P2PSIP Working Group
Internet-Draft
Intended status: Informational
Expires: September 5, 2007

D. Bryan
SIPeerior Technologies and
College of William and Mary
P. Matthews
Avaya
E. Shim
Locus Telecommunications
D. Willis
Cisco Systems

March 4, 2007

Concepts and Terminology for Peer to Peer SIP
draft-willis-p2psip-concepts-04

Status of this Memo

Copyright Notice

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 function is replaced by a distributed mechanism that might 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 open problems that might be addressed by an IETF working group.

Table of Contents

1. Background

One of the fundamental problems in multimedia communications between Internet nodes is that of a discovering the IP address at which a given correspondent can be reached. Correspondents are frequently identified by distinguished names, perhaps represented in the form of a universal resource indicator (URI) [RFC3986].

The Session Initiation Protocol (SIP) [RFC3261] commonly addresses this task assuming that the domain part of the URI indicates an internet host address or internet domain name using the Domain Name System (DNS) [RFC1034]. SIP's location process [RFC3263] assumes that the host part of the URI indicates either the target SIP user agent (UA), or a proxy that has knowledge of how to reach the target UA, presumably gained through SIP's registration process.

This approach, referred to herein as "Conventional SIP" or "Client/Server SIP", assumes a relatively fixed hierarchy of SIP routing proxies (servers) and SIP user agents (clients). The routing proxies are typically resolvable using conventional Internet mechanisms with static IP addresses and associated DNS entries. This structure may not be ideal in all cases, including specifically ad-hoc, serverless, and reduced-administration scenarios. As an alternative, several authors [I-D.bryan-p2psip-dsip] [I-D.shim-sipping-p2p-arch] [I-D.johnston-sipping-p2p-ipcom] [I-D.matthews-sipping-p2p-industrial-strength] have proposed using peer-to-peer (P2P) [I-D.irtf-p2prg-survey-search] approaches to solving the problem of locating the correspondent. The motivations for a P2P approach in SIP have been documented in [I-D.bryan-sipping-p2p-usecases].

This document offers a consolidation of the more important terms and concepts from several of the above documents, presented in the context of a reference model for peer-to-peer SIP (P2PSIP). It is intended that this document serve as a starting point for describing the work needed to resolve a number of open questions such that an IETF working group can do the work needed to resolve these questions and present a standard solution. The authors believe that this goal is roughly consistent with that of a Protocol Model as defined in [RFC4101].

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 implementation of the location service

[RFC3261] for mapping Addresses of Record (AoRs) to Contact URIs. They also provide a transport service by which SIP messages can be transported between any two nodes in the overlay.

A P2PSIP Overlay consists of one or more nodes called P2PSIP 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 service.

The nature of peer-to-peer computing is that each peer offers services to other peers to allow the overlay to collectively provide larger services. In P2PSIP, each individual peer needs to offer a storage service and a transport service to allow the distributed database service and distributed transport service 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]) 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.

An overlay may or may not also include one or more nodes called P2PSIP Clients. The role of a client in the P2PSIP model is still under discussion, with a number of suggestions for roles being put forth, and some arguing that clients are not needed at all. However, if they exist, then people agree that they will also be able to store and retrieve information from the overlay. Section 5.5 discusses the possible roles of a client in more detail.

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. The details of this protocol are still very much under debate: some have suggested that the protocol should be SIP with some extensions, while others have suggested that it should be an entirely new protocol.

If the P2PSIP model also contains clients, then a protocol is needed for client - peer communication. 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.

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.

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 transport service that allows a peer to send a SIP message to any other peer in the overlay - without this service two peers in different address spaces might not be able to exchange SIP messages.

3. Reference Model

The following diagram shows a P2PSIP Overlay consisting of a number of P2PSIP Peers and one P2PSIP Client. It also shows an ordinary SIP UA.

```
                                                  --->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         | Proxy |    /Client\
     |      |    T                              | Peer  |    |___C__|
     |  UA  |    N                              |   Q   |
     | Peer |    A                              +-------+
     |  D   |    T     P2PSIP Peer Protocol           #
     |      |    N                                    #
     +------+    A                                    #
        #        T                                    #
        #        N    +-------+        +-------+      #
        #        A    |       |        |       |      #
        #########T####| Proxy |########| Redir |#######
                 N    | Peer  |        | Peer  |
                 A    |   P   |        |   R   |
                 T    +-------+        +-------+


                  \__/
                   /\
                  /  \
                 / UA \
                /______\
                SIP UA A
```

Figure: P2PSIP Overlay Reference Model

Here, the large perimeter depicted by "#" represents a stylized view of the P2PSIP Overlay (the actual connections could be a mesh, a ring, or some other structure). Around the periphery of the P2PSIP Overlay rectangle, we have a number of P2PSIP 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, two peers

("P" and "Q") which are each coupled with a SIP proxy, and one peer "R" which is coupled with a SIP Redirector. Note that because these are all P2PSIP Peers, each is responsible for storing P2PSIP Resource Records and transporting messages around the P2PSIP 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 P2PSIP Overlay, we have a conventional SIP UA "A" which is not part of the P2PSIP Overlay, either directly as a peer or indirectly as a client. It speaks neither the P2PSIP Peer nor P2PSIP Client protocols. Instead, it uses SIP to interact with the P2PSIP Overlay.

On the right side, we have a P2PSIP client "C", which uses the P2PSIP Client Protocol depicted by "=" to communicate with Proxy Peer "Q". The P2PSIP 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 5.5).

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 P2PSIP 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 P2PSIP 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 P2PSIP 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. <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] 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 services. 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).

P2PSIP Overlay: A P2PSIP Overlay is an association, collection, or federation of nodes that provides SIP registration, SIP message transport, and similar functions using a P2P organization, as defined by "P2P Network" above.

P2PSIP 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.

P2PSIP Peer: A node participating in a P2PSIP Overlay that provides storage and transport services to other nodes in that P2PSIP Overlay. Each P2PSIP Peer has a unique identifier, known as a Peer-ID, within the P2PSIP Overlay. Each P2PSIP Peer may be coupled to one or more SIP entities. Within the P2PSIP 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.

P2PSIP Peer-ID: Information that uniquely identifies each P2PSIP Peer within a given P2PSIP 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).

P2PSIP Client: A node participating in a P2PSIP Overlay that is less capable than a P2PSIP Peer in some way. The role of a P2PSIP Client is still under debate, with a number of competing proposals, and some have suggested removing the concept entirely (see the discussion on this later in the document). If clients exist, then 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: A human that interacts with the overlay through SIP UAs located on peers and clients (and perhaps other ways).

P2PSIP 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.

P2PSIP Service: Actions that a peer (and perhaps a client) can do for the benefit of other peers and clients. Examples include acting as a STUN server, and acting as a voicemail server. 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.

P2PSIP Service Name: A unique, human-friendly, name for a service.

P2PSIP Resource: Anything about which information can be stored in the overlay. Both Users and Services are examples of Resources.

P2PSIP 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.

P2PSIP Resource Record: A block of data, stored using distributed database mechanism of the P2PSIP 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.

P2PSIP Peer Protocol: The protocol spoken between P2PSIP Overlay peers to share information and organize the P2PSIP Overlay Network.

P2PSIP Client Protocol: The protocol spoken between P2PSIP Clients and P2PSIP 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).

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

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

P2PSIP Joining Peer: A node that is attempting to become a P2PSIP Peer in a particular P2PSIP Overlay.

P2PSIP Bootstrap Peer: A P2PSIP Peer in the P2PSIP Overlay that is the first point of contact for a P2PSIP Joining Peer. It selects the peer that will serve as the P2PSIP Admitting Peer and helps the joining peer contact the admitting peer.

P2PSIP Admitting Peer: A P2PSIP Peer in the P2PSIP Overlay which helps the P2PSIP Joining Peer join the Overlay. The choice of the admitting peer may depend on the joining peer (e.g., depend the joining peer's P2PSIP 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.

P2PSIP Bootstrap Server: A network node used by P2PSIP Joining Peers to locate a P2PSIP Bootstrap Peer. Typically, a P2PSIP Bootstrap Server acts as a proxy for messages between the P2PSIP Joining Peer and the P2PSIP Bootstrap Peer. The P2PSIP 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 P2PSIP Bootstrap Server is NOT required to be a peer or client, though it may be if desired.

P2PSIP Peer Admission: The act of admitting a node (the "P2PSIP Joining Peer") into a P2PSIP Overlay as a P2PSIP 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.

P2PSIP 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 P2PSIP Resource-ID as a key.

5. Discussion

5.1. The Distributed Database

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.

5.2. Using the Distributed Database

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] 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].

5.3. NAT Traversal

Two approaches to NAT Traversal for P2PSIP Peer Protocol have been suggested. The working group has not many 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] 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] of NAT Traversal, and in particular the ICE [I-D.ietf-mmusic-ice] 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], the

current recommendation is to use ICE.

5.4. Locating and Joining an Overlay

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.

5.5. Possible Client Behavior

As mentioned above, a number of people have proposed a second type of P2PSIP entity, known as a "P2PSIP client". The question of whether the concept of a "client" is needed and, if it is needed, its exact nature, is still very much under debate. This section presents some of the alternatives that have been suggested for the possible role of a client.

In one alternative, 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. Though interactions with its associated peer, a client can insert, modify, examine, and remove resource records. A client can also send SIP messages to its associated peer for routing through the overlay. In this alternative, 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.

One way to realize this alternative is for a peer to behave as [RFC3261] proxy/registrar. Clients then use standard SIP mechanisms to add, update, and remove registrations and to send SIP messages to peers and other clients. If this is done, there is no need for a separate Client Protocol and no need for P2PSIP to define a distinct "P2PSIP Client" concept.

In a second alternative, a client behaves in a way similar to the way described in first alternative, except that it does store resource records. In essence, the client contributes its storage capacity to its associated peer. A peer which needs to store a resource record may elect to store this on one of its associated clients instead, thus boosting its effective storage capacity.

In a third alternative, a client acts almost the same as a peer,

except that it does not store any resource records. In this alternative, a client has a "peer-ID" and joins the overlay in the same way as a peer, perhaps establishing the same network of connections that a peer would. Clients participate in the distributed database algorithm, and can help in transporting messages to other peers and clients. However, the distributed database algorithm does not assign resource records to clients. The role of a client in this model has been described as "a peer with bad memory".

In these three alternatives, clients can act as SIP entities and make and receive calls on behalf of users. They can also use services offered by peers, and depending on the details of how services are defined, they may also be able to offer services to other clients and peers.

5.6. Interacting with non-P2PSIP entities

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]. 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]) and this peer then looks up the user in the distributed database and proxies the call onto it.

6. Additional Questions

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.

6.1. 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.

6.2. 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?

6.3. Hybrid Domains

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?

7. Security Considerations

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.

8. IANA Considerations

This document presently raises no IANA considerations.

9. Open Issues

Here are some open issues in this document:

1. The word "service" is being overloaded to refer both to what the overlay provides (distributed location service, distributed transport service) and what an individual peer or client provides. How can we fix this?

2. Does P2PSIP provide a distributed location service or an alternative mechanism to RFC 3263? The answer seems to be both, but what is the relationship between these?

10. Acknowledgements

This document draws heavily from the contributions of many participants in the P2PSIP Mailing List but the authors are especially grateful for the support of Spencer Dawkins, Cullen Jennings, and Henning Schulzrinne, all of whom spent time on phone calls about this document or provided text. In addition, Spencer contributed the Reference Model figure.

11. References

11.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[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.

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

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

11.2. Informative References

[I-D.bryan-p2psip-dsip]
Bryan, D., "dSIP: A P2P Approach to SIP Registration and Resource Location", draft-bryan-p2psip-dsip-00 (work in progress), February 2007.

[I-D.bryan-sipping-p2p-usecases]
Bryan, D., "Use Cases for Peer-to-Peer Session Initiation Protocol (P2P SIP)", draft-bryan-sipping-p2p-usecases-00 (work in progress), December 2005.

[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.

[I-D.ietf-behave-rfc3489bis]
Rosenberg, J., "Simple Traversal Underneath Network Address Translators (NAT) (STUN)", draft-ietf-behave-rfc3489bis-05 (work in progress), October 2006.

[I-D.ietf-mmusic-ice]
Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-13 (work in progress), January 2007.

[I-D.ietf-sipping-nat-scenarios]
Boulton, C., "Best Current Practices for NAT Traversal for SIP", draft-ietf-sipping-nat-scenarios-06 (work in progress), March 2007.

[I-D.irtf-p2prg-survey-search]
Risson, J. and T. Moors, "Survey of Research towards Robust Peer-to-Peer Networks: Search Methods", draft-irtf-p2prg-survey-search-00 (work in progress), March 2006.

[I-D.johnston-sipping-p2p-ipcom]
Sinnreich, H. and A. Johnston, "SIP, P2P, and Internet Communications", draft-johnston-sipping-p2p-ipcom-02 (work

in progress), March 2006.

[I-D.matthews-sipping-p2p-industrial-strength]
Matthews, P., "Industrial-Strength P2P SIP", draft-matthews-sipping-p2p-industrial-strength-00 (work in progress), February 2005.

[I-D.shim-sipping-p2p-arch]
Shim, E., "An Architecture for Peer-to-Peer Session Initiation Protocol (P2P SIP)", draft-shim-sipping-p2p-arch-00 (work in progress), March 2006.

[RFC3424] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

[RFC4101] Rescorla, E. and IAB, "Writing Protocol Models", RFC 4101, June 2005.

[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

David A. Bryan
SIPeerior Technologies and College of William and Mary
3000 Easter Circle
Williamsburg, Virginia 23188
USA

Phone: +1 757 565 0101
Email: bryan@sipeerior.com

Philip Matthews
Avaya
1135 Innovation Drive
Ottawa, Ontario K2K 3G7
Canada

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
Cisco Systems
3100 Independence Pkwy #311-164
Plano, Texas 75075
USA

Phone: unlisted
Email: dean.willis@softarmor.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).