P2PSIP Working Group   D. Bryan
Internet-Draft   Cogent Force, LLC
Intended status: Informational   P. Matthews
Expires: August 14, 2016   Alcatel-Lucent
                                                             E. Shim
                                           Samsung Electronics Co., Ltd.
                                                           D. Willis
                                                    Softarmor Systems
                                                         S. Dawkins
                                                        Huawei (USA)
                                                   February 11, 2016

              Concepts and Terminology for Peer to Peer SIP
                      draft-ietf-p2psip-concepts-08

Abstract

   This document defines concepts and terminology for the use of the
   Session Initiation Protocol in a peer-to-peer environment where the
   traditional proxy-registrar and message routing functions are
   replaced by a distributed mechanism. These mechanisms may be
   implemented using a distributed hash table or other distributed data
   mechanism with similar external properties. This document includes a
   high-level view of the functional relationships between the network
   elements defined herein, a conceptual model of operations, and an
   outline of the related problems addressed by the P2PSIP working group
   and the RELOAD protocol and SIP usage document defined by the working
   group.

Copyright Notice

Table of Contents

1. Background

   One of the fundamental problems in multimedia communication between
   Internet nodes is discovering the host at which a given user can be
   reached. In the Session Initiation Protocol (SIP) [RFC3261] this
   problem is expressed as the problem of mapping an Address of Record
   (AoR) for a user into one or more Contact URIs [RFC3986]. The AoR is
   a name for the user that is independent of the host or hosts where
   the user can be contacted, while a Contact URI indicates the host
   where the user can be contacted.

   In the common SIP-using architectures that we refer to as
   "Conventional SIP" or "Client/Server SIP", there is a relatively
   fixed hierarchy of SIP routing proxies and SIP user agents. To
   deliver a SIP INVITE to the host or hosts at which the user can be
   contacted, a SIP UA follows the procedures specified in [RFC3263] to
   determine the IP address of a SIP proxy, and then sends the INVITE to
   that proxy. The proxy will then, in turn, deliver the SIP INVITE to
   the hosts where the user can be contacted.

   This document gives a high-level description of an alternative
   solution to this problem. In this alternative solution, the
   relatively fixed hierarchy of Client/Server SIP is replaced by a
   peer-to-peer overlay network. In this peer-to-peer overlay network,
   the various AoR to Contact URI mappings are not centralized at proxy/
   registrar nodes but are instead distributed amongst the peers in the
   overlay.

   The details of this alternative solution are specified by the RELOAD
   protocol [RFC6940], which defines a mechanism to distribute using a
   Distributed Hash Table (DHT) and specifies the wire protocol,
   security, and authentication mechanisms needed to convey this
   information. This DHT protocol was designed specifically with the
   purpose of enabling a distributed SIP registrar in mind. While
   designing the protocol other applications were considered, and when
   possible design decisions were made that allow RELOAD to be used in
   other instances where a DHT is desirable, but only when making such
   decisions did not add undue complexity to the RELOAD protocol. The
   RELOAD sip draft [I-D.ietf-p2psip-sip] specifies how RELOAD is used
   with the SIP protocol to enable a distributed, server-less SIP
   solution.

2. High-Level Description

   A P2PSIP Overlay is a collection of nodes organized in a peer-to-peer
   fashion for the purpose of enabling real-time communication using the
   Session Initiation Protocol (SIP). Collectively, the nodes in the
   overlay provide a distributed mechanism for mapping names to overlay
   locations. This provides for the mapping of Addresses of Record
   (AoRs) to Contact URIs, thereby providing the "location server"
   function of [RFC3261]. An Overlay also provides a transport function
   by which SIP messages can be transported between any two nodes in the
   overlay.

   A P2PSIP Overlay consists of one or more nodes called Peers. The
   nodes in the overlay collectively run a distributed database
   algorithm. This distributed database algorithm allows data to be
   stored on nodes and retrieved in an efficient manner. It may also
   ensure that a copy of a data item is stored on more than one node, so
   that the loss of a node does not result in the loss of the data item
   to the overlay.

   One use of this distributed database is to store the information
   required to provide the mapping between AoRs and Contact URIs for the
   distributed location function. This provides a location function
   within each overlay that is an alternative to the location functions
   described in [RFC3263]. However, the model of [RFC3263] is used
   between overlays.

2.1. Services

   The nature of peer-to-peer computing is that each peer offers
   services to other peers to allow the overlay to collectively provide
   larger functions. In P2PSIP, peers offer both distributed storage
   and distributed message routing services, allowing these functions to
   be implemented across the overlay. Additionally, the RELOAD protocol
   offers a simplistic discovery mechanism specific to the TURN
   [RFC5766] protocol used for NAT traversal. Individual peers may also
   offer other services as an enhancement to P2PSIP functionality (for
   example to support voicemail) or to support other applications beyond
   SIP. To support these additional services, peers may need to store
   additional information in the overlay. [RFC7374] describes the
   mechanism used in P2PSIP for resource discovery.

2.2. Clients

   An overlay may or may not also include one or more nodes called
   clients. Clients are supported in the RELOAD protocol as peers that
   have not joined the overlay, and therefore do not route messages or
   store information. Clients access the services of the RELOAD

protocol by connecting to a peer which performs operations on the
behalf of the client. Note that in RELOAD there is no distinct
client protocol. Instead, a client connects using the same protocol,
but never joins the overlay as a peer. For more information, see
[RFC6940].

Note that in the context of P2PSIP, there is an additional entity
that is sometimes referred to as a client. A special peer may be a
member of the in the P2PSIP overlay and may present the functionality
of one or all of a SIP registrar, proxy or redirect server to
conventional SIP devices (SIP clients). In this way, existing, non-
modified SIP clients may connect to the network. These unmodified
SIP devices do not speak the RELOAD protocol, and this is a distinct
concept from the notion of client discussed in the previous
paragraph.

2.3. Relationship Between P2PSIP and RELOAD

The RELOAD protocol defined by the P2PSIP working group implements a
DHT primarily for use by server-less, peer-to-peer SIP deployments.
However, the RELOAD protocol could be used for other applications as
well. As such, a "P2PSIP" deployment is generally assumed to be a
use of RELOAD to implement distributed SIP, but it is possible that
RELOAD is used as a mechanism to distribute other applications,
completely unrelated to SIP.

2.4. Relationship Between P2PSIP and SIP

Since P2PSIP is about peer-to-peer networks for real-time
communication, it is expected that most peers and clients will be
coupled with SIP entities (although RELOAD may be used for other
applications than P2PSIP). For example, one peer might be coupled
with a SIP UA, another might be coupled with a SIP proxy, while a
third might be coupled with a SIP-to-PSTN gateway. For such nodes,
the peer or client portion of the node is logically distinct from the
SIP entity portion. However, there is no hard requirement that every
P2PSIP node (peer or client) be coupled to a SIP entity. As an
example, additional peers could be placed in the overlay to provide
additional storage or redundancy for the RELOAD overlay, but might
not have any direct SIP capabilities.

2.5. Relationship Between P2PSIP and Other AoR Dereferencing Approaches

As noted above, the fundamental task of P2PSIP is turning an AoR into
a Contact. This task might be approached using zeroconf techniques
such as multicast DNS and DNS Service Discovery [RFC6762][RFC6763],
link-local multicast name resolution [RFC4795], and dynamic DNS
[RFC2136].

These alternatives were discussed in the P2PSIP Working Group, and
not pursued as a general solution for a number of reasons related to
scalability, the ability to work in a disconnected state, partition
recovery, and so on. However, there does seem to be some continuing
interest in the possibility of using DNS-SD and mDNS for
bootstrapping of P2PSIP overlays.

2.6. NAT Issues

Network Address Translators (NATs) are impediments to establishing
and maintaining peer-to-peer networks, since NATs hinder direct
communication between nodes. Some peer-to-peer network architectures
avoid this problem by insisting that all nodes exist in the same
address space. However, RELOAD provides capabilities that allow
nodes to be located in multiple address spaces interconnected by
NATs, to allow RELOAD messages to traverse NATs, and to assist in
transmitting application-level messages (for example SIP messages)
across NATs.

3. Reference Model

The following diagram shows a P2PSIP Overlay consisting of a number
of Peers, one Client, and an ordinary SIP UA. It illustrates a
typical P2PSIP overlay but does not limit other compositions or
variations; for example, Proxy Peer P might also talk to a ordinary
SIP proxy as well. The figure is not intended to cover all possible
architecture variations, but simply to show a deployment with many
common P2PSIP elements.

```
                                                 --->PSTN
       +------+    N     +------+     +---------+  /
       |      |    A     |      |     | Gateway |-/
       |  UA  |####T#####|  UA  |#####|   Peer  |########
       | Peer |    N     | Peer |     |    G    |      #   RELOAD
       |  E   |    A     |  F   |     +---------+      #   P2PSIP
       |      |    T     |      |                      #   Protocol
       +------+    N     +------+                      #    |
          #         A                                 #    |
       NATNATNATNAT                                    #    |
          #                                            #    |   \__/
       NATNATNATNAT                               +-------+  v   /  \
          #         N                             |       |  |#####/ UA \
       +------+     A        P2PSIP Overlay       | Peer  |    /Client\
       |      |     T                             |   Q   |    |___C__|
       |  UA  |     N                             |       |    |
       | Peer |     A                             +-------+
       |  D   |     T                                 #
       |      |     N                                 #
       +------+     A                                 #  RELOAD
          #         T                                 #  P2PSIP
          #         N    +-------+      +-------+      #  Protocol
          #         A    |       |      |       |      #
          ##########T####| Proxy |########| Redir |#######
                    N    | Peer  |      | Peer  |
                    A    |   P   |      |   R   |
                    T    +-------+      +-------+
                         |                   /
                         | SIP              /
                      \__/  /              /
                       /\  / _____/ SIP
                      /  \/ /
                     / UA \/
                    /_____\
                     SIP UA A
```
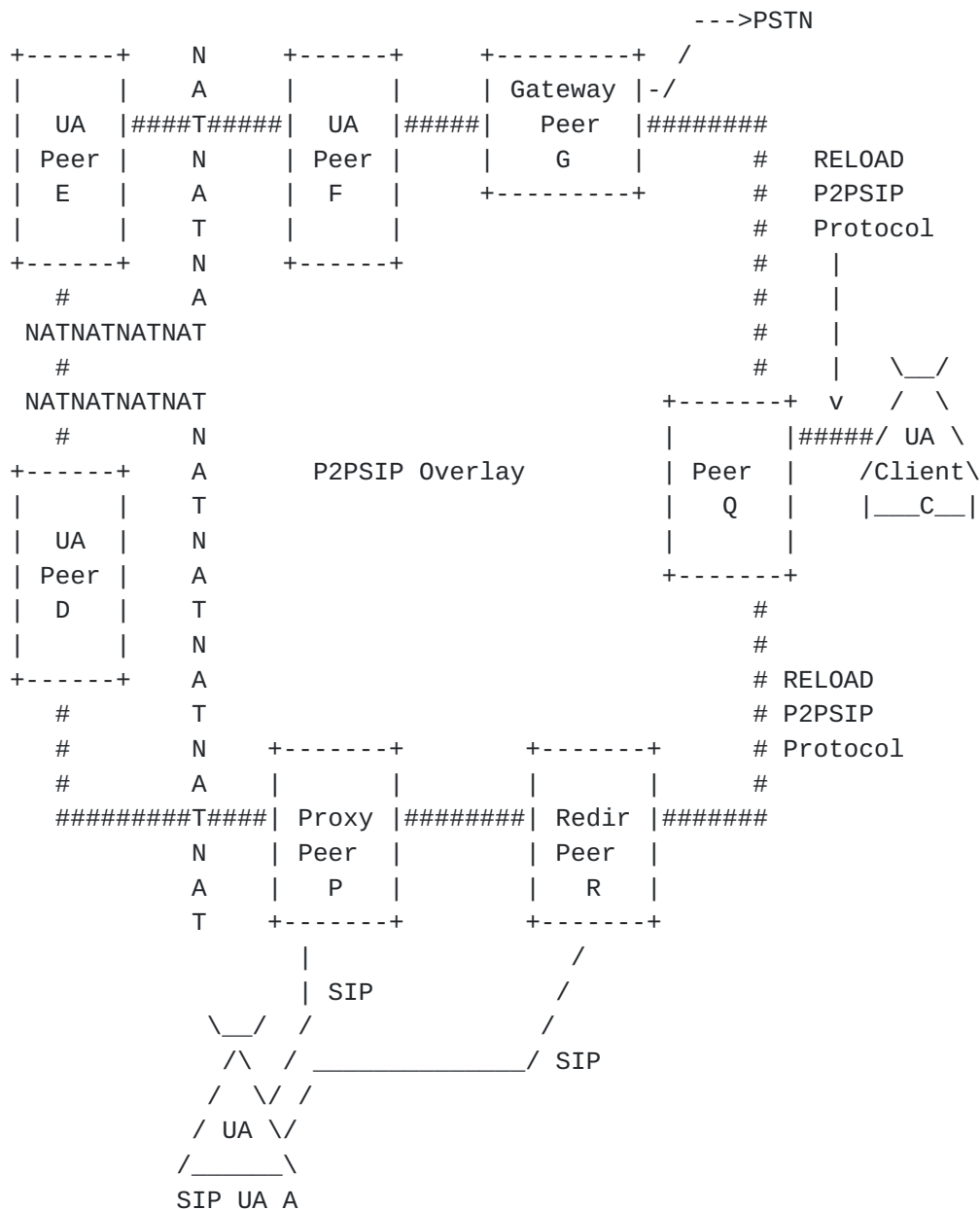
Figure: P2PSIP Overlay Reference Model

   Here, the large perimeter depicted by "#" represents a stylized view
   of the Overlay (the actual connections could be a mesh, a ring, or
   some other structure). Around the periphery of the Overlay
   rectangle, we have a number of Peers. Each peer is labeled with its
   coupled SIP entity -- for example, "Proxy Peer P" means that peer P
   which is coupled with a SIP proxy. In some cases, a peer or client
   might be coupled with two or more SIP entities. In this diagram we
   have a PSTN gateway coupled with peer "G", three peers ("D", "E" and
   "F") which are each coupled with a UA, a peer "P" which is coupled

with a SIP proxy, an ordinary peer "Q" with no SIP capabilities, and one peer "R" which is coupled with a SIP Redirector. Note that because these are all Peers, each is responsible for storing Resource Records and transporting messages around the Overlay.

To the left, two of the peers ("D" and "E") are behind network address translators (NATs). These peers are included in the P2PSIP overlay and thus participate in storing resource records and routing messages, despite being behind the NATs.

On the right side, we have a client "C", which uses the RELOAD Protocol to communicate with Proxy Peer "Q". The Client "C" uses RELOAD to obtain information from the overlay, but has not inserted itself into the overlay, and therefore does not participate in routing messages or storing information.

Below the Overlay, we have a conventional SIP UA "A" which is not part of the Overlay, either directly as a peer or indirectly as a client. It does not speak the RELOAD P2PSIP protocol, and is not participating in the overlay as either a Peer nor Client. Instead, it uses SIP to interact with the Overlay via an adapter peer or peers which communicate with the overlay using RELOAD.

Both the SIP proxy coupled with peer "P" and the SIP redirector coupled with peer "R" can serve as adapters between ordinary SIP devices and the Overlay. Each accepts standard SIP requests and resolves the next-hop by using the P2PSIP protocol to interact with the routing knowledge of the Overlay, then processes the SIP requests as appropriate (proxying or redirecting towards the next-hop). Note that proxy operation is bidirectional - the proxy may be forwarding a request from an ordinary SIP device to the Overlay, or from the P2PSIP overlay to an ordinary SIP device.

The PSTN Gateway at peer "G" provides a similar sort of adaptation to and from the public switched telephone network (PSTN).

[4](#). Definitions

This section defines a number of concepts that are key to understanding the P2PSIP work.

Overlay Network: An overlay network is a computer network which is
   built on top of another network. Nodes in the overlay can be
   thought of as being connected by virtual or logical links, each of
   which corresponds to a path, perhaps through many physical links,
   in the underlying network. For example, many peer-to-peer
   networks are overlay networks because they run on top of the

Internet. Dial-up Internet is an overlay upon the telephone network.

P2P Network: A peer-to-peer (or P2P) computer network is a network that relies primarily on the computing power and bandwidth of the participants in the network rather than concentrating it in a relatively low number of servers. P2P networks are typically used for connecting nodes via largely ad hoc connections. Such networks are useful for many purposes. Sharing content files containing audio, video, data or anything in digital format is very common, and real-time data, such as telephony traffic, is also exchanged using P2P technology. A P2P Network may also be called a "P2P Overlay" or "P2P Overlay Network" or "P2P Network Overlay", since its organization is not at the physical layer, but is instead "on top of" an existing Internet Protocol network.

P2PSIP: A suite of communications protocols related to the Session Initiation Protocol (SIP) [RFC3261] that enable SIP to use peer-to-peer techniques for resolving the targets of SIP requests, providing SIP message transport, and providing other SIP-related functions. At present, these protocols include [RFC6940], [I-D.ietf-p2psip-sip], [I-D.ietf-p2psip-diagnostics], [RFC7374] and [RFC7363].

User: A human that interacts with the overlay through SIP UAs located on peers and clients (and perhaps other ways).

The following terms are defined here only within the scope of P2PSIP. These terms may have conflicting definitions in other bodies of literature. Some earlier versions of this document prefixed each term with "P2PSIP" to clarify the term's scope. This prefixing has been eliminated from the text; however the scoping still applies.

Overlay Name: A human-friendly name that identifies a specific P2PSIP Overlay. This is in the format of (a portion of) a URI, but may or may not have a related record in the DNS.

Peer: A node participating in a P2PSIP Overlay that provides storage and transport services to other nodes in that P2PSIP Overlay. Each Peer has a unique identifier, known as a Peer-ID, within the Overlay. Each Peer may be coupled to one or more SIP entities. Within the Overlay, the peer is capable of performing several different operations, including: joining and leaving the overlay, transporting SIP messages within the overlay, storing information on behalf of the overlay, putting information into the overlay, and getting information from the overlay.

Node-ID: Information that uniquely identifies each Node within a
   given Overlay. This value is not human-friendly -- in a DHT
   approach, this is a numeric value in the hash space. These Node-
   IDs are completely independent of the identifier of any user of a
   user agent associated with a peer.

Client: A node participating in a P2PSIP Overlay but that does not
   store information or forward messages. A client can also be
   thought of as a peer that has not joined the overlay. Clients can
   store and retrieve information from the overlay.

User Name: A human-friendly name for a user. This name must be
   unique within the overlay, but may be unique in a wider scope.
   User Names are formatted so that they can be used within a URI
   (likely a SIP URI), perhaps in combination with the Overlay Name.

Service: A capability contributed by a peer to an overlay or to the
   members of an overlay. Not all peers and clients will offer the
   same set of services, and P2PSIP provides service discovery
   mechanisms to locate services.

Service Name: A unique, human-friendly, name for a service.

Resource: Anything about which information can be stored in the
   overlay. Both Users and Services are examples of Resources.

Resource-ID: A non-human-friendly value that uniquely identifies a
   resource and which is used as a key for storing and retrieving
   data about the resource. One way to generate a Resource-ID is by
   applying a mapping function to some other unique name (e.g., User
   Name or Service Name) for the resource. The Resource-ID is used
   by the distributed database algorithm to determine the peer or
   peers that are responsible for storing the data for the overlay.

Resource Record: A block of data, stored using distributed database
   mechanism of the Overlay, that includes information relevant to a
   specific resource. We presume that there may be multiple types of
   resource records. Some may hold data about Users, and others may
   hold data about Services, and the working group may define other
   types. The types, usages, and formats of the records are a
   question for future study.

Responsible Peer The Peer that is responsible for storing the
   Resource Record for a Resource. In the literature, the term "Root
   Peer" is also used for this concept.

Peer Protocol: The protocol spoken between P2PSIP Overlay peers to
   share information and organize the P2PSIP Overlay Network. In
   P2PSIP, this is implemented using the RELOAD [RFC6940] protocol.

Client Protocol: The protocol spoken between Clients and Peers. In
   P2PSIP and RELOAD, this is the same protocol syntactically as the
   Peer Protocol. The only difference is that Clients are not
   routing messages or routing information, and have not (or can not)
   insert themselves into the overlay.

Peer Protocol Connection / P2PSIP Client Protocol   Connection:
   The TLS, DTLS, TCP, UDP or other transport layer protocol
   connection over which the RELOAD Peer Protocol messages are
   transported.

Neighbors: The set of P2PSIP Peers that a Peer or Client know of
   directly and can reach without further lookups.

Joining Peer: A node that is attempting to become a Peer in a
   particular Overlay.

Bootstrap Peer: A Peer in the Overlay that is the first point of
   contact for a Joining Peer. It selects the peer that will serve
   as the Admitting Peer and helps the joining peer contact the
   admitting peer.

Admitting Peer: A Peer in the Overlay which helps the Joining Peer
   join the Overlay. The choice of the admitting peer may depend on
   the joining peer (e.g., depend on the joining peer's Peer-ID).
   For example, the admitting peer might be chosen as the peer which
   is "closest" in the logical structure of the overlay to the future
   position of the joining peer. The selection of the admitting peer
   is typically done by the bootstrap peer. It is allowable for the
   bootstrap peer to select itself as the admitting peer.

Bootstrap Server: A network node used by Joining Peers to locate a
   Bootstrap Peer. A Bootstrap Server may act as a proxy for
   messages between the Joining Peer and the Bootstrap Peer. The
   Bootstrap Server itself is typically a stable host with a DNS name
   that is somehow communicated (for example, through configuration,
   specification on a web page, or using DHCP) to peers that want to
   join the overlay. A Bootstrap Server is NOT required to be a peer
   or client, though it may be if desired.

Peer Admission: The act of admitting a node (the "Joining Peer")
   into an Overlay as a Peer. After the admission process is over,
   the joining peer is a fully-functional peer of the overlay.
   During the admission process, the joining peer may need to present

credentials to prove that it has sufficient authority to join the
overlay.

Resource Record Insertion: The act of inserting a P2PSIP Resource
Record into the distributed database. Following insertion, the
data will be stored at one or more peers. The data can be
retrieved or updated using the Resource-ID as a key.

5. Discussion

5.1. The Distributed Database Function

A P2PSIP Overlay functions as a distributed database. The database
serves as a way to store information about Resources. A piece of
information, called a Resource Record, can be stored by and retrieved
from the database using a key associated with the Resource Record
called its Resource-ID. Each Resource must have a unique Resource-
ID. In addition to uniquely identifying the Resource, the Resource-
ID is also used by the distributed database algorithm to determine
the peer or peers that store the Resource Record in the overlay.

Users are humans that can use the overlay to do things like making
and receiving calls. Information stored in the resource record
associated with a user can include things like the full name of the
user and the location of the UAs that the user is using (the users
SIP AoR). Full details of how this is implemented using RELOAD are
provided in [I-D.ietf-p2psip-sip]

Before information about a user can be stored in the overlay, a user
needs a User Name. The User Name is a human-friendly identifier that
uniquely identifies the user within the overlay. In RELOAD, users
are issued certificates, which in the case of centrally signed
certificates, identify the User Name as well as a certain number of
Resource-IDs where the user may store their information. For more
information, see [RFC6940].

The P2PSIP suite of protocols also standardizes information about how
to locate services. Services represent actions that a peer (and
perhaps a client) can do to benefit other peers and clients in the
overlay. Information that might be stored in the resource record
associated with a service might include the peers (and perhaps
clients) offering the service. Service discovery for P2PSIP is
defined in [RFC7374].

Each service has a human-friendly Service Name that uniquely
identifies the service. Like User Names, the Service Name is not a
resource-id, rather the resource-id is derived from the service name

   using some function defined by the distributed database algorithm
   used by the overlay.

   A class of algorithms known as Distributed Hash Tables are one way to
   implement the Distributed Database. The RELOAD protocol is
   extensible and allows many different DHTs to be implemented, but
   specifies a mandatory to implement DHT in the form of a modified
   Chord DHT. For more information, see [Chord]

5.2. Using the Distributed Database Function

   While there are a number of ways the distributed database described
   in the previous section can be used to establish multimedia sessions
   using SIP, the basic mechanism defined in the RELOAD protocol and SIP
   usage is summarized below. This is a very simplistic overview. For
   more detailed information, please see the RELOAD protocol document.

   Contact information for a user is stored in the resource record for
   that user. Assume that a user is using a device, here called peer A,
   which serves as the contact point for this user. The user adds
   contact information to this resource record, as authorized by the
   RELOAD certificate mechanism. The resource record itself is stored
   with peer Z in the network, where peer Z is chosen by the particular
   distributed database algorithm in use by the overlay.

   When the SIP entity coupled with peer B has an INVITE message
   addressed to this user, it retrieves the resource record from peer Z.
   It then extracts the contact information for the various peers that
   are a contact point for the user, including peer A, and uses the
   overlay to establish a connection to peer A, including any
   appropriate NAT traversal (the details of which are not shown).

   Note that RELOAD is used only to establish the connection. Once the
   connection is established, messages between the peers are sent using
   ordinary SIP.

   This exchange is illustrated in the following figure. The notation
   "Store(U@A)" is used to show the distributed database operation of
   updating the resource record for user U with the contract A, and
   "Fetch(U)" illustrates the distributed database operation of
   retrieving the resource record for user U. Note that the messages
   between the peers A, B and Z may actually travel via intermediate
   peers (not shown) as part of the distributed lookup process or so as
   to traverse intervening NATs.

```
        Peer B              Peer Z              Peer A
          |                   |                   |
          |                   |          Store(U@Y)|
          |                   |<------------------|
          |                   |Store-Resp(OK)     |
          |                   |------------------>|
          |                   |                   |
          |Fetch(U)           |                   |
          |------------------>|                   |
          |      Fetch-Resp(U@Y)|                 |
          |<------------------|                   |
          |                   |                   |
           (RELOAD IS USED TO ESTABLISH CONNECTION)
          |                   |                   |
          | SIP INVITE(To:U)  |                   |
          |--------------------------------------->|
          |                   |                   |
```
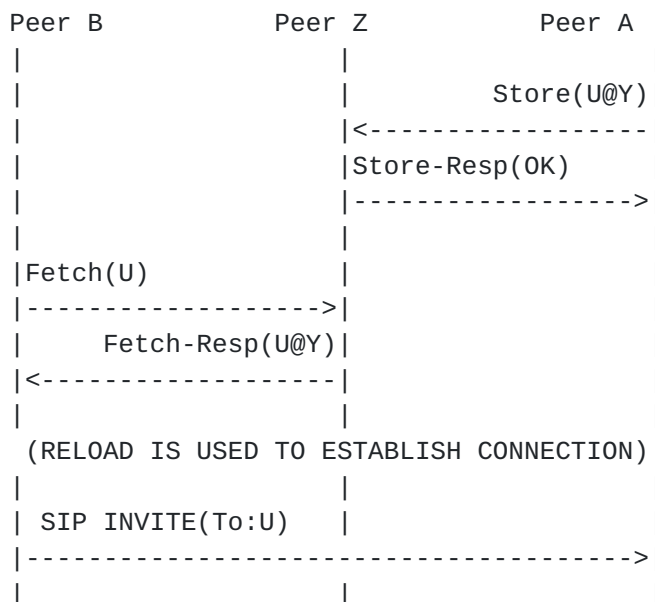
[5.3](). NAT Traversal

NAT Traversal in P2PSIP using RELOAD treats all peers as equal and
establishes a partial mesh of connections between them. Messages
from one peer to another are routed along the edges in the mesh of
connections until they reach their destination. To make the routing
efficient and to avoid the use of standard Internet routing
protocols, the partial mesh is organized in a structured manner. If
the structure is based on any one of a number of common DHT
algorithms, then the maximum number of hops between any two peers is
log N, where N is the number of peers in the overlay. Existing
connections, along with the ICE NAT traversal techniques [RFC5245],
are used to establish new connections between peers, and also to
allow the applications running on peers to establish a connection to
communicate with one another.

[5.4](). Locating and Joining an Overlay

Before a peer can attempt to join a P2PSIP overlay, it must first
obtain a Node-ID, configuration information, and optionally a set of
credentials. The Node-ID is an identifier that will uniquely
identify the peer within the overlay, while the credentials show that
the peer is allowed to join the overlay.

The P2PSIP WG does not impose a particular mechanism for how the
peer-ID and the credentials are obtained, but the RELOAD protocol
does specify the format for the configuration information, and
specifies how this information may be obtained, along with
credentials and a Node-ID, from an offline enrollment server.

Once the configuration information is obtained, RELOAD specifies a
mechanism whereby a peer may obtain a multicast-bootstrap address in
the configuration file, and can broadcast to this address to attempt
to locate a bootstrap peer. Additionally, the peer may store
previous peers it has seen and attempt to use these as bootstrap
peers, or may obtain an address for a bootstrap peer by some other
mechanism. For more information, see the RELOAD protocol.

The job of the bootstrap peer is simple: refer the joining peer to a
peer (called the "admitting peer") that will help the joining peer
join the network. The choice of admitting peer will often depend on
the joining node - for example, the admitting peer may be a peer that
will become a neighbor of the joining peer in the overlay. It is
possible that the bootstrap peer might also serve as the admitting
peer.

The admitting peer will help the joining peer learn about other peers
in the overlay and establish connections to them as appropriate. The
admitting peer and/or the other peers in the overlay will also do
whatever else is required to help the joining peer become a fully-
functional peer. The details of how this is done will depend on the
distributed database algorithm used by the overlay.

At various stages in this process, the joining peer may be asked to
present its credentials to show that it is authorized to join the
overlay. Similarly, the various peers contacted may be asked to
present their credentials so the joining peer can verify that it is
really joining the overlay it wants to.

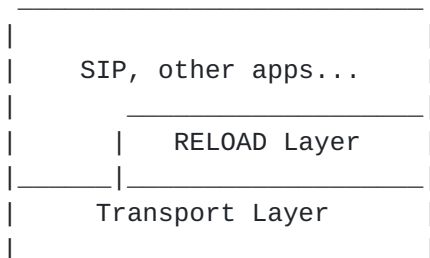## 5.5. Clients and Connecting Unmodified SIP Devices

As mentioned above, in RELOAD, from the perspective of the protocol,
clients are simply peers that do not store information, do not route
messages, and which have not inserted themselves into the overlay.
The same protocol is used for the actual message exchanged. Note
that while the protocol is the same, the client need not implement
all the capabilities of a peer. If, for example, it never routes
messages, it will not need to be capable of processing such messages,
or understanding a DHT.

For SIP devices, another way to realize this functionality is for a
Peer to behave as a [RFC3261] proxy/registrar. SIP devices then use
standard SIP mechanisms to add, update, and remove registrations and
to send SIP messages to peers and other clients. The authors here
refer to these devices simply as a "SIP UA", not a "P2PSIP Client",
to distinguish it from the concept described above.

5.6. Architecture

The architecture adopted by RELOAD to implement P2PSIP is shown
below. An application, for example SIP (or another application using
RELOAD) uses RELOAD to locate other peers and (optionally) to
establish connections to those peers, potentially across NATs.
Messages may still be exchanged directly between the peers. The
overall block diagram for the architecture is as follows:

```
     _____
    |                          |
    |     SIP, other apps...   |
    |      _____  |
    |     |    RELOAD Layer  |  |
    |_____|_____|  |
    |      Transport Layer     |
    |_____|
```

6. Security Considerations

This specification is an overview of existing specifications and does
not introduce any security considerations on its own. Please refer
to the security considerations of the respective specifications,
particularly the RELOAD protocol specification ([RFC6940]) for
further details.

7. IANA Considerations

This document has no actions for IANA.

8. Informative References

[Chord]   Singh, K., Stoica, I., Morris, R., Karger, D., Kaashock,
            M., Dabek, F., and H. Balakrishman, "Chord: A scalable
            peer-to-peer lookup protocol for internet applications",
            IEEE/ACM Transactions on Neworking Volume 11 Issue 1, pp.
            17-32, Feb. 2003, August 2001.

            Copy available at http://pdos.csail.mit.edu/chord/papers/
            paper-ton.pdf

[I-D.ietf-p2psip-diagnostics]
            Song, H., Xingfeng, J., Even, R., Bryan, D., and Y. Sun,
            "P2P Overlay Diagnostics", draft-ietf-p2psip-
            diagnostics-19 (work in progress), November 2015.

[I-D.ietf-p2psip-sip]
            Jennings, C., Lowekamp, B., Rescorla, E., Baset, S.,
            Schulzrinne, H., and T. Schmidt, "A SIP Usage for RELOAD",
            draft-ietf-p2psip-sip-16 (work in progress), December
            2015.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound,
            "Dynamic Updates in the Domain Name System (DNS UPDATE)",
            RFC 2136, DOI 10.17487/RFC2136, April 1997,
            <http://www.rfc-editor.org/info/rfc2136>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
            A., Peterson, J., Sparks, R., Handley, M., and E.
            Schooler, "SIP: Session Initiation Protocol", RFC 3261,
            DOI 10.17487/RFC3261, June 2002,
            <http://www.rfc-editor.org/info/rfc3261>.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation
            Protocol (SIP): Locating SIP Servers", RFC 3263,
            DOI 10.17487/RFC3263, June 2002,
            <http://www.rfc-editor.org/info/rfc3263>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifier (URI): Generic Syntax", STD 66,
            RFC 3986, DOI 10.17487/RFC3986, January 2005,
            <http://www.rfc-editor.org/info/rfc3986>.

[RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local
            Multicast Name Resolution (LLMNR)", RFC 4795,
            DOI 10.17487/RFC4795, January 2007,
            <http://www.rfc-editor.org/info/rfc4795>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment
            (ICE): A Protocol for Network Address Translator (NAT)
            Traversal for Offer/Answer Protocols", RFC 5245,
            DOI 10.17487/RFC5245, April 2010,
            <http://www.rfc-editor.org/info/rfc5245>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using
            Relays around NAT (TURN): Relay Extensions to Session
            Traversal Utilities for NAT (STUN)", RFC 5766,
            DOI 10.17487/RFC5766, April 2010,
            <http://www.rfc-editor.org/info/rfc5766>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762,
            DOI 10.17487/RFC6762, February 2013,
            <http://www.rfc-editor.org/info/rfc6762>.

   [RFC6763]   Cheshire, S. and M. Krochmal, "DNS-Based Service
               Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013,
               <http://www.rfc-editor.org/info/rfc6763>.

   [RFC6940]   Jennings, C., Lowekamp, B., Ed., Rescorla, E., Baset, S.,
               and H. Schulzrinne, "REsource LOcation And Discovery
               (RELOAD) Base Protocol", RFC 6940, DOI 10.17487/RFC6940,
               January 2014, <http://www.rfc-editor.org/info/rfc6940>.

   [RFC7363]   Maenpaa, J. and G. Camarillo, "Self-Tuning Distributed
               Hash Table (DHT) for REsource LOcation And Discovery
               (RELOAD)", RFC 7363, DOI 10.17487/RFC7363, September 2014,
               <http://www.rfc-editor.org/info/rfc7363>.

   [RFC7374]   Maenpaa, J. and G. Camarillo, "Service Discovery Usage for
               REsource LOcation And Discovery (RELOAD)", RFC 7374,
               DOI 10.17487/RFC7374, October 2014,
               <http://www.rfc-editor.org/info/rfc7374>.

Authors' Addresses

   David A. Bryan
   Cogent Force, LLC
   Cedar Park, TX, Texas
   USA

   Email: dbryan@ethernot.org

   Philip Matthews
   Alcatel-Lucent
   600 March Road
   Ottawa, Ontario K2K 2E6
   Canada

   Phone: +1 613 784 3139
   Email: philip_matthews@magma.ca

   Eunsoo Shim
   Samsung Electronics Co., Ltd.
   San 14, Nongseo-dong, Giheung-gu,
   Yongin-si, Gyeonggi-do, 446-712
   South Korea

   Email: eunsooshim@gmail.com

Dean Willis
Softarmor Systems
3100 Independence Pkwy #311-164
Plano, Texas 75075
USA

Phone: +1 214 504 1987
Email: dean.willis@softarmor.com

Spencer Dawkins
Huawei Technologies (USA)

Phone: +1 214 755 3870
Email: spencerdawkins.ietf@gmail.com