5-9-2007

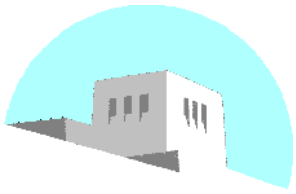
Efficient User Controlled Inter-Domain SIP Mobility: Authentication, Registration, and Call Routing

Joud Khoury

Henry Jerez

Chaouki Abdallah

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING



SCHOOL OF ENGINEERING
UNIVERSITY OF NEW MEXICO

Efficient User Controlled Inter-Domain SIP Mobility

Authentication, Registration, and Call Routing

Joud Khoury Henry Jerez Chaouki Abdallah [1][2]

UNM Technical Report: EECE-TR-07-010

Report Date: January 1, 2007

[1]Joud Khoury and Chaouki Abdallah are with the Department of Electrical and Computer Engineering, of the University of New Mexico, Albuquerque NM 87131, {*jkhoury, cabdallah*}*@ece.unm.edu*. Henry Jerez is with the Corporation for National Research Initiatives, Reston VA 20191, *hjerez@cnri.reston.va.us*.

[2]The work presented in this report is partially funded by the National Science Foundation NSF under the Future Internet Design (FIND) Grant CNS-0626380.

Abstract

Over the past decade, multimedia services have gained significant acceptance and played an important role in the convergence of IP networks. Supporting mobility in IP (Internet Protocol) networks is a crucial step towards satisfying the nomadic communication paradigms on the current Internet. The Session Initiation Protocol (SIP) presents one approach towards supporting IP mobility. Additionally, SIP is increasingly gaining in popularity as the next generation multimedia signaling and session establishment protocol. It is anticipated that the SIP infrastructure will be extensively deployed all over the Internet. In this paper, we explore an efficient approach to inter-domain SIP mobility in an attempt to improve personal and terminal mobility schemes. We succeed in applying a persistent identification framework to application level SIP addressing by introducing a level of indirection on top of the traditional SIP architecture. We refer to our approach as the Handle SIP (H-SIP). H-SIP leverages the current SIP architecture abstracting any domain binding from users. Our approach to mobility is user-controlled. We experimentally prove the efficiency of H-SIP in achieving inter-domain authentication and call routing through modeling and real-time measurements.

Keywords

Mobile environments, Multimedia support, Roaming, Session Initiation Protocol, SIP, Voice-over-IP.

1 Introduction

Low cost, data services and deployment of high speed access networks are pushing service providers and enterprises to adopt packet switched multimedia communication as opposed to current circuit-switched and Cellular alternatives. The industry has recently witnessed a rapid increase in the popularity and deployment of Voice over IP (VoIP) services. Enterprises and mobile operators are currently promoting simultaneous Cellphone/Wi-Fi access by introducing dual-mode phones that switch between the cellular network and the IP packet switched network. Obviously, significant benefits follow for both the consumer, who can save money and get broader coverage, and the operators who can widen their service base for a lower cost. However, identity persistence issues become critical in such networks and need to be addressed in this context.

The Session Initiation Protocol (SIP) [1] and H.323 [2] are among the most widely adopted protocols for IP telephony. While SIP and H.323 have different architectural components, the ability of these two protocols to coexist, and the simpler implementation and open collaboration of SIP, are factors that drive this paper to focus on SIP and to introduce the proposed notions and implementations in the context of SIP. Additionally, SIP has been accepted by the 3rd Generation Partnership Project (3GPP) as a signaling protocol for establishing real time multimedia sessions. The protocol is continuously gaining in popularity and deployment and has been adopted by service providers like Verizon and Sprint to provide IP telephony, instant messaging, and other data services. It is anticipated that SIP will be widely deployed by operators and enterprises, thus populating the Internet with SIP infrastructural components. The widespread deployment of SIP is a major premise of this paper, as we will leverage this idea to propose an efficient inter-domain mobility scheme for SIP environments.

The session initiation protocol (SIP) [1] is a signaling and control protocol for handling multimedia sessions, allowing the establishment and termination of media streams between two or more participants. SIP works in concert with other multimedia protocols due to the independence of the protocol from the underlying transport mechanisms and session types. The SIP architecture allows for its deployment as a centralized system, a distributed system, or a combination of both.

The SIP architecture is also proposed as an efficient candidate that can be reused to provide personal, terminal, and session mobility [3, 4, 5, 6] with a readily available infrastructure. This avoids the redundancy introduced by simultaneous deployment with Mobile IP [7]. The successful reuse of SIP to simultaneously support both multimedia communications and mobility leverages the issues emanating from SIP users *roaming*[1] across multiple SIP domains. These issues are highlighted through a brief overview of the SIP protocol functionality.

SIP handles user location through the use of a Proxy/Location server [2] that accepts user registration requests and updates the respective user location in a location repository. The protocol inherently implements location independence through the use of the uniform resource identifiers (URI) [8], which directly offers personal mobility. A URI acts as a location independent identifier abstracting the actual physical location of a user with respect to the system. So, SIP allows for personal mobility whether through the use of a proxy that sets up the session between the calling parties or through the use of redirection servers. However, the protocol defines a user only within the domain boundaries of the service provider. A user must associate with a specific proxy server that handles user authentication as well as initial traffic routing. The proxy maintains a unique account for the user, who in turn, is expected to coordinate with that same proxy irrespective of his location. This requirement translates into unnecessary loads on the SIP server and on a particular domain. Additionally, it complicates the coordination of *roaming* users who must communicate with a central proxy server while roaming. Despite the possible presence of firewalls and other network restrictions on the foreign domain, roaming users are required to use the central home server instead of using the available local servers. Consequently, while URIs solve the location binding issue, they introduce the domain binding issue. Inefficient traffic routing is a direct consequence of such binding. Besides, the URI identification translates into users needing to be aware of each others' current domain associations. It also brings up the complexity of satisfying calls when initiated from regular keypad terminals.

[1] Throughout this paper, roaming is defined as the SIP inter-domain roaming i.e. the migration of a user between different SIP domains.

[2] We will use the terms SIP proxy, SIP server, SIP registrar interchangeably.

This paper addresses the inter-domain mobility issue by introducing an abstraction framework based on a unique and persistent identification mechanism. As far as the paper is concerned, it only provides an approach that can enhance personal and terminal mobility [5] in current SIP architectures. As to session mobility, the readily available approaches like mid-call mobility [4] or enhancements to that [9] may be used. The framework we propose, which we refer to as the Handle-SIP or H-SIP, can seamlessly fit into the current SIP architecture allowing SIP users to transparently roam across different SIP domains. H-SIP can be gradually deployed and can coexist with the traditional SIP infrastructure. User location and association is abstracted through the use of globally unique and persistent identifiers called *handles* which are part of the Handle System [10, 11, 12, 13]. The Handle System is a distributed system extensively used as an indirection layer for the management of persistent Identifiers. Using the Handle System as an intermediate layer on top of multiple distributed SIP implementations allows us to implement seamless multi-domain authentication and call routing.

The rest of the paper is structured as follows. Section 2 shows how H-SIP is efficiently used to enhance inter-domain SIP mobility. In this section we present a detailed explanation of the proposed inter-domain authentication, registration and call routing mechanisms. In section 3, we develop the implementation test-bed and we discuss the experimental model and performance measurements of H-SIP. Future work is presented in section 4.

2 SIP Inter-Domain Mobility

2.1 Sessions and Mobility

To clarify the SIP inter-domain mobility problem, we will present a simple example. Recall that SIP defines a user as an entity that associates with a particular domain. Figure 1 depicts a simple scenario of a roaming user *r_user* who has a valid association with his home domain *hdomain* but is currently present in a foreign domain *fdomain*. SIP signaling traffic originating from (REGISTER) or terminating at (arrows 1,2,3: arbitrary SIP user trying to INVITE the roaming user) *r_user* must inefficiently pass through his home proxy server. Figure 1 identifies this traffic as traditional traffic flow.

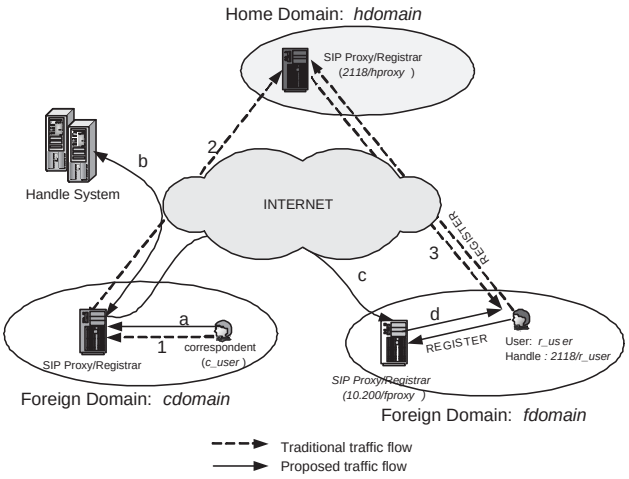


Figure 1: A Reference inter-domain roaming scenario

There are several ways in which roaming issues can be addressed, depending on whether the SIP architecture is roaming-unaware or modified to become roaming-aware. We study these issues and we present our approach by showing a typical flow for INVITE and REGISTER requests. We also compare the different approaches and illustrate the different scenarios in Figure 2. A more elaborate description of these scenarios is presented in section 3.

- The first scenario shows how SIP naturally handles a call flow for a roaming user. A data flow is presented in Figure 2.A. In this case, no roaming logic is injected into the system (system is roaming-unaware). All requests to/from the roaming user must go through the central home proxy server. The home proxy thus treats both roaming and non-roaming users equally and portrays a roaming user as merely a home domain user registering with a foreign contact address. Clearly, if the user is present in another country, his traffic would still have to go through his central home proxy (triangle routing) as depicted in Figure 2.A, despite the availability of a local proxy server in the foreign domain (Foreign Server). This results into significant delays that are not accepted for time sensitive applications. Even with SIP mobility management (SIPMM) [3, 4] support (personal, terminal and session mobility) enabled, the same scenario occurs.

  SIP Mobility allows a user to roam between subnets and domains maintaining accessibility and session continuation using pre-call and mid-call mobility signaling. With pre-call signaling, the mobile user will re-REGISTER with the home proxy anytime his IP address changes. With mid-call signaling, the mobile user will negotiate an address change with the correspondent user while the session is in progress using re-INVITE messages. Mid-call mobility assumes a session is already in progress between the calling parties. Inefficient pre-call traffic routing, and service centralization, are obvious limitations that users roaming in these traditional and Mobile SIP environments have to suffer from. This is the same case also for Mobile IP with Location Registers (MIP-LR) [14, 15], whereas here the SIP proxy servers are replaced with location registers. We argue that our proposed approach to roaming and inter-domain mobility in general, can significantly enhance the SIP personal and terminal mobility performance. Additionally since our approach addresses SIP personal and terminal mobility, we can improve the pre-call portion of any SIP session mobility scheme while other features like mid-call mobility can remain unchanged. For mid-call mobility, current proposals like MIP-LR, SIPMM, or a combination of these two [16]) can be used. These approaches implement mid-call mobility by sending binding updates directly to correspondent nodes without going through Home Agents. Mobile IP (MIP) [7], however, uses Home Agents to forward traffic which creates triangular routing issues. An enhanced version of MIP is MIPv6 [17] that avoids triangular routing and implements route optimization. As to the simultaneous mobility issue, discussed lately in [18], it is left for a future paper to offer a secure framework for simultaneous mobility in the context of H-SIP.

- A second scenario is that of a SIP roaming-aware approach such as the one proposed by Double User Agent Servers [19], that mimics the roaming solution employed in the telecommunication environments. In other words, a user who is roaming outside his home domain, registers with a foreign server. The latter consults the user's home server for redirection, authentication and billing, and proceeds to process the user's transactions. Correspondent users trying to communicate with the roaming user will have to go through his home proxy server which in turn redirects them to the foreign proxy where the user is currently located. Hence, significant signaling overhead results primarily due to the nature of the SIP URI. The URI is composed of a domain part, like in *r_user@hdomain*, thus forcing the calls directed to this user to go through the *hdomain* proxy server first. The data flow for this scenario is presented in Figure 2.B. We argue that this approach is inefficient as it introduces unnecessary overhead and load on the original server .

In the two scenarios above, the use of URIs to identify users and the inherent dependence of the URI on a particular domain, complicates message routing. One solution is to abstract the actual identifier eliminating per-call coordination to minimize the signaling traffic in highly mobile environments.

2.2 H-SIP: Abstraction layer

Our proposed approach uses *handles* as globally unique identifiers to locate and identify SIP architectural elements. This abstraction allows the system to route calls independent of user location and domain association. We refer to the modified SIP framework as the Handle-SIP or H-SIP. Note that we have also exploited this abstraction approach at the level of network devices and services in [20, 21].

Briefly, the Handle System [10, 11, 12, 13] is intended to be a means of universal basic access to registered digital objects [22]. It provides a distributed, secure, and global name service for administration and resolution

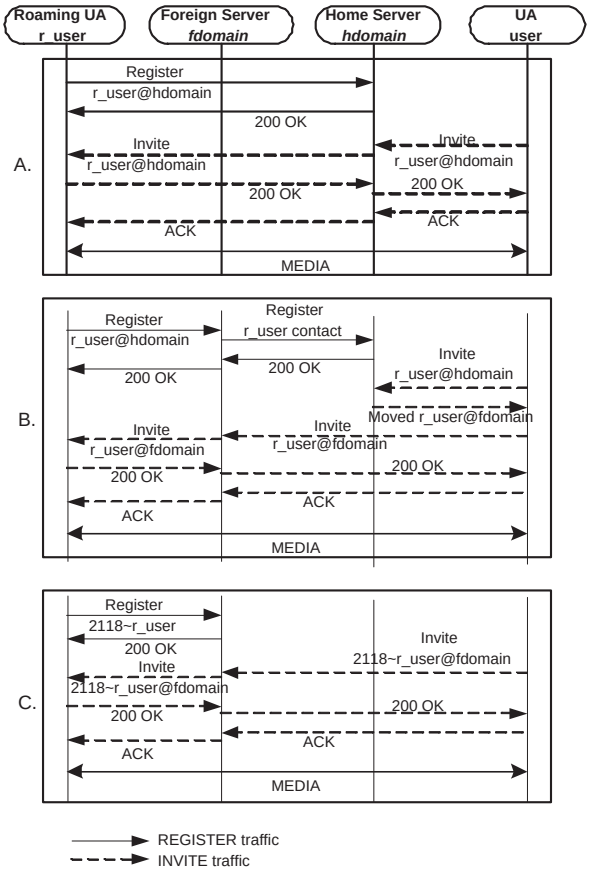


Figure 2: SIP traffic flow A. With no roaming logic B. With traditional roaming logic C. With proposed roaming logic

of *handle*s over the Internet. A *handle* is a persistent name that can be associated with a set of attributes. Some of these attributes describe location, permissions, administrators and state. The fact that *handle*s are defined independently of any of the attributes or public keys of the underlying objects, makes them persistent identifiers [23]. These identifiers are managed and resolved using a secure global name service that guarantees the association of the identifier with its respective attributes over distributed communication.

Security is a crucial property of the Handle System. The system acts as a certification authority assuring that attributes of the name/reference are securely transferred between the communicating ends. Hence, the Handle System allows for secure name resolution and administration in a distributed fashion making it highly scalable and suitable to operate in mobile environments. In our approach, elements of the SIP architecture, SIP users and proxy servers, are identified with *handle*s abstracting any domain binding. Users will identify each other, as well as the SIP servers they associate with using *handle*s instead of URIs and domain names respectively. In Figure 1, the roaming user *r_user* will have his own *handle 2118/r_user* with the necessary administrative privileges over the *handle*. Additionally, the home proxy server has a *handle*[3] *2118/hproxy*, and the foreign proxy server has a *handle 10.200/fproxy*. Note that a *handle* has the form "*prefix/suffix*". The prefix represents the naming authority (NA) while the suffix represents a unique local name under the NA namespace [10], thus rendering the *handle* globally unique. A possible realization of the *handle 2118/r_user* inside the Handle System is depicted in Figure 3. The *handle* has several fields. The HS_ADMIN and HS_VLIST fields determine the administrators of the *handle* who are the naming authority (*0.NA/2118*), the *handle* itself (*2118/r_user*) and the

[3]Please note that a direct mapping between domains and *handle*s exists and is enabled by a handle/DNS proxy approach.

handle	Field Type:index	Value
2118/r_user	HS_ADMIN:100	rwr:0.NA/2118:300
	HS_ADMIN:101	rwr:2118/r_user:200
	HS_VLIST:200	rwr:2118/r_user:300
		rwr:2118/hproxy:300
		rwr:10.200/fproxy:300
	SIP_URL:250	sip:2118.r_user@x.y.z.w
	SIP_PWD:251	password
	HS_PUBKEY:300	00BE0034……..

Figure 3: Sample user *handle* structure

two proxy servers in the HS_VLIST field. Any of these administrators has the privilege to modify the fields inside the *handle* provided the administrator succeeds to authenticate with the Handle System using his private key.

2.3 Authentication and Registration

Currently, the most common authentication mechanism employed by SIP is the digest authentication [24] used by HTTP. When a user associates with a domain proxy server, he obtains an account on that server with a username and password which he uses to authenticate himself to the server if asked to. The digest authentication depicted here is domain dependant i.e. the user's credentials are valid for a particular domain. Briefly, digest authentication proceeds as follows:

1. User sends a REGISTER request to a SIP proxy/registrar server.
2. The server replies with a 401 unauthorized response message challenging the user to authenticate himself for the requested service (realm) through a user and password prompt.
3. The user sends back a message digest of his credentials, which include his username, password etc.
4. The same message digest is computed internally using the server's internal user information and compared to the one sent by the user.
5. Authentication is granted if the two digests match.
6. User registers with the SIP proxy/registrar server.

In our approach, we still use digest authentication for the SIP users due to its wide support by current SIP servers and user agents, although a better authentication mechanism can be designed that would leverage the inherent security that *handle*s expose.

Access to the authentication information is controlled inside the Handle System by the users. Recall that each user owns and administers his own *handle*. As part of this process, the user specifies in the HS_VLIST field, the set of *handle*s that have administrative rights over his *handle*. Among these *handle*s, the user should include *handle*s of any SIP proxy server that he wishes to register with, which could be any foreign server(s) that he trusts.

Two approaches can be exploited to implement the logic needed by the current SIP architecture for supporting *handle* authentication and registration. The first is to modify the actual SIP servers by extending their functionality through a server plug-in. This approach requires no changes to the current User Agent devices whether hardphones or softphones. The devices will adapt seamlessly to the system. Alternatively, a second approach is to modify the User Agent devices instead, which is a more cumbersome task that would require software upgrades for all existing User Agents.

This paper implements the first approach that deals with extending the functionality of the proxy/registrar servers. We present the proposed solution in light of the reference example of Figure 1. In Figure 3, the roaming user *2118/r_user* has granted both SIP proxy servers *2118/hproxy* and *10.200/fproxy* administrative rights over his *handle*. Note that the VLIST could refer to another *handle* containing a list of globally trusted servers. For the roaming user *r_user* present in the foreign domain *fdomain*, the authentication/registration process with the foreign proxy server *10.200/fproxy*, depicted in Figure 2.C and Figure 1 (proposed traffic flow, arrows a,b,c,d), proceeds as follows:

1. *r_user*, after including the *handle 10.200/fproxy* in his *handle* HS_VLIST field, sends a REGISTER request to *fproxy*.
2. *fproxy* challenges *r_user* to authenticate himself.
3. *r_user* uses same digest authentication with username as the *handle 2118˜r_user* and password as the value of the SIP_PWD field that he created in his *handle* as shown in Figure 3.
4. *fproxy* uses the Handle Protocol [12] to resolve the *handle 2118/r_user* into the SIP_PWD field. The server then computes a message digest over the obtained credentials.
5. Authentication is granted if the two digests match.
6. After authenticating *2118/r_user*, the foreign proxy *fproxy* proceeds to create an internal account for *r_user* to be able to use the SIP services on *fproxy*. The internal user account will have a username identical to the *handle* of the registering user with the '˜' replaced by '.' i.e *2118.r_user* in this case.
7. Registration of the user follows. This requires that *fproxy* modifies the *handle 2118/r_user* updating the field SIP_URL to point to the internal account, *2118.r_user@x.y.z.w* in this case, as shown in Figure 3. This means that *r_user* is currently associated with *fproxy*.

handle	Field Type:index	Value
10.200/fproxy	HS_ADMIN:100	rwr:0.NA/10.200:300
	HS_ADMIN:101	rwr:10.200/fproxy:300
	INET_HOST:240	x.y.z.w
	HS_PUBKEY:300	00BE00445……..

Figure 4: Sample proxy *handle* structure

Obviously, our modified authentication algorithm is domain independent. In other words, the user's credentials are valid for all realms provided the correct administrative privileges are set in the Handle System. This property is essential, as it allows a particular authenticated SIP message to traverse multiple domains instead of requiring re-authentication for each domain on the path of the message. Since all communication between the Proxy and the Handle System is secure [12], the proxy can be reasonably certain that the roaming user is indeed who he claims to be by validating his credentials against the secure *handle*. Internally, the proxy server monitors the user accounts created and removes an account (also updating the *handle*) due to unregister requests or account expiration. A sample *handle* for the foreign proxy is shown in Figure 4.
Devices, whether hardphones and softphones are treated similarly. This depends on the ability of the device owner to present the SIP proxy with a username (could be the *handle*) and password for authentication.

With this approach, a user no longer needs to register with a home proxy server, as was required by pre-call mobility [4]. After registering with the foreign server, the user's handle-to-URI mapping remains fresh allowing correspondent users to reach him simply by addressing his *handle* as we will show in the section 2.4.

2.4 Routing

After abstracting any domain binding from users and allowing seamless authentication and registration with local proxy servers, the next step is to permit the user to initiate and receive calls by addressing a particular *handle* with no explicit reference to domain bindings (URIs). In this sense, a SIP user can INVITE any other SIP user provided he knows the latter's *handle*. From the perspective of a user, all other users seem to belong to one local domain and abstraction is complete.

It is extremely important to minimize the call setup time for the time-sensitive and interactive applications enabled by SIP. To explain how the this is achieved, we will go through the steps where an arbitrary SIP user *c_user* (caller) tries to INVITE the roaming user *r_user* (callee) using the latter's *handle 2118/r_user* as shown in Figure 1. The call routing process, presented in Figure 2.C, proceeds as follows:

1. Caller *c_user* sends an INVITE request to *r_user*. The invite request reaches the caller's SIP proxy/registrar containing the following header fields:

   ```
   INVITE sip:2118~r_user@somedomain SIP/2.0
   To:<sip:2118~r_user@somedomain> .........
   ```

   Note: In this message, the domain *somedomain* is irrelevant to our approach. We are only concerned with the *handle* part of the Request-URI. To distinguish between *handle* and non-*handle* requests, we resort to the '/' character[4] in the host name.

2. Proxy checks if the *handle 2118~r_user* is a locally registered user. If not, the server resolves the *handle* into the SIP_URL field which is *2118.r_user@x.y.z.w* in this case as shown in Figure 3.

3. The server then rewrites the target URI of the message to the resolved URI.

4. From this point on, the natural SIP call flow is leveraged and the traditional SIP architecture [1] is utilized for efficient call routing. Note that other proxy servers on the call path treat the request as a normal request i.e. no *handle* resolution is required.

Again, with our approach, correspondent users trying to communicate with the mobile user need not go through a home proxy for session setup or redirection. This renders the call route more efficient eliminating unnecessary overhead and significant round-trip times.

One last point worth mentioning is the ability of a user to register with multiple servers from different devices simultaneously using the same *handle*. In our implementation, the *SIP_URL* field of a particular *handle* can contain a list of bindings (URIs) to enable this attractive property. Exploiting this property is left for future papers.

2.5 User-Controlled Mobility

Our mobility scheme is user-controlled in the sense that the user is responsible for the administration of his SIP identifier now that the latter is domain independent. Consequently, routing the user's calls through a local server simply requires the user to add the server's persistent identifier (*handle*[5]) to her persistent identifier's admin list. This indicates that the user trusts the local server to route her calls.
The other aspect of user control is the distributed service model that potentially eliminates the need for service level agreements SLAs between domains. Instead, the local domain proxy will directly challenge the user and

[4]Since Internet hostnames can not contain the '/' character [25] ascii (0x2F) (essential character in the *handle* Namespace [10]), we replaced it with the '~' ascii (0x7E) character in the examples above for implementation purposes. We also allow the '#' ascii (0x23) character for compatibility with hard IP phones.

[5]We have used the terms persistent identifier and *handle* interchangeably throughout this paper since our current implementation of the persistent identifier is the *handle*.

grant her trust provided sufficient credentials exist within the user's persistent identifier. These credentials can include financial information as well as trust information that the actual server can validate before allowing the user to route traffic through.

3 Implementation

3.1 Test-bed

We have implemented the functionality described in this paper as an extension to two open source SIP servers, the JAIN-SIP Proxy [26] and the SIP Express Router (SER) [27]. The JAIN-SIP proxy server is an open source JAVA based SIP proxy built on top of the JAIN-SIP-1.1 API. SER is an open source, configurable SIP server that is widely deployed in the research community. We have implemented a JAVA based H-SIP API that can be easily called from both proxy servers to expose the H-SIP interface operations. The operations mainly enable inter-domain authentication, registration and call routing using *handle*s. All the results depicted hereafter are based on the JAIN-SIP proxy.

Our test-bed is a realization of the framework depicted in Figure 1. We are running three modified SIP servers on three separate domains:

1. *ece.unm.edu* located at the University of New Mexico, Albuquerque, New Mexico. [Server IP: 129.24.24.106]
2. *cnri.reston.va.us* located in Reston, Virginia. [Server IP: 132.151.9.104]
3. *istec.org* located in Panama. [Server IP: 168.77.202.59]

A roaming user is allowed to move across the domains while establishing connectivity within each domain using the respective local server. The three servers are running Fedora Core 4 kernels and are identical in terms of work load and processing speed (AMD Athlon 1.1 GHz processors).

To use the framework, users are expected to be able to manage their own *handle*s. The Handle System provides a free administration tool [13] for this purpose. This tool is currently implemented in JAVA.

Currently, users wishing to associate with a proxy server are required to specify the latter's IP address or domain name. Since our approach exploits *handle*s instead of domain names, we have implemented a specialized gateway that translates between Domain Name System (DNS) and *handle* protocols. The gateway is responsible for protocol translation, specifically, *handle* to DNS. We refer to this gateway as the Handle-DNS proxy (HDP). HDP is a modified DNS server that communicates using the BIND protocol and implements extra functionality allowing it to associate canonical names and aliases inside its particular naming zone with *handle*s. HDP will therefore resolve canonical names inside its naming zone using the Handle System and will, in addition, allow any common DNS server to resolve DNS entries in the format: <handle>.[DNS proxy domain] to the actual value of the SIP_HOST attribute of that particular *handle*. Details about this gateway which can be extended to become a plug-in for current DNS servers are presented in [20].

The ease of deployment of our framework and success of the conducted experiments encourage us to pursue this work.

3.2 Experimental Model

In this section, we will compare the performance of roaming and non-roaming environments in terms of registration and call establishment delays. The non-roaming case represents traditional SIP signaling between a user and his home SIP server irrespective of roaming while the former case represents our proposed approach. Both cases were tested under the same conditions regarding UA/SIP processor speed and work loads. As UA, we used the Java SIP Communicator [28]. We also tested with the Cisco 7940/7960 IP phones.

Registration

Figure 5 is a magnified image of Figure 2 that focuses on the registration process according to the test-bed. We show a comparison of the registration process for roaming (H-SIP) and non roaming (no H-SIP) environments. *r_user* is located in *fdomain*, *ece.unm.edu*. For the no H-SIP case, we tested with *hdomain* being either *cnri.reston.va.us* (Reston, Virginia) or *istec.org* (Panama). Again, registration here assumes the basic digest authentication with the proxy server. In our model, the UA refreshes its registration with the proxy/registrar continuously (we used a registration TTL of 1 minute).

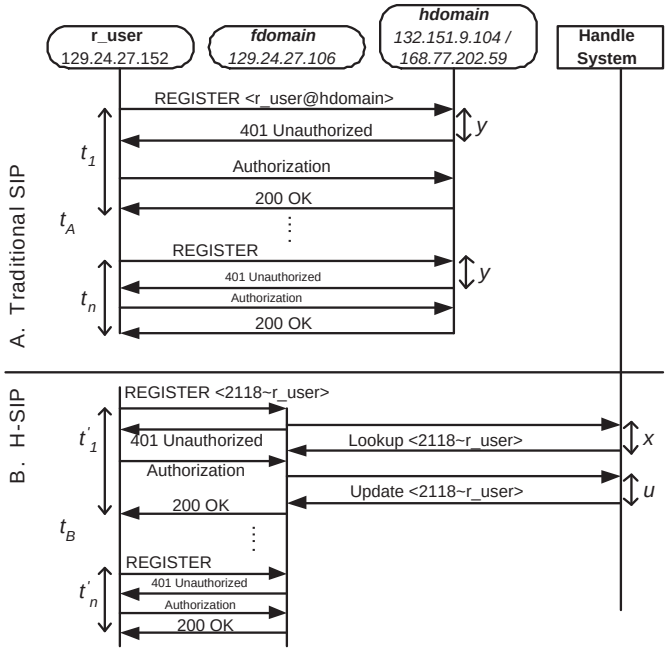


Figure 5: REGISTER message flow A. With no roaming logic C. With proposed roaming logic H-SIP

If we estimate the server's average processing time of the REGISTER request including digest authentication by α , then from Figure 5.A, the average registration time t_A as seen by *r_user* is given by,

$$t_A = \frac{1}{n} \sum_{1 \le i \le n} t_i = t_1 \tag{1a}$$

since,

$$t_1 = t_2 = \ldots = t_n \approx \alpha + 2y \tag{1b}$$

where t_i is time consumed by the i^{th} REGISTER, and y is the average round-trip communication delay between *r_user* and the registering SIP server. Obviously, t_A includes a relatively significant communication delay as a result of the presumably large geographical separation between the roaming user and the home SIP server.

Now, if we consider H-SIP, then from Figure 5.C, the average registration time t_C as seen by *r_user* is given by,

$$t_C = \frac{1}{n} \sum_{1 \le i \le n} t_i^{'} \tag{2a}$$

and,

$$t_1^{'} \approx \alpha + x + u \tag{2b}$$

$$t_2^{'} = \ldots = t_n^{'} \approx \alpha \tag{2c}$$

where $t_i^{'}$ is time to perform the i^{th} REGISTER, x is the average *handle* resolution delay and u is the average *handle* update delay. First, we note that the round-trip delay y is negligible in this case due to the existence of *r_user* and the SIP server on the same local network. Besides, note here that the server will issue one *handle* resolution and one update for the first REGISTER request only. Additionally, the first *handle* resolution is always cached internally on the server. Unless *r_user* moves to another domain or un-registers, no *handle* resolution/update is required. This means that subsequent REGISTER requests will read the cached value and thus, for $i > 1$, $x = u = 0$. This also means that x and u can be discarded for n sufficiently large.

Here, t_A and t_C are measured by *r_user* as the average time between sending the REGISTER request and receiving the 200 OK response. In Figure 5, *r_user* performs a DNS lookup or a Handle-DNS lookup in scenarios A and C respectively. These lookups are excluded from the performance metrics i.e. t_1 and t_2 do not include the DNS lookups for the SIP servers. However, we closely examine and compare the two lookup times and we show the performance results in section 3.3.

Call establishment

Figure 6 is a magnified image of Figure 2 that focuses on the call establishment process according to the test-bed, where *c_user* will try to establish a call with the roaming user *r_user*. For this section, *cdomain* is *ece.unm.edu* [New Mexico]. The *hdomain* is *istec.org* [Panama] and the *fdomain* is *cnri.reston.va.us* [Virginia]. For brevity, the 100 TRYING messages are excluded from Figure 6. We focus on the INVITE message flow and the servers require no authentication. We denote by $rt_{(m,n)}$ the round-trip communication delay between nodes m and n.

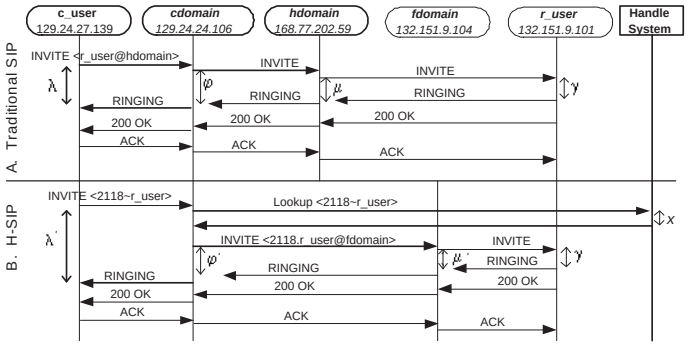


Figure 6: INVITE message flow A. With no roaming logic C. With proposed roaming logic H-SIP

For example, in Figure 6, $rt_{(cdomain,hdomain)}$ is the round-trip delay between the *cdomain* SIP server and that of *hdomain*. If we estimate the server's average processing time of the INVITE request by β, and the UA's average processing time of the INVITE request by γ, then again from Figure 6.A,

$$\lambda \approx rt_{(c\_user,cdomain)} + \beta + \phi \tag{3a}$$

$$\phi \approx rt_{(cdomain,hdomain)} + \beta + \mu \tag{3b}$$

$$\mu \approx rt_{(hdomain,r\_user)} + \gamma \tag{3c}$$

and therefore,

$$\lambda \approx rt_{(c\_user,cdomain)} + rt_{(cdomain,hdomain)} + rt_{(hdomain,r\_user)} + 2\beta + \gamma \tag{3d}$$

where λ is the average call establishment time (the INVITE/RINGING round-trip) as seen by *c_user*, ϕ is the average INVITE/RINGING round-trip time between the *cdomain* and *hdomain* SIP servers and μ is the average

Table I: Average Registration delays with and without H-SIP (as shown in Fig. 5), $n = 1000$, transport=UDP

		Registration Delays [ms]			
		x, u	*y*	t_A	t_C
No H-SIP	hdomain: virginia	NA	86	209	NA
H-SIP	hdomain: panama	NA	116	275	NA
H-SIP		84, 260	0	NA	40

INVITE/RINGING round-trip time between the *hdomain* SIP server and *r_user*.
Now, if we consider H-SIP, then from Figure 6.C,

$$\lambda^{'} \approx rt_{(c\_user,cdomain)} + \beta + x + \phi^{'} \tag{4a}$$

$$\phi^{'} \approx rt_{(cdomain,fdomain)} + \beta + \mu^{'} \tag{4b}$$

$$\mu^{'} \approx rt_{(fdomain,r\_user)} + \gamma \tag{4c}$$

and therefore,

$$\lambda^{'} \approx rt_{(c\_user,cdomain)} + rt_{(cdomain,fdomain)} + rt_{(fdomain,r\_user)} + 2\beta + x + \gamma \tag{4d}$$

where $\lambda^{'}$ is again the average call establishment time as seen by *c_user*, $\phi^{'}$ is the average INVITE/RINGING round-trip time between the *cdomain* and *fdomain* SIP servers, $\mu^{'}$ is the average INVITE/RINGING round-trip communication time between the *fdomain* SIP server and *r_user*, and x is the average *handle* resolution delay. Note here that each INVITE request will require the server to issue one fresh *handle* resolution.
Both λ and $\lambda^{'}$ are measured by *c_user* as the time between sending the INVITE request and receiving the RINGING response.

3.3 Performance Measurements

The measurements of the registration times, call establishment times and communication delays in this section were all averaged from 10,000 samples dispersed over a 10 day period i.e. $n = 1000$ samples a day.

Registration

Examining equations 1a and 2a, we deduce that $t_C \leq t_A$ for sufficiently large n. In the general case of a roaming user, the round-trip delay $2y$ is expensive. Our real-time measurements are listed in table I. Clearly our measurements show that equations 1a and 2a hold for an $\alpha \approx 39ms$. We see here that t_A is approximately $5t_C$ or $7t_C$ for the virginia and panama cases respectively. Obviously, the H-SIP approach outperforms the traditional approach as long as y is significant, which is often the case for roaming subscribers. The value of α directly depends on the implementation of the SIP server which is the JAIN-SIP Proxy server [26] in this case. Besides, x is random and it directly depends on the location of the Local Handle Server (LHS) [11] storing the particular *handle*.

Table II: (a) Comparison of average call setup delays (as shown in Fig. 6) (b) Average Round-trip communication delays , transport=UDP

Call Setup Delays [ms]		
$\Delta\lambda$	$\Delta\phi$	$\Delta\mu$
88	168	106

(A)

Round-trip Delays [ms]		
τ_1	τ_2	x
111	47	85

(B)

Call Establishment

In comparing the call establishment time given by equations 3d and 4d, we are interested in computing the performance enhancement or degradation $\Delta\lambda = \lambda - \lambda^{'}$. We consider the following variables to verify our model:

$$\tau_1 = rt_{(hdomain,r\_user)} - rt_{(fdomain,r\_user)} \gg 0 \tag{5a}$$

$$\tau_2 = rt_{(cdomain,hdomain)} - rt_{(cdomain,fdomain)} \tag{5b}$$

$$\sigma = \tau_1 + \tau_2 > 0 \tag{5c}$$

where σ is the difference in the cumulative round-trip delay between the no H-SIP case and the H-SIP case respectively. Equation 5a is true in general due to the presumably large geographical separation between the roaming user and his home server versus using a local server. We also argue that equation 5c holds in general based on our model i.e. the cumulative round-trip delays for *c_user* to reach *r_user* is smaller in the H-SIP scenario. However, τ_2 in 5b can be positive or negative since it directly depends on the *cdomain-hdomain* separation versus that of *cdomain-fdomain*. For our particular setup, $\tau_2 > 0$.

It follows from equations 3 and 4 that,

$$\Delta\mu = \mu - \mu^{'} \approx \tau_1 \tag{6a}$$

$$\Delta\phi = \phi - \phi^{'} \approx \tau_2 + \Delta\mu \tag{6b}$$

$$\Delta\lambda \approx \Delta\phi - x \tag{6c}$$

Notice here that,

$$\Delta\lambda \approx \tau_1 + \tau_2 - x = \sigma - x \tag{6d}$$

Consequently, H-SIP outperforms the traditional SIP approach, in general, as long as $\sigma > x$. Our conducted real-time measurements are listed in Table II. In Table II, we verify the validity of our model by separately comparing the real-time call setup measurements to the round-trip delays, thus asserting equations 6.

DNS vs Handle-DNS resolution

Obviously, the only performance degradation introduced by our approach is the *handle* resolution overhead. This overhead has been extensively measured by CNRI and its partners and is illustrated in Figure 7. The current performance of the system that oscillates from 3 to 10 ms makes it comparable to the Bind implementation of the DNS protocol. Additionally due to its intrinsic fully distributed administration and resolution, it avoids the pitfalls that plage the current DNS implementation where DNS resolution times can extend to over 100 milliseconds [29].

Overall robustness of the handle-DNS implementation has also been extensively tested along with load assessment. Results, as shown in Figure 8, show that the Handle System can efficiently replace the DNS system .

On top of addressing the domain resolution itself, our approach minimizes the signaling traffic needed by a roaming user to join the SIP infrastructure and be ready to initiate calls. The user is efficiently utilizing the services of a local server with no need for per-call coordination with his home server. If the home server is

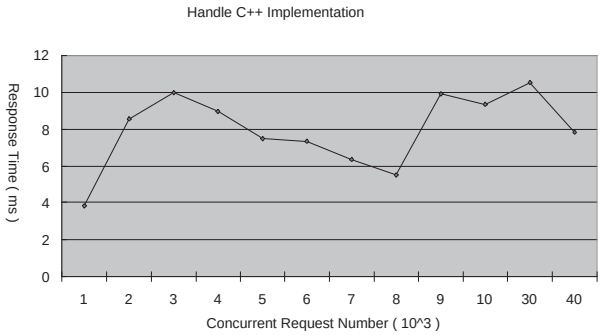


Figure 7: *Handle* implementation performance measurement as of August 2005. Acquired through the courtesy of Mr. Sam Sun and CN-NIC.

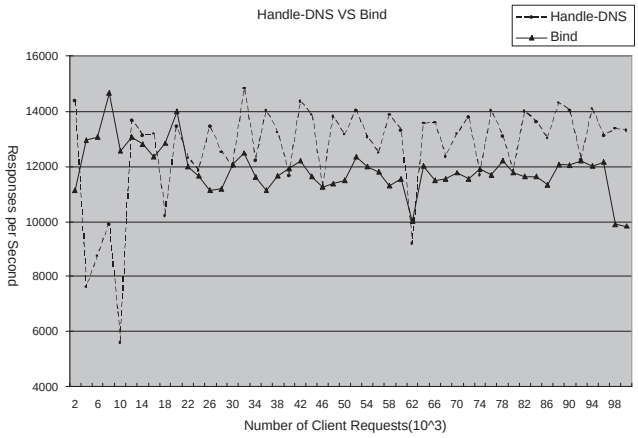


Figure 8: *Handle* load performance measurement as of August 2005. Acquired through the courtesy of Mr. Sam Sun and CN-NIC.

located in another continent per se, the round trip times for registration/redirect messages from the roaming user to home proxy/correspondent user respectively become significant. The proposed approach optimizes the association, authentication and call routing times for the roaming user by guaranteeing that the mobile user will always be addressed through the closest available server in his vicinity.

4 Future Work

This paper used globally unique *handle*s to identify SIP users. It is also possible to use a set of aliases of such *handle*s that can in turn translate into *handle*s. Such a service would be administered by yet another service provider using the *handle* protocol to administer and coordinate the general use of aliases in the system. This may be done using registry-like features for particular *handle*s, and allows the provider to service users that opt into this service. Since *handle*s can also point to other *handle*s, certain intermediate *handle*s may be provided to systems that choose to run private *handle* servers in a way similar to a PBX. The advantage of having a distributed resolution infrastructure that is also domain independent, translates into users being able to run smaller SIP servers that communicate with the Handle System to expedite routing. You could even envision a cellular-like behavior

for SIP systems in which users are able to use the resources of many smaller SIP servers along a user's roaming path. All these ideas need to be investigated as well as possible ways to expedite routing and *handle* resolution are possible future research paths.

We intend to leverage the scalable resolution, security, and administration services of the Handle System and use it to replace the DNS system within the SIP protocol. According to RFC 3263, the main reason SIP needs to use DNS is to enable the originator domain proxy to locate the SIP proxy in the destination domain (IP, port and transport protocol). The other need for DNS in SIP is for the terminating proxy to identify a backup for the originating proxy in the case the latter fails. Replacing the DNS within SIP requires carefully examining all the DNS resolutions performed by UA clients and proxy servers according to RFC 3263 and formalizing the fields to be resolved within the *handle*s. Besides, we envision the future Handle System to be completely decentralized and to be based on concepts like Distributed Hash Tables, where a particular identifier is not necessarily required to be located under a domain hierarchy. Finally, part of our current research is to focus on implementing a structured peer-to-peer form of the Handle System to expedite lookups and resolutions and eliminate single points of failure.

5 Conclusion

In this paper, we outlined the use of an indirection architecture based on the Handle System to address SIP inter-domain mobility. Our approach not only enables roaming controlled by the users rather than organizations, but also provides a faster implementation than traditional approaches currently deployed. Through our work, users are able to dynamically enable their own mobility and benefit from the advantages of a secure distributed persistent identifier network. By disassociating users from DNS domains, while still providing the means to interact with traditional SIP systems, we provide a scalable interchangeable enhancement to the SIP infrastructure.

6 Acknowledgements

We would like to thank the Corporation for National Research Initiatives CNRI for their support as well as the Ibero American Science and Technology Consortium (ISTEC) and the University of New Mexico that provided the VoIP infrastructure for the test bed of this work.

References

[1] J. Rosenberg, H. Schulzrinne, and etal., "RFC 3261: Session initiation protocol," June 2002.

[2] "H.323 : Packet-based multimedia communications systems," http://www.itu.int/rec/T-REC-H.323-200307-I/en.

[3] E. Wedlund and H. Schulzrinne, "Mobility support using sip," in *WOWMOM '99: Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*. New York, NY, USA: ACM Press, 1999, pp. 76–82.

[4] H. Schulzrinne and E. Wedlund, "Application-layer mobility using sip," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 4, no. 3, pp. 47–57, 2000.

[5] R. Pandya, "Emerging mobile and personal communication systems," *IEEE Communications Magazine*, vol. 33, pp. 44–52, June 1995.

[6] A. Dutta, F. Vakil, J. cheng Chen, M. Tauil, S. Baba, N. Nakajima, and H. Schulzrinne, "Application layer mobility management scheme for wireless internet," Apr. 15 2001.

[7] C. E. Perkins, "RFC 3220:ip mobility support for ipv4," January 2002.

[8] T. Berners-Lee, R. Fielding, and L. Masinter, "RFC 2396:uniform resource identifiers (URI): Generic syntax," 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2396.txt

[9] N. Banerjee, S. K. Das, and A. Acharya, "SIP-based mobility architecture for next generation wireless networks," in *PerCom*. IEEE Computer Society, 2005, pp. 181–190.

[10] S. Sun, L. Lannom, and B. Boesch, "Handle system namespace and service definition," RFC 3651, November 2003.

[11] S. Sun, L. Lannom, and B.Boesch, "Handle system overview," RFC 3650, November 2003.

[12] S. Sun, S. Reilly, L. Lannom, and J. Petrone, "Handle system protocol (ver2.1) specification," RFC 3652, November 2003.

[13] "The handle system," http://www.handle.net.

[14] R. Jain, T. Raleigh, D. Yang, L.-F. Chang, C. Graff, M. Bereschinsky, and M. Patel, "Enhancing survivability of mobile internet access using mobile IP with location registers," in *INFOCOM*, 1999, pp. 3–11.

[15] R. Jain, T. Raleigh, C. Graff, M. Bereschinsky, and M. Patel, "Mobile internet access and qos guarantees using mobile ip and rsvp with location registers," vol. 3. ICC International Conference on Communications, June 1998, pp. 1690 – 1695.

[16] K. Wong, A. Dutta, J. Burns, R. Jain, K. Young, and H. Schulzrinne, "A multilayered mobility management scheme for auto-configured wireless ip networks," *Wireless Communications*, vol. 10, no. 5, pp. 62–69, October 2003.

[17] D. Johnson, C. Perkins, and J. Arkko, "RFC 3775: Mobility support in ipv6," June 2004.

[18] K. Wong, A. Dutta, H. Schulzrinne, and K. Young, "Simultaneous mobility: analytical framework, theorems, and solutions," *Wireless Communication and Mobile Computing*, June 2006.

[19] C. Hongtao, Y. Fangchun, and X. Peng, "Analysis on sip mobility of double user agent servers," in *Communications and Information Technology*, vol. 1, ISCIT. IEEE, October 2005, pp. 87–90.

[20] H. Jerez, J. Khoury, and C. Abdallah, "A mobile transient network architecture," 2006, pre-print available at https://dspace.istec.org/handle/1812/55.

[21] J. Khoury, H. Jerez, N. Nehme, and C. Abdallah, "An application of the mobile transient network architecture: Ip mobility and inter-operability," 2006, pre-print available at http://hdl.handle.net/2118/jk_transapp_06.

[22] R. Kahn and R. Wilensky, "A framework for distributed digital object services," Internet Whitepaper http://www.cnri.reston.va.us/k-w.html, January 1995.

[23] S. Sun, "Establishing persistent identity using the handle system," Tenth International World Wide Web Conference, May 2001.

[24] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP authentication: Basic and digest access authentication," RFC 2617, June 1999.

[25] K. Harrenstien, M. K. Stahl, and E. J. Feinler, "RFC 952: DoD Internet host table specification," Oct. 1985.

[26] "Jain-sip proxy (built on jain-sip 1.1 api)," https://jain-sip-presence-proxy.dev.java.net/.

[27] Y. Rebahi, D. Sisalem, J. Kuthan, A. Pelinescu-Onicicul, B. Iancu, J. Janak, and D. Mierla, "The sip express router, an open source sip platform," http://www.iptel.org/ser.

[28] "Sip-communicator 1.0," https://sip-communicator.dev.java.net/.

[29] C. Huitema and S. Weerahandi, "Internet measurements: the rising tide and the dns snag." in *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement Modeling and Management*, ser. IPseminar. Monterrey, CA, USA: ITC, September 18-20 2000.