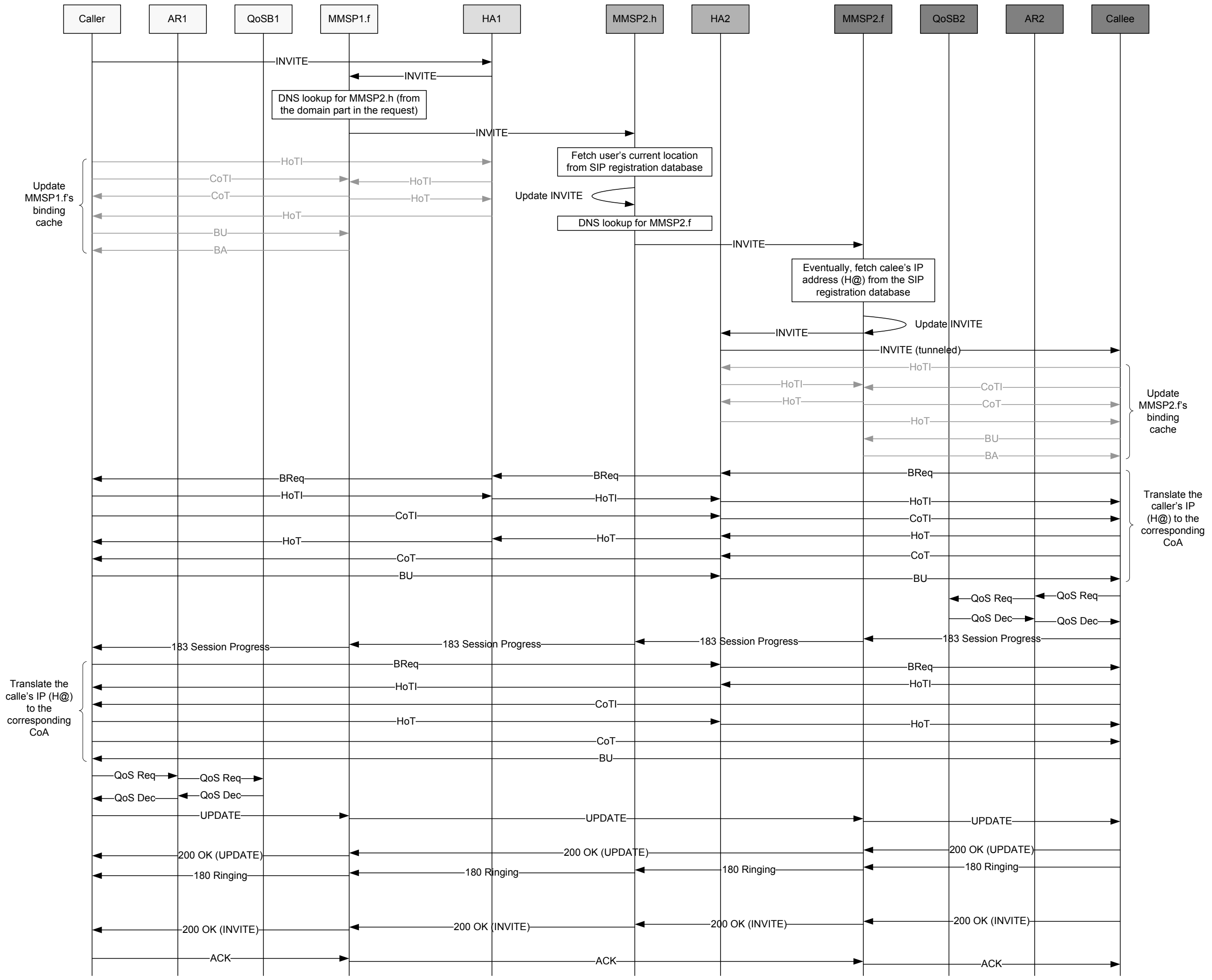


Figure 2. Inter-domain call without optimization (both terminals roaming)

4. Inefficiency of SIP with MIPv6

In this section we analyze the inefficiencies of the joint use of SIP and MIPv6, particularly in an environment where end-to-end resource reservations must be performed. The message sequence for initiating a call between two roaming terminals is illustrated in figure 2. "100 Trying" SIP responses and "PRACK" requests and responses have been omitted in the figure, since they are not in the critical path of signaling.

The sequence is initiated by the caller sending an INVITE with a message body containing an offer with the set of codecs supported by the caller and the corresponding ports (at the caller end only); this message is sent via the outbound proxy, MMSP1.f. If the binding cache of MMSP1.f is not up to date with the caller's current CoA, this INVITE is tunneled to its Home Agent (HA1), from where it is sent to MMSP1.f, introducing an additional delay corresponding to one RTT between the caller's home and foreign domains. The caller may then initiate a return routability procedure (RRP; grayed out since it is not in the critical path of signaling) to MMSP1.f so that further messages between them are optimally routed. If the Home Keygen Token has not expired since registration, only the Care-of Test Init/Care-of Test exchange is necessary, but otherwise a full RRP must be performed. Notice that mobility-unaware applications use Home Addresses (HoA) as endpoints in order for layer-3 mobility to be transparent.

When the INVITE request arrives at MMSP1.f, it must find out the proxy responsible for the callee to send the INVITE. To this end, a DNS lookup is performed, involving a round-trip to a root DNS server, another one to a top level DNS server and one or two[1] to the home domain of the callee, unless the entries are already cached. On receiving the

[1] At least an SRV lookup; however it is usually preceded by a NAPTR lookup.

INVITE, MMSP2.h looks up the registration database and finds out that the user (callee) is roaming; DNS lookups are performed to find out the proxy for the foreign (visited) domain. Notice that service authorization is mandatory, therefore MMSP2.h cannot send the INVITE directly to the callee – packet filtering mechanisms would drop it.

MMSP2.f receives the INVITE and fetches the callee's IP address from its registration database (for the sake of simplicity, we assume that the callee has registered itself with the IP address rather than a hostname). Since regular SIP is not layer-3-mobility-aware, this IP address is a HoA; therefore, the message must go to the callee's home agent (HA), from where it is tunneled to the callee.

When the callee receives the INVITE, it builds a list of the common codecs. In possession of the IPs and ports at both ends (the caller and itself), it may request resources to/from the caller (QoS Req). However, if resources are reserved for more than the wireless link, as in our case, the reservation must be made according to the physical points of attachment of the endpoints, that is, their CoAs. The callee knows its own CoA, but not the caller's, therefore a Binding Request[2] (BReq) must be issued to the caller, which will trigger a return routability procedure and a Binding Update (BU) from the caller to the callee, adding two RTTs between them. Since all of these messages must go through the HA of the callee (the caller has no binding for the callee yet) and some of them (BReq, HoTI and HoT) through the HA of the caller, this translates in 11 inter-domain traversals (considering that HoTI/HoT, and not CoTI/CoT, are in the critical path of signaling, as is most common).

The callee also initiates a return routability procedure and binding update to MMSP2.f, so that future messages need not be tunneled; however, the 183 Session Progress response must still go through the HA (otherwise MMSP2.f would drop it, since it has no binding for the callee).

When the caller receives the 183 Session Progress, it knows its own CoA, but not the callee's; therefore it must send a BReq to the callee (symmetrical of the previously mentioned procedure). Two additional RTTs are, therefore, added (corresponding to 7 inter-domain traversals, not 11 as the previous one, since the callee already has a binding for the caller). Only now the HoA and CoA of the callee are known at the caller side, therefore only now a fully-formed QoS request may be performed at this side. As the amount of available resources may be less than what was reserved at the callee side, an indication of the final codec configuration (counter-answer) must be sent in an UPDATE request in order to synchronize the reservations. Hopefully, by this time all the binding caches are updated, meaning that the UPDATE (as well as all further signaling) travels through optimal paths. The media packets will also use the optimized path, since each terminal has a binding for the media address of the other one (which is usually the same as the signaling address, except in some multi-homed terminals).

It is worth noting that many of the inefficiencies (namely, RTTs between foreign and home domains of the caller, of the callee, and between the foreign domains of both) are due to the SIP protocol's unawareness of layer-3 mobility, and to the need to perform resource reservations combined with this unawareness.

5. Optimizing the Use of SIP with MIPv6 for the Seamless Support of Mobile Multimedia Applications

The call initiation scenario illustrated in the previous section can be much improved by means of very simple procedures and cross-layer interactions. In this section we propose a series of optimizations that allow for a significant reduction in the session setup time, as will be shown in sections 8 and 9.

The first optimization consists on eliminating the need for the INVITE message between the caller and MMSP1.f to go through the HA. While this could be easily accomplished by having the terminal keep the MMSP's cache updated all the time, such approach would lead to a lot of unnecessary signaling, since most of the time it is not actually communicating, and would limit its ability to conserve energy using the paging features of the system. Therefore, we propose a different approach: using the CoA as source IP address of the packet containing the INVITE message. Notice that the INVITE message itself still uses the HoA. Responses to the INVITE will be delivered to the CoA since the proxy adds a *received* parameter with the source IP address of the packet to the *Via* header of a received request whenever the *sent by* parameter in the *Via* header does not match the source address in the IP packet header (in our case, the *Via* header contains the HoA and the source IP address is the CoA). The terminal may then perform the RRP, which is not on the critical path of signaling, and then maintain

[2] Notice that the above mentioned Binding Requests are not compliant with the Binding Refresh Requests defined in [11]. Please see appendix 1 for details.

MMSP1.f's binding cache updated for the whole duration of the call, so that future requests (PRACK, UPDATE, ACK, re-INVITEs, etc.) and their respective responses will always use the optimized path.

Sending the INVITE request in a packet where the source IP is the CoA does not pose security issues; in fact, even in the standard case, all requests sent after a binding is established between the caller and MMSP1.f (e.g., UPDATE) use the CoA as source IP address, only with an added Home Address option containing the HoA.

The goal of the second optimization is to eliminate the need for the INVITE message between MMSP2.f and the callee to go through the callee's HA. Contrary to the previous case, the message is not generated at the mobile terminal. In order to use the callee's CoA as the destination address, MMSP2.f must have knowledge of the mapping between the callee's HoA and CoA, as the HoA is the one used by the application layer. In order to provide this information, we introduce a cross-layer interaction at MMSP2.h: after retrieving the IP address (HoA) of the callee from the registration database, MMSP2.h queries the HA to find out the callee's current CoA. The URI in the request line is then changed to the IP address (HoA), as usual, but with tag containing the current CoA (e.g., "coa=FF1E:03AF::1") appended. Using the CoA from the tag in the request line as the destination IP address of the packet, MMSP2.f may send the INVITE directly to the callee. Notice that the use of the CoA tag by MMSP2.f for direct forwarding does not add any security issue to standard SIP, since the same would occur if MMSP2.h had placed the CoA directly in the request line of the forwarded INVITE.

The third optimization concerns the elimination of the DNS lookup at MMSP2.h when forwarding the INVITE request: if the registration for redirection includes the IP address of the inbound proxy where to forward an incoming INVITE (MMSP.f, in this case), no DNS lookup to find this proxy is necessary. The use of the Path header field described in [10] is recommended for conveying this information, while also providing a simple means of enforcing the traversal of an MMSP at the foreign domain, necessary to perform service authorization and filtering.

The fourth optimization is related to the need to perform network resource reservations concerning more than the wireless link. Since the requests are performed for a path-optimized flow, they must be performed between the physical locations (that is, the CoA) of both terminals. Once again, we rely on the transport of CoA information in the application signaling. However, since there is no guarantee that the media will use the same IP addresses as SIP signaling (particularly with multi-homed terminals), the CoA information used to this end is conveyed not in SIP, but in the protocol used for session negotiation. Inclusion of CoA information in Session Description Protocol (SDP) [11] and SDPng [12] is discussed in section 5.1.

The proposed extensions are backward-compatible in the sense that if a node (terminal or proxy) does not support them, the signaling sequence transparently falls back to a less optimized one without introducing incompatibilities.

One might argue that the inclusion of layer 3 mobility information in an application protocol such as SIP should not be done because it breaks the layering principle; it is worth noting, however, that not only does the standard SIP already include layer-3 information (IP addresses) in its headers, but also that cross-layer information would be required by any protocol with similar characteristics to SIP, namely regarding independence between the signaling and media interfaces. The use of layer-3 information in the application protocol implies that application-level gateways (ALGs) are necessary in transitional scenarios, where parts of the network "speak" IPv4 and other parts "speak" IPv6; this is true in any case where SIP is used (even without mobility), and our proposed optimizations require only minimal changes to the SIP-ALGs, to translate the additional fields where the CoA is conveyed (in addition to all the fields where standard SIP already conveys IP addresses).

5.1 Inclusion of CoA information in SDP(ng)

Media negotiation and configuration for the sessions is performed using either the SDP or its "new generation" successor, SDPng. The inclusion of CoA information requires simple extensions to these protocols.

In SDP, the IPv6 address of the media endpoint is conveyed in the *c*= field [11][13]. Since it is not possible to change the definition of this field without breaking backward compatibility, it contains the HoA only. For conveying CoA information we resort to a newly defined attribute, the standard way of extending SDP. This attribute, named *coa*, has a similar definition to that of the *c*= field:

```
a=coa: <network type> <addr type> <conn addr>
```

The use of the *coa* attribute is illustrated in the following example:

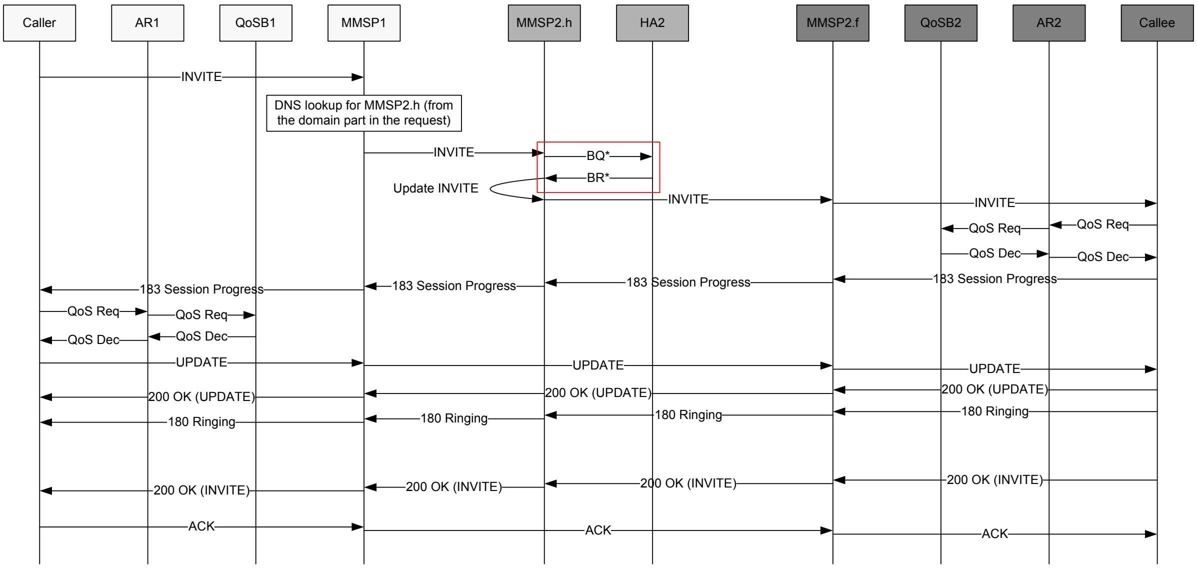


Figure 3. Optimized inter-domain call (both terminals roaming)

```
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
a=coa:IN IP6 FF1E:03AF::1
```

In SDPng, the HoA is conveyed by an *rtp:ip-addr* element. The CoA is conveyed by a newly defined element, *rtp:ip-coa*, as illustrated in the following example:

```
<rtp:udp name="rtp-cfg1" ref="rtp:rtpudpip6">
   <rtp:ip-addr>FF1E:03AD::7F2E:172A:1E24</rtp:ip-addr>
   <rtp:ip-coa>FF1E:03AF::1</rtp:ip-coa>
   <rtp:rtp-port>9456</rtp:rtp-port>
   <rtp:pt>1</rtp:pt>
</rtp:udp>
```

5.2 Optimized Initiation Sequence

Figure 3 shows the message sequence for an optimized multimedia call, with both terminals roaming (messages not in the critical path of signaling are omitted). Since the INVITE is sent with the CoA as source IP address, it goes directly to MMSP1.f. A DNS lookup is performed (there is no way to avoid it) and the message is forwarded to the callee's home proxy. MMSP2.h changes the request line from the URI to the HoA of the callee, adding a *coa* tag with the callee's current CoA (retrieved from HA2) to the request line of the INVITE. Although in the optimal case MMSP2.h and HA2 would be integrated, with the respective location databases merged, even if they are not, communication between them is fast and efficient, since they belong to the same domain and are located close to one another. This communication, however, requires new messages, Binding Query (BQ) and Binding Response (BR), since the standard Binding Refresh Request (BRR) and BU messages are exchanged with the MT, not the HA.

Since MMSP2.h has the IP address of the callee's outbound/inbound proxy, MMSP2.f, there is no need for a DNS lookup. Using the information from the *coa* tag on the request line, MMSP2.f is able to send the INVITE directly to the callee without the need to go through its HA. When this message arrives, the callee retrieves the caller's HoA and CoA from the SDP, and uses this information to request network resources.

After receiving the reservation response, the callee sends a 183 Session Progress response, containing an answer with the set of common codecs and their respective ports at both ends, to the caller. Information on the callee's CoA is included in the SDP; this information is used by the caller to perform the resource reservation on its side.

Usually, by the time the UPDATE is sent, both terminals have already established bindings with their respective proxies. However, the caller may include a *coa* tag with the CoA of the callee to the request line, lest MMSP2.f not have yet a binding for MT2: if this is the case, MMSP2.f uses the tag to send the request directly to the CoA, as it has previously done with the INVITE; otherwise, the tag is ignored. Notice that the UPDATE (and all further requests) does not traverse the home proxy of the callee, since only the local (foreign) proxies, which have responsibilities in service control, have added themselves to the *Record-Route* header of the initial INVITE.

On receipt of the UPDATE with the final configuration, the callee knows that the reservations have been successfully performed and that network resources are available, therefore it may start ringing.

It is worth noting that bindings must still be established between the terminals for the media sessions, meaning that the overhead of both solutions will be comparable (except for a few encapsulated packets in the standard signaling); however, these message exchanges are moved out of the critical path of signaling in the optimized case.

Mid-session mobility is handled exclusively at the layer 3 by MIPv6 (with Fast Handover extensions, in our case). SIP sessions are handled similarly to non-SIP ones based on UDP or TCP, and no re-INVITE message is sent unless a session renegotiation (e.g., for changing the codec or bit rate) is necessary. This way, it is possible to seamlessly support both multimedia and non-multimedia mobile applications in the same architecture.

6. SIP Registration

A user at home registers normally with the local (home) proxy, using the Address of Record[3] (AoR) in the *To* header and the IP address (HoA) in the *Contact* header. A roaming user must register itself with the foreign MMSP, since it will be performing service control, but also with the MMSP of its home domain for location purposes; therefore, MMSP2.f forwards the REGISTER request to MMSP2.h. In the standard case, the user registers itself with MMSP2.f as user@home.com, using the IP address as *Contact*; MMSP2.f changes the *Contact* to user%40home.com@foreign.com and forwards the registration request to MMSP1.h. In the optimized case, the user registers itself with MMSP2.f as user@home.com using the IP address as *Contact*, similarly to the standard case (fig. 4). However, MMSP2.f does not change the *Contact*: instead, it adds a Path header [10] with its own IP address, forcing incoming requests from MMSP1.h to traverse it. Though the Path header is an extension to the basic SIP protocol, it is a standard one.

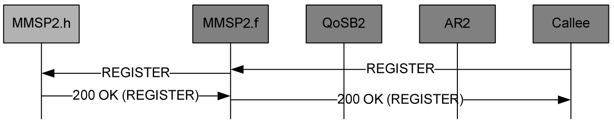


Figure 4. Registration procedure

It is worth noting that, contrary to the standard registration approach where any proxy of the foreign domain may be traversed by an incoming request, the optimized approach associates the terminal with a given proxy. However, being tied to a particular proxy does not degrade fault tolerance when compared to having pool of N proxies: if the probability of failure of each proxy is p_f, then the probability of failure of the one chosen by DNS lookup among a pool of N is the sum of the probability of choosing each of the proxies times the failure probability of that particular proxy, that is, $\sum_{i=1}^{N}(\frac{1}{N}p_f)=p_f$, the same as that of any individual proxy. Moreover, if the MT periodically contacts the proxy in order to check that it is “alive,” failure will be detected, and the MT may re-register with a different one; in this case, resilience is actually increased by sticking to a particular inbound proxy. On the other hand, proxy pooling could still be achieved by providing MMSP2.h an anycast address instead of the regular unicast address of MMSP2.f at registration time, using a method like the one proposed in [14].

7. Issues with Dormancy/Paging

Energy is a scarce resource in mobile terminals, particularly in the smaller and lighter-weighted ones. Therefore, any architecture where small and low-power devices are foreseeable must provide some mechanism for dormancy/power saving. In our architecture, support for dormancy is provided by the Paging Controller (PC). This entity provides an alternate CoA to the MT. When packets arrive, the PC buffers them and informs the terminal that it must wake up; when the MT wakes up, the buffered packets are delivered and the MT starts using its new, real CoA (more details in [15]).

Support for dormancy/paging disallows keeping fresh the binding cache of correspondent entities, namely MMSPs: the terminal only acquires a real CoA when it wakes up; therefore, only after waking up it can update the correspondents’ binding caches. If the terminals are idle for long periods of time, as usually happens with mobile phones, the probability that the newly acquired CoA differs from the one the terminal had before going asleep is pretty high, even with slow mobility patterns. Dormancy/paging also introduces an additional delay in any message received, corresponding to the time it takes for the terminal to be located and waken up. Other than these, dormancy/paging has no issues: the alternate CoA provided by the PC may be used for location purposes as does the real CoA. Notice that this dormancy/paging affects only the reception of the initial INVITE.

[3] Sometimes erroneously called Network Access Identifier (NAI)

8. Delay Analysis

In this section we perform a comparative analysis of the dial-to-ringtone delay with standard and optimized signaling for a call between two roaming terminals (processing delays at the nodes are not accounted for). In order to simplify our analysis, the following assumptions have been made:

- Inter-domain delays are symmetrical, i.e., it takes about the same time to go from A to B as from B to A.
- Compared to the delay a message suffers in the wireless link or in inter-domain trips, the delay in intra-domain wired links is minimal and may, therefore, be neglected.
- In the RRP, the HoTI/HoT exchange takes longer than the CoTI/CoT.
- An RRP from a terminal to a local MMSP takes less time than the same procedure to a remote terminal, provided the HA is the same in both cases.
- The BU from the Caller arrives at MMSP1.f before the 183 Session Progress in the sequence of figure 2, meaning that the 183 S.P. will not go through the HA even in the standard case. This is almost always true, failing only if the inter-domain delay between the home and foreign domains of the caller is disproportionately large compared to the other delays.

In this analysis we will use the following notation: T_{W1} and T_{W2} are the delay at the wireless links of the caller and the callee, respectively; T_{F1F2}, T_{F1H1}, T_{F1H2}, T_{F2H1} and T_{F2H2} are the inter-domain one way trip delays (between combinations of the Foreign and Home domains of the caller – 1 – and the callee – 2); T_{DNS1} and T_{DNS2} are the delays for DNS lookups of the home and the foreign domains of the callee, respectively. Notice that if the entries are not cached, the DNS lookups imply at least one RTT to the DNS registrar, to find out the DNS server of the domain to be resolved, and another one or two to that domain, to find out the address of a SIP proxy (SRV record and, eventually, NAPTR record).

8.1 Standard Case

With standard, non-optimized signaling (refer to fig. 2), the INVITE takes

$$\begin{aligned}T_{Inv} &= T_{W1} + 2T_{F1H1} + T_{DNS1} + T_{F1H2} + T_{DNS2} + \\ &+ 3T_{F2H2} + T_{W2}\end{aligned} \tag{1}$$

to go from the caller to the callee. The QoS request can only be initiated after the caller's CoA has been found, which takes

$$\begin{aligned}T_{CoA1} &= 4T_{W1} + 4T_{W2} + 4T_{F2H2} + 3T_{H1H2} + \\ &+ 3T_{F1H1} + T_{F1H2}\end{aligned} \tag{2}$$

The QoS request/response at the callee side takes

$$T_{QoS1} = 2T_{W2} \tag{3}$$

Then the 183 Session Progress takes

$$T_{SP} = T_{W1} + T_{W2} + T_{F2H2} + T_{F1H2} \tag{4}$$

to go from the callee to the caller. Finding out the callee's CoA takes

$$T_{CoA2} = T_{W1} + 4T_{W2} + 3T_{F1H2} + 3T_{F2H2} + T_{F1F2} \tag{5}$$

The QoS request/response at the caller side takes

$$T_{QoS2} = 2T_{W2} \tag{6}$$

Finally, the PRACK is sent to the callee with the SDP counter-answer, after which it may start ringing. Until the 180 Ringing arrives at the caller, there is an additional

$$T_{Pra} = 2T_{W1} + 2T_{W2} + T_{F1F2} + T_{F2H2} + T_{F1H2} \tag{7}$$

If we add all of these delays, we get a dial-to-ringtone delay of

$$\begin{aligned}T_{Std} &= 14T_{W1} + 14T_{W2} + 2T_{F1F2} + 5T_{F1H1} + \\ &+ 7T_{F1H2} + 12T_{F2H2} + T_{DNS1} + T_{DNS2}\end{aligned} \tag{8}$$

8.2 Optimized Case

With our proposed optimizations (refer to fig. 3), the INVITE takes

$$T_{Inv} = T_{W1} + T_{DNS1} + T_{F1H2} + T_{F2H2} + T_{W2}$$

to go from the caller to the callee. Then, the QoS request takes

$$T_{QoS1} = 2T_{W2} \tag{9}$$

Then the 183 Session Progress takes

$$T_{SP} = T_{W1} + T_{W2} + T_{F2H2} + T_{F1H2} \tag{10}$$

to go from the callee to the caller. The QoS request/response at the caller side takes

$$T_{QoS2} = 2T_{W2} \tag{11}$$

Finally, the PRACK is sent to the callee, after which it may start ringing. Until the 180 Ringing arrives at the caller, there is an additional

$$T_{Pra} = 2T_{W1} + 2T_{W2} + T_{F1F2} + T_{F2H2} + T_{F1H2} \tag{12}$$

Adding these delays, we get a dial-to-ringtone delay

$$\begin{aligned}T_{Opt} &= 6T_{W1} + 6T_{W2} + T_{F1F2} + 3T_{F1H2} + 3T_{F2H2} + \\ &+ T_{DNS1}\end{aligned} \tag{13}$$

for the optimized signaling case.

8.3 Comparison

Table 1 summarizes the components of the initiation delay in the standard and optimized cases, described in the previous sections. As can be seen, the savings are obtained from a much more efficient delivery of the INVITE request and from the elimination of the need to perform RRPs for obtaining bindings.

Table 1. Delays in the standard and optimized initiation sequences

	Standard	Optimized
T_{Inv}	$T_{W1}+2T_{F1H1}+T_{DNS1}+T_{F1H2}+T_{DNS2}+3T_{F2H2}+T_{W2}$	$T_{W1}+T_{DNS1}+T_{F1H2}+T_{F2H2}+T_{W2}$
T_{CoA1}	$4T_{W1}+4T_{W2}+4T_{F2H2}+3T_{H1H2}+3T_{F1H1}+T_{F1H2}$	—
T_{QoS1}	$2T_{W2}$	$2T_{W2}$
T_{SP}	$T_{W1}+T_{W2}+T_{F2H2}+T_{F1H2}$	$T_{W1}+T_{W2}+T_{F2H2}+T_{F1H2}$
T_{CoA2}	$T_{W1}+4T_{W2}+3T_{F1H2}+3T_{F2H2}+T_{F1F2}$	—
T_{QoS2}	$2T_{W2}$	$2T_{W2}$
T_{Pra}	$2T_{W1}+2T_{W2}+T_{F1F2}+T_{F2H2}+T_{F1H2}$	$2T_{W1}+2T_{W2}+T_{F1F2}+T_{F2H2}+T_{F1H2}$
Total	$14T_{W1}+14T_{W2}+2T_{F1F2}+5T_{F1H1}+7T_{F1H2}+$ $+12T_{F2H2}+T_{DNS1}+T_{DNS2}$	$6T_{W1}+6T_{W2}+T_{F1F2}+3T_{F1H2}+3T_{F2H2}+T_{DNS1}$

If we consider $T_{W1}=T_{W2}=T_W$, $T_{DNS1}=T_{DNS1}=T_{DNS}$, and all inter-domain traversal delays equal to T_{ID}, the dial-to-ringtone delays in the standard and optimized cases become the following:

$$T_{Std}=28T_W+26T_{ID}+2T_{DNS} \quad (14)$$

$$T_{Opt}=12T_W+7T_{ID}+T_{DNS} \quad (15)$$

As can be seen, there is always a more than twofold improvement, independently of the actual values.

Figure 5 illustrates the variation of the dial-to-ringtone delay with standard and optimized signaling when all inter-domain delays for one-way trip are equal and assume values from 2ms to 64ms. The DNS lookups take twice that value plus 5ms (RTT to the DNS registrar). Delay on both wireless links is 10ms.

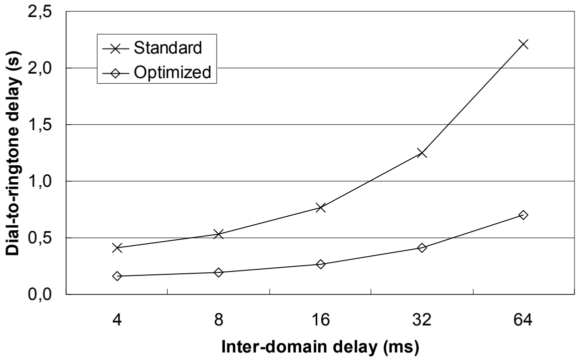


Figure 5. Dial-to-ringtone delay with varying inter-domain delay

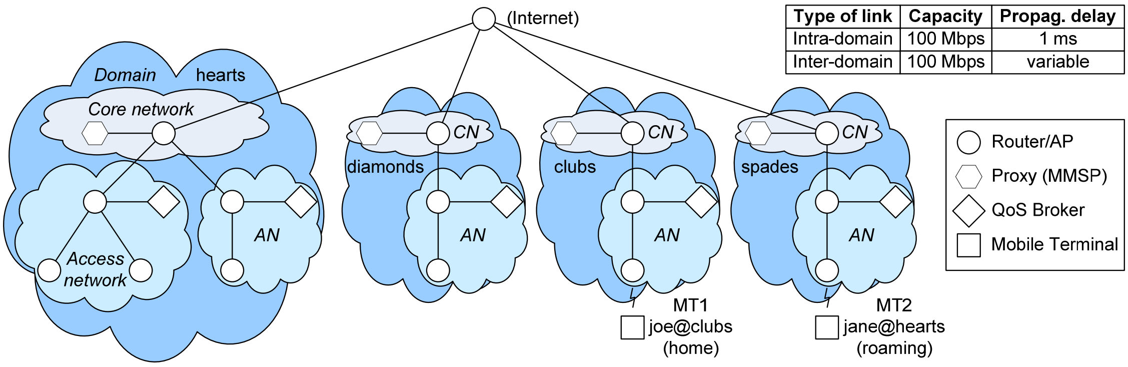
With our proposed optimized signaling, the delay is reduced to about one third of that obtained with standard, non-mobility-aware session signaling, a significant improvement. Without optimizations a user needs to wait more than 2 s to start the multimedia call when the inter-domain delay is 64 ms; with the optimizations, 0.7 s are sufficient for the multimedia call to be initiated under the same conditions.

9. Simulation Results

The efficiency of the standard and optimized signaling scenarios for the initiation of mobile multimedia applications was evaluated using the *ns-2* simulator [16] under Linux. The simulations comprise all possible combinations of: (1) caller terminal at the home domain or roaming; (2) callee terminal at the home domain or roaming; (3) caller and callee physically attached to the same or different domains and; (4) in the first case of (3), caller and callee physically attached to the same or different ANs, therefore representing all possible intra- and inter-domain call scenarios.

The standard *ns-2* simulator supports neither MIPv6 nor SIP. MIPv6 support was provided by the *mobiwan* extension [17], which we further improved by adding several features (reverse encapsulation, RRP, etc.) it did not support. Regarding SIP, although a previous implementation from NIST [18] existed, it is incomplete, difficult to extend, and supports only stateless entities. Therefore, we have performed a new implementation of SIP, layered, with stateful entities, and supporting QoS-aware user agents (UA) and proxies/registrars; it also supports reliability of provisional responses (100rel) SIP extension [19], used in these simulations. Our implementation of SIP for ns-2 is publicly available for download from [20].

Some processing delays are accounted for in the simulation model. Message processing is performed in a FIFO fashion, meaning that processing of each message can only begin after all previous ones have been processed. Processing delays for SIP messages were simulated at both the terminals (15ms) and the MMSP (0.8ms), with an increment for messages with SDP bodies (10ms in the terminals and 0.8ms in the MMSP). QoS request processing at the QoS brokers is also accounted for (1ms). The remaining



Type of link	Capacity	Propag. delay
Intra-domain	100 Mbps	1 ms
Inter-domain	100 Mbps	variable

Figure 6. Topology used in the simulations

processing delays are considered negligible when compared to these, and thus ignored in the simulations. DNS lookups were not simulated for lack of a realistic model for DNS caching. Moreover, since our purpose is the evaluation of signaling, no actual session data was simulated.

Figure 6 shows the topology used in the simulations, containing four domains, the leftmost one containing two ANs, one of which with two ARs. Notice that the total inter-domain delay is twice the inter-domain link delay. Though very simple, this topology allows us to simulate all possible combinations of roaming and non-roaming terminals: physically attached to the same AR, same AN and different AR, same domain and different ANs, or to different domains. 128 terminals were uniformly spread among the access networks, each terminal having a 50% probability of being at its home domain and 50% of being roaming. Random calls were generated between pairs of terminals, with an average duration of 120s and a mean interval between generated call of 15s, for a simulated time of 24 hours (86400s). Between 5 and 10 runs of each simulation, with different seeds, were performed, using different streams of the standard pseudo-random number generator (PRNG) of ns-2.27 for independent events.

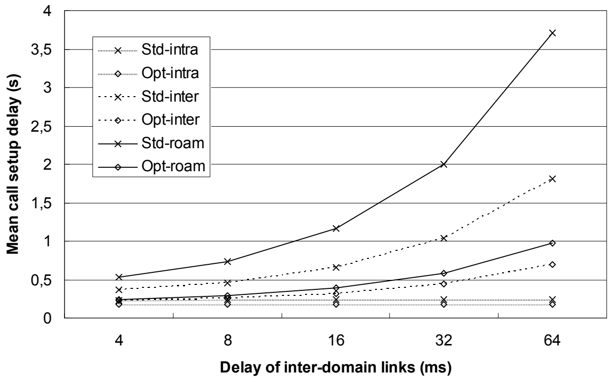


Figure 7. Call setup delay with varying inter-domain link propagation delay

In a first experiment we evaluated the call setup delay with different values of propagation delay for the inter-domain links. The setup delay is evaluated at the caller side, that is, from the moment the INVITE is sent to the moment the 200 OK for the INVITE is received and the ACK transmitted, subtracting the time it takes for the callee to answer the call (delay from sending the 183 Session Progress to sending the 200 OK). The results from this experiment are shown in figure 7 for both the standard and optimized sequences, in three different roaming scenarios (relative locations of the terminals intervening in a call). The roaming scenarios are identified by four letters, *abcd*, where *a* indicates if the caller terminal is at its home domain (*a*=*h*) or roaming (*a*=*r*), *b* holds similar information for the callee, *c* indicates if the terminals are connected to the same administrative domain (*c*=*y* or *c*=*n*), and *d* indicates if they are connected to the same AN (*y* or *n*). For example, *hhnn* means that both the caller and the calle are at their home domains, which are different (they are connected to different domains).

As expected, the setup delay does not vary with the propagation delay of inter-domain links in the *hhyy* scenario, since all signaling is performed intra-domain in this case. The worst scenario in terms of call setup delay is the *rrnn*, where both terminals are roaming and physically attached to different domains (as in figures 2 and 3). In this scenario, the difference in call setup delay between standard and optimized signaling is large, and increases with the propagation delay of inter-domain links: with 64 ms of propagation delay at the inter-domain links, the call setup delay with standard signaling is about 4 times larger than with optimized signaling. The 95% confidence intervals for the mean (5 runs), omitted in the figure for clarity, were less than ±3% of the mean in all cases.

In a second experiment we fixed the inter-domain propagation delay at 16 ms and introduced a varying

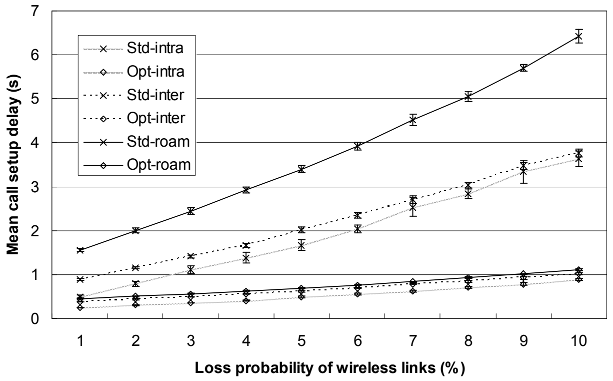


Figure 8. Call setup delay with varying loss probability in the wireless links

loss probability at the wireless links; 802.11 MAC layer retransmissions were disabled so that losses were not compensated. The results of this experiment are shown in figure 8 for different roaming scenarios, including 95% confidence intervals (10 runs).

The figure clearly shows that the non-optimized scenario is much more severely affected by packet loss than the optimized one; this behavior stems from the much larger number of exchanged messages. It is worth noting that even with a packet loss ration of 1%, the mean setup delay of the most favorable roaming scenario (*hhyy*) with standard signaling was larger than that of the least favorable one (*rrnn*) with optimized signaling, a gap that is largely widened as the loss probability increases.

The above presented results show the clear advantage, in terms of call setup delay, of the optimized signaling method over the standard one. The improvement is even more dramatic for long distance calls (larger inter-domain propagation delays) and/or in the presence of packet loss in the wireless links, even though small.

10. Conclusions

This paper identified the sources of inefficiency with the joint use of SIP and Mobile IPv6 (the probable protocols for session initiation and mobility support, respectively, in the next generation telecommunication systems) for the initiation of mobile multimedia applications, particularly when end-to-end resource reservations must be performed for the media. This inefficiency generally stems from SIP/SDP's unawareness of layer 3 mobility, and from the need to perform resource reservations accounting for the physical points of attachment of the terminals combined with that unawareness. A solution for these inefficiencies was proposed, based on the direct use of the Care-of Addresses in some messages (namely for the short-lived message transactions in call initiation) and on a few cross-layer interactions, namely by including layer 3 location information in session setup signaling.

The advantages of the proposed optimizations in session establishment were analyzed, and simulation results have demonstrated that the session initiation sequence is much faster with the optimizations than in the standard case, particularly in the presence of larger inter-domain link propagation delays (long distance calls) or packet loss in the wireless links.

Appendix 1. Binding Requests

Binding Requests (BReqs) in figure 2 behave somewhat differently from the Binding Refresh Requests (BRRs) defined in [1], which states that a mobile node should not respond to BRRs for addresses not in the Binding Update List (BUL). Although it is possible that the CN will respond with a BU if a packet or sequence of packets of any type (e.g., dummy packets) is sent to its HoA when it is roaming, we cannot rely on such solution because:

- There is no guarantee that it will do so.
- If the CN is at home, no BU would ever be received.

Therefore, with the behavior recommended by [1], it is not possible for a terminal to know for sure the physical location of the CN in order to perform an end-to-end reservation.

The reason for rejecting BRRs from nodes not in the BUL is to avoid being subject to a denial of service attack, since state maintenance is required for the RRP at the MT side. Unfortunately, it is not possible to make the procedure completely stateless: while the home and care-of init cookies could be implemented in such a way that the MT would not need to keep them, the first received keygen token (home or care-of) must be stored until the other one arrives. It is possible, though, to implement binding requests as used in this paper by having up to a limited number of low priority entries in the BUL used for replying to binding requests from nodes not in the BUL. These low priority entries are promoted to regular entries only when the conditions that would normally trigger a BU are met. Due to their limited number, these low priority entries do not affect the normal operation of the BUL even under a DoS attack (only a service which is not anyway provided by [1] may be denied); under normal conditions, this additional and potentially useful service is provided.

Acknowledgments

This work is based on results of the IST FP6 Integrated Project DAIDALOS, funded by the European Community's Sixth Framework Programme. It reflects the authors' views, and the EC is not liable for any use that may be made of the information herein contained.

Part of this work was carried out while the first author was with LIACC.

References

[1] D. Johnson, C. Perkins and J. Arkko, Mobility Support in IPv6, *IETF RFC 3775*, June 2004.

[2] 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.

[3] J.W. Jung, D. Montgomery, J.H. Cheon and H.K. Kahng, Mobility Agent with SIP Registrar for VoIP Services, *Web and Communication Technologies and Internet-Related Social Issues — HIS 2003*, Lecture Notes in Computer Science vol. 2713, Springer-Verlag, June 2003, pp. 454–464.

[4] E. Wedlund and H. Schulzrinne, Mobility Support using SIP, *Proc. ACM/IEEE WoWMoM'99*, Seattle, WA, USA, August 1999, pp. 76–82.

[5] R. Koodli (ed.), Fast Handovers for Mobile IPv6, *IETF RFC 4068*, July 2005.

[6] C. Politis, K.A.Chew and R. Tafazolli, Multilayer Mobility Management for All-IP Networks: Pure SIP vs. Hybrid SIP/Mobile IP, *Proc. IEEE VTC'2003 Spring*, Jeju, Korea, April 2003, pp. 2500–2504.

[7] Q. Wang and M. Abu-Rgheff, Integrated Mobile IP and SIP Approach for Advanced Location Management, *Proc. IEE 3G'2003*, London, UK, June 2003, pp. 205–209.

[8] Q. Wang, M. A. Abu-Rgheff and A. Akram, Design and Evaluation of an Integrated Mobile IP and SIP Framework for Advanced Handoff Management, *Proc. IEEE ICC'2004*, vol. 7, Paris, France, June 2004, pp. 3921-3925.

[9] R. Prior, S. Sargento, J. Gozdecki and R. Aguiar, Providing End-to-End QoS in 4G Networks, *Proc. IASTED CCN'2005*, Marina del Rey, CA, USA, October 2005, pp. 188-195.

[10] D. Willis and B. Hoeneisen, Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts, *IETF RFC 3327*, December 2002.

[11] M. Handley and V. Jacobson, SDP: Session Description Protocol, *IETF RFC 2327*, April 1998.

[12] D. Kutscher, J. Ott and C. Bormann, Session Description and Capability Negotiation, *IETF Internet Draft* (draft-ietf-mmusic-sdpng-08), February 2005.

[13] S. Olson, G. Camarillo and A.B. Roach, Support for IPv6 in Session Description Protocol (SDP), *IETF RFC 3266*, June 2002.

[14] R. Engel, V. Peris, D. Saha, E. Basturk and R. Haas, Using IP Anycast for Load Distribution And Server Location, *Proc. 3rd Global Internet Mini Conf.*, Sydney, Australia, November 1998, pp. 27–35.

[15] A. Banchs, I. Soto (eds.) et al., *Detailed Specifications including Complete Interface Specifications*, Daidalos (IST-2002-506997) consortium deliverable D221, January 2005.

[16] The Network Simulator – ns-2, version 2.27. <http://www.isi.edu/nsnam/ns/>

[17] Mobiwan MIPv6 extension to ns-2. <http://www.ti-wmc.nl/mobiwan2/>

[18] NIST IP Telephony Project. <http://snad.ncsl.nist.gov/proj/iptel/>

[19] J. Rosenberg and H. Schulzrinne, Reliability of Provisional Responses in the Session Initiation Protocol (SIP), *IETF RFC 3262*, June 2002.

[20] NS-2 Network Simulator Extensions. <http://www.ncc.up.pt/~rprior/ns/>