A Platform for Location-Aware, Ad-hoc Collaboration in Wireless Networks

O. KAO and U. RERRER
Department of Computer Science, Paderborn University
Fuerstenallee 11, 33102 Paderborn, Germany
{okao|urerrer}@upb.de

Abstract – *In this paper we present a prototype for a collaborative work in wireless networks, which supports a location-aware communication in a neighbourhood defined by a given access point. All participants of this virtual community can ad-hoc – without any previous exchange of contact information – interact via messaging, file-sharing, and conferencing services. Users can join the community as soon as they enter the network zone around the access point or leave it at any time without affecting the platform. For the user interaction a general protocol is developed, which can be adapted to various applications types. A description of the applications messaging and file-sharing in the location-aware environment closes the paper.*

1. Introduction

Recent advances in wireless technology together with demands for an extensive user mobility led to the development and installation of numerous mobile ad hoc networks. Those networks are self-organizing wireless networks where the nodes cooperate to dynamically establish communication using technology like wireless LAN (WLAN), Bluetooth or infrared sensors. Meanwhile, various devices such as PDAs, cell phones, etc. can be used to access and use these types of networks.

In this paper we assume a wireless network with an access point (AP), which receives the connection requirements from mobile devices and supplies these with the necessary access information. A sample is given by a WLAN AP in lecture / conference rooms, coffee shops, airport launches, etc. The AP defines – due to the spatial extend of its signal – a local neighbourhood for the integrated devices. This locality can be used to satisfy communication needs between neighbours and to enhance the collaborative work by providing conferencing tools or sharing facilities limited to this neighbourhood.

The restriction to a *local neighbourhood* is opposite to the main idea of the Internet to bridge the gap between spatially divided users and resources. However, it also addresses a different scenario, which is very common in the everyday life: people contact other people on different locations and start a conversation. They share documents for entertainment (e.g. holiday photos), to learn (e.g. lecture slides) or – most common – to collaborate and to develop products together. All this is possible in the Internet too, but the users have to know the person to be contacted and collect the access information, e.g. e-mail or ICQ number *before* the meeting. Thus, this approach is not suitable for ad-hoc communication. Therefore, we address a scenario, where users contact other members and establish collaboration *without* *knowing each other a-priori*. This applies to all members, who are currently in the range of the AP and are willing to communicate. The latter is announced by installing the developed client and setting the security parameters accordingly. A further grouping of the registered users is allowed as well and can be realised in different ways. For example, users in a video conference establish an own group. The existence of this group is announced to all members in the network zone, who can subsequently join the group. In a simple case no further query is required, as the group is open to all members. The advanced registration mode requires an approval by the creator or any other member of the group, who can decline a connection requirement with the advice, that the group is restricted to certain community due to privacy reasons. The underlying principle is well-known from mailing lists in the Internet, which can be denoted as open or closed. The latter restricts the usage of the list to users knowing the necessary password. Similar approaches are also possible for chatting, shared whiteboard, etc.

An often occurring situation is depicted in Figure 1. All members in the network zone build a global group, which is subsequently divided into sub groups (Group A, B, C in Figure 1). An individual member can – if he/she owns the necessary rights – become a part of a single or of multiple sub groups. The affiliation of each member is thus defined by the community itself.

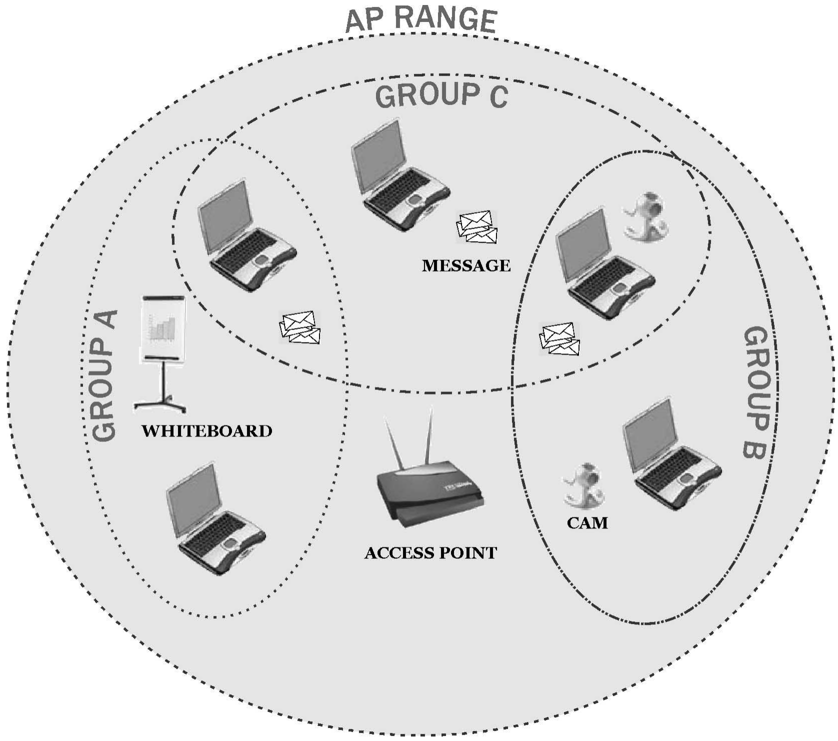


Figure 1: *Member affiliation to the network zone and individual sub groups*

The design of such highly dynamic systems requires application / development of novel approaches for addressing, networking, and collaboration. In particular, all network information must be distributed over all members in the local wireless network, as users can quite the membership at any point of time and leave the network. The redundantly stored information ensures the functionality of the network, until the last member leaves. A general design and a working prototype for collaboration in wireless LANs, which allows the described ad-hoc communication and offers a number of services such as instant messaging, whiteboard sharing, audio/video conferencing and file-sharing is the main goal of this paper.

The rest of the paper is organised as follows. After the introduction of two sample scenarios and the analysis of related work in Section 2, a general overview of the architecture is given in Section 3. Section 4 describes collaboration services such as messaging and files sharing as examples. Section 5 introduces the current work in progress related to additional services for the platform. A presentation of the prototype is found in Section 6.

2. Sample Scenarios and Related Work

This section introduces two often occurring scenarios related to collaboration in a class room and in a public place such as coffee shop. However, similar situation can be found on many other places starting from conference rooms, via airport launches and train departments until hot spots in shopping areas in downtown.

A sample class room scenario starts, when the teacher enters the room and his notebook joins the community of all currently active members (students) in the neighbourhood of the AP. Subsequently, the teacher opens the folder with the documents for the lesson and shares the folder with all members registered by the AP. The students can now load the documents and display them on the local screen of their own notebooks. Moreover, the shared whiteboard replaces the traditional board with the advantage that the board content can be archived locally. Students can ask questions by pointing a specific detail on the whiteboard or they can exchange knowledge by chatting. This scenario is realised without a necessity to agree on a server, to upload the documents, and to distribute passwords before the lecture starts, which is the common way of resource sharing in the Internet.

Another scenario considers a coffee shop supplied with WLAN, which is meanwhile available in large number of shops world-wide. Guests with a mobile device already access the network in order to surf the web, read e-mail, etc. With the developed prototype they can now establish a local community and offer selected personal information to their neighbours, e.g. hobbies, profession, even link to the own home page, etc. This information is visible for all active members in the coffee shop network, who can instantly contact the person by sending a chat/audio/video message. Thus, a real ad-hoc interaction is allowed, which can be followed by exchange of e-mail addresses, phone numbers, etc. This is the only way to stay in touch after leaving the coffee shop, as all access data will be deleted / assigned to the next guest. So, any guest – if desired – can keep the privacy in the real word outside the coffee shop.

A number of systems [1, 2] offer many of the services mentioned in the scenarios. However, most of these systems require a central server, where all users have to register and to log-in. The server manages the contacts with the current status (on-line, off-line, ...), processes connection requests and notifies the user. Additional information regarding personal interests, location information, or grouping the users contacts are all saved on the server. Peer-to-peer systems decentralise the shared information.

KORTUEM ET AL. recently presented the Proem System [3] as a peer-to-peer platform for mobile ad hoc applications, which uses proximity-aware mobile collaboration by detecting other peers called peerlets. KATO ET AL. created Promise [4], which is an architecture adapted to mobile networks with support of a multicast communication and focus on the developed protocol. However, both platforms are designed for collaboration in peer-to-peer networks and create their own protocol. They use peer-to-peer architectures but do not integrate location-aware services which are the main issue in our research.

3. Architectural Overview

The proposed system architecture is an adaptive, universal and extensible platform for collaboration which uses the neighbourhood information of access points to provide location-aware services. It is designed for mobile wireless LANs and includes two fundamental types of components:

- **Peer:** A peer is any mobile host or device equipped with network adapter, which can potentially join the ad-hoc community.
- **Access point:** APs are part of the network infrastructure. An AP sets a border for the current location-aware community by the spatial extend of the radio signal (in following named *network zone*).

The basic workflow starts as soon as a new peer enters the network zone. It sends and receives a sequence of discovery packets, which allow to detect the most suitable AP in that zone and to register subsequently. The MAC address of the new member is stored in a list of the AP, which also contains the MAC addresses of all already registered and active members in the assigned network zone. Depending on the AP manufacturer further meta-information e.g. the corresponding IP addresses, signal quality or bandwidth are also disposable. After registration, the communication middleware downloads the complete list with the information from the AP using the simple network management protocol (SNMP) [5]. This protocol was developed to gather information from network components and to apply configuration changes remotely. Every peer member polls this availability list from the AP in certain, short time intervals (e.g. every 5 seconds). Thus, each member has always an overview about, which peers are active in the network zone and which location-aware services are available.

SENDER	RECEIVERS	TYPE	DATA	OPTIONS
host	host[]	string	object	string[]

host:

NAME	MAC	IP
string	string	string

Figure 2: *Schematic of the general communication protocol*

Furthermore, every member in the network zone can use the list information to send a message to every other member in the zone. For this purpose a general protocol is defined, which builds the essential part of the architecture. The protocol has a simple and adaptive structure shown in Figure 2 and supports solely one universal packet type for message transmission. The packet includes the receiver and sender address, the message type (chat, video, audio, etc.), the message body and additional settings as well.

Each client uses three pre-defined messages to establish/close communication with the community:

- **hello:** This is the first message, when a new peer requires access to the current community. It indicates that the communication middleware is available and ready.
- **open:** This message establishes a communication between two peers in the community. The source peer announces that further content messages will follow.
- **close:** By sending the message close, the communication between two peers in the community is terminated.

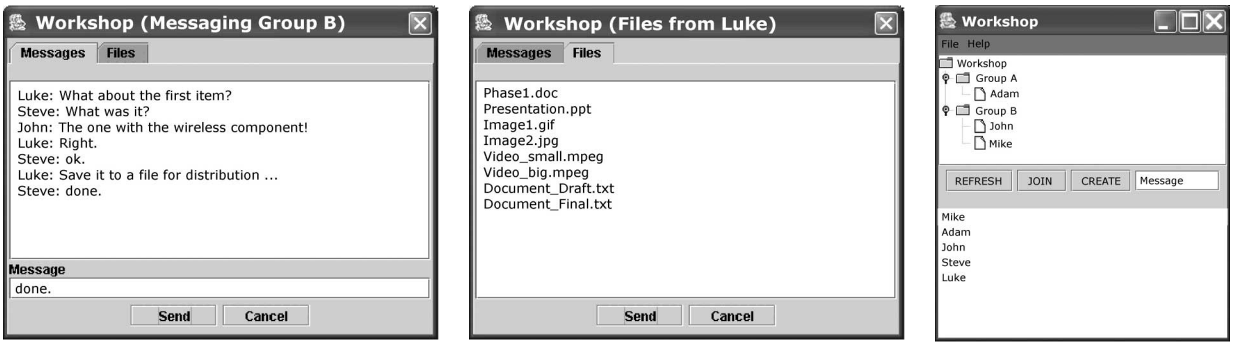
All other messages are specific to the platform extensions / applications described in the next Section.

4. Messaging and File-sharing

In this section two basic applications are presented exemplarily - a messaging service and a file-sharing service.

Sending messages is simple and well-known from chat programs [2, 6, 7]. The software on every peer quotes a GUI for human interaction with the system. The member list is shown with freely defined names for each peer. To create a message a click on the receiver's item – member or group - is necessary. A typing window will appear and stay on top of the GUI for succeeding messages to the same peer. If a message is sent, a window on the receiver's side will appear. Both windows remain displayed on the screen and create a scrollable message history. Closing the windows ends the message-based communication between the peers. A grouping mechanism is also implemented where users can join or create system-wide groups. Sending one message to all group members via multicast mechanism is also available. Figure 3 shows screenshots of the messaging, file-sharing and grouping components of the GUI.

Transferring files is nearly as easy as sending messages but to support reliable large file handling some further considerations have to be made. Similar to the messaging a window concerning the file transport is created. The user has to define his own file list containing the files to be shared with the community. A request *get file* is created by entering a file name into the search dialog and choosing a dedicated file from the displayed results or by selecting a specific file from the users file list. Subsequently, the download is initiated.



(a) (b) (c)

Figure 3: *GUI screenshots for a) messaging, b) file-sharing, and c) grouping mechanism*

The file is divided in packets of fixed size annotated with a checksum. This ensures correct and save transportation over unreliable networks like wireless LAN. The transportation extends the protocol for a number of packet types. The *file list* comprehends the available files of a peer. The *file request* package is send if a user requests a file with a specific name from another peer. To transmit large files three packages are used. *File* is the first chunk of the file initiating the transport followed by a number of *file data* packets containing sequential chunks of the file. A *file end* packet completes the sequence.

Further already developed tools address shared white boards, video/audio conference between the members and shared cache for all current members accessing the Internet.

5. Prototype

The prototype is developed in Java and realises an extensible, module-oriented architecture. New services can be included using existing templates, thus additional functionalities can be integrated easily.

The prototype must be installed in order to establish the collaboration in a given network zone. The software includes the implementation of the general protocol and the applications to be used. Beside the common ways such as distribution on CD, download from a given web site manually or automatically via web installer, it is planned to offer an intuitive and comfortable AP-supported installer. The AP checks for the software availability on a new member of the network. If the collaboration platform is not active, the AP establishes a link to the web page and activates the web component, which will install all required components automatically. This can be achieved by implementing a script into the firmware of the AP or using a software-based access point. Subsequently, the user can join the community immediately.

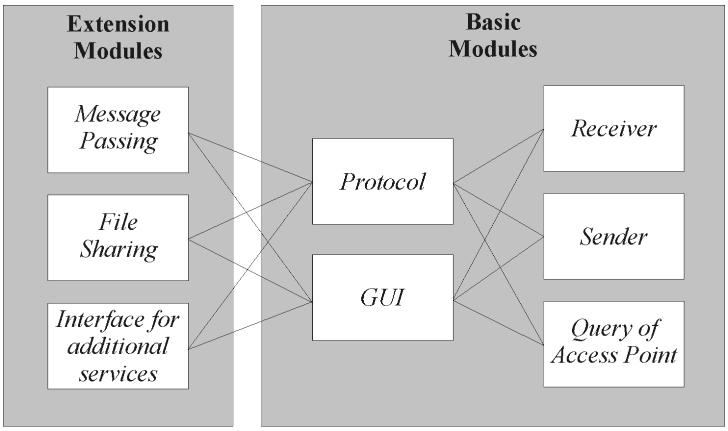


Figure 4: *Overview of the global architecture and the communication between basic and extension modules*

Figure 4 depicts a global overview of the underlying architecture. The modules are divided into basic modules, which implement the communication platform and into extension modules, which realise the additional collaboration services. The basic modules have three main tasks. The processing starts with a registration at the access point and a query for the already registered member in the network zone. The query component implements the SNMP protocol and returns a data structure with the MAC and IP addresses of the other members, which are then used for establishing a connection.

The packet transmission is provided by the modules *sender* and *receiver*. The sender opens a socket connection to the target member and sends the packets, which are prepared by the module *protocol*. The receiver waits for incoming request, accepts the socket connection, receives the packets and forwards these to the module *protocol* for further processing. The module *protocol* provides the packet management and implements the core messages for establishing, realisation and closing of connections as mentioned before.

The acquired contact information of the members in the network zone is used during the direct member communication. Each member can now communicate to any other member by sending the core messages. The receiver analyses the request, compares it to the own security and group policy, before it finally accepts or declines the connection. If the connection is established, the incoming packets are forwarded to the appropriate extension module, in this case messaging or file-sharing.

The interface for additional services contains a template for an intuitive and flexible integration of further collaboration applications. The suitability of this interface was already successfully tested by introducing services for video conferencing and a shared white board to the system.

6. Current Work in Progress

The presented platform implements two basic location-aware services so far, messaging and file-sharing. Using established efficient multimedia techniques other services will be useful for the described scenarios in Section 2. The aspect of location-awareness in a wireless context enables a great deal of possible services and applications. Currently, following two multimedia services – video/audio conferencing tool and a shared whiteboard for a local WLAN environment – are already implemented. Moreover, an infrastructure for a seamless access to output devices such as printer is realised, but still not extensively evaluated. Finally, a central information service for the current WLAN neighbourhood is currently designed. In following, the extensions are briefly described.

- **Video/audio** conferencing tool. For computer supported collaboration systems more than messaging and file-sharing is required in order to simulate a "natural" working environment known from offices in the real world. Seeing and hearing the collaboration partner transports significant additional information through mimics and gestures than a simple text. Tests on the prototype platform showed promising results in performance for multiple video connections over a single AP using standard protocols (Real Time Protocol, RTP) [8, 9] and easy-to-use frameworks (Java Media Framework, JMF) [10].
- For brainstorming techniques and efficient conversations in small groups a **whiteboard** is essential. In the workshop or the teaching scenario discussed earlier an interactive shared whiteboard would cover this aspect. As in Microsoft's Netmeeting [1] features like using different colours per user, inherit objects from offline members, supporting tools for drawing, marking, erasing and writing have to be provided. A significant design change is caused by eliminating the central server, thus the content of the shared board have to be stored locally on each mobile device and every change have to be sent by broadcast to all members of the whiteboard group. Thereby, additional synchronisation aspects have to be considered.
- Also useful are services that arrange without further configuration access to local I/O devices like **printer** or **scanner**. The Java extension Jini manages the discovery and sharing usage of transparent resources in a highly dynamical, distributed environment. Considering more than one AP a user could e.g. submit a printing job to the communication platform. The software discovers due to the users location and printing requirements (dpi, colour depth, format) the nearest available printer. The job is automatically configured and executed at the selected hardware and the user receives an acknowledgment message containing a map with the printer location.
- An **information and discovery service** for each network zone leads to a new, higher level in the design hierarchy, as it contains information about all low-level services but also serves as a gateway to the world-wide Internet. The main goal of such a service is to provide the user in the network zone with information regarding the local

environment, for example the position of the next printer, a map of the building with marked WLAN hot spots, restaurants, rescue stairs, etc. Moreover, it should be able to get the departure plans for the bus/train station nearby or display the cinema schedule on the mobile device. All this has to be provided without necessity for log-in or for additional server information. As soon as the user enters the network zone, a connection from the mobile device to the discovery service is automatically established and the user can select any of the provided services.

These ideas will be integrated in further extensions to the platform in the next time. Also the described services offer many aspects for system enhancement.

7. Conclusions

The presented prototype inverts the Internet-based interaction model and allows an ad-hoc communication in specific network zones defined by an access point in a wireless environment. Participants in the range of the AP can register in the virtual community and immediately interact with other members without any prior exchange of contact information or agreeing on a dedicated central server. In this paper the general architecture of the communication platform together with two sample applications – messaging and file-sharing – for collaboration are described in detail.

Future work includes the development of additional services, enhancement of the software download from the access point, and integration of discovery services in the virtual community.

References

[1] Website Microsoft Netmeeting, http://www.microsoft.com/windows/netmeeting

[2] Website ICQ, http://www.icq.com

[3] G. Kortuem, J. Schneider, D. Preuitt, T. G. C. Thompson, S. Fickas and Z. Segall: *When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad Hoc Networks*, International Conference of Peer-to-Peer Computing (2001).

[4] N. Ishikawa, T. Kato, H. Sumino, J. Hjelm, Y. Yu, Z. Zhu: *A Platform for Peer-to-Peer Communications and its Relation to Semantic Web Applications*, Proceedings of the Workshop on Semantics in Peer-to-Peer and Grid Computing (2003).

[5] M. Schoffstall, J. Case, M. Fedor and J. Davin: *A Simple Network Management Protocol*, RFC 1157, IETF Standard (1990)

[6] Website AOL Instant Messegner, http://www.aim.com

[7] Website Microsoft MSN Messenger, http://messenger.msn.com

[8] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson: RTP: *A Transport Protocol for Real-Time Applications*, RFC 3550, IETF Standard (2003)

[9] H. Schulzrinne, S. Casner: *RTP Profile for Audio and Video Conferences with Minimal Control*, RFC 3551, IETF Standard (2003)

[10] Java Media Framework API Guide, SUN Microsystems Inc. (1999)