

SUITABILITY OF DHT-BASED PEER-TO-PEER SESSION INITIATION PROTOCOL FOR WIRELESS DISTRIBUTED SERVICES

Otso Kassinen Erkki Harjula Mika Ylianttila
University of Oulu University of Oulu University of Oulu
Oulu, Finland Oulu, Finland Oulu, Finland

ABSTRACT

Providing both analytical and empirical results, we analyze the suitability of DHT-based Peer-to-Peer Session Initiation Protocol (P2PSIP) overlay networking as a substrate for wireless, distributed multimedia services. Session initiation is just one application of this framework. First, we discuss the potential of P2PSIP for different applications and analyze the security and reliability of P2PSIP. Second, we present our prototype mobile P2PSIP implementation and a P2PSIP-based distributed calendar application for wireless user groups. Third, we measure average DHT hop count and retransmission count in up to 2000-node P2PSIP overlays. Fourth, we analyze numerically the scalability and robustness of P2PSIP and client-server SIP in some generic networking scenarios. Our results demonstrate the feasibility of P2PSIP as a standardized peer-to-peer networking protocol for distributed Internet applications, both fixed-line and mobile.

I. INTRODUCTION

Recent years have witnessed a substantial increase in the popularity of rich Internet communication services such as online games, VoIP, instant messaging, and social networking. All these services involve communications between two or more end-user terminals on top of a suitable infrastructure. Increasingly, the access devices are wireless.

Communication services can be roughly divided into two categories by their model of information exchange: 1) *mediated exchange*, where information is posted to a shared application environment accessible by both endpoints, for example over HTTP to a Web site; and 2) *direct exchange*, where application-specific information is transferred in real-time between application instances running at the endpoints.

When endpoints want to interact using a direct-exchange oriented application, a session needs to be established between them. A prominent standardized solution for session management is the Session Initiation Protocol (SIP), an application-layer signaling protocol for creating, modifying and terminating sessions between two or more endpoints. SIP has a client-server architecture. A SIP network consists of a set of (fixed-line) SIP servers with different kinds of functionalities; SIP user agents, i.e. clients, use the servers to initiate direct exchange. In this paper, the traditional SIP architecture is referred to as *client-server SIP*. Several extensions for client-server SIP exist, including instant messaging, presence, and emergency services, and there are proposals for advanced modifications such as transparent session transfer between terminals for mobile users [1].

In both fixed-line and mobile access networks, *peer-to-peer* (P2P) networking has long been a subject of intensive study. In consequence, the Internet Engineering Task Force (IETF) has recently formed the *Peer-to-Peer SIP* (P2PSIP) working

group for standardizing serverless, decentralized operation of SIP [2]. Distributed hash tables (DHT) are a key technology for building the efficient P2P overlays needed. An early DHT-based approach similar to P2PSIP is presented in [3].

It is possible to use SIP on top of a decentralized P2PSIP overlay network. However, P2PSIP has a wider scope of applicability. Whereas SIP was designed for direct-exchange communications, P2PSIP also standardizes mediated exchange; P2PSIP enables the storage, discovery and access of digital resources as its core functionalities. Session signaling is just one application of the P2P framework.

Wireless use of P2PSIP (wired-wireless convergence) has been considered in the design from early on. Thus, P2PSIP provides a standard platform for both direct- and mediated-exchange oriented mobile services. However, the mobile P2PSIP related research efforts in the literature have focused on the protocol's properties for direct exchange applications, such as the VoIP call setup delays evaluated in [4].

Content sharing systems are a typical example of mediated exchange. In [5], the popular content sharing system called BitTorrent, a DHT-based P2P system, is analyzed. In addition to content sharing, mediated exchange can be used among other things for implementing P2P mailbox-type messaging services: in [6], measurements of performance in a mobile publish/subscribe P2P system are provided. DHT algorithms – a key enabler for mediated exchange in P2P systems – have been extensively studied; for example in [7], four algorithms are evaluated in terms of bandwidth usage and lookup latency as a function of the essential parameters of the algorithms.

In this paper we analyze the potential of P2PSIP for providing generic functionality for mediated-exchange services, in addition to the more established work on SIP-based, mostly direct-exchange type services on top of the decentralized platform. Our results are based on qualitative and quantitative analysis, and on an evaluation of P2PSIP overlay networks run with an actual P2PSIP implementation.

II. P2PSIP FEASIBILITY ANALYSIS

A. Scope and Potential for Different Network Applications

The usefulness of traditional SIP networks lies not just in the unified format of the session-signaling messages. Before any negotiation for session establishment can take place, the endpoints must be able to locate each other. In an IP-based network, this ultimately translates into resolving, in one way or another, the IP address of the remote party. In SIP systems, every user has a SIP URI (address-of-record) that acts as a user ID, similar to an e-mail address. The SIP servers map their users' SIP URIs to the users' current IP addresses.

This enables initiating a session with a user whose IP address is unknown. However, perhaps an equally profound

benefit is the *management of identities*. The same identity can be used across a multitude of IP-based applications and be bound to an entity regardless of location. The SIP URI is a “phone number” not restricted to a single type of application.

The management of sessions and identities is also possible with P2PSIP. P2PSIP allows legacy SIP-based terminals and applications to work on top of a P2PSIP network, by storing the <SIP URI, IP address> pairs in the overlay. An “adapter” above a P2PSIP protocol stack allows access from SIP-based applications. To summarize the overlapping functionality of SIP and P2PSIP, both provide an identification and location framework that is well suited for application-level, not to forget human-level, usage. Abstracting away the IPv4 or IPv6 network addresses, these systems provide the applications a way of contacting another “client” node for a session, if the remote user-ID is known to the initiator of the call.

In addition to this, P2PSIP provides a general-purpose tool for the decentralized storage and discovery of data resources in mobile and fixed-line networks, for mediated exchange.

Most of the planned P2PSIP systems are *structured*, i.e. DHT-based, peer-to-peer systems. DHTs are algorithms for building a decentralized storage and message routing system. In a DHT, nodes and resources are arranged by their numeric IDs in a logical overlay network according to a mathematical relationship between the IDs. DHT systems are widely applied; besides P2P routing over IP networks, they can be utilized for node discovery in dynamic ad-hoc networks [8].

Resource storage can also be realized with client-server SIP. Data can be transferred in message bodies or over a separate TCP connection, with signaling based on SIP methods such as PUBLISH and SUBSCRIBE. This would, however, be a system-specific addition (with dedicated storage sites) and not a core property of SIP.

Being decentralized, P2PSIP is robust against host failures. While a SIP server forms a single point of failure, P2PSIP, with its data replication and dynamic routing mechanisms, is able to maintain its operation in the unreliable Internet, at the cost of a higher overall complexity.

As its functionalities are not limited to support only session management, P2PSIP is an attempt to *standardize P2P networking* for more generic usages. This includes the use of mediated information exchange patterns. P2PSIP also brings unified user identities into areas where it was not previously considered feasible. For example, P2PSIP allows for professional or leisure-time scenarios where a group of wireless P2P users collaborates for achieving a common goal in the digital world, such as the creation of a presentation, or shares real-life activities of the group with media snapshots.

While many services are possible also with client-server SIP at the level of “sending an invitation to a remote user-ID”, SIP services requiring mediated information exchange are problematic. Is it needed to set up a dedicated storage site for every service? How much capacity should be allocated? As P2PSIP inherently provides mediated exchange features, and at the same time requires minimal maintenance effort, it significantly lowers the threshold for providing rich services.

The P2P-style approach allows users to form interest groups in an ad-hoc manner without the need for external service provisioning. If a P2PSIP overlay is set up among a

group of trusted mobile users, the users can share with each other their personal content, or other trusted information, such as context information. Client-server SIP extensions exist for instant messaging and context, but P2PSIP provides all this using the shared intelligence of the users’ personal devices.

P2PSIP is not the first solution to the mentioned technical problems; it will not immediately replace the plethora of different P2P protocols in the Internet. Thus, one challenge faced by P2PSIP is to define a common way of interaction with *already deployed* applications. The ability to use the new standardized P2P framework through a “plug-in” for existing software would motivate users to take P2PSIP into use.

B. Security and Reliability

When comparing P2PSIP to client-server SIP, potential issues that deserve a lot of attention are the security and reliability of the framework. They have an impact on how critical applications can be built using the protocol. Security is key in P2PSIP-based systems. Due to the distributed, largely uncontrollable message routing scheme of P2PSIP networks, a rogue node might be able to corrupt severely the routing or other mechanisms of an overlay. Threats such as voice spam motivate the creation of security mechanisms.

In [9], P2PSIP security threats and requirements for counteracting them are identified. Threats include, among other things, discarding service requests, corrupting the data the node is responsible for, inappropriate usage of the services that depend on P2PSIP signaling, the alteration of messages sent between other nodes, and the provision of compromised services such as a SIP proxy set up for eavesdropping calls.

Countermeasure requirements include the encryption of communication, control over the admission of new nodes and identities, signatures for verifying the origin of P2PSIP messages, and the possibility to expel malicious nodes from the overlay. The document does not provide solutions but identifies problems to be tackled. In [10] however, a secure P2PSIP name service solution is outlined. The solution contains mechanisms that help to counteract at least Sybil attacks (i.e. the creation of a large number of peer-IDs), Eclipse attacks (i.e. the selection of a specific peer-ID), the malicious modification of data resources, and the flooding of publish messages into the overlay.

C. Considerations for Mobile Usage

A node in a P2PSIP network can be a *client* (analogous to a SIP user agent) that only consumes the services of the P2PSIP overlay, or it can be a *peer* (full-fledged participant in the distributed overlay) that provides the message routing and data storage services upon which the clients, and the entire P2PSIP network including the other peers, depend.

Somewhat similar to mobile clients in client-server SIP, mobile P2PSIP terminals with their scarce resources will often operate as clients. Wireless peers however are needed, for example, when an overlay needs to be created quickly for temporary use, or a long-term overlay is needed for purposes such as for a group of hobbyists or travelling business users.

It should be noted that all technically possible usages do not need to be available in all live P2PSIP networks. Thanks

to the configurability of P2PSIP, a policy may for example limit the size of shared resources in a mobile-node overlay.

For practical deployment of mobile P2PSIP, a mobile terminal, within the limits of its battery capacity and other factors, must be able to function as a P2PSIP peer for a reasonable amount of time. The general feasibility of mobile P2PSIP usage is discussed and demonstrated in [4] and [11]. The results from [12] demonstrate the battery life of an actual mobile P2PSIP peer node; with the evaluated implementation, battery life of 4 to 6 hours is attainable for a peer.

Wireless usage of P2P protocols is different from usage in fixed-line networks. The conditions in the radio layer affect the performance. For example, near a base station in a city center, the radio environment is notably different from that in a sparsely populated rural area. Several radio channel properties, such as multipath propagation, shadowing, and path loss are reflected in the upper layer protocols' operation.

The radio environment, along with vertical handovers where the radio interface in use is changed on the fly, tend to cause frequent disconnections for mobile peers. Moreover, due to power saving requirements, a mobile device cannot be online for long periods while on battery power. Together these factors increase *churn*. Churn is used as a parameter in our simulations in section IV; the selected values represent a high amount of churn in order to simulate a challenging network environment with a large number of mobile nodes.

III. IMPLEMENTED MOBILE P2PSIP SYSTEM PROTOTYPE

A. Mobile P2PSIP Stack

For empirical evaluation, we have implemented a prototype *Peer-to-Peer Protocol* (P2PP) stack for mobile devices. The P2PP draft was one of P2PSIP's peer protocol candidates in the IETF; it was later merged to the RELOAD draft. The merged protocol (RELOAD) has been approved as the base protocol for P2PSIP. Our mobile P2PP implementation is called *MP2PP*. It has been written in cross-platform C++ that works in Symbian OS, in mobile Maemo Linux on the Nokia 810 Internet Tablet, as well as in desktop or server Linux.

Thus, equivalent MP2PP instances can be tested on heterogeneous terminals. This enables studying performance in cases where, for example, an overlay has a variable ratio of fixed vs. mobile nodes. MP2PP also supports pluggable peer protocols and pluggable DHT algorithms; currently it only supports the P2PP protocol and the Kademia [13] algorithm.

B. GroupCalendar: Proof-of-Concept P2PSIP Application

As an example of applications that benefit from a general-purpose P2P framework, we present our mobile application GroupCalendar, which is a collaboration enabler: members of a user group share calendar events that bear significance within the group (e.g. a hobby group or a work team), where the schedules of the users must be synchronized. The software utilizes the MP2PP protocol stack and runs on Maemo Linux.

When a user creates a new calendar event, it is published as an XML-encoded resource object into the P2PSIP overlay. Changes appear in the calendar views of the other members of the group, as their applications learn about the new event-

object. A user can belong to more than one user-group. GroupCalendar uses storage and messaging services through the MP2PP library. Benefits of decentralized operation include the lack of a single point of failure and the lack of a dedicated storage server, as the responsibility of data storage is transparently shared between the participating mobile hosts.

GroupCalendar can be coupled with real-time features, such as opening a voice conference for group members in the context of a calendar event; thus GroupCalendar demonstrates P2PSIP as a unified enabler for both direct- and mediated-exchange communication. Screenshots are provided in Fig. 1.

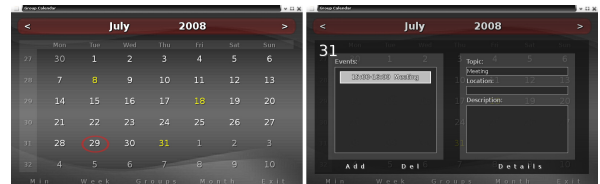


Figure 1: Month view and day view in GroupCalendar.

IV. EMPIRICAL P2PSIP DHT EVALUATION

With the MP2PP stack, we have run simulations of DHT-based P2PSIP overlays. To find out the average hop count in a realistic setting, we measured the average hop count H_{avg} with $N=200$ and $N=2000$ peers. The KeepAlive interval, routing table update interval, and resource publish interval were 10s, 30s, and 30s respectively. There were two variable parameters: the time how long a peer stays online or offline (exponentially distributed with mean value t_{churn} [s]) and the time t_{lookup} [s] between the lookup requests that are sent at a steady rate by every peer to random target peers. It should be noted that a lower t_{churn} means higher churn (plenty of joining and leaving). Each simulation with a specific combination of parameters was run for 30 minutes on a Linux server array.

The hop counts H_{avg} for join, publish, and lookup messages are shown in Fig. 2. The observed H_{avg} is clearly lower than the worst-case hop count $\log_2(N)$. The DHT works well also in the sense that when N is fixed, H_{avg} varies only little.

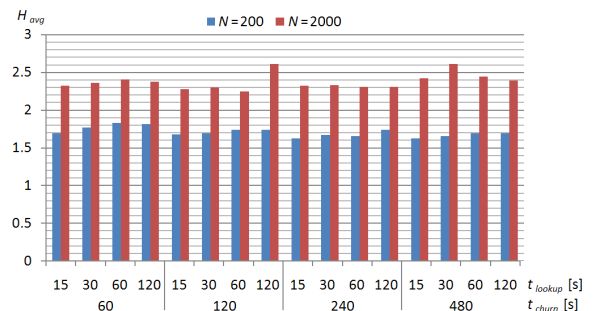


Figure 2: Average DHT hop count with varying parameters.

V. ANALYTICAL P2PSIP DHT EVALUATION

A. Scalability of Application Data and Session Messaging

The storage services of client-server SIP are essentially limited to the maintenance of node contact information, while

P2PSIP inherently provides a scalable distributed database. This affects the load inflicted on the nodes of a network.

Let us consider a scenario where an application of N users stores relatively large amounts of data online. With client-server SIP, a separate storage service is needed. Let us assume it is co-located with the SIP server. This centralized service stores B bytes of data in total, on one (server) node.

If the same application is realized using a P2PSIP overlay, storing the same amount of data, $a_{peers}\%$ of the total N users being peers (not clients), and each data item being replicated r times, the amount of data stored by each peer on average is

$$B_{peer} = \frac{rB}{\frac{a_{peers}}{100}N}. \quad (1)$$

With an example overlay of 10,000 nodes online, 10% participation as peers (90% as clients), and 2-fold replication, the data stored on one peer is 0.002 times the amount stored on a centralized data server. As shown here, P2PSIP offers a good distribution of storage responsibility, removing the need for a single relatively large-capacity storage site.

Another numeric metric to evaluate is the amount of network traffic in a given area (e.g. subnet or domain) of the network in a case where an application repeatedly needs to resolve different SIP URIs to entity addresses. The action of client-server SIP goes as follows in a case with a single domain and a single mobile operator. The initiating SIP client sends the message to the domain's proxy, which then issues a request to the domain's registrar server that knows the target's IP address. After retrieving the IP address, the proxy forwards the original request to the target client. There is a response for each request. No re-directs are used. We also assume that there is a one-to-one relationship between SIP requests and responses (which would not be the case if non-terminating responses and INVITE ACKs were involved).

In the case of P2PSIP we assume the DHT to be Kademlia-based [13], thus the maximum hop count is $\log_2(N)$. Messages are distributed evenly among all network areas, if peers are well distributed. Let there be an average of N_{sub} peers in one network area. Moreover, Q queries per time unit are made in the overlay, and maintenance traffic (KeepAlives, routing table updates) sent by each peer is M messages per time unit. The resulting traffic load in terms of the number of P2PSIP messages sent per time unit in a given area of the network is

$$n_{message-sub} = N_{sub} \left(\frac{2Q \cdot \log_2\left(\frac{a_{peers}}{100}N\right)}{\frac{a_{peers}}{100}N} + M \right). \quad (2)$$

A worst-case number of hops is assumed. The coefficient 2 in front of Q takes into account both requests and responses. To keep the comparison between client-server SIP and P2PSIP meaningful, we only consider the functionalities that are available in both SIP and P2PSIP (we do not consider, for example, the publish and lookup messages of P2PSIP).

It should be noted that the load from received traffic is also $n_{message-sub}$ per area. The (sent-)traffic load $n_{message-sub}$ inflicted on one network-area within the earlier example overlay of

10,000 nodes (1,000 peers) with a constant $M=10$, applying a varying Q and a varying N_{sub} , is depicted in Fig. 3. The graph also presents the sent-traffic load of a client-server SIP system with the corresponding amounts of activity. The graph shows that for large values of Q , the most widely dispersed P2PSIP overlays (i.e. those with a small average size of the network areas such as subnets) have the smallest per-network-area traffic load as opposed to the less dispersed P2PSIP overlays and client-server SIP.

The example parameters $M=10$, $Q=1000$, and $N_{sub}=10$, result in $n_{message-sub}=300$ in P2PSIP. This is 0.1 times the per-network-area number of messages sent in SIP ($6Q/2=3000$), with the simplifying assumption that the involved SIP nodes are in exactly two areas of the network (one large area containing all the clients plus the proxy's public network interface, and one separate network segment where the registrar is located). The P2PSIP nodes are located in multiple IP network areas, not just e.g. in the network of one operator.

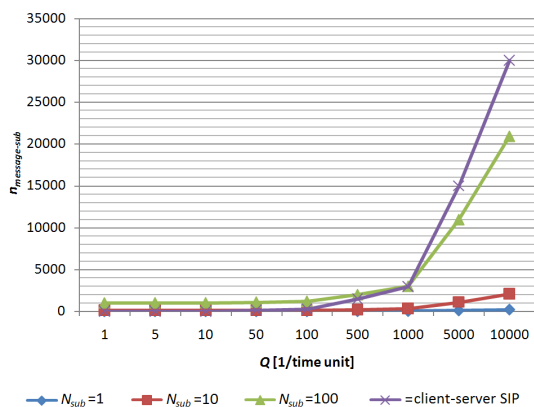


Figure 3: Number of messages sent per network area as a function of activity in P2PSIP and client-server SIP.

B. Robustness against Ungracefully Failing Nodes

While SIP servers usually have high availability, P2PSIP peers can go offline unpredictably. P2PSIP systems achieve reliability with DHT-based routing and resource replication.

If nodes leave the overlay *gracefully*, the service quality of P2PSIP stays excellent, as the node's data objects are transferred to other nodes and routing tables are updated. Overlays also recover after *ungraceful* leaves, and sufficient replication makes the permanent loss of stored resources improbable. Periodic routing table updates and KeepAlives ensure that messages seldom terminate due to a broken path.

After an ungraceful leave, all messages that are destined to the node, or would be routed through the node, are terminated. A peer with a routing table entry about the dead peer learns about the leave, when it next time sends the routing table update or KeepAlive request and gets no reply.

The probability of message terminations as a consequence of ungraceful leaves in P2PSIP can be numerically evaluated against the probability of terminations in client-server SIP.

Let us consider a case where the success of a SIP operation requires S servers up-and-running. If a server is up with probability $p_{server-up}$, the probability of success is $(p_{server-up})^S$ that is not a very interesting figure, if server downtime is rare.

However, if P2PSIP with unstable (possibly wireless) peers is used for the service, success probability is more interesting to observe. A hop-by-hop path's success probability can be determined, if we know the probability p_{term} of a peer to terminate ungracefully during a minute. Graceful leave is notified to the known peers with a Leave request; thus the following only concerns ungraceful leaves, though a graceful leave can also cause stale routing data for short time periods.

A peer sends KeepAlive messages each known peer once every t_k minutes. We assume KeepAlives to be more frequent than routing table update messages, and ignore routing table updates in our calculations. As peers are not synchronized, the average time that (at any given moment) has passed since the last successful KeepAlive to one peer is $t_k/2$. During $t_k/2$ min, the peer has left ungracefully with the probability

$$p_{termKA} = 1 - (1 - p_{term})^{t_k/2 \cdot \text{min}^{-1}}. \quad (3)$$

If an ungraceful leave happened during the $t_k/2$ min period, the other peer – from whose viewpoint we are observing the situation – has not yet learned about the leave. The reception of any message from a known peer also updates the KeepAlive timer associated with that node. We excluded this from the reasoning as the relatively short KeepAlive timer is assumed to be the dominating factor in disseminating peers' status information. Thus, based on the calculated p_{termKA} , the probability that the use of a given hop-by-hop message path succeeds (assuming the worst-case number of hops) is

$$p_{success} = (1 - p_{termKA})^{\log_{10}(\frac{a_{peers}}{100} N)}. \quad (4)$$

A graphical presentation of $p_{success}$ with a varying p_{term} and a varying number of peers in the overlay is provided in Fig. 4. In the graph, the time interval t_k between KeepAlive requests is set to be 0.167 min (10 s). There is no path parallelism: if a path fails, the message fails immediately. (The formula does not take into account that with a very high p_{term} , peers cannot gather enough stable route knowledge even to start with.)

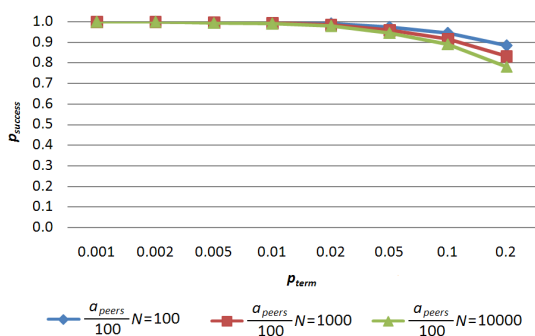


Figure 4: Probability of the successful use of a hop-by-hop path in P2PSIP as a function of p_{term} , with $t_k=0.167$ min.

Ungraceful leave events can be common in P2PSIP. This slightly speaks in favor of client-server SIP. Nevertheless, P2PSIP is usually able to remedy the overlay rapidly after a peer leaves the system, thanks to redundancy of information and periodical maintenance messages. This speaks in favor of

P2PSIP when the probability of ungraceful leaves is bearable or the use of client-server SIP is infeasible for some reason.

VI. CONCLUSION

When standardized, P2PSIP has potential to become a unified P2P solution for session and data management. We studied the properties of P2PSIP with qualitative and numerical analysis and with overlay network simulations. The results indicate that P2PSIP is a feasible technology, not only for direct-exchange, but also for mediated-exchange services.

VII. ACKNOWLEDGEMENT

This work has been financially supported by the Finnish Funding Agency for Technology and Innovation (TEKES), Ericsson, Nokia, and Nethawk.

REFERENCES

- [1] R.-H. Hwang, M.-X. Chen, and C.-J. Peng, "SSIP: Split a SIP session over multiple devices," *Computer Standards & Interfaces*, vol. 29, no. 5, pp. 531–545, July 2007.
- [2] D. A. Bryan and B. B. Lowekamp, "Decentralizing SIP," *ACM Queue*, vol. 5, no. 2, pp. 34–41, March 2007.
- [3] D. Bryan and B. Lowekamp, "SOSIMPLE: A SIP/SIMPLE based P2P VoIP and IM system," white paper, Computer Science Department, College of William and Mary, Williamsburg, VA, USA, Nov. 2004.
- [4] M. Matuszewski and E. Kokkonen, "Mobile P2PSIP: Peer-to-Peer SIP communication in mobile communities," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, 2008, pp. 1159–1165.
- [5] J. A. Pouwelse, P. Garback, D. H. J. Epema, and H. J. Sips, "The BitTorrent P2P file-sharing system: Measurements and analysis," *Lecture Notes in Computer Science*, vol. 3640, pp. 205–216, Nov. 2005.
- [6] T. Kunz, A. Gaddah, and L. Li, "Mobility support in a P2P system for publish/subscribe applications," *B.-C. Seet (ed.) Mobile Peer-to-Peer Computing for Next Generation Distributed Environments: Advancing Conceptual and Algorithmic Applications*. IGI Publish, Hershey, USA, pp. 68–93.
- [7] J. Li, J. Stribling, T. M. Gil, R. Morris, and M. F. Kaashoek, "Comparing the performance of distributed hash tables under churn," *Lecture Notes in Computer Science*, vol. 3279, pp. 87–99, Jan. 2005.
- [8] M. Mani, W. Seah, and N. Crespi, "Super nodes positioning for P2P IP telephony over wireless ad-hoc networks," in *Proceedings of the 6th International Conference on Mobile and Ubiquitous Multimedia*, 2007, pp. 84–89.
- [9] H. Song, M. Matuszewski, and D. York, "Security requirements in Peer-to-Peer Session Initiation Protocol (P2PSIP)," IETF Internet Draft, work in progress, Nov. 2008.
- [10] I. Baumgart, "P2PNS: A secure distributed name service for P2PSIP," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, 2008, pp. 480–485.
- [11] E. Kokkonen, S. Baset, and M. Matuszewski, "Demonstration of Peer-to-Peer Session Initiation Protocol (P2PSIP) in the mobile environment," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, 2008, pp. 1221–1222.
- [12] I. Kelényi and J. K. Nurminen, "Energy aspects of peer cooperation—Measurements with a mobile DHT system," in *Proceedings of the Cognitive and Cooperative Wireless Networks Workshop in the IEEE International Conference on Communications*, 2008, pp. 164–168.
- [13] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, 2002, pp. 53–65.