Mobile P2PSIP

Peer-to-Peer SIP Communication in Mobile Communities

Marcin Matuszewski, Esko Kokkonen
Nokia Research Center
Helsinki, Finland
marcin.matuszewski@nokia.com, esko.kokkonen@nokia.com

Abstract—**This paper presents a realization of a multi-service P2P overlay architecture that allows for establishing SIP multimedia sessions between mobile community members with little or no need for centralized servers of any kind. The overlay architecture forms a cost efficient platform for providing innovative mobile services. As an example, this paper presents an implementation of a mobile P2PSIP VoIP service. The implementation allows a mobile phone user make voice calls using resources of the P2P overlay. The mobile phone user may freely use standard call manager and contact list in a mobile phone. The paper also presents a method for locating relay servers that are used in NAT traversal.**

Keywords: Mobile Peer-to-Peer Session Initiation Protocol (P2PSIP); NAT traversal; Distributed Hash Table (DHT);

I. INTRODUCTION

The concept of Peer-to-Peer (P2P) has been much discussed during last years. The number of P2P applications is growing every day. A common characteristic of all P2P applications is the full distribution of resources and the minimum intervention of centralized servers.

The first and the most widely adopted P2P applications have enabled internet users to share digital content such as pictures, videos and music files. P2P content sharing applications were also deployed in the mobile environment [1, 2, 3, 4]. Some of the mobile P2P content sharing applications allowed mobile phone users to form groups and share content inside communities [5].

Internet real-time communication is typically realized using SIP communication model. Session Initiation Protocol (SIP) [6] is a signaling protocol that is used to establish, tear down and control multimedia sessions. SIP uses an Address-of-Record (AOR), to locate a user. An AOR is a SIP or SIPS URI that includes user and a domain part. Sip:marcin.matuszewski@nokia.com is a typical SIP URI, where the user part is separated from the domain part with @. Although two SIP User Agents (SIP UAs) can communicate directly without any SIP infrastructure, most deployments require proxy and registrar servers that assist SIP UAs in rendezvous, session initiation and multimedia session management. Proxy servers are used to help route SIP requests to the location of other SIP UAs, authenticate and authorize users for multimedia services as well as implement call-routing policies. Registrars allow users to upload their current locations (typically in the form of IP address and port) to the location server. This information is used by the proxy servers to route SIP request to the destination. In practice registrars, proxies and location servers are collocated in one computer called a SIP server.

In a typical SIP communication model (called sometimes a trapezoid model) each SIP UA is registered in a particular domain. If a caller registered in domain A wants to call a callee registered in domain B it sends a SIP request to its own proxy server. The proxy server resolves the domain part of SIP URI of the callee (in this case domain B) using DNS and forwards the request to the proxy server of domain B. The proxy server in domain B forwards the received request to the callee's endpoint using contact information stored in a location server of domain B.

As we can see, typical SIP communication requires a number of centralized servers. Because of this design decision the extension of P2P content sharing applications with real-time communication functionality would require deployment of SIP infrastructure such as SIP proxies, registrars, and other servers.

Besides many GPRS and WLAN networks today are behind NATs (Network Address Translators) or/and firewalls that restrict opening connections towards the mobile endpoints. Only a few nodes, which are made reachable from the Internet, are supplied with public IP addresses and are allowed to receive connections originated from other mobile endpoints. This certainly limits the reachability of a mobile endpoint at the time of joining a P2P network and establishing multimedia sessions with other mobile endpoints.

From these reasons there was a need to design a distributed communication mechanism that could interwork with conventional mobile SIP endpoints and networks, and provide a solution for firewall and NAT traversal. The idea of Voice over IP (VoIP) communication over a P2P network was first utilized in Skype [7] with a proprietary signaling protocol. Shortly after Skype was launched, the internet community started a standardization effort of a distributed mechanism for SIP real-time communication. A standard P2P SIP protocol is currently being designed by the P2PSIP Working Group (WG) [8] in the Internet Engineering Task Force (IETF)[9].

This paper presents an architecture of a multi-service mobile P2P overlay and an implementation of a mobile P2P

1-4244-1457-1/08/$25.00 © IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

SIP system that distributes the task of locating SIP proxies, SIP endpoints and other servers such as media relays to a P2P overlay network. The architecture allows combining P2P content sharing and real-time SIP communication networks. The paper also discusses a method for NAT traversal with the use of relay servers.

The paper is organised as follows. The second section introduces the architecture of the mobile multi-service P2P overlay. Section 3 presents methods for publishing, searching and retrieving information from the P2P overlay that is required to establish multimedia sessions between two SIP UAs. The implementation of the architecture and signalling flows are presented in Section 4. Section 5 discusses the results of the measurements of the mobile P2PSIP system. The last section concludes the paper.

II. Architecture

This section presents a multi-service P2P overlay architecture that allows for establishing SIP multimedia sessions between community members with little or no need for centralized servers of any kind. The architecture is presented in Figure 1.

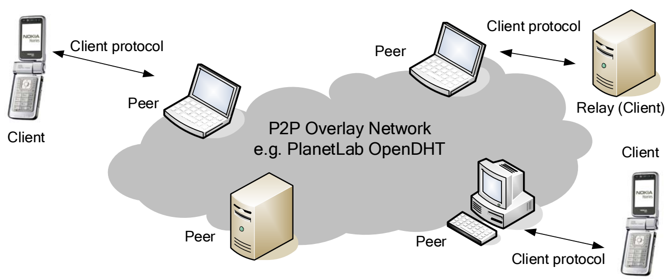

Figure 1. A multi-service P2P overlay

In this architecture, there are two types of entities:

1. Peers - maintain a P2P overlay network by running a P2P protocol and storing data such as contact information. In principle, they provide methods to allow applications and clients to insert, retrieve, modify and delete data stored in the overlay.

2. Clients - connect to peers and insert or retrieve information to/from the overlay network. Any device that does not want or cannot act as a peer can become a client. In Figure 1 we have two types of clients, i.e. mobile phones and a relay server that assist peers and clients in NAT traversal.

A SIP UA can be collocated with a peer or a client. The peer or the client software may reside in mobile endpoints. In this work we assume that mobile endpoints act as clients whereas peers are personal computers in the internet.

Mobile clients connect to peers that act as front-end agents towards the DHT overlay network. Clients and peers communicate with each other using a general purpose client protocol such as XML RPC. The general purpose client protocol allows publishing, modifying, deleting, and retrieving information about users, groups, services, and shared content.

The P2P overlay is based on Distributed Hash Table (DHT), however the architecture also allows unstructured P2P network topologies. DHT is a structured overlay network architecture that allows locating any piece of data stored in the overlay network using a limited number of messages, typically log(N) messages. Each peer is assigned a unique peer ID, sometimes also called a node ID, to enable it to join the overlay. If the DHT is organized in a ring topology, the hash table's keyspace is taken to be circular, and peer IDs are typically 128-bit or 160-bit unsigned integers representing position in the circular keyspace. Example DHT algorithms are presented in [10], [11], [12].

In this architecture the P2P overlay network acts as a general purpose distributed database. It may store different types of data such as user contact information, information about groups, services and content that overlay participants would like to share with others. Data stored in the DHT is indexed using data IDs. The data IDs are distributed among DHT nodes. Each DHT node stores a subset of the data IDs. The Data IDs are formed from a user ID selected by a user during enrollment to the overlay, or a telephone URL, or a SIP URI.

The data stored in the DHT is organized in records. The DHT stores different types of records such as a user, group or service record. Data belonging to users is stored in user records. At minimum, the user records store contact information of the user. In addition they may also store information about social groups the user belongs to as it is explained in [5], or content the user would like to share with others.

There can be two types of groups in the P2P overlay: formal and virtual. Formal groups require specific enrollment of the user into the group. The group is identified by a group ID, and the group keeps track of its members by keeping a list of user IDs that belong to it. Formal groups are registered in the overlay. Information about formal groups is stored in group records. The formal group records stored in the overlay include data about group members and some other additional data.

On the contrary, virtual groups do not require enrollment, and there is no allocated group ID to identify the group. Virtual groups are just created by a user in, e.g., a local Contacts application in user's device, when it ties one user ID with another one that shares certain interests.

The overlay may also store information about services such as STUN or TURN service. Information about services is stored in service records. STUN or TURN servers may publish their contact information to the P2P overlay. Nodes that require assistance from those servers may locate them in the P2P overlay and connect to them directly.

III. Processes

In this section we analyze the three basic processes that enable the architecture presented above. We divided those processes into three categories: relay location procedure, publication of own connectivity information, and session establishment. In our presentation we assume that some mobile endpoints may require assistance of a relay server in NAT

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

traversal procedure. The following sections provide an analysis of these three parts.

A. Relay location procedure

Relays publish their contact information in the P2P overlay e.g. a relay in Finland publishes information that it is located in Finland and is reachable under a certain IP address and port. Relays may be a standalone network entities or be collocated with peers in the P2P overlay. In this scenario we assume that relays are separate network entities that communicate with peers using the client protocol. The physical location of the relay gives an approximate value of the round trip time to the relay.

A node behind a NAT that requires media or/and signaling relaying first searches P2P overlay in order to retrieve the address of a relay (TURN server [12] or any other relay). When a node retrieves the address of the relay is known, it connects to the relay and requests for a public IP address and port pair. The relay reserves a port for the node and responds with its own IP address and the reserved port to the mobile endpoint. Consequently, the relay forwards all messages send to this IP and port allocated to the mobile endpoint to the mobile endpoint. The relay may reserve separate IP addresses and ports for signaling and media traffic.

B. Publication of own connectivity information towards other network nodes

After the mobile endpoint has obtained a public IP and port pair, this data must be published towards/registered in the overlay. The connectivity information of a mobile endpoint is stored in a user record that belongs to the user of the endpoint. The user record is indexed using hashed user ID. In our implementation user ID is equivalent to SIP URI assigned to a user. The publication guarantees that other nodes are able find the user's current contact information in the overlay and contact a user even if his device is behind a NAT.

C. Session establishment

When setting up a session, the contact information of the callee is retrieved from the overlay. The contact information of the callee can be retrieved from user or group records stored in the overlay. If the contact information points to an IP address and a port of a relay the caller sends SIP requests to the relay. The relay forwards all messages to the callee. SDP headers contained in SIP requests are modified to reflect the IP address and the port obtained from the relay.

IV. Implementation

We implemented the proposed architecture. Our system includes the following elements: DHT P2P network, mobile clients and a relay.

1) DHT P2P network

Our implementation uses PlanetLab OpenDHT system, which consist of several peers typically running on servers provided by universities or other (research) organizations. The mobile clients do not take part in the DHT ring themselves; instead they just put/get data to a P2P overlay network. Mobile clients and relays use XML-RPC as a client protocol. The use of HTTP makes it ideal for all clients that are behind firewalls.

PlanetLab OpenDHT stores the location of signaling/media relays and registration data of individual mobile SIP clients.

2) Mobile client

A mobile client is a software component that handles interaction with the rest of the P2P overlay and other clients. The mobile client consists of a P2PSIP module and a native VoIP software that is installed in Nokia mobile phones. P2PSIP module acts as a SIP registrar and proxy and provides SIP to DHT interface. The session parameters in SIP messages are easily modifiable in the proxy. The separation of the VoIP client from P2PSIP module allowed hiding the P2PSIP functionality from a mobile phone user. A mobile phone user uses standard phone UI when it wants to make voice calls using the resources of the P2P overlay. A mobile phone user may freely use standard call manager and contact list in a mobile phone.

In the alternative deployment scenarios a VoIP client may integrate the P2PSIP module so no SIP signaling is visible in the interaction with the overlay. Alternatively, the P2PSIP module can be collocated with peers in the P2P network. In this case a conventional VoIP client connects to the P2P overlay network using conventional SIP signaling and the P2PSIP module translates SIP signaling to P2P signaling. This alternative does not require any changes in the VoIP clients that are available in current SIP enabled mobile phones as well as installation of any new software on those devices.

3) Relay server

Relay servers are used for NAT/firewall traversal. The protocol between a mobile client and a relay is a proprietary protocol. It uses HTTP CONNECT to create a TCP connection to the relay, or a direct TCP connection if the connection is possible without an HTTP proxy. This TCP connection is used to forward UDP (SIP or RTP) packets to the client, bypassing any firewalls or NATs.

There may be several relays in the system, and users do not have to use the same one in order to connect to each other. Relays can either be separate entities or be integrated with peers. The second scenario is preferable because it allows for reduction of operational expenses (OPEX). In this scenario the network of peers would then act as a relays. In our implementation we used standalone relay servers because we could not modify peers which were operated in PlanetLab.

There are also many scenarios where relays are not needed. In real deployments, standard NAT traversal mechanism such as ICE and STUN can also be used to assist nodes in NAT traversal. We used relay servers because we wanted to demonstrate that the P2P overlay can also be used to locate relay servers that can assist clients and peers in NAT traversal.

The following sections describe message flows that are used to realize basic processes described in Section 3.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

A. *Relay location and contact information publication*

The following picture describes relay location and publication procedures assuming that the mobile client includes a VoIP client and the P2PSIP module and a P2P overlay network is based on DHT algorithm.

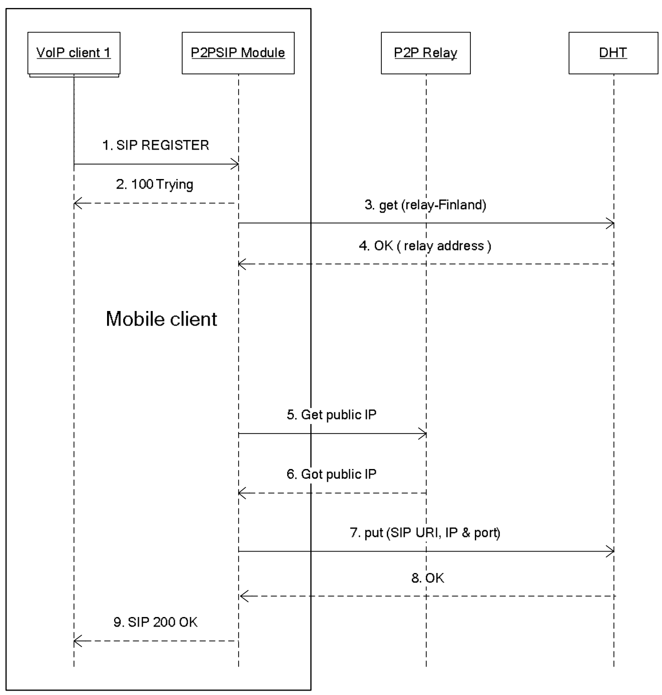

Figure 2. Registration

In the registration phase the VoIP client 1 sends a SIP REGISTER request to the P2P SIP module that acts as a SIP registrar (1). The P2PSIP module replies with a 200 TRYING provisional response to the received SIP REGISTER message (2). Then it tries to locate a relay. It sends a get (relay-Finland) request to the P2P DHT network using the XML RPC protocol (3). When the get request reaches a peer, the peer forwards the request to another peer according to the DHT algorithm. As a result the request reaches a peer that stores contact information of a relay. The DHT P2P overlay network responses with an address of the relay to the P2PSIP module in the mobile client (4). There can be many relays registered in the P2P overlay. If necessary, the DHT P2P overlay network can be asked for several possible relays.

In the next step the P2PSIP module establishes a TCP connection towards the relay using the address returned by the DHT P2P overlay network. When connection is established, the SIP to DHT interface module sends "Get public IP" message to the relay requesting allocation of a public IP address and port (5).

The relay allocates an IP address and a port for the mobile client, and returns them to the P2PSIP module with "Got public IP" message (6). The TCP connection to a relay stays alive. This connection will be later used for SIP signaling.

After receiving the IP address/port pair the P2PSIP module registers the new contact information of a mobile client by sending a PUT request to the DHT P2P overlay network (7). The put request is addressed to the AoR of a user and carries new contact information of the mobile endpoint. DHT P2P overlay network accepts the data and acknowledges it with a response.

After a successful registration in the overlay a 200 OK response is sent to the VoIP client.

B. *Session set-up*

Figure 3 presents a message flow of a session set-up.

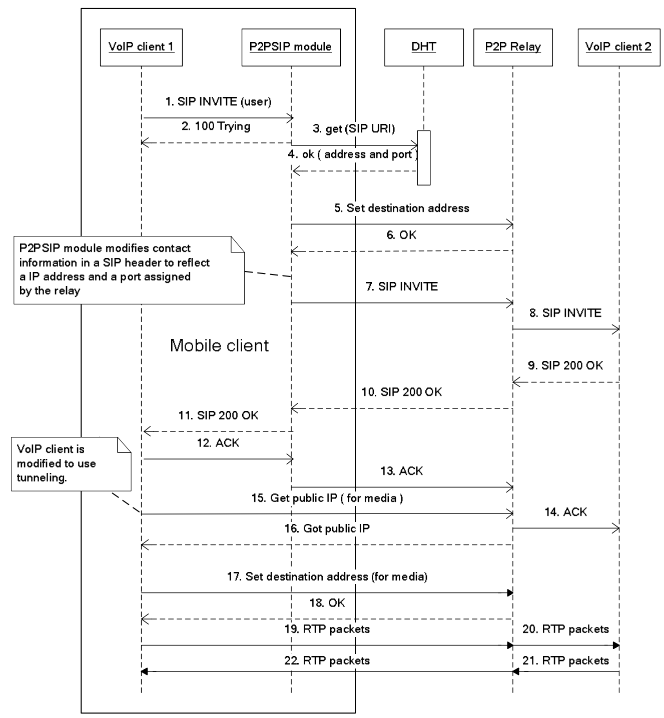

Figure 3. Originating a session

The VoIP client 1 sends a SIP INVITE request to the P2PSIP module (1) that immediately responds with a 100 Trying response (2). The P2PSIP module finds out a destination IP address and a port by sending get (SIP URI of a callee) request to the DHT P2P overlay network over XML RPC protocol (3). The SIP URI is retrieved from a Request-URI in the received INVITE request. The DHT finds an IP address and port of the user and returns them to the P2PSIP module (4). When the destination address for SIP signaling is known, the P2PSIP module configures the relay to send all SIP messages received over the signaling tunnel to that address (5). The relay acknowledges setting the destination (6).

Before forwarding SIP INVITE to the relay, the P2PSIP module modifies the contact information of the caller to the address and port that the relay allocated to the client. SIP headers (Via) are modified so that SIP responses are sent to the relay that forwards them to the mobile client. The SIP INVITE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

is combined with a proprietary tunneling header and sent trough the TCP connection (7). The relay forwards the received SIP INVITE message to the callee using UDP (8).

The callee replies with an appropriate response. For simplicity, Figure 3 only shows 200 OK response (9,10). In the reality, the callee also sends a 180 Ringing response before 200 OK. 180 Ringing is handled similarly to 200 OK in this messaging flow. The SIP response is sent to the relay (due to Via-header modification in step 7) that forwards it to the P2PSIP module trough the signaling tunnel.

After receiving a 200 OK response the P2PSIP module forwards the response to the VoIP client 1 (11). The VoIP client 1 responds with ACK that is forwarded to the callee trough the relay (12,13,14).

Before a media session can be set-up, the VoIP client 1 must request a public IP address/port pair for media from the relay. It creates a TCP connection to the relay, and sends “Get public IP” (15). The relay allocates an IP address and port for the VoIP client 1, and sends them to the VoIP client 1 (16).

After receiving 200 OK the VoIP client 1 knows the destination address for media. It configures the relay to forward packets from the media tunnel to that destination. Therefore, the VoIP client sends “Set destination address” trough the media tunnel to the relay (17). The relay confirms that the destination address in now set for the media tunnel.

The VoIP client 1 sends RTP packets trough the media tunnel. The RTP packets are encapsulated with tunneling headers and sent trough the TCP connection to the relay (19). The relay receives the encapsulated RTP packets from the media tunnel, puts them to UDP packets and sends them to the destination (20) (agreed on step 17).

The terminating party sends RTP packets to the relay (21). More specifically, those packets are sent to the address and port that the relay has reserved for media originated from or destined to the VoIP client 1 in caller's endpoint. Relay forwards the received RTP packets trough the media tunnel to the VoIP client 1 (22).

C. Incoming SIP session

Figure 4 present a signaling flow for an incoming SIP session.

An originating VoIP client (VoIP client 2) retrieves contact information of the user of VoIP client 1 from a DHT P2P overlay network (this operation is not shown in Figure 4). The contact information indicates an IP address and port of a relay. Next, VoIP client sends SIP INVITE to the relay (1). Alternatively the client sends SIP INVITE request to its outbound SIP proxy that forwards the request to the proxy of the DHT P2P overlay network or a peer that implements SIP server functionality. The proxy or a peer forwards the message to the relay.

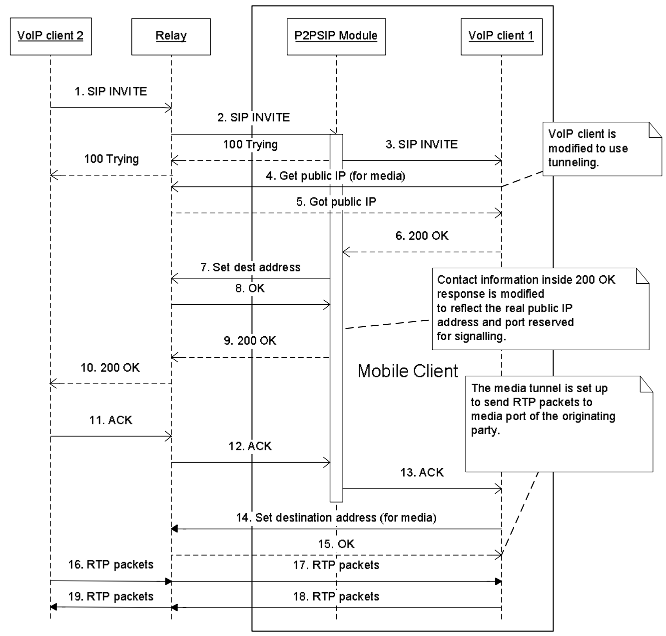

Figure 4. Incoming SIP session

The relay encapsulates SIP INVITE with tunneling headers and sends it trough signaling tunnel to the receiving side P2PSIP module (2) that forwards SIP INVITE to the VoIP client 1 with UDP (3). The VoIP client 1 creates a TCP connection to a relay (typically the same relay that was used for signaling) and sends “Get public IP” request (4). The relay allocates a port for the client for media, and sends the IP/port pair to the VoIP client 1 trough the media tunnel (5). The VoIP client 1 generates 200 OK response and sends it to the P2PSIP module (6)(180 Ringing message could be send before 200 OK, but in this diagram it is left out for simplicity).

For sending the response trough the signaling tunnel, the relay must know the destination address. P2PSIP retrieves the destination address from the received 200 OK response, and sends “Set destination address” message to the relay with the IP/port of the caller (7). The relay acknowledges setting of the destination address (8). The response is sent trough the signaling tunnel to the relay (9). The relay forwards the 200 OK response to the originating VoIP client 2 over UDP (10). VoIP client responds with ACK (11).

For sending media, the relay must know where to forward outgoint packets. Therefore, VoIP client 1 sends “Set destination address” message to the relay trough the media tunnel (14). The relay acknowledges setting of the destination address (15). The originating VoIP client starts sending RTP packets to the address signaled with SIP and SDP, namely the public address and port reserved for media in the relay. Incoming RTP packets are encapsulated with tunneling headers and sent to the receiving VoIP client 1 trough the media tunnel.

V. MEASUREMENTS

We performed measurements of the implemented system. The measurement setup consisted of two Nokia E61

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

multimedia computers, a relay and the PlanetLab OpenDHT overlay network.

We measured performance of the system in two case scenarios: with 3G cellular network and WLAN access network. The WLAN access network consisted of a WLAN access point which was connected directly to the public Internet. The 3G cellular network was provided by Elisa, a Finnish mobile operator.

We extended P2PSIP module software described in Section 4 with measurement capability. The extended software is able to measure registration, address discovery, and call setup delays. The amount of data transmitted was measured with the Ethereal protocol analyzer. The measurements include 21 measurement samples for each measured parameter.

Figure 5 shows measurements of a registration delay for 3G network. Figure 6 shows similar measurements for WLAN access network.

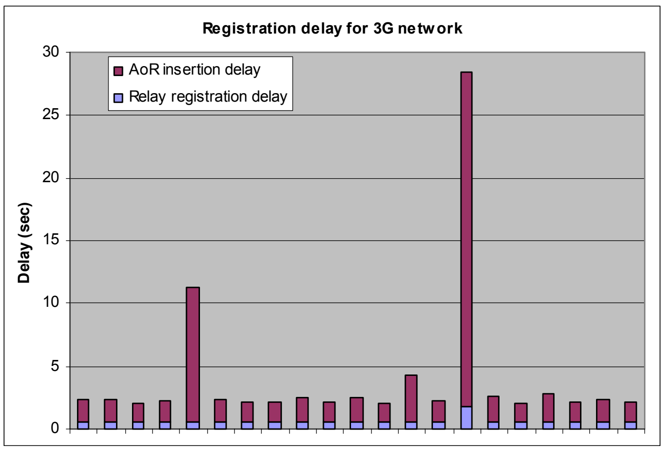

Figure 5. Registration delay in 3G network

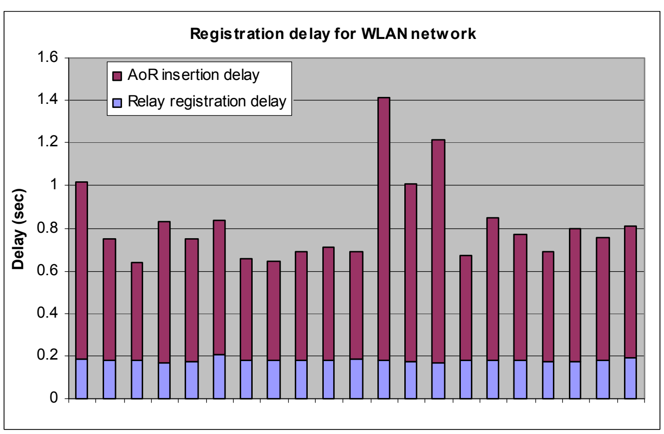

Figure 6. Registration delay in WLAN network

The registration delays presented in Figure 5 and Figure 6 have two components: a relay registration delay (marked with blue color) and a AoR (Address of Record) instertion delay (marked with red color). The relay registration delay consists of creating a TCP connection to the relay and requesting a public IP address and port from the relay. The AoR insertion delay represents the delay incurred during the process of insertion of a record to Planetlab OpenDHT with the XML RPC protocol.

As it was easy to predict, an average delay is significantly higher in the 3G network than in the WLAN network. As can be observed, the peak delay in 3G network setup may be significantly higher than the average delay.

Figure 7 shows call setup delays in the 3G network and Figure 8 shows the call setup delays in the WLAN access network.

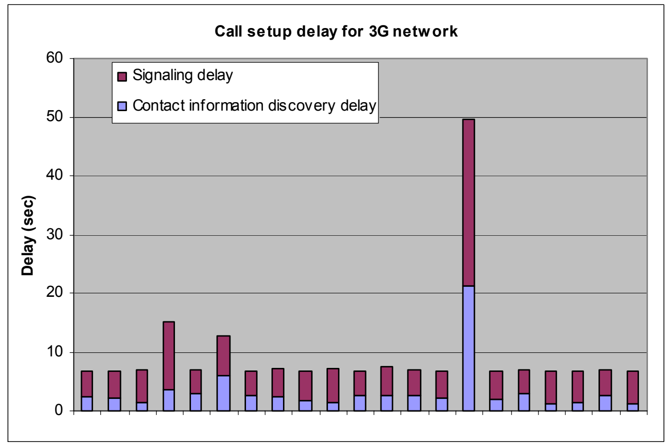

Figure 7. Call setup delay in 3G network

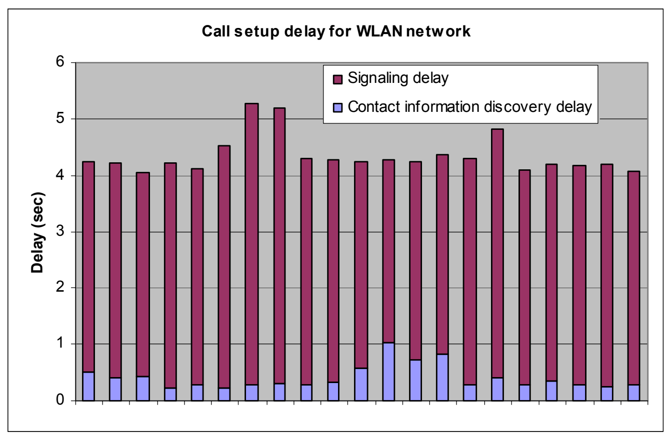

Figure 8. Call setup delay in WLAN network

The call setup delay has two components: a contact information discovery delay (marked with blue color) and a signaling delay (marked with red color). The contact information discovery delay is the time spent on fetching terminating party contact information from the PlanetLab OpenDHT network. The signaling delay is the time spent on sending SIP signaling between endpoints trough the relay. Signaling delay is calculated as follows: a timer starts when mobile client software sends the SIP INVITE request and stops when the software receives the SIP 200 OK response.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.
Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.

The average call setup delays suggest that the contact information discovery is much faster in WLAN than in 3G networks. When there is only a little variation of delays in WLAN network, there are some peaks in the 3G case. We could not find a clear explanation of those peaks. This may be caused by a DHT reconfiguration mechanism or congestion in 3G network. The averages of the delay incurred in the registration and the call setup are shown in Table 1.

TABLE I. AVERAGE REGISTRATION AND CALL SETUP DELAY

Average registration delay (ms)		
	3G	**WLAN**
Relay registration delay	635	179
AoR insertion delay	3426	640
Total	4061	819
Average call setup delay (ms)		
	3G	**WLAN**
Address discovery delay	3299	407
Signaling delay	6304	3944
Total	9603	4351

Table 2 shows the amount of traffic transmitted trough the air interface. DHT GET includes the amount of data transmitted when performing a data query from the PlanetLab OpenDHT network. Such queries are performed in the process of locating a media relay and the discovery of contact information.

DHT PUT describes the amounts of data transmitted when performing insertion of data to the PlanetLab OpenDHT network. In this system, an example of DHT PUT operation is the AoR insertion.

Traffic going through a relay includes traffic generated in the relay registration and the SIP session negotiations and teardowns. The amount of media traffic was not measured.

TABLE II. TRAFFIC TRANSMITTED TROUGH THE AIR INTERFACE

Method	Send (bytes)	Received (bytes)	Total (bytes)
DHT GET	647	423	1070
DHT PUT	647	271	918
Traffic trough relay	1970	1947	3917

As we can see the amount of traffic sent trough air interfaces stays very small in the measurement scenarios. The main problem is related to the peaks in the registration and call setup delays in the 3G network. It seems that there are no major problems with WLAN. The average delays in WLAN seem to be fairly reasonable.

VI. CONCLUSIONS

In this paper we presented a realization of a multi-service P2P overlay architecture that allows for establishing SIP multimedia sessions between mobile community members. The architecture allows extending P2P content sharing networks with real-time communication functionality without a need of maintaining centralized servers of any kind such as SIP proxy, registrar and location servers. The architecture distributes the task of locating SIP proxies, SIP endpoints and other servers such as a relay to a P2P overlay network.

We also presented an implementation of a mobile P2PSIP VoIP application. The implementation allows mobile phone users to make peer-to-peer voice calls in their community networks using resources of the multi-service P2P overlay network. The results of measurements show that registration and call setup delays incurred in the multi-purpose P2P overlay network as well as overhead traffic do not impose any significant restrictions on the commercial implementation of a server-less mobile VoIP service.

In the next steps we are planning to investigate peaks in the registration and call setup delays incurred in the 3G network and research a P2P overlay network that is formed with the use of only mobile phones.

REFERENCES

[1] Matuszewski, M., Beijar, N., Lehtinen, J. and T. Hyyryläinen, “Content Sharing in Mobile P2P Networks: myth or reality?”, International Journal of Mobile Network Design and Innovation, Vol. 1, Nos. 3/4, pp.197–207.

[2] Matuszewski, M., García-Martín, M. A., Beijar, N. and J. Lehtinen, “Resource Sharing and Discovery on Top of IMS”, Proc. of IEEE Consumer Communications and Networking Conference 2007, Las Vegas, NV USA, January 2007.

[3] SymTottent, online: http://www.aut.bme.hu/portal/SymTorrent.aspx?lang=en

[4] Hoßfeld, T., Mäder, A., Tutschku, K., Andersen, F-U., Kappler, C. de Meer, H. Dedinski, I. Oberender. J. (2004) ‘An Architecture Concept for Mobile P2P File Sharing Services’, Wurzburg University, Department of Distributed Systems, Institute of Computer Science, Technical Report No. 344, November.

[5] Matuszewski, M. and M.A. García-Martín, "Social Distributed Hash Table: share and collaborate in your social network", Unpublished, Submitted for publication.

[6] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[7] Skype: online: www.skype.com

[8] P2PSIP Working Group, online: http://www.ietf.org/html.charters/p2psip-charter.html

[9] Internet Engineering Task Force (IETF), online: www.ietf.org

[10] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", ACM SIGCOMM 2001, San Diego, CA, August 2001, pp. 149-160.

[11] Maymounkov P. and David Mazières, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric", First International Workshop on Peer-to-peer Systems 2002.

[12] The Bamboo Distributed Hash Table, online: http://bamboo-dht.org/pubs.html

[13] Rosenberg, J., Traversal Using Relay NAT (TURN), draft-rosenberg-midcom-turn-08, IETF (work in progress).

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2008 proceedings.

Authorized licensed use limited to: UNIVERSITAT POMPEU FABRA. Downloaded on November 6, 2008 at 13:15 from IEEE Xplore. Restrictions apply.