Dynamic Peer-To-Peer Overlays for Voice Systems

Krishna Kishore Dhara
Venkatesh Krishnaswamy
Avaya Labs Research
Lincroft, NJ 07738-1526
{dhara, venky}@avaya.com

Salman Baset
Dept. of Computer Science
Columbia University
New York, NY 10027
salman@cs.columbia.edu

Abstract

Different devices, such as mobile phones, soft phones, or desktop phones, have varying processing power, bandwidth, and media capabilities. Heterogeneous P2P voice systems that are built based on a set of capabilities will not be suitable for devices that have different capabilities. In this paper, we present an architecture for P2P voice systems that can dynamically change P2P overlay mechanisms to better suit different device, user, and feature requirements of a P2P voice system. As a first step towards realizing the architecture, we propose a P2P-SIP based architecture that separates out P2P mechanisms from SIP. Our architecture allows dynamic P2P structural changes, limits bloating of the SIP protocol, and lays a foundation for a flexible hierarchical system.

1. Introduction

Peer-to-peer (P2P) networks have traditionally been used for file and information sharing in particular and for resource sharing in general. The key idea behind such P2P systems is to distribute processing and bandwidth requirements by sharing resources across many different peers. This idea has been extended by Skype [4], Kundan [10], and Bryan et al. [5] to demonstrate the possibility of extending P2P networks to voice services. While Skype apparently uses many P2P techniques [4], it is proprietary and closed. However, there are other works that explore inherently peer-to-peer VoIP protocols such as SIP. The works of Kundan [10] and Bryan [5], demonstrate the integration of the P2P Chord [11] algorithm within SIP [9]. Though such an integration through SIP is open, the close tie up with a specific structure, in their case with Chord [11] in the form of SIP headers, is inflexible to new structures and is not open to dynamic changes required by heterogeneous P2P voice systems.

This problem is severe if some peers require different P2P properties based on end-point properties and capabilities. For example, voice peers that are within an enterprise may have properties such as trust relationships or bandwidth requirements that are different than mobile peers, which are part of the same P2P network. Similarly, mobile peers will have different join/leave intervals than desk top phones or PCs. Hence, a separation of P2P properties from the underlying voice and transport protocols is desirable for heterogeneous P2P voice systems. In this paper, we present an architecture that layers P2P overlays and that separates P2P related issues from the underlying voice and/or transport layer. However, these overlays are logical overlays and can be realized with one or more physical overlays.

The main contributions of this paper are as follows.

- A layered architecture for P2P voice systems that allows dynamic change of overlays at any level based on the properties of devices, users, or features of voice systems.
- A new SIP P2P mechanism that separates the P2P overlays and SIP.

The rest of the paper is organized as follows. In the next section, we present the motivation and the architecture for P2P overlays for voice systems. Section 3 presents background and related work. Section 4 presents XML overlay mechanisms over SIP and details of two overlay mechanisms – user trust management overlay and user identity overlay. Examples from our prototype implementation are presented in section 5. Sections 6 and 7 conclude with future work and conclusions.

2. Architecture for P2P Voice Systems

2.1. Why layering?

The performance of a P2P system often depends on the cost of the lookup and on the cost of the formation of the P2P network itself. These bounds depend on the choice of

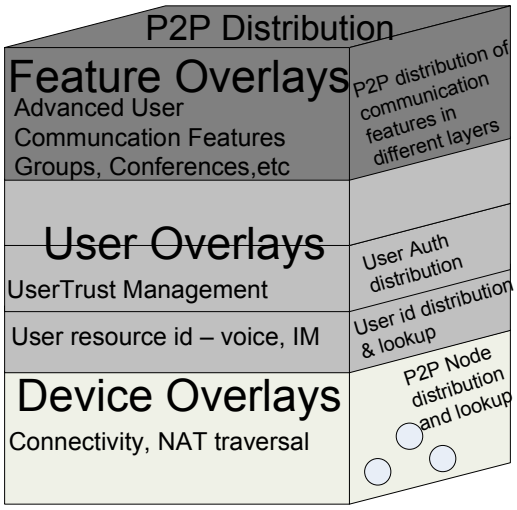

Figure 1. Overlay architecture of a P2P voice network

the structure of the P2P system, which in turn depends on the characteristics of the P2P nodes. While a particular P2P structure may offer a lower bound on the lookup but because of its high cost of maintaining the structure, it might be unsuitable for peers, such as mobile nodes, that dynamically change their network characteristics. For adaptability, we need a framework that allows dynamic changes to P2P overlays.

P2P voice systems require several kinds of lookup services. For example, they need device lookup for establishing communication, user look up for establishing communication sessions, and other service or feature-oriented lookup on a single or a group of users for advanced voice features or services. Choosing a particular P2P structure is difficult as each of the lookup services have different characteristics. We need the ability to choose several P2P structures, one for each layer. Also changing the P2P overlay at one layer, for example at the device layer, should not force a P2P voice deployment to change the P2P overlay at the feature layer or at the user layer. Separation of P2P overlays from each other and from the underlying transport network provides such flexibility for P2P voice systems.

We propose a layered architecture for P2P voice systems that

- allows a dynamic choice of P2P overlays based on the characteristics of devices, users, or features.
- permits P2P voice systems to modify overlays independent of each other.

2.2. Layered Architecture

Figure 1 presents our architecture. We structure a P2P voice system as a hierarchy of layers, each layer with its own P2P distribution and lookup. The layers are grouped in three categories – *device overlays*, *user overlays*, and *feature overlays*. Each of these overlay category can be further layered, for example, the user overlay category is further layered as an authentication layer and a P2P user layer. These layers are logical and can be realized through one or more physical layers. Also, overlays may or may not depend on other overlays.

Typically, the device overlay layer organizes nodes into P2P structures using their network properties such as connectivity, NAT and firewall traversal, bandwidth, and leave and join frequencies. The user layer distributes user information such as user identity, communication abilities, authentication information, etc, across different nodes in the P2P network. The feature overlays layer deals with advanced communication features such as group features, conferencing, and other voice features that are part of any voice network.

2.3. P2P Overlays in SIP

The architecture proposed in Figure 1 is realized by capturing various P2P overlays as XML and using SIP as the transport protocol. The XML content is dependent on the particular P2P mechanism that is an overlay on SIP. For example, if a user trust management overlay is to be realized then we use an XML that captures the trust management and carry it over SIP messages. Similarly, if another feature such as a conferencing is to be implemented as an overlay, then the XML should capture details such as conferencing abilities of its peers and their willingness to mix media, etc. In the next section, we show two such P2P mechanisms that use SIP REGISTER messages to carry XML bodies.

SIP REGISTER method allows a VoIP client to update its contact location with a SIP registrar when it joins the network. The registration creates an address of record (AOR) or a binding of user id with the IP address and port number from where the user can be found. The REGISTER method contains a number of useful headers like 'Expires' and 'Max-Forwards'. The Expires header indicates the time interval for which the binding is valid. The Max-Forwards header indicates the maximum number of hops to which the request can be forwarded. This registration can be periodically refreshed by sending a refresh REGISTER before the expiry of registration time interval. The registration can be terminated by sending a REGISTER with an 'Expires' header of zero.

SIP has built in transaction state machine which means that there is no need to re-create a new transaction state machine for exchanging P2P messages. We reuse SIP headers as much as possible; Expires header indicates when the P2P binding will expire and Max-Forwards is used to indicate the maximum number of peers the request can travel.

3. Background and Related work

3.1. Chord

Chord is a ring-based structured peer-to-peer architecture that uses dynamic hash tables (DHTs). Each node in Chord is assigned an m-bit identifier where m is the number of bits generated by a standard hashing algorithm such as SHA-1. Each node in Chord keeps track of its predecessor, successor, and maintains a table or a state called finger table containing m entries. For a node, n, the i^{th} entry in the finger table contains the identity of the first node, s, that succeeds n by at least 2^{i-1} on the identifier circle, i.e., s = successor$(n + 2^{i-1})$, where $1 < i < m$. Chord differentiates between hash identifiers of nodes and hash identifiers of keys. It computes the identifier of a node by hashing its IP address while it computes the identifier of a key by hashing the key value. A node is inserted in the Chord ring based on the node identifier while the key is stored on successor. We use chord in our examples in the rest of the paper.

3.2. P2P Voice Systems

Skype [4] uses peer-to-peer technologies to offer Voice, IM, and other services through internet. Skype distributes two central resource across its nodes - central database to manage a unique namespace of users and NAT/firewall traversal servers. However, Skype is proprietary, does not work with non-Skype P2P systems, and by its nature does not allow any dynamic P2P structures. Nimcat [1] offers enterprise class peer-to-peer voice systems but like Skype is proprietary.

3.3. SIP P2P Voice Systems

The work by Kundan and Schulzrinne [10] proposes a Chord based VoIP architecture that uses SIP [9] as the underlying transport protocol. Nodes with high capacity and availability become super nodes. Only super nodes form a Chord ring. The low capacity nodes connect to one of the super-nodes. A super-node also acts as a registrar on behalf of its users. The super-nodes exchange DHT information in the SIP protocol itself.

The proposal by David Bryan et al [5] is quite similar to Kundan and Schulzrinne architecture. They use SIP to transport Chord DHT messages to peers. The Chord messages are part of the SIP protocol.

Though both the works of Kundan [10] and Bryan [5] implement P2P overlays for nodes and users, the two overlays are not modular. Also, the close integration with SIP renders them inflexible to adopt to a more suitable P2P protocol in future. We propose a decoupling of P2P protocol and SIP, which allows the flexibility to plug in a different P2P protocol at a later time.

3.4. P2P Protocols

XMPP [6] and JXTA [2] are text-based protocols for exchanging P2P information. XMPP is a text-based IM protocol that has recently been adapted by G-Talk for internet telephony. Functionally, it is quite similar to SIP except that its message syntax is in XML. Since we are targeting SIP based networks our choice is to tailor SIP for P2P needs.

JXTA [2] is an XML based peer-to-peer protocol developed by Sun and provides a solution for creating a peer-to-peer network. We think that the use of JXTA as a P2P IP telephony protocol needs further exploration.

3.5. PKI based P2P Security

Thomas Wolfl [12] proposes a distributed PKI based P2P mechanism for checking authenticity of a node's public key. He proposes an efficient search and transfer of certificates and trust-recommendations among peers. Our focus in this paper is to use existing trust management mechanisms as a P2P overlay and to realize it in a SIP P2P voice system.

4. Overlays for User Authentication and User Identities

In this section we illustrate how the P2P overlay mechanisms depicted in Figure 1 are realized. We focus on the user overlays, specifically two overlays that are illustrated in Figure 1, viz., user trust management overlay and user resource identity overlay. By user trust management overlay, we mean an overlay mechanism that provides trust among peers at the user layer. That is, peers receiving messages from other peer users should be able to verify that the message actually came from them. User resource identity overlay mechanism captures the distribution of user resources in a P2P voice network and their lookup.

To illustrate the independence of the overlays we pick different mechanisms for the overlays described below.

User Trust Overlay: A centralized authentication server with P2P trust verification.

User Identity Overlay: A P2P chord overlay structure.

In P2P SIP networks, each node in the network acts as both a registrar and a proxy. Requests to each node are either forwarded, or accepted with a success or failure. Any new SIP peer willing to join a P2P network should register itself with the P2P structure, that is the SIP peer sends a SIP REGISTER to a (bootstrap) node in the network. The nodes in the network receiving the SIP REGISTER message either forwards it to the closest peer from its finger table or sends back a final reply. This information is contained in SIP reply (OK) message to the REGISTER. This process is

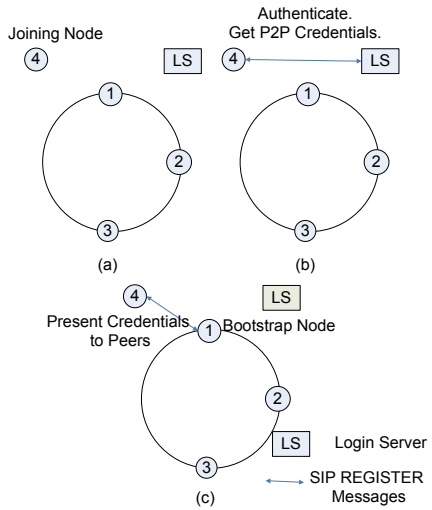

Figure 2. Figures (a), (b), and (c) show the steps in authentication of a joining peer

used in both the trust management overlay and user resource identity overlay mechanisms described below.

4.1. User Trust Overlay

The user trust management overlay uses a simple mechanism for authentication. A joining peer in a P2P network sends a SIP REGISTER with a public key and its node to a login server (LS), which authenticates and issues a certificate to be presented to peers in the network. The joining peer then presents this certificate along with its signature to its peers in further transactions.

Figure 2 refers to the authentication steps required for a joining peer. Figure 2 (a) shows a P2P chord like structure with three nodes, whose identities are represented as 1, 2, and 3. A node with identity 4 is the joining node. LS represents a SIP login server (LS) that can authenticate a user and generate signed public key as a response. The LS ensures a unique namespace for all the peers in the P2P network. The SIP connection parameters of LS can be preconfigured or found out through DNS.

Figure 2 (b) depicts the second step where a joining node sends a SIP REGISTER [9] method that includes its public key and node identity as the body. After the authentication process, in the body of a reply for the REGISTER message, the joining node receives a certified signature that contains its public key signed by LS. Figure 2 (c) depicts the step where the joining node sends its signature and identity to a bootstrap node to find its place in the chord. In our prototype, for simplicity, the bootstrap node's P2P identity along with its connection parameters are passed to the joining node. However, finding out the bootstrap node's parameters can be achieved using other means such as pre-configuring, multicasting, or broadcasting.

The high-level SIP transaction of the joining node and the LS is given below. The SIP header `Content-Type:` identifies the content of the body and `<XML-BODY>` contains the relevant encryption information needed for the transaction.

```
// Joining Node -> LS
// REGISTER MESSAGE WITH XML BODY
REGISTER sip:ls@atlanta.com SIP/2.0
From: sip:alice@atlanta.com;tag=1_F
Content-Type: application/p2p+xml
<XML-BODY>

// LS -> Joining Node
// (after authentication)
// REGISTER REPLY WITH XML BODY
SIP/2.0 200 OK
Content-Type: application/p2p+xml
<XML-BODY>
```

The joining node then can use the certificate obtained from the SIP OK reply along with its signature for further transactions with the P2P nodes. The specific signature/certificate mechanism with the LS or with the peer nodes can be replaced without any changes to the framework. This overlay can be substituted with a mechanism that distributes functionality of the LS across peers and uses SIP REGISTER messages to find the appropriate node to obtain certificates.

4.2. User Identity Overlay

In this section we present an overlay mechanism for distributing user identities across a P2P network.

Figure 3 shows various steps a node has to perform before and after joining a P2P network. The joining node sends a REGISTER message (Figure 3 (a)) to the bootstrap node. The bootstrap node verifies the signature of the joining node. If it is successful, then it checks the identity to see if joining node will be its neighbor. In that case, it updates its finger table with the user identity information from the registration and sends out a reply to the REGISTER message with a *success*XML body along with its signature (Figure 3 (b)) for verification. Otherwise it *forwards* the address and identity of the closest successor from its finger table (Figure 3 (c)).

A joining node upon receiving a reply for its REGISTER message with a P2P body, first verifies the signature and then looks at the P2P XML body for the *forwards* or *success* reply. If it is a P2P *success* reply Figure 3 (b) then it inserts itself into the P2P structure by updating its predecessor and successor nodes, its finger table and by sending out requests to update finger table entries of other relevant nodes (Figure 3 (d)). If it is a P2P *forward* message Figure 3 (c) then the joining node, sends a REGISTER message to

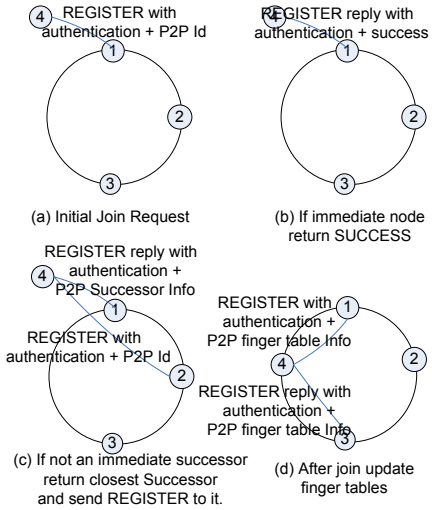

Figure 3. Figures (a), (b), and (c) show the steps in forming a P2P structure. Figure (d) shows the distribution of finger table information.

the address obtained in the P2P body of the REGISTER reply. This step is continued until the joining node receives a P2P *success* or a *failure* in its REGISTER reply.

A node gracefully exiting the P2P network sends a REGISTER message to the peers in the finger table. The nodes that are impacted change their finger table entries and/or their predecessor/successor links appropriately. Nodes exiting ungracefully must be discovered through REGISTER refresh messages. When a node (a predecessor or a successor) discovers a failure, it enters a recovery mode where it tries to form the P2P structure again. The overlay mechanisms in the lower layers, device overlay layer, should also have failure recovery mechanisms.

Call establishment and tear down is performed through SIP INVITEs [9]. A node receiving a SIP INVITE verifies the signature by looking at its body. If it is recipient of the INVITE, it answers the call with its connection parameters and its digital signature in the body. Otherwise, it forwards the INVITE to a closest node in the network by looking at its finger table. If no such nodes are available it sends a SIP 4XX message indicating a failure.

5. Prototype Details

The overlay mechanisms presented in the previous section, to a large extent, have been realized through a prototype SIP P2P system. We use chord [11] algorithm for our P2P structure and use a PKI [3] based trust management scheme. The rest of this section sketches out details of our prototype implementation.

The details of the XML messages in our prototype are specific to our implementation of the chord network. The particular choice of the XML for chord has no bearing on the overlay scheme presented in this paper. Hence, we do not elaborate on the XML itself but instead present a few examples to illustrate the P2P-XML overlays in SIP.

Also, in our prototype, the logical P2P overlays for trust management and user identity are realized through one chord overlay by combining the P2P XML bodies in the SIP REGISTER messages. Section 5.1 presents the initial steps of a node authenticating and obtaining its certificate. Section 5.2 sketches some details of how an authenticated node uses its certificate and signature along with its user identity to join a chord.

5.1. User Trust Overlay

Our client, a SIP P2P voice client (PVC), generates a public-private key pair and uses a user id and password to authenticate with a login server. The PVC in our prototype is preconfigured with a LS and sends out a REGISTER message with its hashed node id and public key as the body of the REGISTER. The following sample message shows a SIP REGISTER message sent from a node with identity 0.

```
// PVC to LS -- Registration Message
// Headers are truncated/some deleted
REGISTER sip:ls@atlanta.com SIP/2.0
From: sip:alice@atlanta.com;tag=1_F
User-Agent: ALP2PNode
Content-Type: application/p2p+xml
<P2Pxml>
<CertificateRequest>
<NodeID>0</NodeID>
<PubKey>43f..truncated</PubKey>
</CertificateRequest>
</P2Pxml>
```

The LS replies back, typically after a REGISTER challenge, with an OK message that generates a certificate for the node and returns the network parameters of the P2P bootstrap node. LS also returns its public key that is used by the PVC to verify signatures presented to it by its peers.

Each PVC has the public key of the LS and can verify the signed digital signatures presented by peers during message exchange. With the bootstrap node's identity, the PVC has enough information and trust credentials to join a SIP P2P network.

5.2. User Identity Overlay

A PVC willing to join a P2P chord overlay network sends out REGISTER message to the bootstrap node to join the network. PVC computes its hash using SHA-1 (our illustrations do not use this mechanism) and uses this as its SIP P2P identity to register to the network. The bootstrap

node replies with a *success* if it is the immediate successor of the joining node otherwise it replies with a *forwards*. The following messages, edited from our prototype, illustrate this.

```
// PVC -- Peer
REGISTER sip:atlanta.com SIP/2.0
From: sip:bob@atlanta;tag=11
Content-Type: application/p2p+xml
<BootstrapRegRequest>
<NodeID>2</NodeID>
<NodeURL>sip:10.8.6.176</NodeURL>
<Certificate>Xj1...truncated</Certificate>
<Signature>v2R...truncated</Signature>
</BootstrapRegRequest>

// Reply -- truncated
SIP/2.0 200 OK
From: sip:bob@atlanta.com;tag=11
Content-Type: application/p2p+xml
<BootstrapOK>
<NodeID>0</NodeID>
<Certificate>fFD...truncated</Certificate>
<Signature>v2p...truncated</Signature>
<SuccessorURL>sip:alice@atl..</SuccessorURL>
<SuccessorID>0</SuccessorID>
<RefreshRate>100</RefreshRate>
<FingerTable>
```

In the reply to the bootstrap REGISTER, the joining node gets its position in the chord network through the predecessor and successor information. It also receives a copy of the predecessor's finger table as a starting point to build its finger table. As part of joining the network and building its finger table it sends out several messages to its peers using REGISTER messages. Corresponding responses to these messages are received in the REGISTER reply.

6. Future Work

Several open research problems need to be solved for a feature-rich P2P voice systems that allow end-points and users with varying capabilities. These include overlays that offer rich features and services, XML schemas that that can abstract various P2P algorithms, and XML abstractions of certificate management. SIP P2P voice systems need to look at ways of optimizing SIP messages perhaps by looking at other SIP event mechanisms for finger table operations, and for status of peers. Several network issues such as NAT and firewall traversal, routing, and P2P optimizations for bandwidth and connectivity need to be solved to successfully accommodate heterogeneous voice end-points.

7. Conclusion

In this paper, we presented a layered architecture for P2P voice systems. The merits of using such an architecture are two fold. It isolates the concerns and restrictions at each layer and allows the choice of a P2P overlay based on a specific set of parameters that are relevant to that layer. We also presented a SIP P2P system that uses SIP as a transport protocol and separates P2P overlay mechanisms from SIP.

References

[1] Nimcat networks. *http://www.nimcatnetworks.com/*, 2005.

[2] Project jxta. *http://www.jxta.org/*, 2005.

[3] C. Adams and S. Lloyd. Understanding pki. Book, 2nd ed., Addison-Wesley, Boston, 2003.

[4] Salman Abdul Baset and Henning Schulzrinne. An analysis of the Skype peer-to-peer Internet telephony protocol. Technical Report CUCS-039-04, Computer Science Department, Columbia University, September 2004.

[5] David A. Bryan, Bruce B. Lowekamp, and Cullen Jennings. Sosimple: A serverless, standards-based, p2p sip communication system. *Appears in AAA-IDEA*, 2005.

[6] P. Saint-Andre Ed. Extensible messaging and presence protocol. RFC 3920, Internet Engineering Task Force, October 2004.

[7] J. Rosenberg, J.Winberger, C.Huitema, and R. Mahy. Stun: Simple traveral of user datagram protocol. RFC 3489, Internet Engineering Task Force, March 2003.

[8] J. Rosenberg, R. Mahy, and C.Huitema. midcom-turn-07. Draft, Internet Engineering Task Force, February 2005.

[9] J. Rosenberg, Henning Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.

[10] Kundan Singh and Henning Schulzrinne. Peer-to-peer internet telephony using sip. In *NOSSDAV*, pages 63–68, 2005.

[11] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. Chord:A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17-32, Feb 2003.

[12] Thomas Wolfl. Public-key-infrastructure based on a peer-to-peer network. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.