P2PSIP
Internet-Draft
Intended status: Standards Track
Expires: July 20, 2012

C. Jennings
Cisco
B. Lowekamp, Ed.
Skype
E. Rescorla
RTFM, Inc.
S. Baset
H. Schulzrinne
Columbia University
January 17, 2012

A SIP Usage for RELOAD
draft-ietf-p2psip-sip-07

Abstract

This document defines a SIP Usage for REsource LOcation And Discovery (RELOAD), The SIP Usage provides the functionality of a SIP proxy or registrar in a fully-distributed system. The SIP Usage provides lookup service for AoRs stored in the overlay. The SIP Usage also defines GRUUs that allow the registrations to map an AoR to a specific node reachable through the overlay. The AppAttach method is used to establish a direct connection between nodes through which SIP messages are exchanged.

Table of Contents

1. Overview

The SIP Usage of RELOAD allows SIP user agents to provide a peer-to-peer telephony service without the requirement for permanent proxy or registration servers. In such a network, the RELOAD overlay itself performs the registration and rendezvous functions ordinarily associated with such servers.

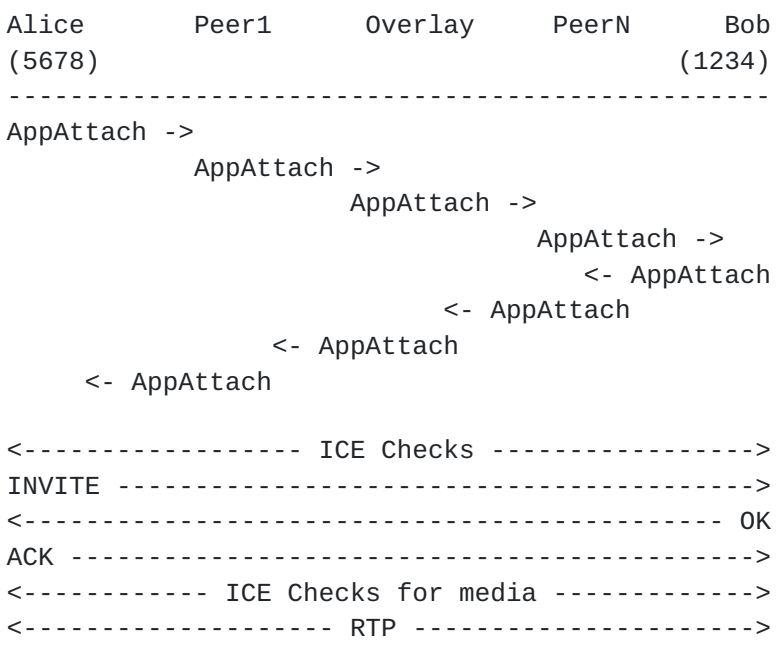
The SIP Usage involves two basic functions:

Registration: SIP UAs can use the RELOAD data storage functionality to store a mapping from their AOR to their Node-ID in the overlay, and to retrieve the Node-ID of other UAs.

Rendezvous: Once a SIP UA has identified the Node-ID for an AOR it wishes to call, it can use the RELOAD message routing system to set up a direct connection which can be used to exchange SIP messages.

For instance, Bob could register his Node-ID, "1234", under his AOR, "bob@dht.example.com". When Alice wants to call Bob, she queries the overlay for "bob@dht.example.com" and gets back Node-ID 1234. She then uses the overlay to establish a direct connection with Bob and can use that direct connection to perform a standard SIP INVITE. The way this works is as follows:

1. Bob, operating Node-ID 1234, stores a mapping from his URI to his Node-ID in the overlay. I.e., "bob@dht.example.com -> 1234".
2. Alice, operating Node-ID 5678, decides to call Bob. She looks up "bob@dht.example.com" in the overlay and retrieves "1234".
3. Alice uses the overlay to route an AppAttach message to Bob's peer. Bob responds with his own AppAttach and they set up a direct connection, as shown below.



It is important to note that RELOAD's only role here is to set up the direct SIP connection between Alice and Bob. As soon as the ICE checks complete and the connection is established, then ordinary SIP is used. In particular, the establishment of the media channel for the phone call happens via the usual SIP mechanisms, and RELOAD is not involved. Media never goes over the overlay. After the successful exchange of SIP messages, call peers run ICE connectivity checks for media.

As well as allowing mappings from AORs to Node-IDs, the SIP Usage also allows mappings from AORs to other AORs. For instance, if Bob wanted his phone calls temporarily forwarded to Charlie, he could store the mapping "bob@dht.example.com -> charlie@dht.example.com". When Alice wants to call Bob, she retrieves this mapping and can then fetch Charlie's AOR to retrieve his Node-ID.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

We use the terminology and definitions from Concepts and Terminology for Peer to Peer SIP [I-D.ietf-p2psip-concepts] and the RELOAD Base Protocol [I-D.ietf-p2psip-base] extensively in this document.

The term AOR is the SIP "Address of Record" used to inditify a user

in SIP. For example, alice@example.com could be the AOR for Alice. For the purposes of this specification, an AOR is consiered not to include the scheme (e.g sip:) as the AOR needs to match the rfc822Name in the X509v3 certificates.

3. Registering AORs

In ordinary SIP, a UA registers its AOR and location with a registrar. In RELOAD, this registrar function is provided by the overlay as a whole. To register its location, a RELOAD peer stores a SipRegistration structure under its own AOR. This uses the SIP-REGISTRATION Kind-ID, which is formally defined in Section 7.
Note: GRUUs are handled via a separate mechanism, as described in Section 6.

As a simple example, if Alice's AOR were "alice@dht.example.com" and her Node-ID were "1234", she might store the mapping "alice@example.org -> 1234". This would tell anyone who wanted to call Alice to contact node "1234".

RELOAD peers MAY store two kinds of SIP mappings:

- From AORs to destination lists (a single Node-ID is just a trivial destination list.)
- From AORs to other AORs.

The meaning of the first kind of mapping is "in order to contact me, form a connection with this peer." The meaning of the second kind of mapping is "in order to contact me, dereference this AOR". This allows for forwarding. For instance, if Alice wants calls to her to be forwarded to her secretary, Sam, she might insert the following mapping "alice@dht.example.org -> sam@dht.example.org".

The contents of a SipRegistration structure are as follows:

```
          enum {sip_registration_uri (1), sip_registration_route (2),
             (255)} SipRegistrationType;

          select (SipRegistration.type) {
            case sip_registration_uri:
              opaque              uri<0..2^16-1>;

            case sip_registration_route:
              opaque              contact_prefs<0..2^16-1>;
              Destination         destination_list<0..2^16-1>;

            /* This type can be extended */

          } SipRegistrationData;

          struct {
             SipRegistrationType   type;
             uint16                length;
             SipRegistrationData   data;
         } SipRegistration;
```

The contents of the SipRegistration PDU are:

type
: the type of the registration

length
: the length of the rest of the PDU

data
: the registration data

- If the registration is of type "sip_registration_uri", then the contents are an opaque string containing the URI.
- If the registration is of type "sip_registration_route", then the contents are an opaque string containing the callee's contact preferences and a destination list for the peer.

RELOAD explicitly supports multiple registrations for a single AOR. The registrations are stored in a Dictionary with the dictionary keys being Node-IDs. Consider, for instance, the case where Alice has two peers:

- her desk phone (1234)
- her cell phone (5678)

Alice might store the following in the overlay at resource "alice@dht.example.com".

- A SipRegistration of type "sip_registration_route" with dictionary key "1234" and value "1234".
- A SipRegistration of type "sip_registration_route" with dictionary key "5678" and value "5678".

Note that this structure explicitly allows one Node-ID to forward to another Node-ID. For instance, Alice could set calls to her desk phone to ring at her cell phone. It's not clear that this is useful in this case, but may be useful if Alice has two AORs.

In order to prevent hijacking, registrations are subject to access control rules. Before a Store is permitted, the storing peer MUST check that:

- The certificate contains a username that is a SIP AOR that hashes to the Resource-ID it is being stored at.
- The certificate contains a Node-ID that is the same as the dictionary key it is being stored at.

Note that these rules permit Alice to forward calls to Bob without his permission. However, they do not permit Alice to forward Bob's calls to her. See Section 8.2.2 for more on this point.

4. Looking up an AOR

When a RELOAD user wishes to call another user, starting with a non-GRUU AOR, he follows the following procedure. (GRUUs are discussed in Section 6).

1. Check to see if the domain part of the AOR matches the domain name of an overlay of which he is a member. If not, then this is an external AOR, and he MUST do one of the following:
   * Fail the call.
   * Use ordinary SIP procedures.
   * Attempt to become a member of the overlay indicated by the domain part, if that overlay is a RELOAD overlay.)
2. Perform a Fetch for kind SIP-REGISTRATION at the Resource-ID corresponding to the AOR. This Fetch SHOULD NOT indicate any dictionary keys, so that it will fetch all the stored values.

3. If any of the results of the Fetch are non-GRUU AORs, then repeat step 1 for that AOR.
4. Once only GRUUs and destination lists remain, the peer removes duplicate destination lists and GRUUs from the list and forms a SIP connection to the appropriate peers as described in the following sections. If there are also external AORs, the peer follows the appropriate procedure for contacting them as well.

Open Issue: Does the RHS of the AORs stored ina given overlay have to match the overlay name?

5. Forming a Direct Connection

Once the peer has translated the AOR into a set of destination lists, it then uses the overlay to route AppAttach messages to each of those peers. The "application" field MUST be 5060 to indicate SIP. If certificate-based authentication is in use, the responding peer MUST present a certificate with a Node-ID matching the terminal entry in the route list. Note that it is possible that the peers already have a RELOAD connection between them. This MUST NOT be used for SIP messages. However, if a SIP connection already exists, that MAY be used. Once the AppAttach succeeds, the peer sends SIP messages over the connection as in normal SIP.

6. GRUUs

GRUUs do not require storing data in the Overlay Instance. Rather, they are constructed by embedding a base64-encoded destination list in the gr URI parameter of the GRUU. The base64 encoding is done with the alphabet specified in table 1 of RFC 4648 with the exception that ~ is used in place of =. An example GRUU is "alice@example.com;gr=MDEyMzQ1Njc4OTAxMjM0NTY3ODk~". When a peer needs to route a message to a GRUU in the same P2P network, it simply uses the destination list and connects to that peer.

Because a GRUU contains a destination list, it MAY have the same contents as a destination list stored elsewhere in the resource dictionary.

Anonymous GRUUs are done in roughly the same way but require either that the enrollment server issue a different Node-ID for each anonymous GRUU required or that a destination list be used that includes a peer that compresses the destination list to stop the Node-ID from being revealed.

7. SIP-REGISTRATION Kind Definition

This section defines the SIP-REGISTRATION kind.

Name SIP-REGISTRATION

Kind IDs The Resource Name for the SIP-REGISTRATION Kind-ID is the AOR of the user. The data stored is a SipRegistration, which can contain either another URI or a destination list to the peer which is acting for the user.

Data Model The data model for the SIP-REGISTRATION Kind-ID is dictionary. The dictionary key is the Node-ID of the storing peer. This allows each peer (presumably corresponding to a single device) to store a single route mapping.

Access Control USER-NODE-MATCH. Note that this matches the SIP AOR against the rfc822Name in the X509v3 certificate. The rfc822Name does not include the scheme so that "sip:" prefix needs to be removed from the SIP AOR before matching.

Data stored under the SIP-REGISTRATION kind is of type SipRegistration. This comes in two varieties:

sip_registration_uri
a URI which the user can be reached at.

sip_registration_route
a destination list which can be used to reach the user's peer.

8. Security Considerations

8.1. Overview

RELOAD provides a generic storage service, albeit one designed to be useful for P2PSIP. In this section we discuss security issues that are likely to be relevant to any usage of RELOAD. In Section 8.2 we describe issues that are specific to SIP.

8.2. SIP-Specific Issues

8.2.1. Fork Explosion

Because SIP includes a forking capability (the ability to retarget to multiple recipients), fork bombs are a potential DoS concern. However, in the SIP usage of RELOAD, fork bombs are a much lower

concern because the calling party is involved in each retargeting event and can therefore directly measure the number of forks and throttle at some reasonable number.

8.2.2. Malicious Retargeting

Another potential DoS attack is for the owner of an attractive number to retarget all calls to some victim. This attack is difficult to ameliorate without requiring the target of a SIP registration to authorize all stores. The overhead of that requirement would be excessive and in addition there are good use cases for retargeting to a peer without there explicit cooperation.

8.2.3. Privacy Issues

All RELOAD SIP registration data is public. Methods of providing location and identity privacy are still being studied.

9. IANA Considerations

IANA [shall register/has registered] code point TBD to represent the SIP-REGISTRATION kind, as described in Section 7.

10. Acknowledgments

This draft is a merge of the "REsource LOcation And Discovery (RELOAD)" draft by David A. Bryan, Marcia Zangrilli and Bruce B. Lowekamp, the "Address Settlement by Peer to Peer" draft by Cullen Jennings, Jonathan Rosenberg, and Eric Rescorla, the "Security Extensions for RELOAD" draft by Bruce B. Lowekamp and James Deverick, the "A Chord-based DHT for Resource Lookup in P2PSIP" by Marcia Zangrilli and David A. Bryan, and the Peer-to-Peer Protocol (P2PP) draft by Salman A. Baset, Henning Schulzrinne, and Marcin Matuszewski.

Thanks to Michael Chen for his contributions.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[I-D.ietf-p2psip-base]

Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-19 (work in progress), October 2011.

11.2. Informative References

[I-D.ietf-p2psip-concepts]
Bryan, D., Matthews, P., Shim, E., Willis, D., and S. Dawkins, "Concepts and Terminology for Peer to Peer SIP", draft-ietf-p2psip-concepts-04 (work in progress), October 2011.

Appendix A. Change Log

A.1. Changes since draft-ietf-p2psip-sip-06

- Added Open Issue

Authors' Addresses

Cullen Jennings
Cisco
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

Phone: +1 408 421-9990
Email: fluffy@cisco.com

Bruce B. Lowekamp (editor)
Skype
Palo Alto, CA
USA

Email: bbl@lowekamp.net

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Phone: +1 650 678 2350
Email: ekr@rtfm.com

Salman A. Baset
Columbia University
1214 Amsterdam Avenue
New York, NY
USA

Email: salman@cs.columbia.edu

Henning Schulzrinne
Columbia University
1214 Amsterdam Avenue
New York, NY
USA

Email: hgs@cs.columbia.edu