# Peer-to-Peer Internet Telephony using SIP

Kundan Singh and Henning Schulzrinne
Department of Computer Science, Columbia University
1214 Amsterdam Ave, Mail Code 0401
New York, NY 10027, USA

{kns10,hgs}@cs.columbia.edu

## ABSTRACT

P2P systems inherently have high scalability, robustness and fault tolerance because there is no centralized server and the network self-organizes itself. This is achieved at the cost of higher latency for locating the resources of interest in the P2P overlay network. Internet telephony can be viewed as an application of P2P architecture where the participants form a self-organizing P2P overlay network to locate and communicate with other participants. We propose a pure P2P architecture for the Session Initiation Protocol (SIP)-based IP telephony systems. Our P2P-SIP architecture supports basic user registration and call setup as well as advanced services such as offline message delivery, voice/video mails and multi-party conferencing. Additionally, we give an overview of our implementation.

## Categories and Subject Descriptors

C.2.2 [**Computer Communication Networks**]: Network Protocols—*Distributed Systems, Applications*

## General Terms

Algorithms, Design, Experimentation

## Keywords

peer-to-peer, Internet telephony, SIP

## 1. INTRODUCTION

The existing Internet telephony client-server architecture based on IETF's Session Initiation Protocol (SIP [14]) employs a registration server for every domain. The majority of the system cost is in maintenance and configuration, typically by a dedicated system administrator in the domain. It also means that quickly setting up the system in a small network (e.g., for emergency communications or at a conference) is not easy. On the other hand, peer-to-peer (P2P) systems [11] are inherently scalable and reliable because of the lack of a single point of failure. P2P systems are robust against global, catastrophic failure, although single nodes may fail.

We propose a P2P Internet telephony architecture using SIP. There are two main motivations for P2P-SIP: a fully distributed model to increase robustness, and the ability to deploy without modifying *controlled* infrastructure such as DNS. We analyze various design alternatives and present our P2P-SIP endpoint using Chord [18] as the underlying distributed hash table (DHT). Our novel hybrid architecture allows both traditional SIP telephony as well as user lookup on P2P network if the local domain does not have a SIP server. We use SIP to implement various DHT functions in P2P-SIP such as peer discovery, user registration, node failure detection, user location and call setup by replacing DNS [13] with P2P for the next hop lookup in SIP.

We have implemented a P2P-SIP adaptor, SIPPEER [17], that allows existing or new SIP user agents to connect to the P2P-SIP network without modifying the user agent. For example, SIPPEER can run on the same host as the PC-based SIP user agent and act as its outbound proxy. SIPPEER can also act as a standalone SIP user agent, proxy or registration server with command line user interface. Our modular design allows reusable and replaceable components. For example, Chord could be replaced by another DHT without affecting the rest of the implementation. The open architecture allows installing new services without affecting the existing design. For example, a new voice mail module can be added to the existing node.

Besides the P2P scalability and reliability, we claim the following additional benefits for P2P-SIP:

**No maintenance or configuration:** The system works out-of-the-box without requiring any tedious server installation, including NAT and firewall configuration. Our work extends the goals of the IETF Zeroconf [3] Working Group to multimedia communication and collaboration systems.

**Interoperability:** Unlike other P2P systems such as Skype [2], our architecture uses SIP messages for communicating with other peers. This readily interworks with any existing IP telephony infrastructure such as SIP-PSTN gateways or server-based IP PBX such as Asterisk.

These advantages come at the cost of increased *resource lookup delay* and security threats. Unlike $O(1)$ lookup cost in a classical client-server based systems, the P2P lookup cost can be much higher. A reliable framework for authentication and reputation without centralized elements is outside the scope of this paper.

We provide background on P2P and SIP related work in Section 2. Section 3 gives an overview of our P2P-SIP architecture, user registration and call setup. We give an overview of our implementation and discuss some design issues such as naming and authentication in Section 4. Section 5 describes advanced services in P2P-SIP. Section 6 predicts performance of the system in terms of scalability, reliability and call setup latency. Section 7 lists vari-

*NOSSDAV'05,* June 13–14, 2005, Stevenson, Washington, USA.
Copyright 2005 ACM 1-58113-987-X/05/0006 ...$5.00.

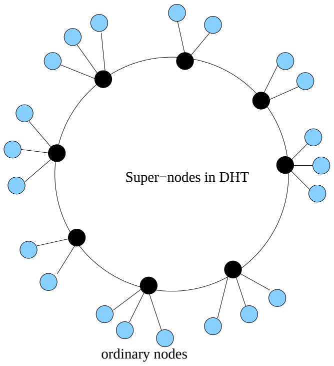

**Figure 1: Nodes attached to super-nodes of Chord**

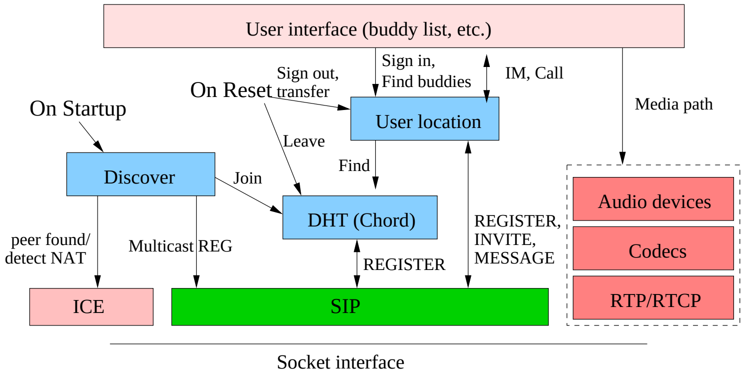

**Figure 2: Block diagram of a P2P-SIP node**

ous open issues such as security threats and deployment scenarios for further study. Finally, Section 8 presents our conclusions and future directions.

## 2. BACKGROUND AND RELATED WORK

**Chord:** Chord [18] is a ring-based distributed hash table (DHT) for structured P2P systems where each node stores at most $log(N)$ entries (or state) in its *finger table* to point to other peers. Lookup is done in $O(log(N))$ time. The *iterative* and *recursive* lookup styles in Chord directly map to the *redirect* and *proxy* behavior, respectively, in SIP. Research in DHT is complementary to our work, since our architecture can use new innovations or optimizations in the underlying DHT.

**Skype:** Skype [2, 4] is a free P2P application based on the Kazaa architecture for Internet telephony and instant messaging. The protocol is proprietary unlike SIP. Secondly, it has centralized elements for login authentication [4]. In a way, the Skype architecture is no different from the classical SIP telephony architecture, except that the Global Index Server assigns a *super-node* for a new joining node. The super-node, similar to the SIP registrar, proxy and presence server, maintains the presence information for this node, and locates other users by communicating with other super-nodes. A node that has enough capacity and availability can become a super-node. We believe that the lookup is based on some variation of flooding, similar to Kazaa, instead of using the provably efficient DHT-based lookup. The main advantage of Skype is that it implements the equivalent of STUN and TURN servers in the node itself to handle NAT [12], unlike explicit server configuration in existing SIP applications.

**SIP:** Unlike P2P, existing SIP-based telephony [14] has a client-server architecture. SIP telephony can be treated as a P2P system with static set of super-nodes (SIP servers) where the lookup is based on DNS instead of a hash key. However, using a pure P2P architecture instead of static set of SIP servers improves the reliability and allows the system to dynamically adapt to node failures.

**P2P and SIP:** More recently work has been started on combining SIP and P2P [15, 10, 8]. SIP can be combined with P2P in two ways: (1) replace the SIP user registration and lookup by an existing P2P protocol, and additionally (2) implement this P2P algorithm using SIP messaging. The former approach uses an existing P2P protocol [8], whereas we focus on the latter approach that builds the P2P network among the peer nodes using standard SIP messages with no change in message semantics [15, 16]. The disadvantage of the second approach is in larger transport message size. Its advantages include (1) use of existing SIP components such as voice mail service, (2) no dependence on existence of external P2P networks, and (3) built-in media relays for firewalls and NATs. SoSIMPLE [6] started with the first approach but is moving towards the latter architecture [5]. SIPshare [1] is an unstructured P2P file sharing application using the SIP SUBSCRIBE and NOTIFY messages.

**Difference with file sharing:** Table 1 summarizes the similarity and differences between file sharing and multimedia conferencing in the context of P2P. In particular, for Internet conferencing, data storage is not an issue. A single P2P-SIP node can handle many more requests than a file sharing node due to low data volume. Caching of location information is not useful because compared to the file access pattern which follows the Zipf distribution, the call access pattern is more uniformly distributed. Moreover, most residential users are likely to get new DHCP IP address every time they connect to the Internet making the cache entry for this user location stale. The file sharing and directory lookup-based systems can tolerate high lookup latency due to the fact that the user does not need to actively wait for the file to download, and the actual file download time tends to be larger than the lookup latency. On the other hand, an IP telephony caller actively waits for the phone on the other side to ring. For file sharing applications, multiple almost-exact copies of a popular file may be available (e.g., independently ripped by different peers). So node reliability does not matter. On the other hand, in the case of IP telephony, we want to talk to the right person, and not some similar person!

**Table 1: Different applications of P2P**

| Properties | File sharing | Conferencing |
|---|---|---|
| Data storage | Yes | No |
| Caching | Yes | No |
| Delay sensitive | No | Yes |
| Reliability | Multiple copies | Does not work |

## 3. ARCHITECTURE

We distinguish three designs for using a DHT. On one extreme, the DHT can be used in the server farm among the servers while still maintaining the client-server architecture. On the other extreme, all the nodes become part of DHT. We choose an intermediate design as shown in Fig. 1 where some of the nodes with high capacity (bandwidth, CPU, memory) and availability (uptime, public IP address) are made super-nodes and form the DHT, whereas other ordinary nodes attach to one or more super-nodes without being part of the DHT.

Fig. 2 shows the proposed block diagram of the different components in our P2P-SIP node. When the node starts up and the user signs in with her identifier, the discover module is activated to initiate NAT and firewall detection [12], peer discovery and SIP registration. Multicast SIP registration, cached peer addresses from last boot cycle and pre-configured bootstrap addresses are used to discover an initial set of nodes. The user interface module keeps track of the user's "friends list" and invokes the user location module to locate these friends. User location is obtained using the SIP module or, if this node joins the DHT, the DHT module. The DHT module maintains the peer information (e.g., Chord *finger table*) and performs DHT operations such as find, join and leave.

SIP is used as the underlying protocol for locating another user or node, joining the DHT, registering the user, call setup and instant messaging. Once the user has been located, the call setup or instant messages can be sent directly via the SIP module to the user's phone. SIP REGISTER refresh and OPTIONS messages are used to detect node failure. When a super-node shuts down or fails, the registrations are transferred to other super-nodes in the DHT as appropriate. Other SIP functions such as third-party-call control and call-transfer can be implemented in the similar way. The media path (audio device, codecs and transport) is independent of the P2P-SIP operation.

Some DHTs (e.g., CAN) may allow parallel search to multiple peers, unlike the sequential search in Chord. In this case the super-node may act as a back-to-back user agent (B2BUA) and propagate the SIP message to the neighboring peers. However, parallel search should be avoided to prevent flooding the network, except possibly in the case of emergency call routing, such as 911 calls in the United States.

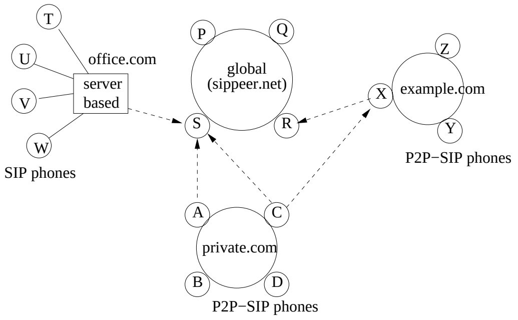

**Figure 3: Example of hybrid systems**

In a real deployment, it is useful to allow multiple P2P-SIP networks (DHTs) to be interconnected. Our hybrid architecture allows both the P2P-SIP network clouds and server-based SIP infrastructure to coexist. There are two approaches: cross register all the users of one network with all the other networks, or locate the user in the other network during call setup. The former method works for small number of known P2P-SIP networks. The latter approach can be implemented using a global naming service such as DNS, or an hierarchy of P2P-SIP networks. In the first case, every P2P-SIP network is represented by a domain name. This is not different than a server-based SIP network where the domain name resolves to one or more bootstrap nodes in that network [13]. In the second case, P2P-SIP is used instead of DNS to resolve the domain name. For example, individual large organizations can have local P2P-SIP network which is connected to the global (public) P2P-SIP network as shown in Fig. 3. The local domain-specific DHT has representative server nodes that are also reachable in the global DHT. For example, key private.com maps to nodes A and C in the global DHT. Any node in the domain-specific DHT can reach the global DHT, and any node in the global DHT can reach the domain-specific DHT via the representative server nodes in the domain.

The hybrid architecture allows the user to register with her provider's SIP server, if available, as well as the P2P-SIP network. Call setup is sent to the SIP destination, if resolved via DNS, as well as to the P2P-SIP network.

# 4. DESIGN AND IMPLEMENTATION

We give an overview of interesting design issues based on our implementation. Details can be found in [17].

## 4.1 Naming

Node and user identifiers are represented using SIP URIs. For example, if a node is listening at transport address 192.1.2.3:8054 for SIP messages and the Chord's hash function gives the key as 17, then the node's URI becomes sip:17@192.1.2.3:8054. A node identifier or key (e.g., 10) in the domain example.com, whose transport address is not known is represented as sip:10@example.com. Every local P2P-SIP network is represented using a DNS domain name, whereas example.invalid is used for the key that has no domain, e.g., in the global DHT. Such node identifiers are useful for DHT maintenance, e.g., to query another node's transport address to populate this node's finger table entries.

User identifiers can be randomly assigned by the system, chosen by the user as a screen name (e.g., alice172@sippeer.net) or chosen by the user as her valid email address (e.g., alice@example.com). The first two approaches allow the user to choose her password, but it is not clear how the P2P node can get the password from the user. We use the last approach as it allows the system to generate a random password and email it to the user for authentication. In the first two approaches, if a password is randomly generated by the system, it can be mailed to the user if the Contact header in the SIP REGISTER request has an email address.

## 4.2 Authentication

When a user signs up with the P2P-SIP network for the first time, we need to verify that the user identifier is valid and indeed belongs to the user. In the absence of public key infrastructure (PKI), the system can generate a new password and send it in an email to the user. This password is used in REGISTER authentication for subsequent sign in. A usable time-to-live, say one month, can be used. The information is refreshed when the user subsequently signs up.

## 4.3 SIP messages

The SIP REGISTER message is used for both user registration and DHT maintenance by the node. The user registration message is similar to the server-based registration with the To header representing the user identifier and the Contact header representing the user contact location.

The SIP REGISTER message is used in two context by the node: *query* and *update*. If a Contact header is present in the message, then it is an update request indicating that the sender wants to update the bindings for the node identifier in the To header. Otherwise, it is a query request, where the sender is requesting to get the Contact information of the node identifier in the To header. In a Chord network of P2P-SIP nodes, the Contact information of the node includes its own transport address, the successors addresses and the predecessor address.

## 4.4 DHT discovery and join

The node sends a SIP REGISTER message with request-URI as sip:224.0.1.75 (SIP REGISTER multicast IPv4 address) and the To header as the local node identifier to discover other P2P-SIP peers in the local network. Additional mechanisms such as service location protocol (SLP) and pre-configured bootstrap node addresses can also be used. The node caches the list of the discovered peer addresses for subsequent reboots.

Once the node discovers a peer, it joins the DHT by sending a SIP REGISTER query to that peer with To header as this node identifier. The successful response contains the successor and predecessor of this node in the existing DHT, which allows this node to update its Chord data structures.

Once the node knows its neighbors in the Chord ring, it sends SIP REGISTER update to them (successor and predecessor), so that they can update their data structures.

Chord stabilization is achieved by periodically sending SIP REGISTER messages to update the successor and predecessor data structures, and to query the finger table entries to verify the local data structures.

## 4.5 SIP message routing

Every node in Chord is responsible for a subset of the key space based on its location in the Chord ring. When the node receives a SIP request, it extracts the destination key as the To header URI for the REGISTER request and request-URI for any other request. For the REGISTER request, if the destination key belongs to the key space of this node, then this node should be the registrar for the destination key. If the user record for this key is present, then a success response is sent, otherwise a failure response is sent. The success response contains the user contact locations or node contacts (local transport address, successors and predecessor addresses) for the user or node registrations, respectively. If the node receives a non-REGISTER request, it proxies or redirects the request to the user contact locations available for the destination user. If the destination key does not belong to the key space of this node, then the request is proxied to the next hop node based on the Chord algorithms and data structures.

## 4.6 Reliability

Chord provides reliability against node failure by storing $log(N)$ successor addresses and replicating keys at some constant ($K$) number of successive nodes. In P2P-SIP, the node update response contains all the $log(N)$ successor addresses, and user registrations are replicated at $K$ successive nodes.

When a node gracefully leaves the network, it unregisters with its successor and predecessor so that they can update their Chord data structures. It also transfers all the registrations to the successor. When a node fails abnormally, its successor and predecessor detect the failure and update their data structures. The stabilization algorithm ensures that the information gets propogated to other relevant nodes in Chord over a period of time.

When the registration is transfered from node A to node B, node B can authenticate node A if it trusts node A, otherwise node B regenerates a new password and sends it to the user's email address. We believe that once we have a P2P reputation system, only the trusted nodes will be present in the DHT. The problem is still there if the registrar node is malicious, and can cause denial of service (DoS).

The P2P-SIP node that stores the user registration, also proxies the call request to that user. Once the call setup is complete, the P2P-SIP node is not needed in the call path.

## 4.7 Adaptor for existing SIP phones

A SIP user agent can use the P2P-SIP node as an outbound proxy and take part in the P2P-SIP network. We have tested our P2P-SIP adaptor, SIPPEER, with various SIP user agents such as the Columbia University's sipc, the Cisco IP phone 7960, the Pingtel IP phone, Xten Networks' X-Lite client v2.0 and Microsoft Windows Messenger.

Some phones do not implement outbound proxy as per the SIP specification [14], which says that the outbound proxy should be treated as a pre-loaded route set. In particular, if the outbound proxy does *not* record route the initial INVITE request, then the subsequent request in the dialog such as BYE should not be sent to the proxy. Suppose the sipc user, alice@example.com, INVITEs the Cisco phone user, bob@example.com, using P2P-SIP. After the call, bob hangs up. The Cisco phone sends the BYE request to the outbound proxy (P2P-SIP node) but the request-URI contains alice@pc2.example.com:5060. The P2P-SIP node may not be able to proxy the request because this URI may not be registered in the P2P-SIP network causing the DHT lookup to fail. We work around this problem in SIPPEER by proxying the request to the request-URI instead of doing a DHT lookup in this case.

# 5. ADVANCED SERVICES

Besides user registration and call routing, our P2P-SIP architecture also supports advanced services such as offline messages and conferencing. Many services can be specified using SIP URIs. For example, sip:staff-meet@office.com can indicate the pre-scheduled conferencing service by the office.com domain, or sip:dialog.voicexml@ivr.net can reach the generic interactive voice response service. Such services can be built transparently in the basic implementation. For example, a SIP conference server can register all the pre-scheduled conferences in the P2P network, an answering machine module can register to receive incoming calls on behalf of all the registered users, and a VoiceXML browser can register the specific voice dialog service such as voice mail access.

## 5.1 Offline messages

Existing persistent P2P file storage systems are not sufficient for IP telephony message storage, because IP telephony also needs message waiting indication. We combine storage at the sender as well as intermediate DHT nodes, to provide a more reliable architecture [16].

## 5.2 Multi-party conferencing

There are three ways to do conferencing using P2P-SIP. One of the participating members can become the mixer for small scale ad hoc conferencing. Alternatively, a completely decentralized SIP conferencing can be used to establish a full-mesh signaling and media relationship among the participating members. Finally, a multicast media distribution tree can be used assuming a small number of senders at any instant. The tradeoff is in terms of reliability (dependence on single node for mixing), complexity and bandwidth utilization, and requires further study.

## 5.3 NAT and firewall traversal

In an ideal world, ISPs and corporate system administrators should enable their NAT and firewall devices with SIP proxies or application level gateways (ALG). However, in practice, this is rarely done. This forces the application developers to write customized kludges to work around the NAT and firewall [12].

Our P2P-SIP node implements the Interactive Connectivity Establishment (ICE) algorithm [12] for NAT traversal. Every node has a built-in STUN and TURN server.

## 5.4 Directory service

One key feature of online chat applications is that it allows people to search for keywords or names. For example, I can search for all the users whose screen names are of the form “bob*” and then pick the one that I want to talk to. This kind of wild-card search is not possible in a DHT based system.

For IP telephony applications, usually people will not want to search using wild-card but may use a combination of first and last names, or may want to search within a few degree (e.g., two) of acquaintances. These searches are possible by registering the first and last name combination in the DHT, and doing blind search with a small hops-to-live value on the acquaintances graph rather than the DHT.

# 6. PERFORMANCE PREDICTION

We plan to do performance measurement for our P2P-SIP user registration and call setup implementation. This section provides some predictions on the scalability, reliability and call setup latency for P2P-SIP.

## 6.1 Scalability

Scalability of the P2P-SIP network depends on the capacity (bandwidth, CPU, memory) of the individual participating super-nodes. Suppose there are $N$ super-nodes in the Chord ring, identifier space is $m$-bit long (i.e., the identifier range is 0 to $2^m - 1$), number of registered users in the system is $n$ (such that number of keys stored per node is approximately $k{=}\frac{n}{N}$), REGISTER refresh rate to successor and predecessor to keep the Chord ring correct is $r_s$, refresh rate for finger table entry is $r_f$, call arrival is poisson distribited with mean $c$ per node, user registration is uniformly distributed with mean interval $t$ per user, and node joining and leaving are poisson distributed with mean $\lambda$. Because average lookup in Chord travels through $O(log(N))$ nodes [18], the finger refresh messages, call arrival messages and user registration refresh messages travel $O(log(N))$ hops. There are $O(log(N))$ finger table entries per node. Node join and leave generate $O((log(N))^2)$ messages. The average message rate per node is sum of the message rates due to refresh, call arrival, user registration and node join or leave, which can be given as:

$$M = \{r_s + r_f(log(N))^2\} + c.log(N) + \frac{k}{t}log(N) + \frac{\lambda(log(N))^2}{N}$$

The message rate in the node determines the bandwidth and CPU utilization for the node. If each node can handle $C$ requests per second, then the equation $C = M$ gives the maximum possible number of nodes, $N_{max}$, in the system, which roughly translates to $N_{max} = 2^{\frac{C}{r+c}}$ for large $N$, where $r$ is the refresh rate and $c$ is the call rate. Note that $\lambda$ is low because nodes which often join and leave are not made super-nodes.

Suppose the node supports 10 requests per second (which is much less than the typical SIP proxy capacity of hundreds of requests per second) with minimum refresh interval of one minute ($r = \frac{1}{60}$) and call rate of one call per minute per node, then the maximum number of nodes in the system can be $2^{10*30}$. We use the high refresh rate in this example compared to the typical one hour SIP registration refresh interval to allow the NAT binding refreshes, if any, or to expedite the node failure detection in the DHT. If more nodes join the system, the super-nodes become overloaded and may deny some incoming call, registration or proxy requests. However, large values of $N$ also increases the call setup latency as we describe below.

## 6.2 Reliability

When a node fails the user registrations stored on that node are lost. To achieve reliability, the refresh rate can be increased (so that node failure detection happens quickly), the user registration refresh rate can be increased (so the the user record is unavailable only for a brief period of time) or the user registration record can be replicated at multiple nodes (e.g., store the user registrations at $K$ successive nodes in Chord). We plan to quantify the effect of each factor on mean time to recover (MTTR) from node failures for a given user record. The equation for average message rate does not change if $\lambda$ includes failure rate along with node join and leave rates.

## 6.3 Call setup latency

The P2P advantages come at the cost of increased call setup latency. For example, with 10,000 nodes in Chord, the average lookup path length is six hops [18], so P2P call setup will take about six times more than the traditional client-server call setup in SIP. With good network condition, single lookup (INVITE response) in SIP is expected to take less than 200 ms. So one or two seconds delay before the phone rings in P2P-SIP is tolerable given that on an average the phone will ring for much longer before the callee picks up.

Due to P2P synchronization latency which depends on refresh rate and node join, leave and failure rates, there may be delay in updating the user records. In this case, it may take multiple retransmissions before call setup is complete. This further increases the call setup latency. Successful user location in Skype takes about three to eight seconds [4].

Some kind of hybrid system may be implemented that takes the advantages of many different structured and unstructured P2P algorithms to further reduce the latency and maintenance cost. For example, there is a proposal on one hop lookups for P2P [7] assuming large storage space in the peer nodes.

# 7. OPEN ISSUES

In addition to the performance measurement of P2P-SIP, we plan to explore additional open issues as described in this section.

## 7.1 Security, trust and reward

A distributed P2P architecture makes the system more prone to *security* issues such as trust (privacy and confidentiality), malicious node behavior (e.g., call dropping) and DoS attacks. For example, a malicious DHT node may not forward the call requests correctly or may log all call requests for future misuse. Hop-by-hop routing of request and responses where each hop (peer) changes the source identifier can be used to provide some confidentiality. Existing P2P reputation systems focus on file sharing (not real-time), have centralized components, assume co-operating peers or have problems of collusion and multiple identities.

The proprietary protocol of Skype makes it difficult for other people to build software that communicates with the Skype clients. Hence, a Skype client can trust the validity of another Skype client (this is not impossible, as Kazaa-Lite showed). On the other hand, P2P-SIP based on open protocols can not trust the validity of another peer. Redundant lookup paths can be used to reduce the risk in structured P2P networks. A BitTorrent-like approach is useful: if a peer can be a supernode, then it can connect to other nodes only if it also routes calls. It will be interesting to answer questions like “how many independent lookups are needed for 99.99% success rate, if at most 5% of the randomly distributed peers are malicious?”

In addition to security threats, P2P-SIP may lose some of the traditional IP telephony services. For example, some of the programmable call routing techniques such as SIP-CGI [9] available for SIP telephony can not work in the P2P-SIP system as we do not want to run potentially malicious script uploaded by some peer on our machines.

Assuming the user identifier to be a valid email address moves the problem of user identity assertion from P2P-SIP to email. Alternate ways to assert user identity is for further study.

Finally, the system should reward the nodes serving in the DHT and discourage "free riding". This requires a P2P electronic credit or debit service.

### 7.2 Media routing

In the presence of NAT and firewalls, the media relay that gives the lowest delay in the media path between the caller and callee endpoints should be located. If the media relay node fails or leaves the DHT, an alternate relay should be located and used in the same call.

### 7.3 Deployment

Besides the Internet wide P2P-SIP network, P2P-SIP can be used within a LAN to save infrastructure cost of setting up an enterprise VoIP system, or among the servers of an Internet telephony service provider (ITSP) to distribute load.

## 8. CONCLUSIONS AND FUTURE WORK

We propose a pure P2P architecture for SIP telephony. The architecture provides reliability and scalability inherent in P2P systems, in addition to interoperability with existing SIP infrastructure.

We have implemented a P2P-SIP node [17] for multimedia communication using our SIP C++ library. We will be doing performance measurement for reliability and scalability on our actual system instead of using simulations.

More work is needed in advanced services such as large scale application level multicast conferencing using P2P, distributed reputation system for peers, and PSTN interworking related issues such as authentication and accounting. There should be a reasonable incentive to become a super-node to provide services to other peers.

Some kind of hybrid system may be implemented that takes the advantages of many different structured P2P algorithms to further reduce the latency and maintenance cost [7]. Other issues such as regulatory and economic impact, security as well as reliable 911 services are for further study.

Finally, we conclude on a note that unless the SIP servers (proxies, registrars) are widely deployed, we will need P2P-based interoperable IP telephony tools so that everyone can use the system. Such P2P-SIP architecture can be extended to other protocols such as H.323, or other DHTs such as Content Addressable Network (CAN).

## Acknowledgment

Salman Abdul Baset helped in understanding the Skype architecture. The work is supported by grant from SIPquest Inc., and the National Science Foundation under Grant No. 0202063.

## 9. REFERENCES

[1] SIPshare: SIP Beyond Voice and Video. http://www.research.earthlink.net/p2p/.

[2] Skype: Free internet telephony that just works. http://www.skype.com.

[3] Zero configuration networking (zeroconf). http://www.ietf.org/html.charters/zeroconf-charter.html.

[4] S. Baset and H. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. Technical Report CUCS-039-04, Computer Science Department, Columbia University, New York, NY, Sep 2004.

[5] D. Bryan and C. Jennings. A P2P Approach to SIP Registration. Internet Draft draft-bryan-sipping-p2p, Internet Engineering Task Force, Jan 2005. work in progress.

[6] D. Bryan and B. Lowekamp. Sosimple: a sip/simple based p2p voip and im system. White paper, Computer Science Department, College of William and Mary, Williamsburg, VA, Nov 2004.

[7] A. Gupta, B. Liskov, and R. Rodrigues. One hop lookups for peer-to-peer overlays. In *HotOS IX: The 9th workshop on hot topics in operating systems*, Lihue, Hawaii, USA, May 2003. USENIX.

[8] A. Johnston. SIP, P2P, and Internet Communications. Internet Draft draft-johnston-sipping-p2p-ipcom-00, Internet Engineering Task Force, Jan 2005. work in progress.

[9] J. Lennox, H. Schulzrinne, and J. Rosenberg. Common gateway interface for SIP. RFC 3050, Internet Engineering Task Force, Jan. 2001.

[10] P. Matthews and B. Poustchi. Industrial-Strength P2P SIP (requirements). Internet Draft draft-matthews-sipping-p2p-industrial-strength-00, Internet Engineering Task Force, Feb 2005. work in progress.

[11] D. Milojicic, V. Kalogeraki, R. M. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. technical report HPL-2002-57 20020315, Technical Publications Department, HP Labs Research Library, Mar. 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html.

[12] J. Rosenberg. Interactive connectivity establishment (ICE): a methodology for nettwork address translator (NAT) traversal for the session initiation protocol (SIP). Internet draft, Internet Engineering Task Force, July 2003. Work in progress.

[13] J. Rosenberg and H. Schulzrinne. Session initiation protocol (SIP): locating SIP servers. RFC 3263, Internet Engineering Task Force, June 2002.

[14] J. Rosenberg, H. 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.

[15] K. Singh and H. Schulzrinne. Peer-to-peer internet telephony using SIP. In *New York Metro Area Networking Workshop*, New York, NY, Sep 2004.

[16] K. Singh and H. Schulzrinne. Peer-to-peer Internet telephony using SIP. Technical Report CUCS-044-04, Department of Computer Science, Columbia University, New York, NY, Oct. 2004.

[17] K. Singh and H. Schulzrinne. SIPpeer: a session initiation protocol (SIP)-based peer-to-peer Internet telephony client adaptor. White paper, Computer Science Department, Columbia University, New York, NY, Jan 2005. http://www.cs.columbia.edu/ ˜kns10/publication/sip-p2p-design.pdf.

[18] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, San Diego, CA, USA, Aug 2001.