TOC

P2PSIP Working Group	D. Bryan
Internet-Draft	SIPeerior Technologies
Intended status: Informational	P. Matthews
Expires: January 8, 2009	Unaffiliated
	E. Shim
	Locus Telecommunications
	D. Willis
	Softarmor Systems
	S. Dawkins
	Huawei (USA)
	July 07, 2008

Concepts and Terminology for Peer to Peer SIP
draft-ietf-p2psip-concepts-02

Status of this Memo

Abstract

This document defines concepts and terminology for 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 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 open problems being addressed by the P2PSIP working group. As this document matures, it is expected to define the general framework for P2PSIP.

Table of Contents

1. Author's Notes and Changes To This Version
    1.1. Author's Notes
    1.2. Changes from Previous Version

2. Background

3. High Level Description
    3.1. Services
    3.2. Clients
    3.3. Protocol
    3.4. Relationship of Peer and Client Protocols
    3.5. Relationship Between P2PSIP and SIP
    3.6. Relationship Between P2PSIP and Other AoR Dereferencing Approaches
    3.7. NAT Issues

4. Reference Model

5. Definitions

6. Discussion
    6.1. The Distributed Database Function
    6.2. Using the Distributed Database Function
    6.3. NAT Traversal
    6.4. Locating and Joining an Overlay
    6.5. Possible Client Behavior
    6.6. Interacting with non-P2PSIP entities
    6.7. Architecture

7. Additional Questions
    7.1. Selecting between Multiple Peers offering the Same Service
    7.2. Visibility of Messages to Intermediate Peers
    7.3. Using C/S SIP and P2PSIP Simultaneously in a Single UA
    7.4. Clients, Peers, and Services
    7.5. Relationships of Domains to Overlays

8. Security Considerations

9. IANA Considerations

10. Acknowledgements

11. References
    11.1. Normative References
    11.2. Informative References

§ Authors' Addresses
§ Intellectual Property and Copyright Statements

1. Author's Notes and Changes To This Version

TOC

1.1. Author's Notes

TOC

The editors are currently considering a rather substantial revision to this document to better reflect the evolving direction of the working group. This version incorporates only minor revisions from the -01 version of the document.
In particular, the authors intend to make the following more substantial changes, and solicit the opinion of the WG on these changes, as well as to solicit suggestions for text for the new sections:

- *Document the current view of the working group that the protocols being developed in P2PSIP should be more broadly applicable than just for peer-to-peer networks of SIP endpoints.

- *The authors plan to add a section that documents the history of various design decisions, and at the same time remove this discussion from other parts of the text. The authors feel that this historical information is important, but also feel that a reader needs to be able to quickly see what the current state of the P2PSIP work is today. An exception would be an early explanation of the fact that P2PSIP doesn't use SIP for the peer protocol, a frequent source of confusion to many people new to the WG.

- *The definition text is somewhat out of date, and should be revised (with some terms added and others eliminated, as appropriate)

- *Incorporate the descriptions of the applications scenarios currently described in draft-bryan-p2psip-app-scenarios-00 into this document.

1.2. Changes from Previous Version

TOC

Changes to this version include removal of the prefix "P2PSIP" before each definition, and clarification on the issue of clients, reflecting the consensus of the WG.

2. Background

TOC

One of the fundamental problems in multimedia communication between Internet nodes is that of discovering the host at which a given user can be reached. 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.) this problem is expressed as the problem of mapping an Address of Record (AoR) for a user into one or more Contact URIs [RFC3986] (Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” January 2005.). 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] (Rosenberg, J. and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” June 2002.) 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 currently being worked out in the P2PSIP working group. This document describes the basic concepts of such a peer-to-peer overlay, and lists the open questions that still need to be resolved. As the work proceeds, it is expected that this document will develop into a high-level architecture document for the solution.

3. High Level Description

TOC

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] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.). 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 peers in the overlay collectively run a distributed database algorithm. This distributed database algorithm allows data to be stored on peers and retrieved in an efficient manner. It may also ensure that a copy of a data item is stored on more than one peer, so that the loss of a peer 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] (Rosenberg, J. and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” June 2002.). However, the model of [RFC3263] (Rosenberg, J. and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” June 2002.) is used between overlays.

3.1. Services

TOC

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 storage and transport services to allow the distributed database function and distributed transport function to be implemented. It is expected that individual peers may also offer other services. Some of these additional services (for example, a STUN server service [I-D.ietf-behave-rfc3489bis] (Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, “Session Traversal Utilities for (NAT) (STUN),” July 2008.)) may be required to allow the overlay to

form and operate, while others (for example, a voicemail service) may be enhancements to the basic P2PSIP functionality.
To allow peers to offer these additional services, the distributed database may need to store information about services. For example, it may need to store information about which peers offer which services, and perhaps what sort of capacity each peer has for delivering each listed service.

3.2. Clients

TOC

An overlay may or may not also include one or more nodes called clients. The role of a client in the P2PSIP model is still under discussion, with a number of suggestions for roles being put forth. The group has reached consensus that clients will be able to store and retrieve information from the overlay. Section 6.5 (Possible Client Behavior) discusses the possible roles of a client in more detail.

3.3. Protocol

TOC

Peers in an overlay need to speak some protocol between themselves to maintain the overlay and to store and retrieve data. Until a better name is found, this protocol has been dubbed the P2PSIP Peer Protocol. While the use of SIP for this protocol was proposed as the working group was forming, the group is currently working toward a new protocol.

3.4. Relationship of Peer and Client Protocols

TOC

To allow clients to communicate with peers, another protocol is required. Until a better name is found, this protocol has been dubbed the P2PSIP Client Protocol. The details of this protocol are also very much under debate. However, if the client protocol exists, then it is agreed that it should be a logical subset of the peer protocol. In other words, the syntax of the peer and client protocols may be completely different, but any operation supported by client protocol is also supported by the peer protocol. This implies that clients cannot do anything that peers cannot also do.

TOC

3.5. Relationship Between P2PSIP and SIP

Since P2PSIP is about peer-to-peer networks for real-time communication, it is expected that most (if not all) peers and clients will be coupled with SIP entities. 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, we think of the peer or client portion of the node as being 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, and some proposed architectures include peer nodes that have no SIP function whatsoever.

3.6. Relationship Between P2PSIP and Other AoR Dereferencing Approaches

TOC

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 (as in Apple's Bonjour protocol), link-local multicast name resolution [RFC4795] (Aboba, B., Thaler, D., and L. Esibov, “Link-local Multicast Name Resolution (LLMNR),” January 2007.), and dynamic DNS [RFC2136] (Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE),” April 1997.).
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.

3.7. NAT Issues

TOC

Network Address Translators (NATs) are impediments to establishing and maintaining peer-to-peer networks, since NATs hinder direct communication between peers. Some peer-to-peer network architectures avoid this problem by insisting that all peers exist in the same address space. However, in the P2PSIP model, it has been agreed that peers can live in multiple address spaces interconnected by NATs. This implies that Peer Protocol connections must be able to traverse NATs. It also means that the peers must collectively provide a distributed transport function that allows a peer to send a SIP message to any other peer in the overlay - without this function two peers in different IP address spaces might not be able to exchange SIP messages.

4. Reference Model

TOC

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 in this document.

```
                                                    --->PSTN
       +------+    N     +------+     +---------+  /
       |      |    A     |      |     | Gateway |-/
       |  UA  |####T#####|  UA  |#####|   Peer  |########
       | Peer |    N     | Peer |     |    G    |       #   P2PSIP
       |  E   |    A     |  F   |     +---------+       #   Client
       |      |    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                                    # P2PSIP
          #        T                                    # Peer
          #        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", 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.

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 speaks neither the Peer nor Client protocols. Instead, it uses SIP to interact with the Overlay.

On the right side, we have a client "C", which uses the Client Protocol depicted by "=" to communicate with Proxy Peer "Q". The Client “C” could communicate with a different peer, for example peer "F", if it establishes a connection to "F" instead of or in addition to "Q". The exact role that this client plays in the network is still under discussion (see Section 6.5 (Possible Client Behavior)).

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 overlay Peer 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).

5. Definitions

TOC

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. http://en.wikipedia.org/wiki/P2P_overlay

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 (see http://en.wikipedia.org/wiki/File_sharing) containing audio, video, data or anything in digital format is very common, and realtime data, such as telephony traffic, is also exchanged using P2P technology. http://en.wikipedia.org/wiki/Peer-to-peer. 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] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) 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. The exact contents of this protocol suite are still under discussion, but is likely to include the P2PSIP Peer Protocol and may include a P2PSIP Client Protocol (see definitions below).

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.

Peer-ID: Information that uniquely identifies each Peer within a given Overlay. This value is not human-friendly -- in a DHT approach, this is a numeric value in the hash space. These Peer-IDs are completely independent of the identifier of any user of a user agent associated with a peer. (Note: This is often called a "Node-ID" in the P2P literature).

Client: A node participating in a P2PSIP Overlay that is less capable than a Peer in some way. The role of a Client is still under debate, with a number of competing proposals (see the discussion on this later in the document). It has been agreed that they do have the ability to add, modify, inspect, and delete information in the overlay. Note that the term client does not imply that this node is a SIP UAC. Some have suggested that the word 'client' be changed to something else to avoid both this confusion and the implication of a client-server relationship.

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. It is expected that not all peers and clients will offer the same set of services, so a means of finding peers (and perhaps clients) that offer a particular service is required. Services might include routing of requests, storing of routing data, storing of other data, STUN discovery, STUN relay, and many other things. This model posits a requirement for a service locator function, possibly including supporting information such as the capacity of a peer to provide a specific service or descriptions of the policies under which a peer will provide that service. We currently expect that we will need to be able to search for available service providers within each overlay. We think we might need to be able to make searches based on network locality or path minimalization.

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.

Client Protocol: The protocol spoken between Clients and Peers. It is used to store and retrieve information from the P2P Overlay. The nature of this protocol, and even its existence, is under discussion. However, if it exists, it has been agreed that the Client Protocol is a functional subset of the P2P Peer Protocol, but may differ in syntax and protocol implementation (i.e., may not be syntactically related).

Peer Protocol Connection / P2PSIP Client Protocol Connection: The TCP, UDP or other transport layer protocol connection over which the Peer Protocol (or respectively the Client protocol) is transported.

Neighbors: The set of P2PSIP Peers that either 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) 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.

6. Discussion

TOC

6.1. The Distributed Database Function

TOC

A P2PSIP Overlay functions as a distributed database. The database serves as a way to store information about things called 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.
It is expected that the P2PSIP working group will standardize the way(s) certain types of resources are represented in the distributed database.

One type of resource representation that the working group is expected to standardize is information about users. 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 might include things like the full name of the user and the location of the UAs that the user is using.

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. The User Name is not a Resource-ID, rather the Resource-ID is derived from the User Name using some mapping function (often a cryptographic hash function) defined by the distributed database algorithm used by the overlay.

The overlay may also require that the user have a set of credentials. Credentials may be required to authenticate the user and/or to show that the user is authorized to use the overlay.

Another type of resource representation that the working group is expected to standardize is information about 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.

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.

It is expected that the working group will standardize at least one service. For each standardized service, the working group will likely specify the service name, the nature and format of the information stored in the resource record associated with the service, and the protocol used to access the service.

The overlay may require that the peer (or client) have a set of credentials for a service. For example, credentials might be required to show that the peer (or client) is authorized to offer the service, or to show that the peer (or client) is a providing a trustworthy implementation of the service.

It is expected that the P2PSIP WG will not standardize how a User Name is obtained, nor how the credentials associated with a User Name or a Service Name are obtained, but merely standardize at least one acceptable format for each. To ensure interoperability, it is expected that at least one of these formats will be specified as "mandatory-to-implement".

A class of algorithms known as Distributed Hash Tables http://en.wikipedia.org/wiki/P2P_overlay are one way to implement the Distributed Database. In particular, both the Chord and Bamboo algorithms have been suggested as good choices for the distributed database algorithm. However, no decision has been taken so far.

6.2. Using the Distributed Database Function

TOC

There are a number of ways the distributed database described in the previous section might be used to establish multimedia sessions using SIP. In this section, we give four possibilities as examples. It seems likely that the working group will standardize at least one way (not necessarily one of the four listed here), but no decisions have been taken yet.

The first option is to store the contact information for a user in the resource record for the user. A peer Y that is a contact point for this user adds contact information to this resource record. The resource record itself is stored with peer Z in the network, where peer Z is chosen by the distributed database algorithm.

When the SIP entity coupled with peer X 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 Y, and forwards the INVITE onward.

This exchange is illustrated in the following figure. The notation "Put(U@Y)" is used to show the distributed database operation of updating the resource record for user U with the contract Y, and "Get(U)" illustrates the distributed database operation of retrieving the resource record for user U. Note that the messages between the peers X, Y 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 X           Peer Z           Peer Y
      |                 |                |
      |                 |       Put(U@Y) |
      |                 |<---------------|
      |                 | Put-Resp(OK)   |
      |                 |--------------->|
      |                 |                |
      | Get(U)          |                |
      |---------------->|                |
      |    Get-Resp(U@Y)|                |
      |<----------------|                |
      | INVITE(To:U)    |                |
      |--------------------------------->|
      |                 |                |
```

The second option also involves storing the contact information for a user in the resource record of the user. However, SIP entity at peer X, rather than retrieving the resource record from peer Z, instead forwards the INVITE message to the proxy at peer Z. The proxy at peer Z then uses the information in the resource record and forwards the INVITE onwards to the SIP entity at peer Y and the other contacts.

```
     Peer X           Peer Z           Peer Y
      |                 |                |
      |                 |       Put(U@Y) |
      |                 |<---------------|
      |                 | Put-Resp(OK)   |
      |                 |--------------->|
      |                 |                |
      | INVITE(To:U)    |                |
      |-----------------| INVITE(To:U)   |
      |                 |--------------->|
      |                 |                |
```

The third option is for a single peer W to place its contact information into the resource record for the user (stored with peer Z). A peer Y that is a contact point for the user retrieves the resource record from peer Z, extracts the contact information for peer W, and then uses the standard SIP registration mechanism [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) to register with peer W. When the SIP entity at peer X has to forward an INVITE request, it retrieves the resource record and extracts the contact information for W. It then forwards the INVITE to the proxy at peer W, which proxies it onward to peer Y and the other contacts.

```
     Peer X           Peer Z           Peer Y           Peer W
      |                 |                |                 |
      |                 |       Put(U@W) |                 |
      |                 |<---------------------------------|
      |                 | Put-Resp(OK)   |                 |
      |                 |--------------------------------->|
      |                 |                |                 |
      |                 |                |                 |
      |                 |                | REGISTER(To:U)  |
      |                 |                |---------------->|
      |                 |                |             200 |
      |                 |                |<----------------|
      |                 |                |                 |
      |                 |                |                 |
      | Get(U)          |                |                 |
      |---------------->|                |                 |
      |    Get-Resp(U@W)|                |                 |
      |<----------------|                |                 |
      | INVITE(To:U)    |                |                 |
      |--------------------------------------------------->|
      |                 |                |    INVITE(To:U) |
      |                 |                |<----------------|
      |                 |                |                 |
```

The fourth option works as in option 3, with the exception that, rather than X retrieving the resource record from Z, peer X forwards the INVITE to a SIP proxy at Z, which proxies it onward to W and hence to Y.

```
     Peer X           Peer Z           Peer Y           Peer W
      |                 |                |                 |
      |                 |       Put(U@W) |                 |
      |                 |<---------------------------------|
      |                 | Put-Resp(OK)   |                 |
      |                 |--------------------------------->|
      |                 |                |                 |
      |                 |                |                 |
      |                 |                | REGISTER(To:U)  |
      |                 |                |---------------->|
      |                 |                |             200 |
      |                 |                |<----------------|
      |                 |                |                 |
      |                 |                |                 |
      | INVITE(To:U)    |                |                 |
      |---------------->| INVITE(To:U)   |                 |
      |                 |--------------------------------->|
      |                 |                |    INVITE(To:U) |
      |                 |                |<----------------|
      |                 |                |                 |
```

The pros and cons of option 1 and 3 are briefly discussed in [Using-an-External-DHT] (Singh, K. and H. Schulzrinne, “Using an External DHT as a SIP Location Service,” .).

6.3. NAT Traversal

TOC

Two approaches to NAT Traversal for P2PSIP Peer Protocol have been suggested. The working group has not made any decision yet on the approach that will be selected.
The first, the traditional approach adopted by most peer-to-peer networks today, divides up the peers in the network into two groups: those with public IP addresses and those without. The networks then select a subset of the former group and elevate them to "super peer" status, leaving the remaining peers as "ordinary peers". Since super peers all have public IP addresses, there are no NAT problems when communicating between them. The network then associates each ordinary peer with (usually just one) super peer in a client-server relationship. Once this is done, an ordinary peer X can communicate with another ordinary peer Y by sending the message to X's super peer, which forwards it to Y's super peer, which forwards it to Y. The connection between an ordinary peer and its super peer is initiated by the ordinary peer, which makes it easy to traverse any intervening

NATs. In this approach, the number of hops between two peers is at most 3.

The second approach treats all peers as equal and establishes a partial mesh of connections between them. Messages from one peer to another are then 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.

The first approach is significantly more efficient than the second in overlays with large numbers of peers. However, the first approach assumes there are a sufficient number of peers with public IP addresses to serve as super peers. In some usage scenarios envisioned for P2PSIP, this assumption does not hold. For example, this approach fails completely in the case where every peer is behind a distinct NAT.

The second approach, while less efficient in overlays with larger numbers of peers, is efficient in smaller overlays and can be made to work in many use cases where the first approach fails.

Both of these approaches assume a method of setting up Peer Protocol connections between peers. Many such methods exist; the now expired [I-D.iab-nat-traversal-considerations] (Rosenberg, J., “Considerations for Selection of Techniques for NAT Traversal,” October 2005.) is an attempt to give a fairly comprehensive list along with a discussion of their pros and cons. After a consideration of the various techniques, the P2PSIP working group has decided to select the Unilateral Self-Address Fixing method [RFC3424] (Daigle, L. and IAB, “IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation,” November 2002.) of NAT Traversal, and in particular the ICE [I-D.ietf-mmusic-ice] (Rosenberg, J., “Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols,” October 2007.) implementation of this approach.

The above discussion covers NAT traversal for Peer Protocol connections. For Client Protocol connections, the approach depends on the role adopted for clients and we defer the discussion on that point until the role becomes clearer.

In addition to Peer Protocol and Client Protocol messages, a P2PSIP Overlay must also provide a solution to the NAT Traversal problem for SIP messages. If it does not, there is no reliable way for a peer behind one NAT to send a SIP INVITE to a peer behind another NAT. One way to solve this problem is to transport SIP messages along Peer and Client Protocol connections: this could be done either by encapsulating the SIP messages inside Peer and Client Protocol messages or by multiplexing SIP with the Peer (resp.Client) Protocol on a Peer (resp. Client) Protocol connection.

Finally, it should be noted that the NAT traversal problem for media connections signaled using SIP is outside the scope of the P2PSIP working group. As discussed in [I-D.ietf-sipping-nat-scenarios] (Boulton, C., Rosenberg, J., Camarillo, G., and F. Audet, “Best Current Practices for NAT Traversal for Client-Server SIP,” February 2010.), the current recommendation is to use ICE.

6.4. Locating and Joining an Overlay

TOC

Before a peer can attempt to join a P2PSIP overlay, it must first obtain a Peer-ID and optionally a set of credentials. The Peer-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 will not standardize how the peer-ID and the credentials are obtained, but merely standardize at least one acceptable format for each. To ensure interoperability, it is expected that at least one of these formats will be specified as "mandatory-to-implement".

Once a peer (the "joining peer") has a peer-ID and optionally a set of credentials, it can attempt to join the overlay. To do this, it needs to locate a bootstrap peer for the Overlay.

A bootstrap peer is a peer that serves as the first point of contact for the joining peer. The joining peer uses a bootstrap mechanism to locate a bootstrap peer. Locating a bootstrap peer might be done in any one of a number of different ways:

- By remembering peers that were part of the overlay the last time the peer was part of the overlay;
- Through a multicast discovery mechanism;
- Through manual configuration; or
- By contacting a P2PSIP Bootstrap Server, and using its help to locate a bootstrap peer.

The joining peer might reasonably try each of the methods (and perhaps others) in some order or in parallel until it succeeds in finding a bootstrap peer.

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 in 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.

6.5. Possible Client Behavior

TOC

As mentioned above, a number of people have proposed a second type of P2PSIP entity, known as a "P2PSIP client". The consensus of the group is that the need for entities to store and retrieve information from the Overlay without participating is recognized, but that for now, little time will spent. This section presents some of the alternatives that have been suggested for the possible role of a client.

In one approach, a client interacts with the P2PSIP overlay through an associated peer (or perhaps several such peers) using the Client Protocol. The client does not run the distributed database algorithm, does not store resource records, and is not involved in routing messages to other peers or clients. Through interactions with its associated peer, a client can insert, modify, examine, and remove resource records. A client may also send SIP messages to its associated peer for routing through the overlay. In this approach, a client is a node that wants to take advantage of the overlay, but is unable or unwilling to contribute resources back to the overlay. This may be achieved using a subset of the Peer Protocol. Such a device need not speak SIP.

For SIP devices, another way to realize this functionality is for a Peer to behave as a [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.) 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.

6.6. Interacting with non-P2PSIP entities

TOC

It is possible for network nodes that are not peers or clients to interact with a P2PSIP overlay. Such nodes would do this through mechanisms not defined by the P2PSIP working group provided they can find a peer or client that supports that mechanism and which will do any related P2PSIP operations necessary. In this section, we briefly describe two ways this might be done. (Note that these are just examples and the descriptions here are not recommendations).

One example is a peer that also acts as a standard SIP proxy and registrar. SIP UAs can interact with it using mechanisms defined in [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, “SIP: Session Initiation Protocol,” June 2002.). The peer inserts registrations for users learned from these UAs into the distributed database, and retrieves contact information when proxying INVITE messages.

Another example is a peer that has a fully-qualified domain name (FQDN) that matches the name of the overlay and acts as a SIP proxy for calls coming into the overlay. A SIP INVITE addressed to "user@overlay-name" arrives at the peer (using the mechanisms in [RFC3263] (Rosenberg, J. and H. Schulzrinne, “Session Initiation Protocol (SIP): Locating SIP Servers,” June 2002.)) and this peer then looks up the user in the distributed database and proxies the call onto it.

6.7. Architecture

TOC

There has been much debate in the group over what an appropriate architecture for P2PSIP should be. Currently, the group is investigating architectures that involve a P2P layer that is distinct from the applications that run on the overlay.

```
           __________________________
          |                          |
          |    SIP, other apps...    |
          |       ___________________|
          |      |   P2P Layer       |
          |______|___________________|
          |     Transport Layer      |
          |__________________________|
```

The P2P layer implements the Peer Protocol (and the Client Protocol, if such a protocol exists). Applications access this P2P layer for various overlay-related services. Applications are also free to bypass this layer and access the existing transport layer protocols (e.g., TCP, UDP, etc.) directly.

A notable feature of this architecture is that it envisions the use of protocols other than SIP in the overlay. Though the working group is primarily focused on the use of SIP in peer-to-peer overlays, this architecture envisions a future in which other protocols can play a role.

The group initially considered another architecture. In this alternative architecture, the Peer Protocol was defined as an extension

to SIP. That is, that the necessary operations for forming and maintaining the overlay and for storing and retrieving resource records in the distributed database were defined as extensions to SIP. Each peer in the overlay was viewed as a SIP proxy that would forward the overlay maintenance and distributed database query messages (expressed in SIP) on behalf of other peers.
This architecture was eventually rejected by the working group for the following reasons:

- The architecture was totally focused on SIP, and made it difficult to use other protocols in the overlay.

- In SIP, proxies are assumed to be trusted parties. Relying on the peers to route the message as proxies exposes the SIP messages to attacks from untrusted proxies that SIP's design does not anticipate. A design that does not allow the peers to modify the SIP message and ideally prevents them from reading it is preferable.

- SIP was seen as a "heavy-weight" protocol for this task. SIP uses a text-based encoding which is very flexible, but leads to both large messages and slow processing times at proxies. This was seen to be a poor match for P2PSIP, where a distributed database lookup operation requires O(log N) peers to receive, process and forward the message.

More discussion on this alternate approach and why it was rejected can be found on the P2PSIP mailing list in a thread that started on 20 March 2007.

7. Additional Questions

TOC

This section lists some additional questions that the proposed P2PSIP Working Group may need to consider in the process of defining the Peer and Client protocols.

TOC

7.1. Selecting between Multiple Peers offering the Same Service

If a P2PSIP network contains two or more peers that offer the same service, then how does a peer or client that wishes to use that service select the peer to use? This question comes up in a number of contexts:

- When two or more peers are willing to serve as a STUN Relay, how do we select a peer that is close in the netpath sense and is otherwise appropriate for the call?

- When two or more peers are willing to serve as PSTN gateways, how do we select an appropriate gateway for a call that is both netpath efficient and provides good quality or inexpensive PSTN routing?

It has been suggested that, at least initially, the working group should restrict itself to defining a mechanism that can return a list of peers offering a service and not define the mechanism for selecting a peer from that list.

TOC

7.2. Visibility of Messages to Intermediate Peers

When transporting SIP messages through the overlay, are the headers and/or bodies of the SIP messages visible to the peers that the messages happen to pass through? If they are, what types of security risks does this pose in the presence of peers that have been compromised in some way?

TOC

7.3. Using C/S SIP and P2PSIP Simultaneously in a Single UA

If a given UA is capable of operating in both P2PSIP and conventional SIP modalities (especially simultaneously), is it possible for it to use and respond to the same AOR using both conventional and P2PSIP? An example of such a topology might be a UA that registers an AOR (say, "sip:alice@example.com") conventionally with a registrar and then inserts a resource record for that resource into a P2PSIP topology, such that both conventional SIP users and P2PSIP users (within the overlay or a federation thereof) would be able to contact the user without necessarily traversing some sort of gateway. Is this something that we want to make work?

TOC

7.4. Clients, Peers, and Services

1. Do all peers providing routing, storage, and all other services, or do only some peers provide certain services?

2. What services, if any, must all peers provide?

3. How we can we describe the capacity of a peer for delivering a given service?

7.5. Relationships of Domains to Overlays

TOC

1. Can there be names from more than one domain in a single overlay?

2. Can there be names from one domain in more than a single overlay? If so, how do we route Client/Server SIP requests to the right overlay?

3. Can the domain of an AoR be in more than one overlay?

4. Should we have a "default overlay" to search for peers in many domains?

8. Security Considerations

TOC

Building a P2PSIP system has many security considerations, many of which we have only begun to consider. We anticipate that the protocol documents describing the actual protocols will deal more thoroughly with security topics.

One critical security issue that will need to be addressed is providing for the privacy and integrity of SIP messages being routed by peer nodes, when those peer nodes might well be hostile. This is a departure from Client/Server SIP, where the proxies are generally operated by enterprises or service providers with whom the users of SIP UAs have a trust relationship.

9. IANA Considerations

TOC

This document presently raises no IANA considerations.

10. Acknowledgements

TOC

This document draws heavily from the contributions of many participants in the P2PSIP Mailing List. Particular thanks to Henning Schulzrinne and Cullen Jennings who spent time on phone calls related to this text.

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).
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” STD 66, RFC 3986, January 2005 (TXT, HTML, XML).

11.2. Informative References

TOC

[I-D.bryan-p2psip-reload]	Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD)," draft-bryan-p2psip-reload-04 (work in progress), June 2008 (TXT).
[I-D.camarillo-hip-bone]	Camarillo, G., Nikander, P., and J. Hautakorpi, "HIP BONE: Host Identity Protocol (HIP) Based Overlay Networking Environment," draft-camarillo-hip-bone-01 (work in progress), February 2008 (TXT).
[I-D.iab-nat-traversal-considerations]	Rosenberg, J., "Considerations for Selection of Techniques for NAT Traversal," draft-iab-nat-traversal-considerations-00 (work in progress), October 2005 (TXT).
[I-D.ietf-behave-rfc3489bis]	Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for (NAT) (STUN)," draft-ietf-behave-rfc3489bis-18 (work in progress), July 2008 (TXT).
[I-D.ietf-mmusic-ice]	Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," draft-ietf-mmusic-ice-19 (work in progress), October 2007 (TXT).
[I-D.ietf-sipping-nat-scenarios]	Boulton, C., Rosenberg, J., Camarillo, G., and F. Audet, "Best Current Practices for NAT Traversal for Client-Server SIP," draft-ietf-sipping-nat-scenarios-10 (work in progress), February 2010 (TXT).
[I-D.jiang-p2psip-sep]	Jiang, X. and H. Zhang, "Service Extensible P2P Peer Protocol," draft-jiang-p2psip-sep-01 (work in progress), February 2008 (TXT).
[I-D.li-p2psip-node-types]	Wang, Y., "Different types of nodes in P2PSIP," draft-li-p2psip-node-types-00 (work in progress), December 2007 (TXT).
[I-D.matthews-p2psip-id-loc]	Cooper, E., Johnston, A., and P. Matthews, "An ID/Locator Architecture for P2PSIP," draft-matthews-p2psip-id-loc-01 (work in progress), February 2008 (TXT).
[I-D.pascual-p2psip-clients]	Pascual, V., Matuszewski, M., Shim, E., Zhang, H., and S. Yongchao, "P2PSIP Clients," draft-pascual-p2psip-clients-01 (work in progress), February 2008 (TXT).
[I-D.zheng-p2psip-client-protocol]	Yongchao, S., Jiang, X., Zhang, H., and H. Deng, "P2PSIP Client Protocol," draft-zheng-p2psip-client-protocol-01 (work in progress), February 2008 (TXT).

[RFC2136]	Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE),” RFC 2136, April 1997 (TXT, HTML, XML).
[RFC3424]	Daigle, L. and IAB, “IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation,” RFC 3424, November 2002 (TXT).
[RFC4485]	Rosenberg, J. and H. Schulzrinne, “Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP),” RFC 4485, May 2006 (TXT).
[RFC4795]	Aboba, B., Thaler, D., and L. Esibov, “Link-local Multicast Name Resolution (LLMNR),” RFC 4795, January 2007 (TXT).
[Using-an-External-DHT]	Singh, K. and H. Schulzrinne, “Using an External DHT as a SIP Location Service,” Columbia University Computer Science Dept. Tech Report 388). Copy available at http://mice.cs.columbia.edu/getTechreport.php?techreportID=388/

Authors' Addresses

TOC

	David A. Bryan
	SIPeerior Technologies
	3000 Easter Circle
	Williamsburg, Virginia 23188
	USA
Phone:	+1 757 565 0101
Email:	bryan@sipeerior.com
	Philip Matthews
	Unaffiliated
Phone:	+1 613 592 4343 x224
Email:	philip_matthews@magma.ca
	Eunsoo Shim
	Locus Telecommunications
	111 Sylvan Avenue
	Englewood Cliffs, New Jersey 07632
	USA
Phone:	unlisted
Email:	eunsooshim@gmail.com
	Dean Willis

Softarmor Systems
3100 Independence Pkwy #311-164
Plano, Texas 75075
USA
Phone: unlisted
Email: dean.willis@softarmor.com

Spencer Dawkins
Huawei Technologies (USA)
Phone: +1 214 755 3870
Email: spencer@wonderhamster.org

Full Copyright Statement

TOC

Intellectual Property