Achieving "Always Best Connected" Through Extensive Profile Management

Lucian Suciu[1], Jean-Marie Bonnin[1], Karine Guillouard[2], and Bruno Stévant[1]

[1] École Nationale Supérieure des Télécommunications de Bretagne
BP 78 - 2, rue de la Châtaigneraie, 35512 Cesson-Sévigné, France
{lucian.suciu, jm.bonnin, bruno.stevant}@enst-bretagne.fr
[2] France Télécom R&D, DMR/DDH Laboratory
BP 59 - 4, rue de clos Courtel, 35512 Cesson-Sévigné, France
karine.guillouard@francetelecom.com

Abstract. The integration of various access networks into a ubiquitous, yet heterogeneous, wireless environment is on the way. This evolution of the mobile network will give the end-user a greater choice of access technologies, and, therefore, the decision to select the "best" interface and access network from many possible combinations has to be taken. The decision will depend on information such as: performances and capabilities of the available networks, requirements from applications, user preferences, or network operators' constraints. Our work focuses on an advanced middleware which deals with profile management to support the interface automatic configuration and selection. Furthermore, the proposed mechanism supports the dynamic (re)mapping of the application flows by taking into consideration multiple selection criteria.

1 Introduction

The main characteristics of the next-generation all-IP mobile architectures can be foreseen by carefully considering the current trends. First of all, we notice a greater choice of access networks and simultaneous multi-access of these networks, including IEEE 802.11a/b/g WLAN, IEEE 802.15 WPAN (embracing Bluetooth), IMT-2000, IEEE 802.20 MBWAN, and so on. Then, there is an increasing number of multimedia communicators and mobile terminals with outstanding performances, such as smartphones, PDAs, tablet PCs, and laptops. Finally, we reckon that there will be a great demand of advanced yet simple to use mobile services comprising mobile commerce, adaptive and self-configuring services, context aware applications, user profiling and personalisation, etc.

This heterogeneous communication environment has already opened up new research areas, e.g. the Mobile IP and its micro-mobility suite, the Quality of Service ([1], [2]), or the Authentication, Authorization and Accounting. However, we believe that two basic requirements have already clearly emerged. The first one states that users should be provided with seamless roaming amongst various access networks (e.g. [3]), including simultaneous or successive connections to several access technologies. The second one mandates that users should

I. Niemegeers and S. Heemstra de Groot (Eds.): PWC 2004, LNCS 3260, pp. 421–430, 2004.

be allowed to always stay connected through the "best" access network (e.g. [4], [5]).

To achieve these goals we propose, design and implement an add-on middleware which is adaptable and reconfigurable. It is adaptable because the terminal always considers the current context (e.g., users' preferences, terminal resources, networks' conditions and applications' needs) and it tries to continuously adjust to the context when communicating. It is reconfigurable because the user or the network operator can redefine their preferences, subscribe to new services, add new configurations for the network interface cards, and so on.

As for adaptive applications we provide a clearly defined API towards them; we have also worked out a solution which handles the legacy applications, i.e. the applications which are unaware of our add-on middleware.

The rest of the paper is organised as follows. We will first present the related work in Section 2. Then will we describe our proposed architecture in Section 3. In the next section we will show the Profile Manager (PM) in detail and will look into our Selection Decision Algorithm (SDA) in Section 5. Section 6 will present the implementation and the results obtained so far. We will reveal the future work and conclude with Section 7.

2 Related Work

There is a growing number of research and standardisation efforts related to profile definition and handling. For example, in [6], and references therein, the notion of Generic User Profile (GUP) is defined as being a collection of data stored and managed by different entities such as the user equipment, the home or visited networks, and which affects the way the end-user experiences the different services offered. Then, the WAP User Agent Profile (UAProf), defined in [7], is concerned with capturing classes of device capabilities and preferences. The Composite Capability/Preference Profiles (CC/PP) framework (see [8]) is yet another mechanism for handling the preferences associated to users and user agents accessing the World Wide Web.

The Information Society Technologies (IST) AQUILA project tries to provide dynamic control to DiffServ based traffic; an objective of this project is to define and manage application profiles which contain the concrete application descriptions. Moreover, the IST-TRUST project tries to understand the users' requirements related to reconfigurable radio systems. It also defines a layered architecture which contains a policies and profiles management component. These profiles are further refined within IST-SCOUT project.

On the other hand, in the recent years, the interface selection problem for a multi-interface terminal, communicating in heterogeneous wireless environment has gained importance (e.g., [9], [10], [11]). However, to our knowledge, less work has been done to integrate the profile management with the optimal interface selection issue.

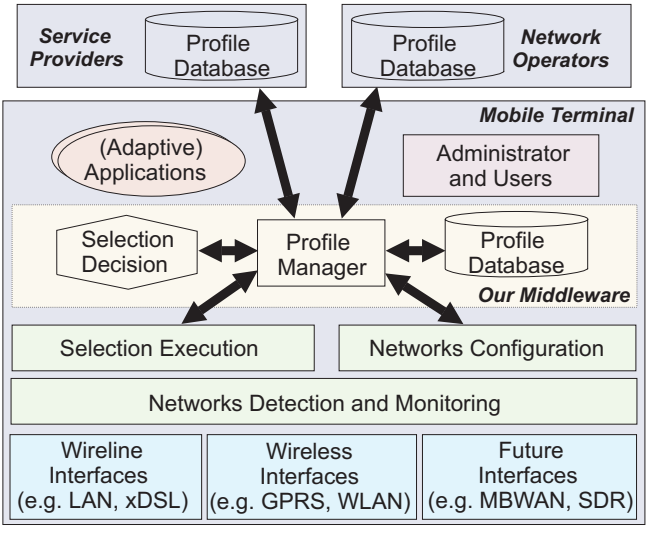

Fig. 1. Proposed architecture

3 The Proposed Middleware

Currently, after terminal start-up, the user surveys the communication and, if several access networks exist, the user decides which one to employ. To overcome this inconvenient, our approach relies on investigating the access networks capabilities and the applications limits and also interacting with the users in order to obtain their preferences.

We will gather in a structured way all these capabilities, requirements, and preferences in well-defined profiles. Then, the decision to use or not to use an access network will be based on these profiles. The final goal is to provide for each application flow the "best" access technology within multi-interface mobile terminals. Fig. 1 shows the envisaged mobile terminal architecture, its components and the possible interactions amongst them. Interfaces above our middleware collect the users, the operators and the service providers preferences and handle the applications' requirements. The layers below the middleware detect the available networks, provide real-time information about communication interfaces and access networks capabilities, or perform on-demand network interface configurations. Furthermore, a separate component handles the selection execution process, i.e., it actually maps the application flows on particular interfaces. Our middleware will control all these "low-level" layers by initiating network configurations and performing interface selection decisions.

The architecture is split in various functional bricks because this modular design facilitates implementation and testing and it also permits the gradual integration of better selection decision algorithms, of novel network detection and monitoring techniques (e.g. [12]), or of fine-grained selection execution modules (e.g. Per-Flow Movement).

Between our middleware, i.e., specifically the Profile Manager, and the external blocks we have used clearly defined interfaces. For example, as the Network Detection and Monitoring component needs to deal with various access technologies, it has to convert the collected information into a generic format which is then sent to the PM. In fact, it is this generic format that allows the comparison of the capabilities of various access networks. Furthermore, the Profile Manager needs to periodically inform the per-flow Selection Execution module about the preferred network interfaces to be used for communication. As for the "high-level" layers, the PM provides bi-directional interfaces towards them, i.e., users, applications, and service providers have to make their requirements and have to be informed about the changes within the system.

4 Profile Manager

An important part of the proposed terminal architecture will be dedicated to the definition and the management of profiles. Profiles are files stored in Profile Databases (PDB) and they summarise key information about the components of the system and its interactions with the environment, i.e., users, applications, access networks, or service providers. Specifically, the profile handling mechanism serves the following purposes:
– triggers and assists the SDA when it makes the choice of the "best" access;
– automatizes the selection of an access network by maintaining all the necessary information for proper interface configuration;
– sets forth a solution which works both for adaptive applications and for unaware applications.

We propose three kinds of profiles within the Profile Databases: generic, specific and active profiles. The generic profiles describe what information could be stored in the various types of profiles, i.e., they can be seen as patterns or schemas. We consider four generic profile types within our Profile Databases:

1. Preferences and Resources Profile (PRP): it has been noticed that one's preferences depend on the currently existing resources or the present situations. Thus, the generic PRP specifies how the system should behave based on the available resources or the current context. The preference parameters considered here are, e.g., selection mode, selection goal, and preferred and forbidden access networks. The system itself may provide the information about the current context, e.g., battery status, geographical location, or subscription type.
2. Flow Description Profile (FDP): the first part of this profile holds the application's QoS requirements (e.g., service class, minimum necessary bit rate, typical delay expected, maximum delay variation) and the second part contains the QoS monitored by the system (e.g., mean bit rate, bit error rate, average latency). We also propose default parameter values for some of the common applications.
3. Network Interface Profile (NIP): it comprises network interface card parameters that can be obtained from technical specifications (e.g., maximum

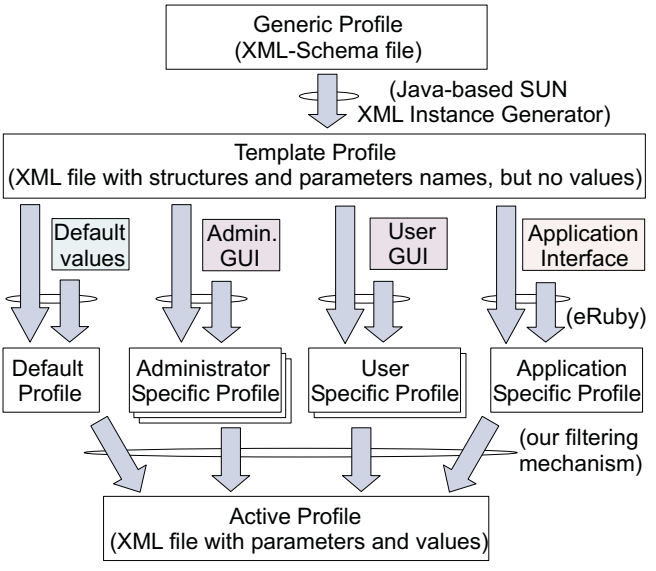

Fig. 2. Obtaining the active profile

theoretical throughput supported), or throughout offline statistical measurements (e.g., maximum real throughput).

4. Access Network Profile (ANP): it specifies all the necessary information, for both Layer 2 and Layer 3, required to successfully configure and use an access network. The Layer 2 part of the ANP contains the mandatory parameters required to associate with the network, the authentication and encryption protocols, the QoS settings and the billing and charging parameters. The Layer 3 part comprises the IPv6 address configuration if needed so, the Mobile IPv6 settings, and various tunnelling configurations. The accounting information from the ANP is intentionally simplified for the time being (i.e., a flat cost model is used), and it is globally defined for an access network, regardless of the traffic type (i.e., business or leisure).

Likewise, the administrators, the network operators, the users and the applications can instantiate a generic profile (i.e., automatically provided) to build specific profiles corresponding to specific cases (e.g. Jean/Player/802.11/Office).

Finally, the third kind of profile is the active profile which is obtained by filtering different specific profiles of the same type. The profile handling mechanism described above is depicted in Fig. 2. Because the administrator (or network operators), the users and the applications (or service provides) can have their own specific profiles, we need to filter somehow the various values defined for the same parameter in different specific profiles of the same type. The following rules apply to the parameter values when an active profile is inferred:
– all parameters have a value type, i.e., mandatory or proposed, and the mandatory value for a parameter has always priority over the proposed value; – the

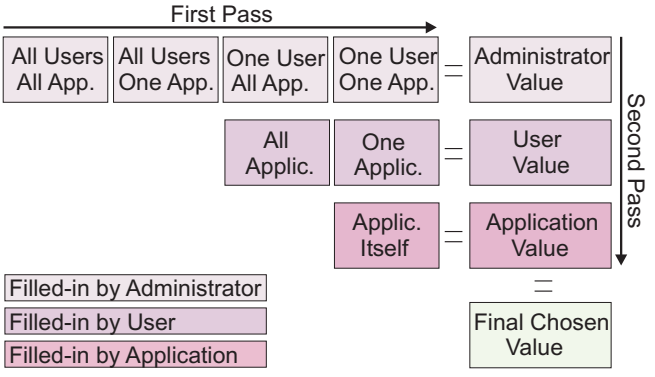

Fig. 3. Two-pass filtering mechanism

priority for a proposed parameter value increases from administrator to user, then to application;
– if there are mandatory parameter values, the priority decreases from administrator to user, then to application;
– some of the parameters may obey to the mutual exclusion rule, e.g., if the administrator sets a forbidden network for a user, the user can set the same access network to "preferred", but its preference is ignored in this case.

In Preferences and Resources Profile case we need to further extend the process to a 2-pass filtering mechanism. This is necessary as the administrator (or e.g., the network operator or the service provider) needs to designate its preferences on a more fine-grained basis: she or he can define a specific PRP for all users and all applications, for a user and all applications, for all users and an application, or for a user and an application. Furthermore, the users themselves can define their preferences as being common for all applications they use, or just for a particular application.

Thus, for each parameter within specific PRPs, we need to construct a kind of hierarchy when an active PRP needs to be obtained, as shown in Fig. 3. Then, we perform the first pass and construct the input for the second pass (i.e., the last column) by picking up the last proposed parameter value or the first mandatory parameter value on each line. Next, we execute the second pass and we chose as final value the last proposed parameter value or the first mandatory parameter value found in the last column.

The information stored within all profiles is managed in a uniform and extensible manner using the XML paradigm. Furthermore, as depicted in Fig. 1, the profiles may be distributed amongst different entities and thus, e.g., a SOAP or an XML-RPC protocol will be required to handle them. However, it can be pointed out that the Profile Manager acts as a dispatcher within our architecture: it interacts with all the entities which supply the profiles, it knows which information needs to store in the Profile Database, and it implements the filtering mechanism. Moreover, after updating the PDB, the PM determines if

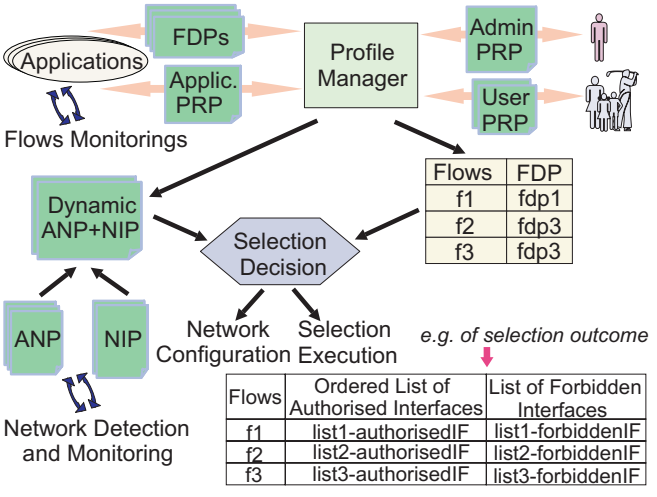

Fig. 4. Profile interdependencies

the SDA should be triggered or not. Fig. 4 shows the relations amongst various profiles defined within our framework and the selection algorithm outcome.

5 Selection Decision Algorithm

Most of the interface selection algorithms or the current handover algorithms take into consideration just one selection criterion, usually the Received Signal Strength (RSS) or the Signal-to-Noise Ratio (SNR). More recently the multiple selection criteria algorithms have emerged as a better alternative.

The Profile Manager determines if the Selection Decision Algorithm needs to be informed or not about the changes within the terminal. Thus, the SDA does not need to know how the parameters are collected within profiles or how the selection decisions are enforced.

The PM must restrain the plethora of various triggers, otherwise the SDA could be activated too often and it will exhaust the CPU or the battery. Based on the triggers received from the Profile Manager, the SDA interrogates the Profile Database and starts its computational procedure. Thus, various active profiles (i.e., the ones resulted after the filtering process) are used as input by the SDA in order to select the "best" interface for each application flow. The procedure used by the SDA is to define access network score functions and application utility functions which are both maximized. The score is calculated for each network i and the utility for each network-flow i, j 2-tuple. Then, to solve this multiple-goal problem, we use the weighting objectives method.

$$Score(i) = \sum_{i=1}^{N} (w_i \cdot \|monitoredValue_i\|); \text{ where } \|\| \text{ is the } ln \text{ or } tanh \text{ function}$$

$$Utility(i,j) = \sum_{i=1}^{N} (\|monVal_i - minNecVal_j\|) \cdot (monVal_i \otimes minNecVal_j);$$

where $X \otimes Y = 1$ if $X \geq Y$ or 0 otherwise

The suggestion to employ or not an access network is made on a flow-per-flow basis, but the SDA can also propose only one interface for all flows. Furthermore, in order to cope with various constraints and to satisfy the users and applications requirements, the Selection Decision Algorithm could decide that it is better to (re)map some of the existing flows on other interfaces.

The current selection algorithm provides possible mappings only for the outgoing flows and it makes no assumption about the incoming flows. Yet, a distributed SDA could also suggests, together with the correspondent nodes or the networks, a global flow mapping which considers the incoming flows as well.

The SDA outcome, as shown in Fig. 4, consists of two lists of interfaces for each application flow: the ordered list of preferred interfaces and the list of forbidden interfaces. The SDA only offers middle term (i.e., hundreds of milliseconds) handover decisions. Nevertheless, our framework supports an extensible interface with the per-flow Selection Execution component, which actually maps the flows and makes short-term (tens of ms) adaptations when needed. This happens, e.g., when RSS/SNR drops below the communication sustainable limit and the Selection Execution module immediately re-maps the flows on the next preferred interface from the list provided by the SDA.

6 Implementation and Results

To implement and test the proposed architecture we have chosen two terminals: IPaq 3970 with Familiar Linux, and Dell Latitude C610 running Debian Linux. To support the L3 mobility we installed the Mobile IPv6 for Linux distribution (i.e., MIPL). As wireless access technologies, we have 802.11b and Bluetooth access points and a commercially available GPRS network.

We prefer to store all the profiles within the local Profile Database for the time being. Nonetheless, we implement all four types of profiles using XML and XML Schemas. The handling and filtering of profiles within the Profile Manager and PDB, and the Selection Decision Algorithm are implemented using Ruby1.8, which is a portable, lightweight, object-oriented scripting language.

The inter-module communication within our middleware is done through Distributed Ruby, and with external modules through an XML-RPC-like protocol. To detect and configure the network interfaces we employ bash scripts.

We use the following selection initiation triggers: interface ready, interface not ready, preferences and resources changed, flow created, and flow deleted. As decision criteria, we use from active PRP the list of forbidden access networks for user and application, the monetary cost vs. QoS goal parameter, the required security level for application, and the battery status. From ANPs we take the monitored bit rate, the average bit error rate, the cost per byte and the security level of the access network. Finally, from NIPs we obtain the theoretical bit rate,

and from FDPs we use the minimum necessary bit rate, the supported bit error rate and the maximum delay. Fig. 5 illustrates two of the most common use-cases, application launch and, respectively, interface status change (either the interface loses the connection with the network, or it gets associated).

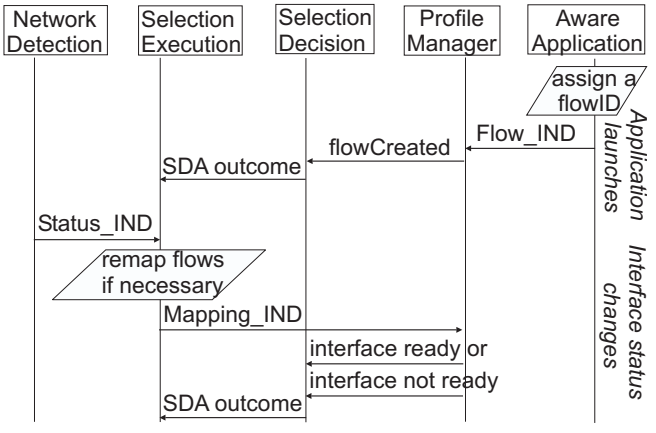

Fig. 5. Two common use-cases

The results obtained so far are encouraging. The selection latency (i.e., starting from selection initiation trigger till the phase when decision to (re)map a flow was taken) depends on trigger type, and it takes 2-3 seconds on the PDA and under a second on the laptop. Yet, we need to do a thoroughly evaluation of our architecture by taking into account, e.g., the selection cost gain (in general the monetary cost gain, but the aggregate bandwidth gain or security gain should not be overlooked), the number of flow re-mapped in similar conditions and during a well-defined period of time, the number of cut off flows, and so on.

However, as we already have a full functioning prototype, it is quite easy to update the profiles with new parameters and to test new selection algorithms.

7 Conclusions and Future Work

In this paper we have presented an adaptable and reconfigurable architecture for the mobile terminals supporting multiple access network interfaces. Our first ambition was to provide seamless access over heterogeneous networks, including simultaneous or successive connections to several access technologies. The second goal was to allow the user to always stay connected through the "best" access network. The proposed architecture includes adaptation mechanisms and relies on tight interactions amongst the different layers, from the application layer to the data link layer.

Thus, we argue that future multi-criteria handover algorithms will be more complex than nowadays, and they need to be divided into two parts: a contributory middle-term handover algorithm (i.e., the one which uses various profiles, as presented in this paper) and an essential short-term handover algorithm (i.e., the one traditionally based on RSS or SNR).

Our on-going work focuses on further refinement of the profiles, e.g., uniform monetary cost representation. In addition, more selection strategies need to be investigated and the most promising of them will be implemented and evaluated. Specifically, in our opinion, three areas should be thoroughly examined: initiation triggers (i.e., when to start the selection algorithm), decision criteria (i.e., which parameters to collect and use), and selection algorithms (i.e., how to combine the chosen parameters). In addition, more tests are needed in order to grasp all the benefits of our terminal architecture.

References

1. Fu, X., Karl, H., Kappler, C.: QoS-conditionalized handoff for Mobile IPv6. In E. Gregori et al (Eds.): Networking 2002, LNCS 2345, May 2002
2. Wisely D., et al.: Transparent IP radio access for next-generation mobile networks. IEEE Wireless Communications, August 2003
3. Zahariadis T.: Trends in the path to 4G. Communications Engineer, February 2003
4. Gustafsson E., Jonsson A.: Always best connected. IEEE Wireless Communications, February 2003
5. O'Droma M., et al.: "Always best connected" enabled 4G wireless world. IST Mobile and Wireless Communications Summit 2003, June 2003
6. 3GPP TS 24.241 V0.4.0: 3GPP generic user profile (GUP) common objects; Stage 3. 3GPP, August 2003
7. WAP 248-UAPROF V20: User agent profiling specification. WAP Forum, October 2001
8. W3C: Composite capabilities/preference profiles (CC/PP): structure and vocabularies. W3C Working Draft, March 2003
9. Park T., Dadej A.: Adaptive handover control in IP-based mobility networks. Proc. of 1st Workshop on the Internet, Telecommunications and Signal Processing (WITSP'02), December 2002
10. Zhang W., Jaehnert J., Dolzer K.: Design and evaluation of a handover decision strategy for 4th generation mobile networks. 57th Semiannual Vehicular Technology Conference VTC 2003-Spring, April 2003
11. F. André et al.: Optimised support of multiple wireless interfaces within an IPv6 end-terminal. Smart Objects Conference, May 2003
12. F. Adrangi (Ed.): Network discovery and selection within the EAP framework. Work in progress, Internet Engineering Task Force, draft-adrangi-eap-network-discovery-and-selection-00.txt, October 2003