25. Spontaneous Collaboration in Mobile Peer-to-Peer Networks

Andreas Heinemann, Max Mühlhäuser
(Darmstadt University of Technology)

25.1 Introduction and Motivation

The field of *mobile* Peer-to-Peer networks (MP2P) has various forms and currently there exists no coherent view on what is understood by it. The term *mobile* emphasizes that nodes/peers in the network are mobile, and therefore need to be equipped with some kind of wireless communication technology. Examples of nodes include pedestrians with mobile devices [284] or vehicles with wireless communication capabilities [632]. Since all mobile Peer-to-Peer networks construct an overlay over an existing wireless network, implementations range from MP2P over mobile ad hoc networks (MANETs) [156] to MP2P over cellular based networks [299, 298].

This chapter looks into a specific class of application for mobile Peer-to-Peer networks. Here the Peer-to-Peer network is formed by humans carrying mobile devices, like PDAs or mobile phones, with ad hoc communication capabilities. All presented applications exploit the physical presence of a user to support digital or real-life collaboration among them. The integration of wireless communication technologies like Bluetooth or IEEE 802.11b WiFi into mobile devices makes this kind of mobile Peer-to-Peer networks feasible.

As stated by Dave Winer, *"The P in P2P is People"* [624]; most of the Peer-to-Peer systems rely on the users' will to contribute. This could be observed in the success of first generation file-sharing applications like Napster [438] or Gnutella [252].

The key issue of user contribution prevails even more in mobile Peer-to-Peer networks, where in general anonymous users form the network with their personal devices. Resources on the device are typically limited. Especially battery power can be a problem. A user risks draining his battery by contributing its resources to other users. The device may also become unavailable for personal tasks, like accessing the calender or making phone calls. However user contribution may be stimulated by the usefulness of an application.

Currently, we see the emergence of several mobile Peer-to-Peer applications, both as commercial products and in research, as described in Section 25.2. As stated above, all these applications make use of the physical presence

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 419-433, 2005.

of a user to support collaboration among participating parties. This leads us to our hypothesis:

> Often, people in close proximity share a common goal or have a related motivation. Some of them may have valuable information relevant but unknown to others.

The common goal for these mobile Peer-to-Peer applications is to make this information available to other interested parties. Through this, we see new forms of spontaneous collaboration that can be distinguished in *active* and *passive* collaboration (cf. Section 25.1.2).

25.1.1 Mobile Peer-to-Peer Networks and Mobile Ad Hoc Networks

A thorough comparison of (infrastructure-based) Peer-to-Peer and mobile ad hoc networks (MANETs) has been carried out in [82]. The fundamental commonalities between the two are:

- *Decentralized architectures.* Neither network type relies on a central component (a centralized server).
- *Transient connectivity.* In both kinds of networks, nodes connect and disconnect unpredictably. In addition, MANET nodes are mobile and they move in and out of the communication range (which may appear to other nodes as a disconnection).
- *Heterogeneity of resources.* A mobile ad hoc network may be formed by different mobile devices, such as a laptop, mobile phone, or PDA. These devices typically differ in battery life, CPU power, and storage capacity. Likewise, computers that run the same Peer-to-Peer application typically vary in their specification.
- *Sharing of resources.* In both network types, a user actively shares her resources (battery power, CPU power, storage capacity, network connection) with others and the network exploits these resources to provide its services.
- *Identity management.* Both networks have to address the identification of any entity (e.g., nodes, peers, users, and content). This may also include privacy protection, in order to allow users to act anonymously within the network or application.
- *Routing and message forwarding.* Although MANETs handle routing on the network layer and Peer-to-Peer networks handle it typically on the application layer, routing remains a critical issue for both kinds of network.

On the other hand, there exist the following differences:

- *Network size.* Whereas MANETs are generally concerned with networks of a few hundred nodes, modern Peer-to-Peer networks can span much larger networks

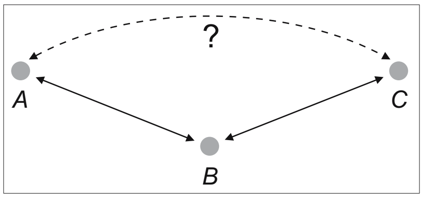


Fig. 25.1: Multi-hop communication.

– *Focus in the ISO/OSI Model.* Research and development of mobile ad hoc networks particularly focuses on the network layer. Several multi-hop routing algorithms have been proposed (see [484, 327]). Conversely, research in Peer-to-Peer networks addresses more the construction of overlay networks in the application layer.

As stated in the introduction, this chapter focuses on *mobile* Peer-to-Peer networks and applications. In many ways, mobile Peer-to-Peer networks are similar to MANETs, since they are also formed by individuals carrying a mobile device. Hence, the networks are of a similar size. However, there are certain important differences between the traditional MANETs and the mobile Peer-to-Peer networks we consider here. MANETs have been investigated in the context of military networks, emergency response, and sensor networks. These networks have several key characteristics in common, namely that *all nodes* in the network are strongly related to each other, trust each other, and share a goal they want to accomplish.

Mobile Peer-to-Peer networks, as we consider them, are formed between *anonymous groups of individuals*. This poses several additional challenges to the network. Consider the situation in Figure 25.1 with A, B, and C as mobile nodes, that is, individuals equipped with mobile devices. A is in communication range of B but not in range of C, who, on the other hand, is in communication range of B. If A wants to communicate with C, all traffic has to be routed via B. Bearing in mind that A, B, and C, a priori, do not know each other, two questions arise:

– What is the incentive for node B to route messages between A and C? Why should node B be willing to donate part of her battery power to enable communication between A and C ?
– Why should node A and C trust and rely on node B for their communication? Node B could easily eavesdrop, manipulate, or just reject messages.

Without an incentive scheme and extra security mechanism built in, the standard multi-hop communication schemes of MANETs appear to fall short of providing for communications in mobile Peer-to-Peer networks.

The alternative to the multi-hop MANETs are the so-called one-hop networks. In these networks, information is not forwarded over several hops;

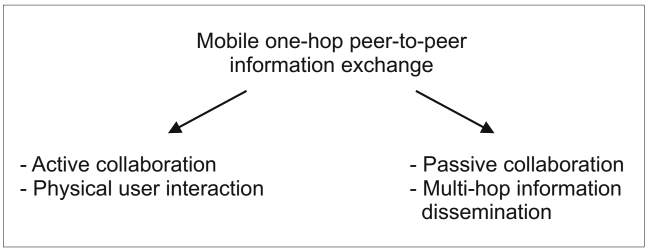


Fig. 25.2: Design space of mobile one-hop Peer-to-Peer communication

instead all communications always happen between two directly connected nodes, both of which have a *direct incentive* to participate in this communication.

We define mobile Peer-to-Peer networks to be such one-hop communication networks, where the individual nodes participate only when they have a direct interest in participating. This avoids the problems of incentives for traditional MANETs in standard situations. These one-hop networks will be the focus for the rest of this chapter. Note that limiting the communications to a single hop on *the network layer* does not limit multi-hop information propagation on *the application layer*, as we demonstrate in Section 25.4.1.

25.1.2 One-Hop Peer-to-Peer Design Space

Starting from the one-hop communication approach, the Peer-to-Peer design space for applications is divided into two general areas, as shown in Figure 25.2:

- **Active collaboration** focuses on the physical proximity of users. In addition to the exchange of digital information with users nearby, this allows using the device as a *link* to the user itself. Via non-intrusive user notification, this could lead to real world collaboration, such as a conversation. *Tap* and *tickle*, two form of digital gestures from the Socialight project (see Section 25.2.4), are a good example for non-intrusive user notification.
- **Passive collaboration** aims to collect and pass any kind of information to users in the vicinity without user interaction. Shark [553] and AdPASS [577] are the most prominent examples for this kind of application scenarios. Passive collaboration leads to on-the-fly information dissemination. In other terms, it is a form of digital *word-of-mouth* communication.

Applications that include both forms of collaboration are also possible.

The second important aspect in the design space is the relation among the users participating in the system. Questions to be answered are:

– Is the number of users known/fixed or unknown/open?
– Are the users identifiable or do they work with pseudonyms or even act totally anonymous?

Answers to these questions have impact on the usage and suitability of the mobile Peer-to-Peer building blocks we present in Section 25.3.

25.1.3 Chapter Overview

The rest of this chapter is organized as follows. Section 25.2 presents emerging applications for mobile Peer-to-Peer networks, namely in the following domains: enterprise knowledge management, spontaneous recommendation passing, conference collaboration, spontaneous encounter with friends and foafs[1], and spontaneous advertisement passing. Common building blocks for mobile Peer-to-Peer networks and applications are derived from these examples in Section 25.3. Following this analysis, the *iClouds* project is presented in Section 25.4. iClouds provides an architecture that includes building blocks for easier mobile Peer-to-Peer application development. This chapter concludes with Section 25.5.

25.2 Application Domains and Examples

This following list of applications is not meant to be exhaustive. Earlier work include the 7DS system [471, 472] and similar ideas can be found in more recent work [158, 256]. Nonetheless the examples present a variety of interesting and useful applications and are sufficient for the identification and extraction of common functionality.

25.2.1 Shark

Shark [553] supports management, synchronization and exchange of knowledge in mobile user groups. The system distinguishes three components: the *Shark Mobile Station*, the *Shark Central Station* and the *Shark Local Station*.

The mobile station runs on a mobile phone and is able to exchange knowledge with a nearby central station, local station or another mobile station. Bluetooth is used for ad hoc communication. The knowledge is stored using TopicMaps [319]. The central station manages the complete knowledge base.

[1] foaf = friend of a friend

Local stations store and manage only location relevant knowledge. This allows for simple location based services. Each local station synchronizes its knowledge with the central station.

One possible application for Shark is enterprise knowledge management. Here, each mobile staff member is equipped with a mobile station. Local stations are installed at subsidiaries. A customer (with his own mobile station) is able to learn about all kinds of information about the enterprise, e.g., product descriptions, price lists or white papers, while talking to a mobile staff member or visiting a subsidiary or shop.

25.2.2 MobiTip

The MobiTip [529] system allows its users to express their opinions on anything of interest in the environment. Opinions are aggregated and presented to the users as tips or recommendations. Opinions are entered in free text form on the user's device (a mobile phone) and shared in a Peer-to-Peer manner on-the-fly with users nearby using Bluetooth.

A typical example is a shopping mall, where MobiTip users share their personal views on certain shops or product offers.

The core MobiTip system can be extended by so-called *connection hotspots*. A connection hotspot is placed at a selected location, e.g., the entrance of a shopping mall, to collect tips and pass them to future visitors. This idea is similar to the time-shifted communication in the Socialight system based on *Sticky Shadows* (see Section 25.2.4).

25.2.3 SpotMe

A quite advanced collaboration system and tool for conferences, symposia, and corporate meetings is called SpotMe [561]. With a special purpose handheld device (with similar size to a mobile phone), each user can exchange information with other users in a Peer-to-Peer manner. Communication to a local server via base stations is also possible.

A user's device is personalized during the conference registration process. This step includes taking a photo of the attendee. The photo with other contact information is stored on the device and in a central database. Users can query this database during the event. Thus, they learn who is actually on site. The *radar* function allows a user to scan all other attendees in a range of 30 meters. With this information, a user can look for a conversational partner or simply identify the people sitting nearby at lunch. A user can specify special interest in another user. The device will then give a notice when this other user is nearby. This may help to start a conversation.

Several other functions require the communication with the local server, e.g., *last minute agenda update*, *news dissemination* and *questionnaires*.

SpotMe includes some basic post event services. The collected data and contact information is made available for every participant online on the web and sent as e-mail.

25.2.4 Socialight

Socialight [332], a mobile social networking platform that uses mobile phones, supports spontaneous encounter and interaction with friends and friends of friends. Using the current or past location of friends, Socialight enables real-time and time-shifted communication.

Location of users is determined by infrastructure based technology (GPS and Cell-ID), or ad hoc by signal recognition of Bluetooth devices nearby. Users have to register on a central platform before using Socialight. The platform also stores information about the social network of users.

Peer-to-peer communication among users may happen via *Tap & Tickle* or *Sticky Shadows*. Tap & Tickle are two digital gestures that allow users to exchange information by vibration of their devices. Pressing a button on a user's phone will make his friend's phone vibrate once (Tap) or rhythmically (Tickle). This is meant as a non-intrusive way to communicate with nearby friends.

With Sticky Shadows, users can attach digital information to a certain location. This digital information is recognized by friends when they pass the same location at a different time. Examples include restaurant reviews for friends, sales or shopping recommendations, and educational purposes, where teachers set Sticky Shadows for students.

25.2.5 AdPASS

AdPASS [577] is a system to spread digital advertisements (ads) among interested users. Each user specifies his interests in a profile that is stored on the mobile device. The communication scheme resembles the way information is spread by word of mouth between human beings, e.g., when recommending something to someone else.

As an incentive for users to take part in the system, AdPASS provides an anonymous bonus point model that rewards a user who carries an advertisement on the way down from the vendor to a potential customer.

AdPASS has three kinds of participants:

– A *merchant* disseminates digital advertisements within its vicinity. For example, there are several fixed nodes located in a merchant's shop. These *information sprinklers*, which are stationary transmitter units, are described

in [288]. Customer devices learn about advertisements while their owners are browsing the shop.
- A *customer* carries a mobile device (PDA). This device collects advertisements and transports or passes the ads to other interested customers. Ideally, some of them come to the shop and buy the advertised good.
- A *mediator* keeps track of the users' accumulated bonus points. It works similar to a central database that both the merchant and the customer can access (e.g. via the Internet). Thus, it guarantees reachability to the customers In addition, the mediator acts as an "anonymizer" to guarantee customers' anonymity.

Figure 25.3 illustrates the different communication steps. The participants interact as follows:

1. Customer A visits a merchant. While being in the shop, his device learns about several advertisements and filters them against his personal profile. The advertisements are stored on the user's device. In the example, customer A learns about a DVD advertisement.
2. Customer A (after leaving the shop) encounters another potential customer B (on the street for instance), who is interested in the ad. B stores the ad and then later passes it on to another interested user C.
3. C itself is taken with the ad, goes to the shop and buys the advertised good. C also passes information to the merchant about how he has learned about the ad – in our example, via A and B.
4. The merchant informs the mediator which customers should be rewarded bonus points.
5. Assuming that there is a Internet connection available, A, B, and C can download their bonus points from the mediator's server onto their devices. This can happen for example during a PC-to-device synchronization operation.

A detailed description about AdPASS, including how the bonus point system works and how user anonymity is provided, is given in [577].

25.3 Building Blocks for Mobile Peer-to-Peer Networks

Analyzing the example applications in Section 25.2 identifies a number of common functionalities among them. These building blocks for mobile Peer-to-Peer applications are described as services in this section.

Presence Awareness Service

Provides the application with information about which other nodes or users are currently active and in communication range. This service typically also provides some kind of neighbourhood information.

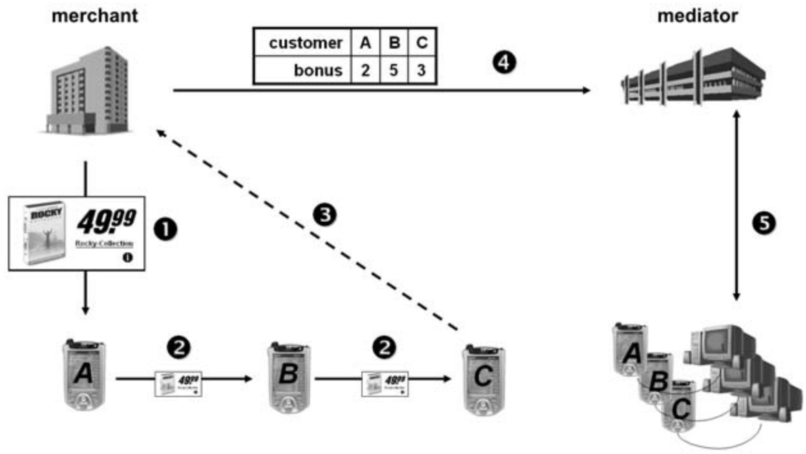


Fig. 25.3: Communication steps

Message Exchange Service
: A service that allows sending messages to and receiving messages from peers in the neighbourhood.

Information Filtering Service
: Since in mobile Peer-to-Peer applications there is also the danger of SPAM, there should be a way to filter out information that might not be relevant to the user. This functionality is provided by an information filtering service.

Information Distribution Service
: The information distribution service offers three functional choices. A peer can give information it receives straight away to other peers. The user may also review a received piece of information and decide on a per item basis whether to share it with other peers. Finally, the information may not be shared at all.

Security Service
: To support data or communication integrity, the security service offers sign and encrypt operation on information items. This may involve public/private-key cryptographic operations, based on some PKI or other trusted sources.

Identity Management Service
: The system design has to specify how a user appears in the system. Users can act anonymously, under a pseudonym or with assigned identities. The identity management service supports this design criteria.

	Project				
Service	Shark	MobiTip	SpotMe	Socialight	AdPASS
Presence Awareness	✓	✓	✓	✓	✓
Message Exchange	✓	✓	✓	✓	✓
Info. Filtering	✓	✓	✓	✓	✓
Info. Distribution	✓	✓			✓
Security					✓
Identity Management	✓	✓	✓	✓	✓
Incentive Schemes					✓
Reputation					
User Notification			✓	✓	

Table 25.1: Common mobile Peer-to-Peer service integration

Service for Incentive Schemes
As stated in the introduction, user contribution is an important issue in mobile Peer-to-Peer applications. Contribution can be stimulated by introducing some kind of incentive schemes for individual users. These schemes may vary from application to application, but may rely on common basic service functionality, e.g., accounting, to implement a certain incentive scheme.

Reputation Service
This service allows individual users to build a reputation within an application. Other users might value received information based on the reputation of the user who sends out information. This may especially be needed for systems with anonymous users. Currently, none of the presented projects in Section 25.2 use a reputation service. Nonetheless, this is an important service for future mobile Peer-to-Peer applications.

User Notification Service
This service instantly notifies the user of incoming information that may require some sort of instant reaction. For example, a real-life conversation with a discussion partner can only happen while the partner is nearby.

Table 25.1 summarizes the common services and their conceptual usage in the presented sample applications.

The next section presents the iClouds project. The project goal is to design a sound and coherent architecture for mobile Peer-to-Peer applications. This architecture integrates the identified set of common building blocks/services for mobile Peer-to-Peer application.

25.4 The iClouds Project

Within the scope of mobile Peer-to-Peer networks, the iClouds Project [289, 287] investigates several kinds of collaboration among mobile users. Based on the one-hop communication paradigm, the iClouds architecture separates common mobile Peer-to-Peer application requirements as services. Making use of these services, the architecture facilitate easy and rapid application development in this emerging area. We will describe the architecture in Section 25.4.3.

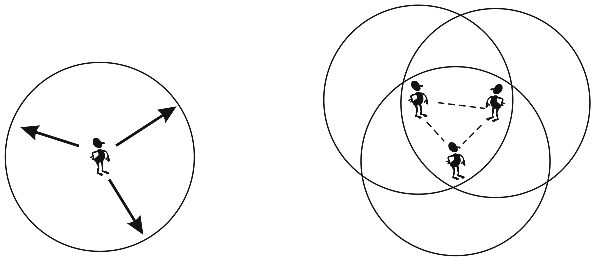
iClouds devices are small mobile devices, such as a PDA or mobile phone, with ad hoc communication support (Bluetooth or 802.11b WiFi). There is no need for any central servers in the iClouds architecture; instead, each device is completely independent. A special kind of iClouds devices are *information sprinklers*. An information sprinkler is mounted at a dedicated location to support simple location based services. Local station (Shark), base station (SpotMe) and connection hotspot (MobiTip) reflect similar concepts.

The diameter of the iClouds communication horizon (see Figures 25.4(a) and 25.4(b)) should not exceed a few hundred meter. iClouds users are given the opportunity for spontaneous collaboration. When two iClouds users “see” each other, they should be within a short walking distance from each other (a couple of minutes at maximum).

Following the assumption form the introduction (cf. Section 25.1.1), we specifically exclude multi-hop communication on the network layer. All communications happen directly between the concerned parties. On the application layer, iClouds supports indirect multi-hop information dissemination,as described below.

25.4.1 Multi-hop Information Dissemination

Information in iClouds can pass through several nodes, as shown in Figure 25.5 and Figure 25.6, assuming certain conditions are met. We call the first scenario *information passing*. Information can only pass between A and B or between B and C, since A and C are not in communication range. Their communication horizons overlap, but the devices themselves do not fall into the other’s communication horizon. In this situation, information can pass from A to B and on from B to C if and only if it passes from A to B, and it would get to C if B and C were in communication range and have interest in the same information provided by A.



(a) Communication horizon

(b) iClouds with 3 peers

Fig. 25.4: Information clouds

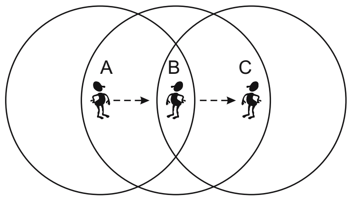


Fig. 25.5: Information passing in iClouds

The second scenario in which information is transported with iClouds involves physical movement of nodes, as shown in Figure 25.6. This is called *information moving*. In Figure 25.6, B gets some information from A. B later meets C, who is interested in the same information and B passes it to C. Again, a prerequisite for the information to get from B to C is that B and C share interest in the same information.

25.4.2 Data Structures and Communication Semantics

The two most important data objects found on the iClouds device are two information lists (*iLists* for short):

- *iHave*-list (information have list or information goods):
  The iHave-list holds all the information the user wants to contribute to other users.
- *iWish*-list (information wish list or information needs):
  In the iWish-list, the user specifies what kind of information he is interested in.

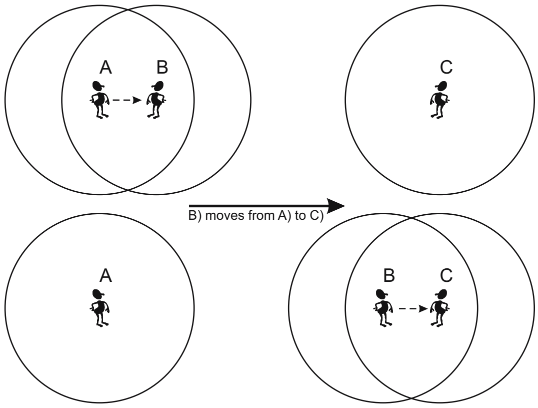


Fig. 25.6: (Physical) Information moving in iClouds

	pull (from Bob)	push (to Bob)
iHave-list	Standard search	Advertise
iWish-list	Active service inquiry	Active search

Table 25.2: Information flow semantics (from Alice's point of view)

Each iClouds device periodically scans its vicinity to see if known nodes are still active and in communication range and also to see if any new nodes have appeared. Information about active nodes is stored in a neighbourhood data structure.

By exchanging iLists, the iClouds devices align their information goods and needs. Items on the iWish-lists are matched against items on the iHave-lists. On a match, information items move from one iHave-list to the other.

For example, consider two iClouds users, Alice and Bob, who meet on the street. When their iClouds devices discover each other, they will exchange their iHave-lists and match them locally against their iWish-lists. If an item on Bob's iHave-list matches an item on Alice's iWish-list, her iClouds device will transfer that item onto her iHave-list.

There are two main communication methods for transferring the iLists. Peers can either *pull* the iLists from other peers or they can *push* their own iLists to peers they meet. Either of these two operations is applicable to both lists, which gives four distinct possibilities of communication. Table 25.2 summarizes these possibilities, along with their real-world equivalents.

In each of the four cases shown in Table 25.2, the matching operation is always performed on the peer who receives the list (Alice's peer in pull and Bob's peer in push). A key strength of iClouds is that each of the four possible combinations corresponds to an interaction in the real world:

– *Standard search.*
  This is the most natural communication pattern. Alice asks for the information stored on Bob's device and performs a match against her information needs (specified in her iWish-list) on her device.
– *Advertise.*
  This is a more direct approach. Alice gives her information goods straight to Bob and it's up to Bob to match this against the things he is interested in. As an example, consider an iClouds information sprinkler mounted on shopping mall doorways pushing advertisements onto customer devices when they enter the building. This is implemented in the AdPASS system (cf. Section 25.2.5).
– *Active service inquiry.*
  This is best suited for shopping clerks. They learn at a very early stage what their customers are interested in. An example of this query could be: "*Can I help you, please tell me what are you looking for?*".
  In general, especially for privacy reasons and user acceptance, we believe it is a good design choice to leave the iWish-list on the iClouds device. Hence, this model of communication would likely be extremely rare in the real world.
– *Active search.*
  With active search, we model the natural "*I'm looking for X. Can you help me?*". This is similar to the standard search mechanism, except that the user is actively searching for a particular item, whereas in the standard search, the user is more passive.

25.4.3 Architecture

Figure 25.7 shows the architecture that is proposed and used in iClouds. There is a general distinction between a communication layer and a service layer. The communication layer provides simple one-hop message exchange between peers in communication range. A neighbourhood data structure keeps track of active peers in the vicinity.

The common services are located on the next layer. Each service can use functionality provided by other services or by the communication layer below. Note that the service layer is extensible for new services that might be needed by future applications.

The applications reside on the topmost. To fulfil its purpose, an application has access to both the service and the communication layer.

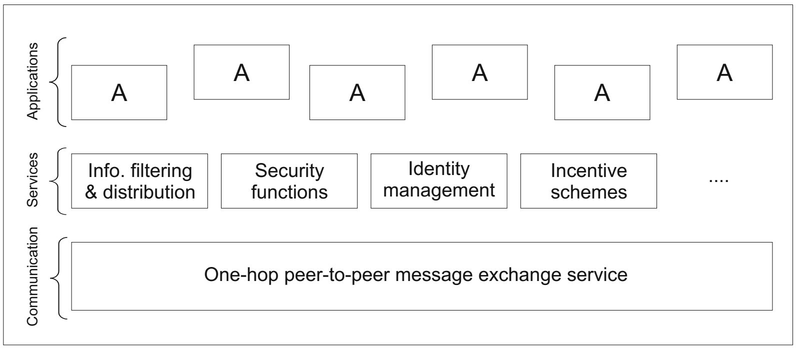


Fig. 25.7: iClouds architecture

25.5 Conclusion

This chapter points out that there are several similarities in mobile Peer-to-Peer applications. The analysis of emerging applications in this area identifies a set of common services that serve as basic building blocks.

The iClouds architecture aims to provide a framework for mobile Peer-to-Peer application developers who do not want to re-invent common functionality over and over again. The architecture is implemented in Java as a lightweight set of classes and runs on Java2 Micro Edition compliant mobile devices with 802.11b WiFi communication support.