8-6-2007

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

Chaouki T. Abdallah

Joud S. Khoury

Henry N. Jerez



## Recommended Citation

Abdallah, Chaouki T.; Joud S. Khoury; and Henry N. Jerez. "Efficient User Controlled Inter-Domain SIP Mobility Authentication, Registration, and Call Routing." *Fourth Annual International Conference on Mobile and Ubiquitous Systems* (2007): 1-7. doi:10.1109/MOBIQ.2007.4451064.

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

Joud S. Khoury * Henry N. Jerez† Chaouki T. Abdallah*

*School of Electrical and Computer Engineering, 1 University of New Mexico, Albuquerque, NM 87131-0001
{jkhoury, chaouki}@ece.unm.edu
†Corporation for National Research Initiatives, Reston, VA 20191
hjerez@cnri.reston.va.us

***Abstract*— Over the past decade, multimedia services have gained significant acceptance and played an important role in the convergence of IP networks. The proliferation of mobile devices and the nomadic user and computing lifestyles on current networks make *mobility support* a crucial ingredient of current IP-based multimedia systems. 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, and the SIP infrastructure is anticipated to be extensively deployed all over the Internet. We have lately proposed an approach to inter-domain SIP mobility which we call H-SIP. H-SIP is a user-controlled mobility scheme that improves personal and terminal mobility. H-SIP uses persistent identifiers and leverages the traditional SIP architecture to abstract any domain binding from users. This paper expands on our previous work and experimentally proves the efficiency of H-SIP in achieving inter-domain authentication and call routing through modeling and real-time measurements.**

## I. Introduction

The industry has recently witnessed a rapid increase in the popularity and deployment of packet switched multimedia services due to its low cost and broader data services. Additionally, mobile operators are currently promoting simultaneous Cellphone/Wi-Fi access [1] by introducing dual-mode phones that switch between the cellular network and the IP packet switched network. Efficient user mobility and identity persistence issues become critical in such networks and need to be carefully addressed in this context.

SIP [2] is a signaling and control protocol for handling multimedia sessions. The SIP architecture is 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 multimedia communications and mobility leverages the issues emanating from SIP users *roaming* across multiple SIP domains. Briefly, SIP handles user location through the use of a proxy/location server 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 due to its location independence. However, *the SIP service model and identification scheme defines a user only within the domain boundaries of the service provider* i.e. 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 the user's 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.

The Handle-SIP (H-SIP) [9] proposal addresses the inter-domain mobility and identification issues by introducing an abstraction framework on top of the traditional SIP architecture. User location and association are abstracted through the use of globally unique and persistent identifiers called *handles* that are part of the Handle System [10], [11], [12], [13]. This allows efficient load distribution and traffic routing as well as seamless and secure roaming. The major premise of H-SIP is the widespread deployment of the SIP infrastructure over the Internet. H-SIP solely enhances personal and terminal mobility [5] in traditional SIP architectures. It may be gradually deployed and fits seamlessly within the traditional SIP architecture allowing users to transparently roam across different SIP domains. This paper expands on our previous work with H-SIP [9] and experimentally proves the efficiency of H-SIP in achieving inter-domain authentication and call routing through modeling and real-time measurements.

The rest of the paper is organized as follows. Section II briefly overviews the principal design aspects of H-SIP and the mechanisms it employs to implement inter-domain authentication, registration and call routing. Section III details the implementation test-bed and section IV discusses the experimental model used for the evaluation of H-SIP. We then describe the performance results in section V. Finally,

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

Section VI outlines our future work before we conclude in section VII.

## II. Overview of H-SIP Mobility Scheme

This section briefly overviews the H-SIP mobility scheme [9].

### A. Sessions and Mobility

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 her 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 her home proxy server. Figure 1 identifies this traffic as traditional traffic flow. All requests to/from the roaming user must go through the central home proxy server. Clearly, if the user is present in another country, her traffic would still have to go through her central home proxy (triangle routing) as depicted in Figure 1, 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. 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.

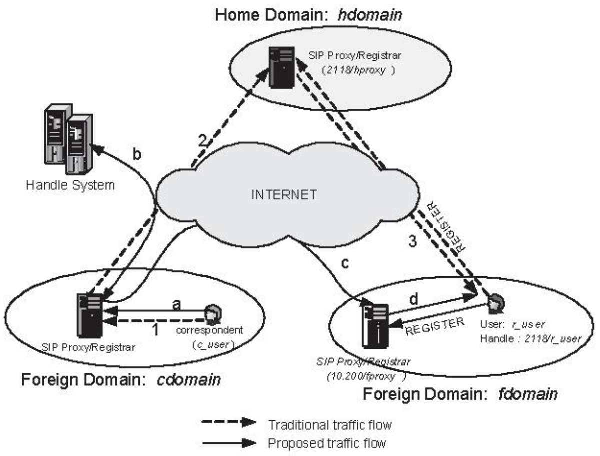


Fig. 1. A Reference inter-domain roaming scenario

Obviously, 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.

### B. Abstraction layer

H-SIP uses *handles* as globally unique identifiers to locate and identify SIP architectural elements. Users will identify each other, as well as the SIP servers they associate with using *handles* instead of URIs and domain names respectively. This abstraction allows the system to route calls independent of user location and domain association. A *handle* is a persistent name that can be associated with a set of attributes. Some of these attributes can describe location, permissions, administrators and state. The fact that *handles* are defined independently of any of the attributes or public keys of the underlying objects, makes them persistent identifiers [14].

Briefly, the Handle System [10], [11], [12], [13] is intended to be a means of universal basic access to registered digital objects [15]. It provides a distributed, secure and global name service for administration and resolution of *handles* over the Internet. The Handle System is extremely scalable[1] and suitable to operate in highly mobile environments.

A possible realization of the *handles* for user *r_user* and proxy server *fproxy* is depicted in Figure 2. The *handle* has several fields including administration and security information (public or secret keys). Administrators have the privilege to modify the fields inside the *handle* provided the administrator succeeds to authenticate to the Handle System using the private/secret key.

A. handle: 2118/r_user

| Field Type:index | Value |
|---|---|
| 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........ |

B. handle: 10.200/fproxy

| Field Type:index | Value |
|---|---|
| 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........ |

Fig. 2. Sample *handle* structures, A. user handle B. proxy handle (dashed arrow shows relation between the two handles)

### C. Authentication and Registration

Recall that with the H-SIP [9] abstraction layer, each user owns and administers her own *handle*, specifying the set of administrators (also *handles*) that have rights over her *handle*. Among those administrators, the user should include the *handle* of any SIP proxy server that she wishes to register with, which could be any foreign server(s) that she trusts (dashed arrow in Fig. 2 shows that *r_user* trusts *fproxy*). Authentication proceeds by the user providing credentials to the local proxy (Figure 1), which in turn validates the credentials against the Handle System. Note that the user has picked the password and made it part of her handle (Fig. 2.A, SIP_PWD), thus eliminating the need to maintain internal accounts on the proxy server. The user's credentials are valid for all realms provided the correct administrative privileges

[1] The largest individual Handle System implementation to date is deployed at the Los Alamos National Laboratory, which is intended to support more than a half billion identifiers while providing internal resolution services to one of the largest archival collections in the United States.

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.

### D. Call Routing

After authenticating and registering with the foreign server, the user's handle-to-URI mapping (Fig. 2.A, SIP_URL) remains fresh allowing correspondent users to reach her simply by addressing her *handle*. From the perspective of a user, all other users seem to belong to one local domain and abstraction is complete. In the scenario of Figure 1, call routing proceeds when the caller *c_user* tries to INVITE the roaming user *r_user* (callee) using the latter's *handle 2118/r_user* as shown in Figure 1 (arrows a,b,c,d). The proxy server resolves the *handle* into the SIP_URL field which is *2118.r_user@x.y.z.w*, and forwards the call accordingly using the natural SIP call flow. Consequently, H-SIP eliminates the need to 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.

### E. User-Controlled Mobility

The H-SIP mobility scheme is user-controlled in the sense that the user is responsible for the administration of her 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*[2]) 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 grant her access provided sufficient credentials exist within the user's persistent identifier. These credentials can include trust information as well as financial information that the actual server can validate before allowing the user to route traffic through.

## III. Implementation and Test-bed

We have implemented the H-SIP functionality as an extension to two open source SIP servers, the JAIN-SIP Proxy [16] and the SIP Express Router (SER) [17]. 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. Our implementation is 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 *handles*. All the results depicted hereafter are based on the JAIN-SIP proxy. Note that the H-SIP mobility scheme requires no modifications to the User Agents (UAs) and is solely deployed through extensions to the SIP proxy servers.

Our test-bed is a realization of the roaming framework depicted in Figure 1. The roaming user *r_user* is located in a foreign domain *fdomain* while her home domain is *hdomain*. We study the registration process that follows from the model. We also study the call establishment process in which a correspondent user *c_user* attempts to establish a call with the roaming user.

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 *handles*. 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 *handles* instead of domain names, we have implemented a specialized gateway that translates between 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 *handles*. 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 INET_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 [18]. The ease of deployment of our framework and success of the conducted experiments encourages us to pursue this work.

## IV. 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 her home SIP server irrespective of roaming, while the former case represents our proposed H-SIP approach. Both cases were tested under the same conditions regarding UA/SIP processor speed and work loads. As User Agents, we used the Java SIP

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

Communicator [19]. We also tested with the Cisco 7940/7960 IP phones.

### A. Registration

Figure 3 shows a comparison of the data flow involved in the registration process for roaming (H-SIP) and non-roaming (traditional SIP) environments. The roaming user *r_user* is located in *fdomain*, *ece.unm.edu*. For the non-roaming case, we tested with *hdomain* being either *cnri.reston.va.us* (Reston, Virginia) or *istec.org* (Panama). Registration here assumes the basic digest authentication [20] with the proxy server. The model considers how the UA refreshes its registration with the proxy/registrar continuously (we used a registration TTL of 1 minute). If we estimate the server's average processing time

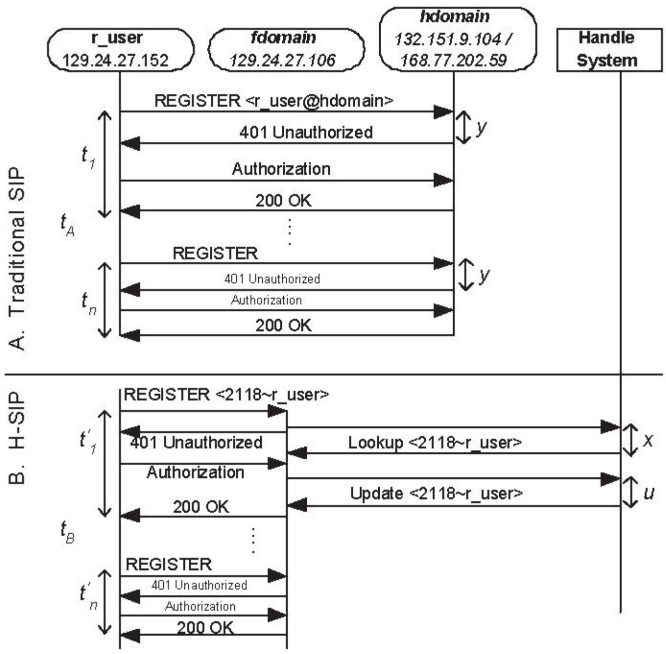


Fig. 3. REGISTER message flow, A. Traditional SIP versus B. H-SIP

of the REGISTER request including digest authentication by $\alpha$ , then from Figure 3.A, the average registration time $t_A$ as seen by *r_user* is given by,

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

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 her home SIP server.

Now, if we consider H-SIP, then from Figure 3.B, the average registration time $t_B$ as seen by *r_user* is given by,

$$t_B = \frac{1}{n} \sum_{1 \leq i \leq n} t'_i \tag{2a}$$

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 ($t'_2$ to $t'_n$) will read the cached value. So, for $i > 1$, $x = u = 0$.

Here, $t_A$ and $t_B$ are measured by *r_user* as the average time between sending the REGISTER request and receiving the 200 OK response. In Figure 3, *r_user* performs a DNS lookup or a Handle-DNS lookup in scenarios A and B 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 V-C.

### B. Call establishment

Figure 4 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*. Again we compare the INVITE message flow of H-SIP to the traditional SIP. 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 4. We focus on the INVITE message flow and the servers require no authentication. We denote by $rt_{(m,n)}$ the round-trip

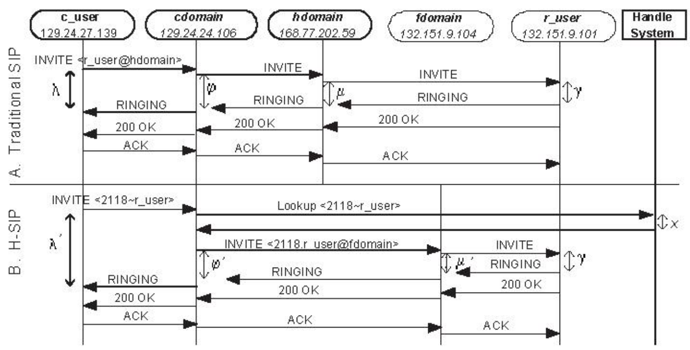


Fig. 4. INVITE message flow, A. Traditional SIP versus B. H-SIP

communication delay between nodes $m$ and $n$. For example, in Figure 4, $rt_{(cdomain,hdomain)}$ 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 $\beta$, and the UA's average processing time of the INVITE request by $\gamma$, then again from Figure 4.A,

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

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

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

and therefore,

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

where $\lambda$ is the average call establishment time (the INVITE/RINGING round-trip) as seen by *c_user*, $\phi$ is the average INVITE/RINGING round-trip time between the *cdomain* and *hdomain* SIP servers and $\mu$ is the average INVITE/RINGING round-trip time between the *hdomain* SIP server and *r_user*.
Now, if we consider H-SIP, then from Figure 4.B,

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

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

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

and therefore,

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

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 $\lambda$ and $\lambda^{'}$ are measured by *c_user* as the time between sending the INVITE request and receiving the RINGING response.

## V. 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[3].

### A. Registration

Examining equations 1a and 2a, we deduce that $t_B \leq t_A$ 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 $5t_B$ or $7t_B$ 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 $\alpha$ directly depends on the implementation of the SIP server which is the JAIN-SIP Proxy server [16] 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*.

[3]The experiments were conducted during the month of December in 2006

TABLE I

Average Registration delays with and without H-SIP (as shown in Fig. 3), $n = 1000$, transport=UDP

| | | Registration Delays [ms] | | | |
|---|---|---|---|---|---|
| | | $x$, $u$ | $y$ | $t_A$ | $t_B$ |
| traditional SIP | hdomain: virginia | NA | 86 | 209 | NA |
| | hdomain: panama | NA | 116 | 275 | NA |
| H-SIP | | 84, 260 | 0 | NA | 40 |

### 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:

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

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

$$\sigma = \tau_1 + \tau_2 > 0 \quad (5c)$$

where $\sigma$ is the difference in the cumulative round-trip delay between the traditional 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 her 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, $\tau_2$ in equation 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,

$$\Delta\mu = \mu - \mu^{'} \approx \tau_1 \quad (6a)$$

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

$$\Delta\lambda \approx \Delta\phi - x \quad (6c)$$

Notice here that,

$$\Delta\lambda \approx \tau_1 + \tau_2 - x = \sigma - x \quad (6d)$$

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.

### C. DNS vs Handle-DNS resolution

Obviously, the only performance degradation introduced by our approach is the *handle* resolution overhead. The *handle* resolution measurements that this paper reflected are all

TABLE II

(A) COMPARISON OF AVERAGE CALL SETUP DELAYS (AS SHOWN IN FIG. 4) (B) AVERAGE ROUND-TRIP COMMUNICATION DELAYS , TRANSPORT=UDP

| Call Setup Delays [ms] | | | Round-trip Delays [ms] | | |
|---|---|---|---|---|---|
| $\Delta\lambda$ | $\Delta\phi$ | $\Delta\mu$ | $\tau_1$ | $\tau_2$ | $x$ |
| 88 | 168 | 106 | 111 | 47 | 85 |
| (A) | | | (B) | | |

based on the current JAVA implementation of the Handle System [13], which is relatively slow[4]. However, a more powerful and faster C++ implementation is available but is not public domain yet. The C++ Handle System implementation performance has been extensively measured by CNRI and its partners (CN-NIC) and is illustrated in Figure 5. The 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 [21]. Clearly, using the C++ instead of the JAVA implementation of the Handle System will allow H-SIP to further outperform the traditional SIP performance. Overall robustness of the handle-DNS imple-

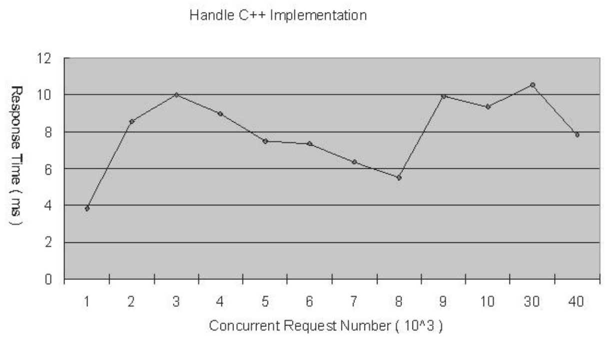


Fig. 5. *Handle* implementation performance measurement as of August 2005. Acquired through the courtesy of Mr. Sam Sun and CN-NIC.

mentation has also been extensively tested by CNRI along with load assessment. Results, as shown in Figure 6, are very encouraging as they show that the Handle System can potentially replace the DNS system.

## VI. FUTURE WORK

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

[4]Our measurements show an average resolution time of around 83ms using the Handle JAVA library.

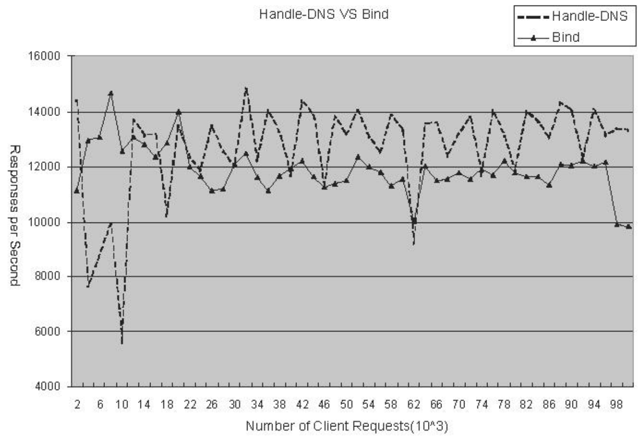


Fig. 6. *Handle* load performance measurement as of August 2005. Acquired through the courtesy of Mr. Sam Sun and CN-NIC.

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. 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.

## VII. CONCLUSION

In this paper, we have reviewed the H-SIP mobility scheme and experimentally proven its efficiency in achieving inter-domain authentication and call routing through modeling and real-time measurements. On top of addressing the domain resolution itself, the H-SIP 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 her home server. If the home server is 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. H-SIP optimizes the association, authentication and call routing times for the roaming user by always trying to address the mobile user through the closest available server in her vicinity.

## VIII. 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] S. Lawson, "T-mobile merges wi-fi, cellular." [Online]. Available: http://www.networkworld.com/news/2006/102406-t-mobile-merges-wi-fi.html

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

[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] J. Khoury, H. Jerez, and C. Abdallah, "H-sip inter-domain sip mobility: Design," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 2007 4th IEEE*, Las Vegas, NV, January 2007.

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

[15] 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.

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

[17] 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.

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

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

[20] 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.

[21] 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.