SCOPE: A Prototype for Spontaneous P2P Social Networking

Mehdi Mani, Anh-Minh Nguyen, Noel Crespi
Wireless Networks and Multimedia Services Department
Institut TELECOM, Telecom SudParis
Evry, France
{*mehdi.mani, anh_minh.nguyen, noel.crespi*}*@it-sudparis.eu*

***Abstract*—This paper explores the architecture and implementation details of *SCOPE* platform. SCOPE is our prototype for spontaneous P2P social networking. It provides customized social networking application for local use cases. Below the network level, SCOPE relies on 802.11 ad-hoc mode and needs no infrastructure. SCOPE follows the hierarchical P2P model. Some nodes with higher computing capability become super-nodes. super-nodes form an overlay and provide the distributed data management system for the P2P social network. Client nodes (CNs) connect to super-nodes and rely on them for sharing their contents or accessing to the shared information. We have developed *What's UP* as our client agent tool which provide a vast range of applications from simple text/link sharing to P2P IP Telephony.**

***Keywords*-Implementation; Social Networking; P2P; DHT; Spontaneous Networks.**

I. Introduction

Advanced features of social networking such as content recommendation and community based environment enhance people's communications and access to the information. Along with the wave of web 2.0, plenty of social networking applications are popped out in the internet. Today, Facebook with around 300 millions active users [2] is the most visited web-site per day in the internet. Moreover, P2P technologies have been very popular during last decade and accounted around 20% of the internet traffic.

On the other hand, increasingly, people carry their treasured multimedia content in small mobile devices like smart phones and PDAs. Tremendous interesting customized applications can be imagined with combination of social networking and P2P technology to provide spontaneous social networking over mobile devices in the events/locations such as conferences, expositions, galleries, stadiums, bars and restaurants. Enabling people to share their experiences, to communicate and to have access to the comments of others without need to have internet access and with minimum required infrastructure is the main idea behind P2P spontaneous social networking. Considering, auto-organization, scalability and distributed architecture, P2P technology can play the key role in creating such kind of social networks.

SCOPE is the prototype we have developed as the core of such kind of P2P social networking. SCOPE provides the distributed data base and lookup services deployed on DHT technology. It runs on nodes with higher capabilities (super-nodes) which are connected in ad hoc 802.11 mode. Moreover, we have developed *What's UP* as our client agent (i.e. User Interface) tool. What's UP provides advanced features of social networking in a distributed architecture with no need to access to the internet. Technically speaking, SCOPE is the code run in the background of super-nodes and What's Up is our Graphical User Interface running on both of super-nodes and Client Nodes (CNs). Three different versions of our prototype is already demonstrated in different conferences [9].

The rest of the paper is organized as follows. In section II we present the SCOPE general architecture and how the P2P overlay is organized to form our social network. We explore the services that our client agent, What's UP, provides in section III. Section IV is devoted to the implementation details of SCOPE and software architecture of super-nodes. Sections V and VI discuss our experimental work for performance validations and measurements. Finally, before the conclusion, the related work will be reviewed in section VII.

II. SCOPE General Architecture

Figure1 shows the general architecture of SCOPE. SCOPE provides a distributed data management system for our P2P social network. The core of a social network, is a data management system which i) enables users to publish and share their contents; and ii) alerts the subscribed users of any updated contents/events of interest.

To construct and organize the SCOPE overlay, certain devices, among all, are selected to serve as the super-nodes. super-nodes are the devices with enough computing/storage capacity. The DHT code is run on them and they share their computing and storage resources and route the clients' requests to the proper peer. In contrary, a Client Node (CN) does not run the DHT code and connects to at least one super-node to register its ID and join the social network overlay. In continue we describe the details of SCOPE.

Our system is supposed to work on mobile devices spontaneously without any dedicated network resources. Hence, we deploy a Distributed Hash Table (DHT)to provide the distributed data management system in the lack of dedicated central resources and servers. The DHT defines the rules for information management and creates our social networking

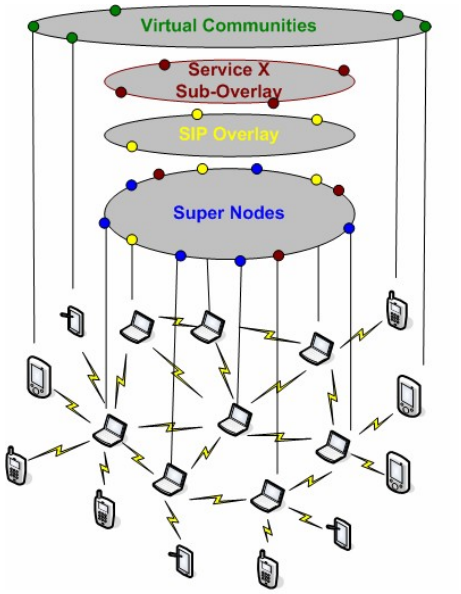

Figure 1. SCOPE General Architecture

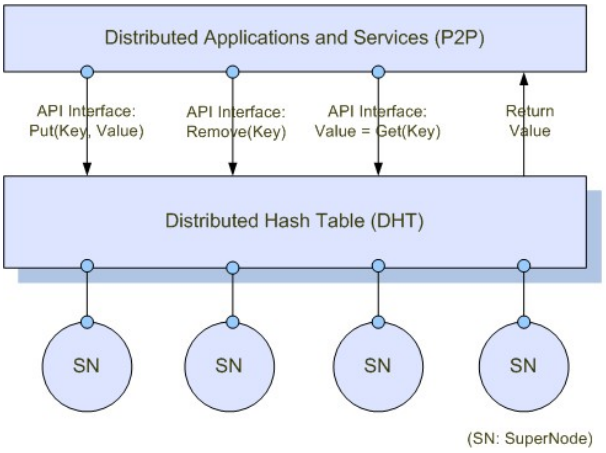

Figure 2. DHT API

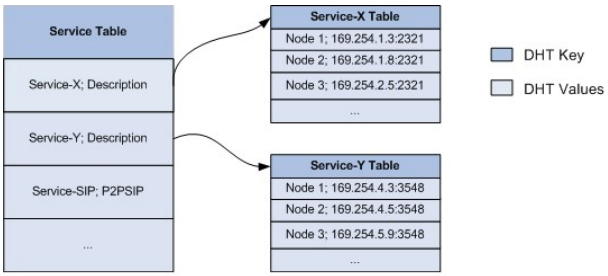

Figure 3. Service Table Structure in SCOPE

overlay. This includes the Publish/subscribe and lookup services.

A. Application Programming Interface(API)

Clients access to SCOPE DHT overlay via a set of DHT APIs as shown in Figure 2. In a DHT system, three functions of *Put*, *Get* and *Remove* provide the possibility of sharing, retrieving and removing a content, respectively. We provide these three functions using a set of APIs under XML format via XML-RPC [5] mechanism. XML is the data representation for efficient exchange of data in heterogeneous systems. Using HTTP as the transport, it is independent of programming language and platform.

To share a content (value), v, the client sends a message with the format of *put(k,v,s,t)* to a super-node. The content (v) is bound to key k and is identified by this key. The key is obtained by hashing on the value and indicates which super-node should store this content v. The time-to-live, t, indicates the lifetime of the content and can be removed before the expiration using the secret, s. Several contents may be bound to the same key and the *get(k)* function gives back a list of values associated with k. For example, the *get(k1)* returns 2 values v1 and v2 if the key pairs *(k1,v1)* and *(k1,v2)* has been already *put* in the DHT. The value of the key, k can be removed using *remove(k,v,s,t)*.

B. Service Classified Overlay

A social network may include a vast range of services from simple content sharing like link sharing to advanced multimedia services like video streaming. Indeed, an interesting aspect for a social network is to enable the users to develop and share their own services. In SCOPE we implement a DHT Overlay with unified API as the core of our P2P social network. We call this the *global overlay*. Then for each specific service, we create a virtual overlay on top of the global overlay (Figure 1). We assign a unique service identifier (*Service-ID*) to each service. In the global overlay, a list called *global service list* includes all of the *Service-ID*s. Then there is a dedicated list for each service Id. This list points to all the nodes which provide this service.

A user is able to get the list of all the services provided in the P2P social network by sending a request *Get(k=HService List)* to the global overlay. When a node wants to join a service overlay (or look for a specific service such as a SIP proxy), it looks up for specific *Service-ID* and obtains the list of nodes that provide this service. The *get* request regarding this lookup will be *Get(k=Service ID*.

To form a new service overlay, a service node puts new *Service-ID* with its contact address (Put(k=Service-ID, v=contact address)) into global overlay. In case that this node joins to the existent service overlay it will update Service Table and add its address into the list of contact nodes for this specific service. For example a node with IP address 169.254.3.1, which is able to provide SIP services via the port 5061, calls function *Put(k="P2PSIP",v="169.254.3.1:5061")* via XML-RPC in order to advertise itself on global overlay.

In SCOPE, we provide communication services (IP Telephony) based on IETF P2PSIP draft [6]. The communication services include VoIP, instant messaging, presence and video call. We explore the details of SCOPE implementation in section IV.

III. Client Agent: What's UP

The What's UP client agent will be installed on all the users including super-nodes and CNs and allow them to benefit from a vast range of P2P social networking services. The list of the provided services is listed in following.

- **P2P Communications**: IP Telephony services in P2P mode including VoIP, Video call, presence and Instant Messaging.
- **V-Card Sharing**: Sharing the business card and enabling the possibility to be found based on the area of the interest, profile and company name.
- **Content Sharing**: Text, link and image.
- **P2P Community management**: Creating a community, joining a community, adding news, subscribing for the events, receiving event alerts, receiving community feeds, commenting and polling.
- **Search**: finding people, content, service and community.

A snapshot of What's Up menu is depicted in Figures 4.

All the requests a user submit to the DHT overlay via What's Up client agent will be transformed to Put/Get messages. This includes all the activities such as creating user profile, setting up a community, sending a messages, searching a content etc. In the time of a profile creation, User-name is used as the DHT key and user's information considered as the DHT value. Then any other activity such as joining a community, creating a community and subscribing for a service will be bound to the user profile. Figure 5 shows how information lists are organized in our DHT system. This Figure shows how all the distributed information concerning a specific user are bound to its profile.

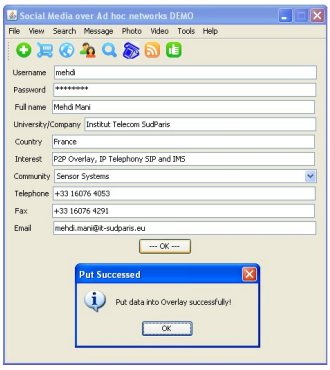

Figure 4. Create a Profile

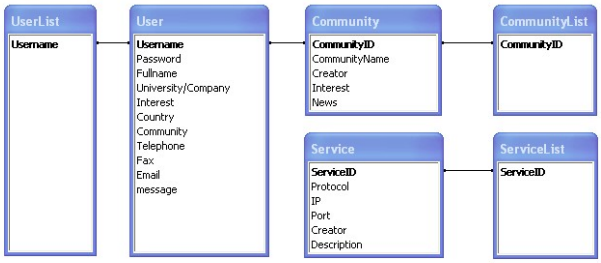

Figure 5. Structure of Information Lists in SCOPE

IV. SCOPE Implementation Details

In our platform, DHT functionality relies on Bamboo open source [1]. Bamboo with its public name OpenDHT [11] has been deployed on Planetlab [3] with hundreds nodes worldwide. The geometry and routing algorithm in Bamboo are identical to Pastry [12]. We have integrated Bamboo open source in the scope super-node code to provide a distributed lookup system based on DHT over the peers' devices. We have modified Bamboo codes based on the researches presented in [10] to adapt it to Ad-hoc networks. Hence SCOPE is able to work efficiently on Ad-hoc networks as well as conventional network configurations.

Furthermore, in SCOPE, we provide the communication services using P2PSIP based on the IETF draft in [6]. We develop the P2P SIP proxy codes for the super-nodes. This code run on some super-nodes and create the communication service virtual overlay.

In this section, we focus on software architecture and implementation deployed on our super-nodes.

A. super-nodes with DHT and P2PSIP Overlay

The software architecture is illustrated in Figure 6. DHT API in XML-RPC provides put/get and remove *(key,value)* pair. HTTP Handler (WebInterface) in Figure 6 is a software module which listens and receives requests from CNs. Upon receiving a request, this module calls DataManager and DHT router module to route (key,value) pair to the correct super-node. HTTP handler can also be seen as a very small web server. StorageManager is responsible to store data to Berkeley DB. Finally, the data is stored to hard disk. DHT routing is a complex module implemented in Bamboo DHT; it is identical to Pastry [12]. As such, it have to manage leaf set and routing table in DHT algorithm.

To route messages through the super-nodes, UDP is used to encapsulate and transfer data. Since UDP lacks congestion control, UDPCC [8] (UDP with Congestion Control) has integrated to DHT software to control the congestion of these UDP packets for better performance. This has been implemented in Bamboo DHT that we utilize.

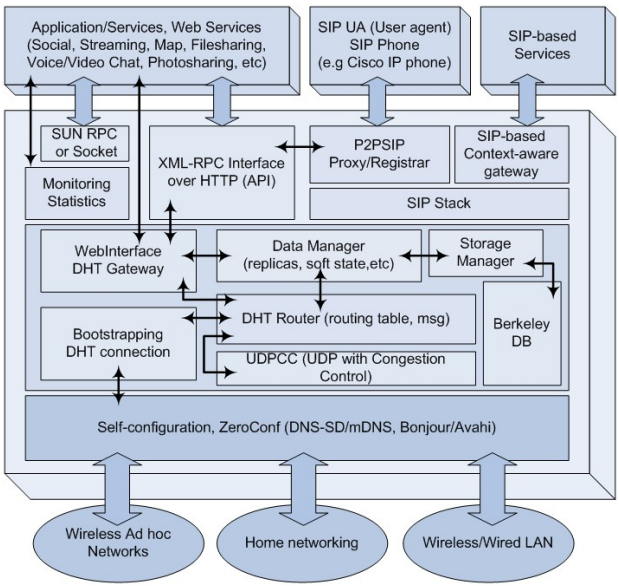

Figure 6. Super-node Software Architecture in SCOPE

B. Communication Services Virtual Overlay

In our platform, we propose and implement a Hybrid P2P SIP architecture which provides P2PSIP distributed Proxy/Registrar and form a communication service overlay. The idea of hybrid P2PSIP is to provide SIP based services not only for P2PSIP clients with put/get API but also for conventional SIP user agents. This enables users to use any traditional SIPUA (*(SIP User agent)* for IP telephony services. On the other hand the users with P2PSIP UAs, will initiate the IP Telephony sessions based on Put/Get DHT signalling.

Assume that Bob is using a traditional SIP UA. First, Bob needs to register its SIP identifier in the overlay. He sends a SIP register request to a SIP super-node. This SIP super-node transforms this request to a DHT Put request and register the SIP identifier of Bob with a super-node based on the DHT rules (Figures 7). If Bob was a P2P SIP UA, he would directly send a put request to register his identifier.

To initiate a call with Alice, Bob sends an INVITE message to the SIP super-node. This request will be transformed to a get request with "key= Alice SIP URI" and forwarded to the super-node with which Alice has registered her SIP identifier. The ip address and port number of Alice will be returned to the SIP super-node as the requested value. Then this SIP super-node sends an INVITE message to Alice and initiate the call for Bob. A P2PSIP UA can send a get request and obtain the IP address of the callee directly.

V. DHT Put/Get Experimental Performance Validation

In this section we present our experimental results to validate the performance of SCOPE. Considering the fact

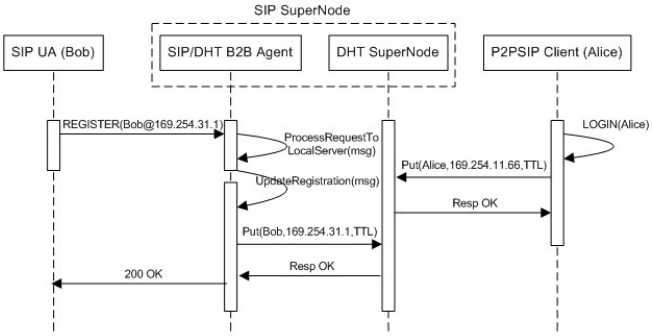

Figure 7. Register to P2PSIP Proxy/Registrar

that all users activities in our p2p social networking system will be translated to a *put* or *get* request, we need to make sure that this process is working correctly in our prototype. As previously mentioned, each data in our architecture is put to DHT overlay with a *(key,value)* pair. The value should be routed and stored in the correct super-node whose hostID is numerically closest to the key. In this section we present the results of our experiments run to make sure that this process of Put is working properly in our system.

For our experiment, we setup an overlay with 11 super-nodes. These super-nodes are configured to work on 802.11 ad-hoc mode and are located randomly in an area of 4x10 m^2. Our super-node software presented in section IV.A is run on these super-nodes and provide DHT and P2PSIP services. We assign a unique identifier to each super-node by hashing its IP address.

We registered 200 clients in this system. The clients interact with the system and use the social networking services that What's UP provides for them. We trace all the Put/Get requests and validate that the values are correctly placed/retrieved in the DHT.

The DHT-put operation in Figure 8 shows that a request is received via XML-RPC interface at DHT gateway with key *0x249450cb* and *103 bytes* of data value associated with *TTL=3600 seconds*. DHT-put latency from starting the put request untill get the response for data value took *84 ms*

DHT Lookup for this key via a DHT-get is presented in Figure 9. The key *0x249450cb* is return from super-node *0x19d5e887* whose Host ID is the closest to the key in *DHT ID Space*.

Figure 10 shows a case where a user on the device with IP address *"192.168.0.14"* registers his SIP identifier to a P2PSIP super-node. P2PSIP super-node parses *SIP REGISTER* message and get SIP ID of the user from *Contact field* in *SIP Header*. Then, the P2PSIP super-node hashes User's SIP ID and obtain k=*"0x2feb35c4"*. Then this key associated with the value *"192.168.0.14:7168"* is put in the DHT overlay.

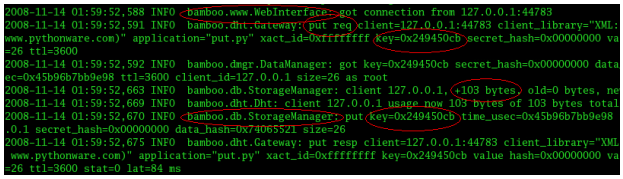

Figure 8. Put(key,value) pair

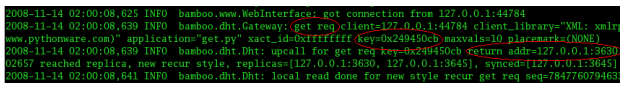

Figure 9. Get(key) returns values

VI. Performance Measurement

Each Put/Get message should be routed to the super-node which hosts the requested value. Each super-node upon receiving a request, verify if the value is hosted (for a Get request) or should be hosted (for a Put request) in it. If the super-node is not responsible for this value, it will relay the request to a neighbor super-node according to the DHT rules. In a Get request, this relaying process makes users experience a random delay before they retrieve the requested content. On the other hand, in a Put process, the new content (value) will not be available to other users before this period of relaying delay. This delay, which we call it *DHT latency*, could be critical for some time sensitive applications we have developed in our social network such as IP Telephony (communication services). In these applications, *session setup delay* is defined to qualify how good they are serving users. *Session setup delay* indicates the delay between the time that a session setup request is submitted and the time that the session is established; it is obviously affected by DHT latency.

We carry out the measurements to see the impact of DHT latency on session setup delay (Get) and registration time (Put). We set up our platform with 9 laptops connected with each other in Wireless Ad-hoc 802.11g. The laptops form a 3x3 grid network with 6 meters of distance between two adjacent nodes. Each laptop is able to serve as up to 8 virtual machine (super-nodes). With this possibility, we increase the number of super-nodes from 9 to 72 to see the impact of the number of super-nodes on DHT latency and session setup delay.

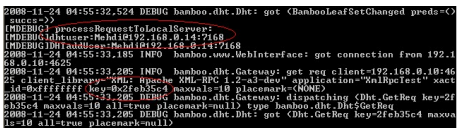

Figure 10. P2PSIP super-nodes parse SIP message and put user's IP address to DHT

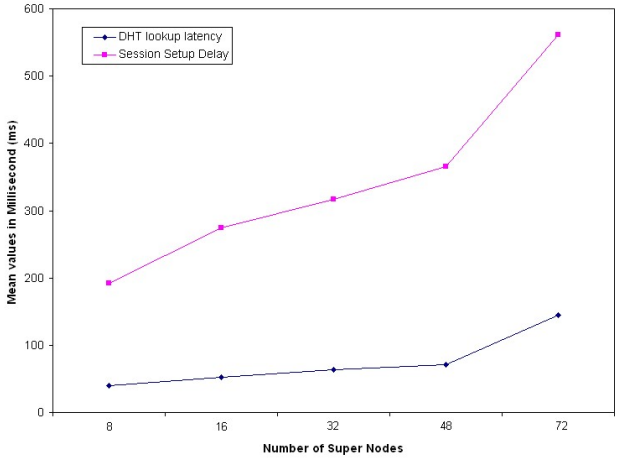

Figure 11. Measurement of Session Setup Delay in P2PSIP scenario

A. Session Setup Delay

Session Setup Delay is an important metric for IP Telephony service. Technically speaking, for SIP signalling. The session setup delay is the time elapsed between the time that the caller submit the INVITE request and the time that he receives the Ringing message.

The measurement results for DHT latency and session setup delay is displayed in Figure. 11. As expected, when the number of super-node increase, DHT lookup latency increase and affects the Session Setup Delay. In a DHT overlay, the lookup hop is O(logN) where is the number of super-nodes. In our measurements, according to our test-bed environment, with the increase in the number of super-nodes from 9 to 72, the DHT latency augments from 8 to 32 ms and occupies about 19.8% of Session Setup Delay time.

In a real system, many factors affects on DHT latency and eventually on Session Setup Delay. In following, we give a list of these factors.

- Mean DHT lookup hop count
- The number of ClienNodes (CNs)
- The path length between super-nodes
- The parallel calls between CNs at the same time
- The number of Services in the network using the same DHT overlay
- Position and hit-rate of super-node in the DHT id. space
- The link failure probability and the quality of connectivity
- The load balance of key distributes over DHT overlay

B. Registration Delay

Similar to session setup delay, registration delay is also affected by DHT latency. When a SIP User Agent submits a register request to a P2PSIP super-nodes, it takes a while to get the confirmation *SIP message "200 OK"*. This process directly depends on DHT-put latency. Although Registration

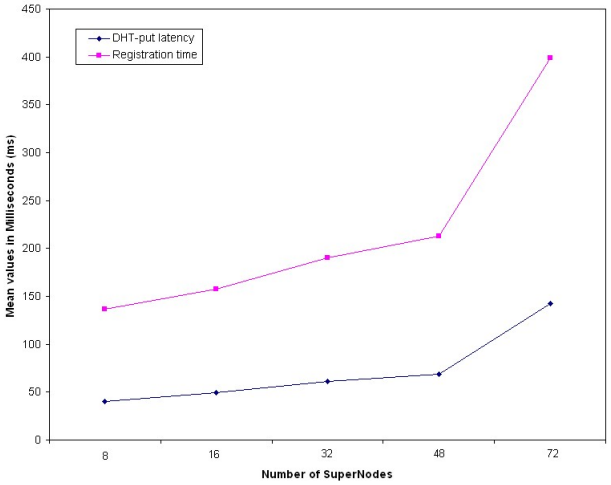

Figure 12. Measurement of Registration time

delay is not as important as Session Setup Delay, however it also reflects the performance quality of the social network and specially SIP-based Services.

Figure 12 depicts the measured results. Similar to Session Setup Delay, when the number of super-nodes increase, DHT-put latency increase and influence on Registration Time. In our measure according to our testbed, the DHT-put latency occupies about 32.1% of Registration time.

VII. Related Work

A platform of middleware and user interaction tool, called MyNet, is proposed in [7]. It allows users to share their device, services and content without requiring central service support. MyNet use its own "MyNet messaging layer" to propagate update messages. Although there is not much detail about how MyNet messaging layer works, but it seems that it works like a simple multicast group. When there is a update message, the messaging layer queues the update and sends it to the list of users that should receive it. This makes MyNet unscalable and limits its use-case for limited social interactions.

OpenVoIP [13] is another P2P platform to provide VoIP and Instant Messaging (IM) services. OpenVoIP is a 500 node overlay deployed on PlanetLab [3]. This platform is created to run on the internet and focus on the aspects like robustness against churn and NAT and Firewall traversal.

Turtle [4] is a prototype for sharing small contents based on special type of peer-to-peer network in which all your communication goes only to your friends, and then to their friends, and so on, to the ultimate destination. This type of networks are called friend-to-friend (F2F) networks. Turtule enables simple content sharing and lookup.

VIII. Conclusion

In this paper we explored how SCOPE provides a P2P and spontaneous solution for social networking in local areas. SCOPE deploys DHT functionalities to form a distributed data management system as the core of our P2P social network. Moreover to be able to provide session based communication services, P2PSIP register functionalities are implemented in super-nodes. Our client agent "What's UP" connects provides a rich menu of social networking services from simple link/text sharing to P2P IP Telephony. What's UP, via the xml-rpc API, connects the usr to the DHT overlay.

References

[1] The bamboo distributed hash table - a robust, open-source dht, http://bamboo-dht.org.

[2] Facebook statistics, www.facebook.com/press/info.php?statistics.

[3] Planetlab, an open platform for developing, deploying and accessing planetary-scale services, http://www.planet-lab.org.

[4] Turtle f2f open source, http://www.turtle4privacy.org/new/.

[5] Xml-rpc specifications, http://www.xmlrpc.com.

[6] Resource location and discovery (reload) base protocol, draft-ietf-p2psip-base-06, November 2009.

[7] D.N. Kalofonos, Z. Antoniou, F.D. Reynolds, M. Van-Kleek, J. Strauss, and P. Wisner. Mynet: A platform for secure p2p personal and social networking services. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 135–146, March 2008.

[8] E. Kohler, M. Handley, and S. Floyd. Designing DCCP: congestion control without reliability. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–38. ACM New York, NY, USA, 2006.

[9] M. Mani, A.-M. Ngyuen, and N. Crespi. What's up 2.0: P2p spontaneous social networking. In *INFOCOM 2009, IEEE*, pages 1–2, April 2009.

[10] Mehdi Mani, Winston Seah, and Noël Crespi. Super nodes positioning for p2p ip telephony over wireless ad-hoc networks. In *MUM '07: Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*, pages 84–89, New York, NY, USA, 2007. ACM.

[11] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: a public DHT service and its uses. *ACM SIGCOMM Computer Communication Review*, 35(4):73–84, 2005.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. *Lecture Notes In Computer Science*, 2218:329–350, 2001.

[13] Gaurav Gupta Salman A. Baset and Henning Schulzrinne. Openvoip: An open peer-to-peer voip and im system. In *SIGCOMM'08 (demo)*, August 2008.