

# Context-Aware Handover Based on Active Network Technology

Qing Wei<sup>1</sup>, Károly Farkas<sup>2</sup>, Paolo Mendes<sup>1</sup>, Christian Prehofer<sup>1</sup>, Bernhard Plattner<sup>2</sup>,  
and Nima Nafisi<sup>1\*</sup>

<sup>1</sup>DoCoMo Communications Laboratoires Europe  
Landsberger Str. 308-312, 80687 Munich, Germany  
{ wei, mendes, prehofer, nafisi }@docomolab-euro.com  
<sup>2</sup>Computer Engineering and Networks Laboratory, ETH Zürich  
Gloriastr. 35 8092 Zürich, Switzerland,  
{ farkas, plattner }@tik.ee.ethz.ch

**Abstract.** Context-aware computing can play a major role to improve the services of mobile networking systems. In this paper, we focus on optimizing handover decisions based not only on the signal quality, but also on the knowledge about the context of mobile devices and networks. Since context information and context processing evolves fast, we propose a flexible, integrated approach for context management, which can adapt in several ways. Our architecture encompasses active platforms in network nodes and mobile devices, distributed context management components on these platforms, and service deployment for network services. This flexible architecture is able to actively deploy different handover services. It can manage different kinds of context information and allow mobile devices to be always connected to the most suitable access network. Our architecture is validated in a prototype implementation.

## 1 Introduction

With mobile networks rolling out in our daily life, offering better support for wireless services becomes an important topic. To optimize the services in this heterogeneous environment and fulfil the user needs, context-aware computing is essential. Context-aware computing is defined as a pattern where the application uses knowledge related to a set of environment states to determine and change the application behaviour.

In this paper, we focus on *Handover* (HO) decisions based on context information. The concept of context-aware HO can be defined as follows: a HO procedure that selects a target *Access Point* (AP), based not only on the signal quality or explicit advertisements sent by the AP, but also on the knowledge of the context information of the *Mobile Node* (MN) and the network, in order to take intelligent and better decisions. However, implementing context-aware HOs poses some problems, namely because context information is diverse, dynamic and distributed among network nodes and MNs. Besides this, fragile and/or low-bandwidth wireless links constrain the exchange of context information.

---

\* Currently at King's College London, nima.nafisi@kcl.ac.uk

Therefore, it is required to ensure that the correct context information is available at the right place and at the right time. For this, the context information should be provided proactively to the MN, i.e., before the HO takes place and when the MN has good radio connectivity. Moreover, the type of context information may change over time, and new services emerging continuously. This requires the use of new algorithms to collect and process the context information.

This paper describes a flexible architecture which is able to satisfy the described requirements in order to provide efficient context-aware HOs. The proposed architecture actively deploys different HO support modules, which are able to manage different context information, allowing the MN to be always connected to the most suitable access network.

The proposed architecture includes a framework to manage diverse, dynamic and distributed context information to support context-aware services. This framework is built upon the active infrastructure. It consists of two parts: active platforms installed in network nodes and MNs which allow the flexible installation and use of software modules; second, a service deployment framework capable of deploying different software modules implementing context-aware handover services.

The main contribution of this paper is an integrated architecture for context-aware HO and its evaluation. It uses two layers for flexibility and performance. First, customized modules for efficient context exchange and context-aware decisions are used for context-aware handover. Secondly, the service deployment framework supports the flexible update of the customized modules on the active nodes, when needed. Furthermore, we show practical evaluation results by a prototype including the above three parts of the architecture (i.e., active platform, service deployment framework and context management framework). The evaluation of the proposed architecture is based on a prototype scenario that considers the context information of location of MNs and the traffic load of access networks except for the signal strength.

The remainder of the paper is organized as follows. Section 2 provides an analysis of the requirements of flexible active context-aware architectures. In Section 3, we describe the proposed context-aware HO architecture based on active network technology. The functionality of the proposed architecture is illustrated and evaluated in Section 4. In Section 5, we take a look at research issues and prior work related to context-aware services and service deployment, and Section 6 concludes the paper.

## 2 Requirements of Context-Aware Handover Service

Our goal is that the mobile node behaves in an optimal way depending on the context. This requires a mechanism to efficiently collect and manage the context information and an appropriate platform to use the context information for an optimal decision.

Context information may be classified as static or dynamic, depending on the frequency and cause of changes, and based on the location where such information is maintained. Table 1 explains our classification of context information. It is just a snapshot, as novel kinds of context information may appear. For instance, in future networks, new kinds of context information like user groups may be relevant. Some items, such as the user profile, appear twice, as the information is often spread over the user device, the operator and possibly several service providers. As an example,

the user profile may include subscribed services and service preferences, e.g. which services have to be downgraded or dropped if sufficient resources are not available.

**Table 1.** Context Information Classification

	Context information on mobile device	Context information on network side
Static	User settings and profile Application settings	User profile and history Network location Network capabilities and services Charging models
Static within a cell	Reachable APs	Potential next AP
Dynamic	Type of application Application requirements Device status (battery, interface status, etc)	Location information and location prediction Network status Network load

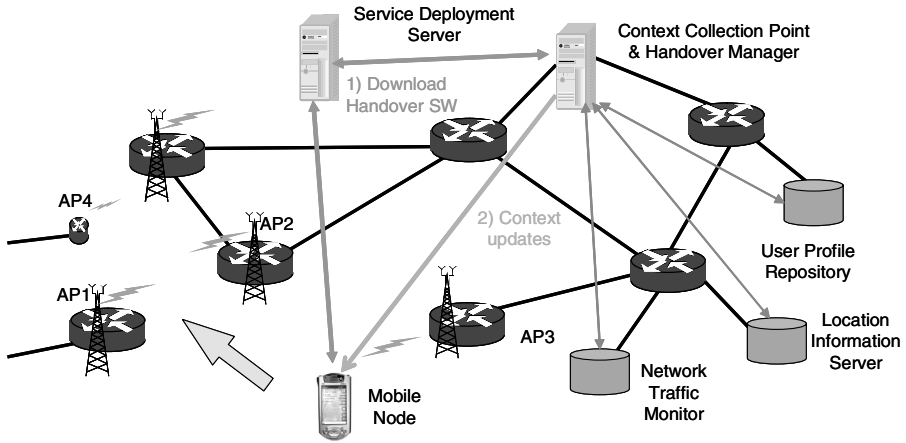
However, context information is not readily available to the involved entities in a HO for several reasons. Firstly, context information is distributed. For instance, some pieces of context information may be available in user's home network, some may be available in the visiting network and some resides on the terminal. Secondly, dynamic context information may change frequently or lose accuracy over time. For instance, it is tempting to convey information about the current AP loads, yet its relevance decreases quickly over time. Thirdly, what is relevant about context information and the methods to interpret it may evolve over time. Hence algorithms for interpreting the context data need to be adapted to new requirements.

Therefore, we need a context management framework which assures that the right information is available just in time. Still, information exchange between network and MNs should be minimized to save wireless resources. Furthermore, a context-aware handover requires an appropriate execution platform which is flexible enough to adapt to the changing requirements of this service. It should be able to cope with continuously changing context and automatically alter the handover decision policy or algorithm in use. It should support continuously the context exchange between the nodes involved in the handover service in an efficient way. It should also enable the mobility management of the mobile node to take the right handover decision.

### 3 An Integrated Approach for Context Management

This section introduces an approach which integrates a context management framework, an active platform and a service deployment scheme to provide the functionalities needed for context-aware HO. The context management framework is in charge of collecting the relevant context information for different services and managing the context information. The active platform is used to exchange and process the context information. The service deployment scheme is used to synchronize and manage the working of involved active nodes. In our architecture shown in Fig. 1 context information is stored in context information repositories (e.g.,

*Location Information Server (LIS), Network Traffic Monitor (NTM), user profile repository, etc).*



**Fig. 1.** Architecture for Context-aware HO Using AN Technology

The LIS is responsible to track the position of each mobile device in the provider's network and has the knowledge of nearby APs, while the user profile repository stores the user profiles as seen by the network service providers. We introduce a *Handover Manager (HM)*, which controls the handovers carried out in some part of an access network. The HM is responsible for filtering and processing HO-related context information. Finally, a *Service Deployment Server (SDS)* is used to manage and install the service modules needed in the network nodes and mobile nodes.

The realization of this service can be divided into three steps, which are explained in the message sequence diagram shown in Fig. 2. Step one (phase A,B,C in Fig. 2) is service deployment, which includes fetching the right service components for a required service, installing them on the appropriate network node and confirm the successful installation of all the components for this service. Step two (phase D in Fig. 2) collects relevant context information. In step three (phase E in Fig. 2) the context information is evaluated and the HO decision is made.

Fig. 3 shows the execution platforms of the nodes used in our scenario (see Section 4), which is the execution environment where active modules are installed. For example, to realize a context-aware HO service, we need to install a context exchange protocol - HO support module (HSM) and a HO decision mechanism - HO decision module (HDM) in the related nodes. More specifically, here we need to install appropriate versions of both the HSM and the HDM in the active platform of the MN and the HM. This is realized by the service deployment in Phase A, B and C. The signalling used by the HDM to request the needed context information is detailed in phase D. After the HDM made the decision on AP based on the collected context information, the decision is sent to the *mobility management component* of the mobile node to execute the handover (phase E); in our approach, mobility management is done with Mobile IP.

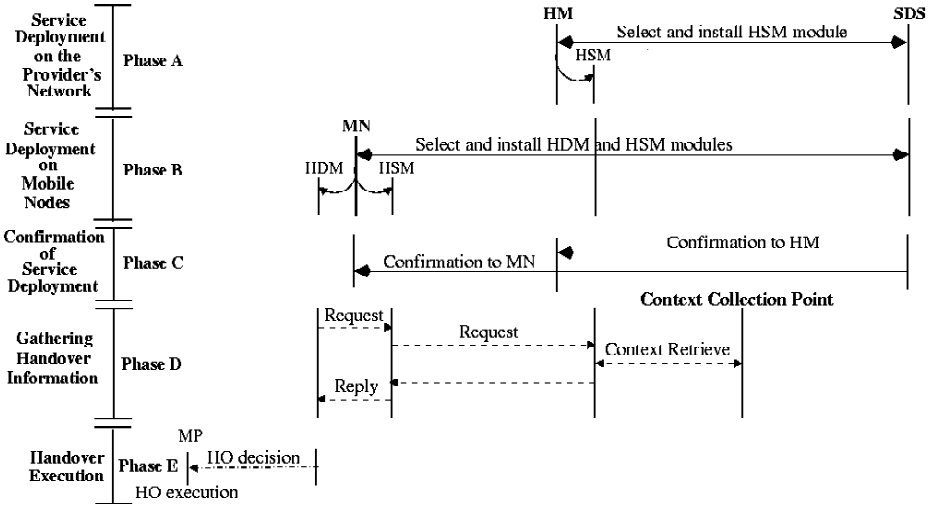


Fig. 2. Sequence of Signalling and Processing for Context-aware Hos

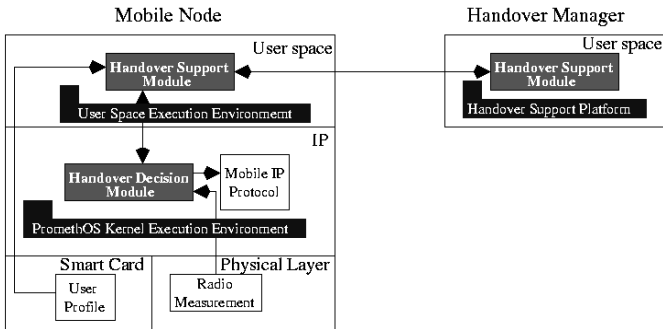


Fig. 3. Active Modules for Context-aware HO

### Context Management Framework

Our context management framework is in charge of context collection, compiling, exchanging and evaluation. Details of this framework can be found in [10]. It defines the following main entities:

- The *context collection point* on the network side, which collects and compiles the relevant context information from different sources. In our scenario, the context collection point is placed in the HM shown in Fig. 1.
- The *HO decision point* decides which AP is to be selected for the handover. In our scenario, the HO decision point is placed in the MN, as shown in Fig. 1.

The HO decision point uses simple algorithms (e.g., rule-based logic) for interpreting the data delivered by the context management framework. It is the task of the service

deployment service to assure that both the HO decision point and the context collection point are proactively supplied with the appropriate algorithms.

The algorithms need to be altered if there are structural changes of the context information, e.g., when the context format, user profile or context processing algorithm changes. Notice that software updates can be applied in different ways. For instance, each time the user profile changes, e.g. when entering a car new software modules are installed. The context information will be exchanged when needed using context exchange protocol. In this way, up to date context information is used for HO decision. The detailed steps for context-aware HO are:

1. Prepare a software module at the context collection point with the algorithm and the collected context data needed at the handover decision point.
2. Download the software module proactively to the HO decision point before HO.
3. Exchange context when needed.
4. The software module is invoked at the HO time. It makes the decision on AP with the input of the dynamic terminal context (e.g., reachable access points, application requests and sessions), and transferred dynamic context from the network.

In the architecture described in section 3, the HO decision algorithm is implemented as HDM which is installed in MN, and the context exchange protocol is implemented as HSM which is installed in both MN and HM.

### **Active Node Platform and Service Deployment**

In the following, we describe the node platform for our architecture. We use active networking technology to meet the requirements in section 2. Since the decision algorithms for optimal AP are context-dependent, the network elements and the mobile end systems involved in this process need to be programmable. Active networking technology is a good candidate to fulfil this requirement. Our active node consists of the basic processing hardware, a node operating system and several execution environments, in which active applications will execute the handover algorithms. The node has to support the dynamic installation of a handover decision module at run-time, without interrupting the node's proper working. The active node architecture and implementation we use for our system is PromethOS [8]. It is a generic platform for running active applications in a Linux environment, allowing for on-demand installation of user- or kernel-space modules.

While PromethOS provides a basic active node platform, our application scenario also calls for a framework capable of handling the selection, installation, configuration, and management of the service components. The Chameleon service deployment framework [7] can accomplish these functions. However, Chameleon, in its current implementation, only provides node level service deployment functions. We therefore extended it with a simple, centralized network-level service deployment framework called Octopus. In the rest of this section, we first give a brief overview about our use of Chameleon and the new Octopus framework.

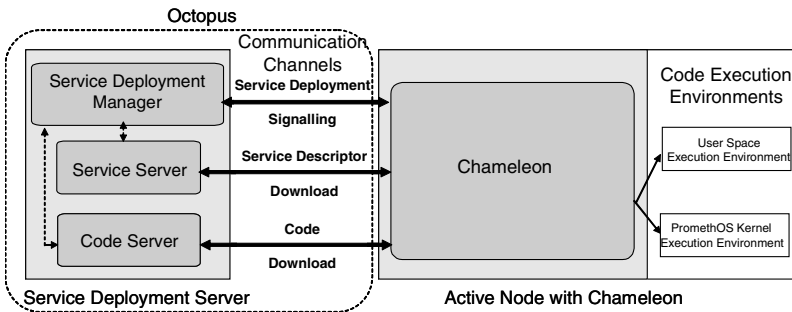
### **Chameleon, a Node Level Service Deployment Framework**

Node-local service deployment comprises selecting, downloading and installing implementations of service components on an active node, such that these

components jointly provide the specified service. Chameleon uses a service specification – to be generated by the network-level service deployment – and a description of the intrinsic properties of the active node to determine which implementations of service components need to be installed on the active node. The service specification is given as an XML document, which follows an XML document type definition; the latter defines the structure and format of service specifications. Chameleon resolves the service specification against the description of the node properties, thereby creating a tree-like structure representing all possible implementations of a service. In our application of Chameleon, such service specifications are generated by Octopus, our network-level service deployment scheme supporting context-aware handovers.

### Octopus, a Network Level Service Deployment Scheme

The core of Octopus is a central management entity, called SDS, as illustrated in Fig. 4. The SDS contains a *Service Deployment Manager* (SDM) module which controls the network-wide signalling and all related synchronization functions needed during service deployment. Moreover, it contains a *Service Server* (SS) which stores the descriptors of the services known to the system and a *Code Server* (CS), which stores the implementations (code modules) of the service components available in the provider's network. These servers are managed by the SDM; they can be located anywhere in the network.



**Fig. 4.** Communication between the SDS and a Chameleon Node

Fig.4. shows how Octopus and Chameleon communicate. Three signalling channels are used: (1) Service deployment signalling for requesting and/or sending the proper service specification to the Chameleon instance on the active node, (2) Service descriptor download signalling for retrieving service descriptions, to which the service specification (1) refers; the download is initiated by Chameleon while the service specification is resolved; and (3) Code download signalling, which is used by Chameleon to retrieve the implementation modules needed from the Code server for execution of the service; this retrieval, again, is initiated by Chameleon.

The specific operation of Octopus varies depending on where Chameleon is executed.

- If the node is a network node such as HM in our context-aware HO service, then the SDM module of SDS establishes a connection to the node's Chameleon on the signalling channel and requests the installation of the given service. Subsequently, the Chameleon instance on the HM autonomously resolves the service specification and installs the component implementations.
- If the node is an end user node as the MN in our context-aware HO service, then this node's Chameleon initiates the service deployment by requesting a list of available services and selecting a service from this list; the SDM then sends the requested service specification and Chameleon proceeds as in the previous case.

Service deployment can be either provider-initiated or user-initiated. In the former case, the service is deployed in the network proactively, before the arrival of any service user. While in the latter case, service deployment is on demand. The arrival of the first service user initiates the installation of the service. The extended Chameleon with Octopus can cope with both policies and the active nodes can act either as network node or end user node by setting the parameter of Chameleon.

## 4 Prototype and Evaluation

In this section, we evaluate the proposed architecture by using a prototype that illustrates a scenario, in which a provider has three access points, and a user is driving or traveling by train along a trajectory leading through an area where all three networks intersect. Without context aware HO mechanisms, the selection of a new access network during a handover would be driven by Mobile IP version 6 (MIPv6) [16] route advertisements. In this case, the mobile device may register with any of the possible destination networks. However, if active networking is used, the user's MN can request the utilization of a HO service based on context information.

To illustrate the behaviour of the proposed architecture, two types of context information are considered: the user's location, speed and trajectory, which depends on how the user moves, and the *Quality of Service* (QoS) required by the application. Considering the user's location-context, the HO may be made to a network that has a better coverage along a train track or to a network with small range, but high QoS if the user is walking. Additional criteria could be the network traffic load, which normally would have an effect on the QoS perceived by the user.

### Prototype and Scenarios

The used prototype, illustrated in Fig. 5, encompasses one HM that controls the three access networks. To reduce the complexity of the prototype, the behaviour of the LIS and NTM is simulated with static information, and only one type of wireless interface is used. We make the distinction between overloaded and non-overloaded networks, by generating more traffic to specific networks, using the MGEN tool-set [17].

Four scenarios are used in the prototype. Scenario one is based on MIPv6 HO. The remaining three scenarios aim to show the advantages of using context information and active networking to choose a right AP, while the hypothetical user travels in a train. Scenario two uses a service based on the user's location-context. Scenario three



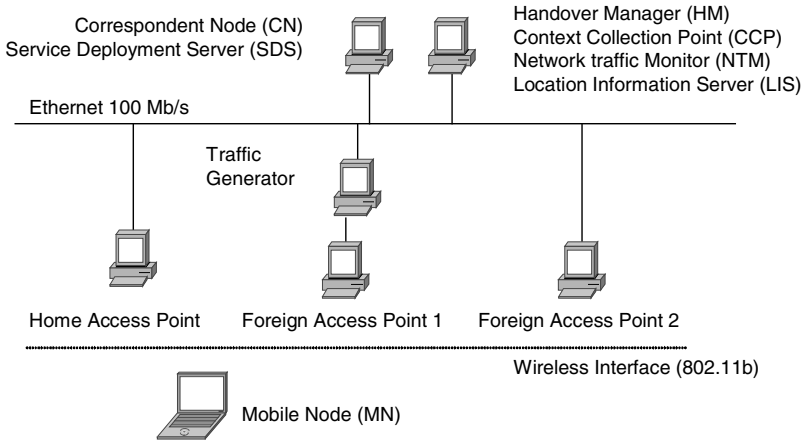


Fig. 5. Prototype for Context-aware HO

also considers the user's location-context, but in this case the user is receiving a streamed video sent by the CN. In the fourth scenario, the user is also receiving a streamed video, but we also consider the QoS requirements of this application. For scenarios three and four, *Foreign Access Point* (FAP1) is first unloaded and then is heavily loaded, while in any scenario the *Home Access Point* (HAP) and FAP2 are always unloaded. In each scenario, a context-specific HSM and HDM, able to process information about the user's location-context and/or QoS-context are downloaded, installed and used by the HM and MN.

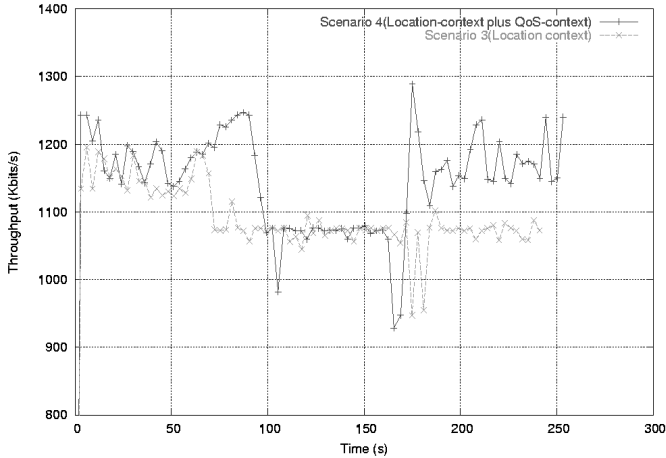
## Experimentation and Evaluation

The goal of this analysis is twofold: first, to exemplify the behaviour of the proposed architecture, as a proof of concept. The second goal is to evaluate the signalling overhead of the proposed architecture and to understand how much time it requires to deploy context-aware services and to collect customized HO data.

In what concerns the behaviour of the proposed architecture, scenarios one and two show that with context information MNs have higher control over their HOs. In scenario one, the MN registers with the FAP from which it gets the first route advertisement. However, in scenario two, the MN registers with FAP1, independently of the order of received router advertisements. This corresponds to the assumption that the MN is in a train and that FAP1 has the best coverage along the train track.

The results of the experiments done with scenarios three and four, illustrated in Fig. 6 demonstrate the importance of using the appropriate modules.

In scenario three, the MN registers with FAP1, at second 70, since this is the network with wider coverage near the train track. This ends up in the deterioration of the perceived video quality since FAP1 is loaded and the used context-aware modules do not consider the load of networks. In scenario 3 the video remains with low quality, since the MN is only aware of the user's location, which does not change.



**Fig. 6.** Throughput of the Video Stream on Scenarios Three and Four

In scenario 4, the MN also registers with FAP1, because FAP1 and FAP2 are both unloaded in the beginning, and so the HO decision is only dependent on the location context. Starting at second 90, FAP1 is loaded to show that with location-context plus QoS-context modules the MN can react properly to significant changes in the load of networks. As shown in Fig. 6, the MN registers with FAP2 at second 170, after receiving and processing the new context information, and the perceived video quality increases again. The throughput oscillations shown in Fig. 6 are mainly due to the instability on the wireless link.

In the current prototype, the service deployment takes about 850ms in the HM and up to 9.3s in the MN. This long time required to deploy a service in the MN is mainly due to the fact that in the current prototype, the user is allowed to select the service to be installed in the MN based on the list received from the SDS. Concerning the exchange of context information, our measurements show that it takes 1.8 ms.

We also conducted an analysis about the communication overhead involved in collecting context information. The length of the packet sent by the MN to request context information is 64 bytes: 60 bytes of TCP/IPv6 headers and four bytes to indicate the current user's context. As a reply, the HM sends to the MN information about each access network. With location-context-aware modules, such information occupies 20 bytes per access network: 16 bytes for the IPv6 address of the AP and four bytes indicating the priority of the network. With location-context plus QoS-context modules, four bytes corresponding to the load of the network are added to the information sent to the MN. Considering a scenario with 1000 access networks, in which the user is requesting location-context plus QoS-context modules, and the *Maximum Transfer Unit* (MTU) is 1500 bytes, the control information is of 24.4 KB. Considering a worst-case situation where the load of any AP changes every 1s, and the HM is configured to check the load of the networks also every 1s, the control information overhead is of 199.7 kb/s.

## 5 Related Work

In [12][13], sophisticated handover procedures have been considered. However, the parameters, which have been considered for the handover decision, are confined to the type of the radio access technology plus the signal strength. In [14], different handover policies for heterogeneous networks are used, considering as handover parameters mainly the air-interface type and the available bandwidth at the access router. There are several approaches aiming at a more intelligent handover in heterogeneous access network (e.g., [15]). Yet none of them presents a general framework for a handover mechanism that will benefit of the various contexts the mobile node might be confronted with.

In the area of active service deployment several proposals have been made, but often these are restricted to the needs of a specific platform (e.g., the Active Networks Daemon [2] of ABone [3]). Some of them have the potential to be used as a generic service deployment scheme (e.g., ASCP [4]). Recent publications discuss approaches, which are explicitly targeted towards a generic solution for service deployment, e.g. ASDP [5] or pattern based service deployment [6] with Chameleon [7].

## 6 Conclusions

This paper describes an architecture that aims at optimizing mobile network services based on context information. This means to gather information from different network elements and to use this information in mobile nodes for local context-aware and better decisions. Our proposed architecture allows for a flexible use of different protocols to exchange different types of context information, as well as a flexible use of different context-aware decision algorithms in mobile nodes. Our solution ensures that the correct context information is available at the right place at the right time, and handles diverse, dynamic and distributed context information.

Our solution integrates three main parts: first, we use an active platform installed in network nodes and mobile nodes. Second, we integrated a framework capable of performing network-wide and node-local deployment of various context-aware services on these platforms. Third, use an efficient and flexible context management to handle diverse, dynamic and distributed context information.

The active platform and our service deployment framework are not specific to the context-aware hand-over service but it is more generic and can be used in case of other network services. The deployment framework is scalable in two dimensions, concerning the number of services and the size of network (number of involved nodes in a service).

Our evaluation of the proposed architecture, based on a prototype, has shown the feasibility and utility of our approach. Specifically, we show that context awareness increases the efficiency of hand-overs. We also show that the deployment of active modules is fast in the network and discuss the problems of service deployment to mobile nodes over the wireless link. The evaluation results show that customized exchange of context information is quick and that it does not have a significant communication overhead. Only if the context processing or exchange formats change, the modules have to be updated.

## References

- [1] Jamalipour, S. Tekinay (eds.): IEEE Personal Communications Magazine, *Special issue on Fourth Generation Wireless networks and Interconnecting Standards*. October 2001.
- [2] *Anetd: Active Networks Daemon*. ACTIVE project, ISI & SRI, <http://www.sdl.sri.com/projects/activate/anetd/>
- [3] Branden, L. Ricciulli: *PA Plan for a Scalable ABone – A modest proposal*. Technical Report, USC – Information Science Institute, January 1999.
- [4] E. Amir, S. McCanne, R. Katz: *An active service frame-work and its application to real-time multimedia transcoding*. ACM SIGCOMM'98, Canada, 1998.
- [5] M. Sifalakis, S. Schmid, T. Chart, D. Hutchison: *A Generic Active Service Deployment Protocol*. In proceedings of the Second International Workshop on Active Network Technologies and Applications, ANTA, Osaka, Japan, May 2003.
- [6] M. Bossardt, A. Mühlemann, R. Zürcher, B. Plattner: *Pattern Based Service Deployment for Active Networks*. In proceedings of the Second International Workshop on Active Network Technologies and Applications, ANTA, Osaka, Japan, May 2003.
- [7] 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.
- [8] R. Keller, L. Ruf, A. Guindehi, B. Plattner, *PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing*, IWAN 2002, Dec. 2002, Springer-Verlag
- [9] Matthias Bossardt, et al. *Integrated service deployment for active networks*. In International Working Conference on Active Networks(IWAN2002), Zurich, Switzerland
- [10] Christian Prehofer, Nima Nafisi, and Qing Wei. A framework for context-aware handover decisions. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Beijing, China, September 2003
- [11] Christian Prehofer and Qing Wei. *Active networks for 4G mobile communication: Motivation, architecture and application scenarios*. IWAN 2002, Zurich, Switzerland
- [12] M. Stemm and R. Katz, *Vertical handoffs in wireless overlay networks*, ACM Journal on Mobile Networks and Applications, Vol. 3, No. 4, 1998.
- [13] K. Pahlavan, et al, *Handoff in hybrid mobile data networks*, IEEE Communication Magazine, April 2000.
- [14] Helen J. Wang, Randy H. Katz, and Jochen Giese. *Policy-Enabled Handoffs across Heterogeneous Wireless Networks*, In WMCSA 99, IEEE, Feb.1999
- [15] Michael E. Kounavis, et al, *Design, Implementation and Evaluation of Programmable Handoff in Mobile Networks*, ACM Journal on Mobile Networks and Applications (MONET), September 2001
- [16] Johnson, C. Perkins, and J. Arkko, *Mobility Support in IPv6*, IETF Internet-Draft, June 2003, Work in Progress.
- [17] MGEN: The Multi-Generator Toolset, <http://manimac.itd.nrl.navy.mil/MGEN/>