Peer-to-Peer SIP features to eliminate a SIP sign-up process

Toshiya Okabe
System Platforms Research Laboratories,
NEC Corporation, Japan
t-okabe@bx.jp.nec.com

Henning Schulzrinne,
Department of Computer Science,
Columbia University, USA
hgs@cs.columbia.edu

Abstract—We propose two features for Peer-to-Peer SIP in order to eliminate a sign-up process so that users can use the SIP user agent software right after installation. One is a locally generated SIP-URI which eliminates the sign-up process. Another is a feature to exchange contact addresses. It helps users to create new entries in the local address book without typing SIP-URIs. We implemented these features on SIP-Communicator with a peer-to-peer client protocol to work with a Peer-to-Peer SIP overlay network.

Index Terms—Community management, Internet telephony, Peer-to-peer computing, Real-time multimedia services

I. INTRODUCTION

In the Session Initiation Protocol (SIP) [1], a SIP registrar is responsible for registering association of an AOR (Address Of Record) and one or more Contact URIs (Uniform Resource Identifiers). The AOR is a logical identity of a user, whereas the Contact URI indicates a specific host where the user can be reached. The association is called location information and is provided by SIP user agents in a form of a REGISTER message. SIP proxy servers use this location information to route SIP request messages to the current location of the recipient.

The IETF P2PSIP Working Group is working on standard protocols [2] which replace SIP registrar/proxy servers with a peer-to-peer overlay network. In the peer-to-peer overlay network, maintenance of the location information is not centralized but is instead distributed among the peers in the overlay. Another similar approach which combines SIP and a peer-to-peer overlay is reported in [13], where the authors implement a peer-to-peer overlay algorithm using SIP messages without an additional protocol for a peer-to-peer overlay.

This architecture enables a lot of application scenarios. In [12], these scenarios are put into three categories: “the global internet environments,” “the environments with limited connectivity to the Internet” and “the managed, private network environments.” In the category of “the environments with limited connectivity to the Internet”, a scenario of ad-hoc and ephemeral groups is described as “groups of individuals meeting together have need for collaborative communications systems that are ephemeral in nature, have minimum (ideally zero) configuration, and do not depend on connectivity to the Internet.”

Under the environments such as the ad-hoc and ephemeral groups, it will be desirable to eliminate the sign-up process so that anyone can use right after installing the software or turning on the device without any efforts prior to its use.

In this paper, two features for Peer-to-Peer SIP (P2PSIP) overlay networks are proposed. One is a locally generated SIP-URI which eliminates the need for the sign-up process and the servers for this purpose. Since transient networks and ad-hoc networks do not always have stable connection to servers on the Internet, we think it is desirable that users can use the software with as little preparation as possible.

In this paper, we propose a solution to eliminate the sign-up process. Users will be able to use the SIP user agent software right after installation.

II. PROBLEM STATEMENT

Usually a user of a SIP user agent software has to perform three steps prior to making or receiving a call.

Step 1. Install the software.
Step 2. Enroll (sign-up on a website or paperwork)
Step 3. Configure the software

After these three steps, registration of the location information is done by the software automatically.

P2PSIP eliminates the need for dedicated SIP proxy and registrar servers and configuration work related to these servers. However, users have to perform a sign-up process; it is usually paperwork or web registration, where a SIP-URI is chosen by the user or it is assigned by the operator of the P2PSIP network. Since transient networks and ad-hoc networks do not always have stable connection to servers on the Internet, we think it is desirable that users can use the software with as little preparation as possible.

In this paper, we propose a solution to eliminate the sign-up process. Users will be able to use the SIP user agent software right after installation.

III. IDEAS

A. Mechanism to exchange contacts

Once you put someone's addresses into your address book, you will notice that it is not needed to explicitly type or see the address even when you make a call or receive a call to/from others.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.
In the real world, there is a well-known protocol for cell phone users to exchange their contact addresses (Fig. 1). A user, Alice, tells his telephone number to her friend, Bob (1). Bob makes a call to Alice but soon hangs up before she picks the call (2). Then Alice calls back using the call history but hangs up again before Bob picks up the call (3). Through this series of the transaction, both of them have each other’s contact address in their call history lists. Then they register them in their local address books. After that, they can edit the newly created entry ((4) and (5)) and never mind the actual telephone number.

We propose a rendezvous address for exchanging contact addresses. In the protocol above, a telephone number is not used for its original purpose but used for exchanging contact addresses. For this specific purpose, it does not need to be a telephone number but can be an address which is globally unique only while the exchange is being performed. In other words, the lifetime can be shorter and its length, the number of digits, can be shorter. We propose the new address for this purpose. The address works as a substitute of the telephone number. It is shorter in its length (for example, a four-digit number) and easier to use although the lifetime is shorter.

**B. Locally generated SIP-URI**

Instead of an address assigned by a central authority, a locally generated address can be used as a SIP-URI in order to remove the sign-up process. The address has to be globally unique. There are two solutions.

1. To use an existing global unique address like an e-mail address,
2. To generate a random number or a random character string in the local host and make the address from it

In case of using existing addresses, users can choose their familiar ones although it has to be input by users into the software. A problem will arise when P2PSIP networks connect and interoperate with the other networks such as a standard SIP network.

With regard to the second option, the address must be long enough to make the collision in the namespace small and negligible. For this purpose, RFC 4122 [3] defines UUID (Universally Unique IDentifier).

The UUID enables distributed systems to uniquely identify information without central coordination. The identifier is created by using the current time, a pseudo-random number and has a format like

4def145b-e510-4dd4-a3a9-25008bc0ba8a

By using the UUID as a user part of the SIP-URI coupled with a preconfigured domain name, which can make separate name spaces if it is needed, each user agent can have its SIP-URI without the help of the central coordination.

The address is created at the time when the software runs for the first time and is used until it is explicitly removed by the user.

It should be noted that this address is not intended to replace other existing addresses. The proposed approach would not be helpful if users have only a business card or email address. However, a SIP user agent software usually can handle multiple user accounts. The locally generated identifier should be used as an additional and optional account which is useful under the ad-hoc and ephemeral communication environments.

**IV. IMPLEMENTATION**

We implemented the two proposed ideas; one is a feature for exchanging contacts by using the rendezvous address and another is the UUID SIP-URI.

SIP-Communicator [4], which is an open source of Java implementation of a SIP user agent, is used as a platform. We first modified it to make it work without SIP proxy/registrar servers. Then we implemented a Java library for the P2P client protocol which is a subset of the P2P protocol defined in [5] and incorporated it into the platform.

**A. Overview**

In a meeting or conference room in person, when a user wants to exchange his contact address with someone, following steps are taken.

1. Issue a rendezvous address by selecting “Get Contact”
from the popup menu on the GUI (Fig. 2). The popup window shows the newly issued address.

(2) Tell it to the recipient.

And then on the recipient’s host,

(3) The recipient inputs the address on the popup window shown up by selecting “Register Contact” from the popup menu on the GUI.

A new entry is created both in the user’s address book and in the recipient’s address book.

When the user makes a call, he can select an entry in the address book and press the button on the GUI. Through these operations, users do not need to use actual SIP-URIs explicitly.

B. Generating UUID SIP-URI

When the software is started for the first time, a SIP-URI is automatically generated in the local host by concatenating a user part generated by java.util.UUID and a preconfigured domain name like example.com. An example of the SIP-URI is shown below.

4def145b-e510-4dd4-a3a9-25008bc0ba8a@example.com

This SIP-URI is stored in a local file and is used until it is explicitly discarded or overwritten by the user.

C. Rendezvous Address

The rendezvous address is an address to exchange contacts easily and it helps users to use even lengthy addresses like an automatically generated SIP-URI mentioned above. In order to create a unique address in a distributed manner, the rendezvous address is generated in ways shown below.

Generating Rendezvous Addresses

- With a random number generator

In a way similar to IPv4 link-local addresses [6], a host selects an address using a pseudo-random number generator with a uniform distribution in the range from 0000 to 9999. If the address is being used by the other hosts, it repeatedly generates a new address until it finds unused one.

This method can cause noticeable amount of delay. As the number of hosts increases, the chance of the repetition becomes high. As a result, the amount of the traffic and the delay will increase.

This method is used only in a case where the overlay network is used to store only location information and storing other kinds of data are not allowed. To shorten the delay, when the overlay can be used for multiple purposes, the next method is used.

- With an incrementing counter

This method is simple and so effective that it can avoid the delay. When the first host starts the generation of the rendezvous address, it uses a pseudo-random number generator described above. It generates an address, for example, 1234. The host stores a key-value pair shown below in the overlay network in the URN form [7], where the preceding “x-” means that it is experimental. This pair indicates that the address 1234 is the most recently generated rendezvous address. This key-value pair expires in 30 minutes.

(key, value) = (urn:x-rendezvous-address:last, urn:x-rendezvous-address:address:1234)

The next host looks for the value associated with the key “urn:x-rendezvous-address:last”. When the value is found, it increments the value, to 1235 in this example. If the address is not used by anyone, it uses the value. If the address is in use, it repeatedly increments the value until it finds unused one.

Uploading SIP-URI with Rendezvous Address

After generating the rendezvous address, the P2P client software automatically uploads a key-value pair of the rendezvous address and the automatically generated SIP-URI into the overlay network in the form shown below.

(key, value) = (urn:x-rendezvous-address:address:1234, sip:UUID1@example.com)

The key-value pair is uploaded with a lifetime so that it will expire in a specified period. The expiration time must be longer than the period while the contact exchange is being performed by the users. The default value of the expiration time is 30 minutes.

The P2P client software starts monitoring the change of the value associated with the key “urn:x-rendezvous-address:return-address:1234” in a polling manner. This key-value pair also expires in 30 minutes.

The user tells the rendezvous address “1234” to the recipient user in person. The recipient user enters the address on the recipient’s host. The software downloads the value associated with the key “urn:x-rendezvous-address:address:1234”.

Fig. 3. Contact exchange
and creates a new entry for the value “sip:UUID1@example.com” in the address book. After that, a key-value pair shown below which includes the recipient’s AOR is uploaded (Fig. 3).

\[(key, value) = (\text{urn:x-rendezvous-address:return-address:1234}, \text{sip:UUID2@example.com})\]

**Downloading SIP-URI with Rendezvous Address**

The user monitors a value associated with the key “urn:x-rendezvous-address:return-address:1234” and retrieves the value “sip:UUID2@example.com.” The software creates a new entry in the local address book and notifies the user (Fig. 3).

**Broadcasting contacts**

The sequence described above is not applicable if the user wants to convey the contact to many recipients at a time (for example, as a speaker in a conference room does). In this case, one-way transaction can be used; the user tells the rendezvous address to all recipients in person. The recipients receive the user’s SIP-URI although the recipients do not upload anything.

To be noted that these mechanisms do not compromise existing security mechanisms like a digital signature. They will work together without significant modifications.

**D. Call setup**

After exchanging contact addresses, it is possible to make a call by using the address book. A call setup flow is shown below (Fig. 4).

**Uploading the location information**

The location information, which consists of an AOR and a Contact-URI, is stored in the overlay network right after the client software starts (step (0) in Fig. 4). The location information is stored in the form of the URN to specify the meaning of the data.

\[(key, value) = (\text{sip:UUID2@example.com}, \text{sip:Contact2@169.254.30.30})\]

**Downloading the location information**

When an entry in the address book, for example “BOB”, is selected and to which a new call is attempted (step (1) and (2) in Fig. 4), the software looks for the value associated with the key “sip:UUID2@example.com” (step (3) in Fig. 4). Then it finds “sip:Contact2@169.254.30.30” (step (4) in Fig. 4) and makes a call to the Contact-URI. The standard SIP call setup sequence follows (step (5) in Fig. 4).

**V. RELATED WORK**

The link-local address for IPv4 [6] has a similar mechanism although it is used in the IP layer, where an IP address is automatically generated by each host and is used after it is confirmed that the address is not used by the other hosts. This mechanism is useful in situations like a wireless Ad-Hoc network where the Internet is not reachable and there are no DHCP servers.

HIP (Host Identity Protocol) [10] [11] also does not need a central server to generate an identifier. The HIP defines a new name space called the Host Identity namespace. The Host Identifier is an identifier for a computing platform and is independent of network layers. Statistically globally unique and public cryptographic keys of a public/private key pair are used for the identifier. The DNS (Domain Name System) helps to associate the identifier with a different name space like a FQDN (Fully Qualified Domain Name).

Combination of DNS-SD (DNS-based Service Discovery) [8] and mDNS (multicast DNS) [9] can work as if all users in a subnet have a single shared address book. This combination allows users to announce their location information. Usually collision in the namespace is solved in a way that a numeral is automatically added to the original name. For example if "mycomputer.local" is already in use, it will be renamed to "mycomputer2.local". The combination of DNS-SD and mDNS works fine in a subnet such as a wireless ad-hoc network although it does not work in large networks with multiple subnets.

**VI. CONCLUSION**

We proposed two features for Peer-to-Peer SIP. One is a mechanism to exchange contact addresses by using the proposed rendezvous address. Users can use the rendezvous address when they want to create a new entry in the local address book without typing lengthy SIP-URIs. Another is automatically generated SIP-URIs which eliminates the sign-up process so that users can use the SIP user agent software right after installation without manual operation for the sign-up process.
We implemented these features on SIP-Communicator with the P2P client protocol to make them work on a P2P overlay network.

We plan to work on performance analysis and measurement including scalability issues of the overall system.

REFERENCES


