Case Study on a Community-Centric Mobile Service Environment

Timo Koskela, Otso Kassinen, Erkki Harjula, Jani Pellikka, and Mika Ylianttila
MediaTeam Oulu research group
Department of Electrical and Information Engineering
University of Oulu
FIN-90014 University of Oulu
Finland
firstname.lastname@ee.oulu.fi

ABSTRACT: *This paper discusses and evaluates the design and the implementation of a Community-centric Mobile Service Environment (COMSE), where all services are developed with Web technologies. The service environment is built on a distributed communication platform that is implemented using a DHT-based peer-to-peer (P2P) protocol. The interaction between P2P overlay networks and Web-based services is realised with a mobile middleware component called P2P Daemon. By using P2P Daemon as a gateway, Web-based services are able to utilise P2P overlay networks for community management, service publishing and discovery, as well as data storage. This case study highlights the opportunities of using P2P and Web technologies collaboratively in mobile settings. Special focus is on technical evaluation of the COMSE with two pilot services that were tested by genuine users in a real usage environment. As performance criteria, we used nodes' messaging load, success ratio of P2P operations, and latencies of successful P2P operations. The evaluation results show that it is technically feasible for mobile nodes to use Web-based services in community-centric P2P overlay networks. Our findings suggest that the load inflicted on mobile nodes for maintaining small-sized P2P overlay networks is at an acceptable level, since the mobile nodes are able to stay online up to 8 hours before running out of battery.*

Keywords: context information, mobile middleware, peer-to-peer, user communities, Web-based services.

Received: 12 May 2010, Revised 3 June 2010, Accepted 10 June 2010

1. Introduction

Peer-to-peer (P2P) networking has evolved from *unstructured* systems [1] towards *structured* systems [2], where the nodes and their resources are organised into a global data structure according to some rules that are usually based on algorithms called *distributed hash tables* (DHTs) [2]. At the same time, *pure P2P* systems have been supplemented with *hybrid P2P* solutions, where the advantages of both P2P and client-server architectures have been utilised [4][5]. In spite of the flexibility of P2P networking, enormously popular file sharing is sometimes considered its only application [5][6]. Furthermore, P2P has often been treated as a technology that is a complete opposite of the traditional client-server networking paradigm [7]. The given generalisations offer only a limited understanding of P2P networking and clearly exclude the usage of P2P for communication and collaboration, distributed computing, Internet service support, and database system purposes [2][8]. A broader definition given in [2] regards P2P networks as *distributed systems consisting of interconnected nodes that are able to self-organise into network topologies with the purpose of sharing resources*. These resources are not limited to content, but also include services, CPU cycles, and bandwidth, to mention some examples.

In the context of P2P networking, the resource-limited mobile environment has been recently paid a lot of attention [8][9][10][11]. The mobile nodes with limited processing power, bandwidth, and power supply set extensive demands for the implementation of P2P systems, especially for the power management in order to keep the battery life feasible [10][11]. In addition, mobile nodes tend to participate in the network in a transient manner [9], which complicates the collaborative network-maintenance activity, and thus decreases the smoothness of operation in P2P networks. It is also important to distinguish mobile P2P from mobile ad hoc networking (MANET) that occurs at the lower layers of the OSI model [12]. The definition of P2P systems, given in [2], does not exclude MANETs, but in this paper, the term mobile P2P is only used to refer to application layer P2P overlay networking.

The recent research on the collaborative usage of P2P and Web technologies has primarily focused on (1) Web service provision in fixed-line P2P environments (e.g. [13][14][15][16][17][18]), where P2P technologies are used for replacing the centralised Web service repository (e.g. UDDI [17]) in order to increase the scalability and robustness of the Web service provision; (2) P2P-based caching of Web content (e.g. [19][20][21]), where P2P technologies are used for providing replicas of Web content in order to reduce the access latency and the load on Web servers; and (3) access to P2P overlay networks using a Web browser (e.g. [22][23][24][25]), where a Web-based application, in some way, interfaces with P2P overlay networks or the P2P functionality is integrated directly in a Web browser. However, the collaborative usage of P2P and Web technologies for providing community-centric services, especially in mobile environments, has not been paid attention. The usage of P2P technologies in the development of community-centric services seems an inherent choice, as the nodes in an overlay network may correspond to community members.

In this paper, we present a *Community-centric Mobile Service Environment* (COMSE), a hybrid P2P solution that provides an example case how P2P and Web technologies can be collaboratively utilised for community and service management in mobile environments. The COMSE is built on a distributed communication platform that is implemented using a DHT-based P2P protocol. The usage of P2P overlay networks improves fairness of load distribution, scalability and fault tolerance of the COMSE, whereas the usage of Web-based services enables easy development, reuse and composition of platform-independent mobile services. The performance of the COMSE is evaluated with two pilot services that were tested with genuine users in a real usage environment. As evaluation criteria, we use (1) nodes' messaging load; (2) success ratio of P2P operations; and (3) latencies of successful P2P operations. This paper mainly introduces new results on the COMSE, but also summarises some of our previous work [26][27][28][29] in order to provide a full picture of the developed mobile service environment.

The rest of the paper is structured as follows: Section 2 provides an overview of the COMSE, Section 3 presents the mobile middleware called P2P Daemon, Section 4 presents five management components built as part of P2P Daemon, Section 5 introduces some monitoring and management tools, Section 6 presents the two pilot services, Section 7 provides the performance evaluation of the COMSE followed by conclusions and discussion in Section 8.

2. The Community-centric Mobile Service Environment

The developed prototype of the COMSE relies on a distributed communication platform that is implemented using a popular DHT algorithm called Kademlia [30] and a DHT-based P2P protocol called P2PP. The DHT-based scheme was chosen to avoid burdensome flooding of the query messages and to provide deterministic resolution of queries [26]. Our implementation of the P2PP protocol conforms to the Internet Engineering Task Force (IETF) P2PP draft version -01 [31] that was later merged into another P2P protocol draft called RELOAD [32]. Since the implementation of the P2PP protocol is not in the focus of this paper, it is not presented in detail. However, some more information is available in [33].

A layered model of the COMSE is illustrated in Figure 1. On the *protocol layer* reside the P2PP protocol and Sofia-SIP stack with Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) [34]. The P2PP protocol provides low-level functions for managing P2P overlay networks and their resources, whereas the Sofia-SIP stack with SIMPLE provides low-level functions for communicating with a context service. In practise, the Sofia-SIP stack with SIMPLE implements encoding and decoding of SIMPLE-specific SIP messages that carry context information of individual users and user communities. The context service has been implemented as a hybrid solution that uses both a centralised context server and the P2P overlay networks (more details are given in Section 4.5).

On the *middleware layer* reside five management components that interface with the underlying protocol layer in order to provide high-level functions for the *service layer*. The provided high-level functions are categorised into the following management components: (1) Community Management, (2) Community Access Management, (3) Content Management, (4) Service Management, and (5) Community Context Management. The management components are part of a mobile middleware called *P2P Daemon* that is intended to run as a background process on each device participating in the COMSE. The design and implementation of P2P Daemon is presented in Section 3, and the management components in Section 4.

On the service layer reside Web-based services that utilise the high-level functions provided by the middleware layer. Nothing prevents native applications from taking advantage of the COMSE, but our pilot services were implemented only with Web technologies because of their platform independent nature. The COMSE promotes two types of services, *utility services* and

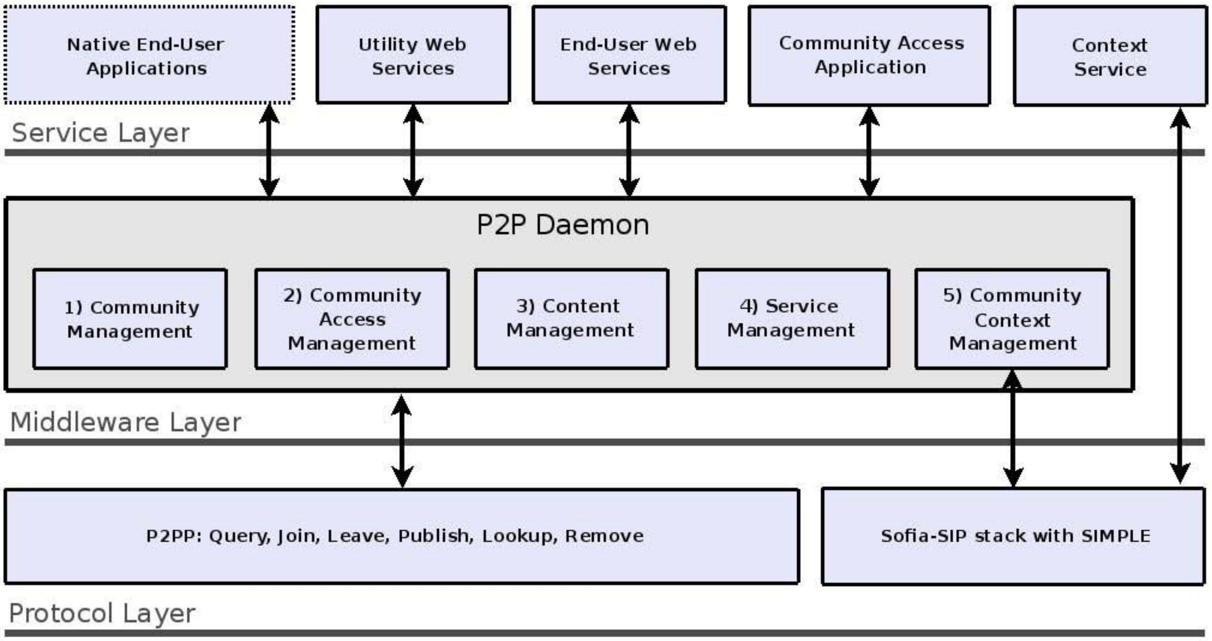

Figure 1. A layered model of the Community-centric Mobile Service Environment

end-user services, that are similar to the two service concepts (utility services and business services) of the Service Oriented Architecture (SOA) [35][36]. Utility services are not intended to provide any value to end-users as such; instead, they are intended to be used as building blocks when composing new end-user services. End-user services can be composed of utility services, but end-user services can also involve other services that are available in the Internet. According to the ideology of Web 2.0, an end-user service can be treated as a mash-up; a composition that connects together services and information from multiple sources [35]. End-user services can be either available to everyone or only to members of a particular user community.

3. Mobile Middleware

The function of the mobile middleware called P2P Daemon is to enable communication between Web-based services, P2P overlay networks and the context server. P2P Daemon implements the communication by providing the following three functionalities: (1) acting as a P2P node in one or more P2P overlay networks; (2) acting as a client for the context server; and (3) providing the communication interfaces for Web-based services. As shown in Figure 2, P2P Daemon supports both the *D-Bus* and the *local socket* communication interfaces. The D-Bus is an open source component that enables communication between local applications on Linux-based platforms [37].

The communication between a Web-based service and P2P Daemon (both the client-and server-side instances of P2P Daemon) is enabled by an interface component called *P2P Bridge*. On the *client-side*, P2P Bridge uses the D-Bus connection for implementing the communication between a Web-based service (running in a Web browser on the client) and client-side P2P Daemon. Client-side P2P Bridge has been realised with an open source D-Bus bridge library that provides the D-Bus Javascript bindings. On the *server-side*, P2P Bridge uses the local socket connection for implementing the communication between a Web-based service and server-side P2P Daemon. Server-side P2P Bridge has been realised as a PHP script. Client-side P2P Bridge also supports relaying requests to server-side P2P Bridge (and further to server-side P2P Daemon) over the HTTP protocol when needed, for instance, when client-side P2P Daemon is not running.

Since P2P Daemon interacts with P2P overlay networks and with the context server over a network connection, the communication is always burdened with some network delay. Hence, all communication between Web-based services and P2P Daemon has been implemented in an asynchronous manner. In other words, the requests initiated by Web-based services do not immediately receive responses. Especially, when the requests are subscription-based, the responses, i.e., notifications arrive only when new data is available. This sets some requirements for Web-based services, since they have to be able to receive and process the response and subscription notifications whenever they arrive. In P2P Daemon, the asynchronous communication has been implemented using threads, whereas Web-based services utilise the P2P Bridge library, which supports the asynchronous communication using Ajax.

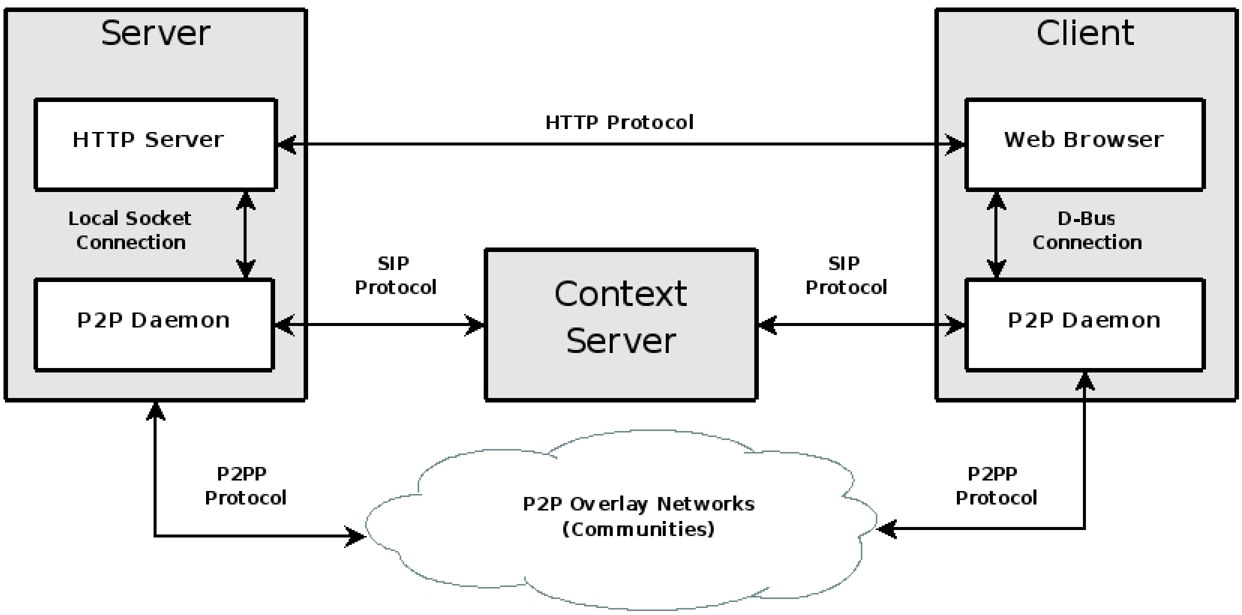

Figure 2. Communication interfaces of P2P Daemon

Normally, there is one instance of P2P Daemon running on the server-side and one instance on each of the clients that use the Web-based service. For alleviating the load on the server-side, Web-based services should utilise client-side P2P Daemon whenever possible instead of P2P Daemon running on the server-side. Server-side P2P Daemon can be used as a fallback if the client has no P2P Daemon running or in some other special situations as seen fit by the service developer. When client-side P2P Daemon is used, the interaction with P2P overlay networks is directly managed by the client without burdening the server.

A simple request-response messaging model is used in the communication between Web-based services and P2P Daemon. The same messaging model is used for both D-Bus and local socket connections. In Figure 3, our messaging model is described with Augmented Backus-Naur Form (ABNF) notation [38]. The capitalised tokens in the figure do not conform to ABNF, but are there for simplification purposes. The *ACTION* presents a specific action that is dependent on the component, since every component possesses its own set of actions. The *PARAMETER* presents any number of parameters which, in turn, are dependent on the specified action. The *RAWDATA* can contain any type of data which depends on the requested action. P2P Daemon also supports storing the context data retrieved from the context server or any resource data from P2P overlay networks directly into a local file for easy access (as opposed to unnecessarily transferring the data over, e.g. the D-Bus connection within the local device, which could often cause performance overhead). This behaviour can be invoked using the *storagemode* option.

```
request = component SP action SP *parameter SP [storagemode]

response = statuscode SP component SP action CRLF [data]

component = "daemon" / “context” / "overlay" / "resource" /
            "community" / "service"

action = ACTION ; multiple choices, dependent on component

parameter = PARAMETER ; dependent on component and action

storagemode = "" / "file://" PATH_TO_A_TEMPORARY_STORAGE

statuscode = 200 / 400 / 404 / 405 / 406 / 407 / 418 / 500 / 501

data = "" / RAWDATA ; dependent on component and action
```

Figure 3. Augmented Backus-Naur presentation of P2P Daemon messaging model

By applying our messaging model, community joining messages, for instance, can be structured as below:

Request: community join CafeOulu

Response: 200 community join CafeOulu

In the example given, the community is the *component*, join is the *action*, and CafeOulu is the *parameter*, which in this case presents the identifier of the community. The identifier is also included in the data part of the response message. The included *statuscode* 200 stands for successful action. In general, the status codes are adopted from the P2PP protocol specification [30]. The status codes are quite similar to the ones used in the HTTP protocol.

Currently, P2P Daemon has been implemented for Linux-based platforms. However, P2P Daemon could be easily ported at least to Symbian S60 (a mobile platform), since P2P Daemon has been implemented using plain C++ code with only standard libraries that are available on both Linux and Symbian S60 platforms. Furthermore, the implementation of the P2PP protocol has already been ported to Symbian S60.

4. Management Components

In this Section, the design and implementation of the management components are described. In addition, some evaluation results of the Community Management, Community Access Management, and Community Context Management components are presented.

4.1 Community Management

The rapidly growing popularity of social networking services, such as Facebook, highlights the importance of user communities. This is also the case with P2P overlay networks that are natural platforms for interpersonal communications. In order to enable easy and secure interaction between community members, a solution for community management is needed.

In our Community Management component, the central idea is to create a separate DHT overlay network for each community, resulting in multiple small overlays that are subsets of a common overlay, from now on called the *main overlay* [21]. Therefore, we end up having a hierarchy of overlays as illustrated in Figure 4.

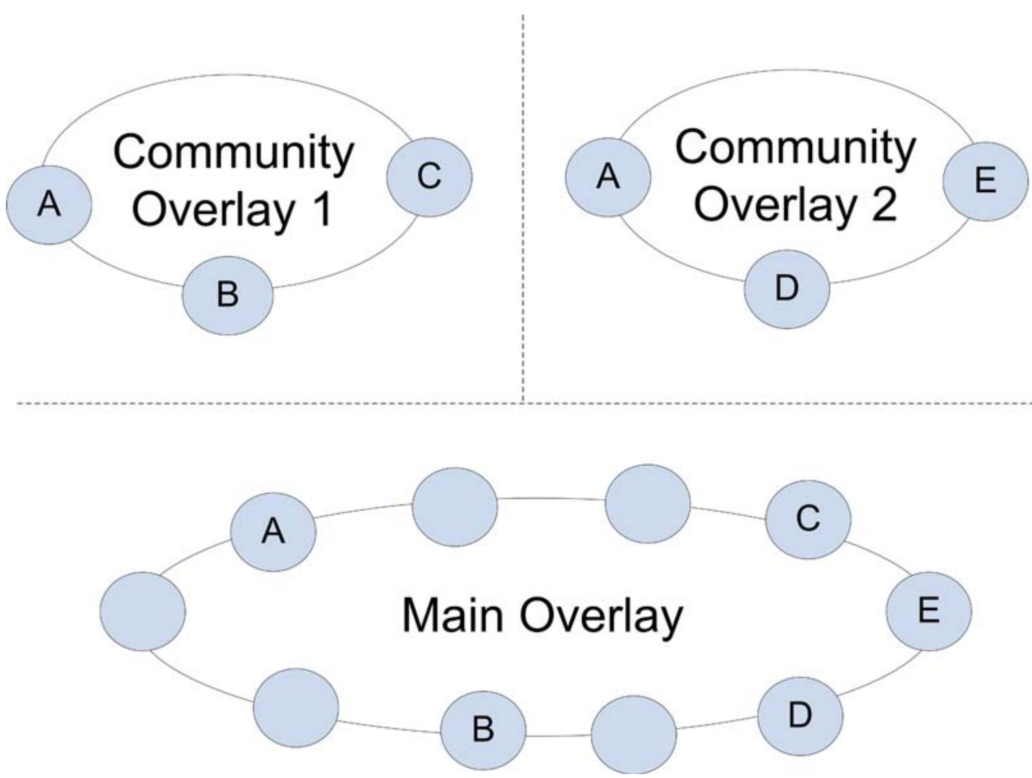

Figure 4. The design of the Community Management component

A new community is established by the following procedure: first, the node that wishes to establish a new community creates a new instance of the P2PP protocol stack (to be run in parallel with the already existing instances that are associated with the main overlay and possibly the other existing communities). Since each community is a self-contained and independent overlay network, the contents of a particular community are separated from the rest of the communities and the main overlay. Second, the existence of the established community, and the community member list that consists only of the creator node itself at this point, are published in the main overlay, embedded in generally known data resources. New members are able to join an established community through the community's creator node or through any of the other member nodes. Joining a community is done by joining the community overlay. Whenever a new node joins the community, it updates the community member list with its own information. It should be noted that both the communities and memberships can also be kept private simply by omitting the publishing of the community-and/or membership-specific data resources in the main overlay.

In the evaluation, conducted in [26], we compared the implemented prototype of the Community Management component with a system, where communities were established within a single DHT-based overlay. According to this evaluation, the Community Management component, with the multiple DHT overlays, performs especially well when communities are relatively small and node activity is high. This primarily results from the two following facts: (1) with Kademlia DHT routing is more efficient in smaller overlays, because having fewer nodes in an overlay results in smaller routing tables, which consequently results in shorter routing paths; (2) with high node activity (i.e. lots of messaging), the exchange of the routing information is more efficient, since there is less need for overlay maintenance-specific messages thanks to the fact that the routing information is embedded in all messages.

4.2 Community Access Management

For a quick and large-scale adoption of a new technology, the technology should be user-friendly. In order to enable easy and intuitive joining to location-based communities, a straightforward solution for community access is needed.

We have implemented the Community Access Management component with support for two short-range communication (SRC) technologies: Radio Frequency Identification (RFID) and Bluetooth (BT). With both SRC technologies, a "landmark" device containing the SRC identifier is placed to the location (or in the proximity of a larger area), to which the location-based community

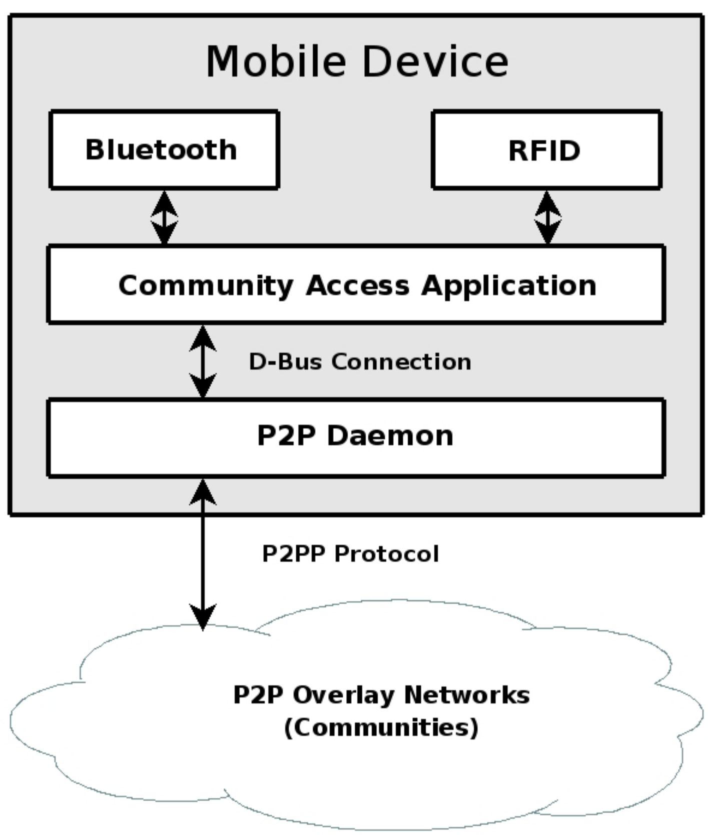

Figure 5. The design of the Community Access Management component

is related. Thus, when the user enters the area, the SRC identifier is detected and retrieved by the user's mobile device. In order to utilise multiple SRC technologies for joining the user communities, we have developed a generic SRC interface within the Community Access Application. The design of the Community Access Management component is illustrated in Figure 5.

With RFID, the unique identifier of an RFID tag, and with BT, the unique identifier of a BT beacon, is used as a DHT-key to retrieve the contact information (IP address and port) of the user community's bootstrap node from the P2P overlay network. The mobile device with a BT transmitter, however, is not paired with the beacon; only the unique identifier is fetched to enable access to the user community. This substantially improves the performance and scalability of the BT-based implementation, since pairing would be a time-consuming operation. In order to enable the usage of RFID and BT technologies, SRC identifiers need to be mapped with the contact information of a community's bootstrap node. This can be done either directly by mapping the SRC identifier with bootstrap information, as shown in Figure 6, or indirectly by binding the SRC identifier with the community identifier. The latter case involves two look-ups: the first look-up operation with the SRC identifier as a key will return the community identifier, and the second look-up operation with the community identifier as a key will return the bootstrap information. This way, the changes in the community bootstrap node's IP and port need only to be updated in a single resource.

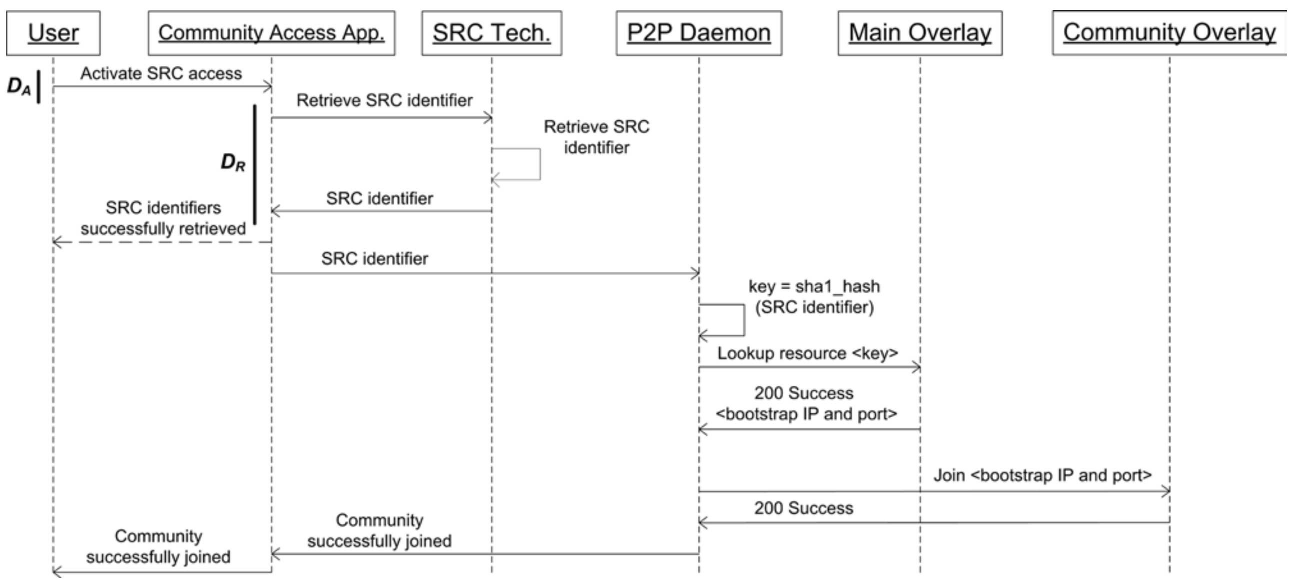

Figure 6. The sequence of the community joining procedure

The performance of the Community Access Management component was evaluated in a laboratory environment using a BT-capable mobile device (Nokia N810 Internet tablet). In the performance evaluation, two delay components were measured: (1) the delay for activating the BT transmitter of the mobile device (D_A); and (2) the delay for retrieving the identifier of the BT beacon (D_R). Both measurements were conducted 20 times. When measuring D_R, the office space was populated with 0, 5, 10, 15, and 20 other devices with BT on in order to evaluate the robustness of the BT-based community joining procedure; this was done because we expected that a greater number of nearby devices with BT on would cause longer delays. The delay caused by P2P overlay networks was not measured, since it was negligible in the laboratory environment (the performance evaluation results of the COMSE in a real-life environment are presented in Section 7). After successfully retrieving the identifier of the BT beacon, the BT transmitter of the mobile device was automatically turned off. The measured delays in the BT-based community joining procedure are summarised in Table 1.

	Average of D_A [s]	**Average / standard deviation of D_R [s]**	**D_A+D_R [s]**
No disturbance	3,24	1,83 / 1,79	5,07
5 other BTs	3,16	2,55 / 1,72	5,71
10 other BTs	3,19	3,43 / 1,86	6,62
15 other BTs	3,13	4,73 / 3,23	7,86
20 other BTs	3,17	4,55 / 3,13	7,72

Table 1. The delay in the BT-based community access procedure

It can be seen in Table 1 that the delay for retrieving the identifier of the BT beacon increases slowly in relation to the number of other devices with BT on. However, when the number of devices with BT on exceeds 15, the standard deviation rises quite rapidly – there were some rather long delays while scanning the BT environment. All in all, the performance of the BT decreases slightly, but not drastically, when the environment is populated with other devices with BT on.

4.3 Content Management

With the rapidly enhancing capabilities of mobile devices and wireless interfaces, the mobile devices are becoming more suitable for hosting Web-based services. However, due to their limited battery life, it is highly beneficial that not all content provided by the mobile Web server is stored locally on the mobile device. In order to lessen the burden on the server-side, a solution for distributed content management is needed.

Web-based services can utilise the Content Management component for sharing data between other services and/or for distributed data storage. Especially for Web-based services running on mobile nodes, it is highly beneficial to place the content data into P2P overlay networks. This way, the content used in a Web-based service can be retrieved from a P2P overlay network instead of the mobile server's local storage. Thus, the total load is distributed in a P2P overlay network. In addition, a P2P overlay network may use a sophisticated load balancing algorithm for further dividing the load according to the node capabilities.

The Content Management component utilises the low-level resource management functions provided by the P2PP protocol. P2P Daemon handles content as a pair of plain text identifier and content data. The P2PP protocol hashes the plain text identifier and publishes the content using the hashed identifier as a resource key and the content data as a resource value. The P2PP protocol achieves the persistence of the published content data with long enough expiration timers and sufficient data replication. Therefore, it is enough for the service provider to re-publish the data only occasionally, and re-publication is mainly needed when the data has been changed. It is also possible that the service provider can be offline for a short period of time without losing the published data items (before the data timers expire). The sequence of publishing content data to a P2P overlay network is shown in Figure 7.

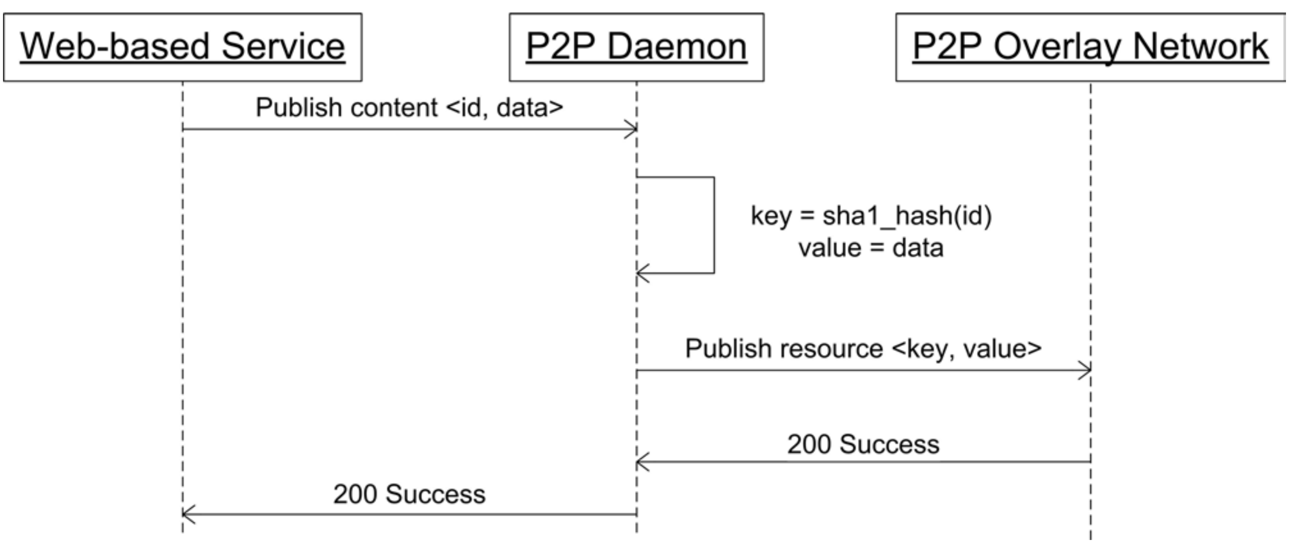

Figure 7. Publishing content data to the P2P overlay network

The creator of a Web-based service can decide whether the content data of a service is stored in a P2P overlay network or locally on the server. The publication of the content data to a P2P overlay network can be implemented, for instance, by a Web-based service or done manually using management tools introduced in Section 5. A Web-based service must know the identifiers (overlay identifier and data resource identifier, i.e. DHT-key) to the content data in a P2P overlay network; the identifiers are passed between the server and the client in a service-specific way. The client passes the content data identifiers to local P2P Daemon that retrieves the data from a P2P overlay network. If the content data is only stored on the server or is not found in the P2P overlay network, the content data is directly transferred between the server and the client. The sequence of content retrieval from the P2P overlay network is shown in Figure 8.

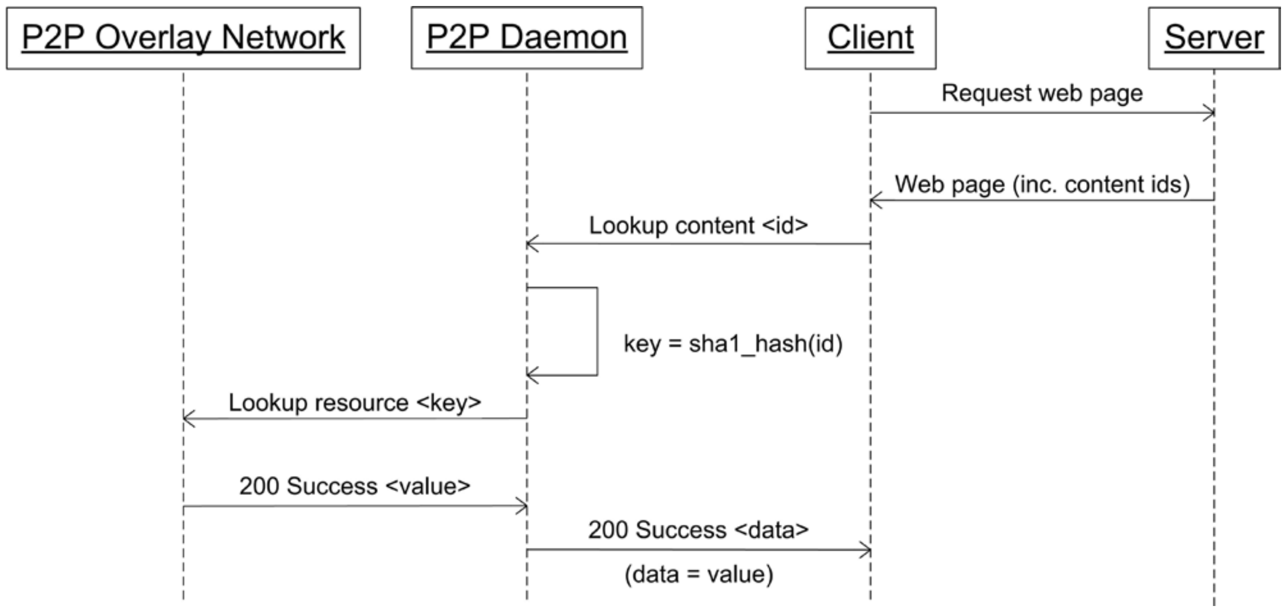

Figure 8. Retrieval of content from the P2P overlay network

4.4 Service Management

In order to enable easy distribution of both commercial and user-created Web-based services in P2P overlay networks, a solution for service publishing and discovery is needed. The implementation of the Service Management component [28] is fairly simple, since our aim was not to develop new service management mechanisms, but to enhance and demonstrate the collaborative usage of P2P and Web technologies in general.

The Service Management component relies on the low-level resource management functions provided by the P2PP protocol. The sequence of service publishing is illustrated in Figure 9a [28], where SERV*utility* stands for utility services and SERV*user* for end-user services. All Web-based services are published as resource objects that embody the following key-value pair: <service identifier, service XML>. Service XML is a simple file that holds metadata such as service description, category, version, and the service's physical address. By examining the provided metadata, the service requestor should be able to determine whether the Web-based service is of interest, and how to use the service.

In Figure 9a, the resource objects that contain (1.1) utility service metadata and (1.2) the list of available utility services are published in the main overlay. Since the P2PP protocol supports attaching multiple values to a single key, a utility service list can be easily implemented by publishing all utility services with a publicly known key identifying the utility service list. This way, the accessibility to utility services can be ensured for all nodes in a P2P overlay network. (2.1) The availability of end-user services is usually published within a particular community overlay. The end-user services are mainly community-specific, but end-user services can also be published in the main overlay. (2.2) The category of an end-user service is also published in a community overlay with a publicly known key. In addition, resource objects that comprise (2.3) the list of available end-user services is also published in a particular community overlay or in the main overlay, and again with a publicly known key identifying a service list. The possible service categories are available as a common list in the main overlay. The available categories can be defined by the infrastructure provider, for instance.

The sequence of service discovery is, in turn, shown in Figure 9b. Utility services can be discovered directly if a particular service identifier (i.e. a key) is known, or indirectly by examining the utility service list. The discovery of utility services takes place in the main overlay. End-user services can be discovered in the same way as utility services, but also the service categories can be used as search parameters. The discovery of end-user services takes mainly place in specific community overlays. In both cases, with utility and end-user services, the interpretation of service XML is done on the service layer.

4.5 Community Context Management

The context information can be defined as any *information that can be used to characterise the situation of an entity* [39]. This entity has been usually taken to mean a single user [40]. However, context information can also be used to describe the situation of an entire user community. The context of a community can, for example, contain collective mood and the music taste of a

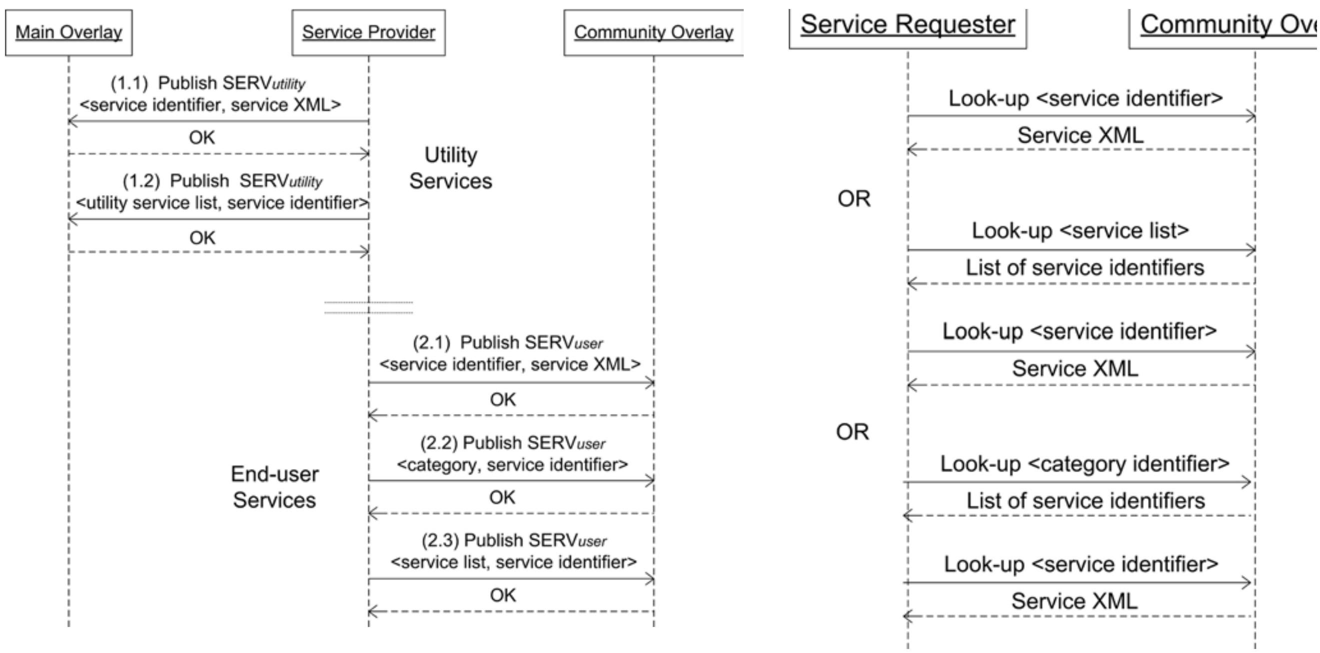

Figure 9a. The sequence of service publishing. Figure 9b. The sequence of service discovery

clientele of a restaurant. The value of the context information increases with the mobility of users. This is due to the fact that the moving users' mobile devices produce more versatile context information, and the moving users more often engage in situations where this context information can be exploited. In order to enable robust storage and delivery of context information, and consequently, easy development of context-aware mobile services, a solution for community context management is needed.

We do not only concern the context of an individual user, but take a broader approach, handling also the context of an entire user community. According to our conception, community context is defined as a raw collection of individual contexts rather than an approximation calculated based on the individual contexts of community members. Because of this approach, Web-based services are free to utilise and interpret the community context in a way that suits their needs best.

The Community Context Management component is a hybrid solution that relies both on a centralised context server and P2P overlay networks. We chose to use a centralised server for the storage and delivery of the context information, since with time-critical context, such as location, P2P overlay networks might cause some disturbing delays. However, in the case of user communities, the member lists are retrieved from P2P overlay networks. The design of the Community Context Management component is shown in Figure 10.

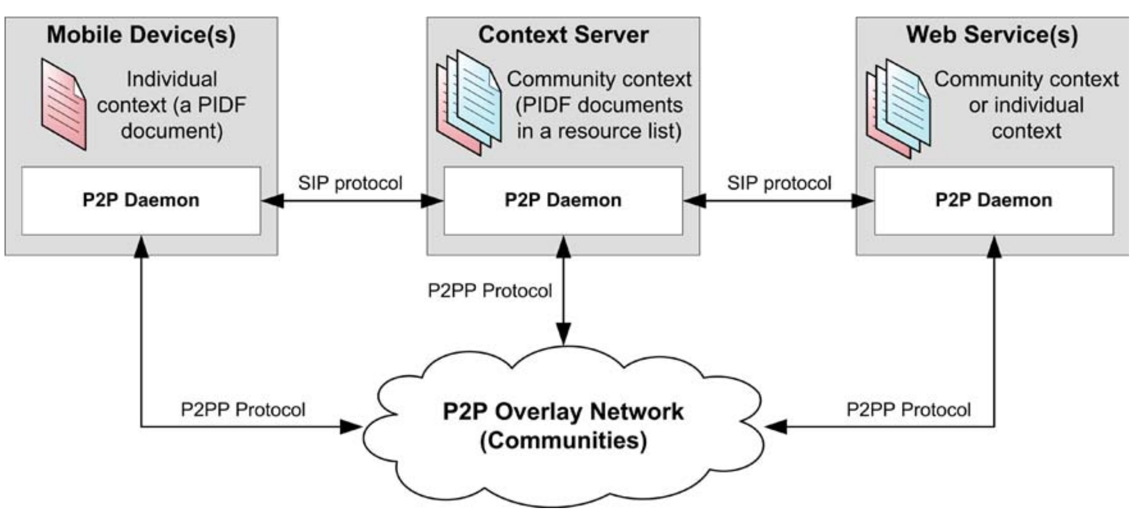

Figure 10. The design of the Community Context Management component

Our approach utilises the context presentation related IETF standards in a novel way [29]. The context representation of an individual community member is based on Presence Information Data Format (PIDF) that is an XML format defined by IETF for storing and distributing all kinds of context payload [41]. PIDF has later been enhanced with Location Object (PIDF-LO) that is especially utilised for carrying location data. The representation of the community context is implemented with the resource list extension for event notifications [42]. The PIDF representations of the community members' context are gathered into a single resource list that is finally transferred within SIP packets to interested services. In our Community Context Management component, the functionality equal to a resource list server is also implemented, but as a distributed solution, since the community member lists are retrieved from a P2P overlay network using P2P Daemon. Furthermore, our Community Context Management component implements publish and subscribe features which have been standardised as part of the work done in an IETF workgroup called SIMPLE [34].

The performance of the Community Context Management component was evaluated by comparing the efficiency of making a subscription to one community context against the efficiency of making separate subscriptions to multiple individual contexts. The tests were run using the Nokia N810 Internet tablet, a mobile device with a WLAN connection. The context server was run on a PC server machine. All the data were transferred using SIP messages over the TCP protocol in an IP-based network.

In the network traffic analysis, the proportions of network traffic generated by the context payload and by the headers of IPv4, TCP and SIP, occurring in a context subscription, were measured. The testing was performed with communities of up to 100 members. The evaluation results [29] show that our Community Context Management component outperforms the traditional method of individual subscriptions in terms of network traffic. Even with a community of two members, it is useful to retrieve community context instead of two individual subscriptions. With larger communities (more than 10 members), the amount of network traffic resulting from a community subscription is even 50% to 60% less than with multiple individual subscriptions.

5. Management Tools

In order to enable service providers, administrators, as well as end-users to manage their activities in the COMSE, they need to be able to manage the operation of the local P2P Daemon. For this purpose, some management tools have been developed.

The operation of P2P Daemon can be manually managed (1) *on startup* with specific configuration data files, and (2) *at runtime* with commands sent over the local socket interface. The *primary* configuration data file contains items for setting up the user name (unhashed peer identifier), the identifier of the main overlay, and some general network settings such as the listening port for the incoming P2PP protocol messages. The *secondary* configuration data file contains items for specifying the actions that are performed automatically when local P2P Daemon is started. These actions can include, for instance, joining to a specific community or publishing some services. Of course, the same actions can also be initiated at runtime through the local socket interface.

For facilitating the management of local P2P Daemon, a Web-based utility has been developed. The functionality provided by the utility has been divided into three interfaces that are *status monitoring*, *management*, and *administration*.

The status monitoring interface displays the state (active or inactive) and the uptime of local P2P Daemon, as well as statistical information about the transactions of the local P2P node. Statistical information contains the average hop count and the success ratio of the transactions that were initiated by the local P2P node. In addition, the list of resources and the routing table entries maintained by the local P2P node are available for each overlay the node is participating in.

The management interface provides tools for controlling local P2P Daemon. In the management interface, (1) the Communities tab provides means for creating and joining community overlays (i.e. overlays with community functionality); (2) the Services tab provides means for publishing new services to the overlays; (3) the Context tab provides means for interacting with the context server; (4) the Overlays tab provides means for creating and joining overlays without the community functionality; and finally, (5) the Content tab provides means for publishing content to the overlays. Optionally, the user can enable the automatic initiation of these actions. As mentioned before, the automatic actions are specified in the secondary configuration data file that is processed on startup by local P2P Daemon.

The administration interface provides tools for adjusting the logging level of the P2PP protocol, and for examining the created log files. The log files are visualised using a graphical log analyser, as shown in Figure 11. By using the graphical log analyser, the user can study all transactions and statistical information of any P2P node participating in the overlay that is currently under examination.

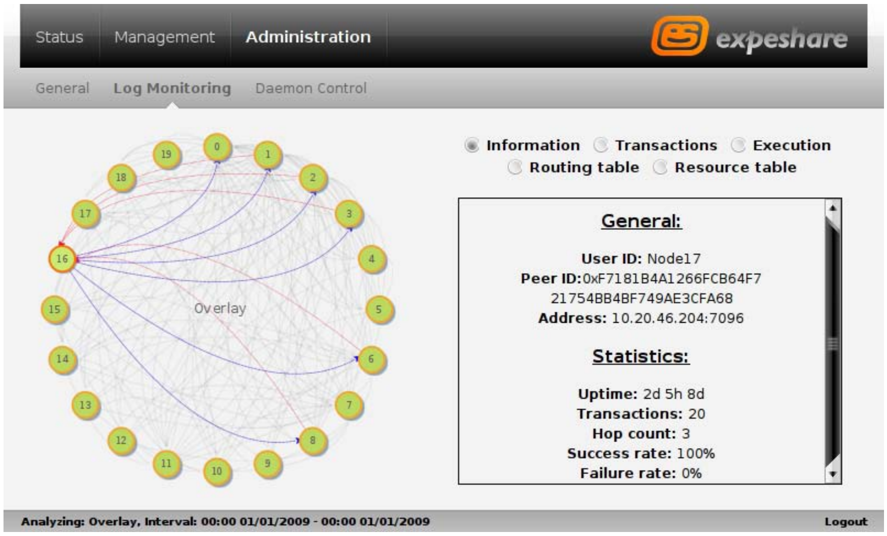

Figure 11. Graphical log analyser

6. Pilot Services

In this Section, two pilot services are introduced. The pilot services were developed in order to enable the performance evaluation of the COMSE with potential real-life service examples.

6.1 Community-centric Music Voting Service

The first prototype service built on the COMSE is called *Community-centric Music Voting* (CCMV) service [43]. The communities are formed by the customers of cafeterias, pubs, or bars. For each of these entertainment premises, we establish a P2P-based community that can be accessed with either RFID or BT technologies. The CCMV service interfaces with a commercial *DjOnline* system (www.djonline.fi/eu/) that provides a huge database of music that is streamed to the client front-ends residing in the commercial premises. The streamed songs are selected automatically by the DjOnline system based on the desired music style of the entertainment premises. The DjOnline system possesses also the functionality for creating music voting events.

With the provided Web-based service shown in Figure 12, the customers can affect the music played in the entertainment premises by voting for their favourite songs. The end-user application displays user community's playlist, which also contains the possible voting events. The implementation of the CCMV service relies on a reusable utility service that provides generic voting mechanisms. In order to achieve a desktop-like application that is interactive and easy to use, we have used AJAX.

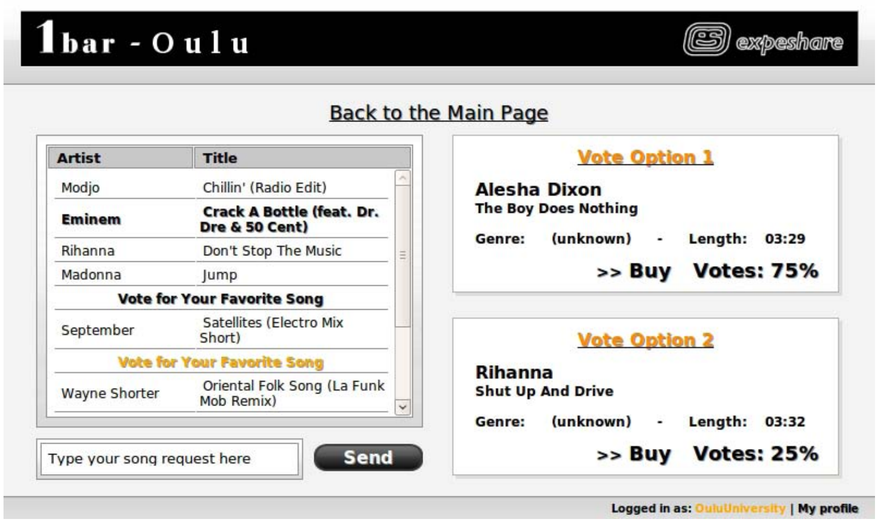

Figure 12. The Music Voting service

The CCMV service has also been enhanced with communal features. If permitted by the users, the member list of the community can be retrieved from the P2P overlay network and displayed along with the individuals' context information such as a person's current mood or interest in some social or game activity. The context information is stored and distributed by the context server.

6.2 City Night Life Service

The second prototype service built on the COMSE is called the *City Night Life* (CNL) service [44]. As part of a convenient user interface, the CNL service utilises GoogleMaps (an example of a utility service from the Internet) that is populated with a variety of entertainment premises, as illustrated in Figure 13. The CNL service also interfaces with the DjOnline system in order to get information about the opening hours, tonight's events, and the music played in the entertainment premises.

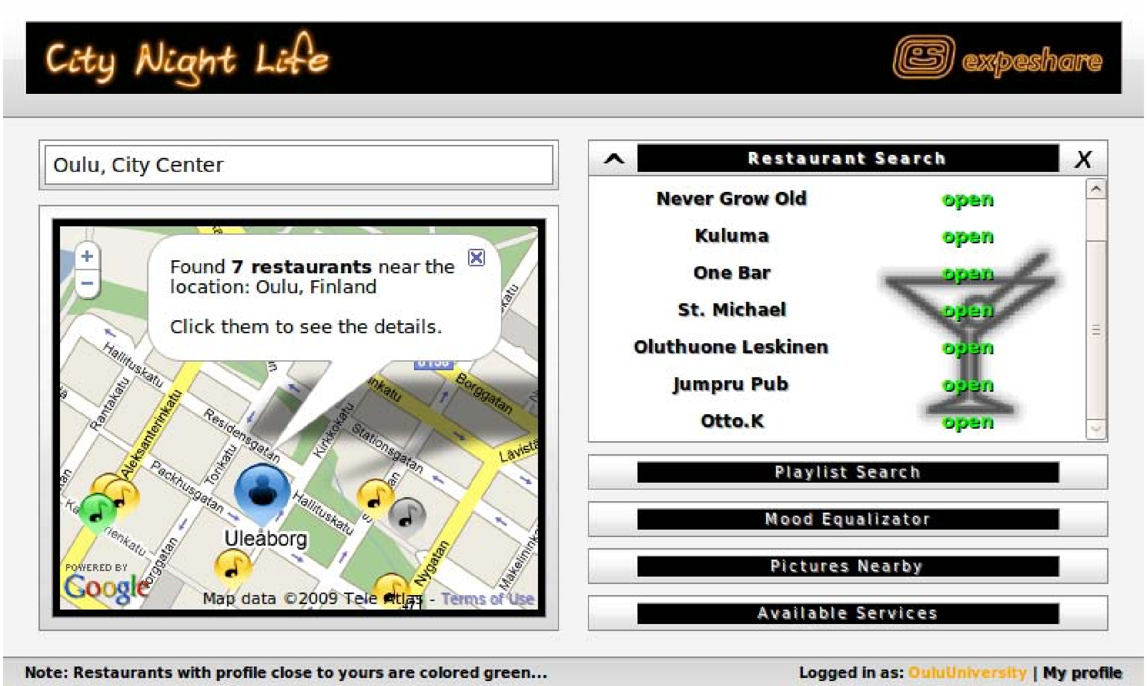

Figure 13. The City Night Life service

The CNL service provides a simple tool for creating personal music profiles that are stored by the context server. Based on the given information, the CNL service is able to compare the individual music tastes against the music played in the entertainment premises, and as a result, provide recommendations where a user could pay a visit. Enabled by the Community Context Management component, the comparison could also be done between the music tastes of the customers of the entertainment premises and an individual user. In this case, the user would be looking for a place, where her own music taste is shared by other people.

7. Performance Evaluation

In this section, the settings and the results of the performance evaluation of the COMSE are presented.

7.1 Evaluation Settings

We evaluated the performance of the COMSE by measuring nodes' messaging load, success ratio of P2P operations, and latencies of successful P2P operations. The measurement data was acquired by observing actual messages generated during the usage of both CCMV and CNL services in a real usage environment. The evaluation took place in the premises of a bar that is situated in the city centre of Oulu, Finland. Altogether, 53 students from various faculties of University of Oulu participated in the testing of the pilot services.

The test users were equipped with N810 Internet Tablets that were connected to the Internet through a WLAN base station (IEEE 802.11b/g). Due to the limited capacity of the available WLAN base station, only approximately 10 test users were allowed to use the system at the same time, resulting in five comparable test events. In every test event, the test users were given 40 minutes time to use both the CNL and the CCMV services.

The established P2P overlay networks, in each test event, comprised approximately of 10 mobile nodes, plus a bootstrap node and a node that hosted the context server, totalling an average of 12 nodes. The bootstrap node and the context server node were run in a single server machine that was connected to the Internet through a fixed line. Altogether, we managed to collect reliable data from 10 fixed line node instances but only from 30 mobile node instances, since some of the mobile nodes crashed several times. The malfunction of the mobile nodes was, on the one hand, caused by the heavily loaded wireless network and the poorly performing WLAN connection of the N810 devices. On the other hand, the implementation of the COMSE is still in a prototype stage, and therefore, prone to failures in unstable environments.

7.2 Results

In the evaluation, the average (AVG) uptime of a mobile node was 28 minutes with a standard deviation (STDEV) of 8 minutes, as shown in Table 2. Thus, the established P2P overlay network was subjected to a reasonable amount of churn that was caused by joining and leaving of the mobile nodes. In addition, more churn was inflicted on the P2P overlay network by the malfunctioning mobile nodes that crashed almost immediately after joining the network. These observations indicate that the established P2P network represented well a real usage environment, where the amount of churn is often significant. The standard deviations in the case of the bootstrap and the context server nodes were not calculated, since these nodes were up and running during the whole testing period.

	Uptime (min)		Messages/min	
	AVG	STDEV	AVG	STDEV
Mobile nodes	28	8	165	34
Bootstrap nodes	60	-	279	39
Context server nodes	59	-	70	9

Table 2. Node uptime and the number of messages

A mobile node processed (i.e. sent or received), in average, 165 messages in a minute with a standard deviation of 34. This result indicates that the P2P traffic was fairly equally distributed among the mobile nodes. If using the client-server architecture instead, the most of this network load would accumulate to the server. According to the measurements done in [26], the mobile nodes could participate in this specific P2P overlay network even up to 8 hours before running out of battery. Compared to the mobile nodes, the average load on the bootstrap nodes was higher, since these nodes acted as gateways for joining the P2P overlay network. On the contrary, the average load on the context server nodes was significantly lower, since these nodes did not belong to any user communities and did not use any of the available services.

In Table 3, the success ratio of the P2P operations is shown. Here, a *P2P operation* is defined as a series of messages starting with an initial request, which is followed by other request, response, and acknowledgement messages sent between the various nodes of an overlay in order to fulfil the initial request. Examples of P2P operations are resource publishing and resource look-up. Acknowledgement messages were used, because only the UDP protocol was implemented as a transport protocol for the P2PP stack; the acknowledgement messages were required to compensate for the unreliable nature of the UDP protocol. The average success ratio of P2P operations for all nodes in the P2P network was as high as 88%. However, in 32% of all P2P operations (including successful and expired P2P operations), one or more retransmissions of messages were required during the operation.

	Successful P2P operations	Expired P2P operations	P2P operations that needed retransmissions
Mobile nodes	90 %	10 %	35 %
Bootstrap nodes	86 %	14 %	30 %
Context server nodes	85 %	15 %	22 %
All nodes	88 %	12 %	32 %

Table 3. Success, expiration and retransmission ratio of the P2P operations, as a percentage of all initiated operations

The latency is sometimes considered as a challenging issue in P2P overlay networks, since routing of messages might require multiple logical hops for a successful operation. We measured the latencies of the successful P2P operations to assure that the latencies are low in small-sized user communities that are location-based (i.e. mainly sharing the same network). The results are presented in Table 4.

	Latency (s)		
	< 1	**1-5**	**> 5**
Mobile nodes	70 %	28 %	2 %
Bootstrap nodes	68 %	31 %	1 %
Context server nodes	79 %	19 %	2 %
All nodes	70 %	28 %	2 %

Table 4. Latencies of the successful P2P operations

According to our measurements, more than 70% of the P2P operations were completed within a time-frame of one second that is acceptable for fluent use of Web-based services. Since the established communities were small-sized, most of the P2P operations required only a single hop. This result corresponds with the measurements done in [11], where the average one-hop-latency was measured to be 0.9 seconds in a large-scale DHT-based P2P network. Next, less than 30% of the P2P operations were completed within a timeframe of 1 to 5 seconds. This delay might be noticeable, but it did not notably interfere with the usage of the Web-based service, since the P2P operations are carried out in an asynchronous manner. By examining both Table 3 and Table 4, it seems that the latencies exceed one second (the two rightmost columns in Table 4) when re-transmissions (the rightmost column in Table 3) are needed to complete the operations: the percentage of P2P operations with retransmissions is approximately the same as the percentage of operations with long latencies. Since the retransmission delay is caused by the underlying physical network, this delay is independent of the used application layer communication technology, P2P in our case.

Although the overall success ratio of the P2P operations was as high as 88%, the success ratio of the look-up operations remained at 55%. For instance, in a publishing operation, the resource is stored by a node whose hashed peer identifier is currently the nearest one to the hashed identifier of the published resource according to the DHT algorithm. In time, when the structure of the P2P overlay network changes, the published resources are assigned to new nodes. Although the resource still exists in the overlay, the lookup operation can be considered successful only if the searched resource is actually found – the look-up may fail if the routing information is not up-to-date or the resource has been erased due to an ungraceful leaving of a node. This explains why the success ratio of look-up operations is usually lower than the overall success ratio. These findings are supported, for instance, by [46] and [47], where NetHawk EAST test automation and traffic generation tool and P2PSim simulator, respectively, were used for simulating the operation of DHT algorithms in the presence of churn. In these studies, the success ratio of look-up operations varied approximately from 60 to 90 percent in scenarios with high levels of churn. The reason for our lower look-up success ratio presumably results from the two following facts: (1) the WLAN connection was under heavy stress, which caused a relatively high number of retransmissions; (2) even though the average uptime of a mobile node was as high as 28 minutes in our measurements, as mentioned before, a lot of churn was inflicted on the P2P overlay network by the malfunctioning mobile nodes that crashed almost immediately after joining the network.

The proportion of the service-specific P2P traffic was in average 39% of the total traffic. This proportion includes all P2P traffic that is related to the user communities and the available Web-based services. The relatively high amount of maintenance traffic (61%) originates mainly from the keep-alive messaging that is used to maintain the integrity of the P2P overlay network. It is important to notice that in our implementation of the P2PP protocol, the proportion of the keep-alive messaging decreases when the service-specific messaging increases. This occurs, because the nodes learn about the state of other nodes, also when receiving service-specific messages from them. As a result, less keep-alive messages are needed as they are substituted by the service-specific messages.

Finally, it should be noted that the tested services utilise the P2P overlay networks only as means for managing user communities and for service discovery. Using the P2P overlay networks also for data storage would clearly increase the proportion of service-specific P2P traffic.

8. Conclusions and Discussion

In this paper, we examined the design, the implementation and the evaluation results of a Community-centric Mobile Service Environment (COMSE), where all services are provided with Web technologies. The COMSE relies on a distributed communication platform that is implemented using a DHT algorithm called Kademlia and a DHT-based P2P protocol called P2PP. The interaction between Web-based services and the P2P overlay networks has been realised with a mobile middleware component called P2P Daemon. The usage of P2P overlay networks improves fairness of load distribution, scalability and fault tolerance of the COMSE, whereas the usage of Web-based services enables easy development, reuse and composition of platform-independent mobile services.

The performance of the COMSE was evaluated with the following criteria: (1) nodes' messaging load; (2) success ratio of P2P operations; and (3) latencies of successful P2P operations. The evaluation was conducted with two pilot services that were used altogether by 53 test users in a real usage environment. In five comparable test events, we established P2P overlay networks that consisted of approximately 10 mobile and 2 fixed nodes. A reasonable amount of churn was inflicted on these P2P overlay networks by the mobiles nodes' joining and leaving.

According to the evaluation results, the load inflicted on mobile nodes for maintaining the P2P overlay networks was at an acceptable level and quite evenly distributed. In the established P2P overlay networks, the mobile nodes could have been operable even up to 8 hours without running out of battery. This is a result from the following two facts: (1) the established user communities, in the case of our two pilot services, were small-sized resulting in small hop count; and (2) the P2PP protocol is capable of exchanging routing information within all messages, which decreases the need for sending maintenance-specific messages. Even though the used WLAN network was under heavy load, the success ratio of P2P operations was as high as 88%. Moreover, 70% of the latencies were less than one second, which is an acceptable delay for the smooth operation of Web-based services. A delayed P2P operation did not notably interfere with the usage experience, since the P2P operations were carried out in the background, in an asynchronous manner. The lookup success ratio, however, remained at 55% due to the inconsistency of routing information caused by the high level of churn. In the case of the two pilot services, the proportion of the service-specific (i.e. other than overlay maintenance-specific) P2P traffic was in average 39% of the total traffic.

This case study provided examples how P2P and Web technologies can be collaboratively used in mobile environments. According to the evaluation results, it is technically feasible for mobile nodes to use Web-based services in community-centric P2P overlay networks. Furthermore, the evaluation results indicate that today's high-end mobile devices are capable of participating in (at least small-sized) DHT-based P2P overlay networks. It is also important to notice that Kademlia (as most of the DHT algorithms) scales well to large P2P overlay networks. When the number of nodes in a P2P overlay network grows linearly, the complexity of routing length and routing table size, and consequently, the load inflicted on nodes, grows only in a logarithmic scale [45]. Therefore, we can predict that the performance of mobile devices would not significantly decrease in larger P2P overlay networks.

The future work could include in-depth evaluations of community-centric mobile services that utilise P2P overlay networks in a more extensive manner. In addition, the COMSE could be enhanced with features that enable user-originated service creation and composition, taking the concept of reusable service templates to the next level.

9. Acknowledgements

The financial support from the National Technology Agency of Finland (Tekes) and Graduate School in Electronics, Telecommunications and Automation (GETA) is gratefully acknowledged. In addition, this work has been supported with scholarship grants by Foundation of Technology, Nokia Foundation, Foundation of Walter Ahlström, Research and Educational Foundation of TeliaSonera, Research Foundation of HPY, and Foundation of Tauno Tönning. Finally, the authors would like to thank the fellow researchers in the ITEA2 Expeshare and DECICOM (for providing the P2PP protocol implementation) project consortiums, and especially, Janne Julkunen who assisted in the implementation work and testbed preparation.

References

[1] Portmann, M., Sookavatana, P., Ardon, S., Seneviratne, A (2001). The cost of peer discovery and searching in the Gnutella peer-to-peer file sharing protocol. *In: Proc. of the 9th IEEE International Conference on Networks*, pages 263-268. Oct. 2001.

[2] Androutsellis-Theotokis, S., Spinellis, D (2004). A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys* 36(4) 335-371.

[3] Peng, Z., Duan, Z., Qi, J., Cao, Y., Lv, E (2007). HP2P: A Hybrid Hierarchical P2P Network. *In: Proc. of the International Conference on the Digital Society*, pages 18-22. Jan. 2007.

[4] Kalofonos, D.N., Antoniou, Z (2008). A hybrid P2P/Infrastructure platform for personal and social Internet services. *In: Proc. of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1-6. Sept. 2008.

[5] Bisignano, M., Modica, G.D., Tomarchio, O (2004). An infrastructure-less peer-to-peer framework for mobile handheld devices. *European Transaction on Telecommunications* 15(6) 599-612.

[6] Li, Z., Huang, D., Liu, Z., Huang J (2003). Research of peer-to-peer network architecture. *In: Proc. of the International Conference on Communication Technology*, pages 312-315. Apr. 2003.

[7] Gehlen, G., Pham, L (2005). Mobile Web Services for Peer-to-Peer Applications. *In: Proc. of the 2nd Conference on Consumer Communications and Networking*, pages 427-433. Jan. 2005.

[8] Christensen, B.G., LightPeers: A Lightweight Mobile P2P platform. *In: Proc. of the International Conference on Pervasive Computing and Communications Workshops*, pages 132-136. Mar. 2007.

[9] Anirban, M., Kumar, M.S., Masaru, K (2007). Research issues and overview of economic models in Mobile-P2P networks. *In: Proc. of the 18th International Conference on Database and Expert Systems Applications*, pages 809-813. Sept. 2007.

[10] Zhiyi, Q., Yuanting, G., Kaiyuan, J., Tiangang, L., Zhong, W (2007). A Study on the Model of Mobile P2P File-Sharing System is 3G Network. *In: Proc. of the 3rd International Conference on Semantics, Knowledge and Grid*, pages 306-309. Oct. 2007.

[11] Kelényi, I., Nurminen, J.K (2008). Optimizing Energy Consumption of Mobile Nodes in Heterogeneous Kademlia-based Distributed Hash Tables. *In: Proc. of the International Conference on Next Generation Mobile Applications, Services, and Technologies*, pages 70-75. Sept. 2008.

[12] Tutschku, K., Hoßfeld, T., Berl, A., de Meer, H., Oberender, J.O (2007). Mobile P2P Applications in Infrastructure-Based Mobile Networks, *Technical Report* No. 398. Jan. 2007.

[13] Emekci, F., Sahin, O.D., Agrawal, D., Abbadi, A.E (2004). A Peer-to-Peer Framework for Web Service Discovery with Ranking. *In: Proc. of the International Conference on Web Services*, pages 192-199, Jul. 2004.

[14] Yu, J., Su, H., Zhou, G., Xu, K (2007). SNet: Skip Graph based Semantic Web Services Discovery. *In: Proc. of the ACM Symposium on Applied Computing*, pages 1393-1397. Mar. 2007.

[15] Lin, Q., Rao, R., Li, M (2006). DWSDM: A Web Services Discovery Mechanism Based on a Distributed Hash Table. *In: Proc. of the International Conference on Grid and Cooperative Computing Workshops*, pages 176-180. Oct. 2006.

[16] Chen, G., Low, C. P., Yang, Z (2008). Coordinated Service Provision in Peer-to-Peer Environments, *IEEE Transaction on Parallel and Distributed Systems* 19(4) 373-384.

[17] Li, R., Zhang, Z., Wang, Z., Song, W., Lu, Z (2005). WebPeer: A P2P-based system for publishing and discovering Web services, *In: Proc. of the IEEE services computing (SCC)*, pages 149-156. Jul. 2005.

[18] He, Q., Yan, J., Yang, Y., Kowalczyk, R., Jin, H (2008). Chord4S: A P2P-based Decentralised Service Discovery Approach, *In: Proc. of the IEEE services computing (SCC)*, pages 221-228. Jul. 2008.

[19] Wanlei, Z., Casey, J (2008). Reducing Cache Lookup and Publish Overhead through Locality Awareness in P2P Web Caching, *In: Proc. of the IEEE International Conference on High Performance Computing and Communications*, pages 296-303. Sept. 2008.

[20] Iwamaru, A., Itokawa, T., Kitasuka, T., Aritsugi, M (2009). Introducing Group Participation Support into P2P Web Caching Systems, *In: Proc of the International Conference on Advanced Information Networking and Applications*, pages 868-875. May 2009.

[21] Cheng, H., Gu, Z., Ma, J (2007). IntraCache: An Interest group-based P2P Web Caching System, *In: Proc. of the IEEE International Parallel and Distributed Processing Symposium*, pages 1-8, May 2007.

[22] Ribeiro, H.B., Lau C.L., Santin, A.O., Brisola, N.L (2007). Implementing a Peer-to-Peer Web Browser for Publishing and Searching Web Pages on Internet, *In: Proc. of the International Conference on Advanced Information Networking and Applications*, pages 754-761. Nov. 2007.

[23] Deshpande, M., Amit, A., Chang, M., Venkatasubramanian, N., Mehrotra, S (2007). Flashback: A Peerto-PeerWeb Server for Flash Crowds, *In: Proc. of the International Conference on Distributed Computing Systems*, pages 1-8. Jun. 2007.

[24] Terrace, J., Laidlaw, H., Liu, H.E., Stern, S., Freedman, M.J (2008). Bringing P2P to the Web: Security and Privacy in the Firecoral Network, *In: Proc. of the International Conference on Peer-to-peer Systems*, pages 1-7. Sept. 2008.

[25] Deepnet Explorer. URL: http://www.deepnetexplorer.com.

[26] Koskela, T., Kassinen, O., Ou, Z., Ylianttila, M (2010). Improving Community Management Performance with

Two-level Hierarchical DHT Overlays, *Journal of Internet Technology, Special Issue on Internet Resource Sharing and Discovery* 11(2) 167-181.

[27] Koskela, T., Kostamo, N., Kassinen, O., Ohtonen, J., Ylianttila, M. (2007). Towards Context-Aware Mobile Web 2.0 Service Architecture. *In: Proc. of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 41-48. Nov. 2007.

[28] Koskela, T., Julkunen, J., Korhonen, J., Liu, M., Ylianttila, M. (2008). Leveraging Collaboration of Peerto-Peer and Web Services. *In: Proc. of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 1-6. Oct. 2008.

[29] Pellikka, J., Koskela, T., Ylianttila, M. (2008). Partially Decentralised Context Management for P2P Communities. *In: Proc. of the 2nd IEEE Conference on Exhibition on Next Generation Mobile Applications, Services and Technologies*, pages 89-94. Sept. 2008.

[30] Maymounkov, P., Mazieres, D (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. *In: Proc. of the International Workshop on Peer-to-Peer Systems*, pages 1-6. Mar. 2002.

[31] Baset, S., Schulzrinne, H., Matuszewski, M (2007). Peer-to-Peer Protocol (P2PP), *IETF Internet Draft*, Work-in-progress. Nov. 2007.

[32] Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., Schulzrinne, H (2008). Resource Location and Discovery (RELOAD), *IETF Internet Draft*, Work-in-progress. Jun. 2008.

[33] Kassinen, O., Harjula, E., Korhonen, J. Ylianttila, M. (2009). Battery Life of Mobile Peers with UMTS and WLAN in a Kademlia-based P2P Overlay. *In: Proc. of the 20th Personal, Indoor and Mobile Radio Communications Symposium 2009 (PIMRC09)*. Sept. 2009.

[34] Rosenberg, J (2008). SIMPLE made Simple: An Overview of the IETF Specifications for Instant Messaging and Presence using the Session Initiation Protocol (SIP), *IETF Internet Draft*, Work-inprogress. Oct. 2008.

[35] Schroth, C., Janner, T (2007). Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IEEE IT Professional Magazine* 9(3) 36-41.

[36] Erl, T (2005). Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR. 792.

[37] Browser DBus Bridge. URL: http://sandbox.movial.com/wiki/index.php/Browser_DBus_Bridge.

[38] Crocker, D., Overell, P (2008). Augmented BNF for Syntax Specifications: ABNF. *IETF RFC* 5234. Jan. 2008.

[39] Dey, A., Abowd, G (1999). Towards a Better Understanding of Context and Context-Awareness. *In: Proc. of the International Symposium on Handheld and Ubiquitous Computing*, pages 304-307. Sept. 1999.

[40] Chen, G., Kotz, D (2000). A Survey of Context-Aware Mobile Computing Research, *Technical Report* TR2000-381. Nov. 2000.

[41] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., Peterson, J (2004). Presence Information Data Format (PIDF), *IETF RFC* 3863. Aug. 2004.

[42] Roach, A.B., Campbell, B., Rosenberg, J (2006). A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists, *IETF RFC* 4662. Aug. 2006.

[43] Koskela, T., Julkunen, J., Kassinen, O., Ylianttila, M (2010). User Evaluation of a Community-centric Music Voting Service, *Journal of Digital Content Technology and its Application* 4(3) 43-57.

[44] Koskela T., Järvinen S., Liu M., Ylianttila M (2010). User Experience in Added Value Location-Based Mobile Music Service. *In: Proc. of the 8th IEEE International Conference on Web Services.* Jul. 2010.

[45] Harjula, E., Hautakorpi, J., Beijar, N., Ylianttila, M (2009). Peer-to-Peer SIP for Mobile Computing: Challenges and Solutions. *Seet B-C (ed.) Mobile Peer-to-Peer Computing for Next Generation Distributed Environments: Advancing Conceptual and Algorithmic Applications*, Hershey, USA: IGI Publish.

[46] Ou, Z., Harjula, E., Kassinen, O., Ylianttila, M (2010). Performance Evaluation of a Kademlia-based Communication-oriented P2P System under Churn. *Elsevier Journal of Computer Networks* 54 (5) 689705.

[47] Herrera, O., Znati, T (2007). Modeling Churn in P2P Networks, *In: Proc. of the 40th Annual Simulation Symposium*, pages 33-40. Mar. 2007.

Authors Biographies

Timo Koskela received his M.Sc. degree in Information Engineering in 2005, and his M.Sc. degree in Management and Organization in 2009, from the University of Oulu, Finland. He is now working as a research scientist in MediaTeam Oulu research group at the University of Oulu, and focusing on his Ph.D in the area of community-centric mobile services. His professional interests include mobile applications and services, context-awareness, P2P overlay networking, user evaluations, and community management.

Otso Kassinen is working as a research scientist in the MediaTeam Oulu research group at the University of Oulu, Finland. He received his M.Sc. degree in Information Engineering from the University of Oulu, Finland, in 2007. He is currently working towards a Ph.D. degree in the University of Oulu. In his research work, he is focusing on mobile middleware solutions for the management of device resources, the establishment of communication sessions, and scalable service provision especially in DHT-based P2P networks.

Erkki Harjula is working as a research scientist in the MediaTeam Oulu research group at the University of Oulu, Finland. He received his M.Sc. degree in Electrical and Information Engineering from the University of Oulu, Finland, in 2007. He is now studying on a Ph.D. degree at Oulu University, Finland. The research interests include P2P protocol optimizations for mobile and other constrained-capacity networks.

Jani Pellikka received his M.Sc. degree in Information Engineering from the University of Oulu, Finland, 2009. He is currently working as a research scientist at Centre for Wireless Communications, University of Oulu, and is conducting his Ph.D. research on Internet security and IP mobility protocols.

Dr. Mika Ylianttila is a Docent in Information Engineering and Information Network Systems at the Computer Science and Engineering laboratory and Associate Director at the Media Team Oulu research group, at the University of Oulu, Finland. He received his doctoral degree in 2005, and was appointed docent 2007. He has acted as pro tem professor 2005-2009, being responsible of Information networks degree programme, and has supervised 65 master thesis and one doctoral thesis. He has published more than 70 peer-reviewed publications. His research is focused on multimedia communication in ubiquitous information network systems. Networking research considers overlays, social and peer-to-peer networks, as well as content delivery and service computing. He is a Senior Member of IEEE.