Internet Engineering Task Force
INTERNET-DRAFT
draft-schulzrinne-sip-register-01.ps

SIP WG
Schulzrinne
Columbia U.

April 16, 2001
Expires: October 2001

SIP Registration

Status of this Memo

Copyright Notice

Abstract

SIP registration provides personal, pre-call terminal and service mobility. We describe the registration process in detail, considering different options for roaming users.

1 Introduction

The SIP [1] REGISTER request is the core mechanism for supporting personal, service and pre-call terminal mobility.Here, we define *personal mobility* as the ability for a user to be reachable under the same identifier while using different terminals, possibly several at the same time. *Service mobility* refers to the ability to obtain the same services regardless of where a user may be roaming.For VoIP services, service mobility may include the ability to use the same speed dial functionality, preprogrammed user interface elements and possibly even the user interface itself even when using a terminal owned by a third party, e.g., an Internet "payphone" or kiosk. A related aspect of service mobility is the ability to maintain the same set of services when changing providers or proxies. This is supported, for example, by uploading CPL [2] or cgi [3] scripts to the local proxy server, either via HTTP or as REGISTER bodies [4]. This aspect is beyond the scope of this document.

Pre-call terminal mobility describes the ability of a terminal to dynamically acquire IP addresses, but remain reachable under the same application-layer identifier.

This document does not add functionality to SIP [1]. Rather, it spells out in more detail possible implementations and suggests where additional functionality is needed.

2 Assumptions

We assume that each terminal is configured with a user address, a SIP URI, such as `alice@wonderland.com`. This identifier may be embedded by configuration into the communications device (e.g., for an Ethernet

phone, personal laptop or workstation), established via local login into a shared computer or associated temporarily with a device by some token carried by the user. Examples of such tokens include SIMs, smartcards, iButtons, PDAs or magnetic swipe cards.

This address also implies a *home registrar*, where the home registrar is derived by the DNS SRV [5] lookup of the host portion of the SIP URI, `wonderland.com` in this example.

In addition, each device has at least a temporary network address which can be used to identify it during a session. This address is provided in the Contact header. However, the temporary address may not be directly reachable by everybody, due to firewalls and network address translators.

Networks are identified by their domain name, independent of whether they belong to the same autonomous system, multicast scope or link-layer local area network. The same physical network may share several such domains. For example, while `cs.columbia.edu` and `columbia.edu` are part of the same autonomous system and organization, but they are different domains. `hgs@cs.columbia.edu` would be *visiting* the `columbia.edu` domain as soon as he obtains a Columbia, rather than Computer Science, IP address. A user's local domain is defined by the domain name option configured via DHCP. Some domains do not have a DHCP server, such as the addresses administered by virtual SIP domains [TBD: need better terminology - this refers to domains such as yahoo.com or hotmail.com that offer the equivalent of web-based email, without any association to a physical network.]

We define a *traveling user* or *visitor* as a SIP end point that is visiting a domain other than the domain indicated in its SIP URI. This could be a mobile device with an embedded identifier visiting a foreign network or a local device that is personalized via a token of a visitor, as described above. Thus, a SIP payphone would always be a visitor.

The outbound proxy and registrar server in the visited network are called the *local proxy* and *local registrar*, respectively. That network is referred to as the *visited network*, while the user's domain is called the *home network*, which has *home proxy* and *home registrar*.

In any network, a SIP end system needs to establish two SIP-related configuration parameters, namely the local registrar and whether there is an outbound proxy. There are many possible ways this information can be configured, but manual configuration is ill-advised. It is RECOMMENDED that the end system obtain local proxy information via the SIP server DHCP option [6]. In this approach, the local proxy is assumed to know where the local registrar is located, if it is not co-located with the proxy.

In the absence of DHCP or manual configuration, a SIP end system has to assume that there is no outbound proxy.

3 Registration in Visited Network

In the examples, we let `alice@wonderland.com` visit the network `visited.net`.

Home registration only: In this model, the visiting user simply acquires a local IP address in the visited network and sends a registration with a Contact header indicating that address.

```
REGISTER sip:wonderland.com SIP/2.0
To: <sip:alice@wonderland.com>
From: <sip:alice@wonderland.com>
Contact: sip:alice@128.59.16.1
```

It makes no difference here whether the visited network provides SIP services or not. An outbound

proxy can be used, but it simply forwards the REGISTER request based on its request URI.

This approach works only if the visited network does not use a firewall. It also means that every location update has to go back to the home network. (This is likely to matter only if IP address changes are frequent.)

The proxy in the visited network can still provide localized services such as emergency calling [7] by remapping these addresses.

Outbound proxy intercept: Here, the outbound proxy intercepts the registration request and any other outbound requests and changes the Contact address to its own address. It also has to forward the request to the local registrar. It has to create a new temporary user identifier that allows it to identify incoming requests for that visiting user. This could be a random identifier or the concatenation of the visitor's address and the proxy's domain, such as `alice%40wonderland.com@visited.net`, where the %40 is the URL-escaped "@" symbol. We call the latter the canonical visitor name. (The proxy cannot just replace the host part and keep the user identifier as there may be several users, local and visiting, by that name.)

This approach has the advantage that it forces incoming requests to use the proxy server and thus solves the firewall problem.

If the registrar and proxy are not co-located, the REGISTER request forwarded to the registrar has to use the "real", local Contact address and the REGISTER request forwarded to the visitor's home address contains the address of the visited proxy.

A rogue user can easily override the registration of the visiting user, although the proxy can provide some security by discarding any registrations where the registration fails in the visiting user's home network. Thus, the visited registrar MUST only act on the registration after a 200 (OK) response has been returned by the home registrar. This approach is vulnerable to response spoofing, unless the response is also authenticated by Digest authentication or cryptographic signatures.

The visiting user could also provide a random basic password when first registering and then be forced to re-use this secret on subsequent registrations. This would limit registration spoofing to those intruders that can snoop the initial registration. A Diffie-Hellman generated key may also be useful, as long as the intruder cannot insert itself into the middle of the registration exchanges. It is probably safest if the local proxy has access to the local AAA mechanism, as that mechanism has verified the visiting user and knows which IP address has been assigned to it.

As a simple precaution, proxies in visited networks can simply disallow changes of IP addresses for visiting users; however, that then only allows a single instance of a visiting user per visited network.

User-initiated proxy registration: This is a variation of the previous approach. The visitor recognizes that it is in a foreign network by comparing its URI domain to the domain returned by DHCP in the domain name (Option 15, Section 3.17 of [8]) or the SIP server option [6]. If they differ, it uses the address of the SIP server returned by the DHCP SIP server option as its Contact address. This assumes that this address is externally reachable, but even if the domain has it own local DNS and address space, only the name has to be the same, as it will be resolved by DNS SRV records. In most cases, this entry will simply be the domain name.

The outbound proxy server intercepts the REGISTER request and updates its internal registration.

User-initiated proxy registration has the advantage that it does not interfere with cryptographically signing registration requests. However, it does require minor adjustments in SIP UAs and additional functionality in SIP registrars.

To avoid adding numerous configuration options, this only works if *all* outbound proxy servers can handle such registrations without prior configuration of the user identifier. This method has the same spoofing vulnerability as the previous one.

Dual registration: In dual registration mode, the visiting UA sends two REGISTER requests, one to the local registrar, e.g., via multicast or the DHCP-configured outbound proxy, and another to the home registrar. The registration to the local registrar uses the canonical visitor name to avoid collisions, while the registration at home follows the same rules as the "user-initiated proxy registration" case, except that the proxy server can simply proxy the REGISTER request not addressed to it, rather than having to also interpret it.

This approach has the advantage that error handling is simplified, as each registration operation can fail individually.

However, the visitor generally has no credentials to authenticate the local registration, unless the registrar and UA somehow "borrow" credentials from some AAA mechanism, e.g., a CHAP secret. This is not likely to work across network types. (For example, it does not work in the common case where visitors are allowed to plug in laptops in a local area network while visiting a university or research lab.)

This approach has the disadvantage that it requires two messages between UA and local registrar, which is undesirable particularly for bandwidth-constrained environments. It also requires changes in current SIP UAs.

Third-party registration: The home registrar registers the visitor in the visited network, supplying its own credentials. The home registrar uses the domain name supplied in the Contact header of the visitor. This can obviously only work if the UA supplies a domain name rather than a numeric IP address.

This approach has the fundamental architectural flaw that the home registrar is now acting as a UA.

Note that while the first INVITE in a session uses the outbound proxies, the regular Route mechanism ([1], Section 6.38) takes over for subsequent requests.

4 Aliases

Often, SIP UAs have several names, such as a SIP URI derived from the user's email address (e.g., `alice@wonderland`, a name reflecting a telephone extension (e.g., `4567@wonderland.com`) to ease dialing on IP phones equipped only with a numeric keypad and possibly an E.164 address (e.g., `1-212-555-4567@wonderland.com`).

UAs may not always know their domain name, so that configurations derived from user logins may produce identifiers such as `alice@rathole.wonderland.com`. However, for registration in the visited network, the proxy or registrar in the visited network has no way of knowing whether these two identifiers are indeed the same user, so that these two identities cannot be mapped to the same registration.

Particularly with telephone extensions, some care needs to be taken, since extensions have traditionally referred to physical lines, not users. Thus, the extension may be associated with a particular device or line.

Rather than making aliases visible at the protocol level, it may be preferable to have the SIP UA simply register the same Contact for each of these aliases. The registrar then uses the user profile or rewriting rules to associate several different To values with the same internal registration record.

Similarly, the location server MAY also, without registration, translate the request URI in incoming requests from various alias forms into a canonical user identifier. If the location server can perform this translation, it removes the need for multiple registrations. (TBD: are there cases where this is not the case?)

5 Home Services while in Visited Network

For some applications, the user would like to employ services of the home network while generating outbound requests in the visited network. The visiting UA needs to detect that it is in a foreign network and insert a Route header pointing to its home proxy server. The UA has to include the address of callee in the Route URI and the network address of the home server in the maddr parameter. For example, if Alice calls Bob, she would include the following in her outgoing requests:

```
From:  <sip:alice@wonderland.com>
Route: <sip:bob@macrosoft.com;maddr=wonderland.com>
```

The home proxy can either be configured statically, based on the user's From domain, as in the example above, or could be obtained via some configuration information. The details remain to be worked out.

6 SIP Naming

It is RECOMMENDED that a user have a single identifier for email, SIP and as a network access identifier (NAI) [9]. Thus, every SIP URI SHOULD also be usable as an email address. Note that this implies that the algorithm for resolving aliases in proxy servers and SMTP servers SHOULD be the same.

7 Registration Proxying

8 Registrar Redundancy

9 Stale Registrations

It can occur that a device does not have the opportunity to remove a registration for a particular IP address before being powered down or otherwise being unable to communicate. Registrations will expire automatically, but the expiration time can be sufficiently long that such "orphan" registrations can cause requests to be directed to a network address that has in the mean time been reassigned to another user.

Recipients of misdirected requests SHOULD respond with 404 (Not Found), which then allows the proxy to remove the registration.

Also, since registrations are additive, a UA that could not remove a registration at a previous network, will just add the new registration, causing requests to be forked to both the new and the "stale" registration. The UA will obtain all current registrations, but if a single user has multiple devices, it is not easy for the UA to detect stale registrations and remove them.

One possible solution is to add a unique "tag" parameter to the Contact header of REGISTER requests for those Contact fields where the UA is the authoritative source. The tag value is selected to be independent of the UAs current IP address and only depend on its device identity. Thus, tags are selected such that it never makes sense to have two registrations with the same tag value. The registrar keeps track of the tags associated with a registration and then replaces rather than adds registrations that duplicate existing Contact header tag values.

> Using the tag parameter in the To header field was considered, but since a registration may contain many Contact headers, it is not clear whether it should apply to all of them. This UAC-initiated use of the tag parameter also violates the UAS-initiated basic usage in other requests.

10 Security Considerations

It is RECOMMENDED that the user name in Basic and Digest is the same as the To header field, rather than a different user name, to simplify the use of global user databases in multi-domain SIP servers.

Digest authentication does not protect the Contact header against alteration by an adversary. This allows the adversary to redirect calls to another location if it can alter requests. The Authentication-Info header field contains a response digest, but it only protects the response entity body, not header fields. It may be feasible to create a new qop value, "auth-header", that includes all headers of the request except those marked with "c", "a" or "m" in Tables 4 and 5 of [1]. (TBD: this is not particularly easy to implement since it's not clear what to do with unknown headers. Do the kludge that only headers before Authorization are included?)

A2 = Method ":" digest-uri-value ":" H(entity-body) ":" H(e2e-headers)

11 Changes Since Version 00

- Clarification on local proxy configuration.

12 Acknowledgements

This draft is based on discussions with Jonathan Lennox, Jonathan Rosenberg and Stinson Mathai.

13 Author's Address

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
electronic mail: schulzrinne@cs.columbia.edu

References

[1] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: Session initiation protocol,” Internet Draft, Internet Engineering Task Force, Aug. 2000. Work in progress.

[2] J. Lennox and H. Schulzrinne, “CPL: a language for user control of internet telephony services,” Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

[3] J. Lennox, J. Rosenberg, and H. Schulzrinne, “Common gateway interface for SIP,” Internet Draft, Internet Engineering Task Force, May 1999. Work in progress.

[4] J. Lennox and H. Schulzrinne, “Transporting user control information in SIP REGISTER payloads,” Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

[5] A. Gulbrandsen, P. Vixie, and L. Esibov, “A DNS RR for specifying the location of services (DNS SRV),” Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.

[6] G. Nair and H. Schulzrinne, “DHCP option for SIP servers,” Internet Draft, Internet Engineering Task Force, Apr. 2000. Work in progress.

[7] H. Schulzrinne, “Providing emergency call services for sip-based internet telephony,” Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[8] S. Alexander and R. Droms, “DHCP options and BOOTP vendor extensions,” Request for Comments 2132, Internet Engineering Task Force, Mar. 1997.

[9] B. Aboba and M. Beadles, “The network access identifier,” Request for Comments 2486, Internet Engineering Task Force, Jan. 1999.

Full Copyright Statement