Peer-to-Peer Session Initiation Protocol Standardisation Progress

Erasmus Tyapa
Email: g10t3627@campus.ru.ac.za

July 4, 2010

Abstract

Peer-to-Peer Session Initiation Protocol (P2PSIP) is ongoing effort of developing a Peer-to-Peer (P2P) version of the Session Initiation Protocol (SIP) protocol that uses any Distributed hash Table (DHT)-based P2P overlay network to locate resources. The main motivation behind P2P based SIP is to support ad-hoc communication, to simplify the configuration of SIP networks, to make SIP networks more scalable and to provide services independently of other network components such as Domain Name System (DNS). This paper is based on literature study and the P2PSIP working group drafts, in the first section we will introduce P2PSIP looking at its goals and main concepts. The second section gives an overview of super-DHT distribution model with following section describing the approaches to solve NAT drawbacks. In the fourth section will look at possible architecture that can be used to combine P2P and SIP. In the fifth section we will look at REsources LOcation And Discovery (RELOAD) with the following section describing some of the available implementations. The next two section will address the closed and open issues.

1 Introduction

In legacy Voice over IP (VoIP), a centralised call server is subject to all the shortcomings of the client-server model of communications. The server may be the single-point of failure. In actual practice, it also requires deployment, maintenance, and configuration redundancy. Furthermore, client devices designated for different Internet Protocol (IP) telephony services may not be able to interoperate with each other because of speficic customisations added to their respective servers. All these issues call for a decentralised, server-less model of communications. Relatively new DHT algorithms such as Chord and Pastry have made P2PSIP possible. P2PSIP has been a topic of discussion, debate, and development for sometime. It collapses some of the more complex server functions into the User Agents (UAs) themselves and relies on the SIP philosophy that intelligence in communications solutions should reside in the endpoint. P2PSIP origins started in academia in 2004, some of the papers published by then were [3] and [22]. For past years, P2PSIP has been explored, discussed and debated both outside and inside Internet Engineering Task Force (IETF), where a P2PSIP working group was formed in March 2005 as a follow-up on the Columbia University projects SIPpeer [21] and the SOSIMPLE [2, 3] project at William and Mary College. The initiators of P2PSIP claim higher robustness against failure as well as easier configuration and maintenance as the main motivation for P2PSIP. The P2PSIP working group is an IEFT working group [15] in charge of standardisation of the decentralised SIP architecture. In short, to incorporate the advantages of Peer-to-Peer (P2P) into the client-server based SIP and form P2PSIP protocols that is not entitled to only one organisation like Skype. P2PSIP is an open-standard's answer to Skype. P2PSIP effectively distributes the registration, location and lookup steps of SIP. It handles three functions:

1. Registering a user with the P2P overlay network.
2. Looking up a user in the P2P overlay network (when a call to a user is made).
3. Dynamically sharing information when peers join and leave, so that the load is balanced across peers, and so that the sudden loss of one or more peers does not cause the P2P network to lose track of its current registrars.

section P2PSIP IETF working group The P2PSIP working group is the group in charge of the standardisation of the decentralised SIP architecture. The group is main task is to define a P2P based VoIP communication that uses SIP. Moreover, it addresses issues such as security and privacy in a P2P communications network. In short, the manadate of the group is to incorporate the advantages of P2P into the client-server based SIP and form P2PSIP that is not tied to only one organisation like Skype. Essentially, P2PSIP is an open standard's answer to Skype.

Any person with interest in P2PSIP can join the group and follow up what is going on. Group members propose different ideas, which, once accepted and mature, are formulated in drafts. Group members can read the drafts at the working group website and the discussion on the issues about the proposed designs are carried via a mailing list.

1.1 Goals

The main goal of the P2PSIP working group is to eliminate centralised servers currently deployed in SIP architecture. Basically, this means developing a platform that allows for a decentralised approach to deploy voice and multimedia services. A good example is a Skype-like system that uses non-proprietary protocols but is capable of working in any conditions and has good scalability.

The charter of the IETF P2PSIP working group [25] outlines the following as primary tasks of the group:

- Producing an overview document explaining concepts, terminology, rationale, and providing use cases for the remaining work.
- Writing a proposed standard for the P2PSIP peer protocol.
- Writing a proposed standard for the P2PSP client Protocol, the protocol used between a P2PSIP peer and a P2PSIP client. (The terms peer and client will be explained later).
- Writing statements that will address how the previously defined protocols, along with existing IETF protocols can be used to produce systems to locate a user,

identify appropriate resources to facilitate communication (such as media relays), and establish communications between the users, without relying on centralised servers.

We will now look at some of the concepts produced by the P2PSIP working group

1.2 Main Concepts

The main concepts used in P2PSIP are defined in the, so called, concepts draft [4]. This section describes the most important concepts to the work done in this research.

- P2PSIP: It is the set of protocols that extends SIP for P2P. It only includes two protocols: the P2PSIP Peer Protocol used between P2PSIP Peers, and the P2PSIP Client Protocol used between a P2PSIP Client and a P2PSIP Peer.

- The P2PSIP Overlay: This refers to a network of nodes that participates in data distribution and provides SIP registration, SIP request routing, and other services.

- A P2PSIP Peer: It is a node participating in a P2PSIP overlay that provides storage and routing services to other nodes in the same P2PSIP overlay. A P2PSIP peer can be located behind NATs and still be fully functional. It can perform several operations like joining and leaving the overlay, routing requests within the overlay, storing information, inserting information into the overlay and retrieving information from the overlay.

- A P2PSIP Client: It is a node participating in a P2PSIP overlay that does not store resources, run the distributed database algorithm, and is not involved in routing messages to other peers or clients. A P2PSIP client is like a simpler peer. A client insert, modify, examine, and remove records by interacting with a peer of that same overlay.

- P2PSIP Peer Enrollment: This refers to the process a P2PSIP peer follows to get an identifier and credentials for a given P2PSIP overlay. The process is done outside the P2PSIP overlay and is only needed at regular intervals or when the P2PSIP peer looses its identifier or its credentials.

2 Distribution Model in P2PSIP

The ultimate aim of P2PSIP is to get rid of centralised proxies and registrars in SIP and distribute their functionalities among the participating nodes. There are different ways of distributing these functionalities depending on which kind of the nodes participate in the distribution.

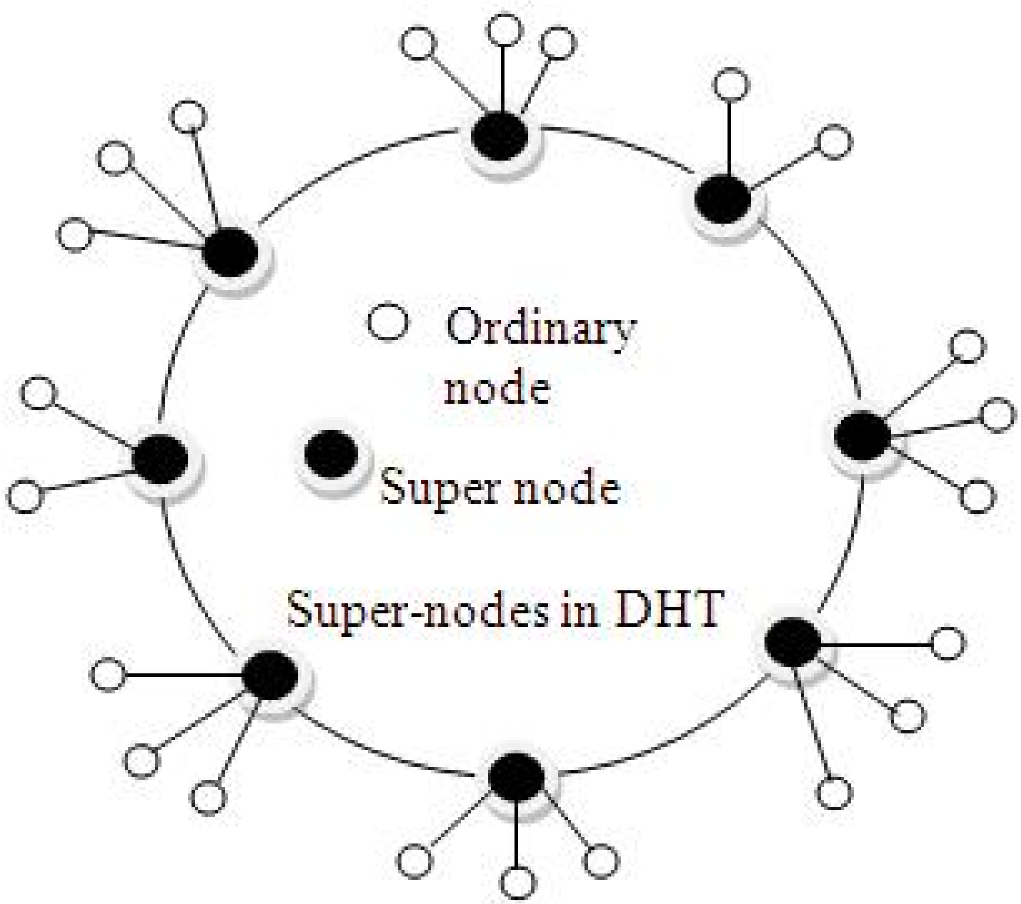

Figure 1: Super-nodes in DHT distribution model (Adapted from [20])

Figure 1 shows a distribution model that has been chosen by the working group as a distribution model to be used in P2PSIP. The model in Figure 1 is the P2P overlay of nodes but not all nodes have equal capacity and availability. Some of the nodes have high capability (bandwidth, Central Processing Unit (CPU), memory) and availability (uptime, public IP address) [20] than the other nodes. These nodes are called super-nodes and the other nodes with fewer capabilities are called ordinary nodes. The super-node performs the duties of the SIP registrar and proxy, it maintains the location information for the joining node, and locates other users by communicating with other super- nodes.The working group claim that the connection to a P2PSIP peer behind a Network Address Translation (NAT) is the main motivation for using the super-node distribution model. Singh[20] gives an example that support the statement, that a node with low bandwidth connection to the Internet or those behind a firewall or NAT may not be able to fully function in a DHT because it may need in-bound connections, significant bandwidth for forwarding P2P messages or significant memory or CPU to maintain DHT State. The super-node and ordinary node distinction is when the node has joined the DHT. So basically a P2PSIP node enrolls in the P2PSIP system, and then acts in either client or peer mode depending on the node capability and availability mentioned above. This can happen at the startup or when the node detects that it has enough capacity and capability. Skype , is belived to use the same model as in Figure 1.

The P2PSIP overlay is required to interwork with conventional SIP networks. The super node in the DHT model solves the problem of inter-domain connectivity by letting each domain have at least one super node. These super-nodes connect each other to form upper layer overlay, which provides help when communication is needed between peers in different P2PSIP domain.

The difference in capabilities between super-nodes and ordinary nodes is very similar to the way in which the P2PSIP charter introduces a P2PSIP peer and P2PSIP client. The idea of this model is to select weaker and unstable peers for the lower layer, hence making the system more scalable, and guarantees the peer or resources lookup in the higher overlay. P2PSIP message flow in the overlay network should comply with a few routing styles. In the next section we will be looking at different ways of solving NAT traversal which is major drawback in the progress of P2PSIP standardisation.

3 NAT Drawbacks

The P2PSIP overlay must be able to function and provide services, even when some of the peers are behind NATs. Therefore a selection of routing algorithms that allow peers to participate in the P2PSIP overlay even when non-transitive connectivity exists because of NATs. One approach the working group can use to solve the problem of NAT and security issue is Host Identity Protocol (HIP), For more on this see draft [5].

3.1 Session Traversal Utilities for NAT (STUN) approach

This approach define a new network element called STUN server, used in the middle of two endpoints to learn the NAT status (e.g., existence of NAT, NAT type, public endpoints address, port, etc.). With the above mentioned information, two endpoints might be able to establish the session directly. STUN [19] work for certain types of NATs or if only Transmission Connection Protocol (TCP) is allowed.

3.2 Traversal Using Relay NAT (TURN) approach

TURN [13] solves the problems associated with STUN, it work in both TCP and User Datagram Protocol (UDP). TURN define a new network element called TURN server that relay data traffic during the connection and transmission. Interactive

Connectivity Establishment (ICE) approach: This approach uses the combination of TURN and STURN approach. ICE [18] firstly select STUN for handling, while turns to TURN if STUN is not available. Besides, ICE supports the negotiation of session establishment (e.g. latency, jitter measurement, error handling, best route, etc.) between end-points. This approach is the one which the working group is using to solve the NAT drawbacks.

3.3 Interactive Connectivity Establishment (ICE) approach

This approach uses the combination of TURN and STURN approach. ICE firstly select STUN for handling, while turns to TURN if STUN is not available. Besides, ICE supports the negotiation of session establishment (e.g. latency, jitter measurement, error handling, best route, etc.) between end-points. This approach is the one which the working group is using to solve the NAT drawbacks.

4 Protocol Layering for P2PSIP

The debate on which model for protocol layering is the most appropriate for P2PSIP has been debated in the P2PSIP working group and different proposals were brought forward to support the notion. The P2PSIP working group has chosen one of the two architecture of combining SIP and P2P, proposed by the Singh [20] for P2PSIP telephony. The two architectures are: P2P-over-SIP, SIP-using-P2P. The two architectures will be described next.

4.1 P2P-over-SIP

In P2P-over-SIP architecture, SIP messages are not used only for registering users, resource lookup, and establishing session, but also for maintaining a P2P network. Although SIP was not designed with P2P in mind, the design is extensible. To cater for P2P traffic SIPPEER [21] and dSIP[9] have proposed extensions of SIP. In this proposal, SIP REGISTER requests are used to join, build, and pass information between peers. However, tunnelling all P2P messages over SIP REGISTER causes high overhead. P2P-over-SIP architecture is not flexible as it is not interoperable with any other P2P applications without requiring them to implement a SIP stack. Because of the many drawbacks mentioned, the P2P-over-SIP has lost its popularity and was not chosen.

4.2 SIP-using-P2P

The problem of overloading SIP with further functionality it was not designed to do e.g., maintaining a P2P/DHT network, can be solved by SIP-using-P2P. SIP-using-P2P architecture uses two separate stacks: a SIP layer for registering users, resource lookup, establishing session and a P2P layer for maintaining a distributed network. Therefore, it diminishes the application overhead and complexity. The SIP stack and P2P stack in P2PSIP applications can be implemented on the same or different nodes. In other words, P2PSIP application can implement DHT on its own, or deploys an external DHT service. Singh and Schulzrinne [22] propose using OpenDHT [8] as a SIP location service. They use a partial P2P architecture where OpenDHT node acts as a server offering a storage service to other clients who does not support P2P functions. As shown in Figure 2(a), DHT layer and SIP layer are clearly separated in SIP-using-P2P. In SIP-using-P2P the P2P wire protocol is independent of SIP, and the SIP entities just use the services provided by P2P layer, e.g., storage, lookup and routing. This made SIP-using-P2P as the appropriate choice by the working group. There are many proposed designs that combine SIP and P2P using SIP-using-P2P, such as RELOAD which will be described in the next section.

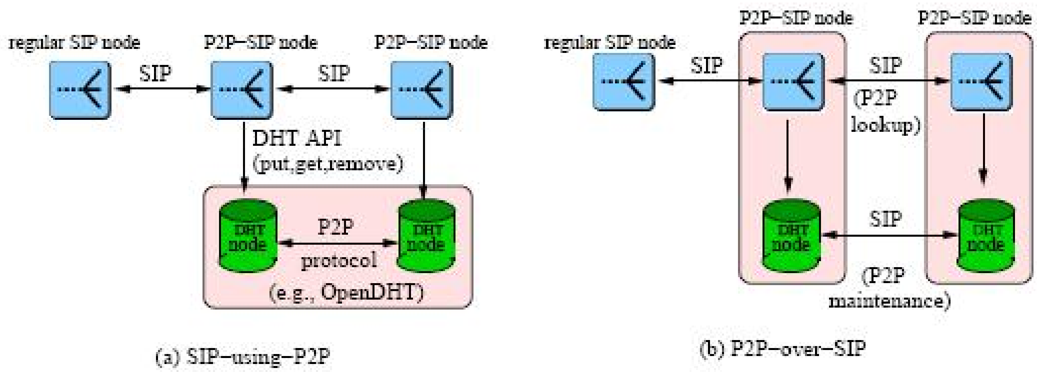

Figure 2: Difference between SIP-using-P2P and P2P-over-SIP architectures (From [20])

5 REsources LOcation And Discovery

In 2007, there were many competing proposals for the P2PSIP peer protocol; RELOAD, Address Settlement by peer-to-peer (ASP) [11], Service Extensible P2P Peer Protocol SEP) [12], Extensible Peer Protocol (XPP) [14] and Host Identity Protocol (HIPHOP) [6]. In February 2008, Peer-to-Peer Protocol (P2PP) [1] was merged to the combined RELOAD/ASP protocol.

RELOAD [10] was adopted by the P2PSIP working group as its starting point for the primary P2PSIP protocol. RELOAD can be used for other P2P applications since it has two separate stacks, SIP stack and P2P stack. RELOAD has been designed with an abstract interface to the overlay layer to simplify implementing a variety of structured (DHT) and unstructured overlay algorithms. This promotes interoperability and selection of overlay algorithms optimised for a particular application.

5.1 RELOAD Architecture

Figure 3 shows the basic architecture of RELOAD. Each application that wishes to use RELOAD defines a RELOAD usage; examples include a SIP Usage, Extensible Messaging and Presence Protocol (XMPP) Usage or any other. These usages all talk to RELOAD through a common Message Transport API. The applications can use

RELOAD to store and retrieve data, as a service discovery tool or to form direct connections in P2P environments. Currently defined usages are the SIP usage, the certificate store usage, the Traversal Using Relays around NAT (TURN) [13] server usage and the diagnostics usage. Transport layer provides a generic message routing

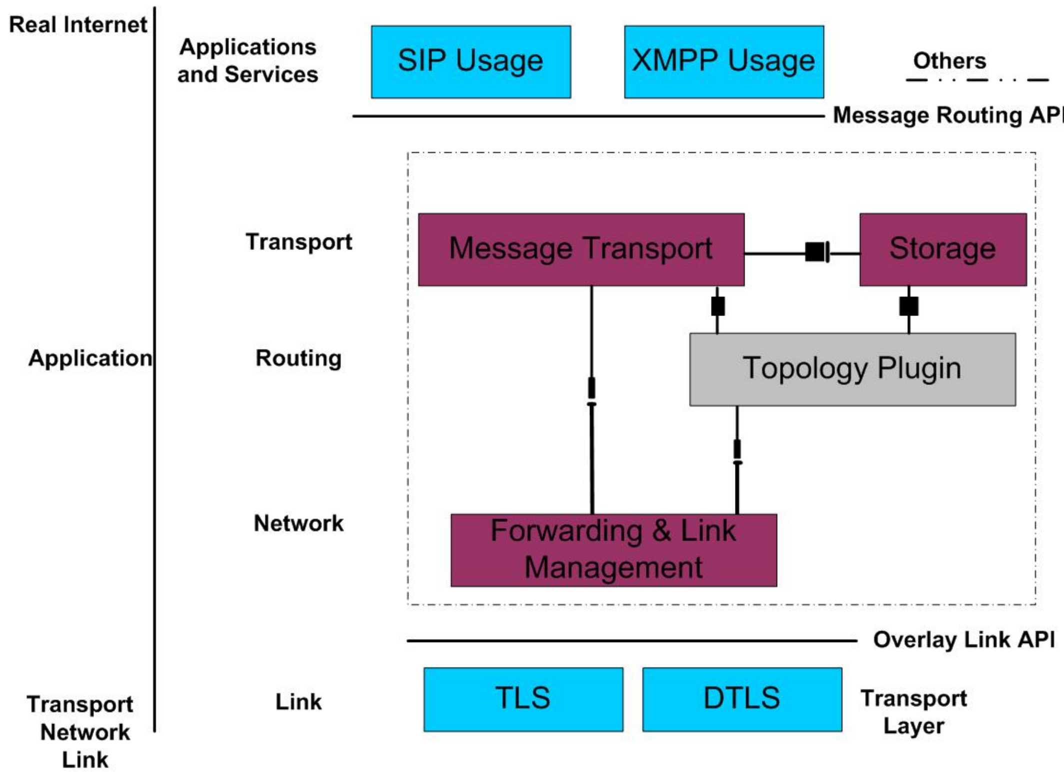

Figure 3: Major Components of RELOAD (From [10])

service for the overlay that is sending and receiving messages from peers. The storage component is responsible for processing messages relating to storage and retrieval of data. RELOAD is specifically designed to work with a variety of overlay algorithms. However, for interoperability reasons, RELOAD defines one algorithm, Chord, that is mandatory to implement. The topology plugin defines the content of the messages that will be used in RELOAD, the various procedures to join and leave an overlay, the hash algorithm used (the default algorithm used is Secure Hash Algorithm 1 (SHA-1)), the replication strategy and the routing procedures. Forwarding and link management layer provide packet forwarding services and help setting up connections across NATs using Interactive Connectivity Establishment (ICE) [18]. Currently Transport Layer Security (TLS) over TCP and Datagram Transport Layer Security (DTLS) over UDP are the link layer protocols supported by RELOAD for hop-by-hop communication.

5.2 RELOAD Network

The RELOAD Network is very similar to any other P2P network. However, one of the differences is that not all members of the network are peers, in fact there are also clients. A node might act as a client simply because it does not have the resources defined in [10] or does not implement the topology plug in required to act as a peer in the overlay. Still, a client uses the same RELOAD protocol as the peers, knows how to calculate Resource-IDs and signs its requests in the same manner as peers. RELOAD is going to have a credential and an enrollment server too. The routing mode in RELOAD is symmetric recursive, which is similar to recursive routing. Iterative routing is not possible since a message may need to be sent to a peer that is not present in the routing table, which requires a new direct connection to be established, making latency too high for the communication to be efficient. The pure recursive routing cannot be applied either for similar reasons; if a node behind a NAT receives a message response that has not been previously requested, the NAT will drop the message, making the communication impossible.

6 P2PSIP Implementation

P2PSIP is new technology that just emerged few years back, hence there are few P2PSIP open source project implemented so far but most of the available P2PSIP system are proprietary make them diffcult for academic research purpose. We are going to discuss few of the P2PSIP open system, SIPDHT,SIP2P, and a P2P framework that was implemented at Rhodes University.

6.1 SIPDHT

SIPDHT [7, 24, 26] is one of the few P2PSIP open source implemented project so far. It was started by an active member of P2PSIP IETF working group, Enrico Marocco from Telecom Italia and others in May 2006. There are two version of SIPDHT, the first version followed Chord algorithm when implementing its DHT. However, the second version used a different algorithm called Passive Content Addressable Network (pCAN) [17] as a DHT algorithm, a modified version of CAN. The pCAN is passive in nature because the client does not participate actively in the overlay. They only participate when they are invited by an existing peer in the overlay. The peers in the overlay take the job of SIP proxies while pCAN functionality is added to the SIP clients. Standard SIP clients can connect to the network by configuring any of the SIPDHT nodes as a server, but will not carry any part of the pCAN. The peers in overlay are used to locate nodes on which a user is located by forwarding the requests to the appropriate peers. The overlay is maintained using the lightweight binary protocol using pCAN as its underlying DHT. pCAN provides functionalities for setting up sessions, exchanging information with the peers assisting new clients to join the overlay and performing all tear downs.

Marocco cited the following as the reason of choosing CAN over other DHTs.

- CAN is symmetric, which allows the connection to be accessed equally by both connecting peers.
- In CAN the peers maintain a stable routing table with limited number of entries.

Pundkar [16] cited some drawbacks with CAN, when using the CAN algorithm performances is relatively low. Further, while other popular algorithms claim to always be logarithmic in complexity, overlays based on CAN cannot scale indefinitely. CAN based overlays need to be configured during deployment with parameters depending on the size of the intended network. To achieve acceptable performance, it has been argued in [16] that only a subset of interested nodes in the SIPDHT P2P overlay should provide service.

6.2 SIP2P

SIP2P [23] is an open-source project published at Sourceforge.net that was initiated by WU Wien (University of Economics). It provides experimental peer-to-peer SIP implementation in the form of a small application that makes use of a Kademlia library and allows placing calls between SIP clients. It is written in C++ and currently supports the REGISTER and INVITE methods, which are directly translated into a *put/get* operation on the Kademlia network.

6.3 OverCord

The eagerness in the P2PSIP has led to many people wanting to experiment with P2PSIP protocol. By the time the work of P2PSIP was still new in the IETF, a project was being carried out at Rhodes University. The OverCord framework, a P2P framework was developed as a part of master thesis at Rhodes University. The OverCord framework is based on a SIP-using-P2P. The OverCord framework separates the SIP and P2P. Figure 4 show the architecture of the architecture of the OverCord.

At the top most layer of the OverCord framework is the SIP applications and other distributed services other than telephony. The middle layer is the P2P layer that consists of resource database, discovery, plug-in management, overlay plug-in and overlay repository layers. The OverCord framework again demonstrates the importance of combining P2P and SIP using SIP using-P2P which the group has adopted as the best approach.

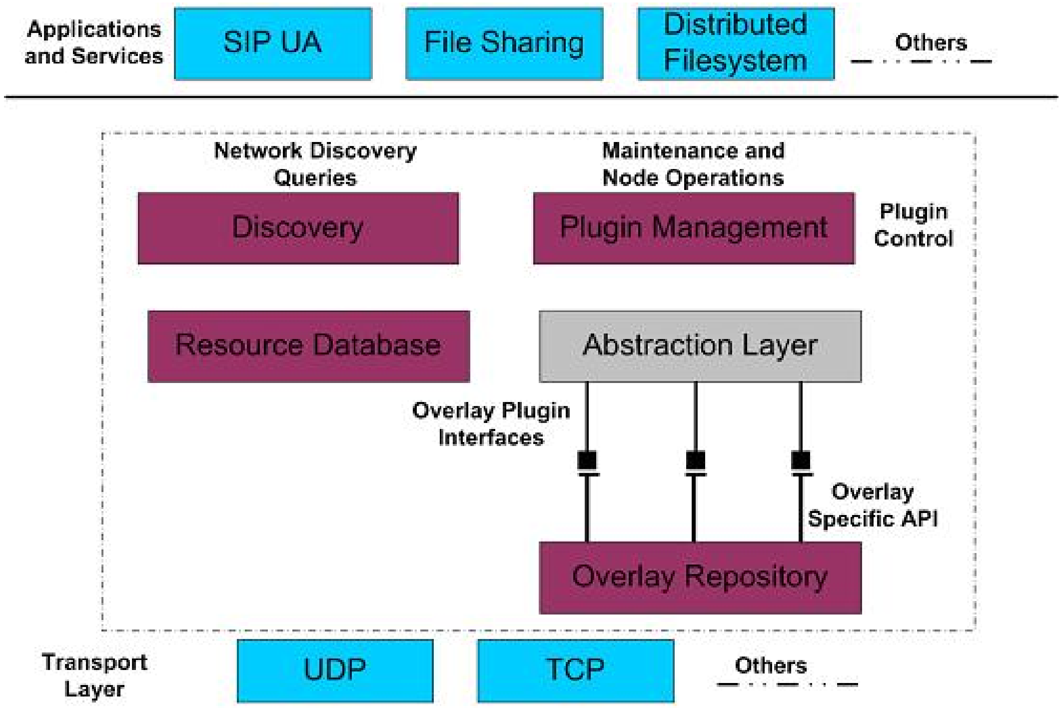

Figure 4: The layered architecture of the OverCord framework (From [26])

7 Closed issues

There was a debate in the working group whether the P2PSIP peer protocol is necessary at all and why clients and peers cannot use the same protocol but the group later decided that the P2PSIP client protocol rather be an optional. Most of protocol proposal presented support the notion of both clients and peers using the same protocol.

The P2PSIP working group has agreed that the distribution model to be used in the P2PSIP is the super-node in DHT model; hence the client and peer distinction is only in when the node has joined the DHT. So basically a P2PSIP node enrolls in the P2PSIP system, and then act in either client or peer role depending on the node capability and availability mentioned above. Furthermore, this answers the questions of whether there is a need of separate credentials for peers and user. Another model that has been accepted by the working group is the fully P2P model.

The selection of the DHT algorithm is left to the developer, has the P2PSIP protocol must support multiple DHTs. The P2PSIP peer protocol should be extensible to accommodate different overlay technologies such as pastory, kademlia, chord and others, including future algorithms that may appear. Even though this debate is closed, there are still some things to be done by the working group the selection mandatory of the DHT algorithm and other issues not related to DHT , see next section for some of those issues.

8 Open issues

P2PSIP has gone a long way but there still some issues to sort out. There are issues that are still being discussed by the working group. There is a debate on whether P2PSIP RELOAD is suitable to be leverage for both P2P live streaming and Voice on Demand (VoD) services. Apparently the evaluation has proved that DHT or RELOAD is not suitable for the chunk dicovery in P2P streaming especially the live media streaming. Even though the working group agreed that the choice of the DHT algorithm is left to the implementer, currently, Chord is the only standardised DHT algorithm in the peer protocol and Chord is not as good as other DHT algorithms in some ways.Therefore, the P2PSIP working group need to standardise more DHT algorithms. Another issues being debated, is the issues of security and NAT traversals, where solutions are currently proposed in a draft [5], this draft is under Host Identity Protocol (HIP) working group. The only question is whether the P2PSIP working group need to or will it have time to adapt HIP in future as HIP has not been deployed widely.

9 Conclusion

The P2P networking architecture attraction has led to its adoption for SIP, which led to the formation of the P2PSIP working group in the IETF. Even though, this paper summarised some of the work that has been achieved so far, work on P2PSIP is still going on. Some of the works are put on hold, such as P2PSIP client protocol. This paper concentrated more on answering the questions that has been asked to bring up the requiremenst for P2PSIP protocols. We have discussed some of the major break-throughs in the working group such as the distribution model and RELOAD. The work in the P2PSIP working group is still going on especial on the aspects of security and NAT drawbacks but we expect the P2PSIP protocols to come out soon.

References

[1] S. Baset and H. Schulzrinne. Peer-to-peer protocol (p2pp). Internet-Draft (work in progress), draft-baset-p2psip-p2pp-01, Available online, URL: http://tools.ietf.org/html/draft-baset-p2psip-p2pp-01, November 2007. Accessed on: 14 June 2010.

[2] D. Bryan and B. Lowekamp. Sosimple: A sip/simple based voip and im system, Novemebr 2004.

[3] D. Bryan and B. Lowekamp. Sosimple: A serverless, standards-based. In *P2P SIP Communication System, Proc. of the International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications*, pages 42–49. IEEE Press, 2005.

[4] D. Bryan, P. Matthews, E. Shim, D. Willis, and S. Dawkins. Concepts and terminology for peer to peer sip. Internet-Draft (work in progress), draft-ietf-p2psip-concepts-02, Available online, URL: http://tools.ietf.org/html/draft-ietf-p2psip-concepts-02, July 2008. Accessed on: 04 April 2010.

[5] G. Camarillo, P. Nikander, J. Hautakorpi, A. Keranen, and A. Johnston. Hip bone: Host identity protocol (hip)based overlay networking environment. Internet-Draft (work in progress), draft-ietf-hip-bone-06.txt. Available online, URL: http://tools.ietf.org/html/draft-ietf-hip-bone-06, April 2010. Accessed on: 03 June 2010.

[6] E. Cooper, A. Johnston, and P. Matthew. A distributed transport function in p2psip using hip for multi-hop overlay routing. Internet-Draft (work in progress), draft-matthews-p2psip-hip-hop-00. Available Online, URL: http://tools.ietf.org/search/draft-matthews-p2psip-hip-hop-00, June 2007. Accessed on: 03 June 2010.

[7] A. Fandrianto. Sipdht: An open source project for p2psip, cisco systems,. Implementation report, Cisco Systems, April 2007. Available online; URL: http://sipdht.sourceforge.net/sipdht2/report-fandrianto-sipdht2.pdf, Accessed on: 21 March 2010.

[8] OpenDHT Homepage. Available online, URL: http://opendht.org/. Accessed on: 14 March 2010.

[9] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne. A p2p approach to sip registration and resource location. Internet-Draft (work in progress), draft-bryan-p2psip-dsip-00. Available online, URL: http://tools.ietf.org/html/draft-bryan-p2psip-dsip-00, February 2007. Accessed on: 03 June 2010.

[10] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne. Resource location and discovery (reload) base protocol. Internet-Draft (work in progress), draft-ietf-p2psip-base-08. Available online, URL: http://tools.ietf.org/html/draft-ietf-p2psip-base-08, March 2010. Accessed on: 14 June 2010.

[11] C. Jennings, J. Rosenberg, and E. Rescorla. Address settlement by peer to peer. Internet Draft (work in progress), draft-jennings-p2psip-asp-0. Available online, URL: http://www.p2psip.org/drafts/draft-jennings-p2psip-asp-00.txt, July 2007. Accessed on: 14 June 2010.

[12] XingFeng. Jiang, HeWen. Zhenga, C. Macian, and V. Pascual. Service extensible p2p peer protocol. Internet-Draft (work in progress), draft-jiang-p2psip-sep-01. Available online, URL: http://tools.ietf.org/html/draft-jiang-p2psip-sep-01, February 2008. Accessed on: 14 June 2010.

[13] R. Mahy, P. Matthews, and J. Rosenberg. Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun), April 2010.

[14] E. Marocco and E. Ivov. Extensible peer protocol (xpp). Internet-Draft (work in progress), draft-marocco-p2psip-xpp-01. Available Online, URL: http://tools.ietf.org/html/draft-marocco-p2psip-xpp-01, November 2007. Accessed on: 03 June 2010.

[15] P2psip Status Pages. Peer-to-peer session initiation protocol (active wg). Available online, URL: http://tools.ietf.org/wg/p2psip/. Accessed on: 21 October 2010.

[16] S. Pundkar. Peer to peer communication over the internet: An open approach. Technical report, Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, INDIA, Developed at Telecom Italia, Turin, Italy, May-July 2007. Available online, URL: http://sipdht.sourceforge.net/sipdht2/swapnil-report.pdf, Accessed on: 14 June 2010.

[17] S. Ratnasamy, P. Francis, M. Handley, Richard Karp, and S. Shenker. A scalable content-addressable network. In *IN PROC. ACM SIGCOMM 2001*, pages 161–172, 2001.

[18] J. Rosenberg. Indicating support for interactive connectivity establishment (ice)in the session initiation protocol (sip), April 2010.

[19] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing. Session traversal utilities for nat (stun), October 2008.

[20] K. Sighn. *Reliable, scalable and Interoperable Internet Telephony.* Phd project, Columbia University, Department of Computer Science, New york, 2006.

[21] K. Singh and H. Schulzrinne. Sippeer: A session initiation protocol(sip)-based peer-to-peer internet telephony client adaptor. Implementation report, Columbia University, Department of Computer Science, Columbia University, April 2004. This is a full TECHREPORT entry, Available online, URL: http://kundansingh.com/papers/sip-p2p-design.pdf, Accessed on: 17 September 2010.

[22] K. Singh and H. Schulzrinne. Peer-to-peer internet telephony using sip. In *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pages 63–68, New York, NY, USA, 2005. ACM.

[23] a P2P SIP Proxy sip2p. Available online, URL: http://sourceforge.net/projects/sip2p/. Accessed on: 14 October 2010.

[24] SIPDHT. Available online, URL: http://sipdht.sourceforge.net/. Accessed on: 21 August 2010.

[25] Peer to Peer Session Initiation Protocol (p2psip). Available online, URL: http://datatracker.ietf.org/wg/p2psip/charter/. Accessed on: 21 March 2010.

[26] M. Tsietsi. Prototyping a peer-to-peer session initiation protocol user agent. Master's project, Rhodes University, Department of Computer Science, Grahamstown, South Africa, March 2008.

This Document was original written on 4 July 2010, and was later modified on the 17 October 2010.