A NETWORK COMPONENT ARCHITECTURE FOR COLLABORATION IN MOBILE SETTINGS

Bernd Eßmann, Thorsten Hampel, Thomas Bopp
Heinz Nixdorf Institute
University of Paderborn
Fürstenallee 11, 31102 Paderborn

Keywords: Mobile Computing, Spontaneous Collaboration, Distributed Knowledge Spaces, CSCW.

Abstract: Today *Computer Supported Cooperative Work* (*CSCW*) is used in broad areas of human cooperation. With the propagation of radio-based communication and ad hoc networking it may enter new areas of human cooperation. One important aspect is the new quality in CSCW of being independent from special network-enabled places. Another aspect is the more intuitive support of face-to-face cooperation utilizing personal mobile devices. To open this field of collaboration our approach featuring *Distributed Cooperative Knowledge Spaces* specifically addresses conceptual issues pertaining to the transition from classical, server-centered to mobile, distributed collaboration environments. With this concept we introduce *persistent* and *personal knowledge spaces* as well as so-called *temporary knowledge areas* and *groups*. Our prototypical application for spontaneous collaboration implements this approach. We are able to draw here on many years of experience in the development and testing of our concept of *Cooperative Virtual Knowledge Spaces*.

1 INTRODUCTION

In recent years, the CSCW world has created several cooperation environments to support inter-human work using network technology provided by the Internet. We, for our part, have developed the *sTeam* system, based on the concept of *Cooperative Virtual Knowledge Spaces*. *Knowledge Spaces* allow users to structure their knowledge spatially in virtual rooms (also called "*areas*"). *sTeam* is designed as a client/server system with a central server providing the cooperation environment, which is accessible from any device connected to the Internet. The idea is that users should be able to cooperate independently of specific locations such as conference rooms.

The need for collaboration environments that are independent of specific locations is the reason for our conviction that network-supported cooperation will change in the near future. Ad hoc networks are encouraging the emergence of this feature. Such networks allow devices to be connected spontaneously without planning an infrastructure. This makes network-based cooperation environments independent of network-connected locations.

One problem we face when using these new network topologies is that data is often distributed across several devices. For our cooperation environment,this means that the *Knowledge Spaces* of the cooperation partners are also distributed across their devices. The original *sTeam* system ensured consistency of the underlying data by managing it through one server. It was not planned to store the data on multiple servers. This means that areas of the *Distributed Knowledge Spaces* can be connected across several devices.

An additional problem is that the availability of devices in an ad hoc network is not guaranteed because its structure is highly dynamic. This means that parts of the *Knowledge Spaces* could also disappear when hosted on a vanishing device.

One of our goals, then, is to expand our architecture to cover these network requirements, allowing us to create *Knowledge Spaces* across several devices without much need for administration by the users. Another is to support both the classical, proven concepts of CSCW and new concepts for spontaneous networking. By doing so, we seek to transfer our well-proven concept of *Cooperative Virtual Knowledge Spaces* to such a network environment.

After consideration of related work, Section 3 of this paper describes future network structures based on *MIP* and *Multihop Ad Hoc Networks* (*MANETs*).

Such a new network structure will change the pos-

Eßmann B., Hampel T. and Bopp T. (2004).
A NETWORK COMPONENT ARCHITECTURE FOR COLLABORATION IN MOBILE SETTINGS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 337-343
DOI: 10.5220/0002622503370343

sible concepts for the desired cooperation environment. For this reason, we discuss the new concept of *Distributed Knowledge Spaces* in Section 4. This new concept extends *Cooperative Virtual Knowledge Spaces* to include distributed environments like *MANETs* without neglecting existing structured networks like the Internet. In addition, this concept is capable of combining both network topologies and enriching the concepts for classical infrastructures.

These two factors, the network structure and the concepts, influence the design of the application for the cooperation environment. We introduce and discuss this design in Section 5. We present the technologies used, *JXTA*, *mDNS* and *sTeam*, as well as initial prototypes for evaluating the technology and our concepts.

In the final section, we discuss our experience so far and look at the tasks ahead.

2 RELATED WORK

Using mobile devices in cooperation scenarios is an attractive prospect. Our research focuses on developing a collaborative working environment using spontaneously connected devices with the option of accessing services in structured networks like the Internet. Many projects explore the use of spontaneous networks.

The *ELAN* project[1] aims to develop an infrastructure for e-learning in mobile ad hoc networks (Lauer and Matthes, 2002). Unlike our approach featuring distributed knowledge spaces for collaborative knowledge construction in mobile settings, this project focuses on pure e-learning issues in these networks.

The project *ConcertStudeo* being conducted by the *Fraunhofer Integrated Publication and Information Systems Institute's (IPSI) Concert work group* combines an electronic blackboard with handheld devices for enhancing face-to-face learning using interactive learning tools (Wessner et al., 2003). This approach is based on special tools (e.g. voting tools) for specific learning settings. The cooperation concept focuses on e-learning only; it does not use ad hoc networks for the learning environment.

Another *IPSI* project, called *MIRIAM*, aims to set up a reference installation for Mobile IP over IPv4 (MIPv4). Concrete installations based on MIPv4 are evaluated in an application scenario for scientists (Berier 2001). The results of this project can help us to select a suitable network for our research needs. Although the project analyzed mobile networking, it did not address ad hoc networks or focus on e-learning scenarios.

The *MIT* has a research project called *Oxygen* addressing various aspects of pervasive computing. *Oxygen* seeks to enable pervasive, human-centered computing by combining specific user and system technologies. The term ”ambient intelligence” is used in this context (Dertouzos, 1999; Hanssens et al., 2002). The goal is to build an intelligent ambient environment enriched with a collaboration and knowledge-access subsystem. This project does not exploit the potential of *Cooperative Virtual Knowledge Spaces* for knowledge structuring and management as we do.

The above projects focus on one or two aspects of the research needed to build, as our approach seeks to, an environment based on ad hoc established and structured networks allowing users to tailor their own *Cooperative Virtual Knowledge Spaces* with a minimum of administrative effort. What we need is a comprehensive concept of *Cooperative Virtual Knowledge Spaces* as in classical sTeam, ported to mobile distributed environments. This concept must take into account hardware issues, network technology and application frameworks, enabling scaling from powerful hardware down to ultra-portable devices like PDAs.

3 NETWORK TOPOLOGIES SUPPORTING MOBILITY

An important factor in the design of the collaborative work environment is the nature of the underlying network structure. Put another way, the design of the collaborative system influences the choice of network structure. In our case, the idea is to use available network technology, adapting it to match future network structures and meet the needs of mobile learning environments. In our view, the network infrastructure of the future must offer both maximum independence from established infrastructures and the option of accessing classical networks like the Internet.

As cooperation occurs spontaneously, it should not depend on existing network infrastructures. The network infrastructure must be built from scratch, with no need to configure the network devices manually. Ad hoc networks meet these requirements. They set up the network devices and establish routing schemes appropriate to the network structure. Ad hoc networks are able to maintain changing network infrastructures on the fly when devices move within the network or disappear.

Although ad hoc networks would appear to be the ideal solution for spontaneous collaboration, mobile users still benefit from connecting to classical net-

[1]The ELAN project is part of the *German Research Foundation's* (*DFG*) *Priority Programme 1140 ”Basic Software for Self-Organizing Infrastructures for Networked Mobile Systems”*.

work infrastructures like the Internet. Many useful services are only available in classical networks, and remote cooperation over long distances can be achieved through Internet connection. This is why we wish to retain the option of accessing the classical *sTeam* server, when reachable.

By combining these technologies, three criteria are met:

1. spontaneous networking without existing network infrastructure
2. the use of existing network structures, if reachable
3. availability of the technology in near future, based on already existing network structures and technologies

In a setting incorporating ad hoc and structured networks, reachable devices must be contactable through both network topologies. A unique worldwide contact address is needed because a device can move freely within structured and ad hoc networks. Movement within structured networks is called *macro* or *inter-domain mobility*, movement within ad hoc networks *micro* or *intra-domain mobility*.

The *Internet Engineering Task Force* (*IETF*) work group's *Mobile IP* (*MIP*) is a network technology providing macro mobility and reachability, irrespective of the network point to which the device is connected (Perkins, 1997; Perkins, 1998). MIP was originally designed for IPv4 (*MIPv4*) and can be used for IPv6 (*MIPv6*) (Johnson et al., 2003).

While MIP supports macro mobility, it is not suitable for micro mobility and lacks independence from established networks. Mobile ad hoc networks (MANETs) are therefore used for building networks without the need to connect to established networks (Perkins, 2001). If an MIP agent is in the range of the MANET, devices able to connect to the agent can provide access to the structured network as a service to the other devices in the MANET (Tseng et al., 2003). In this case, a bridge between the two networks is established. As soon as this bridge exists, all clients of the ad hoc network are able to connect to their home agent and are reachable from outside . Conversely, the loss of the bridge does not affect the ad hoc network; communication between the ad hoc-attached devices is still possible (Campell et al., 2002).

In recent years, several routing protocols for MANETs have emerged. They allow establishment of the previously discussed spontaneous networks. So far, however, none of them has been established as a standard for ad hoc networking[2]. The only protocol for establishing ad hoc networks available in most operating systems is the *link-local protocol* (Cheshire et al., 2003)[3]. The main drawback of the link-local protocol is that it is not a multi-hop protocol. This means that it is not able to build larger ad hoc networks. Our approach is to use it for spontaneous networking in small groups, which meet at one location and are within range of all other radio network devices.

For bigger scenarios, it should be possible to establish multi-hop networks like the one mentioned in (Perkins, 2001), where every device can be a router for other devices, which would allow ad hoc networks to be established over large areas. This would allow connection to the Internet with mobile IP. Our short-term goal is to create some test installations of this network to evaluate them in everyday use.

In the described networks, client/server applications would make no sense because loss of the server from the network would render all affected clients useless. Unlike client/server structures, peer-to-peer technology meets the needs of such a network topology perfectly.

Knowing the characteristics of the network infrastructure for the learning environment, we must integrate our current cooperative knowledge spaces into this combination of structured networks and MANETs. Our goal is to integrate the current concepts of cooperative knowledge spaces into this innovative network topology.

4 CHALLENGES OF COOPERATIVE KNOWLEDGE SPACES IN MOBILE ENVIRONMENTS

The concept of *Cooperative Virtual Knowledge Spaces* has been successfully applied for a number of years. Our work aims to provide learners with *Distributed Knowledge Spaces* based on innovative network infrastructures for mobile devices. We begin by evaluating the potential of handheld devices to provide user interfaces for virtual knowledge spaces, go on to address the task of building an environment for mobile collaborative learning, and finally discuss our concepts for distributed cooperative knowledge spaces and spontaneous learning sessions.

Our goal is to facilitate state-of-the-art mobility-supporting network technology, as described in Section 3, and evaluate it in terms of our aims. Based on these networks, we seek to build an application that allows users to cooperate with a minimum of administrative effort.

[2]The *Ad hoc On-Demand Distance Vector (AODV) Routing* is specified in an RFC (Perkins et al., 2003), but is not available as standard protocol stack in current operating systems.

[3]The *link-local* protocol is part of MacOS X, Linux and newer Windows versions

Handheld devices are small enough to support portability, often have enough power to compute the software for these environments, and the wireless network interfaces now frequently included allow connectivity with other devices. To evaluate the special characteristics of PDAs, we developed clients for our *sTeam* client/server environment. While the first prototypes were HTML-based (Eßmann and Hampel, 2003), the newer generation is Java-based, to benefit from *sTeam's* event mechanisms.

While the developed user interfaces meet the needs of PDAs? small displays and pen-based input, they are still classical clients in a client/server infrastructure. Parallel to this, we built the middleware for finding other peers and communicating with them in *MANETs* and structured networks. We describe this middleware in Section 5 and it is the main focus of this paper. Our application is thus designed as a peer-to-peer architecture capable of running on mixed network topologies .

The novel feature of using CSCW environments for mixed network structures is not only the advantage of independence from specific locations but also the loss of guaranteed services within these spontaneous networks. This loss of guarantees poses a number of challenges in terms of adapting concepts from classical CSCW environments. **In peer-to-peer topologies, all resources for needed services must be able to be hosted on all clients**. An important problem are *persistent knowledge areas* of classical *sTeam*, in which learners cooperate by communicating and managing documents. We need a concept to distribute the knowledge space over the mobile devices in such a way that the required knowledge areas are accessible to all users.

Spontaneous networks are by nature highly dynamic. The existence of a node in this network is not guaranteed. **The node containing the desired knowledge area must not be a member of the network or may suddenly disappear from it**. Whole parts of a network may also be separated from each other. In (Feeney et al., 2001), this problem is called network partitioning.

Since users can move around, the availability of a certain knowledge area is not predictable. Given the uncertain availability of knowledge areas on foreign hosts, users may want to take interesting areas with them. To do so, **users need their own instances of interesting knowledge areas to work with in case they are disconnected from their co-workers**. This results in redundant storage of the objects involved. If one user changes the content of an object within a certain area, all other instances become outdated. **Problems may arise if two instances are combined again when two peers reconnect with each other**. This is what (Feeney et al., 2001) calls network merging. If the instances of an area change, this poses the challenge of synchronizing them between the affected nodes.

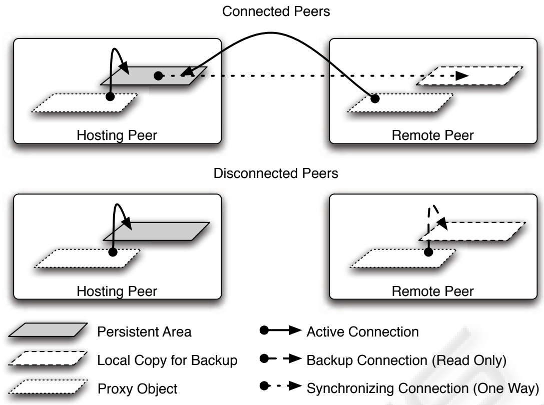

Figure 1: Remote and hosting peer when connected and when disconnected

In order to explore practical solutions to these problems, we began by adopting the pragmatic approach of creating persistent and personal knowledge areas. Our next step was to design temporary knowledge areas and temporary user groups.

4.1 Persistent and Personal Knowledge Areas

In classical client/server environments, all users work on the same instance of a knowledge area. This instance remains persistent on the server until an authorized user destroys it. These persistent knowledge areas can be guaranteed because all users work on the same objects. In ad hoc networks, there is no such guarantee, not even for personal work areas. In *Distributed Knowledge Spaces*, we distinguish between *personal* and *persistent knowledge areas*.

Personal knowledge areas are hosted only on the learners' personal device and are only accessible to them. Unlike personal knowledge areas, *persistent knowledge areas* are areas for cooperative work hosted on precisely one node in the network (*hosting peer*). All connected nodes (*remote peers*) may perform changes within this area through a proxy object pointing to the original area. As shown in Figure 1, nodes are also allowed to create a synchronized local read-only instance in case they are disconnected. When offline from the hosting peer, the remote peer may read-access the local copy.

The problem of local copies becoming outdated poses a challenge that it is hard to meet with technical solutions. Classical synchronization for mobile clients as in *PalmSync* or *CPISync* is designed to synchronize changing datasets between only two or three

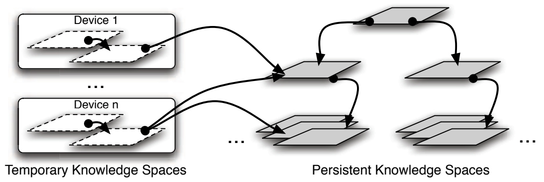

Figure 2: Temporary knowledge areas may reference persistent knowledge areas but not vice versa

hosts in one or the other direction (Trachtenberg et al., 2002). Dynamic synchronization of a complete network's nodes is an even tougher task. Our concept enables this problem to be evaded.

4.2 Temporary Knowledge Areas and Groups

Another difference from classical learning environments is the fact that in mobile scenarios collaborative work is often not planned with the goal of creating persistent knowledge spaces for future work but to use knowledge spaces for spontaneous knowledge construction. Consequently, the developed knowledge space can be destroyed as soon as the session is finished. It may be the case that one or more participants in the session wish to keep some of the results or even the whole area. Since this cannot be decided when the session starts, it must be possible to transfer the results of the session, or even parts of it, to a persistent knowledge space. The same applies to the spontaneously established groups because the decision to hold a subsequent session may have been reached after the group was established.

For this reason, our concept allows temporary groups to be established with an appropriate knowledge area. This mechanism is specially designed for spontaneous face-to-face learning sessions, where all participants meet at a certain location and wish to collaborate. The users only need their connected devices.

An important restriction on temporary knowledge areas is the linking of objects. While in classical *sTeam* systems any object may be referenced by a link, links to temporary knowledge areas are prohibited (see Figure 2. To reference a temporary knowledge area, it must first be transformed into a persistent knowledge area.

Like the area, the temporary group exists as long as at least one user is participating in the session. When the last participant leaves, the group is automatically destroyed. If the hosting device withdraws from the network, the session still exists but the other participants have only a read-only copy, as in the case of persistent knowledge areas. These conventions ensure that only one active instance of the area exists.

This concept is evaluated by implementing prototypes for the cooperation environment, as described in the following section. Evaluation results relating to the concept of *Cooperative Virtual Knowledge Spaces* can be found in (Hampel and Keil-Slawik, 2003).

5 AN ARCHITECTURE FOR MOBILE COLLABORATION

The last two sections discussed the network environment for our application and described collaboration techniques matching such network environments.

The devices the application has to run on may be as heterogeneous as the network. They include any network-connected device, ranging from cell phones and handheld devices to full-featured PCs and servers. This heterogeneous situation yields some important criteria for our application:

Open Architecture The application must be portable to almost any platform/device. Standardized protocols and interfaces must therefore be used. Also, it must scale to the capabilities of the hosting devices.

Automatic Configuration The application must be able to configure to meet the requirements of the actual environments with a minimum of administrative effort. Also, it must reconfigure dynamically for changing environments.

Spontaneous Networking The application must able to establish spontaneous *sTeam* networks within heterogeneous IP-based network environments. This includes the option of sharing resources within the network and connecting to classical *sTeam* servers.

The goal of our development process is to build an open, self-configuring and well-scaling architecture. Our architecture is based on our work on the classical client/server *sTeam* system, which has been shown to work well in common IP-based network infrastructures. Since every resource in *sTeam* is represented by an *steam-object*, which can be both a user in the system and a knowledge area or document, we speak only of objects, regardless of the kind of object we are talking about.

A node in a *sTeam* a peer-to-peer network may have limited and possibly low resources. For this reason, nodes must manage resources very carefully. Typical critical resources on mobile nodes are memory, computing power (CPU), battery life and bandwidth (Meyer, 1995).

One component in our architecture manages the resources of a node. We call this component *Resource Handler*. The *Resource Handler* warns about low resources and is configured with *policies* on how to

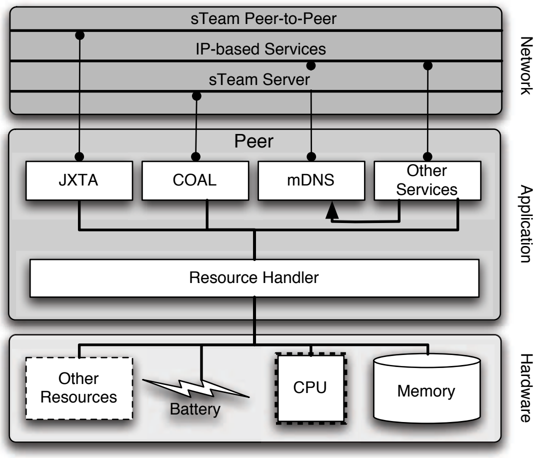

Figure 3: The *Resource Handler* controls the network components the peer may offer to the network according to the available hardware resources

take care of resource conflicts. An example policy is used for peers with low battery power. The *Resource Handler* invokes the migration of shared objects from the low-energy peer to another peer with better battery resources. This allows the drained peer to save battery power by reducing its network activity while other remote peers can still work on the shared objects. Even if the node disappears from the network because of its low battery power, it does not affect the remotely working peers. Similar policies exist for low memory, CPU performance, bandwidth and any other resource monitored by the *Resource Handler*. The *Resource Handler* balances available resources between nodes by communicating with its counterparts in other *sTeam* peers via the network.

Another way to save a node?s resources is to activate/deactivate several services to the network, as shown in Figure 3. Thus, a node on a low-featured device like a mobile phone would merely access other remote resources but offer no services to other nodes. Better equipped devices can even offer conventional services to access a knowledge area (e.g. ftp access). We will discuss this option later on.

To build our application, we opted for the *JXTA* peer-to-peer framework, which implements a overlay peer-to-peer network that allows communication between any network-connected device, from cell phones and handheld devices to personal computers and servers. Figure **??** shows a small *JXTA* network. By using *JXTA* applications are able to communicate on a peer-to-peer basis without knowing the underlying network structure (Gong, 2001).

An important requirement for our application is the ability to connect automatically to its counterparts to form a spontaneous application network. *JXTA* shows this behavior even in a hybrid architecture of structured and ad hoc networks.

While *JXTA* offers services to other *JXTA* nodes, other IP-based applications are unable to access the resources on the *sTeam* peer. To be as cooperative as possible with other parts of the network, our application is able to offer services to classical clients, e.g. web browsers or ftp clients. These additional services can be provided depending on the capabilities of the node. Furthermore, the *sTeam* peer is able to use standard IP-based services for its purposes. This leads to the problem of finding services in ad hoc networks where no fixed namespaces exist and where access to DNS servers cannot be guaranteed.

For publishing and finding such services, we use *Multicast DNS* (*mDNS*). *mDNS* broadcasts information about provided services to the network. This information can be read by any *mDNS*-capable application to determine which IP address and port to connect to for which service. *mDNS* is part of the *Zeroconf* specification carried out by *IETF's Zeroconf Working Group* and is described in (Cheshire and Krochmal, 2003).

The second optional network protocol is the *COAL protocol*. This is the proprietary standard *sTeam* protocol for connecting to classical *sTeam* servers and clients. Thus, the peer can communicate with *sTeam* servers and provide objects to classical *sTeam* clients.

The use of optional services allow peers to scale their provided services to suit their resource capabilities. In addition to the above-mentioned resources, the *Resource Handler* also controls the provided network services according to available bandwidth, computing power, memory, etc. This leads to high scalability. The application can scale both on the *JXTA* part by scaling from *Minimal Peer* up to *Rendezvous* or *Relay Peer* (Gong, 2001) and on the *sTeam* part by scaling from a pure *remote peer*, only working on knowledge areas hosted on other nodes, up to a network server, offering access to its knowledge areas via http, ftp, *COAL*, etc. This concept enables the application to adapt to devices like cell phones, full-featured laptops or even workstations.

As part of our development process, we built a prototype of an *sTeam* peer node using the *JXTA* technology. The prototype is quite a simple peer, which is able to send messages containing XML structures, *steam-objects* or Java objects from one peer to another. This is done without any administrative effort on the part of the users. The peers find each other autonomously without any configuration. This prototype shows that it is feasible to build an *sTeam* peer-to-peer network with the basic required communication mechanisms using the *JXTA* framework.

Parallel to this, we built a Java-based framework for implementing *mDNS*-capable Java applications. It is based on the *Multicast DNS* implementation *jRen-*

dezvous of *Strangeberry Inc.*. Our framework provides a Java interface class, which must be implemented by an application to provide an *mDNS* service. As with the *JXTA* prototype, instances of the chat application are able to find each other and establish connections without any configuration by the users. With our *mDNS* interface and the *JXTA* prototype, we now have the basic components for our *sTeam* peer-to-peer application. We hope to be able to present the results of our ongoing work at the conference.

6 CONCLUSION

The described prototype for collaboration in mobile ad hoc networks constitutes a further step toward network-supported, location-independent collaboration. Our concept enables the users of mobile collaboration environments to cooperate via their mobile devices without the need for any existing network infrastructure, and at the same time allows them to use classical services when available. By doing so, we avoid the trade-off involved in using spontaneous peer-to-peer networks or established client/server solutions.

Both concepts are consolidated in our approach.

Our prototype demonstrated that it is possible to transfer our well-proven concept of *Cooperative Virtual Knowledge Spaces* from common network infrastructures to the innovative hybrid network structures described in Section 3. To do so, we are adapting *Cooperative Virtual Knowledge Spaces* to dynamic network structures. Our goal is to develop a concept for *Distributed Knowledge Spaces*. *Distributed Knowledge Spaces* allow *Knowledge Spaces* to be distributed over multiple dynamic connected nodes in spontaneous and classical networks.

Computer-supported cooperation can thus accompany users through the sort of situations encountered in their daily lives. This is our primary research goal.

REFERENCES

Campell, A. T., Gomez, J., Kim, S., Wan, C.-Y., Turanayi, Z. R., and Valko, A. G. (2002). Comparison of ip micromobility protocols. *IEEE Wireless Communications*, 9(1):72–82.

Cheshire, S., Aboba, B., and Guttman, E. (2003). Dynamic configuration of link-local ipv4 addresses. Http://todo/draft-ietf-zeroconf-ipv4-linklocal-08.txt (june 03 2003), Zeroconf/DNSEXT Working Group (IETF).

Cheshire, S. and Krochmal, M. (2003). Performing dns queries via ip multicast. http:/todo/draft-cheshire-dnsext-multicastdns-02.txt (june 20 2003), Zeroconf/DNSEXT Working Group (IETF).

Dertouzos, M. (1999). The oxygen project. *Scientific American*, 282(3):52–63.

Eßmann, B. and Hampel, T. (2003). Human computer interaction and cooperative learning in mobile environments. In *Human Centred Computing Cognitive, Social and Ergonomic Aspects. Proceedings of the HCI International 2003*, pages 694 – 698.

Feeney, L. M., Ahlgren, B., and Westerlund, A. (2001). Spontaneous networking: An application-oriented approach to ad hoc networking. *IEEE Communications Magazin*, 39(6):176–181.

Gong, L. (2001). Jxta: A network programming environment. *IEEE Internet Computing*, 5:88–95.

Hampel, T. and Keil-Slawik, R. (2003). Experience with teaching and learning in cooperative knowledge areas. In *Proceedings of the Tracks Education, Global Community, Industrial, Practice and Experience,Web Services*, pages 125–132. WWW2003. The Twelth International World Wide Web Conference.

Hanssens, N., Kulkarni, A., Tuchinda, R., and Horton, T. (2002). Building agent-based intelligent workspaces. In *Proceedings of the ABA Conference*, page To Appear.

Johnson, D., Perkins, C. E., and Arkko, J. (2003). Mobility support in ipv6. Internet draft, draft-ietf-mobileip-ipv6-24.txt, june 30, 2003, IETF.

Lauer, M. and Matthes, M. (2002). Elan: An e-learning infrastructure for ad-hoc networks. In *Proceedings of the eighth ACM International Conference on Mobile Computing and Networking*, volume 7, pages 53–55.

Meyer, A. (1995). Pen computing. *SIGCHI Bulletin*, 27(3):46–90.

Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (aodv) routing (rfc 3561) (july 2003). Rfc, IETF Network Working Group.

Perkins, C. E. (1997). *Mobile IP: Design Principles and Practices*. Addison-Wesley.

Perkins, C. E. (1998). Mobile networking in the internet. *Mobile Networks and Applications*, 3(4):319–334.

Perkins, C. E., editor (2001). *Ad Hoc Networking*. Addison Wesley, Boston, USA.

Trachtenberg, A., Starobinski, D., and Agarwal, S. (2002). Fast pda synchronization using characteristic polynomial interpolation. In *Proceedings of the INFOCOM 2002. 21th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1510–1519.

Tseng, Y.-C., Shen, C.-C., and Chen, W.-T. (2003). Integrating mobile ip with ad hoc networks. *IEEE Computer*, pages 48–55.

Wessner, M., Dawabi, P., and Fernández, A. (2003). Supporting face-to-face learning with handheld devices. In *Designing for Change in Networked Learning Environments, Proceedings of the International Conference on Computer Support for Collaborative Learning 2003*, pages 487–491.