Abstract

The P2PSIP working group’s charter charges the group to define a P2PSIP peer protocol that defines how peers "collectively provide for user and resource location in a SIP environment with no or minimal centralized servers." The charter also states that the group may define a peer protocol that is syntactically based on SIP. This document outlines the motivation and merits of using conventional SIP messages as the syntax to encode the P2PSIP peer protocol and
discusses arguments made against this design decision.

Table of Contents

1. Introduction .................................................. 3
2. Requirements .................................................. 3
3. Why SIP should be used ........................................ 3
   3.1. Proposed SIP uses ...................................... 4
4. Why SIP implementation is easy .............................. 5
5. Arguments against SIP as the Peer Protocol ............... 5
6. Summary ....................................................... 7
7. Security Considerations ...................................... 8
8. IANA Considerations .......................................... 8
9. Informative References ....................................... 8
Authors’ Addresses ............................................... 9
Intellectual Property and Copyright Statements ............. 10
1. Introduction

There are several reasons to use SIP messages to implement the P2PSIP Peer Protocol. In order to properly implement a P2PSIP protocol, it is necessary to have mechanisms to store, retrieve, and query the locations of resources, as well as to route information. Pluggable DHT algorithms require the Peer Protocol to support negotiation of protocol extensions. Several NAT traversal and security mechanisms must be supported by the Peer Protocol. We have found that SIP offers mechanisms that meet all of these requirements today, has well defined security mechanisms, and additionally works well with the IETF suite of NAT traversal techniques: STUN, TURN, and ICE. Because all this work would need to be redefined in a new P2PSIP Peer Protocol, and because all P2PSIP devices must, by definition, implement SIP anyway, we feel the only reasonable syntax choice for the P2PSIP Peer Protocol is SIP.

2. Requirements

The following requirements are based on the P2PSIP working group’s charter.

1. A P2PSIP Peer Protocol must provide a distributed mechanism for locating resources (users) in a SIP environment with minimal use of centralized servers.
2. The protocol must work, as much as possible, in environments that contain NATs and firewalls.
3. SIP must be used for session establishment. The charter further states that P2PSIP protocols cannot modify baseline SIP behavior or create a new or similar protocol for session establishment.
4. Peer protocols should define one base P2P algorithm but be designed for pluggable P2P algorithms to be interchanged.
5. Peer protocols should provide security mechanisms that authenticate data being stored in the overlay, allow clients to establish authenticated signaling dialogs through untrusted relays, and possibly address privacy issues of peers storing data or routing requests for other peers in the system.
6. The charter states that the scope of the working group is to create P2P SIP protocols and explicitly forbids (first thing excluded on the charter) creating a more general distributed lookup for use with applications other than SIP.

3. Why SIP should be used

Fundamentally, the DHT overlay used in P2PSIP is a distributed registrar. In this sense, queries that would generally be sent to a
conventional registrar are now performed within the DHT. Similarly, once the target resource has been located, further communication proceeds directly between the UAs (or designated adapter peers) as with conventional SIP communications. We see no reason that any messages used to register, locate, or communicate with other users should be implemented in P2PSIP by anything other than SIP.

The second class of messages that must be exchanged in a P2PSIP overlay is inter-peer communication. These messages are used to join, leave, and maintain the overlay; transfer registrations between peers; and to route SIP messages between endpoints to establish communication. We see three reasons to use SIP for these messages:

- Joining and maintaining the overlay primarily consists of establishing the connections for each peer’s routing table that form the topology of the overlay. These connections form long-lived sessions over which the peers exchange information and route requests. As such, these sessions are appropriately established using SIP, the Session Initiation Protocol.
- As the registrations stored are intended for use by SIP, they must be stored in a SIP format, and using SIP to exchange and refresh them seems natural.
- The overlay is needed to route messages between endpoints for NAT traversal. Because the messages between P2PSIP endpoints will be SIP messages, routing them using SIP and leveraging the routing capabilities already built into SIP makes sense.

NAT traversal must work for both inter-peer traffic and end-to-end session establishment traffic. If routing behavior is needed for relay anyway, it is a clear win to have both DHT messages and the session establishment messages get routed by the peers -- INVITEs between Alice and Bob get routed by the same peers that pass DHT SIP messages. The NAT traversal problem needs only be solved once if routing of DHT and session establishment messages are both SIP and the solution can leverage existing NAT traversal solutions proposed for conventional SIP.

3.1. Proposed SIP uses

Two of the ways in which SIP can be used to encode P2PSIP Peer Protocols are to 1) use newly defined headers for SIP messages to pass DHT information and 2) embed the DHT information in the body of the SIP messages. In either case, the work already done in the areas of NAT traversal, security, etc. are leveraged by the peer protocols using SIP as the syntax.

The dSIP [I-D.bryan-p2psip-dsip] P2PSIP Peer Protocol proposes the first approach where newly defined headers are used to pass DHT
information. The dSIP draft outlines using SIP REGISTER messages to manipulate and maintain the DHT. dSIP and RFC 3261 both use REGISTER messages to "add, remove, and query bindings." The difference between dSIP and SIP, is that in addition to distributing the conventional SIP registrar functionality across the overlay, dSIP uses the REGISTER messages to bind neighbors in DHT maintenance operations.

We feel that the similarity of registering a Contact mapping and registering a peer entry in a routing table are similar enough to justify the use of REGISTER, but acknowledge that whether the REGISTER method or a new method is used for DHT maintenance purposes is still an open question. One example of such a new method is in the P2PSIP Peer Protocol, written by Hautakorpi and Camarillo[I-D.hautakorpi-p2psip-peer-protocol], proposes using a new SIP method, LOCSE, with a well-defined XML message body.

4. Why SIP implementation is easy

P2PSIP Peer Protocols that use SIP syntax do not have to implement new stacks to handle P2P maintenance messages. P2PSIP Peer Protocols must use SIP for session establishment, so a SIP stack will be part of P2PSIP implementations. Rather than implement multiple stacks, the SIP stack can be minimally modified to support P2P while not breaking interaction with conventional SIP clients and servers. This saves development time and also saves the overhead (processing time, runtime footprint, memory) that would be involved in implementing an additional stack.

5. Arguments against SIP as the Peer Protocol

Baset and Schulzrinne devote Section 4.2 of their P2PSIP peer protocol draft [I-D.baset-p2psip-p2pcommon] to argue why SIP should not be used as the underlying syntax for a P2PSIP protocol.

Argument 1: SIP is a session establishment protocol SIP is a protocol for initiating, modifying, and terminating interactive sessions[RFC3261]. Communicating with other nodes to form and maintaining an overlay does not create an interactive session.

Response: Communicating or registering your user information with a SIP server/proxy does not create an interactive session either. In plain SIP and P2PSIP, registering resource information does not create the session, but rather enables you to create that session when needed.
Note: we’ll classify "communicating with other nodes to form and maintain an overlay" as dht maintenance for simplicity in the remainder of this document.

Argument 2: SIP is not a general Remote Procedure Call (RPC) A peer-to-peer protocol designed on top of SIP will merely use it as a remote procedure call (RPC) mechanism. The SIP guidelines document [RFC4485] prohibits the use of SIP as a RPC mechanism.

Response: The argument that registering a contact for an AoR, as done by the conventional SIP REGISTER, is not an RPC, but registering a peer in a DHT routing table is an RPC seems dubious. DHT maintenance does not use SIP as an RPC mechanism. The dht maintenance between nodes is an (distributed) equivalent of registering resources with SIP servers. If you don’t keep a list of how/where to contact resources, sessions can’t be established at all. The point is that maintaining information on who to send INVITEs to is done via dht maintenance in P2PSIP and therefore, using SIP messages is appropriate.

The justification in RFC 4485 for not supporting general-purpose RPC is that "None of its user discovery and registration capabilities are needed for RPC, and neither are most of its proxy functions." For establishing connections between peers in dSIP, the discovery, registration, and proxy functions are all integral and necessary parts of the DHT maintenance phase.

Quoting RFC 4485, "The primary purpose of SIP is a rendezvous function, to allow a request initiator to deliver a message to a recipient wherever they may be. Such a rendezvous is needed to establish a session, but it can be used for other purposes related to communications, such as querying for capabilities or delivery of an instant message." DHT maintenance is a necessity for discovery and registration -- it implements a rendezvous function which is clearly called out as being in scope of SIP by name in RFC 4485. The P2PSIP working group is specifically chartered to create a P2P rendezvous for SIP, so using SIP as the underlying messages (i.e. in dht maintenance) in the P2P rendezvous function seems to be very appropriate.

Argument 3: SIP is not a general purpose transfer protocol One possible way to design the peer-to-peer protocol is to incorporate it as a SIP message body, possibly in XML, leaving the SIP headers unchanged. The peer-to-peer protocol is, however, unrelated to SIP’s operation and the SIP guideline document [RFC4485] prohibits sending large amounts of data unrelated to SIP’s operation. Such a mechanism would use SIP as a RPC, which, as stated earlier, is prohibited by the SIP guidelines document.
Response: DHT maintenance information is related to SIP’s operation as a registrar in a distributed fashion. It is rendezvous and user location, explicitly called out as in scope and the task that the P2PSIP working group is chartered to convert to P2P.

Argument 4: SIP is not a lookup protocol  SIP is not a lookup protocol; it relies on DNS to locate an appropriate SIP server. Insert and lookup functionality is essential for any peer-to-peer protocol. Using SIP as a peer-to-peer protocol requires integrating lookup functionality into SIP which goes against its design philosophy.

Response: To address this argument, we will use the following distinction between lookup and discovery: lookup is similar to looking up a value for a known key in a hash table, whereas discovery is like getting a new key as well as its value.

Saying that SIP does discovery but not lookup is a little disingenuous. While one could certainly make the case that SIP endpoints only do discovery, they in general rely on proxies and registration servers, and these servers certainly do lookups. Because peers are performing the functions of a SIP registrar, the lookup aspect is just more obvious.

Taken another way, is this argument claiming we need something other than SIP so it can do general lookup? We are not doing (or at least should not be as chartered!) a general P2P lookup. We are making a P2P rendezvous for SIP, and the maintenance of this is part of sharing that information among entities in the system, just as register is used today to share that information between endpoints and the registrar in conventional SIP.

Argument 5: SIP headers overhead  SIP has headers that are not necessarily needed for a peer-to-peer protocol. Examples of such headers are Call-ID, and Contact.

Response: It’s really too early to tell if Call-ID and Contact (or others) are headers that aren’t useful for p2psip using SIP as the underlying protocol. dSIP relies on both of these headers in its messages. In general, many of the headers will be used for the same purposes because the purposes are very similar: identifying dialogs, request/reply/replay prevention, routing options, security, etc.

6. Summary

We find there to be significant and compelling reasons to use SIP as
the Peer Protocol for P2PSIP. SIP is a mature, established protocol that already has well-defined mechanisms to support the necessary capabilities of routing options, security, NAT traversal, and extensibility. SIP can be extended for use as the Peer Protocol through the addition of a small number of new headers, message bodies, or methods as demonstrated by several submitted drafts. We question whether an entirely new protocol with similar power and flexibility could be designed that would be significantly simpler and easier to implement than the necessary components of SIP. Furthermore, because all P2PSIP UAs must already have a SIP implementation, using SIP as the Peer Protocol is likely to be simpler and lower-cost than including a separate stack for an unrelated Peer Protocol. dSIP [I-D.bryan-p2psip-dsip] and The Peer Protocol [I-D.hautakorpi-p2psip-peer-protocol] present two possible methods for implementing a Peer Protocol over SIP, and we hope the working group will produce and evaluate a number of candidates to determine the best mechanism for implementing the Peer Protocol in the near future.

7. Security Considerations

There are important security considerations in any Peer Protocols. Each addresses its issues in their respective drafts.

8. IANA Considerations

This document has no actions for IANA.

9. Informative References

[I-D.baset-p2psip-p2pcommon]

[I-D.bryan-p2psip-dsip]

[I-D.hautakorpi-p2psip-peer-protocol]


Authors’ Addresses

Marcia Zangrilli
SIPeerior Technologies
3000 Easter Circle
Williamsburg, VA  23188
USA

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

Bruce B. Lowekamp
SIPeerior; William & Mary
3000 Easter Circle
Williamsburg, VA  23188
USA

Phone: +1 757 565 0101
Email: lowekamp@sipeerior.com
Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

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

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

Acknowledgment

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