

# Bridging the gap between peer-to-peer and conventional SIP networks

Mosiuoa Tsietsi, Alfredo Terzoli, George Wells  
 Department of Computer Science  
 Grahamstown, South Africa  
 Tel: +27 46 603 8291  
 hezekiah@rucus.ru.ac.za  
 {A.Terzoli, G.Wells}@ru.ac.za

**Abstract**—The Peer-to-Peer SIP (P2P SIP) protocol, which is still being standardised in the IETF, holds much promise for enabling many diverse forms of realtime multimedia communications in completely, or partially, decentralised environments. The final specification will likely feature a distributed storage API for manipulating records belonging to the SIP overlay network, as well as a multi-protocol routing layer. Though great pains have been taken to add support for different types of peer-to-peer platforms in order to realise a truly protocol-agnostic architecture, significantly less effort has been invested in resolving the issue of how these networks will interoperate, if at all, with conventional SIP networks. This paper describes the precise nature of the problem of interoperation among these different types of networks, and presents a solution which is based on dynamic updates in DNS. The solution is described within the context of the P2P SIP architecture called OverCord, and describes how this architecture was extended to support this type of interoperation.<sup>1</sup>

**Index Terms**—Internet telephony, peer to peer, overlays, DHTs

## I. INTRODUCTION

P2P SIP is a term which is used to describe the protocols and mechanisms that can be used to provide realtime multimedia communication services based on SIP, where the traditional services such as routing and storage are performed by a collection of intelligent endpoints [1]. One benefit of such an

architecture is that, since the network is decentralised, single points of failure are avoided which ultimately improves the availability of the network. Secondly, peer-to-peer architectures such as the one described in this paper are topical for modern wireless networks which overlay traditional centralised structures.

The P2P SIP working group in the IETF - which has been tasked with developing these protocols and mechanisms - is nearing the stage where a draft proposal can be adopted as the group's official solution to the problem. Currently, the main scope encompasses issues such as overlay management (locating and joining overlays), differentiating between different roles nodes can play (clients and peers), the provisioning of services (such as offline voicemail and NAT traversal) and a distributed database function to store overlay resources.

While these are crucial issues deserving much attention, one critical feature which has explicitly been deemed out of scope by the working group, at least for the moment, are techniques for interoperating decentralised overlays with conventional ones. It is evident that the key to providing this form of interoperation is to be able to reliably calculate the next hop for routing messages from one type of network to the other. A particularly challenging feature of peer-to-peer systems is that, due to high rates of churn, the next hop mapping may change frequently. Consequently, there is a need for a mechanism flexible enough to provide correct mappings in a constantly changing environment. The use of dynamic DNS updates has been suggested in the P2P SIP working group charter [1] as a possible tool to use for achieving this, and after some investigation, we concluded that it seemed to have the properties

<sup>1</sup>This work was undertaken in the Distributed Multimedia Center of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Amatole Telecommunication Services, Tellabs, THRIP, Mars Technologies, Stortech, Comverse and the National Research Foundation

needed. However, there was a lack of detail on how dynamic DNS could be used in these kinds of scenarios, and as such we set about investigating this opportunity.

This paper is a report on our experiences in utilising dynamic DNS as a supporting technique for providing interoperation within the context of our own P2P SIP architecture called OverCord. The purpose of this investigation was purely to develop a proof of concept system where only at a later stage, once the concept had been validated with an implementation, performance testing such as signaling overhead could be measured.

In this paper, we begin with a discussion on the currently existing proposals for solving the interoperation problem, which is the subject of section II. Section III briefly describes the OverCord architecture and section IV explains how OverCord was extended with dynamic DNS support. Section V describes how the reworked OverCord node behaves, and section VI provides results of the extension.

## II. BACKGROUND

The Internet draft written by Shim *et al* [2] was one of the first to investigate the question of how nodes in decentralised SIP overlays would interact with those in client-server overlays. Figure 1 shows a hypothetical SIP overlay network composed of both peer-to-peer and client-server entities. When user agent 2 (which embeds a peer-to-peer node) joins the network, it inserts data (such as location records) into the distributed database which is provided by the peer-to-peer cloud. Similarly, when user agent 1 (a client-server entity) joins, the network's SIP registrar handles the client's registration, and inserts a location record for this client into a central database. The SIP registrar, however, embeds a peer-to-peer node, and is able to use the peer-to-peer distributed database methods to subsequently insert a corresponding location record for the client into the peer-to-peer cloud.

It is easy to support communication from the peer-to-peer node to the client-server node. In this case, user agent 2 simply obtains the location record that was created by the SIP registrar from the distributed database, and uses this to appropriately create and send a SIP INVITE message. In the opposite case, where the requesting entity is the client-server node, Shim *et al* introduce a SIP proxy with peer-to-peer abilities. Its responsibility is to examine the

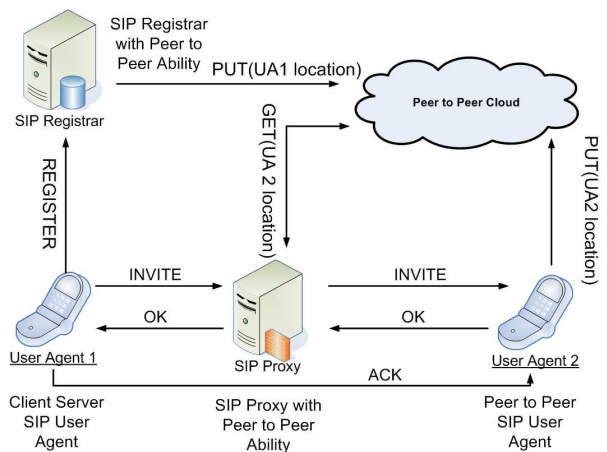


Fig. 1. A solution for interoperation. Source: [2]

incoming request from client-server nodes, and use the peer-to-peer data methods to perform a lookup in the distributed database for the location of user agent 2. The proxy then forwards the request to the appropriate destination and the SIP session can be established.

While the design presented in this draft was novel, it was also problematic. Firstly, while it answers the question as to how heterogeneous devices can co-exist in a common SIP overlay network, it does not answer the question as to how interoperation can be approached when two or more physically remote domains are involved. Secondly, the design relies very heavily on the existence of modified SIP servers such as registrars and proxies. Enforcing this requirement could be difficult since most currently existing SIP servers do not embed peer-to-peer logic.

A second solution is provided in the draft by Marocco and Bryan [3]. This draft specifies two main entities, a P2P SIP proxy peer and a relay agent peer, which are both members of a peer-to-peer overlay. The overlay name is assumed to be a fully qualified domain name, where the P2P SIP proxy has a binding in DNS. In Figure 2 the P2P SIP proxy peer performs the proxy function for users in the peer-to-peer overlay, and both it and other proxies on the Internet can use the mechanisms for locating SIP servers described in RFC 3263 [4] to locate each other and pass messages to the users they are serving. The relay agent peer uses protocols such as TURN (Traversal Using Relay NAT) [5] to provide relay transport addresses to allow media flow between user

agents behind NATs.

This design does not suffer from the shortcomings of the one proposed by Shim *et al*, since it works with unmodified SIP servers and can be used to support communication between remote domains. The problematic aspect of the design, however, is that it relies substantially on DNS, which assumes quasi-static relationships between domain names and the entities they map to. This assumption is at odds with what is typical in peer-to-peer systems where the uptime of nodes is very short, ranging from an hour to as little as a minute [6]. Ideally, for the network to be truly decentralised, the identity of the proxy peer would not be fixed, but at different times, different nodes would either be elected by the overlay or volunteer themselves to perform the proxy function. Thus the mapping between the serving proxy peer and its IP address would change frequently.

From these two approaches, a number of basic requirements can be derived for performing interoperation. Firstly, a design is needed that can work with unmodified SIP entities. Secondly, it is evident that the method must interact with DNS, as conventional systems make extensive use of DNS in order to calculate the next hop destination. Thirdly, a connection point is needed into the peer-to-peer overlay in the form of a proxy peer. Lastly, the proxy peer should rather be a role and not the identity of a dedicated, fixed node. The next two sections will describe an architecture that allows a node to fulfill the proxy peer role, and explain how dynamic DNS was used in order to support communication in the presence of transient proxy peers.

### III. OVERCORD - A P2P SIP NODE FRAMEWORK

OverCord is an architecture for decentralised object storage and location that can be used to develop P2P SIP user agents [7]. Its architecture is depicted in Figure 3. From the figure, it can be observed that OverCord consists of several discrete modules. The discovery layer consists of techniques used to locate overlays. The resource database is a node cache that contains overlay related data items in a key-value format. The overlay repository consists of different overlay implementations. An abstraction layer is used to hide the underlying overlay implementation in use, by defining a generic interface which is implemented by middleware objects we call DHT plug-ins. These

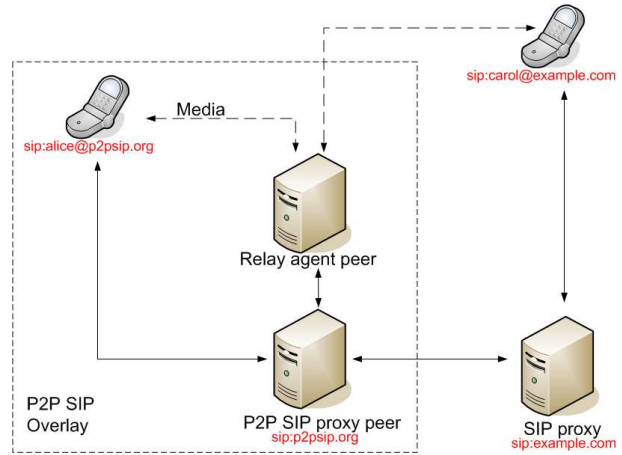


Fig. 2. A solution for interoperation. Source: [3]

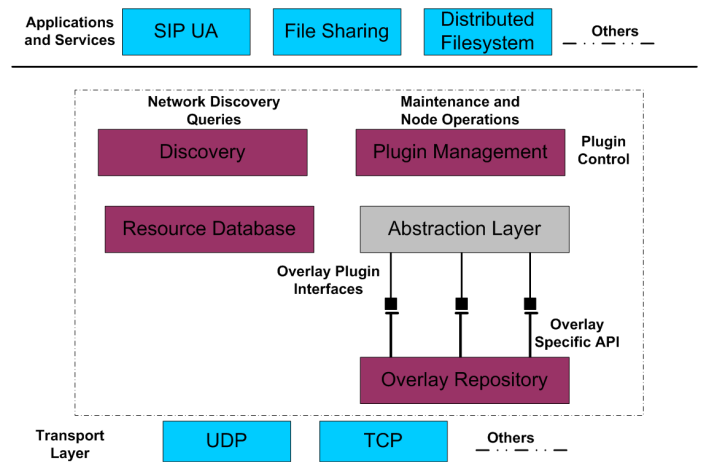


Fig. 3. The OverCord node architecture.

plug-ins assist in the translation between the single generic API and the overlay-specific APIs of the individual overlay implementations. It has been proved that interoperation between heterogeneous peer-to-peer overlays can be achieved [8] using these plug-ins. The plug-in management layer interacts directly with the higher level application, and is responsible for the detection, verification, creation and control of resident plug-ins. OverCord is well suited for P2P SIP as it can be embedded into currently existing SIP user agents to add peer-to-peer logic for the creation and maintenance of decentralised overlays and to support many forms of realtime multimedia communications [9].

Figure 4 shows how a SIP session is established between OverCord nodes. User A performs a recursive search in the local overlay using peer-to-peer

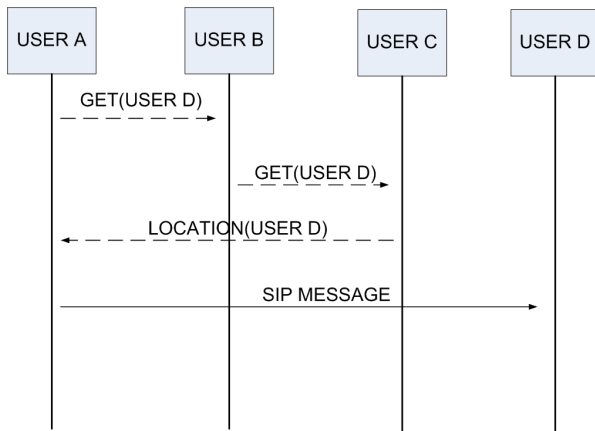


Fig. 4. Messaging sequence in OverCord.

methods in an attempt to retrieve the location record for User D. Once obtained, this information will be used to transmit the SIP message accordingly. This approach only works for users in a common peer-to-peer overlay, since the search space can only contain local resources if the peer-to-peer algorithms are to function properly. There is therefore a need for OverCord to be extended to be able to obtain the location records of users in centralised domains, which are typically stored in DNS, and vice-versa.

#### IV. EXTENDING THE OVERCORD ARCHITECTURE

##### A. Phase 1 - Importing a Dynamic DNS Client

A logical approach to solving the problem is to import a currently existing dynamic DNS software client into the OverCord architecture. One such client is the Perl-based `ddclient`. This client comes with a configuration script which can be customised according to user's needs by assigning values to several parameters, such as the name of the dynamic DNS provider, the dynamic DNS domain name, the updated IP address and the user's credentials. As a discovery and lookup tool, the client fits in the discovery layer in OverCord.

In order to use this client, we setup an account with DynDNS.com which provides free DNS services for both static and dynamic IP addresses. A domain name was created called `chord.dyndns.org`, and a default mapping to a machine was created for it. An example of a DNS record for `chord.dyndns.org` domain is given in Figure 5. The record has a very short TTL to ensure that DNS servers do not cache the name

```

chord.dyndns.org: type A, class IN,
addr 146.231.123.55
Name: chord.dyndns.org
Type: (Host address)
Class: IN (0x0001)
Time to live: 1 minute
Data length: 4
Addr: 146.231.123.55
  
```

Fig. 5. DNS record for `chord.dyndns.org` domain.

resolution for longer than one minute, which suits the dynamic requirements of a peer-to-peer overlay.

While this method achieves the required purpose, it is undesirable for two main reasons. Firstly, it introduces external software dependencies on the user agent that embeds OverCord. In particular, `ddclient` requires that a working Perl installation exist on the host device. Secondly, it was considered superfluous given the relatively simple function that the client provides.

##### B. Phase 2 - HTTP Requests

Upon further investigation, we were able to determine more clearly how the `ddclient` program works. We observed that whenever it was running and attempted an update, it would send an HTTP GET request to the web address `http://members.dyndns.org/nic/update` and append query string arguments to the end of it that contained the name of the domain, and the updated IP address to use. Since these parameters could be determined from the user agent embedding OverCord, it became possible to do away with `ddclient`, and simply create a new special class that was responsible for collecting this information, and execute the HTTP GET request. Due to the fact that the dynamic DNS provider will present a security challenge to the request asking for authentication details (which would typically be recorded in `ddclient`'s configuration script), a Java Authenticator class was used to respond to such challenges, and pass the credentials back to the web server.

#### V. NODE BEHAVIOUR

When an OverCord-enabled user agent attempts to join a particular peer-to-peer overlay, it will try to discover if the incumbent P2P SIP proxy peer is

online and accepting connections from other peers. By using the domain name configured into the user agent, the node obtains the IP address of the proxy peer by using a DNS lookup. It will then attempt, at most, two tests. The first determines if the proxy peer is reachable. It achieves this by probing the proxy peer, and waits for a specific amount of time for a response. If the proxy peer is not reachable, then the node which is joining the overlay will assume the proxy peer is no longer available, in which case it updates the dynamic DNS binding. If the indicated IP address is reachable, a second test is attempted, by which the joining node determines if the indicated node has confirmed that it is playing its role by listening on the default port number for incoming connections (in the experiment, port 49152). This port number is a private port number which is used for accepting probes from other nodes in the overlay. If the joining node discovers that the port is not open, it performs a dynamic DNS update and begins listening for connections on the default port and also updates the database record in the overlay for the location of P2P SIP proxy peer. If on the other hand, the proxy peer is accepting connections, then the node starts as normal, but creates a timer that schedules regular probes to the proxy peers location.

The P2P SIP proxy peer is the first point of contact into the overlay from the global Internet. When a request is received by the proxy, it must first examine the incoming request in order to determine the target. If the Request URI in the SIP message indicates the proxy peer itself, then it processes the request as per RFC 3261 [10]. If the Request URI does not belong to the proxy peer, it must consult the distributed database service, and use the binding (or bindings) stored therein to forward the request to the appropriate destination (or destinations).

## VI. RESULTS

In order to test the success of the use of dynamic DNS for interoperation, the popular open source SIP server called SIP Express Router (SER) was used. In the experiment, an OverCord-enabled application was started, which was assigned to a hypothetical user called Alice, a member of the chord.dyndns.org domain. Since her user agent creates the overlay, the host address of her machine is entered in the DNS records for chord.dyndns.org. The user Alice

has a hypothetical friend called linphone, a member of a centralised domain known as deebee.dsl.ru.ac.za, which is served by an instance of SER. Firstly, presence subscription and advertisement were attempted between Alice and linphone. The user linphone registers with SER and subsequently sends a PUBLISH method to the server. The PUBLISH message attaches Presence Information Document Format (PIDF) data which contains presence information, which SER will store. The user then sends a request to obtain the presence status of the user Alice who is in the chord.dyndns.org overlay. SER proxies the request to the decentralised overlay by accessing the current P2P SIP proxy peer in chord.dyndns.org, which happens to be Alice's host itself. In this case, signaling was successful.

If Alice agrees to accept the subscription request from the remote user linphone, she sends a notification to that user. Alice will subsequently generate a request of her own, subscribing to the presence status of linphone. SER allocates a lease on the presence status of linphone to Alice and sends a NOTIFY message to Alice, informing her of the present status of this user. The SIP application must handle the refreshing of these leases since the SIP stack does not do this on the behalf of the user agent itself. The contact list on the user linphone's user agent appears below in Figure 6 showing the insertion of the user Alice and her status.

The screenshot shown in Figure 6 demonstrates that it is possible to achieve interoperation from centralised to decentralised overlay networks using dynamic updates in DNS. OverCord is able to facilitate routing of external messages to overlay nodes, by virtue of the proxy behaviour embedded in the P2P SIP proxy. To prove that the design is able to achieve this, another user is introduced, named Bob, who joins the chord.dyndns.org overlay after Alice, but while Alice's node is still the P2P SIP proxy peer. In the experiment, it was possible to provide presence and instant messaging communication between linphone and Bob through Alice's user agent performing the proxy function, as shown in Figure 7.

## VII. CONCLUSION

This paper has described the extension of the OverCord peer-to-peer framework with dynamic DNS abilities to provide interoperation with conventional



Fig. 6. Presence status of a peer-to-peer user agent displayed on a client-server user agent.

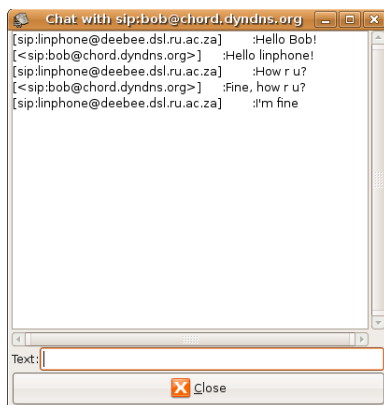


Fig. 7. Instant messaging supported by a P2P SIP proxy peer.

SIP systems. This feature would be an important addition to the P2P SIP protocol and would allow a much needed break-out function into conventional SIP. The described solution will also allow communication within a single, organisational SIP overlay network that consist of both centralised and non-centralised entities. This would be useful in situations such as failover of centralised servers or ad-hoc group meetings. The solution aligns itself with the current trend of assigning special roles to nodes that is prevalent in some P2P SIP draft proposals that describe how services are provided in the overlay by specific nodes.

## REFERENCES

- [1] "P2PSIP Working Group Charter," Available Online, 2007, uRL:<http://www.ietf.org/html.charters/p2psip-charter.html>.
- [2] E. Shim, S. Narayanan, and G. Daley, "An Architecture for Peer to Peer Session Initiation Protocol (P2P SIP)," Available Online, February 2006, internet Draft: draft-shim-sipping-p2p-arch-00, work in progress, <http://www.p2psip.org/drafts/draft-shim-sipping-p2p-arch-00.txt>.
- [3] E. Marocco and D. Bryan, "Interworking between P2PSIP Overlays and Conventional SIP Networks," March 2007, internet Draft: draft-marocco-p2psip-interwork-01, work in progress, <http://tools.ietf.org/html/draft-marocco-p2psip-interwork-01.txt>.
- [4] J. Rosenberg and H. Schulzrinne, "RFC 3263: Session Initiation Protocol (SIP)- Locating SIP Servers," June 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3263.txt>
- [5] J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," Available Online, November 2007, internet Draft: draft-ietf-behave-turn-05, work in progress, <http://tools.ietf.org/html/draft-ietf-behave-turn-05>.
- [6] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," University of California Berkeley, Tech. Rep. UCB/CSD-03-1299, 2003.
- [7] M. Tsietsi, "Prototyping a Peer-to-Peer Session Initiation Protocol User Agent," Master's thesis, Rhodes University, South Africa, March 2008.
- [8] M. Tsietsi, A. Terzoli, and G. Wells, "Prototyping a p2p sip user agent with support for multiple overlays," in *MP2P '08*. IEEE Computer Society, 2008, pp. 474–479.
- [9] —, "A Peer to Peer Layer for SIP Based Realtime Multimedia Communication," in *SATNAC Conference Proceedings*, 2007.
- [10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP: Session Initiation Protocol," 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3261.txt>

**Mosiuoa Tsietsi** is currently reading towards a PhD degree in Computer Science at Rhodes University, Grahamstown

George Wells is currently Head of Department of Computer Science at Rhodes University, Grahamstown

Alfredo Terzoli is currently Project Director at the Center of Excellence at Rhodes University, Grahamstown