

Analysis of Peer-to-Peer SIP in a Distributed Mobile Middleware System

Erkki Harjula, Jussi Ala-Kurikka, Douglas Howie, Mika Ylianttila
MediaTeam Oulu Group, Department of Electrical and Information Engineering,
Information Processing Laboratory, P.O. Box 4500, FIN-90014 University of Oulu, Finland

Abstract- The seamless and flexible interconnection of the existing and emerging protocols and networks is essential to the success of the new generation mobile applications and services. For exploiting the functional diversity, we have developed a middleware for mobile devices called the Plug-and-Play Application Platform (PnPAP). We have previously studied the Session Initiation Protocol (SIP) as an interconnecting protocol for the PnPAP nodes. In this paper, we analyze the usage of a new variant of SIP called Peer-to-Peer SIP (P2P SIP) for this purpose. Our interconnection architecture utilizes layered (hierarchical) P2P SIP to establish a self-organizing, failure tolerant overlay signaling network between PnPAP nodes. Signaling latency and scalability of the layered P2P SIP signaling is compared in a simulation environment with both the traditional client-server SIP and fully distributed P2P SIP. Simulation results illustrate the benefits of the layered P2P SIP solution where only the supernodes participate to the DHT management, enabling light-weight peer node implementation for mobile nodes.

I. INTRODUCTION

Mobile devices are developing rapidly today, but they still have many restrictions when compared to fixed computing devices. However, there are a number of protocols available for mobile devices with rich set of diverging functionalities. Typically today's intelligent mobile device also has multiple connectivities available for communication, and the number is growing. In order to keep the application development straightforward and simple, intermediate middleware logic is needed to manage the protocols and connectivities so that applications do not need to directly take charge of using them. Middleware also facilitates optimizing the usage of the available communication protocols and connectivities.

PnPAP [1] is an intelligent middleware platform that offers a uniform interface for applications' communication needs. PnPAP manages the protocols and connectivities dynamically in order to provide optimal performance to applications during runtime. PnPAP focuses particularly on using P2P protocols, taking full advantage of their distributed nature. New protocols can be downloaded and taken into use on the fly, which makes the platform modular. This kind of modularity makes the platform lightweight for applications that do not require rich functionalities, as well as versatile for applications that require diverse functionalities and therefore more protocols. Applications can set QoS requirements for PnPAP so that it can serve them optimally.

For optimizing performance, modularity and compatibility with multiple networks, PnPAP nodes need to exchange protocols and control-, status-, and context information with

each other. A protocol for connecting PnPAP nodes into a single overlay network, PnPAP Network, is needed. It must be lightweight, guarantee interoperability with a variety of mobile networks and have the ability to carry required information with acceptable overhead. Figure 1 illustrates the PnPAP network structure. This paper continues the preliminary work made for harnessing SIP for autonomous mobile P2P networking [2], by utilizing the coming P2P SIP.

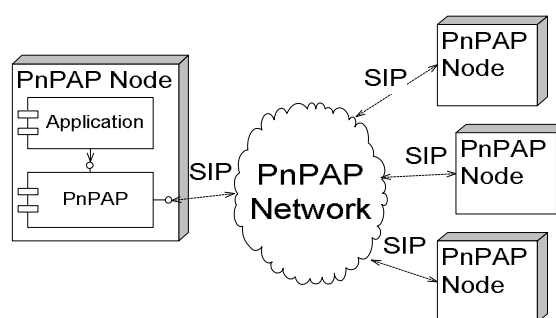


Figure 1. PnPAP overlay network structure

II. SESSION INITIATION PROTOCOL

SIP has been chosen as the protocol for IP-based multimedia call control for 3G wireless networks by the Third Generation Partnership Project (3GPP). This makes it a common standardized basis for session management, context exchange and instant messaging in the new generation mobile networks. SIP can be considered as one enabling technology for the Application Supernetworking. As concluded in [2], SIP is also a feasible option for PnPAP intercommunication.

A. Basic SIP

In its basic form, SIP is a client-server based control protocol above the transport layer for creating, modifying and terminating sessions between two or more participants. The core protocol of SIP [3] supports the following fundamental features: user registration, session management (initiation, modifying and termination), and user mobility (call redirection). Since the core protocol was made extendable, SIP has later been extended with numerous new features, such as instant messaging, presence, conferencing and emergency services. Recently, the emerging popularity of P2P has also made it a valid topic within the SIP standardization track. The new P2P SIP working group is currently being formed within the Internet Engineering Task Force (IETF) for adapting P2P with features suitable for SIP.

B. P2P SIP

The goal of P2P SIP is to allow true P2P networking between SIP peers by using SIP as the signaling protocol. The concept behind P2P SIP is to leverage the distributed and failure-tolerant nature of P2P by building an overlay network above SIP to provide P2P services for a network of SIP peers. This removes the need for centralized server components and allows true P2P networking between the SIP nodes. The focus of P2P SIP is constrained to the node and resource location and the essential supportive functions, such as overlay management, query routing, bootstrapping, registration, NAT- (Network Address Translator) and FW (firewall) traversal, authentication, security etc. The session management, after the remote peer or resource is located, is realized by normal SIP session management methods and is thus out of the scope of P2P SIP.

There are no standardized P2P SIP protocols yet. However, several tentative internet-drafts (I-Ds) have been made for grounding the P2P SIP work. I-D [4] focuses in describing the overall architecture and the environment of the system, whereas I-D [5] presents the more specified architectural design of the overlay network. There are also drafts presenting the problematics caused by the NATs and firewalls between the nodes and networks, e.g. I-D [7]. Together, these drafts constitute the foundation on which the P2P SIP will be based on.

I-D [4] presents the layered super-peer architecture for P2P SIP. In this model, the nodes are either ordinary peers or super-peers, wherein only super-peers contribute to the overlay. The ordinary peers connect the other peers through their super-peers. The Distributed Hash Table (DHT) is used for the management and routing of the overlay. Peers can become super-peers if they fulfil the terms that are e.g. peer's willingness to contribute, stability, performance or resource sufficiency. Accordingly, this model falls in P2P categories partially decentralized (third generation) structured architectures [6]. As proposed in [5], the P2P SIP uses Chord-based DHT routing algorithm for overlay management. In a nutshell, Chord [8] forms a decentralized, symmetric ring of peers, wherein each node only maintains information about $O(\log N)$ nodes for effective routing. Chord's main advantage is its robustness in handling frequent node failures and re-joins.

When started up, P2P SIP peer needs to either join the existing overlay or create a new overlay. In order to join to an existing overlay, peer must first locate some other peer that is already a part of the overlay, which is referred to as the bootstrap node. P2P SIP drafts do not define the exact lookup mechanisms, but they do list some choices for it. In priority order, these choices are service location, cached addresses, last good address and pre-configured bootstrap servers. After bootstrapping, peer must authenticate, possibly traverse NATs and firewalls and register itself to the overlay in order to be able to retrieve and update information. The authentication method and NAT traversal methods are not defined yet. NAT traversal will probably be the most challenging task in P2P SIP standardization process. This problem is discussed in I-D

[7]. I-D [4] presents the preliminary overlay API, which specifies the functions for the peer and resource insertion, deletion, updating and location. For implementing these functions, I-D [5] proposes using SIP REGISTER messages.

The layered super-peer architecture is a suitable platform for mobile peers, since as the ordinary peers they don't need to participate to the overlay tasks, requiring lots of signalling. For this reason, we use the layered super-peer architecture P2P SIP concept as the basis for PnPAP intercommunication architecture.

C. SIMPLE

SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) is an IETF working group that has defined a set of extendable features for SIP for presence exchange and instant messaging. Specifically, the RFCs [9] and [10] describe the architecture for asynchronous, push-type event notifications, such as presence information updates and session changes. SIP Event Notification [9] enables SIP nodes to subscribe to dynamically changing information so that whenever this information is changed, the subscriber node is informed. In a nutshell, event notification works as follows: a SIP node A (consumer), interested in event notification, sends a subscription message to SIP node B (producer). When B answers in the affirmative, the dialog is established. B sends now a notify message to A every time the event to which A has subscribed changes, until A leaves, B cancels the subscription or the subscription is expired. A Presence Event Package for SIP [10] is an instantiation of [9] and presents the client-server architecture for presence delivery. However, the original SIP event notification does not limit the architecture to client-server type alone.

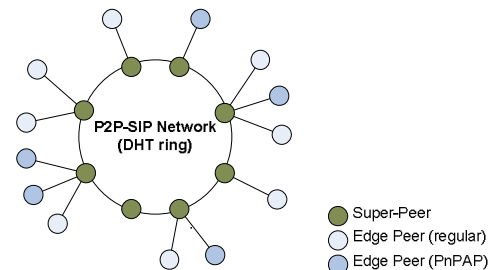


Figure 2. PnPAP network using P2P SIP overlay

III. PNPAP INTERCOMMUNICATION ARCHITECTURE

PnPAP intercommunication architecture is our novel concept architecture for connecting all PnPAP nodes to a single backbone network. Figure 2 illustrates the PnPAP intercommunication architecture utilizing layered super-peer type P2P SIP. PnPAP has three main objectives: 1) To enable control messaging between PnPAP nodes, 2) To provide applications a uniform messaging architecture for co-operation, and 3) To enable the exchange of data and context information between PnPAP nodes. The failure tolerance and freedom of maintenance are important goals for the PnPAP intercommunication. On this account, it is important that PnPAP intercommunication architecture incorporates as few

centralized components as possible. SIP has already been proven a feasible option for implementing PnPAP intercommunication in [2]. The measurement results in [11], for one, show that the overhead caused by SIP is tolerable when used as the signaling protocol for mobile P2P. Thus, P2P SIP appears to be a promising architecture since it provides the same features as client-server SIP, but is not dependent of centralized servers.

As mentioned earlier, in P2P SIP, super-peers handle most of the networking functionalities, leaving ordinary peers as little functionality as possible. As PnPAP is a mobile middleware architecture, all PnPAP nodes are mobile devices that are most probably only transiently connected to network, usually behind NATs and firewalls and have low-capacity network connections. Due to these restrictions, PnPAP nodes would not normally act as super-peers. Furthermore, because of the requirement of lightness, it is justifiable to leave super-peer functionalities unimplemented in PnPAP nodes, and use P2P SIP overlays managed by existing super-peers. In the following subsections we describe our proposal for using layered P2P SIP as the enabling protocol for PnPAP intercommunication on a more detailed level. Super-peer functionalities are not described in very detailed manner, as the behavior of them is outside the scope of this paper.

A. Initialization, registration, shutdown and failures

When PnPAP is started, it initializes the required connectivities and contacts the network. PnPAP makes use of the P2P SIP peer initiation described in [4]. The first action is to look up the closest super-peers. For locating these peers, [4] recommends using the following methods with the following priority order: 1) service location by multicast in local network, 2) cached addresses, 3) last good address, 4) publicly known bootstrap servers. However, as we are trying to avoid excessive signalling load to the mobile network, we use the caching of super-peer addresses as the primary method, multicast service location as the secondary method and pre-configured bootstrap server as the last method.

Once super-peer(s) have been detected, PnPAP peer attempts to associate and authenticate itself with closest of them. These methods are not defined by P2P SIP drafts at the moment, but we will follow the P2P SIP specifications when they become available. After successful authentication, P2P SIP peer must test the NAT and firewall e.g. by methods defined by ICE (Interactive Connectivity Establishment), with a super-peer providing STUN (Simple Traversal of UDP) server functionalities. Additional NAT/FW traversal configurations may be needed depending on the strictness of the firewall settings in the NAT node. However, authentication and NAT/FW traversal are outside the scope of this document and later on we assume that the NATs and FWs do not significantly affect the communication.

Once PnPAP peer has been successfully authenticated, it registers using SIP “REGISTER” message. Accordingly, super-peers serve as SIP registrars for the associated ordinary peers, including PnPAP peers. After successful registration, super-peer generates a user location record for PnPAP peer

and inserts it in the overlay network [4]. From this moment, PnPAP peer starts sending periodically the SIP “REGISTER” message to its super-peer and waits for the answer SIP “200 OK” to detect possible super-peer failures. Super-peers monitor the aliveness of the associated ordinary peers, including PnPAP peers by periodically sending a SIP “OPTIONS” message to them. PnPAP peer must answer this with a “200 OK”. To keep the signalling overhead minimal, the polling intervals are sparse, rather minutes than seconds.

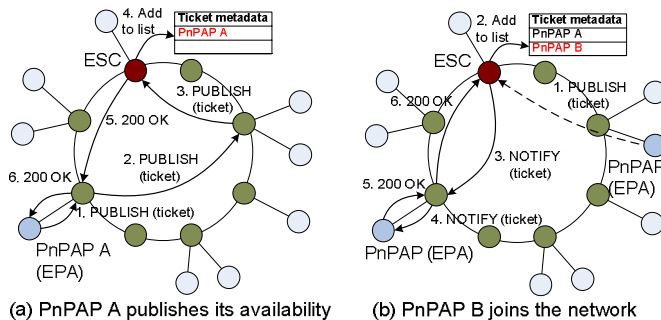


Figure 3. PnPAP list management examples

After joining the overlay, PnPAP peer must learn about the other PnPAP peers in the overlay and make itself visible in order to be able to communicate with them. For this, we exploit the resource sharing mechanism presented in I-D [12] and [13]. We assume that PnPAP peers implement the EPA- (Event Publishing Agent) and super-peers implement the ESC (Event State Compositor) functionalities, described in [12].

First, for publishing its availability to others, PnPAP peer shares a resource, later referred as ticket, using the SIP “PUBLISH” message (Figure 3a). Sharing the ticket, which is identical in all PnPAP peers, indicates that the ticket’s host is a PnPAP peer. The ESC, which holds the hash space containing the hash of the ticket, lists the peers (that are sharing the ticket) to the resource metadata describing the shared ticket. Thus, this metadata contains the list of all online PnPAP peers.

Next, PnPAP peer (EPA) subscribes to the ticket via its super-peer by using SIP “SUBSCRIBE” message. The overlay routes the subscription to the ESC super-peer hosting the metadata of the ticket. The ESC super-peer responds with SIP “200 OK” and returns the metadata containing the list of PnPAP peers to the PnPAP peer using SIP “NOTIFY” message, as defined in [9]. From now on, every time the metadata is changed, i.e. peers join to or leave from the overlay, ESC notifies the subscribed nodes. Figure 3b provides an example of the case of PnPAP peer joining the previously one-node sized PnPAP overlay network.

PnPAP peer has to renew the publication periodically. If the publication is not renewed before its expiration, PnPAP peer’s entry is removed from the ticket metadata. This occurs, for example, when PnPAP peer has failed. PnPAP peer must renew, in addition, the subscription to the ticket periodically to keep it alive. To keep the signalling overhead minimal, the expiration times are relatively long, rather several minutes than seconds. If the ESC super-peer leaves the network, the

resources it holds, including the ticket, are transferred to the super-peer next to it. In case the ESC super-peer fails, the ticket metadata is lost. However, with the next publication- or subscription refresh attempts of PnPAP peers, the messages are routed to the new ESC node. The new list is populated just as in the case when the peers join the PnPAP network for the first time. For making this scenario possible, PnPAPs use the full resource document (instead of partial documents) in refreshing the publications.

When PnPAP peer leaves the network normally, it unpublishes the ticket and ESC removes its entry from the ticket metadata. PnPAP peer also terminates the subscription to the ticket. After this, PnPAP peer unregisters from the overlay by sending a SIP “REGISTER” message with a zero value in its “Expires” field to its super-peer. The super-peer then requests the overlay to remove PnPAP peer’s user location record from the overlay. PnPAP’s own super-peer can also leave the network or fail. In this situation PnPAP looks up the next closest super-peer, authenticates and registers with it.

B. Application Supernetworking and Control Messaging

PnPAP intercommunication is designed to provide different applications with a channel for interacting directly with each other as well as a channel for PnPAPs’ mutual control messaging, e.g. in exchanging requests and responses concerning protocol and connectivity management. In our proposal, PnPAPs use the SIP “MESSAGE” [14] for carrying these messages and replies to them in their payload.

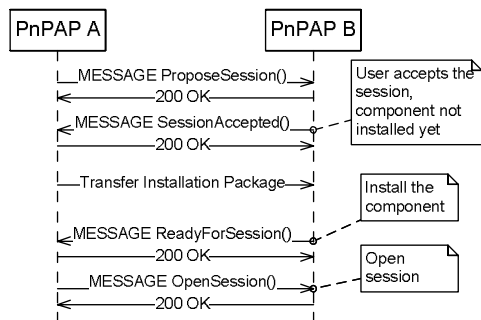


Figure 4. ACPC messaging using P2P SIP

Agile Content Push Control (ACPC) [15] is our concept for P2P-based mobile content distribution for PnPAP. We use ACPC as the example case for PnPAP inter-communication messaging. Figure 4 illustrates the example ACPC messaging scenario, in which the user of PnPAP A proposes a game session to the user of PnPAP B, who does not have the game in question. When the user of PnPAP B accepts the query, the game is automatically downloaded and installed to PnPAP B node, and the game is then started. The scenario is described more specifically in [14].

C. Exchanging PnPAP Components

As mentioned in chapter IIIA, we use SIP “SUBSCRIBE” and “NOTIFY” messages for locating resources, and SIP “PUBLISH” message for sharing resources. The message routing in overlay (i.e. between super-peers) seems to differ

between I-Ds [4] and [5]. However the differences do not affect the PnPAP functionality requirements as they do not become super-peers in any circumstances. As I-D [4]’s proposal follows the Chord algorithm, we use it in our example sequence, which is illustrated in Figure 5.

In Figure 5, an application using PnPAP A requests a function that is not currently supported by any installed protocol. First PnPAP A sends a query for the protocol that would implement the requested function to its super-peer by using the “SUBSCRIBE” message (message 1 in figure 5). The super-peer routes the message further to the P2P SIP overlay (messages 2-3). In this example case, one remote PnPAP node (PnPAP B) has a protocol that provides the required functionality so the super-peer holding its resource metadata responds to PnPAP A with a “200 OK” (4,5) and “NOTIFY” messages (6-7) for which PnPAP A answers with “200 OK” (8,9). When PnPAP A has received the “NOTIFY” revealing the location (PnPAP B) of the protocol installation file, it initiates the session for downloading the file by sending “INVITE” directly to PnPAP B (10). PnPAP B accepts the session by “200 OK” (11) and the file is then downloaded (12). When the protocol has been downloaded, PnPAP installs it and then executes the requested function.

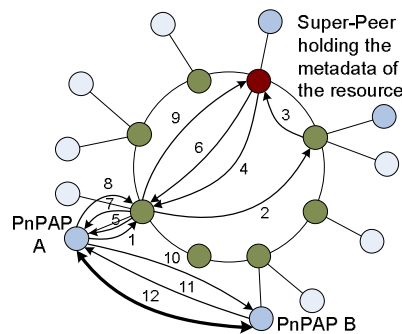


Figure 5. PnPAP Protocol fetch using P2P SIP

D. Presence and Context Exchange

PnPAP intercommunication also provides the possibility of exchanging context information (CI) between the peers. The general CI exchange messaging follows the sequences presented in Chapter 3A, but the exchanged resources are the certain types of CI of a known PnPAP peer (instead of the common ticket resource). For enabling CI exchange, PnPAP nodes must provide the Presence Use Agent (PUA) and EPA functionalities. Super-peers are assumed to provide Presence Agent (PA) and ESC functionalities. These functionalities are specified in RFC [14] and I-D [12].

Exchanging CI can be roughly divided to two types: synchronous (pull-type) and asynchronous (push-type). In the first case, the consumer node requests the CI from producer node separately each time it needs the CI. In the second case, the consumer node subscribes to the producer node to get asynchronous updates to the CI, as described in [9]. We use the SUBSCRIBE/NOTIFY mechanism for both types, but in the pull-type CI exchange the subscription is made only for single response.

IV. RESULTS AND DISCUSSION

We have built a PnPAP prototype for the Nokia Series 60 platform based on the Symbian OS. The environment sets strict requirements for the PnPAP intercommunication. The platform has limited hardware resources, which prevents using components that incur heavy processing and memory load. Mobile devices running PnPAP are also most likely connected to the Internet using unreliable, expensive and low-capacity wireless connectivity. This prevents using protocols that waste network capacity due to large overhead. For analyzing the feasibility of our approach, we have made some comparative performance estimates and measurements of the client-server SIP and P2P SIP.

A. Performance estimation

There are major functional differences between client-server SIP and P2P SIP. The P2P SIP characteristics depend on whether the architecture is totally decentralized or partially decentralized (layered super-peer architecture), where the PnPAP peers would be ordinary peers. In Table 1, we have listed estimates of some differing key features from the viewpoint of mobile peers and networks. These estimates are based on the specifications and drafts [3], [4] and [5]. As the table points out, P2P SIP architectures seem advantageous when looking at reliability features, whereas client-server SIP fares better in performance features.

TABLE I. CLIENT-SERVER SIP AND P2P SIP COMPARISON

Feature	Client-server SIP	P2P SIP (layered)	P2P SIP (decentralized)
Need for maintenance	yes	no	no
Failure tolerance	poor	good	good
Resource discovery	no	yes	yes
Suitability for mobile peer	good	good	poor
Joining & registration latency	good	average	poor
End-to-end latency	good	average	average

B. Simulated performance measurements

We made measurements to compare the performance of client-server SIP and P2P SIP using live network simulation. PnPAP peers ran on Nokia Series 60 phones with low-capacity GPRS Internet connections, and other peers were hosted by desktop PCs with typical home Internet connections, all located in the same region. For the measurements, we assumed the following. N is the number of peers participating in DHT in the case of P2P SIP and the number of servers in the case of client-server SIP. It should be noticed that in client-server SIP, the number of servers in the network does not basically affect the signalling routes between the end-nodes at all. We also assumed that each DHT peer in P2P SIP can store all the needed keys. Using these assumptions, the average path length M for searching a key in DHT network of N nodes can be calculated using the following formula [8]:

$$(1) \quad M = (1/2) \log_2 N$$

The average path length is used for simulating the performance of different models for each network size. We made two measurements in our simulations: registration latency and the message delivery time between the peers in cases when the DHT peer already has (familiar) or does not have (unfamiliar) the address of the target peer. It is assumed that in the case of client-server SIP, the server knows the target peer address in any case. In the registration simulation, we did not include the PnPAP peer discovery, bootstrapping, NAT traversal or authentication, since there are a number of different ways to carry them out with each models. The average message delivery times for different network sizes are presented in Figure 7. The values present the average of 100 test runs for each network size.

As expected, Figure 7a shows that the registration latency of totally decentralized P2P SIP is the longest and the latency grows with network size, as it has to join the DHT, which we assume requiring averagely $(1/2)\log_2 N$ message redirections before finding the correct location in the DHT ring. There are no big differences in the registration latencies between the two latter models since the layered P2P SIP needs to register only for the closest super-peer, which makes the sequence basically identical to client-server SIP.

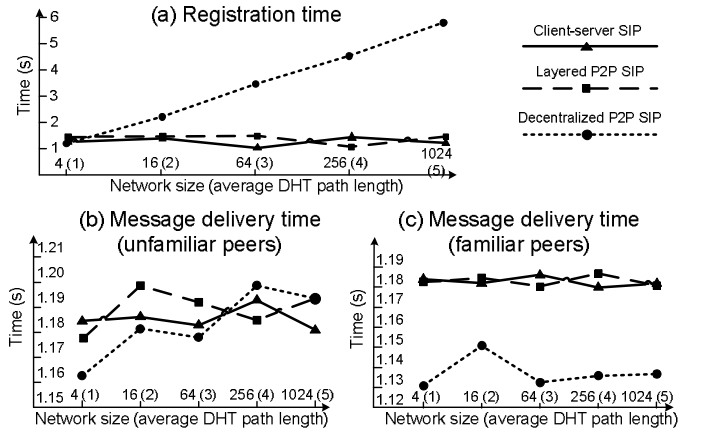


Figure 7. Measurement results

In the case of message delivery time between unfamiliar (uncached) peers (Figure 7b), the differences seem to be marginal. The variance of latency is inside 40ms whereas the total latency is more than one second. This can be explained by the fact that the first low quality wireless links dominate the results. The delivery time of client-server SIP seems to stay constant, as the number of servers does not affect its performance. P2P SIP models' delivery times seem to have a slightly growing trend as the network size grows. The total average path length between two ordinary peers in layered P2P SIP is the average path length of the overlay plus the hops between ordinary peer and super-peer in both ends, calculated by formula (2). In totally decentralized P2P SIP, the total average path length is calculated by formula (1).

$$(2) \quad M = (1/2) \log_2 N + 2$$

However, this difference between the P2P SIP models can barely be noticed from the Figure 7b. It can be seen that the

differences are unsubstantial with the used network sizes, but with larger networks they would probably be more easily recognizable. It should also be noticed that in the same-sized networks, the number of peers participating the DHT is significantly lower in layered P2P SIP than totally decentralized P2P SIP, as significant number of peers are ordinary peers. This makes the actual performance of totally decentralized P2P SIP worse than Figure 7b shows.

In the case of message delivery time between familiar (cached) peers (Figure 7c), the client-server SIP and layered P2P SIP seem rather equal, whereas the delivery time of totally decentralized P2P SIP is slightly smaller. This is explained by the absence of intermediate peers (servers or super-peers) in totally decentralized P2P SIP when compared to the other models.

C. Discussion and future work

In theory, the benefits of P2P SIP come at the cost of performance, as stated in [16]. However, when analyzing our measurement results, we can see that the performance cost is significant only in the registration of totally decentralized P2P SIP model. The low performance of the first link between the mobile device and the network dominates the total performance with high latencies making the signalling between the mobile device and the network critical from the viewpoint of performance, as the measured latencies illustrate. Partially centralized, layered P2P SIP architecture handles this problem by setting low-performance peers as ordinary peers and thus relieving them of the DHT management. As can be seen from the registration latencies, the performance of layered P2P SIP is identical to client-server SIP in that sense. From these findings, we can make two conclusions: 1) For mobile PnPAP nodes, layered model of P2P SIP provides averagely better performance than totally decentralized model, and 2) The performance of layered model of P2P SIP is very close to client-server SIP with the measured network sizes.

The security issues of PnPAP overlay management and the impact of NATs and firewalls need to be further investigated. Without any security mechanisms, the provided method is prone to different types of security attacks. Some NAT traversal mechanisms might also require changes for the methods presented in this paper. The future work also includes studying the possibility of adding super-peer functionalities to PnPAP peers as the mobile device capabilities are getting better.

V. CONCLUSIONS

In this paper, we analyze the usage of a new variant of SIP called Peer-to-Peer SIP (P2P SIP) for interconnecting Plug-and-Play Application Platform (PnPAP) nodes. First, we introduced PnPAP intercommunication architecture based on layered P2P SIP and analyzed it on a more detail in different communication scenarios. Then, we analyzed the functionality and performance issues from the standpoint of mobile middleware and compared the P2P SIP models with traditional client-server SIP to illustrate the feasibility of our approach.

Signaling latency and scalability of the layered P2P SIP signaling was compared in a simulation environment with both the traditional client-server SIP and fully distributed P2P SIP. It was seen that since in the layered P2P SIP solution only the supernodes participate to the DHT management, light-weight peer node implementation for e.g., mobile nodes can be enabled. This is one of the key benefits of layered and hierarchical approach in comparison to fully distributed P2P SIP where all nodes have similar roles.

In comparison to client-server SIP, closely similar performance results were obtained in a GPRS network due to dominating role of the first-hop latency. In a WLAN network, where significantly lower first-hop latency is expected, the situation can be different. Analyzing this and modelling other communication scenarios remains as future work.

REFERENCES

- [1] E. Harjula et al, "Plug-and-Play Application Platform: Towards Mobile Peer-to-Peer", in the Proceedings of the 3rd International conference on Mobile and Ubiquitous multimedia (MUM2004), College Park, Maryland, USA, October 2004, pp. 63-69.
- [2] D. Howie et al., "Harnessing SIP for Autonomous Mobile Peer-to-Peer Networking", in the Proceedings of the IEEE Global Telecommunications Conference (Globecom05), St. Louis, Missouri, USA, November, 2005, Vol.2., pp. 879-883.
- [3] J. Rosenberg et al., "SIP: Session Initiation Protocol", RFC3261, Internet Engineering Task Force, June 2002.
- [4] E. Shim et al., "An architecture for Peer-to-Peer Session Initiation Protocol (P2P SIP)" (work in progress), IETF draft, draft-shim-sipping-p2p-arch-00, February 2006.
- [5] D. Bryan, B. Lowekamp "A P2P Approach to SIP Registration and Resource Location", (work in progress), IETF draft, draft-bryan-sipping-p2p-02, March 2006.
- [6] S. Androutsellis-Theotokis, D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies", ACM Computing Surveys, vol. 36, No. 4, December 2004, pp. 335-371.
- [7] E. Cooper, P. Matthews, "The Effect of NATs on P2P SIP Overlay Architecture", (work in progress), IETF draft, draft-matthews-p2psip-nats-and-overlays-00, February 2006.
- [8] I. Stoica, et al., "Chord: A scalable peer-to-peer lookup service for Internet applications," in SIGCOMM, San Diego, CA, USA, August 2001, pp. 149-160.
- [9] B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification", RFC3265, Internet Engineering Task Force, June 2002.
- [10] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC3856, Internet Engineering Task Force, August 2004.
- [11] J. Lehtinen, "Design and Implementation of Mobile Peer-to-Peer Application", MSc. Thesis, Helsinki University of Technology, January 2006.
- [12] M.Garcia-Martin et al., "A Framework for Sharing Resources with the Session Initiation Protocol (SIP)", (work in progress), IETF draft, draft-garcia-sipping-resource-sharing-framework-00, June 2006.
- [13] M.Garcia-Martin, M. Matuszewski, "A Session Initiation Protocol (SIP) Event Package and Data Format for Describing Generic Resources" (work in progress), IETF draft, draft-garcia-sipping-resource-event-package-00, June 2006.
- [14] B. Campbell et al., "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC3428, Internet Engineering Task Force, December 2001.
- [15] O. Kassinen et al., "Group-Based Content Push with Dynamic Session Startup", Proc. of the 4th International conference on Mobile and Ubiquitous multimedia (MUM2005), The University of Canterbury, Christchurch, New Zealand, December 2005, pp. 135-141.
- [16] K. Singh, H. Schulzrinne, "Peer-to-peer Internet Telephony using SIP", Columbia University Technical Report CUCS-044-04, New York, NY, Oct 2004.