AD HOC SERVICE GRID
A SELF-ORGANIZING INFRASTRUCTURE FOR MOBILE COMMERCE

Klaus Herrmann, Kurt Geihs, and Gero Mühl
Berlin University of Technology,
Institute for Telecommunication Systems - iVS,
EN6, Einsteinufer 17, D-10587 Berlin, Germany,
Tel: +49 30 314 25102, Fax: +49 30 314 24573,
Email: {kh, geihs, gmuehl}@ivs.tu-berlin.de

Abstract: The provisioning of location-specific services in medium-sized facilities like shopping malls, hospitals, and trade fares using WLAN access points or cellular phone systems has some drawbacks. It is rather inflexible or results in high running costs. We propose a new self-organizing infrastructure called *Ad hoc Service Grid* (ASG) that is based on a mobile ad hoc network to resolve these issues. In this paper, we define the necessary concepts, structures and components for realizing ASGs. We present our approach to building a Serviceware that enables an AGS to self-organize. We evaluate the effectiveness of our mechanisms for service migration, replication, and lookup based on simulation results.

Key words: Mobile services; Ad hoc Networking; Ad hoc Service Grid; Serviceware.

1. INTRODUCTION

Providing wireless electronic services in public places becomes increasingly important for users and companies. Currently, there are essentially two alternatives for providing such services: IEEE802.11 (WLAN) access points can cover a limited area and let people gain access to the Internet. Cellular phone networks like GSM and UMTS provide ubiquitous access to global resources. However, it turns out that both WLAN and cellular phone networks fail to provide a valid coverage technology for local services at medium-sized locations such as shopping malls, hospitals, and construction sites which span areas of a few thousand square meters. Such locations are too large to be covered efficiently with WLAN. The main reason for this is that, in a standard setup, IEEE802.11 access points need to be wired to some networking infrastructure. Here, the paradox situation occurs that the wiring of several wireless access points costs large amounts of money. This was reported in 2001 by the company lesswire who covered a hall at the CeBIT fare with Bluetooth access points (Kraemer and Schwander, 2003). Other experiments with WLAN also resulted in the costs for wiring exceeding the costs of the wireless equipment by far. Moreover, wired access points limit the flexibility of the network since extending or restructuring the network is cumbersome.

The alternative of using mobile phone networks for service provisioning introduces another problem. The services that are subject to the coverage are in most cases local by nature. For instance, in a shopping mall a shop owner wants to provide a product information service, or a music store offers excerpts of current chart hits for download and lets the customer compose his own CD. Using these services requires communication between two parties at the same physical location (the shopping mall). However, using mobile phone networks implies global communication at a relatively high price and at a comparably low data rate (even with UMTS). Therefore, they seem inappropriate too.

Thus, there appears to be a gap between WLAN coverage and mobile phone network coverage, which cannot be served adequately by either of the two. The solution is a mobile ad hoc network (MANET). A MANET consists of devices that are equipped with a short range radio interface like Bluetooth. Two devices can only connect to each other if they are within each other's transmission range (usually between 10 and 100 meters). The devices may be mobile. Thus, the communication network among them is more or less dynamic. Connections may be setup and torn down frequently and there is no pre-existing infrastructure through which one device may connect to some other device reliably. Many different routing algorithms have been designed which enable two devices to communicate over a multi-hop route if they are not direct neighbors.

In this paper, we propose a new self-organizing service provisioning infrastructure called Ad hoc Service Grid (ASG) based on MANET technology (Giordano, 2002). ASG fills the gap between small-scale WLAN access points and global-scale mobile phone networks and presents a promising approach to the problem of covering medium-scale locations with local services. ASG uses MANET technology to connect individual PC-class devices (called Service Cubes) dispersed over the location. Together, the Service Cubes provide the basic infrastructure and the resources for service provisioning. The communication within this infrastructure is free of charge and at the same time the installation is quick and easy. The ASG also introduces a new way of service provisioning in terms of provider and client roles. In an ASG, these roles are not fixed. Multiple different providers should be able to provide their services to locally present users as indicated in the small shopping mall example.

The remaining paper is organized as follows: Section 2 presents an overview of the research work related to the ASG vision. In Section 3 we discuss the basic communication infrastructure that is based on ad hoc networking between autonomous Service Cubes. The notion of Location-specific Services is introduced in Section 4. Section 5 discusses the mechanisms and functions of a self-organizing Serviceware for ASGs in greater detail. The simulation results given in Section 6 for our service migration and replication mechanism and for our service lookup and discovery protocol prove the effectiveness of our approach. Finally, Section 7 concludes the paper and presents our research agenda.

2. RELATED WORK

MANETs have originated from a military background. Recently, they have drawn much attention also in the civilian domain. Most people involved with MANETs assume that they consist of a group of mobile users that move relative to each other while trying to communicate. The question of service provisioning in MANETs has been studied especially with respect to two issues: Some research projects deal with the problem of optimally positioning a service on one of the devices in order to cover the largest possible number of the users (Li, 2001; Wang, 2001; Wang and Li, 2002, Lau et al., 2001). However, it remains questionable if this is a relevant problem at all since the authors only give a vague indication of the nature of the services and the scenarios they relate to. Other projects deal with the problem of designing service lookup protocols in MANETs (Cheng, 2002; Chakraborty et al., 2002). Again, the motivation for running a service on a resource-limited and battery-powered device of a mobile user remains unclear. Infostations (Goodman et al, 1997) are another approach to providing services in ad hoc networks. However, Infostations are not necessarily connected to each other. Instead, they represent islands covered by WLAN that may exchange information with mobile users passing by. These islands are connected in a logical sense by the users moving between them and by the information that may be exchanged indirectly via several users and Infostations. An area that has received some attention lately are Peer-to-Peer (P2P) services in MANETs (Datta, 2003; Klemm et al., 2003). The common ground that both P2P and MANETs build on is the lack of a hierarchy and predefined roles. There are no dedicated servers and clients. Instead, every node in the network acts as a server and a client at the same time. While this is an appealing approach for extending common P2P systems into the wireless world, such systems do not seem to have any potential for service provisioning on a commercial level, because the resources available through mobile devices are very scarce and the expected dynamics in a network purely made of mobiles is too high.

To the best of our knowledge, applying the MANET technology in a more conservative way to supply an alternative service provisioning infrastructure has not been subject to research yet.

3. COMMUNICATION INFRASTRUCTURE

The basic communication infrastructure of Ad hoc Service Grid consists of so-called *Service Cubes*. Service Cubes are devices that offer computational resources comparable to standard personal computers. They need a power supply and can connect to each other over a short-range wireless network interface (e.g. Bluetooth). At a facility that is to be covered with wireless services, a number of Service Cubes are dispersed and automatically set up an ad hoc network (Figure 1). This network provides two things:

1. An access infrastructure that connects all the Service Cubes, and
2. A distributed pool of resources (the Service Cubes themselves) to host value-added services.

As opposed to existing wireless service provisioning infrastructures, in an Ad hoc Service Grid there is no real distinction between the access infrastructure and the system running the services. The whole infrastructure is modular with the Service Cubes providing networking *and* computing resources. The ASG follows a philosophy we call *Drop-and-Deploy*: A new Service Cube can be added to the network by simply putting it in an appropriate position (within communication range of at least one other Service Cube) and switching it on. The Serviceware that we envision recognizes the new resource and integrates it into the ASG. The aim is to reduce the overhead associated with planning, installation, and maintenance. The MANET technology is the key to this Drop-and-Deploy feature.

Users access the services via their own handheld device. A network connection to the nearest Service Cube is established automatically, and the user is provided with a list of available services. If the Service Cube network is dense enough, the user is always connected to at least one Service Cube, and thus, able to access the ASG's services while he moves through the location.

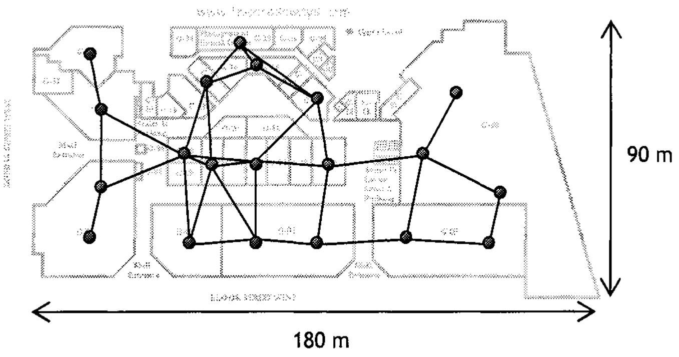

Figure 1. Example of an ASG Network in a shopping mall.

4. LOCATION-SPECIFIC SERVICES

The term *Location-based Service* is used to describe services that capture the user's current position and present content pertaining to this position. These services do not enable the user to interact with his current location in any way. The ASG, in contrast, offers *location-specific services*. Such a service allows the user to interact with his current environment. It is provided *by* the location instead of *on behalf of* the location (as with location-based services). Thus, a typical location-specific service runs locally (e.g. in the shopping mall) and can be used by users in close proximity. It is provided by a local party. For example, in a shopping mall, the operator could provide a *navigation service* that helps visitors in finding the right shop. Shop owners might offer *product information services*. A music store might offer music clips for free and let customers compile and purchase their own CD.

The ASG presents a model that is quite different from classical service provisioning architectures. Most notably, the increased flexibility implies that the roles of the participants are somewhat blurred. To clarify things, we introduce three basic roles:

- The *operator* is an institution (a single person or a group of people) that is responsible for administering the location or facility that is subject to service coverage.
- *Providers* are parties that are locally present at the facility. They may potentially provide electronic services to others (e.g. individual shop owners in a mall). Providers are divers and only loosely coupled.
- *Customers* are temporarily visiting the facility and are potential users of provided services. Customers are also called *clients* or *users* in this paper.

Basically, all three groups may offer and use location-specific services. Albeit, the ASG infrastructure implicitly assumes that customers will be at the receiving end of service provisioning in most cases.

5. A SELF-ORGANIZING SERVICEWARE

Like existing Grid technology, an Ad hoc Service Grid provides the resources and the control structures necessary for accomplishing diverse tasks in a distributed, collaborative fashion. However, classical Grid tasks consist of calculations with a limited life span that need to be processed as quickly as possible and produce a (partial) result that is returned to the client. In contrast to this, in an Ad hoc Service Grid the tasks are services which typically run over a longer, indefinite period of time. Moreover, these services normally have extensive interactions with numerous users.

The ASG changes dynamically when new Service Cubes are added or existing ones are moved or removed. Thus, maintaining a global view on the overall system state is difficult. Additionally, dynamically changing usage patterns (dictated by mobile clients) necessitate an ongoing adaptation process in order to optimize the resource usage. Managing the ASG manually is therefore inappropriate. Allocating resources statically, i.e. installing services on fixed nodes will result in suboptimal resource usage since a proper initial placement of services will most likely become suboptimal when usage patterns change. The service might end up running on a node that is far away from the bulk of interested clients resulting in excessive overhead as messages are routed through the entire network.

Because of the Service Cube dynamics, the lack of an overall system view, and the changing usage patterns, the ASG has to organize many aspects of the service provisioning on its own, without manual intervention. It has to be *self-organizing*.

We aim to encapsulate the bulk of self-organization tasks inside a software layer that we call *Serviceware*. It is based on the middleware MESH*Mdl* (Herrmann, 2003) which is specifically tailored to self-organizing applications in MANETs. It employs mobile agents (Picco, 2001) and asynchronous communication based on Tuple Space technology (Gelernter, 1985; Ahuja, 1986). Every client device and each Service Cube runs a MESH*Mdl* Engine that is able to execute application components (mobile agents) and provides asynchronous communication via a MESH*Mdl* *Event Space* (special enhanced realization of the Tuple Space paradigm in MESH*Mdl*). Data items in this Space (also called *Entries*) can also be sent to remote devices via a multi-hop routing infrastructure. ASG services are implemented as mobile agents, and thus, are able to move between Service

Cubes if necessary. The three basic mechanisms used for dynamic service distribution are:

- *Migration*: Moving a service from one Service Cube to another. In most cases this is done to reduce the distance between clients and a service replica.
- *Replication*: Creating a replica of a service and migrating the original service instance and its new replica to different Service Cubes. This is necessary if an existing replica in unable to handle the local request load and if traffic patterns imply that high volumes of requests come in from different directions.
- *Recombination*: Removing a dispensable service replica and possibly merging its state into another replica. Due to dynamic changes, two or more service replicas in the same vicinity may end up serving a small number of clients. This excessive service capacity causes suboptimal resource usage. Thus, these service replicas may recombine into one single instance.

In the following subsections, we explain some of the core functions provided by the Serviceware in more details.

5.1 Network Clustering

To enforce a structure on the amorphous ad hoc Service Cube network and to implement basic networking functions, we adapted the clustering algorithm published by Basagni (1999). It partitions the Service Cube network into clusters, each having a cluster head node. As a consequence, every *ordinary node* has exactly one *cluster head* and the cluster heads are distributed evenly over the whole network. Cluster heads provide core Serviceware functions while ordinary nodes carry value-added services. Thus, the cluster assigns basic roles to the nodes in terms of service provisioning and introduces a separation of concerns.

Based on the clustered structure, we implemented a routing algorithm that is based on *Dynamic Destination-sequenced Distance Vector Routing* (DSDV) (Perkins and Bhagwat, 1994). Since the Service Cube network is not very dynamic, this simple algorithm is adequate. Clusters define subnetworks with the cluster head serving as a router. Thus, only routes between the cluster heads need to be set up. Each node is assigned a node address consisting of the node ID and its cluster head ID. This overlay network is used to route *Entries*, messages used by MESH*Mdl* for inter-agent communication (Herrmann, 2003), between different Service Cubes.

While classical distributed systems hide the network structure and issues like routing from the applications, the ASG Serviceware takes a different approach. To render the Serviceware self-organizing, structural information about the network and the message flow is vital. Indeed, the flow of requests from clients to services is an important stimulus for adaptation that enables, for example, the adequate placement of services replicas inside the ASG. Thus, we implement routing at the Serviceware layer to make this information accessible. The clustered structure must be perceivable for the Serviceware since it decides about the higher-level functions provided by a Service Cube.

5.2 Service Replication and Migration

A core function of the ASG Serviceware is to accept the installation of new services at an arbitrary Service Cube and move it to a Location that provides the necessary resources for its execution. The overall goal is to balance the computational load on the set of available Service Cubes and to minimize the networking load on the ad hoc communication network. Both can be achieved by dynamically allocating Service Cubes to services (i.e. by making services mobile) and by replicating services. Figure 2 depicts the combination of service migration and replication. Please note that the squares do not necessarily stand for individual clients, but rather for client request hot spots (i.e. Service Cubes that receive a high number of requests).

We have developed an algorithm for *service migration and replication* that lets a service migrate gradually towards the direction of the highest client request traffic. If the service receives a high number of requests via different network connections (see Figure 2b), it replicates and the two replicas move towards the directions of highest request traffic.

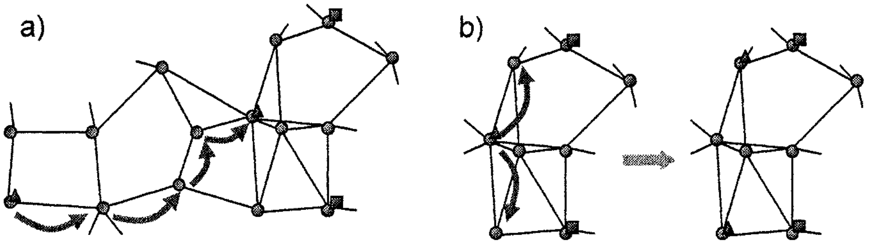

Figure 2. a) Moving services (depicted as triangles) towards the centers of client requests (indicated by squares) and b) service replication.

The algorithm executed for a service on node s_0 is defined as follows:

Let $S = \{s_1,\ldots,s_n\}$ be the set of neighbor nodes of s_0 via which s_0 has received requests for the service. Let S be in descending order with respect to the number of requests $m(s_i)$ received from s_i. Let T_A be the *adaptation threshold* and let T_M denote the *migration threshold.* T_A is measured in *requests per time unit* and indicates the volume of incoming requests that will trigger an adaptation. An adaptation can either be a migration (the service moves to a neighbor node) or a replication (the service clones itself and the two replicas are moved to different neighbor nodes). T_M is used to decide which of the two actions is to be invoked. Finally, let R be the set of currently active service replicas and let L_R be the maximally allowed number of replicas. The algorithm in Figure 3 is used to achieve the autonomous distribution of services.

```
1: if $|S| > 0$ and $m(s_1) \geq T_A$
2:     if $\left(|S| = 1 \text{ or } \frac{m(s_1)}{m(s_2)} > T_M\right)$ and $m(s_2) < T_A$
3:         migrateTo($s_1$);
4:     elsif $|R| < L_R$
5:         replicateTo($s_1, s_2$);
6:     endif
7: endif
```

Figure 3. Autonomous service distribution algorithm.

The algorithm is periodically executed in parallel by all active service replicas. In line 1 the algorithm checks if there have been any requests at all ($|S| > 0$) and if the number of requests received from the top sender has reached the adaptation threshold ($m(s_1) \geq T_A$). The *if* statement in lines 2 to 6 decides whether a migration or a replication should be executed. If we have only one sending neighbor ($|S| = 1$) or the ratio of the number of messages sent by s_1 over the number of messages sent by s_2 exceeds the migration threshold $\left(\frac{m(s_1)}{m(s_2)} > T_M\right)$ we will migrate to s_1, except, if the number of messages received from second highest sender s_2 also exceeds the adaptation threshold. The later case covers the situation that the service receives a high volume of requests via two different neighbors which should results in a replication. Note that if the ratio of received messages exceeds the replication threshold, s_1 is forwarding considerably more requests than s_2. If there is no indication for a migration, then the *elsif* statement is executed. It checks whether the maximum number of replicas has already been reached. If this is not the case, then a replication is triggered. The information about the number of currently active service replicas is maintained by the distributed lookup service that is explained in the next section. This information yields an approximation of the replica count. The results of simulating this algorithm in a network of 100 Service Cubes is presented in Section 6.

5.3 Service Lookup and Discovery

In the ASG, established service lookup protocols like that of Jini (Sun Microsystems, 2001) fail because services are mobile. We have developed a lookup service that can handle service mobility. The basic idea is to position a lookup service at every cluster head node in the network. Thus, at every Service Cube, clients and service replicas always have a lookup service within one-hop distance. A moving service can always easily notify its local lookup service about its new location. However, propagating such a change to the other lookup services without causing a broadcast storm on the network is not as easy. We have developed a *reply-driven signaling protocol* that minimizes the use of update broadcasts and uses a combination of piggybacking and message snooping in order to reduce the message overhead. Whenever a new service replica is started, it registers with its nearest lookup service. This service broadcasts the existence of the new service to all lookup services (running on cluster heads). If a registered service replica moves, client requests arriving at its old location are redirected locally to its new location by the Serviceware. Information about this redirect is stored in the request. The receiving service copies this meta information into its reply and sends it back to the client. As the reply is routed back, the nodes on this route *snoop* the redirect information stored in the reply and update their own lookup services with the new location of the service. Thus, the information about the service migration is propagated along the reply paths to all the clients and eventually reaches the clients themselves. The efficiency and effectiveness of this protocol is investigated in Section 6.

Initial service discovery is also done via the distributed lookup service. A nearby client can request a list of service instances that are running close to its location.

6. SIMULATION RESULTS

The migration and replication (MR) algorithm and the distributed lookup service have been implemented in a simulation environment to evaluate their effectiveness. Figure 4 depicts the reduction in the number of overall messages being transmitted in the network as the MR algorithm is executed. The two graphs in this figure represent two experiments with different service lookup strategies that will be explained below. In the context of service migration and replication, the striking property of both graphs is the downward trend and the fact that the overall network traffic is reduced by approximately 50% when the algorithm is applied.

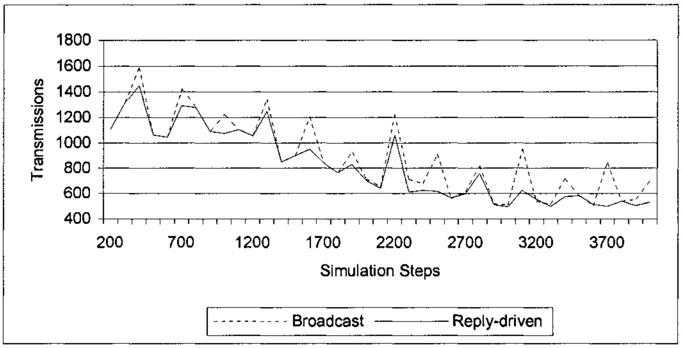

Figure 4. Overall transmitted messages for different lookup signaling strategies.

While the service is being replicated, the load is distributed over the individual service replicas. This is shown in Figure 5: It takes the service about 1300 steps to move to the *center of the network* (in terms of client requests). Then it starts replicating (replications indicated by vertical lines) and the replicas attract the requests that originate nearest to them. Consequently, the load is shared among them and effectively balanced within the network. Thus, the load on the individual services reduces.

The effect of the lookup service signaling protocol on the overall number of messages transmitted is shown in Figure 4. The Figure compares it with a broadcast protocol that simply broadcasts updates to all lookup services. It can be seen that the overall reduction in transmitted messages is still achieved. Thus, the new lookup signaling protocol proves to be effective and clients still find the services nearest to them. At the same time, the broadcast bursts caused by the simple protocol are avoided.

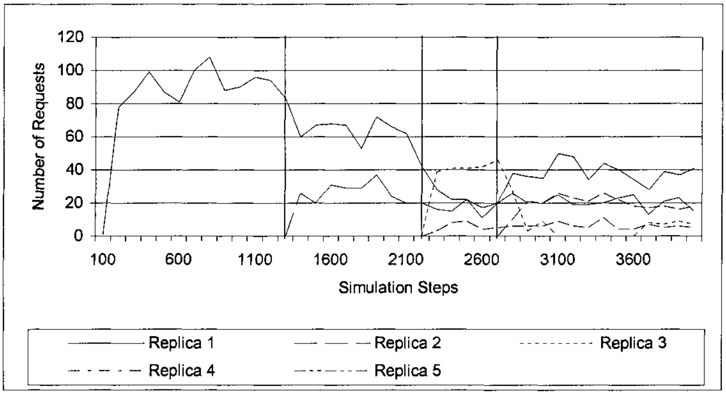

Figure 5. Reduction in per-service load as services move towards clients and replicate.

The reply-driven lookup signaling protocol reduces of the number of transmitted messages at the cost of some lookup services not receiving the correct information directly. Figure 6 depicts the fraction of lookup services that hold the correct information over time. While the broadcast protocol achieves 100% correctness shortly after every service migration, the reply-driven protocol slowly converges towards full correctness as the number of service replicas, and thus replies, increases. The services temporarily holding incorrect information have no clients in their local vicinity. If a new client arrives in such a location, it sends its requests to a wrong destination. However, the forwarding protocol will redirect them to the new location of the service and upon receiving a reply, the information in the lookup service will be corrected. This form of *lazy update* suits the nature of the ASG very well since trying to maintain overall correctness is costly and unnecessary.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a new infrastructure for providing facility-specific services to mobile users. We argue that Ad hoc Service Grids are able to fill the gap that is observable between small-scale WLAN access point systems and large-scale mobile phone systems. We presented our vision of an Ad hoc Service Grid and pointed out the possible structures and concepts necessary for the realization of the infrastructure. We have evaluated our current work on *service migration and replication* as well as a *distributed lookup and discovery service* for the ASG Serviceware. The simulation results demonstrate that decentralized control mechanisms can be employed for the self-organization of ASG services. These mechanisms are based on the flow of user requests through the network and indicate that this stimulus can be used to trigger self-structuring behavior in the ASG.

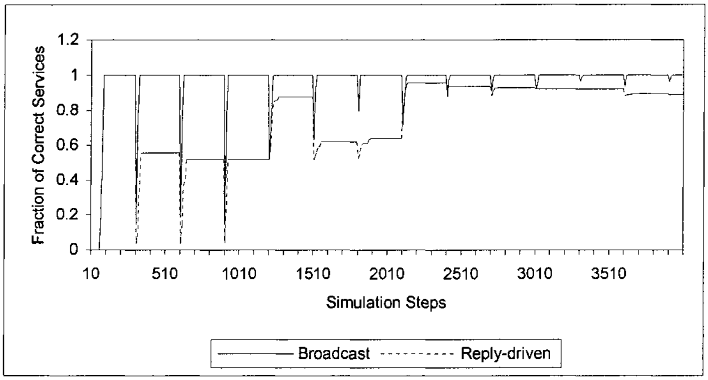

Figure 6. Fraction of lookup services holding correct information as a function of time.

Our current research efforts are directed towards refining the presented mechanisms. Moreover, we are developing concepts that allow stateful services to maintain data consistency among their replicas despite their mobility.

Realizing the Ad hoc Service Grid vision takes efforts from other fields. Is has to be pointed out that especially the basic ad hoc networking support provided by current standards is not sufficient. Furthermore, indoor positioning is an issue.

REFERENCES

Ahuja, S., Carriero, N., and Gelertner, D., 1986, Linda and friends, *IEEE Computer*, pp. 26–32.

Li, B, 2001, QoS-aware Adaptive Services in Mobile Ad-hoc Networks, In: *Proceedings of the Ninth IEEE International Workshop on Quality of Service (IWQoS 01)*, also Lecture Notes in Computer Science, Springer-Verlag, Vol. 2092, pp. 251-268, Karlsruhe, Germany.

Basagni, S., 1999, Distributed Clustering for Ad Hoc Networks, In: *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN).*

Chakraborty, D., Joshi, A., Yesha, Y., Finin, T, 2002, GSD: A Novel Group-based Service Discovery Protocol for MANET, In: *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN'02).*

Cheng, L, 2002, Service advertisement and discovery in mobile ad hoc networks, In: *Proceedings of the Workshop on Ad hoc Communications and Collaboration in*

Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work, New Orleans, Louisiana, USA, November 16-20, 2002.

Datta, A, 2003, MobiGrid: P2P Overlay and MANET Rendezvous - A Data Management Perspective, In: *Proceedings of CAiSE 2003 Doctoral Symposium.*

Gelernter, D., 1985, Generative Communication in Linda, *ACM Transactions on Programming Languages and Systems*, 7(1):80--112, Jan. 1985.

Giordano, S., 2002, Mobile ad hoc networks, In *Handbook of Wireless Networks and Mobile Computing*, ed. I. Stojmenovic. John Wiley & Sons. 2002.

Goodman, D.J., Borras, J., Mandayam, N.B., and Yates, R.D., 1997, INFOSTATIONS : A New System Model for Data and Messaging Services, In: *Proceedings of IEEE VTC'97*, vol. 2, pp. 969-973, Phoenix, AZ. May 1997

Herrmann, K., 2003, MESHMdl – A Middleware for Self-Organization in Ad hoc Networks, In: *Proceedings of the 1st International Workshop on Mobile Distributed Computing (MDC'03)* May 19, 2003. Providence, Rhode Island USA.

Klemm, A., Lindemann, C., and Waldhorst, O., 2003, A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks, In: *Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, FL, October 2003

Kraemer, R., and Schwander, P., 2003, Bluetooth based wireless Internet applications for indoor hot spots: experience of a successful experiment during CeBIT 2001, *Computer Networks*, vol. 41, no. 3, 2003

Lau, W. H. O., Kumar, M., and Venkatesh, S., 2001, A Cache-based Mobility-aware Scheme for Real-Time Continuous Media Delivery in Wireless Networks, In: *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo (ICME2001)*, August 22-25, 2001, Waseda University Tokyo, Japan

Perkins, C. E., and Bhagwat, P., 1994, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, In: *Proceedings of SIGCOMM 94*, pages 234–244, August 1994.

Picco, G. P, 2001, Mobile Agents: An Introduction, *Journal of Microprocessors and Microsystems*, vol. 25, no. 2, pp. 65-74, April 2001.

Sun Microsystems, 2001, Jini Architecture Specification. Version 1.2. December 2001

Wang, K. H., 2001, Adaptive Service Provisionings in Partitionable Wireless Mobile Ad-Hoc Networks, M.S. thesis, University of Toronto, Department of Electrical and Computer Engineering, October 2001.

Wang, K. H., Li, B, 2002, Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks, In: *Proceedings of IEEE INFOCOM 2002*, Vol. 2, pp. 1089-1098, New York City, New York, June 23-27, 2002.