P2PSIP WG D. Bryan/SIPeerio-editor
Internet Draft S. Baset/Columbia U.
Intended status: Informational M.Matuszewski/Nokia
Expires: Jan 2008 H.Sinnreich/Adobe

P2PSIP Protocol Framework and Requirements
draft-bryan-p2psip-requirements-00.txt

Status of this Memo

Copyright Notice

Abstract

Though both SIP and various peer-to-peer (p2p) protocols have been widely deployed, there is no operational experience with SIP using overlay networks. Also, most p2p networks developed in the research community do not deal with NAT traversal. This document attempts to list the design requirements for a P2PSIP protocol taking into account the experience gained from both the p2p and the SIP

communities. Special emphasis has been put on the SIP-DHT interface, on the overlay performance, on NAT traversal and on security.

Conventions used in this document

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

Table of Contents

1. Introduction

Recently, the IETF has established a P2PSIP WG [2] to discuss the development of a protocol to allow the establishment of SIP connections with little or no need for centralized servers of any kind. The WG (and individuals now composing it prior to the official formation) have focused on developing a DHT based system that can be used to achieve these ends. A concepts document [3] has been produced that outlines the basic ideas and concepts of how such a system would be structured.

This document attempts to accomplish a few things. It attempts to combine and present requirements for designing the peer and (if needed) client protocols, attempts to provide some insight into selection of mandatory DHTs, and attempts to present requirements for other DHTs that may be added as well.

Much of what is presented here is also a collection of insights and references to material assembled by the authors that will assist in these decisions. As the authors all have different (and sometimes conflicting!) ideas about how these systems should be implemented, this document may present multiple alternatives, or even mention points of contention. As the group reaches more consensus, it is expected that these differences will diminish.

This is a first revision of this document, and is the result of work by authors on several continents. As such, inconsistencies and even outright errors are likely to appear. The authors recognize this and are happy to have the work criticized to help improve it. In addition, the document is certainly incomplete, and there are areas (most notably the security section) where additional help is required to complete this document. We hope the discussion generated can offset the early stage of this work.

This document on the requirements for the p2p protocols is based on published research and also on running code for p2p systems. We have carefully avoided articulating any requirements that cannot be substantiated by both usage scenarios and by published work based on running code, simulations and measurements on deployed P2P systems. We have provided for this reason references that are pertinent to each specific requirement.

This document does NOT attempt to replace the concepts [3] in any way, but rather extend it to discuss requirements.

OPEN ISSUE: Should this document eventually be split (either into two documents or simply within this document) into sections on P2PSIP protocol decisions vs. DHT decisions?

OPEN ISSUE: Should this document eventually be split (either into two documents or simply within this document) into sections on P2PSIP protocol decisions vs. DHT decisions?

2. Deployment Scenario Examples

Generic applications for P2PSIP have been discussed in [4], [5] and we are adding some more detailed deployment scenarios here.

The use of innovation in the market is usually hard to predict. Still, P2PSIP can be deployed in many scenarios; out of which we have selected here examples that could be most frequently related to present SIP deployments. New scenarios may however emerge with increased use of P2PSIP.

An overall view of possible deployment scenarios would require a table that may be too big to fit into the ASCII format of this memo. For this reason, we will rather enumerate here the main categories that are encountered in deployment scenarios:

- o Degree of mobility: Stable (PC, gateway), nomadic (laptop), mobile (phone, PDA)

- o NAT scenario: Any type of NAT, BEHAVE compliant NAT, no NAT

- IP connectivity: Stable, intermittent.

Of special interest to wireline (DSL, cable, fiber) service providers is the deployment scenario where the peer nodes are located in the network access devices of their customers, be they residential gateways or enterprise gateways.

Peers located in access devices enjoy major benefits, such as:

- High bandwidth on the Internet side
- Stable connectivity
- No NAT in some provider network designs.

P2PSIP nodes in the access devices seem to be a valid replacement of the conventional client-server SIP VoIP infrastructure, while combining all the benefits of the end-to-end principle of the Internet with the control desired by service providers or network administrators via the software in the access device.

Mobile service providers can reap the benefit of P2PSIP by placing peer nodes into the base station network and using a battery sparing client protocol in the mobile devices.

The deployment scenarios may suggest a wide variety of protocol parameters or several DHT types for various deployments.

In the following sections, we will use the terms P2P network and P2P overlay interchangeably.

2.1. Peer Protocol Deployment

Though the mobility, churn rate and session time are quite different for adapters and access devices versus a laptop, it still make sense to use a single peer protocol:

- The IP address changes even for fixed devices
- Devices with higher churn rate and low session time will benefit

from the existence of stable peers

- o IP connectivity may be very good for one peer but may be less satisfactory for other remote peers.
- o Interoperability will be improved, making it more likely that many devices can function together in one overlay.

Designers may be tempted to pick the simplest DHT peer protocol for the deployment scenario at hand, but run the risk of loosing the flexibility offered by a full featured peer protocol. Additionally, choosing a protocol that only works in limited environments means those devices cannot be used in more challenging environments.

2.2. Client Protocol Deployment

The small battery capacity of handheld mobile devices requires reducing the amount of messages between the mobile device and the fixed network to an absolute minimum. The client protocol has therefore to be designed to avoid the large message exchange required for the maintenance of the P2P overlay.

A single standard client protocol is desirable for interoperability.

Wireless access points, wireless radio base stations may also contain DHT peers so as to facilitate mobile devices running only the client protocol.

To date, the working group has not decided if such a protocol is needed, or if unmodified SIP can serve this role.

3. NAT Traversal

As mentioned in the Concepts and Terminology document on P2PSIP, it is expected the majority of SIP peers will reside behind NAT. This raises the issue to what extent can P2PSIP be successfully deployed, since it will depend on effective NAT traversal techniques.

The state of P2P communications across NAT is discussed in [6]. In the light of this document, P2PSIP must be a NAT-friendly application, that is:

"A NAT-friendly P2P application registers with a well-known rendezvous server, used for node registration and peer node discovery purposes. Pursuant to registering with rendezvous server, a P2P-friendly application uses its private endpoint, public endpoint, or a combination thereof to establish peering sessions."

P2PSIP can be a NAT-friendly application, by using ICE [7]. According to surveys such as [8], practically all consumer grade NAT devices are P2P-friendly, so ICE can be used effectively for NAT traversal consumer devices.

Symmetric NAT is however not P2P-friendly and NAT traversal may not succeed without using STUN/TURN relay servers [9]. Symmetric NAT is almost always deployed in private enterprise IT networks. Since P2PSIP must be used in enterprise networks with the consent and cooperation with the IT network administrator, measures can be taken in private IP networks to deploy IETF BEHAVE standard compliant NAT devices that are P2P friendly. The use of TURN may be required for facilities where this is not possible, and SHOULD be included in proposals.

4. Bootstrap and Other Servers

A small number of servers is required for the proper operation of the overlay network. We discuss here briefly for clarity only these types of servers though they do not affect the P2PSIP protocol design.

- The P2P overlay is established with the help of bootstrap servers.

- As mentioned in the section on NAT traversal, P2PSIP must be NAT-friendly and thus also requires well-known rendezvous servers, such as specified for the ICE protocol.

- Security for the overlay must be based on strong authentication provided by an authentication server. P2PSIP does not depend on the particular implementation of the authentication server and therefore the authentication server is not discussed in this document.

The bootstrap servers and the rendezvous servers may or may not be placed on the same machines and this is a design decision that does not affect the P2PSIP protocol.

A recent I-D on NAT traversal specifically for P2PSIP has been published [10], showing that NAT traversal using ICE techniques are quite effective.

5. SIP-P2P Interface

In a typical SIP trapezoid, a caller in domainA wishing to contact a callee in domainB sends a request to its proxy server (proxyA), which following the guidelines in [1] forwards the request to the proxy server of domainB (proxyB). When the request arrives at proxyB, it consults its location service (typically a database) to find the IP address of the callee, and forwards the request to callee SIP user agent.

A characteristic of this architecture is that the typical SIP hop-count is only a few hops; the caller only needs to send its request to its outbound SIP proxy, which in turn needs to locate the SIP server of the callee. The key assumptions of this architecture are that SIP proxy servers use DNS to locate other proxy servers, maintain a database of SIP user agents in its domain, are fairly stable and trusted.

Peer-to-peer systems on the other hand distribute the task of locating the SIP proxy servers, and proxy servers locating SIP user agents to the peer themselves. This means that SIP requests no longer traverse through reliable and trusted proxy servers and the SIP hop-count of a request can be significantly greater than a few hops; in a DHT it is log(N), where N is the number of peers.

The above discussion suggests at least two paradigms for SIP operation in a p2p setting: the end-to-end paradigm where a SIP user

agent uses the p2p location service to discover the location of callee, and then send the SIP message directly to the callee, or a hop-by-hop paradigm where each peer forwards the SIP request to a peer which is more 'closer' to the callee. The former can be thought of as a RPC whereas the later can be thought of as a local procedure call to determine the next hop.

SIP uses an address-of-record (AOR), to locate the user. An AOR is a SIP or SIPS URI and frequently thought as the "public address" of the user. DHTs are a prime motivation for P2PSIP and they convert a blob of text to a unique hash value. A DHT will generate a unique hash for a SIP or SIPS URI for the same user, and store it on different peers. Since a peer-to-peer system may suffer through churn, it may lead to a situation where a SIP URI is reachable but SIPS URI is not or vice versa. The impact of such distributions due to the underlying P2P service must carefully be analyzed.

A SIP user agent can participate in the traditional c/s SIP or a P2PSIP network at the same time. Such a UA must have a predictable ordering of what to do when presented with a SIP URI.

SIP relies on STUN, TURN and ICE for NAT traversal. The P2P service will likely incorporate NAT and firewall mechanisms for the maintenance of the overlay. The SIP application may take advantage of NAT traversal mechanisms provided by the underlying P2P service.

It is important to note that the SIP-DHT interface and the peer protocol should not depend on one particular DHT. While having one DHT as must implement is necessary for interoperability, there is a danger that specifics from that DHT may creep into the overall design. Moreover, there is no 'clear winner' out of the DHTs proposed so far. The research in DHTs is ongoing and therefore the must-implement DHT should in no way restrict the SIP-DHT interface and the peer protocol from incorporating a DHT in the future that performs significantly better. Any attempt to incorporate DHT-specific information will greatly undermine the flexibility of the SIP-DHT interface and the peer protocol. The Peer-to-Peer Protocol (P2PP) [27] is a proposal that attempts to incorporate these concerns.

Req 5-1: The P2PSIP protocol MUST have a clearly defined interface between the SIP and overlay layers.

Req 5-2: The interface between SIP and the overlay MUST support the use of various types of overlays.

6. APIs for DHT Usage

DHTs provide two conceptual operations, namely, 'Lookup' to locate peers responsible for a key, and 'Publish' to insert a key-value pair in the DHT. In addition, a 'Remove' operation allows the publisher to explicitly remove a key-value pair. These operations suffice for nodes that do not participate in the DHT overlay. For nodes that do participate in the overlay, additional operations of 'Join' and 'Leave' are necessary. Thus, at the very least, the following API must be provided by the P2P layer:

- Join
- Leave
- Lookup
- Publish
- Remove

If a client protocol is implemented, the following API must be used:

- Lookup
- Publish
- Remove

Peer protocol will likely implement mechanisms for overlay maintenance during churn, and replication. However, an application using DHT does not need to be aware of them. If there is a need to fine tune these parameters, an API similar to set[get]sockopt() can be defined.

OPEN ISSUE: Is there a need to separately define the secure and non-secure version of these APIs?

7. P2P Overlay Requirements

SIP can be used with various types of P2P overlay networks. This document focuses on requirements for overlays on the Internet to leverage the global reach, inherent mobility, resilience and other attributes of the Internet.

At the same time, these requirements take also into account the impairments found on the public Internet, the lack of transparency due to NAT, as well as the various exposures to security for P2PSIP.

An important feature of the P2P design for global reach over the Internet is the self organizing characteristic of P2P networks, in

the sense that no network management and no peer node configuration is required to be performed manually.

Smaller P2PSIP networks such as found in private IP networks may have different requirements that are not addressed in this document.

As mentioned in below, designers can have a fair choice of the DHT they may want to deploy and this can be accommodated by the SIP-DHT interface discussed in Section 5.

Though the focus of this memo is on SIP applications, we take into account, whenever is possible, the fact that P2P overlays can be used for many other non-SIP applications. Implementers of P2PSIP may have compelling reasons to use an overlay for other applications beyond SIP that are however out of scope for the P2PSIP WG. For this reason, the flexibility of using a DHT for other applications must also be considered.

The default protocol must be a DHT but we should not preclude other overlay protocols from consideration.

7.1. Architecture and Virtual Geometries

One of the main reasons for selecting DHT overlays for large-scale systems is their resilience in the event of node failures.

Research on structured, DHT based overlays has revealed multiple architectures with various pros and cons that go beyond the scope of this memo, since the DHT routing architecture is only one of several selection criteria and by itself, is not of exclusive importance.

In this memo we focus only on structured, DHT functionality on Internet-like scale for the overlay network that can be used for SIP based multimedia communications.

For illustrative purposes, we assume a large DHT address space such as 160 bits that can be represented on a virtual circular space. A virtual circle is however not the only possible geometry. Pastry and its successor Bamboo also display a virtual tree-like geometry that facilitates routing based on increasingly matched prefixes.

Various tree-like routing architectures can be used for application level multicast, for conferencing, collaboration and for prefix based search - all of which are of interest for use in distributed SIP conferencing. Highly scalable application layer multicast systems have been developed in the research community [11]. Application level multicast can support a wide variety of applications [12].

7.2. Structured Overlays

Req 7-1: For scalability and predictability, a structured DHT overlay MUST be used as the default P2P protocol.

Examples of structured overlays include CAN, Chord, Bamboo, Kademlia, Pastry, Tapestry, and Viceroy [13].

DHT overlays have natural limitations that must be taken into account [14]. Two very different deployment scenarios illustrate the limitations a particular DHT has when trying to address all problems:

- Highly dynamic membership of peers: In this case only small amounts of data can be stored, limited by the amount of bandwidth available for the update of data stored in peers. Besides the maintenance of the overlay network structure may require additional resources such as bandwidth, CPU and memory. This is especially evident in the high network churn scenarios.

- Stable membership of peers: In this case large amounts of data can be stored since no frequent updates are required.

System designers must have the flexibility to target their design anywhere between these two extremes.

Req 7-2: The peer protocol MUST allow for flexibility in the DHT selected, to allow for deployments with various tradeoffs for churn, data size and bandwidth for reliable data storage. One (or a small number) of DHTs specified as mandatory MUST be supported, but protocol MUST allow for new DHTs to be added.

7.3. Data Replication

Most DHT schemas provide data replication in neighbor nodes that have overlay identifiers numerically close to the target node.

High reliability for the availability of replicated data can be assured because of the wide geographic diversity, IP network ownership and jurisdiction, etc, of neighbor nodes that have close neighbor identities [15], [16].

There is a trade-off between data replication being controlled by the data publisher or by the node storing the data.

Req 7-3: The P2PSIP Overlay MUST assure that no data placed in the overlay is lost before the scheduled timeout. The scheduled timeout for data on the overlay is a design choice. The protocol selected MUST allow for flexibility in how redundant a system is, since some deployments may require higher (or lower to conserve resources) levels of redundancy.

7.4. Load Balancing

The load balancing mechanism in DHT based overlays is provided by the statistical nature of the hash function and the algorithm of the DHT itself. Refinement in overlay protocols, such as reported for PAST over PASTRY [15] file storage have been described that attempt to improve the fairness of disk sharing among peers.

There are several other flavors of load balancing: One is to distribute data request among several replications. The other is to make the request routed through different overlay paths and reduce burden of hot spots.

Req 7-4: Dedicated load balancing schemas beyond the natural load balancing of the DHT MAY be used for fair disk storage sharing between peers as well as for balanced network usage. The DHT SHOULD allow an efficient load balancing algorithm that distributes the load on P2PSIP network entities that are responsible for storing frequently queried resources. While the authors would prefer to make this a MUST requirement, it is recognized that some specialized DHTs may be designed to prefer other factors (for example, geographical nearness) over fair load balancing. The default DHT MUST provide fair load balancing.

7.5. Overlay Performance

While other designs may be possible, there are two components of most DHTs that define how routing works:

1. Neighbor nodes for correct routing to the destination. The selection of neighbor nodes can take into account network proximity metrics such as latency, hop count or bandwidth of the neighbor nodes.

2. A DHT routing algorithm for fast, greedy search, generally a mechanism for determining which neighbor node to send the message to.

These two components of the DHT architecture will be discussed here in more detail.

7.5.1. Routing Performance

One of the criteria for selecting a specific DHT geometry is routing performance, meaning the average number of hops as a function of overlay network size. If we denote the complexity by O and the number of nodes in overlay by N, most architectures assure a number of hops proportional to O*log(N).

The routing performance MAY be improved by parallel queries, which reduce the delay impact of contacting a dead node.

Req 7-5: The peer protocol MUST accommodate a DHT for a fast routing algorithm that minimizes hop count to the root node. The routing algorithm MUST assure real time information retrieval. This also means that the delay of the information retrieval MUST be acceptable to users.

There are two fundamental types of neighbor nodes:

- Neighbor nodes in the hash table key space. Chord for example defines its neighbors as the successor nodes from itself [17], while Pastry [15] and Bamboo [18] define the so called leaf nodes that are at an equal distance before and after itself on the circle. Kademlia uses the mathematical distance based on XOR metric [19]. Such neighbor nodes may however be widely dispersed across the underlying IP network and this fact guarantees good geographic diversity.

- Neighbor nodes learned from the routing process in various searches that have terminated on the target root node. In Pastry and Bamboo the target root node makes a neighborhood list and can apply various routing metrics such as delay, number of IP hops or bandwidth to select the closest known peers. This is an extremely valuable feature for real time communications.

Req 7-6: The number of neighbor nodes MUST be selected high enough to insure the the loss of connectivity to a few nodes does not disconnect the peer and cause loss of data (although replication should mitigate this).

Req 7-7: The protocol DHTs MUST have the capability of applying various routing metrics in selecting neighbor nodes. Latency based neighbor selection (for example for use in NAT traversal) for SIP based real time communications SHOULD be supported by the peer protocol.

7.5.2. Routing Tables and Routing State

The number and size of routing tables varies for the different DHT types. Some DHTs keep only a table of logarithmically distributed peers, where others keep additional information. For example, Pastry and Bamboo have the following:

- o A routing table of known live peers. This table is used to execute fast search requests from other peers.
- o A list of the leaf set with its identifiers and IP addresses. The leaf set is used for accuracy in finding the root.
- o A neighborhood set selected by some routing metric, such as delay.

Routing state is understood here to mean the total data amount stored in the routing tables.

The routing state has to be maintained under churn. Devices may learn this by a number of ways, including from the routing events or/and by querying other nodes for their availability. The DHT maintenance mechanism may require setting up and maintaining TCP or TLS connections with neighbor nodes or other nodes in the routing tables.

In all battery-powered devices such as laptops, mobile phones, PDAs, or WiFi phones, power management is one of the key issues. The power consumption may be impacted by maintenance of the routing state and by the handling of lookups. These two mainly affect the idle state. The idle state represent a state when P2PSIP peer software runs in a battery-powered device but a user does not use any services provided by the P2PSIP overlay. The power consumption in the idle state determines how long a battery-powered device can stay online without recharging. Similarly, small capacity devices such as low-end desktop phones may have limited CPU and/or memory capacity. These limitations may result in different requirements for DHTs.

Req 7-8: The size of the routing state, that is the greedy routing table plus the tables of neighbor nodes SHOULD be kept equal or minimally larger than required by the routing table for the [14] routing protocol and MUST not grow faster than the log of the P2P overlay size. Note however that this does NOT mean that it must be

the log of the maximum size possible (for example, 160 neighbors for systems using 2^160 hash size) of the overlay, since this may be impractical (and undesirable) for small networks such as ad-hoc or enterprise. It also does NOT mean that additional neighbors may not be stored to improve efficiency if capacity is available. Because routing efficiency and local storage used are tradeoffs that may vary for different deployments, the balance of this is left to the various DHTs used (or the overlay designer)

Req 7-9: The maintenance of the routing state MUST consume minimum node and network resources.

7.5.3. Iterative and Recursive Routing Styles

Routing in the hash identifier space can be iterative or recursive, each having its advantages and drawbacks depending on the application.

o Iterative routing allows the source to control the routing process. The source peers can apply logic for security by checking the routing integrity. The number of NAT traversals is smaller and NAT traversal may be therefore easier. Besides the iterative routing offers increased robustness against message loss since the message is not relayed by many nodes in the overlay. The disadvantage of iterative routing is higher message traffic at the source and a slightly higher search delay. Further, iterative routing over TCP may result in establishing a per-hop TCP connection.

o Recursive routing is performed hop by hop and its advantages are that it is faster and has lower message traffic at the source. The disadvantage of recursive routing is a higher vulnerability to malicious nodes, since the routing integrity cannot be checked at the source, lower robustness against message loss and possibly higher number of required NAT traversals.

Other mechanisms (such as strictly forward routing) may also be appropriate and should be considered for incorporation here.

Note that because various circumstances (such as the presence of NATs) may only occur on some links, the protocol must allow a mixture of these routing mechanisms.

7.5.4. Handling of Join/Leave (Churn)

We define here the metrics of churn [20] that is relevant for SIP applications using as background the application scenarios for P2PSIP. There are two distinctive metrics for churn:

- Time online: The length of time when a peer node is online for the purpose of a real time communication session. This includes:
    - The length of time a mobile device is online
    - A PC online where the user is monitoring the presence list for buddies of interest or some other SIP events not related to real time communications
    - A network access device for a residential or enterprise network, which may be very nearly "always on".

- Lifetime: The length of time during which a peer node may appear online. This happens between the time of enrollment and when leaving the P2P service.

Churn in P2P networks has been studied extensively for various applications, but unfortunately not for SIP applications. At this point in time, we have to rely on studies on the handling of churn that have been done for other applications on various P2P systems.

The handling of peer nodes joining and leaving the overlay is a most critical decision. Various DHT algorithms have very different procedures for new nodes joining the overlay and leaving, mostly suddenly the overlay. Dealing with nodes leaving the overlay is more difficult. For example:

- o Chord has a stabilization protocol that runs in the background to check the successor nodes.

- o Pastry and Bamboo perform periodic check to verify the existence of the neighbor nodes and modify the routing tables accordingly.

Both Chord and Pastry may have additional algorithms for handling churn.

Reported test results include [23] comparing Chord and Bamboo for the latency of routing versus session time between active nodes. Note that such comparisons do not provide the whole picture in the presence of network impairments.

There are too many differing factors to compare between differing architectures and the effects of various features are hard to isolate. For this reason, it is not practical comparing the architectural differences of the various DHT implementations but rather focus on the important issues in handling churn. Simulations and testing can eliminate DHT types that show no promise and help in selecting a candidate DHT for the "must implement" DHT. Again, requiring modularity in DHT selection for the protocol can prevent the protocol from being tied to a poor choice or limited to only be appropriate for certain deployments.

Req 7-10: The default DHT selected for the peer protocol MUST incorporate techniques for handling churn. Joining, leaving, or searching the P2P overlay under high churn condition SHOULD not be significantly slower than these operations in the absence of churn.

7.5.5. Enabling Mobility

Handling nodes leaving the overlay introduces an additional traffic overhead in the overlay and also requires additional processing power to modify information in routing tables. When a peer rejoins the overlay network the overlay may have to perform more operations than in the case when a peer leaves a network. Additionally, updates to the DHT do not take place immediately after a node leaves.

Laptop computers, mobile phones, PDAs, and WiFi phones may move from one network to another by doing handover between different access networks or moving from one WLAN access point to another. When doing handover they may loose connectivity to the overlay for a short period of time. Most important for mobility is the fact they may receive a new IP address from the new access network.

In some DHTs a peer ID is based on the IP address. This means that when a node receives a new IP address it must also receive a new peer ID. This is equivalent to a node leaving the overlay network and joining it again. This may cause network instability and require additional resources to handle network adaptation (as explained above). This has impact on battery life of battery-powered as well as on system reliability. In the transition period search delay for user contact information may be higher, delaying the call establishment. For these reasons, the node ID SHOULD not depend on the IP address.

The peer protocol SHOULD support a mechanism for DHTs that allows network nodes to freely move from one access network to another without a need to reconfigure their peer IDs and without changes in the overlay network topology such as changes in neighboring sets or contacts in the routing tables. This requires only that the new address of a moving node propagates in real-time to the rest of the peers that need that information.

Req 7-11: The peer protocol MUST not interfere with mobility. A DHT MUST be available that supports mobility to allow network nodes to freely move from one access network to another without causing changes in the overlay network topology and decreasing routing performance. The information about new address of a node MUST propagate in real-time to the rest of the peers that need that information.

7.5.6. Fault Tolerance to Non-Transitive Connectivity

The various DHT architectures assume all peer nodes can communicate at all times but this is not true in practice [21]. The Internet has transient failures of connectivity due to link failures, route flapping due to equipment maintenance, faulty BGP routing table updates and ISP disputes. During such transient failures, pairs of nodes such as node A and node B can communicate with node C, but cannot communicate with each other. This is called non-transitivity. In addition, the presence of NATs can lead to systemic and semi-permanent non-transitivity.

Extensive testing on Planet Lab that includes nodes on Internet1, Internet2 and multihomed nodes has shown amounts of non-transitivity than can lead to significant problems for various types of DHT. Note this occurs even on Planet Lab, which is relatively closed, high quality, NAT-free academic environment which is far more forgiving than real-world deployments are likely to be.

Non-transitivity is actually rather the rule than the exception on the Internet since most peer nodes are likely to reside behind one or more NATs. The section on NAT traversal deals more with this aspect.

Firewalls can be another cause for non-transitivity, but we assume the network administrator will configure firewalls to pass P2PSIP messages if this behavior is desired. If local policy should prohibit the firewall to pass P2PSIP messages then it is not our intent to write standards requirements to circumvent local firewall policy.

The problems arising due to non-transitivity include the following:

- Invisible nodes when a peer learns about a node but cannot communicate with it
- Routing loops when a search skips a node around the virtual circle in some implementations
- Broken return paths. This may happen when the root node holding the information is found by using recursive routing and the result is sent back directly to the source node, but there is no connectivity between them.
- Inconsistent roots. The correctness of the key space can be affected if two nodes that cannot see each other believe each to be the correct root during a lookup.

Various DHT systems have different specific remedies to counter non-transitivity.

Req 7-12: DHTs designed for use with the P2PSIP protocol MUST take into consideration non-transitivity on the Internet.

7.6. Implementation Experience in Selecting DHTs

A significant aspect in choosing a DHT for P2PSIP is the glaring contrast between the huge number of research papers versus the extremely limited DHT overlays that have actually been deployed in large scale systems on the Internet.

There are other considerations as well, besides the routing aspects in choosing a DHT, such as:

- To what degree has a particular DHT been deployed and tested in large scale systems over the Internet?
- To what a degree has a particular DHT been researched in depth and the research results made public?
- To what degree has a DHT been used for various significant applications and can their use be extrapolated for P2PSIP?

Bamboo was deployed on the openDHT network on Planet Lab. Kademlia is the only DHT that was deployed in a large scale commercial systems.

Req 7-13: The DHT selected for the default DHT MUST be based on operational experience from large scale systems and measurements on the Internet or significant testing and research involving multiple peers.

7.7. Simplicity of Implementation in Selecting a DHT

Selecting a DHT as the "must implement" must also take into account the fact that many developers will be independently building implementations. While any algorithm, even one of arbitrary complexity, may work, and may even be more efficient, the likelihood of two independent implementations functioning properly together decreases greatly as the algorithm's complexity increases. While efficiency is extremely important, the "must implement" should be a safe fall back that can be easily and correctly implemented. More complex DHTs can be employed as additional DHTs.

Req 7-14: The DHT selected as the default DHT MUST be as simple and easy to implement as possible, while addressing the majority of requirements above.

This selection MUST be driven by an understanding of the difficulty of different developers producing interoperable, running code.

7.8. Discussion on the Selection of a DHT Overlay

Picking one or a small selection of DHT overlays for P2PSIP is a difficult proposition, given the very large number of solutions developed in the research community. To overcome this difficulty it is useful to look first at the important properties of various DHT without going into all the specific details for each. After choosing the key properties, various DHT can be compared as has been done in [22], [23]. The selection of a DHT for P2PSIP MUST be based on a thorough comparison and discussions in the IETF P2PSIP WG.

There are a large selection of DHT geometries, including the ring geometry, the XOR geometry, the tree and hypercube. Some DHT such as Pastry feature a dual geometry of both ring and tree. It is beyond the scope of this memo to discuss in detail the various geometries and we prefer to reference key literature on this topic. A detailed mathematical analysis of the various geometries in [23] shows the ring and XOR geometries to have the best performance using the above criteria, with the ring geometry showing a slight advantage.

8. Client Protocol Requirements

Note: While it is not yet clear if a Client protocol will be required (a number of members of the WG, and even some of the authors of this document, feel conventional SIP may be enough, although it is too early to tell), if one is required, the following requirements would apply to this protocol.

One assumption of DHT networks on the Internet could be that the network nodes are homogenous, and they have enough processing power and memory to run DHT networks without any limits. This assumption does not hold however if we consider mobile devices such as mobile phones, PDAs, or WiFi phones. Mobile devices are heterogeneous. Low-end devices may have a little memory, a slow CPU and may not support fast packet radio interfaces. On the other hand, high-end devices may have significantly higher capabilities: Fast radio interfaces, more memory and faster CPU then they their less advanced counterparts.

From this reason we can classify network nodes into two categories: Mainly those that have enough CPU power, memory and fast network interface to run P2PSIP peer protocol and those that may not have enough capabilities to become P2PSIP peers.

Additionally, if we consider the issue of NAT traversal, a very powerful mobile device can be behind a restrictive NAT. This may require the device to establish a TCP or TLS connection towards a TURN server. If the connection has to be maintained for a long period, it may drain the device battery. From these reasons, not all battery powered devices though having enough capabilities to run the peer protocol will still not be able to become P2PSIP peers.

Essentially, any DHT overlay network can support two basic operations: PUT and GET. In the P2PSIP overlay a PUT operation is used to insert, modify, or delete data in the overlay,such as user or resource records. A GET operation is used to retrieve data stored in the overlay. Clients should be able to store, modify, delete and retrieve user and resource records from the overlay; in the P2P language they must support PUT and GET operations.

Req 8-1: The client protocol MUST allow nodes that are not eligible to become peers to store, modify, delete and retrieve user and resource records stored in the P2PSIP overlay.

In all kinds of battery-powered devices such as laptops, mobile phones, PDAs, or WiFi phones power management is one of the key issues. As discussed in this section and section 8.7.2 traffic introduced by the overlay protocols may significantly reduce the battery life of battery powered devices. This does not allow some of the nodes to become peers. Those devices will have to act as clients and implement the client protocol that conserves the client's battery.

Req 8-2: The client protocol must be efficient enough to conserve the client's battery.

As mentioned, there are many situations when powerful mobile devices cannot be elected as peers. This may happen, e.g. if the access network provides a small bandwidth interface or if the user is not willing to assume the cost of increased bandwidth consumption related to the P2PSIP application.

Req 8-3: The client protocol MUST introduce low message traffic to preserve bandwidth.

User and resource records may include different data that may be of no use to some applications. Transferring all of the data stored in user and resource records over the network interface (especially over the air interface) may be undesirable in some situations since it will increase traffic in the network. Additionally a user may make only small changes to its own records stored in the overlay. In this case it should be possible to update only a part of the user record stored in the overlay without a need of uploading the whole user record to the responsible peer.

From these reasons, there should be a mechanism to allow clients to indicate what data they want to modify, delete, or retrieve.

Req 8-4: The client protocol MUST enable peer clients to selectively indicate the data they are interested on. The clients MUST be able to retrieve, modify or delete a selected part of user and resource records.

In order to support the iterative routing style the client protocol must support redirection from a peer to another peer or other network node. The peer must be able to send redirect response to a request originated by a client with the address of a node that is better suited to handle the request.

Req 8-5: The client protocol MUST support redirection of requests from one peer to another network node.

8.1. Discussion about the Client Protocol

The realisation of the client protocol may have many forms. The widely deployed protocols on the Internet that are used to manipulate data stored remotely are using HTTP as the transport and such as the XML data format as the encoding. A Remote Procedure Call (RPC) protocol such as XML RPC is one of the options for the client protocol. XML RPC [24] is used in OpenDHT [25] deployed in Planet Lab [26] that is designed to support long-running services for a client base. This protocol is also proposed in [28] as an ASCII based client protocol. A binary protocol is yet another possibility.

In this option SIP is used for multimedia session establishment between endpoints whereas XML RPC or a similar protocol is used as a lookup protocol providing an access to SIP data stored in the P2PSIP overlay that is required to setup a multimedia session.

Another option is to send XML documents in SIP messages or modify SIP to support functionality needed for the client protocol. It has been noted that SIP is a session establishment protocol, not a generic lookup protocol; therefore it should not be used for general Remote Procedure Call (RPC).

It seems to be a more reasonable solution to support conventional SIP UAs (unmodified SIP devices). In this alternative the P2PSIP overlay would have to appear as a conventional SIP network to SIP UAs. This solution does not require any changes to the existing SIP devices, which can use SIP services provided by the overlay without even noticing its existence. However it introduces the very strict requirement that every peer in the P2PSIP overlay implements the SIP proxy and redirect server functionality.

The P2PSIP overlay may include STUN/TURN servers that do not implement a SIP stack but may assist peers/SIP UA in NAT traversal. These servers must be allowed to register to the DHT and advertise their services.

It is also possible to combine a client protocol with support for conventional SIP devices. In this scenario some of the peers would include a co-located SIP proxy server which implements a P2PSIP API that allows them to PUT/GET information to/from the overlay on behalf of unmodified SIP devices. All of the peers (or the subset of thereof) would implement the (different) client protocol to support requests.

End users and P2P operators may have an equal or even bigger interest in various other non-SIP applications compared to P2PSIP only. For this reason, the Client Protocol may be enabled with other interfaces to the DHT, such as described in [30]. Such interfaces are however beyond the scope of P2PSIP WG.

9. Security Considerations

P2PSIP security [29] can be decomposed into several parts:

- DHT security
- SIP security
- Client protocol security
- Security of the SIP-DHT interface.

It is a good idea to keep these parts as independent as possible, though as we will see, DHT security can be enhanced by strong authentication provided in the SIP application layer.

9.1. DHT Security

The specific security issues for large public P2P networks stem from the fact that mutually distrusting parties must be able to join the overlay network in spite of having possibly conflicting interests. This is also true for large closed P2P networks [30].

A fraction of the nodes may act maliciously and cause the following problems:

- Misrouted, corrupted or dropped messages,
- Corrupted routing information,
- False identities being presented to the other peers,
- Corrupting or deleting data they are supposed to store.

1. The attacker should not be able to easily corrupt, delete, or overwrite data stored in the P2PSIP overlay, the data stored in routing tables as well as user records. In order to achieve this, the node ID assignment process should make sure that a single user receives only one node ID. Once assigned, the node ID should be difficult to change. These two requirements prevent the so-called Sybil [31] attacks, where a malicious party obtains a large number of valid node IDs. Several measures can be taken by the enrollment process, for example:

rd o Strong identi

o Charging for the enrollment,
o Trust may be a useful tool in very large P2PSIP networks [32].

2. It should be possible to verify integrity of data stored in the overlay by comparing data with neighbor nodes or by checking the consistency of the routing tables.

3. It should be possible to verify integrity of data stored in the overlay by comparing data with neighbor nodes or by checking the consistency of the routing tables.

4. Signature mechanisms can be used to verify data has not been tampered with while stored.

The security mechanisms should scale from a few nodes to an overlay of many millions of nodes across the globe. Because of the nature of P2PSIP the security mechanism SHOULD NOT depend on centralized servers except for a central certificate authority such as from the P2P overlay operator. Self signed certificates may also be deployed.

Req 9-1: The security of the underlying DHT in P2P SHOULD use one or a combination of all four DHT specific security techniques:

1. Secure node IDs obtained from the enrollment server
2. Secure routing table maintenance by checking the routing tables with the neighbors
3. Secure message forwarding using various failure tests, such as comparing responses from different nodes when using different source nodes to launch a request.
4. Originator signed data.

Trust [32] MAY also be used to reduce the reliance on the above security tools.

9.2. SIP Security

The minimal requirements for SIP security for both client-server and peer-to-peer modes are detailed in [33].

9.3. Client Protocol Security

The client protocol must allow nodes that are not eligible to become peers to securely store, modify, delete and retrieve user and resource records stored in the P2PSIP overlay. This includes the following operations:

1. Authentication

2. Access rights assignment

3. Integrity protection

4. Data encryption

The client protocol must support a method for clients and peers to authenticate each other. This is important in a distributed scenario where a client may connect to different unknown peers. The client should be able to verify that the peer it is connecting to belongs to the desired overlay. On the other hand the server should also be able to authenticate the client. Some widely deployed cryptographic protocols such as (D)TLS use a certificate based authentication.

Req 9-2: The client protocol must support mutual authentication between peers and clients. The authentication method MUST be implemented by peers and clients and SHOULD be used by these entities.

The client protocol must support a method that allows setting access rights to the resources stored in a DHT. When inserting a new user or resource record into the DHT, the client should be able to allow one or more network entities to perform one or more of the following operations: modify, delete, and override the inserted record.

Req 9-3: The client protocol MUST allow setting access rights to the resources stored in a DHT.

The peer should be able to validate that the data received from a client was not altered. It should be also possible for a client to verify integrity of data retrieved from the overlay.

Req 9-4: The client protocol MUST support integrity protection of the data being inserted or retrieved to/from an overlay.

The clients should be provided with option to encrypt data exchanged with peers, and vice versa.

Req 9-5: The client protocol SHOULD support data encryption.

9.4. Security of the SIP-DHT Interface

Besides the security of the P2PSIP overlay itself that we have discussed in the sections 10.1 and 10.2, the security of the interface between SIP and the DHT is also a matter of concern.

Two main attack scenarios have been identified between the DHT and applications that use it [30]:

- Squatting: Use a valuable key ("coca-cola.com"), similar to domain names

- Drowning: Putting a vast number of values for the same key.

The security design for the interface to deal with the above can be summarized as in the following example

1. Authentication
   - Assumes the existence of a private + public crypto key pair
   - Verify that an authorized and/or trusted entity wrote a value
   - A node protects its own value from overwriting.

2. Publish security
   - The publishing node signs the key/value pair.

3. Lookup security
   - A node verifies the data returned by the Lookup operation.

4. Remove security

- A node verifies the credentials of the requestor to remove data.

Req 9-5: The interface between SIP and the DHT MUST be secured in case the P2PSIP client is connecting to peer nodes of the P2PSIP networks or when an external DHT is used:

9-5 a. Authentication credentials MUST be presented, and

9-5 b. All commands to the peer must be secured by encryption and digital signatures.

10. IANA Considerations

This document has no actions for IANA.

11. Conclusions

The requirements for P2PSIP point to a complex protocol, especially when choosing a high performing DHT layer that includes system design based on operational experience and measurements.

NAT traversal and security add to the complexity of P2PSIP.

Given however that all peers run essentially the same software and P2PSIP is a self organizing network, the operational complexity and cost is expected to be much less that conventional client-server SIP networks.

12. Acknowledgments

The authors would like to thank Kundan Singh for useful input to this document and also for reviewing parts of the draft. In addition, the many people of the IETF P2PSIP WG that have contributed to discussions or drafts were invaluable in assembling this document.

Jiang XingFeng has made valuable comments to this document.

This document was prepared using 2-Word-v2.0.template.dot.

13. References

13.1. Normative References

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

13.2. Informative References

[2] Home page for the P2PSIP WG: http://tools.ietf.org/wg/p2psip/

[3] Bryan, D., Matthews, P., Shim, E. and Willis, D: "Concepts and Terminology for Peer to Peer SIP", draft-willis-p2psip-concepts-04 (work in progress), http://www.ietf.org/internet-drafts/draft-willis-p2psip- concepts-04.txt, March 2007.

[4] D. Bryan et al: "Use Cases for Peer-to-Peer Session Initiation Protocol (P2P SIP)", I-D, IETF November 2005.

[5] David A. Bryan, Bruce B. Lowekamp, and Cullen Jennings, SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System (pdf), Proceedings of the 2005 International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA 2005), June 2005

[6] "State of P2P Communication Across NATs" by P. Srisuresh et al. Internet Draft, IETF, February 2007.

[7] "Interactive Connectivity Establishment (ICE)" by J. Rosenberg. Internet Draft (Version 15), March 2007.

[8] "NAT Classification Results using STUN" by C. Jennings. Internet Draft, IETF, October 2004 (archive).

[9] "Obtaining Relay Addresses from Simple Traversal Underneath NAT (STUN)" by J. Rosenberg. Internet Draft, IETF, March 2007, work in progress.

[10] "The Effects of NATs on the P2P Overlay Architecture" by E. Cooper and P. Matthews. Internet-Draft, IETF March 2007, work in progress.

[11] "Scribe: A large-scale and decentralized application level multicast infrastructure" by M. Castro et al. IEEE Journal on Selected Areas in Communications, October 2002. http://freepastry.org/PAST/jsac.pdf

[12] "A Survey and Comparison of End-System Overlay Multicast Solutions Suitable for Network Centric Warfare" by Cristina Abad et al., NCSA and CS University of Illinois, SPIE 2004.

[13] "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes" by E.K. Lua et al. IEEE, March 2004. http://www.cl.cam.ac.uk/teaching/2005/AdvSysTop/survey.pdf

[14] "openDHT: A Public DHT Service, Ph.D. Thesis by Sean C. Rhea, 2005 http://srhea.net/papers/rhea-thesis.pdf

[15] "Pastry: Scalable, decentralized object location and routing storage for large-scale peer-to-peer systems" by A. Rowstron and P. Druschel. Proceedings of the IFIP/ACM , Nov. 2001.

[16] "Scalable peer-to-peer substrates: A new foundation for distributed applications?" by P. Druschel and A. Rowstron. Rice University and Microsoft Research.

[17] The Chord Project, see http://pdos.csail.mit.edu/chord/

[18] "The Bamboo Distributed Hash Table" web site at http://bamboo-dht.org/

[19] "Kademlia" A Peer-to-peer Information System Based on XOR Metric" by P. Mamounkov and D. Mazieres. Rice University 2002. http://www.cs.rice.edu/Conferences/IPTPS02/109.pdf

[20] "Handling of Churn in a DHT" by S. Rhea et al. USENIX Annual Technical Conference, June 2004.

[21] "Non-Transitive Connectivity and DHTs" by M. Freedman et al. Workshop on Real Large Distributed Systems conference 2005.

[22] David A. Bryan, Marcia Zangrilli, and Bruce B. Lowekamp, Challenges of DHT Design for a Public Communications System (pdf), William and Mary Technical Report WM-CS-2006-03, June 2006.

[23] "The Impact of DHT Routing Geometry on Resilience and Proximity" by K. Gummadi et al. SIGCOMM conference 2003, Karlsruhe, Germany. http://www.sigcomm.org/sigcomm2003/papers/p381-gummadi.pdf

[24] XML-RPC, online at : http://xmlrpc.com

[25] "OpenDHT", online at: http://www.opendht.org

[26] "PlanetLab", online: http:// https://www.planet-lab.org

[27] S. Baset and H. Schulzrinne: "Peer-to-Peer Protocol (P2PP)", Internet Draft, IETF, February 2007, work in progress

[28] K. Singh and H. Schulzrinne: "Data format and interface to an external peer-to-peer network for SIP location service", I-D, IETF, May 2006, expired.

[29] "Security requirements in P2PSIP" by M. Matuszewski et al. Internet-Draft, IETF, Feb. 2007, work in progress.

[30] S. Rhea, et al: "Open DHT: A Public DHT Service and Its Uses", SIGCOMM'05, 2005. http://www.sigcomm.org/sigcomm2005/paper-RheGod.pdf

[31] Douceur, J., "The Sybil Attack", IPTPS '02, March 2002.

[32] "P2P Trust Infrastructure" by R. Nelson et al. UCLA report, 2/13/2005. http://www.cs.ucla.edu/~rlnelson/trust.pdf

[33] "Simple SIP Usage Scenario for Applications in the Endpoints" by H. Sinnreich et al. Internet-Draft, IETF, Sep. 2007, work in progress.

Author's Addresses

David A. Bryan
SIPeerior; William & Mary
3000 Easter Circle
Williamsburg, VA 23188
USA
Email: bryan@ethernot.org

Marcin Matuszewski
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland
Email: marcin.matuszewski@nokia.com

Salman A. Baset
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
Email: salman@cs.columbia.edu

Henry Sinnreich

Adobe Systems Incorporated

601 Townsend Street

San Francisco, CA 94103

USA

Email: henrys@adobe.com

Intellectual Property Statement

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 or 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.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.