Swiss Federal Institute of Technology Zurich
Computer Engineering and Networks Laboratory

SERVICE PROVISIONING IN AD-HOC NETWORKS

Report on Research Topic Proposal

Authors: Károly Farkas
Bernhard Plattner

December 2003

This document was created within the framework of the joint project 'Active Networking for Mobile Networks' of DoCoMo Communications Laboratories Europe GmbH and ETH Zurich in 2003.

Zurich, 14. December, 2003.

1. Introduction

This research proposal intends to be the continuation of the previous successful collaboration between DoCoMo Eurolabs and ETH Zürich which came out with a prototype mobile application (context-aware handover using active networking technology) and a joint conference paper (at IWAN 2003).

During the exploration phase of this proposal we looked over several research areas for revealing interesting open issues. This exploration comprehended the areas of Active Networks, Service Deployment and Management, Next Generation Services on Today's Infrastructures, Self-organized Networks, Autonomic and Proactive Computing, and Context-aware Service Provisioning. Finally we decided to look into the longer-term future and tried to find an exciting problem which has relation to several research areas making possible to exploit our existing experiences, as well. Thus, we selected the problem of distributed, device independent, software-based service provisioning in the environment of ubiquitous wireless devices which are communicating spontaneously forming self-organized mobile ad-hoc networks.

In the remaining part of this report first we introduce and elaborate on the revealed problem, then we briefly survey the state of the art of the related research areas, and finally we outline our approach to solve the problem.

2. Problem Statement

2.1 Constraints of Ad-hoc Environment

Service provisioning in ad-hoc environment requires special attention because generally the traditional techniques developed for permanent infrastructures cannot be applied due to the constraints of ad-hoc mobility. These constraints are:

- *Lack of permanent infrastructure*: in an ad-hoc network there is usually no infrastructure support so the system needs to be completely self-organized.
- *Lack of central management*: due to the dynamic and mobile nature of the devices (can join and left the ad-hoc network any time) permanently available nodes, which could carry out central management tasks, cannot be assumed.
- *High level of heterogeneity*: the mobile devices forming the ad-hoc network are usually heterogeneous ones with different capabilities but they should communicate and work together.
- *Devices with limited resources*: generally these devices are small and have limited capacity but presumably this constraint will be mitigated in the future with the technical improvements.

2.2 Application Scenarios

Services can differ from each other. Several sets of services can be distinguished from the simple, centralized, device oriented service (e.g., network printer) to the complex, distributed, software-based device independent one (e.g., real-time games). These services have different requirements which can incur more sophisticated procedures to deploy and manage them than just the relatively simple resource discovery. Let's consider the following application scenarios:

a) *On-line and distributed group games in a public place to kill waiting time (e.g., waiting for a flight in an airport):* the mobile device joining an ad-hoc network can appear on a virtual play-field of a game (e.g., an RPG - Role Play Game) and the user can join the ongoing game session. If the exfoliated ad-hoc network has connection to the Internet, other remote users can also join the game. This vision can be extended further by using virtual reality equipments (e.g., gloves, glasses, sound systems) on the players such as intelligent sensors attached to the ad-hoc node.

b) *Automatic time organizer*: let's suppose that on an international conference there are a lot of participants. Some of them are working in a common project and they want to organize a project meeting during the conference. Every present project member has a PDA which is able to run the automatic meeting organizer service. Moreover, some project members arrive just later to the conference but their calendars can be reached via the Internet thus they can also participate in the meeting time negotiation.

2.3 Problem Elaboration

To identify the arising problems the scenarios should be investigated more closely. Since these scenarios can reveal similar problems, this time let's consider and elaborate on the game scenario.

The foremost obvious question is in this scenario **how the game service can be specified and described?** The service specification, which describes the functions and connections of service components, is required for service registration[1] and lookup[2] when a node wants to announce and provide a service and when a new node joining the ad-hoc network tries to discover the available services in the network. Moreover, it is used also in service deployment when the selected service has to be installed on a given node.

The next question is **where the game service comes from?** Since this game service is a device independent, distributed software service, at least one participant device has to be in possession of the game software in advance, even before the formation of the ad-hoc network. (Another solution can be a fix service node placed in the vicinity of the evolving ad-hoc network serving as code and service servers but this can restrict the full flexibility of the self-organized game network.) But **how can this 'first' device obtain the game software?** The simplest way is if the owner of the device uploads it to the device beforehand and it is distributed from this device to the nodes participating in the game. However, software distribution can arise also legal issues concerning the game software license rights. If it's a freeware software it can be distributed without restrictions. But, if it's a commercial one **how the licensing can be handled in case of the software deployment?**

At this point the notion of the traditional service model should be rethought, as well. Traditionally, the service provider offers the service to its clients and they are using and presumably paying for it. But in this game service **who is the service provider?** The node which owns the first instance of

[1] We use the terms 'service registration', 'service announcement' and 'service advertisement' in the same meaning in this document

[2] We use the terms 'service lookup' and 'service discovery' in the same meaning in this document

the game software? And touching upon the billing and accounting issues too, **who can charge for the game service in which form** and **who has to pay for it?** It can be imagined that the user, who offers it in the ad-hoc network, doesn't intend to make the distribution of the game software possible for free of charge if it is commercial product and the user had to purchase it. This issue touches upon also the licensing problem. Moreover, how the other users should behave who just obtained the game software in an ad-hoc network, and next time they will provide the same game service in another ad-hoc network? And how the intermediate nodes can be handled who just forward traffic providing the connectivity between the game players? Another interesting question is how the accounting/billing can be realized if the charging is carried out between individuals? Since every node may offer software service it can be the case that every connection between the individual peers should be kept track, which introduces a lot of storing and processing overhead if it's possible at all, and extra distributed procedures are required for making possible secure payments, too.

Returning back to the scenario the next step is that the 'game-aware' device tries to explore some partners or join an already evolved ad-hoc network. After the communication is established it has to announce (register) the game service. On the other hand, if a visitor node (a potential game player) arrives it has to be informed of the game service or it has to be capable to look it up. The questions arising here: **how the service announcement (registration) and lookup (discovery) can be carried out?** The problem is that ad-hoc networks cannot provide a permanent, central directory to where the offered services can be inserted and from where the available services can be read out. The solution can be to replicate this directory on every node connected to the ad-hoc network (which introduces synchronization and propagation overhead to maintain a consistent database on all of the nodes but makes the service lookup very efficient) or to establish a so-called virtual backbone in the ad-hoc network selecting some nodes which store a copy of the service directory. This latter solution introduces less administrative overhead (however, it requires an extra procedure to select and maintain the set of virtual backbone nodes) but makes the service lookup less efficient.

When every node knows about the game service, they can decide whether want to participate in the game or not. If yes, the next step is requesting, downloading, installing, configuring the game software and starting the game. Every node has to accomplish these procedures separately but in a synchronized manner to get a consistent state by when the game starts. This addresses the question **how the game service can be deployed?** Presumably a general service deployment and management scheme should be used to facilitate this procedure, which is capable to run in distributed mode and to synchronize between the player nodes. An interesting issue is also to investigate **how this scheme gets to the ad-hoc network?** Maybe every device must contain it in advance in which case updating the scheme may introduce difficulties and may lead a multi-version environment (which is not a problem by all means).

When the game started the service management issues come to the front. The first problem is here **how the communication among the players can be aided during the game session?** A distributed group game presumably requires a huge amount of multicast-like communication which introduces the deployment and use of multicast routing protocols in the ad-hoc network. But whether this protocol should be handled as part of the game service or independently? If the game has special requirements toward this protocol maybe the former solution is better otherwise the latter (here still the problem is how this protocol gets to the ad-hoc network?). Moreover, **how the service reconfiguration and termination can be accomplished?** The service management scheme should support the leaving and joining the ongoing game session, finishing the ongoing game session and maybe restarting a new one with new player nodes. This requires negotiation and synchronization procedures by the management scheme even during the game session. Furthermore, a related problem is the user identification in the game service: **how the service users can be identified?** Presumably the game service itself has to handle its actual players' identification during the game session (e.g., if private messages have to be posted), but if a visitor node appears as a new player candidate the management scheme has to give support to its admittance or rejection (and the appearing node has to be identified somehow).

If the ad-hoc game network has Internet connection, the participation in the game via the Internet can also be possible. But presumably the service deployment and management scheme has to be modified to give support to this access. Moreover, the existence of Internet connection can make it possible to give alternative answers/solutions for the abovementioned questions. For instance, in this case the game software download or the service management software update can be carried out via the Internet, or different multicast routing protocol may be required to support the seamless participation of Internet nodes in the game session. Thus, the problem is addressed here **how the Internet connection can be handled in the game service and how the service management scheme should be modified to support this?**

And finally, concerning the extended version of the game scenario, the following question can arise that **how the context information of the game service user** (e.g., the virtual reality equipments forming a mobile group attached to the ad-hoc node) **can be collected and used in the service deployment/management and during the service session?** For example, if the player is possession of such equipments he can setup an extended version of the game giving more pleasure for himself (e.g., 3-dimensional virtual space instead of 2-dimensional screen graphic). This presumably requires the use of a context management framework, which is capable to collect the context information of the service user and to store it in a standard format, and its integration into the service deployment and management scheme.

2.4 Problem Abstraction

Summing up the previous elaboration the following problems can be abstracted:

Technical issues:

1. How can the different ad-hoc services be grouped and specified/described using some general service description methodology? How can this description be used in service announcement/discovery/deployment/management?

2. How can a service be advertised/discovered/deployed/managed in the mobile ad-hoc environment without the support of permanent/central infrastructure?

3. How can the communication (which can be real-time, delay sensitive incurring huge overheads) among the service users be realized/sustained/aided during the service session?

4. How can the identification and lookup of the service users be solved in the ad-hoc environment?

5. How can the service announcement/discovery/deployment/management be extended when the ad-hoc network has connection to the Internet?

6. How can the service user's context information be collected and used in the service deployment/management and during the service session?

Legal issues:

1. How can the software licensing be handled in case of a distributed software service in an ad-hoc environment?

Economical issues:

2. Who can charge whom and for what in an ad-hoc service?

3. How can the traditional billing/accounting models be adopted or new ones be developed for the ad-hoc environment?

3. Related Work

In the previous section we identified several related issues (technical, legal, economical) to the basic problem, namely the provisioning of distributed, device independent, software-based service in ad-hoc networks. In this research we plan to focus only on a subset of the technical issues (see the next section) and leave the others for future investigation. Thus, in this section we try to briefly survey the state of the art of areas which we will touch in our work.

Service provisioning in ad-hoc networks is a complex problem and it concerns several, mainly distinct research fields. **Fig. 1** depicts this relation.

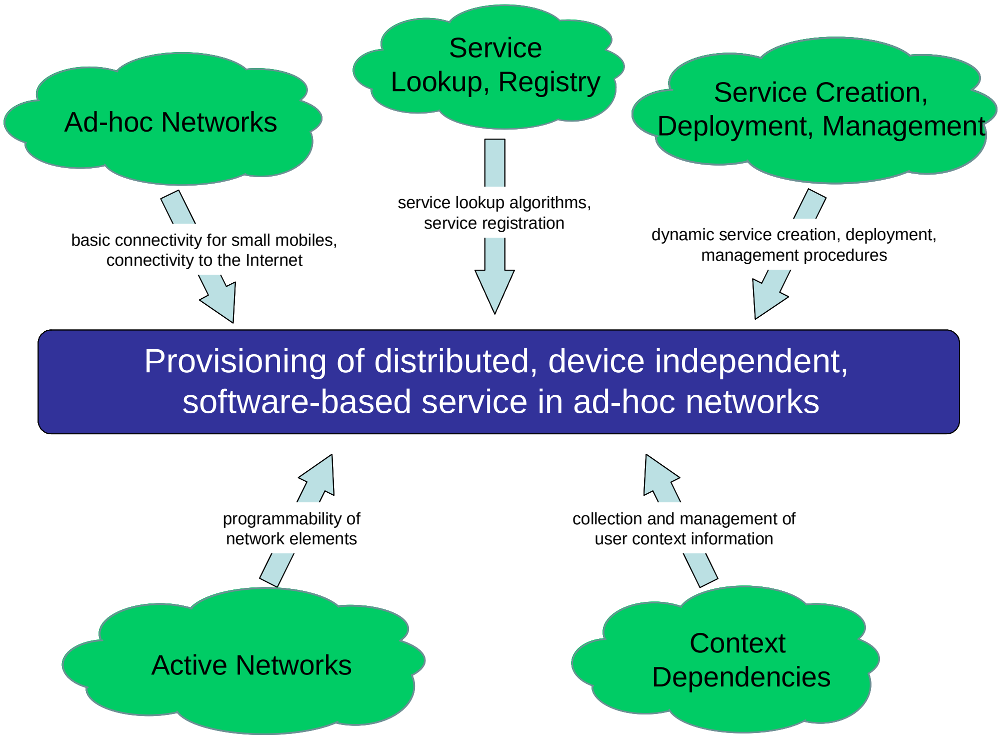
Service Lookup, Registry
Ad-hoc Networks
Service Creation, Deployment, Management
basic connectivity for small mobiles, connectivity to the Internet
service lookup algorithms, service registration
dynamic service creation, deployment, management procedures
Provisioning of distributed, device independent, software-based service in ad-hoc networks
programmability of network elements
collection and management of user context information
Active Networks
Context Dependencies

Figure 1: Service Provisioning in Ad-hoc Networks and the Related Areas

3.1 Ad-hoc Networks

Ad-hoc networking seems to be the vision of future's communications when networks of small, heterogeneous and wireless devices are coming, going and communicating spontaneously. All nodes can move, be connected in an arbitrary manner, and act as routers so ad-hoc networking provides basic connectivity for small mobiles and in some cases connectivity to the Internet. The main problem in this field is routing and addressing. During the last couple of years numerous protocols have been proposed (e.g., AODV, DSR, etc.) for this problem, but still no consensus toward using any of them as a standard (and it seems that it's not possible to develop a general protocol which performs well in every scenario).

3.2 Service Lookup, Registry

The area of service lookup/registry provides algorithms and procedures for discovering and registering services and resources in traditional networks. The problem here is similar to develop a naming and name resolution system. Several proposals have been developed to standardize these algorithms/procedures (e.g., Service Location Protocol (SLP) [1] of IETF, Sun's Jini [2], Microsoft's Universal Plug and Play (UPnP) [3], IBM's Salutation [4], Bluetooth's Service Discovery Protocol (SDP) [5]) but these proposals mainly target infrastructure-aware networks. Moreover, almost all of these protocols include the client-server paradigm and mainly focus on resources/services provided by devices (e.g., printers, fax-machines) which restricts their applications in the envisioned future ad-hoc networks when an entirely new set of possible device independent and distributed services (such as games, context-aware services, etc.) can be introduced. However, recently some new proposals have been appeared directly targeting infrastructure-less networks (e.g., the directoryless Konark system [10] or the directory architecture based on the formation of virtual backbone [11]), which can be good candidates for service lookup/registry in ad-hoc networks[3].

3.3 Service Creation, Deployment, Management

Several research projects have recently focused on dynamic service creation, deployment and management in networks (e.g., Chameleon [6], active pipes [8, 9], hierarchical service deployment [12]), but at the moment they are in different research/development phases and there are still a lot of open issues (e.g., network-wide service deployment, fault tolerant service deployment, on-the-fly service reconfiguration, support for also service management). Moreover, these research works mainly target infrastructure-aware networks and up to now, to the best of our knowledge, there has been no proposal in this area dealing with self-organized networks. Since this field is at the moment in flux, it can provide room for additional research and can be a good candidate as the core of our further research.

3.4 Active Networks

The area of Active Networks provides programmability and on-the-fly reconfiguration of network elements (e.g., PromethOS [7]). The architecture and platform related issues are more or less well explored and don't keep a lot of room for further research in store besides the application of the results in different scenarios/environments.

[3] But not they even provide service deployment and management functions

3.5 Context Dependencies

User context information seems to have more importance in future services than in today's ones. Service sessions can be customized based on the actual user context providing more flexibility and automation to the user. Context information has to be collected and managed to make its usage possible in the service which requires the development of a general context collection and management framework. Recent research projects have been touching this issue (e.g., PERCOM [13], MONTAGE [14], CONTEXT [15], paper of P. Mendes et al. [16]) and adapting the results context-awareness can be introduced in our problem scenario, too.

Concluding this section we can see that what we know today is still not enough to solve our distributed, device independent, software-based service provisioning problem in ad-hoc networks. What missing is the development of a service deployment and management framework, which can cope with the self-organized environment; the adaptation of the recent proposals of the discussed areas (i.e., ad-hoc networks, service lookup/registration, active networks, context dependencies) to the infrastructure-less conditions; and the integration of all these pieces into a common system.

4. Our Approach

In Section 2 we investigated a group game scenario to explore the problems of provisioning distributed, device independent, software-based services in ad-hoc networks. As we mentioned, in this research we plan to focus on a subset of the revealed technical issues. Our basic goal is to work out a complete service provisioning system for the introduced scenario, whose core is a new service deployment and management framework to be developed and which integrates the recent research results of the related areas. We divided this work into two phases. In the first phase we intend mainly to design and develop this new service deployment and management framework, since in the second phase we would like to implement it with a dummy game service using Active Networking technology. Both of these phases have several subtasks to be accomplished which are described in the following.

4.1 Phase 1

First we plan to investigate and extend the service description method used in Chameleon [6] making the specification of our game scenario also possible. In Chameleon the service specification is given as an XML document, which follows an XML document type definition (DTD) defining the structure and format of service specifications. As we mentioned previously, the service description is required for service registration, lookup, deployment and management. Since Chameleon is a node-level stuff, its service description part has to be extended to support also the specification of network-level connections among the service nodes.

For service lookup and registry we plan to examine the recent proposals (the Konark system [10] and the virtual backbone directory system [11]) and adapt (extend) one of them to the requirements of the game scenario. For example, the participating nodes in the game constitute by themselves an overlay network which can be considered as a virtual backbone and the similarity can be exploited.

Constituting the main focus of our research we plan to develop a service deployment and management framework which can cope with our game scenario. In this framework we plan to use Chameleon as the node-level service deployment scheme and we will extend it with service management procedures making service reconfiguration and termination also possible. Moreover, we plan to develop also a network-level scheme which can handle the identification of the service nodes and can manage the communication and synchronization between them.

As the proof of the concept, we plan to implement and evaluate our newly developed service deployment and management framework in a network simulator called ns-2 [17]. The intended purpose of the evaluation is to investigate the framework's scalability and performance (e.g., introduced protocol overhead, service setup time, communication delay, etc.) in different, simulated scenarios.

4.2 Phase 2

In this phase first we would like to implement and evaluate the developed service deployment and management framework with a dummy game application using PromethOS [7] active platform. The reason of introducing the use of Active Networking (AN) here is twofold: on one hand PromethOS provides a handy and flexible node platform to execute Chameleon which allows us to utilize our previous experiences; on the other hand ad-hoc nodes act as not only end systems but routers and in this environment the service related benefits provided by AN (i.e., the dynamic programmability of network elements which makes, for example, the introduction of new multicast protocols into the network possible on-the-fly) can be exploited. Concerning the programming language, a working prototype of Chameleon has been already implemented in Java providing platform independency, so we plan to use this language to code our framework, too.

Moreover, we intend to extend the developed service deployment and management framework and the implemented prototype to support the connection to the Internet, as well. This would allow us to investigate the problem field of service provisioning not only in self-organized networks but also in hybrid ones.

And finally we intend to examine the recent proposals of dealing with context dependencies in service provisioning (the CONTEXT project [15] and the general context management framework [16]) and integrate one of them into our system to make the handling of the extended game scenario possible, too.

References

[1] E. Guttman, C. Perkins, J. Veizades, and M. Day: Service Location Protocol, Version 2, IETF RFC 2608, June 1999.

[2] Sun Microsystems, JINI Architecture Specification, Nov. 1999.

[3] Microsoft Corporation: Universal Plug and Play: Background, www.upnp.org/resources/UPnPbkgnd.htm, 1999.

[4] Salutation Consortium: Salutation Architecture Specification, www.salutation.org/specordr.htm, 1999.

[5] Specification of the Bluetooth System, www.bluetooth.com, Dec. 1999.

[6] M. Bossardt, L. Ruf, R. Stadler, B. Plattner: A Service Deployment Architecture for Heterogeneous Active Network Nodes. Kluwer Academic Publishers, 7th Conference on Intelligence in Networks (IFIP SmartNet 2002), Saariselkä, Finland, April 2002.

[7] R. Keller, L. Ruf, A. Guindehi, B. Plattner, PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing, IWAN 2002, Dec. 2002, Springer-Verlag

[8] Ralph Keller, Tilman Wolf, Jeyashankher Ramamirtham and Bernhard Plattner: Programming Active Networks Using Active Pipes. Washington University in St. Louis, Technical Report WUCS-00-27, October, 2000.

[9] Ralph Keller, Bernhard Plattner: Self-Configuring Active Services for Programmable Networks. Fifth Annual International Working Conference on Active Networks (IWAN 2003), December 10-12, Kyoto, Japan.

[10] Sumi Helal, Nitin Desai, Varum Verma and Choonhwa Lee: Konark – A Service Discovery and Delivery Protocol for Ad-hoc Networks. In Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, March 2003

[11] Ulas C. Kozat and Leandros Tassiulas: Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In Proceedings of the IEEE INFOCOM, San Francisco, USA, April 2003.

[12] Robert Haas: Service Deployment in Programmable Networks. Ph.D. thesis, ETH Zurich, 2003.

[13] H. Bakker, J. Heilesen, G. Schapeler, M. Anagnostou, C. Georgopoulos: The Implementation of PSCS Features in a RACE ATM Demonstrator. In Proceedings of the Third International Conference on Intelligence in Broadband Services and Networks-IS&N'95, Heraklion, Crete, Oct. 1995.

[14] D. Prevedourou, M. Anagnostou, et al.: Use of Agent Technology in Service and Retailer Selection in a Personal Mobility Context. Computer Networks, Vol. 31, pp. 2079-2098, Sept. 1999.

[15] IST-CONTEXT project proposal, Annex 1, http://context.upc.es/index.htm, May 2002.

[16] P. Mendes, C. Prehofer, Q. Wei: Context Management with Programmable Mobile Networks, IEEE Computer Communications Workshop (CCW 2003), Dana Point, California, USA, Oct. 2003.

[17] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/, 2002.