

INTERWORKING BETWEEN LINK LAYER AND APPLICATION LAYER ADAPTATIONS IN A RECONFIGURABLE WIRELESS MIDDLEWARE

E. Mohyeldin¹, M. Dillinger¹, M. Fahrmaier², P. Dornbusch² and W. Sitou²
¹SIEMENS AG, ²TUM

E-mail: eiman.mohyeldin@icn.siemens.de

Abstract

Dynamic reconfiguration aims at maximizing the performance of applications and services under varying wireless channel conditions. In this paper, we describe how to adequately demonstrate the middleware concept for reconfiguration. It is clear that the reconfiguration of reconfigurable terminals requires complex interactions between the terminal and the network entities, and occasionally even the download of new software modules to be installed on the terminals. To enable mobile devices for SDR, a middleware technology will be implemented which follows the developed mode negotiation and switching methodologies. The feasibility of reconfigurable architecture and context/profile based service provisioning is proven in this contribution.

Keywords: SDR, Reconfiguration, Context, Wireless, Middleware, Layer Adaptation.

1. INTRODUCTION

The support of reconfiguration requires the existence of negotiation procedures and trading services connecting the device reconfiguration manager with the corresponding network proxies that will operate between the network proxies and the managed equipment, and will enable the exchange of information (e.g. mode availability, access network capabilities) or the ordering of certain procedures to take place. The task of the reconfiguration is to gather all the required software modules and carry out the individual reconfiguration steps. The reconfiguration manager expects the trading and negotiation services to act as authority that it can consult and which will deliver the necessary profiles and supporting service. The Middleware technology with services as trading services is the subject of investigation in this paper. Middleware is considered here for its potential in addressing the following issues related to software (parameters) download for the reconfiguration of mobile devices: locating of file server, authentication of subscriber and software distributor, capability exchange and transfer of software defined radio parameters.

In previous research work [1] a series of tests was run for the reconfiguration deployment time. The time spans of interest were the time required to deploy a reconfiguration, e.g. how long the client would need to switch between two different service providers to match quality of service used to a change of network conditions from a high bandwidth to a low one. It was found that the time needed to detect and

signal a profile parameter (like application QoS) from the terminal to the profile server is about 18% from the Total Reconfiguration Time (TRT). Signaling profile-changed events “this is similar to synchronization or updating the profile” from server to the reconfiguration manager is again 14% (if profile server and reconfiguration manager share the same location). Acquiring the trading decision from the trading manager is 28%. Signaling back the decision to the terminal is another 18% and switching to the new service for the client takes another 21%.

In this work an online parameters/update of the network profile, the network load and user throughput are included, the update of profile data entries and its trading time is measured.

This paper is organized as follows: section II presents the reconfigurable middleware layers followed by the overview of the link layer adaptation in section III. The system overview and test bed are shown in section IV. Section V deals with the evaluation of the performance parameters. The measured results and analysis are depicted in section VI. Section VII concludes the paper.

2. MIDDLEWARE RECONFIGURATION LAYERS

Reconfigurable Middleware is an evolutionary improvement of middleware that was developed for ease of development of distributed systems like for example CORBA [2]. With the development of middleware, various aspects of mobility and system dynamics in wireless or ubiquitous systems like ad-hoc resource discovery, dynamic binding, and interruption handling can be handled. Compared to traditional distributed systems with static architecture, homogeneous environment and high reliability, dynamic systems have a limited reliability with respects guaranteed wireless resources for the duration of system runtime (e.g. due the high variations of the mobile channel) and hence architecture, functionality or implementation during runtime should be modified and updated to maintain some kind of best effort functionality instead of releasing or dropping the communication [3]. The middleware is involved in different layers of the system reconfiguration; we have identified three layers that are subject for our investigations:

- **The Infrastructural (or Link) Layer:**

The Infrastructural Layer contains reconfiguration of quality of basic services that are shared among several applications. This mainly affects the basic communication (layer 1-4) that is used to connect the distributed subcomponents (terminals,

servers, etc.) into one distributed application. In other words, we speak here of the radio link layer.

- **The Application Layer:**

The Application Layer contains reconfiguration of application functionalities that could directly or indirectly depend on radio resources with an availability that is not controlled by the application itself. Usually this means functionalities that depend on not guaranteed infrastructure like varying communication QOS. For example changing from UMTS to GSM coverage might make the video part of an instant messaging application obsolete and therefore switched off.

- **The User Layer:**

The User Layer contains reconfiguration that might affect several applications at once, for example combining a number of smaller applications into the terminals portal or personalizing the behavior of several applications according to a single user preference (e.g. best price vs. best quality).

It is obvious that the negotiations on different layer types can have different requirements. On the lower layers simple rule based mechanisms for fast decisions and reconfigurations might be the way to go. Whereas more sophisticated and resource consuming decision logic like a learning neuronal network might be the decision type of choice for user level personalization.

But in this paper focus the first two layers. Moreover we concentrate on the question, how the adaptations and reconfigurations carried on the lower layers can affect the application adaptation/reconfigurations.

3. OVERVIEW ON LINK LAYER ADAPTATION

Radio links in mobile communication are always subject to fading effects and high interference. For a given transmission scheme (link mode), i.e. modulation format, spreading factor, coding scheme, etc., a certain amount of received power is required at the receiver side. The so called link adaptation scheme is used by many radio systems, e.g. in GPRS, up to four different coding schemes are defined; nine different modulation and coding schemes are standard in EGPRS which gives user data rate varies from 8.8kbits/s to 59.2kbits/s. 3GPP HSPDA provides even more flexible transmission modes thanks to the code division. The introduction of additional downlink channels (HS-DSCH) in UMTS FDD provides downlink data peak rate up to 10Mbits/s. The optimal link adaptation scheme by *in time* selecting modulation and coding schemes brings higher system performance even in poor radio conditions.

Interworking between radio networks, layers and service types are from great importance for the future radio network. In UMTS WCDMA system with the co-existing different types of services, how the link adaptation algorithm works in order to obtain optimal radio resource allocation is still under investigations. Figure 1 illustrates the behaviour pattern for the network (represented by the

effective network bandwidth) and the usage pattern of the application/services.

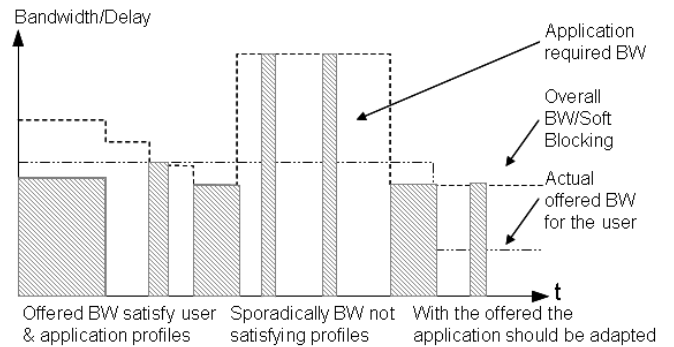


Figure 1: Example for Bandwidth and Service Patterns

In this scenario, the link changes/adapts its throughput (BW), by means of the above mentioned link adaptation methods, and hence, other layers (in this case the application) are affected and should be adapted to the new situation. This must be achieved with minimal interruption time.

4. SYSTEM OVERVIEW AND TEST BED

The main system function groups are:

- Application
- Terminal Management
- Simulation
- Trading
- Profiles
- Evaluation
- Reconfiguration

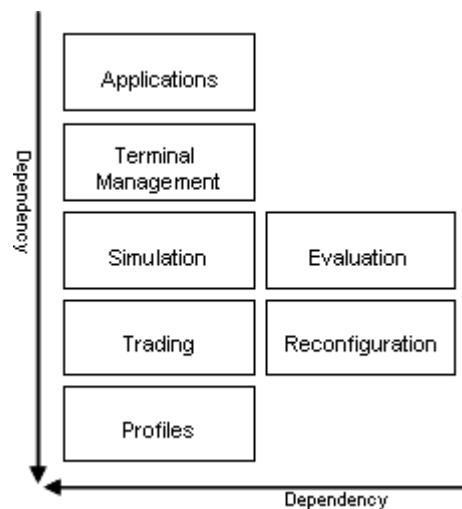


Figure 2: System Components

In Figure 2 a general overview is depicted for the system components and their dependences, details are in [4].

Test platform

The structure of the testing platform is depicted in Figure 3. There is one server on the fixed network that represents the home network and holds the Home Reconfiguration Manager (HRM) (HRM provides support for individual software downloads, coordinates mass software upgrades and communicates with external software suppliers). A laptop server that holds a proxy reconfiguration Managers (PRM) (The PRM serves as download proxy that performs short-term caching) and is associated to an 802.11 wireless network access point represents each location. The PRM laptops can be connected to the HRM over an Intranet or the Internet. This way the demonstrator can easily be presented in different physical locations. The mobile terminals can roam between different PRM networks. The PRM network is also used to determine the location of a terminal. However since the demonstrator itself is modular and scaleable, more than two PRM networks can be added by adding either additional Access Points and Laptop servers or only APs (the laptop server then holds more than one instance of a PRM implementation).

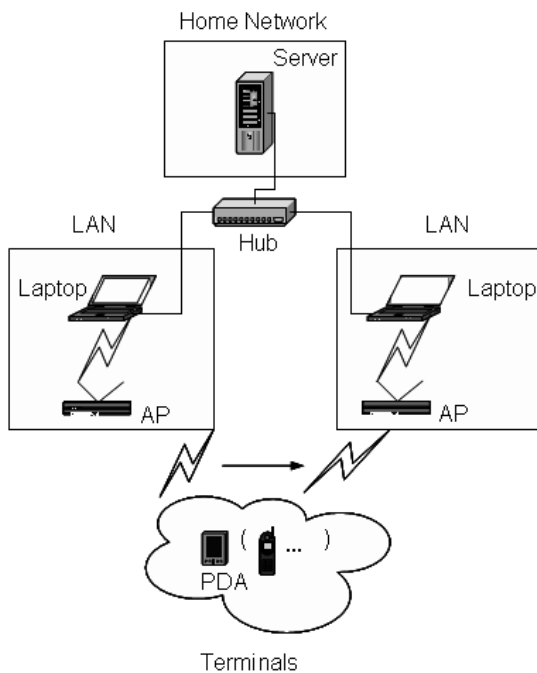


Figure 3: Testing platform Structure

5. EVALUATION OF THE WIRELESS MIDDLEWARE PERFORMANCE PARAMETER

The main objective of the evaluation is to get quality values that can indicate whether the whole concept of a runtime reconfiguration is feasible. Furthermore, how certain performance and resource parameters will scale, how effective certain optimisations compared to others are and

where the bottlenecks of the system and therefore candidates for further optimisations can be found. Here we have only emulated parts of a mobile telecommunication system infrastructure; the absolute numbers measured can not be transferred directly into absolute numbers for a real mobile telecommunications system infrastructure. However, given an appropriate analysis and interpretation of the results, these values can provide some insight of what can be expected if the concept should be deployed in a real production environment. A typical reconfiguration that takes place during the current usage of a service by a client (i.e. runtime) can be described regarding its timing behaviour. During the whole reconfiguration process there are three major important time spans:

- The time needed to update/inform the decision-making unit that decides about a reconfiguration about a profile change, i.e. the time to migrate the profile change from the entity where the profile information is generated to the decision-making unit “Trading Manager”. The overall time span regarding this process is called Context Recognition Time (CRT).
- The time needed for the decision-making unit in deciding about a new configuration. This of course depends on several things like the complexity of the decision itself and any further profile information that might be involved in this decision. The total time span from receiving an event about changed context information in the decision unit until a reconfiguration decision is ready to be deployed is called Trading Time (TT).
- The time needed for the system to propagate and completely change/reconfigure according to the new configuration and until any suspended session was successfully resumed if possible is called Reconfiguration Deployment Time (RDT).

All three time spans together describe the total reconfiguration time ($TRT=CRT+TT+RDT$). The TRT is a technical timing that contains everything from the actual change of a physical parameter (e.g. bandwidth) until the new system configuration has been deployed.

Additionally to this TRT there is a Handover time (HT). The handover time describes the reconfiguration from a user point of view. Depending on middleware intelligence and application type, HT can be significantly smaller than TRT because it only contains time spans of lower (than possible) QoS or interruption visible to the user.

Testing system

The testing subject is a prototype which is an implementation of a mere use case of a reconfigurable architecture. The middleware implementation contains all implementations that are necessary to run a limited number of scenarios. For this prototype the testing environment is a mix mode setup of a real algorithms and hardware as well as simulated modules. Real hardware was used where available

and technically feasible. The rest consists of emulation components, especially for different network types and characteristics.

6. MEASUREMENT ANALYSIS

Profile Access Time

The profile access time is the time needed to update the profile entries and inform the trading manager about the update. This time includes delay from the physical context information change until the change is detected by sensors and signaled to the location where it is processed. Profile size in this test is 1Kbytes. The result for different simulated bandwidth (64kbit/s, 144kbit/s 2Mbit/s and 100Mbit/s) and number of concurrent threads (parallel accesses/number of concurrent users) is depicted in Figure 4. From these results it can be deduce that the signaling part needed to update the profile data and transmit the change to the decision entity is marginal in networks with high bandwidth, therefore the effects of profile data on the trading time is negligible there. The minimum time for profile update was measured to be 0.099% of the total trading time. However in this prototype, the trading manager is located in the same location as the profile database, i.e. low delay is resulting. In typical wireless networks with distributed architecture signal delay should be added.

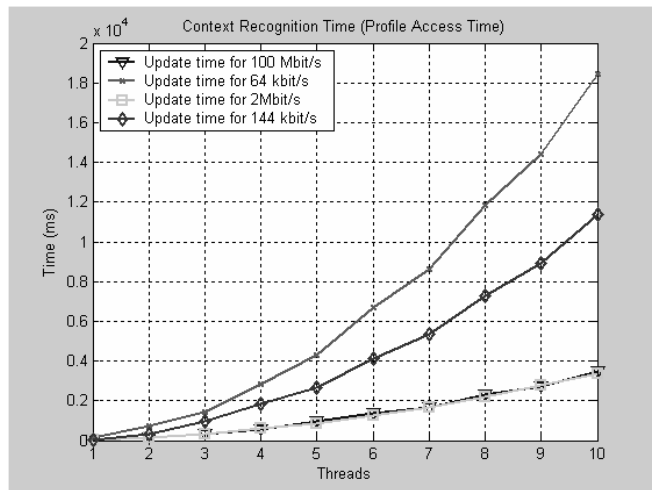


Figure 4: Profile Access Time for different simulated BW

Round Trip Trading

In this test the TRT for a real world reconfigurable HTML application was tested. This includes signaling changes of the bandwidth, trade & negotiates on these changes and signaling back the reconfiguration parameters and deploying it into the application. In our case we had three possibilities to reconfigure the application in accordance with the bandwidth change namely using a network gateway proxy without pictures for reduced or low bandwidth, compressed pictures for medium bandwidth and complete picture and text for high bandwidth situations. Reconfiguring an HTTP based application is one of the more advantageous scenarios

since breaks between viewing two pages can be used to mask the TRT towards the user usually resulting in no or very low handover time (HT). However this is a quite common scenario type for reconfiguration compared to the worse case scenario of real time streaming video.

The TRT measured in the tests ranges from 30ms to about 190ms for the not optimized prototypical implementation of this middleware prototype, see Table 1. This is considerably faster than similar middleware approaches [5] and clearly shows that reconfiguration is feasible in mobile communication environments.

Bandwidth	Average Round Trip Trading Time (ms)
2Mbit/s	36
144kbit/s	84
64kbit/s	195

Table 1: Round Trip trading Time for different Bandwidths

7. CONCLUSION

A fundamental requirement for reconfigurable wireless middleware is a need for advance trading mechanism that allows application and service to adapt to available resource in dynamic and reconfigurable networks. From the test results it is obvious that the profile size effects on the trading time is negligible, however the total trading time is significantly high in networks with low bandwidth (e.g. at 64kbit/s a TRT of 195ms). Strategies for optimization are carried out by reducing the amount of signaling. This requires optimized reconfiguration protocols especial for lower layers with high speed reconfiguration frequency.

8. OUTLOOK

Further work will mainly concentrate on measuring the various effects of the single basic middleware components and their communication mechanisms. Further investigations will include:

- Download time and different profile sizes
- Optimization strategies for compression e.g. on-the-fly compression of XML signalling.

REFERENCES

- [1] IST-2001-34091 Consortium SCOUT, <http://www.ist-scout.org>
- [2] R. Ben-Natan, *CoRBA: A guide to Common Object Request Broker Architecture*, McGraw-Hill, Inc., 1995
- [3] G. Clemo, T. Farnham, N. Georganopoulos, M. Seneclauze, B. Souville, T. Schöller, S. Buljore, "Requirements on Service Discovery, Middleware and Flexible Protocols fore Software Defined Reconfigurable Terminals, IST Summit 2003, 15-18 June 2003, Aveiro- Portugal.

- [4] DuReau P., Redmill D., Mohyeldin E.; Golubicic Z.; Hirschfeld R.; Fahrmaier M.; Salzmann C.; Dornbusch P., "Description and Specification of the SCOUT hardware validators and SCOUT middleware demonstrator" IST-2001-34091 D4.3.1, 2003-11-02.
- [5] Chan and S. Chuang, "MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing", IEEE Transaction on Software Engineering, Vol.29, No. 12, December 2003.