A Cooperative SIP Infrastructure for Highly Reliable Telecommunication Services

Ali Fessi, Heiko Niedermayer, Holger Kinkelin, Georg Carle
Computer Networks and Internet, University of Tübingen
Sand 13, 72076 Tübingen, Germany
{fessi|niedermayer|kinkelin|carle}@informatik.uni-tuebingen.de

ABSTRACT

Voice over IP (VoIP) has been a promising technology for several years. Although it has become very popular, the vision to substitute the "good old" Public Switched Telephone Network (PSTN) is still far from being true. Several issues, such as security, reliability and emergency calls are slowing down the integration of VoIP. Currently, there is a mixture of technologies where VoIP is often used for reducing costs, while still keeping the PSTN if a reliable communication is needed.

Peer-to-Peer (P2P) and overlay networks received a large attention in the research community in the last few years. P2P networks provide self-organization and scalability. Recent work involved P2P-based signaling for lookup services for Voice over IP (VoIP) communication based on the Session Initiation Protocol (SIP). The main motivation behind P2P-basd 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 DNS. However, pure P2P networks do not have only their advantages. They create also several security threats which are hard to solve. Some of them, for example, the Sybil attack, have been proven to be unsolvable without centralized authorities or an additional out-of-band mechanism. P2P-based SIP may also create a high potential for Spam over IP Telephony (SPIT).

In this paper, we present a hybrid solution for telecommunication networks that fills the gap between centralized (and therefore vulnerable to Denial-of-Service (DoS) attacks) telecommunication infrastructures and pure P2P-based telecommunication networks (which are vulnerable to SPIT, Sybil attacks, eclipse attacks, partition attacks, etc.). Our approach works with a SIP server under normal operation, while SIP User Agents (UAs) organize themselves into a P2P network. In the normal case, SIP UAs can profit from the lookup performance provided by the SIP server, which is usually better than the performance provided by the P2P network. In case of a service interruption of the infrastructure, the SIP network will function further due to the interconnection in the P2P network. Since the server and the P2P network cooperate in order to improve reliability, survivability, performance and security, we call this approach Cooperative SIP (CoSIP). We present our prototype implementation of CoSIP and discuss the potential for improvements compared to server-based SIP and P2P-based SIP.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: *distributed systems.*

General Terms

Management, Performance, Design, Reliability, Security

Keywords

SIP, Peer-To-Peer (P2P), P2PSIP, High Reliability, Survivability, Security Threats, Performance

1. INTRODUCTION

Although VoIP has been a promising solution for several years, the deployment is realized very slowly. Some of the major problems that are slowing down the deployment are the reliability of the telecommunication infrastructure and the security threats that show up due to the use of a common medium for the audio/video data and the legacy Internet applications such as email and web browsing. Such threats include viruses, worms, intruders and eavesdroppers. Security threats may be reduced using intrusion detection systems and appropriate authentication, integrity and encryption mechanisms. But there are still many issues to be solved here. Reliability is still a major issue as well. End users are used to the nearly 100% reliable PSTN. Therefore, people prefer to keep using the PSTN as long as VoIP does not provide sufficient reliability. One of the reasons for the lack of reliability is the complexity of the service infrastructure, which includes SIP registrars, SIP proxies, AAA servers, DNS servers, DHCP servers, routers and other network components. Even with a careful design, it comes often to inter-dependencies between these components. The interruption of the DNS service may, for example, often lead to the service interruption of SIP proxies and registrars. Network and service failures may propagate very quickly. Another reason for the lack of reliability is overload situations either due to Denial-of-Service (DoS) attacks or due to flash crowds. Some of the DoS attacks are not even caused on purpose and are just due to some SIP User Agents (UA) that send their registration requests too often to the SIP registrars [30].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IPTCOMM'07, July 19–20, 2007, New York, USA.
Copyright 2007 ACM …$5.00.

In June 2006, the German computer magazine "heise" reported a service interruption of several VoIP providers that lasted for about 2 hours [23]. During that time, it was not possible to make phone calls.

Therefore, there is a clear need for making VoIP services and networks more secure and more reliable. To achieve this goal, we take an approach that tries to combine the advantages of server-based and P2P-based SIP networks. SIP UAs join a P2P network and organize themselves into a Distributed Hash Table (DHT). However, unlike the P2PSIP approach that is currently discussed in the IETF, SIP UAs cooperate with the SIP server in order to achieve higher performance, higher reliability/availability and better security. Due to this cooperation, we call this approach Cooperative SIP (CoSIP).

The rest of this paper is organized as follows. Section 2 provides a short background on SIP and P2P networking. Section 3 describes our approach to improve the reliability of SIP services, with CoSIP. Section 4 outlines some implementation details of the CoSIP prototype. Section 5 discusses the expected improvement with respect to reliability/availability and security. Section 6 provides an overview of related work on this topic and outlines the differences to our approach. Finally, section 7 concludes this paper and outlines future work.

2. Background

2.1 Session Initiation Protocol (SIP)

SIP became very popular in the last few years. It is currently superseding the ITU H.323 protocol. Many commercial providers are already using it. It has been integrated into the 3GPP IP Multimedia Subsystem (IMS). If the reader is very familiar with SIP you may skip the rest of this section.

SIP is an application-layer signaling protocol that is used to establish, modify and terminate application sessions. It is an open standard defined by the IETF. The main standard document is [1]. A large number of related RFCs and Internet-Drafts have been published. An overview can be found on the SIP site of Columbia University [24] or on the Portal dedicated to SIP [26].

The most relevant components in a SIP network are User Agents (UA), proxies, redirect servers and registrars. A SIP UA may generate requests or processes requests from other UAs. In the former case, the UA is called User Agent Client (UAC). In the latter case, the UA is called a User Agent Server (UAS). A SIP registrar is a server that processes the registration of a UA to a certain location. The SIP registrar might use a location database and an Authentication, Authorization and Accounting (AAA) server in the backend to manage the registrations and the location information of its UAs. SIP proxies process and forward requests of SIP UAs. Among others, they work together with SIP registrars in order to establish connections between two different UAs. SIP registrars and proxies are often located together. The separation is only logical. Both of them are essential for the functionality of the SIP network.

Once the signaling for establishing the sessions has finished, the UAs can start to send media data to each other. The media data can flow directly between the two UAs except in some situations where a relay node is used, or all the media communication is routed via a single node, in some environments called a Security Border Controller (SBC). Figure 1 summarizes the main functionality of "traditional" SIP. More details can be found in the literature, for example, in [1], [27] or [28].

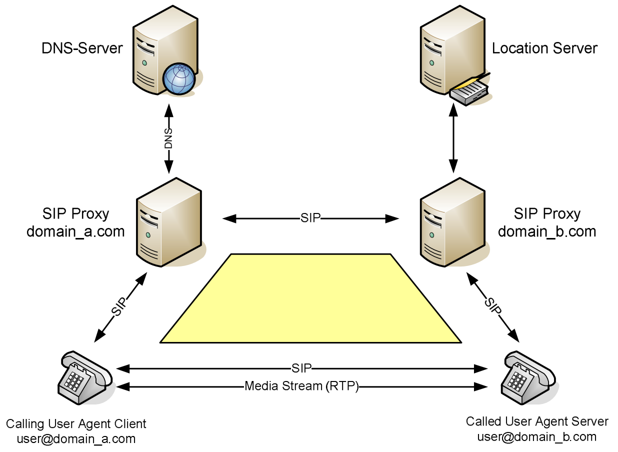

Figure 1: The SIP Trapezoid

2.2 The Peer-To-Peer (P2P) Paradigm

P2P networks are networks among equals. All peers using the system are also contributing to it. The organization and maintenance of the network is completely decentralized and carried out by all the peers.

P2P networks have become popular due to their vast application for file sharing purposes, in particular to overcome censorship and the potential shutdown by the prosecution.

The properties to achieve this are decentralization, self-organization and robustness. Decentralization means that there is no central entity that is required for the functioning of the network. Self-organization means that the network coordinates itself once running. Joins, leaves of single peers and other forms of interruptions are handled. Robustness means that the design is already dealing with an imperfect behavior of the network and especially of the peers. Peers are usually end-user systems that are neither stable nor permanently online or even running.

P2P networks are not only used for sharing files. BitTorrent deals with an efficient transfer of a file to many destinations. Skype [19] provides an instant messaging and VoIP service. Other networks provide application layer multicast and several other services.

One of the main advantages of P2P systems is the exploitation of resources at the edges of the network. End-user PCs that are idle most of the time provide the service. It is thus an easy way to provide a service without expensive dedicated servers or changes to the networking hardware.

P2P networks are also interesting with regard to reliability. Any individual peer may be less reliable than most servers. However, the network consists of many peers that automatically take over the work of a peer once that peer leaves. In a large-scale P2P network, the uptime may outperform any existing server infrastructure. The service rate for individual peers, however, may be lower. Reasons are losses and dynamics of the P2P network.

2.2.1 Distributed Hash Tables (DHTs)

A DHT is a structured P2P system that can be used for storing values in a manner similar to a standard hash table. The basic API of a DHT is therefore rather simple:

- *put(key,data)* is used to store the value "*data"* with the lookup key "*key"*. The system first starts a lookup for the key. All nodes responsible for the key "*key"* will be contacted. Those nodes will then store the value "*data*".
- *data=get(key)* is used to request the value at the key "*key"*. The value "*data*" will be returned. Here again, a lookup for a node that is responsible for "*key"* is performed and then the value returned.
- *remove(key)* is used to remove an outdated value.

Nodes need also methods to join and leave the network and to exchange routing information. Chord, CAN, Kademlia, and Pastry are examples for DHTs. DHTs are also called structured P2P networks. Unlike Unstructured P2P networks, like Skype, DHTs have the following property: it is possible to directly route to a key or node. There is no need for flooding. There are no false negatives. Under normal operation, a key is either found or does not exist.

Algorithms for routing and geometry for structured P2P networks are usually a compromise of state and lookup complexity. State refers to the size of the routing table. Lookup refers to the number of overlay hops to reach a specific key. Typically, if n is the number of nodes in the network, both complexities are $O(\log n)$ in average.

2.2.2 Security of P2P Networks

A major drawback of P2P solutions is their lack of security. P2P networks lack a central control and are, thus, prone to many attacks. Attacks can be on the data stored in the DHTs and on the network structure and connectivity. Without central control, hardly anything can be done to mitigate these problems. A fundamental result by Boyd [31] says that authentication can either be based on an already existing context or on a trusted third party which would be a central element. Using central elements for security-related tasks solves some of these problems. Skype, for example, uses central login servers for authentication [20].

Attacks on the data in a P2P network for Voice over IP could be used to redirect calls to a man-in-the-middle or the wrong person. This could be used as a setup for social attacks.

Another type of social attacks is Spam over Internet Telephony (SPIT). As a VoIP service needs in many use-cases to be open, we need to accept calls without any existing context. Identity management is a way to alleviate the problem, but it does not fully solve it.

Social elements can also be used to secure P2P networks. In case of VoIP services, ZPhone makes use of the interaction of human users to tackle man-in-the-middle attacks in case of initial contacts [36]. In case of a successful authentication the resulting shared-secret is used to identify the contact in future calls.

There are also privacy issues. Storing personal data on arbitrary machines enables targeted attacks against individual users. Private data may be open to the public and even normal non-authorized users may be able to profile and study the behavior of other users.

2.3 P2P-based SIP (P2PSIP)

We found a good brief description of what P2PSIP is about at [25]:

"The concept behind P2PSIP is to leverage the distributed nature of P2P to allow for distributed resource discovery in a SIP network, eliminating (or at least reducing) the need for centralized servers."

Different approaches for P2P-based SIP have been introduced in [5] and [6]. The idea is, instead of using SIP servers, proxies and registrars that manage the location of users and moderate session establishment requests, a P2P network is used. The P2P network consists of nodes where each node is a registrar. The node identifier (Node-ID) in the P2P network can be the hash value of its IP address, possibly combined with the port number. When Alice registers with its current location, the data "*Alice's current location"* with the lookup key *"Alice's URI"* is stored in the P2P network. Precisely, the node whose identifier is the closest to *hash(Alice's URI)* stores this data. When Bob needs to establish a session with Alice it sends a lookup message for *(Alice's URI)* which is propagated in the P2P network until the appropriate node responds to the request. Propagation might be routed recursively or iteratively. If Alice is currently not online, the lookup will fail or the registration information of Alice in the P2P network will be outdated.

The approaches for P2P-based SIP suggested by Bryan *et al.* [5] and by Singh *et al.* [6] have many things in common and few differences. Both of them use Chord as underlying P2P network. However, there is currently a consensus that P2PSIP can use other DHT geometries. Both approaches separate between node registrations and user registrations. Both suggest using SIP for providing the DHT functionality. The SIP REGISTER method is used by a node to "*join"* the DHT and by a user to insert its current location information into the DHT, (which is semantically equivalent to a *"put"* with the data *(user's URI, current location)*). The INVITE method is used to lookup location information of other SIP users in the DHT (which is semantically a*"get"* operation with the key *"user's URI"*).

The few differences between [5] and [6] are, e.g., use of iterative versus recursive routing of requests, the use of Super Nodes versus pure P2P network without Super nodes.

Since almost no security service can be provided in a pure P2P network, Bryan *et al.* [5] recommends the use of a public key infrastructure (PKI). Although authentication based on a PKI may solve many security problems in P2PSIP networks, a number of other attacks, such as attacks on the routing, are still possible.

Seedorf provided an analysis of security considerations for P2PSIP networks [11]. There are still may open issues. Security issues for P2PSIP will be summarized in section 5.2 in order to explain the difference to our approach CoSIP.

Recently, the IETF working group P2PSIP [7] has been chartered and standardization efforts have started. The following main goal of the working group is as follows (quoted from [7]):

"The Peer-to-Peer (P2P) Session Initiation Protocol working group (P2PSIP WG) is chartered to develop protocols and mechanisms for the use of the Session Initiation Protocol (SIP) in settings where the service of establishing and managing sessions is principally handled by a collection of intelligent endpoints, rather than centralized servers as in SIP as currently deployed."

Therefore, it is clear that the focus of the working group is to push the work towards the end nodes and to keep the required infrastructure as minimal as possible.

3. Cooperative SIP (CoSIP)

Instead of taking the P2P approach as by the P2PSIP IETF working group, we consider another approach. Starting from a "traditional" SIP infrastructure with dedicated SIP servers, SIP UAs connect additionally to each other in a P2P network, preferably a DHT. Lookup requests can be resolved either by the SIP server or by the DHT. We call this approach Cooperative SIP (CoSIP). The SIP server cooperates with the SIP UAs to manage user registrations and connection establishments. Compared to traditional SIP, CoSIP should improve the reliability and recoverability from catastrophic failures. Compared to P2PSIP, CoSIP should provide faster response, since the response time of the server is in average and under normal circumstances better than the response time of the DHT. Compared to P2PSIP, CoSIP should also improve security, since there is an entity in the network that is more intelligent than a "simple" Certificate Authority (CA) and can make sure that SIP UAs are behaving well in the P2P network.

In case the server is overloaded, under DoS attack or unreachable due to network failures, the DHT serves as a backup. This provides a significant improvement of the availability of the SIP service.

Centralized SIP infrastructures are vulnerable to DoS attacks. P2PSIP networks are hard to secure and are vulnerable to a number of attacks such as Sybil attacks, eclipse attacks, partition attacks, SPIT, etc. CoSIP is intended to fill the gap between the two extreme solutions in order to benefit from the advantages of both solutions.

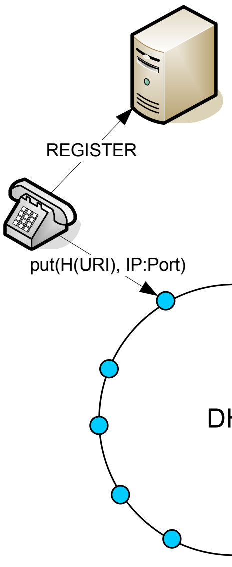

Figure 2: Registration of a SIP UA with CoSIP

Figure 2 and Figure 3 show the basic functionality of CoSIP. When Bob's UA registers to the SIP registrar, it also performs a "*put*" operation in the DHT:

put(hash(SIP-URI), IP:Port)

This data will be propagated in the DHT and can be used to resolve the current location of Bob. By the time, another peer Alice needs to reach Bob, there are two different ways to resolve the current location of Bob:

- Send an INVITE message to the SIP server
- Compute the hash of Bob's URI and send a "*get*" request to the peer that is supposed to be the closest peer to *hash(Bob'URI)* in Alice's routing table in the DHT.

A response from the DHT may take longer depending on the routing algorithm used for the DHT, the number of peers and the stability of the DHT. However, in case the server is down, or is undergoing an attack or an overload situation, the response from the DHT may be much more helpful.

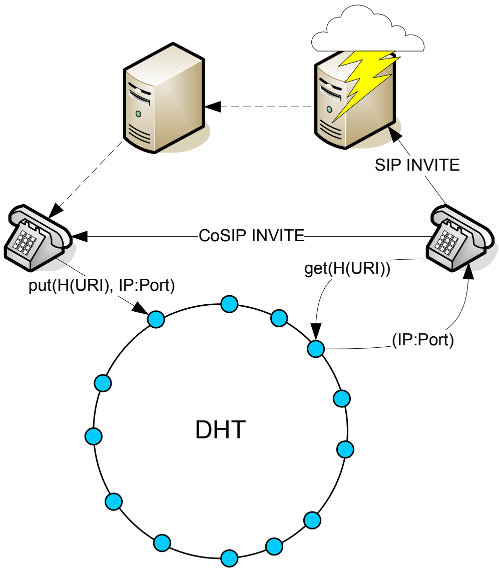

Figure 3: CoSIP Operation in case of a Server Failure

4. Prototype Implementation

4.1 Design Decisions

For a proof-of-concept implementation of CoSIP, we leveraged where possible other existing software components.

4.1.1 Support of Different SIP Clients Software

We implemented CoSIP as a local SIP proxy that processes the SIP signaling of one or more SIP UAs. Implementations of the SIP UA do not need to be aware of CoSIP. The SIP UA just needs to be configured with the CoSIP Proxy as an outbound proxy.

4.1.2 Choice of a DHT

As a DHT, we decided to use the Bamboo DHT [35]. Bamboo uses the concept of the Pastry DHT [2] with improved routing and maintenance algorithms in order to cope with high churn rates [13]. Additionally, Bamboo needs less traffic and bandwidth for maintenance [13]. For the communication between the CoSIP proxy and the DHT node, we use XML-RPC for performing "*put"* and "*get"* requests. The XML-RPC interface is provided by Bamboo to simplify the integration of Bamboo into other projects.

4.1.3 A Super-Node Approach

It is well known that due to the different capacities of peers in a P2P network, such as CPU and memory, there should be a differentiation between “powerful” peers, which are called the Super Nodes, which are connected to the DHT, and “weak” peers that are connected indirectly to the DHT via the Super Nodes. This contributes to the stabilization of the DHT. In CoSIP, we support the Super Node approach by having several SIP UAs connecting to a CoSIP proxy. The CoSIP proxy deals with a few SIP UAs and represents them in the P2P network. However, SIP UAs may also connect to several CoSIP proxies at a time in order to avoid that the CoSIP proxy becomes a single point of failure by itself.

4.1.4 Putting Everything Together

Based on the design decisions presented above, the architecture results into different software components communicating together as presented in Figure 4. The architecture can also be presented in a layered model with three layers (see Figure 5). The topmost UA layer uses the underlying CoSIP layer as a general signaling medium. The UA layer does not need to care about how the signaling is achieved by the CoSIP layer.

The CoSIP Layer uses traditional SIP signaling via the SIP server infrastructure and signaling with the DHT layer.

Finally, as a SIP server, we used the SIP Express Router (SER) [29]. The CoSIP proxy was implemented in Python. As mentioned above, our proof-of-concept implementation supports different SIP clients. In fact, we tested it with Kphone [32] as well as XLite [33].

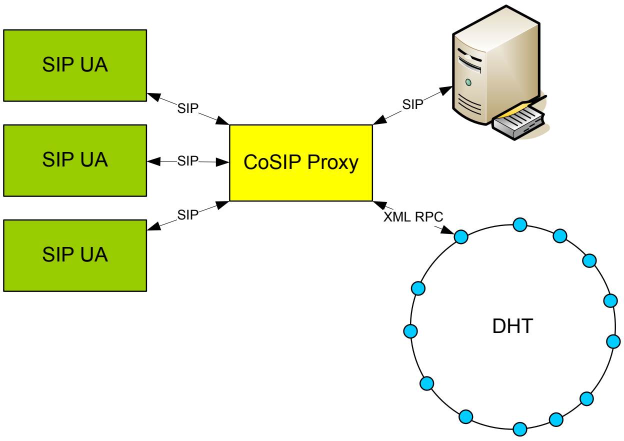

Figure 4: Architecture of a SIP Network with a CoSIP Proxy

4.1.5 Support of P2P-based SIP

As a side effect of our implementation of CoSIP, it was easy to extend the functionality of CoSIP to support a P2P mode without a SIP server. Therefore, our implementation of CoSIP can be also used as a P2P-based SIP solution[1]. We also performed some tests with CoSIP in a P2P mode with OpenDHT [18] and it worked well. Therefore, our implementation of CoSIP supports three different operation modes:

- DHT-only mode: here the CoSIP proxy makes “*put*”’s and “*get*”’s our of REGISTER and INVITE methods coming from UAs. No SIP server is required.
- Cooperative mode: this mode is the main goal behind CoSIP. The functionality is as described above.
- Server-only mode: here the CoSIP proxy forwards the SIP signaling between UAs and the SIP server without any modification.

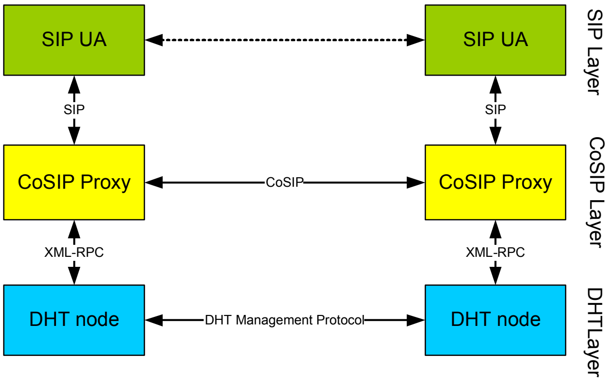

Figure 5: CoSIP Layered Model

4.2 Message Processing

In this section, we outline the processing of SIP messages by the CoSIP proxy in the cooperative operating mode.

4.2.1 UA Registration with CoSIP

If a UA sends an initial REGISTER message via the CoSIP Proxy to the SIP server, a new data structure that represents this UA is created. This data structure manages state machines that are used to keep track of the communication stages of the UA.

Next, the CoSIP Proxy modifies the REGISTER message and sends the modified message to the SIP Server. Modifications to the messages are needed to ensure that all SIP messages will be sent back via the CoSIP Proxy to the UA.

The CoSIP Proxy can be understood as a masquerading SIP proxy. Modifications are taken back in response messages from the server.

A timer for the server registration response will be started when the REGISTER message is sent for limiting the response time of the server to a certain timeout.

If the server does not respond in time, we assume that the server is down, or is in trouble, and register the UA to the DHT only. This means we “*put*” the Contact URI (IP and port) of the CoSIP Proxy to the DHT using the hashed SIP URI as a key.

put (hash(sip:user@sipdomain.org), IP:Port)

If this “*put*” succeeds, we send a “200 OK” SIP message to the UA, which means “registration successful” in this context.

If the server responds in time, we forward the SIP message to the UA and stop the timer. If the newly forwarded SIP message indicates a successful SIP registration, we also “*put”* the Contact

[1] Note here that this solution for P2P-based SIP does not conform to the drafts proposed for a P2PSIP solution in the IETF. One of the main differences is that the IETF goes for a SIP-based DHT management, while our solution re-uses other DHT protocols.

URI of the CoSIP Proxy to the DHT. So the UA is registered to the SIP server and the DHT.

4.2.2 Session Establishment with CoSIP

The processing of an INVITE message, sent by an UAC via the CoSIP Proxy, is straightforward. The CoSIP Proxy sends the message to the SIP server and sends a "100 Trying" message to the UAC in order to notify the UAS that its request is being processed.

Simultaneously, we try to resolve the Contact URI of the CoSIP Proxy of the called UAS using the DHT. So we perform a "*get*" with the hashed SIP URI of the invited UA as a key.

(IP:Port) = get(hash(sip:user@sipdomain.org))

We also start a timer, in order to limit the response time of the SIP server and the DHT to a certain timeout.

If the server responds first, the message is forwarded to the UA, the timer is stopped and we are done.

If the DHT responds first, e.g., when the server is down or is undergoing an overload situation, we use the data received from the DHT to send the INVITE message directly to the Contact URI of the peer CoSIP Proxy responsible for the sought UAS.

The peer CoSIP Proxy will forward the INVITE to the UAS. The UAS will respond back. The response passes through both CoSIP proxies to the UAC. Then we are done.

If neither server nor DHT respond in time, it is not possible to resolve the lookup request. In this case, the CoSIP proxy sends an error message "408 Request Timeout" to the UAC.

4.3 High Level State Machines

In order to handle the state machine of the CoSIP Proxy, we follow a modular approach, which results into a clear and less error-prone state machine. First, a CoSIP Proxy can deal with one or more UAC behind it by logically separating their state machines. Second, we assume that a UAC can have one "REGISTER" session and one or more "INVITE" sessions. Each session is represented by a state machine.

4.3.1 REGISTER Session State Machine

A new REGISTER session is created, when the UA sends a first REGISTER message via the CoSIP Proxy to the SIP server.

The CoSIP Proxy forwards the REGISTER message to the SIP server and starts a "SERVER TIMER" for limiting the server response time. The state of the session is set to PENDING.

If the server requires authentication, it will answer with a "401 Not Authorized" message. In this case, the state is set back to IDLE and we wait for the next REGISTER message that contains authentication data and start a new registration cycle.

If a "200 OK" message is received from the server in state PENDING, we set the state to REGISTERED and start the DHT registration process.

If the "SERVER TIMER" expires in state PENDING, we assume that the server is currently not reachable. The CoSIP Proxy starts a registration to the DHT only and moves to the state PENDING_DHT. If the DHT registration succeeds, the state of the REGISTER session is set to REGISTERED. Note however, that in our implementation, we have different flags that are set or reset in order to know whether the UA is currently registered to the server or to the DHT or both. If the registration fails, the state is reset to IDLE.

The registration of a UA to a SIP server expires within a certain period. Thus, the UA needs to renew the registration periodically. When the CoSIP Proxy receives a REGISTER message in state REGISTERED, a new registration cycle is started.

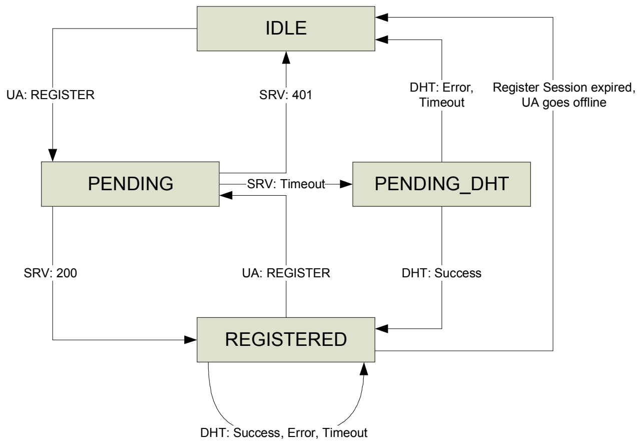

Figure 6: High-Level CoSIP REGISTER State Machine

4.3.2 INVITE Session State Machine

When the UAC sends an INVITE message, the CoSIP Proxy will create a new INVITE session. The INVITE session starts in state IDLE. The first transition to state PENDING_SRV_DHT is done directly after the CoSIP Proxy forwards the INVITE to the SIP server and starts the DHT lookup process. In order to limit the server response or DHT lookup time, the "RESOLVER TIMER" is started.

In the state PENDING_SRV_DHT, we either wait for a response from the server or for the DHT returning the Contact URI of the peer CoSIP Proxy.

Final messages from the server are, e.g., "180 Ringing" messages that indicate that the desired UAS received the INVITE message or a "404 Not Found" message, if the desired UAS is not registered to the server. These messages will set the state of the INVITE session to DONE.

The state DONE will also be reached, if the "RESOLVER TIMER" fires before the SIP server or the DHT returns any results. In this case, the CoSIP Proxy sends an error message to the UAC "404 Not Found".

When the DHT Lookup returns the desired Contact URI in state PENDING_SRV_DHT, the CoSIP Proxy sends immediately an INVITE message to the CoSIP Proxy of the desired UAS, starts the new "TIMER UAS" and changes its state to PENDING_SRV_UAS.

In state PENDING_SRV_UAS, the CoSIP Proxy waits for the first final answer that arrives either sent by the server or sent by the CoSIP Proxy of the UAS. A "180 Ringing" message either from the server from the UAS will set the INVITE session to state DONE.

If the TIMER UAS expires, this will also set the INVITE session to the state DONE.

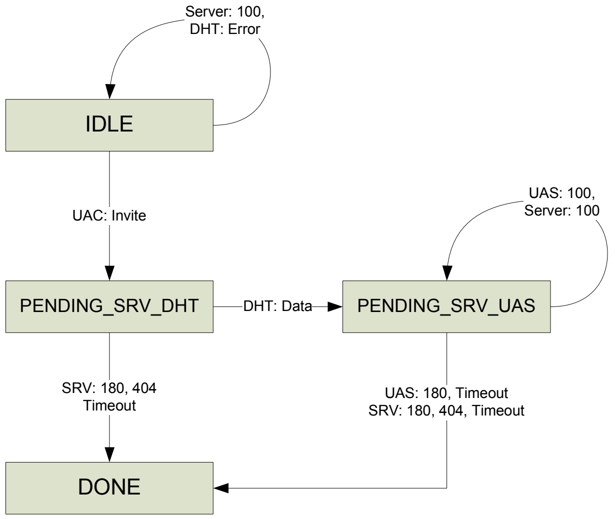

Figure 7: High-Level CoSIP INVITE State Machine

In order to be able to compare the performance of the DHT with the performance of the server, we added two more states to the INVITE session state machine (not shown in Figure 4). These states allow the CoSIP Proxy to wait for a second response when the signaling has been already successful either through the server or the DHT. The second response will enable the CoSIP proxy to build performance profiles of the server and the DHT, which can be used for future evaluation.

5. Evaluation

5.1 Expected Impact on Reliability

A service can only be used when it is up and running. In most cases, a SIP server should be enough to serve the requests for a domain. However, attacks, random failures or simply maintenance may stop the server and, thus, stop the service.

CoSIP adds a system that can replace the server when it is not responding. In fact, the DHT and the server operate in parallel. As long as one system is running, CoSIP is up. Its uptime is higher than the server's or the DHT's uptime.

One important benefit of a P2P approach is that existing computers that are not solely dedicated to the service can be used. Diversity is also an argument for the combination of server and DHT in CoSIP. Diverse systems are less prone to similar errors and correlated failures. Finally, DHTs cannot be shutdown with a DoS attack against some nodes.

The strength of the DHT approach comes from the use of many nodes and their means to deal with unreliable systems. Having several links to successor and neighbor nodes in the DHT contributes to the recovery from failure of several nodes at a time. Data is replicated on a set of nodes (replica sets) and is therefore not lost if a node quits the DHT suddenly.

In case of completely uncorrelated failures the probability for a failure of CoSIP is:

$$Failure_{CoSIP} = Failure_{Server} \bullet \left(Failure_{DHT\_node}\right)^n$$

with *Failure* being the probability for being down and *n* being the number of nodes in the CoSIP's DHT.

This is, of course, too optimistic. A peer in the DHT has only few links to a set of other peers. As this set of peers is smaller than the set of all nodes in the DHT, not all DHT nodes fully contribute to the uptime of the DHT service for a particular node.

A very useful effect on reliability is that maintenance downtimes can be completely bridged by CoSIP. CoSIP might be even an appropriate approach for providing emergency call services. In case of catastrophic failures, the SIP infrastructure may be not working, while the DHT will survive. A pure P2P approach is, however, not an appropriate approach for emergency calls due to the security issues in P2PSIP such as SPIT.

Concluding this section, the uptime of CoSIP can be expected to be significantly higher than the uptime of the pure server solution. CoSIP is based on existing resources and improves the resistance against attacks. Unless there is a failure in the infrastructure, server and DHT failures should be uncorrelated. Thus, a UA will experience hardly any of the failures and a high increase in reliability is achieved.

In our future work, we will also consider geometry and routing algorithms for the DHT in order to provider high connectivity in case of network failures. Imagine, for example, a small enterprise network that is using an external SIP server. Users in the small enterprise network should be able to communicate among each others even if the unlink to the outside is down. This will need the DHT routing and connectivity to be optimized depending on the underlying network.

5.2 Expected Impact on Security

Security mechanisms for CoSIP are not implemented yet. However, in this section, we outline the differences between the security services that can be provided with CoSIP compared to server-based SIP networks and P2P-based SIP networks.

5.2.1 Security Considerations in Server-based SIP Networks

For server-based SIP networks, it is possible to provide sufficient mechanisms for authenticating users, authorizing them, providing data integrity and confidentiality, providing privacy of the user activities, non-repudiation and accounting. Basically, with the presence of a central authority, which may be a SIP registrar/proxy possibly with a AAA server at the backend, it is possible to provide these basic security services. It is very common to use HTTP Digest for authenticating SIP UAs. In [1], the use of S/MIME certificates is also defined as an option.

Denial-of-Service attacks are still an open issue. Computational puzzles may help to reduce the problem. However, with sufficient resources, attackers may still be able to saturate the CPU resources of a SIP server and the bandwidth of the network.

Another issue that is currently a research topic, e.g., in [21], is Spam over IP Telephony (SPIT). Although Identity Management may help to fight it, SPIT is still a major problem that is slowing down the deployment of VoIP.

5.2.2 Security Considerations for P2P-based SIP Networks

As for P2P networks, they have several open security issues. The decentralization comes to the cost of less control of what is happening in the network. Decentralization pushes parts of the tasks to be performed to end hosts that may potentially be malicious. Security threats for DHTs were discussed in [9] and in [10]. Security threats for P2PSIP were discussed in [11]. Here, we summarize security threats in P2P-based SIP networks in order to be able to make a comparison with CoSIP:

- Impersonation: a malicious node may use a fake identity in the P2P network
- Forging messages: a malicious node may insert wrong data, e.g., location information, into the DHT in order to prevent honest users from being reached
- Sybil attack: a malicious node may join the P2P network with several different identities. Other nodes may believe that they are interacting with different nodes, while they are actually interacting with the same node. A malicious node performing a Sybil attack may be able to gain control over a large part of the P2P network.
- Eclipse attack: if an attacker controls a large fraction of the neighbors of honest nodes, it can "eclipse" the honest nodes and prevent correct overlay operation.
- Partition attacks: when a honest node contacts a malicious peer node in order to join the P2P network, the malicious node may provide the honest node with wrong information. The honest node will join a parallel P2P network.
- Invalid lookup: a malicious node may forward lookups to an invalid node, non-existing nodes or existing but a random node. Incorrect lookups will result into a waste of time and bandwidth, and may prohibit a honest peer to reach another peer.
- Propagating wrong routing tables: a malicious node may even propagate wrong routing information and force other honest node to forward requests incorrectly.
- Eavesdropping: a malicious node may passively record activities of other users/nodes in the network. A malicious node may build profiles for activities of its neighbors: when and where they are registered, when and whom they are calling.
- Deleting information from the DHT: a malicious node may delete data from the DHT to prohibit honest peers from being reached.
- DoS attacks: an attacker may launch a DoS attack against one or more nodes in the DHT to abuse their resources.

Current approaches for Node-ID assignment use a hash of the IP address possibly combined with the port number. Using different port numbers will obviously allow malicious nodes to have multiple identities on the same machine. It may also allow an attacker to join and leave the DHT several times until it finds a strategically good position to launch several attacks, for example, to get into the neighborhood of Bob's UA, monitor Bob's activities or launch an eclipse attack on his node.

5.2.3 Improving Security with CoSIP

Research efforts towards P2P-based SIP have reached the stage that it is currently well understood that managing security issues is not possible without a central authority. Bryan *et al.* [5] suggest the use of a Public Key Infrastructure (PKI) for authenticating users. Mutual authentication between UAs will allow for user authentication and message integrity. Message integrity can be achieved either by signing messages with a private key, or using a secure channel, e.g. based on TLS or IPSec, which is more efficient if communication between two peers is not restricted to a single message. Content integrity in the DHT can be achieved by having each peer signing the data that it stores in the DHT.

Those are already a few issues that can be solved by having centralized certificate authorities in the network. CoSIP may, however, provide better security since there is a SIP server in the network and that is more intelligent than a "simple" CA. The SIP server, or let's call it "CoSIP server", since it would be a SIP server adjusted to our needs for CoSIP, may manage the identities of the SIP UAs, such as public keys and Node-IDs in the P2P network. A possible way to manage the identities of the UAs could be, for instance, using the Security Assertion Markup Language (SAML) as suggested by [34].

Managing the identities of the UAs by the SIP server will prevent impersonation and Sybil attacks. By choosing secure random IDs, the server will also prevent malicious nodes from choosing strategically good IDs where they may be able to monitor activities of other users. The SIP server may contribute in providing secure bootstrapping to the P2P network. It might, for example, be itself a bootstrap node to the P2P network. This way, UAs can use the server as a bootstrap node when they register for the first time to the server. Secure bootstrapping will disallow partition attacks.

As for attacks on the routing, such as invalid lookup responses, and populating routing tables with wrong entries, these may be prohibited by validating responses from other peers from time to time by the server.

Note here that some approaches in the literature try to cope with attacks on routing in P2P networks, e.g., by providing different routes over different parallel overlays [12]. However, even if a honest node has detected that another node is not behaving correctly, it has no possibility to react on this or to prevent it in the future. This is simply because there is nothing like a network intrusion detection system (NIDS) in the P2P network that can cope with this. By having centralized authorities in the network, that are more intelligent than a "simple" CA, it should be possible to provide mechanisms for not only observing but also prohibiting these kinds of attacks on the routing.

Finally, as for DoS attacks, it is for further study whether an attacker may be able to launch an attack on the infrastructure as well as on the DHT in the same time. However, such an attack would certainly require a large amount of resources.

5.2.4 CoSIP Security: Conclusions

Security mechanisms for CoSIP have not been implemented yet. However, it is expected that having centralized authorities, such as CoSIP servers that are more intelligent than a simple CA, will cope with the typical security threats in a P2P network. SIP servers can manage the identities of the UAs and make sure they behave correctly. Of course, with scalability and self-organization as main goals in mind, centralized authorities could be considered a step backward. However, it might be not realistic to cope with all security threats in a P2P network without having these centralized authorities. A P2PSIP network may work well on a small scale. However, in a large scale P2PSIP network, it may be very hard to trace back what is going wrong, or who is misbehaving in the P2P network, particularly when the network is not working properly.

6. Related Work

As mentioned above, P2P and overlay networks have received much attention in the last few years. A lot of research effort has been investigated in this field. Mostly, the P2P network consists of end-hosts connecting together to provide the service to each other. Some realizations of the P2P concept are implemented on the carrier side as well. This means, a large number of servers cooperate and organize themselves in a P2P manner in order to provide a highly-scalable service infrastructures. Keromytis *et al.* [37] suggest the use of overlay networks in the infrastructure for providing robustness against DoS attacks. Their call their approach "SOS: Secure Overlay Networks". Andersen *et a.,* [15] discusses an approach for providing resilient paths or alternative paths between two end-hosts based on exchanging routing information on an overlay network and routing the data through the overlay. Their approach, called "Resilient Overlay Networks" (RON), is supposed to recover faster from routing failures than wide-area routing protocols, such as BGP. However, unlike our approach, the intelligence and management of the overlay network is also pushed to the end-hosts and applications.

Skype has become very popular in the last few years as well. However, Skype is a proprietary protocol. Its security is based on its "closed source" status, which follows the "security by obscurity" approach. "Security by obscurity" is rather controversial and has failed several times before.

CoDNS [38] may be the approach most closely to ours. In fact, the name CoSIP is also inspired from CoDNS. Using CoDNS, DNS clients send a DNS lookup to another DNS client in another domain when they notice that their DNS server is encountering problems. However, the authors of CoDNS do not cope sufficiently with security issues. They admit that CoDNS can not cope with falsified responses and that there is no general solution for this issue. Unlike CoDNS, CoSIP copes with most of the security issues by having the SIP server controlling the activities of the peers from time to time and making sure that they are behaving correctly. However, we admit that CoSIP is currently a single-domain approach while CoDNS relies on the cooperation of the different DNS clients from different domains. It is clear cooperation between different domains raises more security and trust issues. We will need to deal with theses issues in our future work on CoSIP.

As for P2PSIP, Bryan *et al.* [5] discusses different application scenarios in an Internet Draft. Failure recovery is mentioned as one of the use-cases. However, the P2P network is again on the infrastructure side. We are currently not aware of a similar approach for SIP to ours where both a P2P-based SIP network and a SIP server cooperate in order to provide better performance, better reliability/survivability and security in the same time.

7. Conclusions and Future Work

In this paper, we presented Cooperative SIP (CoSIP), which is our approach to cope with the reliability/survivability problem of SIP infrastructures. Unlike the P2PSIP approach, CoSIP is an approach where both the SIP server and a DHT cooperate together in order to resolve requests from UAs. CoSIP benefits from the advantages of both server-based and P2P-based SIP networking and should be able to improve the performance, reliability, survivability and security. CoSIP is more likely to survive catastrophic network failures and provides better security than P2PSIP. CoSIP might be even an appropriate approach for providing emergency call services since it provides high reliability and security simultaneously.

Furthermore, we outlined the implementation of a first prototype of CoSIP including state machines for registering users and resolving INVITE requests. As a side effect of our implementation efforts, we gained also a P2P-style SIP implementation. The CoSIP proxy can be just configured to operate in P2P mode, regular server-only mode or cooperative mode.

One of the limitations of CoSIP is that it is currently designed for a single domain. Cooperation between different domains in a CoSIP style will be investigated in our future work. Further future work will also include the performance evaluation that we have just started on PlanetLab [17]. We will try to quantify CoSIP's reliability and recoverability from failures. We will also work on integrating security mechanisms for CoSIP as discussed above.

Another issue is the impact of local network failures on the connectivity of the UA to the P2P network. Improving connectivity in the DHT by making it aware of the underlying network will require modifications in the DHT geometry and routing algorithms.

Finally, we will investigate different degrees of cooperation between server-based and a P2P-based SIP networks. CoSIP can be seen as a concept between pure P2P networks and centralized SIP networks. Therefore, one might think of different application scenarios where the work may be done closer to the server side or the P2P network.

8. REFERENCES

[1] J. Rosenberg, et al. "The Session Initiation Protocol". IETF RFC3261. June 2002.

[2] P. Druschel, A. Rowstron. "Storage management and caching in PAST, a large-scale, persistent P2P storage utility". Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001), Lake Louise, AB, Canada. October 2001.

[3] S. Ratnasamy, et al. "A Scalable Content-Addressable Network". Proceedings of the ACM SIGCOMM 2001, San Diego, CA, USA, August 2001.

[4] I. Stoica et al. "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications". Proceedings of the ACM SIGCOMM 2001, San Diego, CA, USA, August 2001.

[5] D. Bryan, et al. "SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System", International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA), June 2005

[6] K. Singh and H. Schulzrinne. "Peer-to-Peer Internet Telephony using SIP". Columbia University Technical Report CUCS-044-04. October 2004.

[7] "Peer-to-Peer Session Initiation Protocol (p2psip)" IETF Working Group, http://www.ietf.org/html.charters/p2psip-charter.html

[8] K. Gummadi, et al. "The Impact of DHT Routing Geometry on Resilience and Proximity". Proceedings of the ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.

[9] E. Sit and R. Morris, “Security Considerations for Peer-to-peer Distributed Hash Tables”. International Workshop on Peer to Peer Systems. March 2002.

[10] L. Divac-Krnic, R. Ackermann. “Security-Related Issues in Peer-to-Peer Networks”. Peer-to-Peer Systems and Applications 2005: 529-545.

[11] J. Seedorf. ”Security Challenges for Peer-to-Peer SIP.” IEEE Network, Volume 20, Issue 5. Sept.-Oct. 2006.

[12] H. Johansen, et al. “Fireflies: Scalable Support for Intrusion-Tolerant Network Overlays”. Proceedings of Eurosys 2006.

[13] S. Rhea, et al. “Handling Churn in a DHT”. Proceedings of the USENIX Annual Technical Conference. June 2004.

[14] V. Ramasubramanian, E. G. Sirer, “The Design and Implementation of a Next Generation Name Service for the Internet”. Proceedings of the ACM SIGCOMM '04 Conference on Communications Architectures and Protocols. August 2004.

[15] D. G. Andersen, et al. “Resilient Overlay Networks”. Proceedings of the 18th ACM SOSP, Banff, Canada. October 2001.

[16] S. Rieche, et al. “Reliability and Load Balancing in Distributed Hash Tables”, Peer-to-Peer Systems and Applications 2005: 119-135.

[17] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services, http://www.planet-lab.org/

[18] S. Rhea, et al. “OpenDHT: A Public DHT Service and Its Uses”. Proceedings of ACM SIGCOMM 2005, August 2005.

[19] Skype. http://www.skype.com/

[20] T. Berson. “Skype Security Evaluation”. October 2005.

[21] “Spam over Internet Telephony Detection Service”, EU SPIDER project. http://www.fokus.fraunhofer.de/bereichsseiten/projekte/SPIDER/

[22] “Low Cost Tools for Secure and Highly Available VoIP Communication Services”, EU SNOCER project, http://www.snocer.org/

[23] “VoIP-Störung bei United Internet”, Press Article, July 2006, http://www.heise.de/newsticker/meldung/75382

[24] SIP Web site at Columbia University, http://www.cs.columbia.edu/sip/

[25] “P2P SIP”, P2PSIP Web Site at Columbia University, http://www.p2psip.org/

[26] “Tech-invite”, a Portal devoted to SIP and surrounding technologies, http://www.tech-invite.com/Ti-sip-IDs-SIP.html

[27] G. Camarillo, "SIP Demystified", McGraw-Hill, 2002.

[28] A. Johnston, "Understanding the Session Initiation Protocol", ARTECH, 2001.

[29] “SIP Express Router”, http://www.iptel.org/ser

[30] A. Hoffmann. “Securing Large Scale VoIP Infrastructures”. Talk in 3rd VoIP Security Workshop Berlin 2006.

[31] C. Boyd. ”Security Architecture Using Formal Methods”. IEEE Journal on Selected Areas in Communications, 1993; 694−701.

[32] KPhone, a SIP UA for Linux, http://sourceforge.net/projects/kphone

[33] XLite, a SIP UA, http://www.xten.com/

[34] H. Tschofenig, et al. “SIP SAML Profile and Binding”. IETF Internet Draft (work-in-progress), October 2006.

[35] The Bamboo Distributed Hash Table, http://bamboo-dht.org/

[36] “The ZPhone Project”, http://zfoneproject.com/

[37] A. D. Keromytis et al. “SOS: Secure Overlay Services”, Proceedings of SIGCOMM 2002, Pittsburg.

[38] K. Park, et al. “CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups”. Proceedings of the Sixth Symposium on Operating Systems Design and Implementation(OSDI '04).