Decentralized supplementary services for Voice-over-IP telephony

Christoph Spleiß and Gerald Kunzmann

Technische Universität München
80333 Munich, Germany
{christoph.spleiss,gerald.kunzmann}@tum.de

Abstract. As current Voice-over-IP (VoIP) systems encourage a direct communication between the callees they are similar in design to peer-to-peer (P2P) approaches. Therefore, we introduce a framework to build *distributed* supplementary services for VoIP. Some of these services like completion of calls on no reply or holding are already commonly established in public switched telephone networks or realized in centralized VoIP servers like Asterisk [1]. In order to foster the completely decentralization of these features we suggest corresponding services implemented on top of a structured P2P network. Using a resource-based approach new features can easily be deployed and announced.

Key words: Voice over IP, telephony services, Peer-to-Peer, Framework

1 Introduction

Supplementary services for telephony are a fixed part in today's communication world. Everybody uses features like call forwarding, completion of calls on non-reply or holding. These features are provided by a centralized local or private branch exchange. Due to centralization this approach has many disadvantages. Administration and maintenance of an exchange are very expensive as usually every terminal needs a separate line. Furthermore, in case of a breakdown every connected client is affected and the whole system will be unavailable.

In order to save costs and provide resilience, peer-to-peer technology (P2P) provides a reasonable approach to solve these problems. In this paper we describe how we can realize decentralized supplementary telephone services using a structured P2P system in a Voice-over-IP (VoIP) environment. We developed an extensible framework in which we can add supplementary telephone services to common VoIP systems like Skype or SIP. Furthermore, this approach is platform independent and can be used both on stationary and mobile devices.

The paper is organized as follows. Section 2 gives an introduction to P2P technology and a more detailed explanation of the structured P2P system CHORD. We also have a look at previous work done on a resource management framework (RMF) for P2P systems used in our approach. Afterwards we give an overview over supplementary telephony services in current public switched telephone networks which will be realized in our overlay network. Section 3 describes how these services can be realized using CHORD and the RMF. Conclusions and directions for future work are presented in section 4.

1.1 P2P systems

Compared to a client/server architecture like FTP, a pure P2P system hasn't got any central instance like a server. Every participant in such an environment acts both as a server providing data and storage, as well as a client, searching for content. Information is exchanged directly between the peers. Because every peer is sharing its own resources like storage space, a P2P system can store much more data than a single server. Also the data availability is higher than in a client/server system. In case of a breakdown of a few peers, only the data stored on them are lost. Even this risk can be minimized by using replication, that is storing the information on more than one node.

But the equality of all peers is not necessary in order to qualify a system as P2P. Napster for example uses a centralized indexing server to store references on the content of the whole P2P system and to provide a powerful lookup service. Every user willing to share data registers itself to the lookup-server and transmits a list of available files. Peers looking for a certain content gain the network address of peers with the desired content from the lookup-server. The file itself is directly transferred between the two peers. However, in case of lookup-servers' breakdown, the whole system isn't available any more.

Due to resilience problems of centralized P2P architectures the Gnutella protocol was developed which provides a fully decentralized architecture. Every peer holds a few connections to other peers. In order to search for content, queries are forwarded to all connected neighbours. Until the first match every peer forwards this query. To limit the packets' lifetime, a TTL counter is decremented at every hop. If the query is successful, the information is transmitted using the reverse path of the query.

Because of the high bandwidth consumption of Gnutella hierarchical topologies like Fasttrack were developed. As a hybrid P2P system, Fasttrack provides so called supernodes, which form a Gnutella network, and so called leafnodes, which are connected to one of the supernodes. Fasttrack provides the high resilience of Gnutella combined with the bandwidth efficiency of Napster.

All P2P architectures described above belong to the so called unstructured P2P systems. The next generation in P2P systems are structured P2P systems [2], [3] like CHORD [6] which are based on the principle of distributed hash tables (DHT) to provide efficient routing mechanisms. CHORD has got a circular structure whereby specific node features are used to calculate a n-bit hash value m for every node. According to the hash value the nodes are arranged in a circle. The same hashing function is used to provide a key k for the content stored in the network. Every key k is now assigned to the first node whose identifier is greater than or equal to k. To provide routing and lookup mechanisms every peer maintains a routing table with entries pointing to at least its direct successor and predecessor in the ring. To make lookups much faster, CHORD also maintains an additional routing table, the so-called *finger table*. The finger table has got n entries, where each entry i points to the node first that succeeds m by at least 2^{i-1}. These fingers are used to provide shortcuts through the ring during the lookup process. Because the distance to a key k will be at least halved in every step, each query can be resolved in $\mathcal{O}(logN)$ hops, where N is the number of participating nodes.

1.2 Resource Management Framework for P2P networks

In order to define a structure to our content stored in the P2P network we use the idea of metadata representation shown in [4]. Every content needs certain properties to build a defined structure. The mandatory property of a content is the `type` descriptor as it determines the semantic interpretation of further properties. In addition, every content is identified by a unique identifier (`UID`). This `UID` is provided by our DHT function. Links between contents are created by storing the UID of another content in the `linkUID` property.

1.3 Supplementary telephony services in public switched telephone networks (PSTN)

Common communication carriers provide their customers a great variety of supplementary telephony services [5]. These services are realized on centralized exchanges with all disadvantages mentioned before. These supplementary services are for example:

- **Calling Line Identification Presentation/Restriction** is a telephony network service that transmits the caller's telephone number to the called party or blocks the transmission. The state for the indication is set by the caller and evaluated in the telephone exchange. The exchange is responsible for proceeding the correct information to the called party.
- **Completion of Calls to Busy Subscribers/on No Reply** is a telephony network service that allows the calling party to automate the call origination in the telephone exchange for the unavailable subscriber. Because the exchange as a centralized instance knows the state of every participant, it can initiate the callback if the desired destination is available again.
- **Call Waiting** is a feature that a subscriber gets a audio signal during an active connection signaling that another party wants to establish a connection. The called party doesn't get the congestion signal but the ringing signal. The party informed by the audio signal can decide if the waiting call should be rejected, accepted by terminating the current call or accepted by putting the current call on hold.
- **Call forwarding** is a feature where an incoming phone call is redirected to another party. The diversion can occur immediately, after a certain time or if the called party is busy.
- **Three-Party-Conference** is a feature whereby a party can establish a conference between itself and two other parties. Afterwards every party can

hear the others at the same time. This feature is normally controlled by the telephone exchange but can also be provided by the end device of the originator. In the last case the originator needs at least two phone lines.
- **Hold** is a telephony network feature where during a call a party can put the remote party on hold. The party on hold is receiving music on hold from the exchange. The initiated party has the possibility to switch to another phone and resume the connection. This procedure has to be finished within a certain time, otherwise the connection to the party on hold will be terminated by the exchange.

Some of the features shown above are to be realized with a CHORD P2P System. In the next section we will see which features can be realized in the P2P network itself and which have to be realized in the end devices.

2 Implementing supplementary services using P2P technology

Our goal is to provide an overlay network for supplementary telephony services which integrates already common VoIP solutions like Skype or SIP. We do not need to develop an own channel for voice transmission as the overlay is an additional feature to existing communication solutions. This project shall be realized using the basic functions provided by a P2P network: storing and finding of information. In our implementation we used CHORD as the underlying P2P network. Also any other structured P2P network is applicable. The supplementary services are provided on top of this protocol. However, it is not possible to realize all services shown above solely in the P2P network. Thus, calling line identification presentation is a service which can only be provided by the end device, because the communication takes place on a direct P2P connection without an exchange in between. Also three party conference has to be realized directly in the end device as there's no exchange to merge the speech signals. In addition, services are expected to work cross platform, i.e. both mobile and stationary end devices should be able to use services provided by the overlay.

2.1 Searching subscribers

Before communication can start we have to ensure that subscribers can find each other. As interaction between different VoIP systems is desired we cannot directly use Skype or SIP URIs but we have to develop our own lookup process inside the P2P network. For these purposes every subscriber has to enter individual-related data like given name, family name and email address into his end device. Also a dialplan has to be created which will be discussed in the next subsection. The end device is computing five hash values, one used as the UID for the content type called *contact sheet*, and the others used as the UID for four contents of type *profile*. The profiles contain the full name of the subscriber and a linkUID to the contact sheet. They are hashed with the subscriber's given name, family

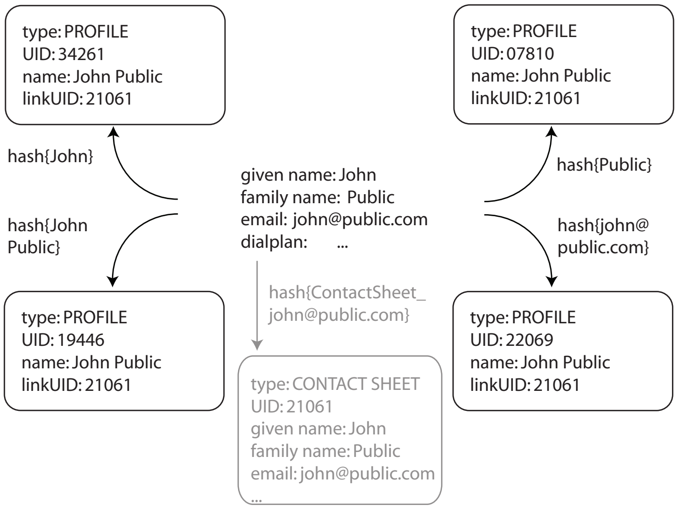
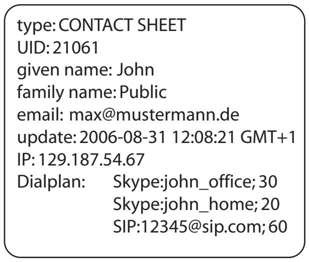


Fig. 1. Four profiles generated out of the users data

name, both names together and email address (Fig. 1) and then published in the P2P network. We are using personal information about the subscriber to hash our profiles so that subscribers searching for others can enter something they know about the desired party like the family name into their end device. After calculating the UID out of the search string, this value is committed to the P2P network which will return all profiles matching the search key. If the desired party has been returned and selected by the user, the contact sheet referred in the linkUID-field of the profile is obtained. As the contact sheet contains the dialplan a connection to the other party can be established according to the dialplan.

2.2 Dialplan and call forwarding

The dialplan is the most important component necessary during the origination. Every subscriber needs to enter a set of rules how he wants to be accessible (via Skype, SIP or others) and in which order. Also a timeout value for every dialplan rule is necessary. If a rule isn't successful within the given timeout value, the next rule will be attempted. The dialplan length is arbitrary as it can be realized as a linked list. This dialplan is saved inside the contact sheet together with further user information such as the users IP-address and a time stamp of the last update (Fig. 2). The UID value for the contact sheet is determined out of the user's email address with the prefix "ContactSheet_". This salted hash is necessary as we already use the hash obtained by the email address for a profile's UID. Based on the fact than an email address is unique there shouldn't be any collisions between contact sheets. After the reception of a contact sheet the



```
type: CONTACT SHEET
UID: 21061
given name: John
family name: Public
email: max@mustermann.de
update: 2006-08-31 12:08:21 GMT+1
IP: 129.187.54.67
Dialplan:    Skype:john_office; 30
             Skype:john_home; 20
             SIP:12345@sip.com; 60
```

Fig. 2. Contact sheet with dialplan attached

recipient tries to establish a connection according to the dialplan. The dialplan is subjected to be changed at any time by the owner. Changes occur immediately as the contact sheet is reloaded at every connection attempt. If a speech system isn't supported by the caller the next eligible rule is taken. This is also the way how we realized the supplementary feature call forwarding. After certain time the next dialplan rule is chosen. In order to provide immediate call forwarding the user is able to select zero as timeout value. This rule will be skipped.

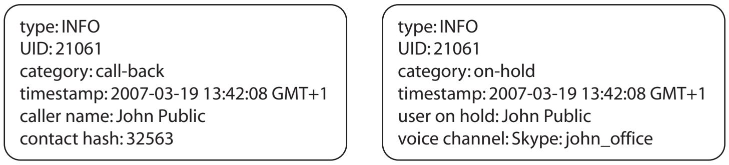
2.3 Completion of calls on no reply

This supplementary feature is realized with content of the so called type `info profile`. If no rule in the dialplan of a subscriber was successful, the caller has the opportunity to send a call-back request to the unavailable subscriber. For this purpose the caller generates a info profile of the category `call-back` which has the same UID as the contact sheet of the unavailable person and publishes it in the P2P network (Fig. 3, left). Furthermore the info profile contains a time stamp, the caller's name and a linkUID to it's contact sheet in order to initiate the call-back.

Every subscriber connected to the overlay periodically checks if there are any info profiles existing for his UID. This is possible because the info profile's UID is known and can be queried. If a subscriber receives a call-back info profile he can choose if he wants to establish the call-back or ignore the request. He can also use multiple devices to receive this info profile as the application is platform independent. In both cases the info profile is deleted after it has been received to avoid multiple notifications of the same event. The major difference between the same service provided by telephone exchanges is that the call-back is not established by the exchange but by the other party.

2.4 Putting calls on hold

Like the supplementary service completion of calls on no reply the service putting calls on hold is realized with info profiles. Subscribers having an active connection can put the other party on hold in order to resume the call from another location.



type: INFO
UID: 21061
category: call-back
timestamp: 2007-03-19 13:42:08 GMT+1
caller name: John Public
contact hash: 32563

type: INFO
UID: 21061
category: on-hold
timestamp: 2007-03-19 13:42:08 GMT+1
user on hold: John Public
voice channel: Skype: john_office

Fig. 3. Call-back and on-hold info profile

To achieve this the hold initiating subscriber generates an info profile of category on-hold. This info profile also contains a time stamp, the UID of the initiators own contact sheet, the full name of the subscriber which is put on hold and the last active voice channel (Fig. 3, right). After this, the info profile is published in the P2P network and the active communication is disconnected.

The initiating subscriber has now again the opportunity to change its end device, e.g. from a PC to a PDA. As soon as the new device is connected to the P2P network it checks for relevant info profiles. If an info profile with on-hold information is received, the user can decide if he wants to reestablish the former connection with the same voice channel used before. The info profile is deleted after reception.

3 Summary and outlook

We have shown that it is possible to provide supplementary telephony services without the need of a centralized instance using only P2P technology. In addition, we have developed a Java-client connected to Skype via the SkypeAPI and to a CHORD P2P network which provides all services shown above. Although Skype is proprietary we have chosen this speech system as it provides an easy to use API and sufficient possibilities to control and monitor call states. Anyway we do not use the supplementary services offered by Skype as other systems may not support them and we want to make the services usable in all imlemented speech systems. This makes is necessary to implement another speech system like SIP to utilize the possibilities of the framework.

Using the resource management framework new features can easily be implemented by simply defining new info profiles for new supplementary services. Possible ideas are for example voice gateways which can convert the speech signal from one VoIP system to another, i.e. from Skype to SIP. Info profiles are used to propagate the existence of these gateway and to announce their facilities. Other services are enhanced messaging services which collect the info profiles of registered subscribers and inform them about events via email or SMS. Also, reminder services and distributed calendars are easy to implement.

In order to protect the overlay from malpractice some security features still need to be integrated. We suggest a PKI based approach as it must be guaranteed that nobody is inserting malicious info profiles or removes info profiles and contact sheets without permission.

References

1. Asterisk: An open source PBX and telephony toolkit, Digium. http://www.asterisk.org
2. Eberspächer, J., Schollmeier, R., Zöls, S., Kunzmann, G.: Structured P2P Networks in Mobile and Fixed Environments. Proceedings of HET-NETs '04 International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks 2004. Ilkley, U.K.
3. Kellerer, W., Kunzmann, G., Schollmeier, R., Zöls, S.: Structured Peer-to-Peer Systems for Telecommunications and Mobile Environments. AEÜ - International Journal of Electronics and Communications 2006.
4. Friese, T., Freisleben, B., Rusitschka, S., Southall, A.: A Framework for Resource Management in Peer-to-Peer Networks. Department of Mathematics and Computer Science, University of Marburg.
5. Siemens Communication Lexicon, Siemens Enterprise Communications GmbH & Co. KG. http://networks.siemens.com/communications/lexicon_en/index.htm
6. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proceedings of the 2001 SIGCOMM conference. ACM Press New York, NY, USA