SIP
Internet-Draft
Intended status: Standards Track
Expires: July 28, 2008

J. Rosenberg
Cisco

January 25, 2008

TOC

Applying Loose Routing to Session Initiation Protocol (SIP) User Agents (UA)
draft-rosenberg-sip-ua-loose-route-02

Status of this Memo

Abstract

A key part of the behavior of the Session Initiation Protocol (SIP) is that SIP proxies rewrite the Request-URI as a request moves throughout the network. Over the years, experience has shown this to be problematic. It makes it difficult to use Request URI for service invocation, complicates emergency services, makes it more complex to support aliases, and so on. Architecturally, it confounds the concepts of address and route. This document proposes to change this through a new mechanism called UA loose routing.

Table of Contents

1. Introduction

TOC

A key part of the behavior of proxy servers in the Session Initiation Protocol (SIP) [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) is that they rewrite the Request-URI of requests as the request moves from the User Agent Client (UAC) to the User Agent Server (UAS). This is particularly true for requests outside of a dialog; requests within a dialog have their path dictated primarily by the Route header fields established by the Record-Routes when the dialog was initiated.

The most basic instance of this behavior is the processing executed by the "home proxy" within a domain. The home proxy is the proxy server within a domain which accesses the location information generated by REGISTER messages, and uses it to forward a request towards a UAC. Based on the rules in RFC 3261, when a home proxy receives a SIP request, it looks up the Request-URI in the location database, and translates it to the contact(s) that were registered by the UA. This new contact URI replaces the existing Request URI, and causes the request to be forwarded towards the target UA. Consequently, the original contents of the Request URI are lost.

In addition to routing of SIP requests based on the contents of the location database, proxies can employ other techniques. It is common in practice to have proxies which perform prefix and number analysis on

the Request URI against configured tables in order to do routing. It is also common practice to rewrite the Request-URI to point to an application server, again based on configured mappings.
Over the years, this practice of rewriting the Request-URI has proven problematic. Section 2 (Problem Statement) describes the problems with this mechanism. Section 3 (Architectural Roots of the Problem) analyzes the architectural issues which drive these problems. Section 4 (Alternative Solutions) discusses alternative solutions. Section 5 (Proposed Solution) describes a proposed solution to this problem, a technique coined 'UA loose routing'. [[OPEN ISSUE: A better name is needed here, since the mechanism applies equally well to targeting proxies.]]

2. Problem Statement

TOC

Several problems arise from the practice of rewriting the request URI.

2.1. Unknown Aliases

TOC

SIP user agents are associated with an address-of-record (AOR). It is possible for a single UA to actually have multiple AOR associated with it. One common usage for this is aliases. For example, a user might have an AOR of sip:john@example.com but also have the AORs sip:john.smith@example.com and sip:jsmith@example.com. Rather than registering against each of these AORs individually, the user would register against just one of them, and the home proxy would automatically accept incoming calls for any of the aliases, treating them identically and ultimately forwarding them towards the UA. This is common practice in the Internet Multimedia Subsystem (IMS), where it is called implicit registrations and each alias is called a public identity.
It is a common requirement for a UAS, on receipt of a call, to desire to know which of its aliases was used to reach it. This knowledge can be used to choose ringtones to play, determine call treatment, and so on. For example, a user might give out one alias to friends and family only, resulting in a special ring that alerts the user to the importance of the call.
However, based on the procedures in RFC 3261, when an incoming call hits the home proxy, the request URI (which contains the alias) is rewritten to the registered contact(s). Consequently, the alias that was used is lost, and cannot be known to the UAS.

TOC

2.2. Unknown GRUU

A variation on the problem in Section 2.1 (Unknown Aliases) occurs with Globally Routable User Agent URI (GRUU) [I-D.ietf-sip-gruu] (Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)," October 2007.). A GRUU is a URI assigned to a UA instance which has many of the same properties as the AOR, but causes requests to be routed only to that specific instance. It is desirable for a UA to know whether it was reached because a correspondent sent a request to its GRUU or to its AOR. This can be used to drive differing authorization policies on whether the request should be accepted or rejected, for example. However, like the AOR itself, the GRUU is lost in translation at the home proxy. Thus, the UAS cannot know whether it was contacted via the GRUU or its AOR.

2.3. Limited Use Addresses

TOC

A limited use address is an SIP URI that is minted on-demand, and passed out to a small number (usually one) remote correspondent. Incoming calls targeted to that limited use address are accepted as long as the UA still desires communications from the remote target. Should they no longer wish to be bothered by that remote correspondent, the URI is invalidated so that future requests targeted to it are rejected.

Limited use addresses are used in battling voice spam [I-D.ietf-sipping-spam] (Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam," July 2007.). The easiest way to provide them would be for a UA to be able to take its AOR, and "mint" a limited use address by appending additional parameters to the URI. It could then give out the URI to a particular correspondent, and remember that URI locally. When an incoming call arrives, the UAS would examine the parameter in the URI and determine whether or not the call should be accepted. Alternatively, the UA could push authorization rules into the network, so that it need not even see incoming requests that are to be rejected.

This approach, especially when executed on the UA, requires that parameters attached to the AOR, but not used by the home proxy in processing the request, will survive the translation at the home proxy and be presented to the UA. This will not be the case with the logic in RFC 3261, since the Request-URI is replaced by the registered contact, and any such parameters are lost.

2.4. Sub-Addressing

TOC

Sub-Addressing is very similar to limited use addresses. Sub-addresses are addresses within a subdomain that are multiplexed into a single

address within a parent domain. The concept is best illustrated by example. Consider a VoIP service provided to consumers. A consumer obtains a single address from its provider, say sip:family@example.com. However, Joe is the patriarch of a family with four members, and would like to be able to have a separate identifier for each member of his family. One way to do that, without requiring Joe to purchase new addresses for each member from the provider, is for Joe to mint additional URI by adding a parameter to the AOR. For example, his wife Judy with have the URI sip:family@example.com;member=judy, and Joe himself would have the URI sip:family@example.com;member=joe. The SIP server provider would receive requests to these URI, and ignoring the unknown parameters (as required by RFC 3261) route the request to the registered contact, which corresponds to a SIP server in Joes home. That server, in turn, can examine the URI parameters and determine which phone in the home to route the call to.

This feature is not specific to VoIP, and has existing in Integrated Services Digital Networking (ISDN) for some time. It is particularly useful for small enterprises, in addition to families. It is also similar in spirit (though not mechanism) to the ubiquitous home routers used by consumers, which allow multiple computers in the home to "hide" behind the single IP address provided by the service provider, by using the TCP and UDP port as a sub-address.

The sub-addressing feature is not currently feasible in SIP because of the fact that any SIP URI parameter used to convey the sub-address would be lost at the home proxy, due to the fact that the Request-URI is rewritten there.

2.5. Service Invocation

TOC

Several SIP specifications have been developed which make use of complex URIs to address services within the network rather than subscribers. The URIs are complex because they contain numerous parameters that control the behavior of the service. Examples of this include the specification which first introduced the concept, RFC 3087 [RFC3087] (Campbell, B. and R. Sparks, “Control of Service Context using SIP Request-URI,” April 2001.), control of network announcements and IVR with SIP URI [RFC4240] (Burger, E., Van Dyke, J., and A. Spitzer, “Basic Network Media Services with SIP,” December 2005.), and control of voicemail access with SIP URI [RFC4458] (Jennings, C., Audet, F., and J. Elwell, “Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR),” April 2006.).

A common problem with all of these mechanisms is that once a proxy has decided to rewrite the Request-URI to point to the service, it cannot be sure that the Request-URI will not be destroyed by a downstream proxy which decides to forward the request in some way, and does so by rewriting the Request-URI.

2.6. Emergency Services

TOC

Another problem that arises from Request-URI rewriting is with emergency services for VoIP. A key requirement of systems supporting emergency calling is that the SIP INVITE request for an emergency call be 'marked' in some way that makes it clear that it is an emergency call, so that it can receive priority treatment [I-D.ietf-ecrit-requirements] (Schulzrinne, H. and R. Marshall, “Requirements for Emergency Context Resolution with Internet Technologies,” March 2007.). However, such a marking needs to be done in a way that it cannot be abused by attackers seeking to get special treatment for non-emergency calls. The solution for this is that the marking needs to be the target address of the request itself, which would unambiguously identify an emergency services calltaker as the target. The solution that has been agreed upon is the SOS URN [I-D.ietf-ecrit-service-urn] (Schulzrinne, H., “A Uniform Resource Name (URN) for Emergency and Other Well-Known Services,” August 2007.) which takes the form urn:service:sos. This URI appears the in the Request-URI of the request emitted by the UA making the emergency services call, and needs to remain in the Request-URI as the request is routed towards the correct emergency services center (ESC) and eventually the target call taker [I-D.ietf-ecrit-framework] (Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, “Framework for Emergency Calling using Internet Multimedia,” July 2009.).
This mechanism will not work if any of the proxies along the way try to rewrite the Request-URI for the purposes of directing the call to a proxy or UA that will handle the call.

2.7. Freephone Numbers

TOC

Freephone numbers, also known as 800 or 8xx numbers in the United States, are telephone numbers that are free for users to call (although the author will note that such notions are becoming less important as billing models evolve, and harken back to an era where phone service depended on global agreement on such billing concepts). In the telephone network, freephone numbers are just aliases to actual numbers which are used for routing of the call. In order to process the call in the PSTN, a switch will perform a query (using a protocol called TCAP), which will return either a phone number or the identity of a carrier which can handle the call.
There has been recent work on allowing such PSTN translation services to be accessed by SIP proxy servers through IP querying mechanisms. ENUM, for example [RFC3761] (Faltstrom, P. and M. Mealling, “The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM),” April 2004.) has already been

proposed as a mechanism for performing Local Number Portability (LNP) queries [RFC4769] (Livingood, J. and R. Shockey, “IANA Registration for an Enumservice Containing Public Switched Telephone Network (PSTN) Signaling Information,” November 2006.), and recently been proposed for performing calling name queries [I-D.ietf-enum-cnam] (Shockey, R., “IANA Registration for an Enumservice Calling Name Delivery (CNAM) Information and IANA Registration for URI type 'pstndata',” September 2008.). Using it for 8xx number translations is a logical next-step.

Once such a translation has been performed, the call needs to be routed towards the target of the request. Normally, this would happen by selecting a PSTN gateway which is a good route towards the translated number. However, one can imagine all-IP systems where the 8xx numbers are SIP endpoints on an IP network, in which case the translation of the 8xx number would actually be a SIP URI and not a phone number. Assuming for the moment it is a PSTN connected entity, the call would be routed towards a PSTN gateway. Proper treatment of the call in the PSTN (and in particular, correct reconciliation of billing records) requires that the call be marked with both the original 8xx number AND the target number for the call. However, in our example here, since the translation was performed by a SIP proxy upstream from the gateway, the original 8xx number would have been lost, and the call will not interwork properly with the PSTN.

Similar problems arise with other "special" numbers and services used in the PSTN, such as operator services, pay numbers (9xx numbers in the U.S), and short service codes such as 311.

3. Architectural Roots of the Problem

TOC

There is a common theme across all of the problems in Section 2 (Problem Statement), and this theme is the confounding of names, routes, and addresses.

A name is a moniker for an entity which refers to it in a way which reveals nothing about where it is in a network. On the Internet, names are ideally provided through Universal Resource Names (URNs). In the problem cases above, the SOS URN and 8xx numbers are examples of names.

An address is an identifier for an entity which describes it by its location on the network. In SIP, the SIP URI itself is a form of address because the host part of the URI, the only mandatory part of the URI besides the scheme itself, indicates the location of a SIP server that can be used to handle the request. Finally, a route is a sequence of SIP entities (including the UA itself!) which are traversed in order to forward a request to an address or name.

SIP, unfortunately, uses the Request-URI as a mechanism for routing of the request in addition to using it as the mechanism for identifying the name or address to which the request was targeted. A home proxy rewrites the Request-URI because that rewriting is the vehicle by which

the request is forwarded to the target of the request. However, this rewritten URI (the contact from the register), is not in any way a meaningful name or address for the UA. Indeed, with specifications like SIP outbound [I-D.ietf-sip-outbound] (Jennings, C., "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)," June 2009.), even the IP address within the registered contact is meaningless since the flow on which the REGISTER is sent is used rather than the IP address. Consequently, the home proxy is fundamentally replacing the address in the Request-URI with a route to reach that UA. This architectural mistake is the cause of the problems described above.

Interestingly, this same mistake was present in RFC 2543 [RFC2543] (Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol," March 1999.) for the handling of mid-dialog requests. It was fixed through the loose routing mechanism in RFC 3261, which used Route header fields to identify each hop to visit for a mid-dialog request, and separated this from the Request-URI, which identified the ultimate target of the request (the remote UA), and remained unmodified in the processing of the request. It is also interesting to note that in RFC 3261, the Request-URI in a mid-dialog request is the contact provided in the INVITE or 2xx, and identifies the UA itself. This is typically a SIP URI containing an IP address and, as has been argued above, its not an address per se, but a SIP hop. That too has proven to be an error, and has been fixed by the GRUU specification [I-D.ietf-sip-gruu] (Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)," October 2007.), which will cause the Contact in INVITE and 2xx to be the GRUU instead. This, in turn, means that mid-dialog requests will contain the GRUU in the request-URI. The GRUU is, in fact, an address.

However, the loose routing fix made in RFC 3261 was not extended to the handling of requests outside of a dialog. There, proxies retain the practice of rewriting the Request-URI when accessing the location service.

4. Alternative Solutions

TOC

There are several existing mechanisms which might be employed to solve this problem.

4.1. What about the To header field?

TOC

When a UA sends a request, it typically populates the To header field and the Request-URI with the target URI. Consequently, when the request arrives at the terminating network, the Request-URI will be rewritten, but the To header field is retained. Thus, when the request arrives at

the UA, the To header field identifies the original target. Could that serve as the obvious solution to the problem?
Unfortunately, it cannot. When a SIP call is forwarded (also known as retargeting), the actual target of the address changes completely, but the To field does not. When a retargeting operation happens, the URI that needs to be delivered to the UAS is the SIP address or name after the most recent retargeting operation. Consider the case of Alice making a call to Bob (sip:bob@example.com). This arrives at Bob's proxy, which has logic programmed in it to forward the call to Jane, a user in a completely different network (sip:jane@example.edu). When this arrive at Jane's proxy, the Request URI is rewritten to her registered contact. In this case, the To header field contains the original target of the request (sip:bob@example.com), but this is not an identifier for Jane. Thus, the SIP URI for which she was targeted (sip:jane@example.edu). Is lost. Another example of this would be a call to one address or number which is later forwarded to an 8xx number.

4.2. History Info

TOC

Another candidate solution is the History Info specification [RFC4244] (Barnes, M., “An Extension to the Session Initiation Protocol (SIP) for Request History Information,” November 2005.). This specification defines a new header field, History-Info, which records a history of redirection and retargeting operations. One solution to this problem is to require every proxy that rewrites the request URI to implement this specification. As a consequence of that, a UAS could examine the History-Info header field and determine the URI used to reach it. Functionally, this can work. However, we would argue that there are some major architectural problems with it.
Firstly, it would cause the Request-URI to be relegated to nothing more than an indicator of the next hop for the request, identical exactly to the purpose of the Route header field. This results in two things in the SIP specification which do exactly the same thing. Worse still, this is not just for some small feature of SIP (where such duplication might not be a big deal), but rather, it would be a duplication of SIP's primary function - routing of a call towards a target.
Secondly, it would require the UA to look through the history info and figure out which of the URI in there represent the target by which it was reached, and which represent hops that were used along the way. The UA may have no easy way to know this, especially if there were many hops within the domain in which the UA resides.

TOC

5. Proposed Solution

The proposed solution is simple. When handling a request, a proxy only rewrites the Request-URI when performing a retargeting operation. If, instead, the proxy is trying to route the request via some entity (whether its a proxy or UA) to reach the target, the Request-URI is retained, and Route header fields are pushed into the request to reach the target.

This introduces an important question: what is a retargeting operation compared to a routing operation? Is a translation of a name (such as an SOS URN) to an address (like a SIP AOR) a retargeting or a routing operation? We propose that the distinction be determined by means of identity, and in particular the type of assertions provided by [RFC4474] (Peterson, J. and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP),” August 2006.). An operation is considered to be a retargeting operation if the entity to which the request is ultimately delivered could not, based on the policies of the domain of that entity, place the URI prior to translation in the From header field, and have an identity service in its domain sign it. The inverse is not true however. If an entity can legitimately claim the identity prior to the translation operation, it may still be a retargeting. In this case, it is a matter of domain policy about whether it is or not.

From this basic rule, several sub-cases can be derived:

1. When a home proxy receives a request and accesses a location service, the resulting contact(s) obtained from the location service are considered the last hop in the route towards the entity addressed by the Request-URI. Since that target, almost by definition, can claim the identity of the URI prior to translation, the operation is one of routing and not retargeting. Consequently, the home proxy would retain the Request-URI, place the contents of any Path headers from the registration into the request as Route header field values [RFC3327] (Willis, D. and B. Hoeneisen, “Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts,” December 2002.), and insert the registered contact as the last Route header field value.

2. When a proxy receives a request whose contents are a name and not an address (for example, a tel URI or an SOS URN), and the proxy determines through some means an address for that name, this operation is not retargeting. The presumption is that the entity managing the database that provides the translation will only translation the name to an address if the SIP resource identified by that address could claim the name as an identity. Consequently, the proxy would push that address as a Route header field value and retain the Request-URI.

3. When a proxy receiving a request identifies a next hop server that is needed to process the request, that next hop server is a route. A next hop server is not a UA and would never be able to claim its identity. Its URI is pushed into a Route header field and the Request-URI is retained. An important use case for this are proxies that select PSTN gateways for call egress to the PSTN. Such selection would place the SIP URI of the gateway into the topmost Route header field value and retain the Request-URI.

4. When a proxy receives a request whose Request-URI is a SIP URI matching the domain of the proxy, and the proxy decides that the call needs to be terminated at a resource in another domain, this is fundamentally a retargeting operation, and the Request-URI is rewritten. It is fundamentally retargeting because an entity in one domain couldn't claim the identity of an entity in another based on the procedures in [RFC4474] (Peterson, J. and C. Jennings, “Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP),” August 2006.)

This definition also lends clarity to how and when History-Info gets used. In particular, a History-Info header field would get added when a request is retargeted, but not when it is routed. That is, only operations which would cause a Request-URI to be rewritten would cause a History-Info header field to be added.
Redirection can then have several different meanings. Consider a client X which sends a request to server Y. Server Y redirects the request. The redirection could have three meanings:

1. The server is asking the client to retarget, so that the recursed request generated by the client replaces the Request-URI with the contents of the redirection.

2. The server is asking the client to route through a different server instead, so that the recursed request generated by the client replaces the topmost Route header with the contents of the redirection.

3. The server is asking the client to route through an additional proxy prior to visiting it, so that the recursed request generated by the client pushes an additional Route onto the Route set.

Today, a 3xx always has the first semantic. To allow redirects to result in a change in the route header field, an additional mechanism is needed. A client which is capable of supporting this mechanism (whether its a proxy or UA), adds a field to the Via header field which indicates that this hop supports the mechanism.

With that in place, the three different redirect behaviors can be achieved. If a server redirects, and the contact in the redirect contains the ;lr parameter, this is a request for the previous hop to ovveride, for this transaction only, the topmost Route header field value with the value of the contact. If the redirect omits the ;lr parameter, it is a normal redirect that replaces the Request-URI (a retarget). A new response code can be defined, used only when the previous hop supports this specification, for telling the upstream client to append the contact to the existing route set (again for this transaction only).

It is important to note that this mechanism will allow for a mid-dialog request to be redirected to a different hop (i.e., a redirect with an ;lr parameter in the contact), and that this will persist just for the duration of the transaction. This mechanism is used in the failover techniques described in [I-D.rosenberg-sip-outbound-discovery-mid-dialog] (Rosenberg, J., "Discovering Outbound Proxies and Providing High Availability with Client Initiated Connections in the Session Initiation Protocol (SIP)," October 2006.).

6. Backwards Compatibility Considerations

TOC

The principal problem to be resolved is how to make this mechanism backwards compatible. There are several solutions that can be used.

The simplest case is the location service case. When a UA registers, it places the "ua-loose" option tag into the Supported header field of its REGISTER request. If the registrar and home proxy support the UA loose routing procedure described here, it adds a Require header field to the response, indicating to the UA that loose routing procedures will be used. This mechanism would permit different UA for the same AOR to be a mix of ua-loose capable and ua-loose incapable.

There are additional complications with the REGISTER case, however. It is possible that the outbound proxy between the UA and the home proxy will be confused by a new request towards the UA. It will now have a Route header field in it pointing to the UA. Based on the procedures in RFC 3261 and RFC 3263 [RFC3263] (Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers," June 2002.), it should work fine, and even an outbound proxy implementing [I-D.ietf-sip-outbound] (Jennings, C., "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)," June 2009.) will properly route the request towards the UA (that routing being based on the received Route header field, in fact). There is some question about whether a P-CSCF based on the IMS specifications will properly work in this case. Being RFC 3261 compliant it ought to; but it requires further investigation.

The more troubling cases are for translations not based on the registration operation, such as name to address or gateway routing

operations. One idea is to use the existing ;lr URI parameter to indicate that a URI is a loose route, and needs to be placed into a Route header field and not cause replacement of the Request-URI. This would work well when configuring proxies compliant with this specification. A URI with the ;lr parameter indicates a routing next hop, and without indicates a retargeting.
For external services that provide next hops, such as ENUM [RFC3761] (Faltstrom, P. and M. Mealling, “The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM),” April 2004.), implementations would assume that any contents received are not loose routes, but rather retargets. Such services would need to define new fields specifically for loose routes.

7. Minting AORs and GRUU

TOC

With loose routing in place, a UA can mint additional URI that are processed by the SIP proxies identically to their AOR or GRUU. This is done by adding a URI parameter, chosen by the UA, to the AOR or GRUU, and handing that out to UA to use.
Strictly speaking, there is no need to even standardize a specific URI parameter. The parameter is inserted by the UA, and used only by the UA. However, it does need to avoid conflicting with any other URI parameters which might have other meaning by the home proxy, unbeknownst to the UA. This would argue for either one or more IANA registered parameters, use of a vendor namespace, or cryptographically random URI parameter names. It does make sense to allow for more than one URI parameter however. This would allow for infinitely nested sub-addressing capabilities, which is highly desirable.

8. Security Considerations

TOC

The UA loose routing mechanism reveals to the UA the address by which it was contacted. Previously, this was hidden from the UA. It may be possible that a UA is not permitted to know the address at which it was contacted. In such cases, the home proxy SHOULD treat such calls as retargets and rewrite the Request-URI.

9. IANA Considerations

TOC

TODO.

TOC

10. Example

Consider the most basic case of a single proxy P and two user agents, UA1 and UA2. A basic flow for registration and call setup is shown in Figure 1.

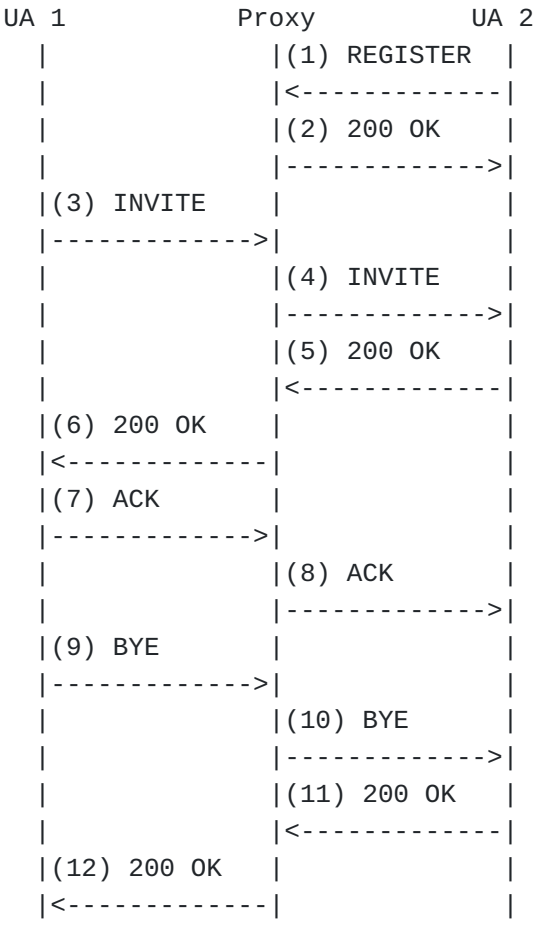

Figure 1

First, UA registers (message 1). It indicates support for loose routing via a Supported header field parameter and also includes an ;lr parameter in its Contact header field. This message would look like, in part (note the usage of both GRUU and sip-outbound; they are not required with UA loose routing but is illustrative of a likely use case):

```
REGISTER sip:example.com SIP/2.0
From: sip:user2@example.com;tag=9asd7d
To:  sip:user2@example.com
Supported: gruu, ua-loose
Contact: <sip:ua2@192.0.2.1;lr>
     ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
     ;reg-id=1
```

The response to the REGISTER (message 2) provides a GRUU to the UA and also indicates that loose routing is to be used:

```
REGISTER sip:example.com SIP/2.0
From: sip:user2@example.com;tag=9asd7d
To:  sip:user2@example.com
Require: ua-loose
Contact: <sip:ua2@192.0.2.1;lr>
     ;+sip.instance="<urn:uuid:0C67446E-F1A1-11D9-94D3-000A95A0E128>"
     ;reg-id=1
     ;expires=3600
     ;pub-gruu="sip:user2@example.com;gr;
      aor-qual=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
```

Next, UA1 generates an INVITE towards UA2 (message 3):

```
INVITE sip:user2@example.com SIP/2.0
From: sip:user1@example.com;tag=555af9g7
To: sip:user2@example.com
```

This arrives at the proxy, which looks up the Request-URI. It finds a single registered contact which is marked as loose routing. Therefore, the request it generates towards UA2 looks like (message 4):

```
INVITE sip:user2@example.com SIP/2.0
Route: <sip:ua2@192.0.2.1;lr>
From: sip:user1@example.com;tag=555af9g7
To: sip:user2@example.com
```

Note that the Request-URI is unmodified and a Route header field has been pushed. The UAS generates a 200 OK (message 5):

```
SIP/2.0 200 OK
From: sip:user1@example.com;tag=555af9g7
To: sip:user2@example.com;tag=6566565
Contact: <sip:user2@example.com;gr;
 aor-qual=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6>
```

Note the presence of the GRUU in the 200 OK. When the BYE comes later on (message 9), it is sent to the GRUU:

```
BYE sip:user2@example.com;gr;
 aor-qual=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
From: sip:user1@example.com;tag=555af9g7
To: sip:user2@example.com;tag=6566565
```

When this arrives at the home proxy, the same thing happens as before. The registered contact bound to the GRUU is a loose route, and so the BYE sent to the UAS would look like (message 10):

```
BYE sip:user2@example.com;gr;
 aor-qual=urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6 SIP/2.0
Route: <sip:ua2@192.0.2.1;lr>
From: sip:user1@example.com;tag=555af9g7
To: sip:user2@example.com;tag=6566565
```

11. References

TOC

11.1. Normative References

TOC

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261, June 2002 (TXT).

[RFC3263] Rosenberg, J. and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” RFC 3263, June 2002 (TXT).

[RFC3327] Willis, D. and B. Hoeneisen, “Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts,” RFC 3327, December 2002 (TXT).

11.2. Informative References

TOC

[I-D.ietf-sip-outbound] Jennings, C., "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)," draft-ietf-sip-outbound-20 (work in progress), June 2009 (TXT).

[I-D.ietf-sip-gruu] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)," draft-ietf-sip-gruu-15 (work in progress), October 2007 (TXT).

[I-D.ietf-sipping-spam] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam," draft-ietf-sipping-spam-05 (work in progress), July 2007 (TXT).

[I-D.ietf-ecrit-requirements] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies," draft-ietf-ecrit-requirements-13 (work in progress), March 2007 (TXT).

[I-D.ietf-ecrit-service-urn] Schulzrinne, H., "A Uniform Resource Name (URN) for Emergency and Other Well-Known Services," draft-ietf-ecrit-service-urn-07 (work in progress), August 2007 (TXT).

[I-D.ietf-ecrit-framework] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling using Internet Multimedia," draft-ietf-ecrit-framework-10 (work in progress), July 2009 (TXT).

[RFC4769] Livingood, J. and R. Shockey, "IANA Registration for an Enumservice Containing Public Switched Telephone Network (PSTN) Signaling Information," RFC 4769, November 2006 (TXT).

[I-D.ietf-enum-cnam] Shockey, R., "IANA Registration for an Enumservice Calling Name Delivery (CNAM) Information and IANA Registration for URI type 'pstndata'," draft-ietf-enum-cnam-08 (work in progress), September 2008 (TXT).

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)," RFC 4474, August 2006 (TXT).

[RFC2543] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol," RFC 2543, March 1999 (TXT).

[RFC3761] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)," RFC 3761, April 2004 (TXT).

[RFC3087] Campbell, B. and R. Sparks, “Control of Service Context using SIP Request-URI,” RFC 3087, April 2001 (TXT).

[RFC4240] Burger, E., Van Dyke, J., and A. Spitzer, “Basic Network Media Services with SIP,” RFC 4240, December 2005 (TXT).

[RFC4458] Jennings, C., Audet, F., and J. Elwell, “Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR),” RFC 4458, April 2006 (TXT).

[RFC4244] Barnes, M., “An Extension to the Session Initiation Protocol (SIP) for Request History Information,” RFC 4244, November 2005 (TXT).

[I-D.rosenberg-sip-outbound-discovery-mid-dialog] Rosenberg, J., “Discovering Outbound Proxies and Providing High Availability with Client Initiated Connections in the Session Initiation Protocol (SIP),” draft-rosenberg-sip-outbound-discovery-mid-dialog-00 (work in progress), October 2006 (TXT).

Author's Address

TOC

Jonathan Rosenberg
Cisco
Edison, NJ
US
Email: jdrosen@cisco.com
URI: http://www.jdrosen.net

Full Copyright Statement

TOC

Intellectual Property

this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.