Prototyping for End-to-End Reconfigurable Equipment

A. Katidiotis, D. Petromanolakis, V. Stavroulaki, P. Demestichas(1)
E. Patouni, A. Kousaridas, P. Magdalinos, N. Andriopoulos(2)
K. El-Khazen, S. Buljore, P. Roux, C. Beaujean (3)
T. Farnham (4)
D. Nussbaum (5)
A. Kountouris, F. Marx(6)

(1) University of Piraeus, Department of Technology Education and Digital Systems, Greece, e-mail: katidiot@unipi.gr
(2) Communication Networks Laboratory, Department of Informatics & Telecommunications, University of Athens, Greece
e-mail: elenip@di.uoa.gr
(3) Motorola Labs, European Communications Research, Gif-sur-Yvette, France, e-mail : karim@motorola.com
(4) Toshiba Research Europe Ltd, Bristol, UK. email: Tim.Farnham@toshiba-trel.com
(5) Eurecom, France, email : Dominique.Nussbaum@eurecom.fr
(5) France Telecom (FTR&D), France, email : apostolos.kountouris@francetelecom.com

Abstract— In the current wireless telecommunications world, various heterogeneous access technologies can co-exist in a diversified infrastructure, forming the so-called Composite Radio Environment. Reconfigurability, aims at bringing the full benefits of the valuable diversity within the Composite Radio Environment, by devising reconfigurable devices and supporting system functions to offer an expanded set of operational choices to all business players in the context of heterogeneous mobile radio systems, i.e. users, applications and service providers, operators and regulators. It is obvious that reconfigurability poses new requirements on the management of wireless systems. More specifically, from the equipment point of view, in order to be able to shift across heterogeneous wireless access technologies, dynamically adapting to the optimum RAT, enhanced management and control capabilities are required. In this direction, part of the work in the IST project End-to-End Reconfigurability (E²R) has focused on reconfiguration issues local to the equipment. In order to obtain proof of concept and measurements demonstrating the effectiveness and efficiency of the equipment reconfiguration management and control several individual prototyping activities have taken place. This paper focuses on prototyping activities conducted in the area of equipment reconfiguration management.

***Index Terms*— End-to-End Reconfigurability, Reconfigurable Equipment, Prototype platform**

I. INTRODUCTION

In the current wireless telecommunications world, various heterogeneous access technologies can co-exist in a diversified infrastructure, forming the so-called Composite Radio environment. In Composite Radio systems, different radio networks, such as GPRS, UMTS, WLAN, WMAN, DVB, etc, can be co-operating components of a heterogeneous wireless infrastructure, increasing the efficiency of service provision and the exploitation of available Radio Access Technologies (RATs). Through such Composite Radio systems, users can be directed to alternate RATs, according to service area regions, time zones, profile and network performance criteria. Reconfigurability, aims at bringing the full benefits of the valuable diversity within the Composite Radio Environment, by devising reconfigurable devices and supporting system functions to offer an expanded set of operational choices to all business players in the context of heterogeneous mobile radio systems, i.e. users, applications and service providers, operators and regulators. It is obvious that reconfigurability poses new requirements on the management of wireless systems. More specifically, from the equipment point of view, in order to be able to shift across heterogeneous wireless access technologies, dynamically adapting to the optimum RAT, enhanced management and control capabilities are required. In this direction, part of the work in the IST project E²R [1] has focused on reconfiguration issues local to the equipment. More specifically, a reconfiguration management framework considering architecture, interfaces and functionality, has been developed, and the extensions for execution environments (capable of hosting the reconfiguration management functionality and the flexible protocol stacks) including the development of reliability/security mechanisms for the equipment management have been realised.

In order to obtain proof of concept and measurements demonstrating the effectiveness and efficiency of the equipment reconfiguration management and control several individual prototyping activities have taken place. This paper focuses on prototyping activities conducted in the area of equipment reconfiguration management.

II. EQUIPMENT MANAGEMENT ARCHITECTURE FOR END-TO-END RECONFIGURABILITY

In general, the management of equipment in end-to-end reconfigurable networks should satisfy certain requirements to ensure an efficient, reliable and secure equipment operation: (i) Monitoring and discovering the available networks. (ii) Negotiating offers with the various available networks and selecting the most appropriate network/RAT. (iii) Providing support for different protocols and protocol features. (iv) Providing appropriate mechanisms for the validation, fault diagnosing as well as error recovery procedures. (v) Control of equipment components to coordinate the realisation of a reconfiguration that requires modifications of various equipment components. (vi) Interworking with corresponding reconfiguration entities on the network side. In order to address flexible, reconfigurable equipment, it is important to enable both the management and control of the functions introduced by the reconfiguration processes. The main functional entities of the equipment management and control architecture are:

- The *Configuration Management Module* (CMM), which manages the reconfiguration processes according to specified semantics, protocols and configuration data model, and
- The *Configuration Control Module* (CCM), which is responsible for the control and supervision of the reconfiguration execution.

These main modules are further split into several sub-modules that cover specific tasks so as to meet the requirements identified in the previous. An overview of the equipment management and control architecture [2], in terms of involved sub-modules is depicted in Fig. 1. As was introduced in the previous several individual prototyping activities have taken place in the area of equipment reconfiguration management. These are described in detail in the following sections.

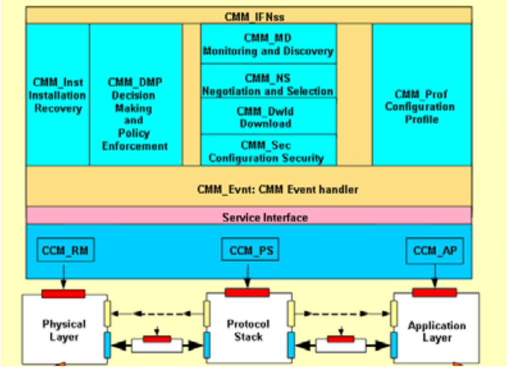
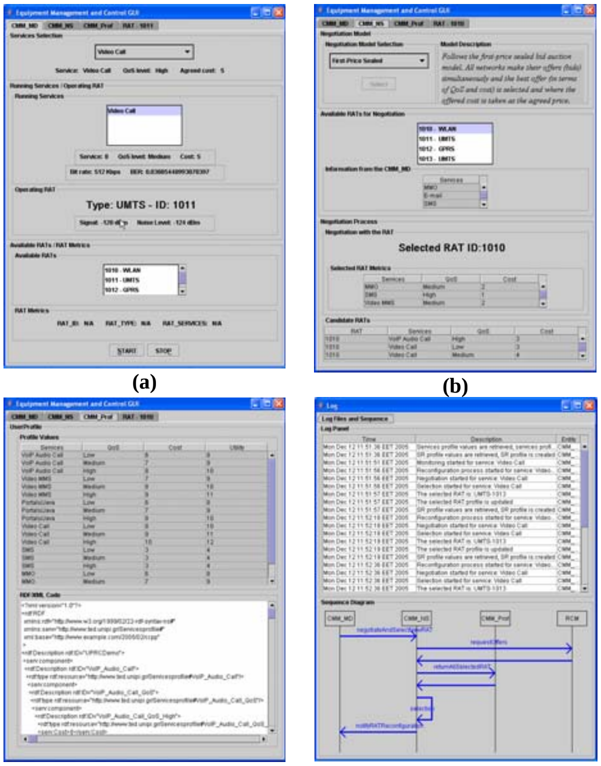


Fig. 1. Equipment Management and Control Architecture

III. CONFIGURATION MANAGEMENT FOR ENABLING THE REALIZATION OF THE SEAMLESS EXPERIENCE

The contribution of this prototyping activity falls within the prototype development of a Management and Control System for Reconfigurable Equipment (MCS-RE) enabling the realization of the seamless experience [3]. A generic MCS-RE, targeted to the requirements identified in the previous, consists of five main modules (corresponding to a subset of the entities of the Equipment Management and Control Architecture). The main modules of the MCS-RE prototype are:

- The *Profiles* module which maintains and provides user and equipment profile information.
- The *Monitoring and Discovery* module whose role is to monitor the environment, identify the available networks in a certain area and monitor their status.
- The *Negotiation and Selection* module which enables the interaction of the equipment with the various networks/RATs of the composite radio infrastructure and is responsible of negotiating (exchanging) offers and selecting the most appropriate reconfiguration action in terms of cost and provided QoS level.
- The *Reconfiguration Implementation Procedures* module that provides all functionality required to implement the actual reconfiguration pattern that has been decided by the *Negotiation and Selection* module.
- The *Configuration Control* module is responsible for the control and supervision of the reconfiguration execution.
- The *Interface with the network support entities* for the exploitation of the interface with the network entities.


(a) (b)

(c) (d)

Fig. 2. MCS-RE GUI: (a) Monitoring and Discovery; (b) Negotiation and Selection; (c) Profiles; (d) Log

The prototype software development is made on a Windows XP operating system and is based on several technologies, including Java, RDF/XML, Xerces [4] (XML Parser), ARP [5] (RDF Parser), Jena [6] (API for creating and updating RDF/XML documents) and JSR 188 [7] (an API provided by SUN for parsing RDF/XML documents using Jena). The

MCS-RE prototype implementation provides a Graphical User Interface (GUI) which offers several actions and options of visualization, such as visualization of profile related information, selecting and starting different services, supervision of running services, information on the status of the current connection and information on the negotiation process. The MCS-RE GUI is divided in four main parts (tabs): the *Monitoring and Discovery* part, the *Negotiation and Selection* part, the *Profiles* part, and the part providing profile information related to the currently selected RAT.

IV. RECONFIGURATION MANAGEMENT FOR ADAPTABLE PROTOCOLS

The contribution of this prototyping activity lies in specifying generic reconfiguration management functionality that will trigger protocol layer reconfiguration. In the context of this prototyping activity, special focus was paid on the Reconfiguration Profiles and Protocol Layer Reconfiguration, which is triggered by advanced policy-based decision making mechanisms; the latter are out of scope for this paper.

A. Reconfiguration Profiles

This section briefly describes the implementation design of “CMM Profiles” (CMM_Prof) entity that is responsible for profile retrieval, profile representation and profile update in the presented prototype architecture. The profile is represented through the Composite Capabilities/Preference Profiles [8] and the contextual information it carries is organized in a Component – Property basis. The Open Mobile Alliance’s User Agent Profile schema ([9]) has been considered as a base. As OMA’s schema primarily focuses on static information, the so called Reconfigurability Schema has been drafted including reconfigurability-related information. This vocabulary extends state-of-art standards, encompassing aspects concerning the operating system, radio access technologies, device software and hardware and identifying specific profile properties for each one of them. The aforementioned extensions can appear either through the definition of one more profile class (reconfiguration profile) or through the addition of parameters to the profile classes already in use. In this prototype the new schema defines six (6) CC/PP components, namely ExecutionEnvironment, HWExecutionEnvironment, RadioAccess, Operating System, InstalledProtocolComponent, and ApplicationProfile. Further, specific properties have been identified and related to the presented components. The final form of the reconfiguration profile is still an open issue. The reconfigurability schema is flexible enough to support each of the presented alternatives. Fig.3 depicts a sample reconfiguration profile as an instance of the reconfigurability schema.

B. Protocol Layer Reconfiguration

This section elaborates on protocol layer reconfiguration concepts. In particular the basic aspects related to this part of this prototyping activity are the following: a) Dynamic fragmentation and reassembly of the existing protocol layers in discrete protocol components, b) dynamic plug-in and replacement of protocol components and c) introduction of management functionality that will control the establishment of the proposed composition, verify its correctness and realize the protocol reconfiguration. To the next step, the above analyzed concepts were mapped to the E²R terminal architecture. In particular, the generic protocol reconfiguration/adaptation management functionality was incorporated in the “CCM Protocol Stack” (CCM_PS) module whereas the intra-layer protocol reconfiguration management is included in the so-called Configuration Binding Control Module (CBCM) [10]. Triggered by the policy-based and decision making mechanisms, an update for a protocol component may occur. The CCM_PS is notified about the forthcoming transformation of the protocol stack. Then it dispatches this notification to the appropriate CBCM that controls the protocol layer that is going to be reconfigured. The CBCM incorporates the necessary mechanisms that guarantee the seamless reconfiguration. The aforementioned procedures are realized through a generic framework for protocol reconfiguration utilizing a queue-based scheme and preservation and application of state information to guarantee operation continuity. With respect to the validation and proof-of concept of the presented protocol layer reconfiguration aspects, a case study was considered for the FTP protocol. The details of this experiment coupled with some first experiences from preliminary performance assessment are further discussed in [10].

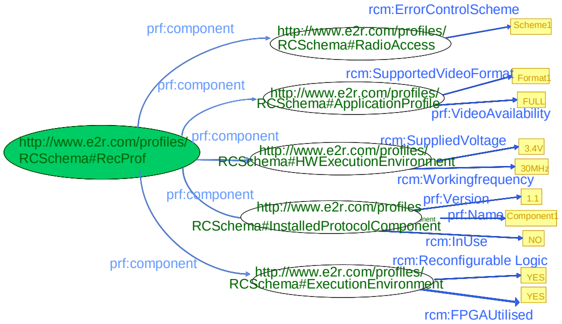


Fig. 3. Sample reconfiguration profile

V. SECURE DEVICE MANAGEMENT

Advanced security features, in a context where several stakeholders interact in the management of reconfigurable device, is still a challenge for wireless technology stakeholders. The deployment of SW from manufacturer, service and third party providers play an important role in future mobile devices to upgrade the equipment or to tune it to personal needs. The main focus of this prototyping activity (Fig. 4) has been the verification of certificates and checking of integrity of the downloaded software and the demonstration of secure device management features. Key issues addressed are secure transfer of data from the server to the equipment protecting from eavesdropping and other attacks, the authorisation of the user/equipment before availing any service and the secure connection between equipment and network.

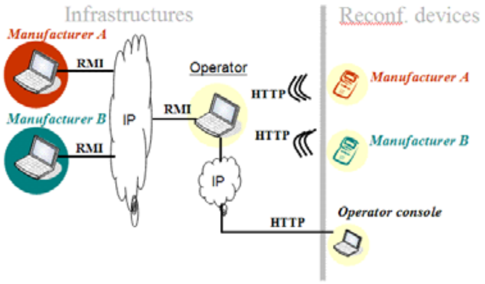


Fig. 4. Secure device management prototype architecture

VI. TCP ADAPTATION

Going into more specific reconfiguration enablers TCP adaptation is addressed. The objective was to reduce the sending rate (with the Slow Start and Congestion Avoidance) after a packet loss less aggressive than with a standard TCP. This allows the improvement of the TCP performances by tuning the protocol parameters with a minimal impact on the overall transport layer behaviour. It is shown that this can be done by tuning: the congestion window: bypass the very first steps of the Slow Start and start sending data at a higher rate than just 1, 2 or 4 segments at a time (this is a kind of "fast start" of the TCP sessions). This is demonstrated under a Linux Mandrake 10.1, kernel 2.6.8.1 for both IPv4 & v6, framework. Finally, evidence on how the TCP Adaptation system can be interfaced and integrated with the other components of the overall equipment management architecture.

VII. SECURE, RELIABLE AND OPTIMISED RECONFIGURATION ENVIRONMENT (SERP)

Secure and reliable (cross-layer) optimisation of equipment reconfiguration (Fig. 5) is the goal of the work conducted in this prototyping area. The architecture for the optimisation consists of Network and Local Optimisation Agent, (NOA and LOA), and the process is performed in a hierarchical manner. The optimisation addresses the protocol stack and execution environment configuration and in particular considers the trade-off between performance and power-consumption. The process of optimisation consist of the NOA computing constraints, given the options and preferences and then the LOA computing the best mode (or modes) of operation based on the constraints imposed by the NOA. A feedback loop is included from the protocol stack and execution environment monitors back to the LOA and NOA to permit continual refinement. Performance results are presented showing the types of trade-off that can be performed with a generic Medium Access Control (MAC) protocol component and configurable execution environments. In particular the results indicate that no single execution environment or protocol stack configuration is suitable for all performance and power consumption requirements. The prototype therefore considers two optimisation objectives, one in which low power consumption is required and the other in which high performance is required. The modes of operation are reconfigured accordingly to meet the objectives. The reconfigurable security mechanisms considered in this section permit the configuration of cryptographic functions in a generic manner. This permits exploitation of accelerated cryptographic implementations when necessary. The API can also support security context management to allow multiple contexts (for instance between different operators) to co-exist on the same device. This prototype currently supports a wide range of cryptographic algorithms in a software implementation. The algorithms can be selected and cryptographic operations performed on demand.

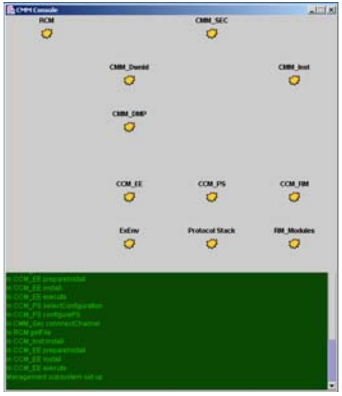


Fig. 5. SERP prototype

VIII. EQUIPMENT POWER MANAGEMENT

This prototyping activity focused on exploring the possibilities to enhance equipment power management by means of run-time management of the transceiver processing complexity. Assuming that there is a direct relation between processing complexity and power consumption it is conceivable that management of the processing complexity coupled to other power management techniques could potentially contribute in minimizing the power consumption on average and thus extend battery time and terminal autonomy. The object of this work is the receiver in the downlink of a 3G UMTS TDD mobile terminal. A smart receiver is considered the structure of which adapts itself to the environment. The goal is to minimize computational complexity and hence power consumption, while satisfying certain performance requirements. The main environmental characteristic considered here is the channel impulse response. The variable parts of the receiver considered is the channel delay spread and the choice of either a channel matched filter or a linear equalizer. The goal is to test the smart receiver on a wireless platform, in conjunction with the specification of the UMTS TDD standard and the platform implementation. This leads to a number of preferred implementation choices. Such issues have an impact on the options to be considered for the development of the smart receiver.

1) *Channel estimation*

In many systems, (e.g. UMTS-TDD), channel estimation is carried out in the frequency domain in principle, involving FFTs (Fast Fourier Transforms). The channel estimate is obtained in the frequency domain by simple division of the FFT of the received signal and the (stored) FFT of the training sequence. The channel estimate in the time domain is obtained by performing inverse FFT and windowing the result to the expected delay spread. The positioning of this window corresponds to a synchronization operation. It is not necessary to continuously leave the synchronization uncertainty to the complete Inverse FFT (IFFT) duration, so for most of the frames the IFFT is computed over a reduced time span, creating some room for complexity reduction via pruned FFTs.

2) *Equalization*

In the case of a non-trivial spreading factor, the receiver implementation computes the convolution of spreading code and channel impulse response and then correlates at symbol rate with this cascade. So reception is performed in the time domain. The equalizer to be considered here is a linear equalizer. If the channel impulse response is represented as an (Finite Impulse Response) FIR filter (and not sparse as in the RAKE) and the (chip) equalizer would be an FIR filter of the same length, then the complexity of performing the reception would be the same for a channel matched filter (MF) or an equalizer. The difference in complexity would be due to the computation of the equalizer (computation of the receiver, as opposed to the operation of reception itself). If the equalizer filter is longer than the channel impulse response filter, then reception with an equalizer becomes more complex than reception with a channel MF. The relative complexity difference is more pronounced if no spreading is used. The use of a longer equalizer is often desirable when the channel length is shorter. The computation of the equalizer is an important issue.

3) *Switching criteria*

Ideally, the performance is a function of the SINR (Signal to Interference and Noise Ratio) at the output of the receiver filter. The SINR is also used as criterion to adjust the power control. Even though the power control will tend to force the SINR to a constant, the SINR can still be used as a switching criterion between channel MF and equalizer since the SINR of the Minimum Mean Squares Error (MMSE) equalizer will in principle always be better than that of the channel MF for the same filter length. Now, the actual desired result is that a certain SINR gets attained, with a minimal complexity, and for a limited transmit power requested from the base station (BS). So the proper optimization criterion will be to minimize computational complexity (and hence terminal power consumption) subject to minimal SINR and maximal BS transmitted power constraints. An important issue is that the computation of the SINR for a certain receiver structure may get quite complex, involving among other things the computation of that receiver structure. It would be meaningless to put in place a switching mechanism to switch between receivers in order to minimize complexity, when the complexity of the switch would be comparable to the reduction in complexity.

4) *Implementation*

The equipment power management functionality has been integrated in a real time flexible UMTS protocol stack. This protocol stack has been developed mainly in C and runs in real time in PC-based platforms thanks to Real-Time Linux [11]. Concretely, four main blocks are developed in the PC-based platform that will enable integration interactions: (i) The channel estimation block code that is developed in "C" and has basically the same prototype as the classical UMTS-TDD channel estimation bloc code. (ii) The equalizer block code is also developed in "C", as the previous equalization block. (iii) The switching criterion block code. (iv) The GUI block code. These blocks have been delivered under the General Public Licence. Moreover, a GUI is under development to illustrate and validate the software. It depicts in a window the BLER (Block Error Rate), the SINR, the window length used for channel estimation, the state of the equalizer, the switching criteria and the complexity of the signal processing in terms of MIPS.

IX. CONCLUSIONS

Part of the work in the IST project E²R has focused on reconfiguration issues local to the equipment. In order to obtain proof of concept and measurements demonstrating the effectiveness and efficiency of the equipment reconfiguration management and control several individual prototyping activities have taken place. This paper presented an outline of the prototyping activities conducted in the area of equipment reconfiguration management in the context of E²R. The prototyping activities will continue and it is envisaged that the corresponding prototypes will be used for further development and evaluation of necessary improvements and will be extended with physical layer reconfiguration capabilities and network support services. Finally, the individual prototypes will be upgraded and will be integrated into the overall E2R proof of concept environment.

ACKNOWLEDGMENT

This work has been performed in the framework of the EU funded project End-to-End Reconfigurability (E²R). The authors would like to acknowledge the contributions of their colleagues from the E²R consortium.

REFERENCES

[1] IST project End-to-End Reconfigurability (E²R), www.e2r.motlabs.com
[2] E²R White Paper: Equipment Management and Control Architecture (July 2005), http://e2r.motlabs.com/whitepapers
[3] V. Stavroulaki, A. Katidiotis, D. Petromanolakis, P. Demestichas, “Equipment Management Strategies in Reconfigurable Networks”, accepted for IEEE VTC2006-Spring, Melbourne, Australia
[4] Xerces Java Parser, http://xml.apache.org/xerces-j/index.html
[5] ARP, http://www.hpl.hp.com/personal/jjc/arp/
[6] JENA, http://jena.sourceforge.net/
[7] JSR 188 CC/PP Processing, http://www.jcp.org/en/jsr/detail?id=188
[8] Composite Capability/Preference Profiles (CC/PP):, http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/
[9] OMA User Agent CC/PP Schema, http://www.openmobilealliance.org/tech/profiles/uaprof/ccppschema-20030226
[10] E. Patouni, N. Alonistioti, and P. Magdalinos, “A Framework for Protocol Reconfiguration”, in the Proceedings of the Seventh IFIP International Conference on Mobile and Wireless Communication Networks, Marrakech, Morocco, Sept. 2005.
[11] FSMlabs Website, http://www.fsmlabs.com