Wireless Pers Commun
DOI 10.1007/s11277-009-9903-0

Towards the Functional Enhancement of 3GPP Networks with Reconfiguration Capacities

Vangelis Gazis · Zachos Boufidis · Nancy Alonistioti · Lazaros Merakos

© Springer Science+Business Media, LLC. 2009

Abstract Over the last decade, several wireless access systems, such as broadband WLAN and WMAN (e.g., IEEE 802.11 and 802.16), broadcast systems (e.g., DAB, DVB-T) and short-range connectivity systems (e.g., Bluetooth, UWB), have emerged to complement existing second (2G) and third (3G) generation cellular systems. The common objective is to migrate to a flexible connectivity platform that will support the disparate requirements of mobile applications in the next-generation era, commonly referred to as 4G. To achieve a versatile multi-radio and multi-protocol access from mobile devices of limited resources in 4G, the original software-radio concept has evolved into across-the-board reconfiguration, which includes not only radio-specific functionalities but supports also the dynamic modification of the entire protocol stack. The present article clarifies the fundamental aspects of reconfiguration and points out the shortcomings of current standards for describing the reconfiguration capabilities (i.e., metadata about the feasible alternative configurations) of mobile network elements. An object-oriented model for reconfiguration metadata is introduced and isomorphically mapped to an RDF vocabulary, which is used to describe reconfigurable protocol stacks as defined for UMTS/WLAN mobile devices. The associated description languages and formats are presented and evaluated with regard to their applicability and suitability for this task and the criteria for choosing one are identified and detailed. Next, the article proposes a novel generic modular architecture for reconfiguration support in 4G mobile networks and elaborates on its functional components. We investigate the integration of our logical architecture in the 3GPP mobile network architecture and illustrate the

V. Gazis (✉) · Z. Boufidis · N. Alonistioti · L. Merakos
Communication Networks Laboratory (CNL), Department of Informatics & Telecommunications,
University of Athens, Panepistimiopolis, 15784 Athens, Ilisia, Greece
e-mail: gazis@di.uoa.gr

Z. Boufidis
e-mail: boufidis@di.uoa.gr

N. Alonistioti
e-mail: nancy@di.uoa.gr

L. Merakos
e-mail: merakos@di.uoa.gr

Published online: 01 January 2010

allocation of its functional elements over the logical domains of 3GPP system Release 6 and onwards. Furthermore, we show the signaling between terminal-side and network-side architecture components for software download and protocol stack reconfiguration. Finally, we summarize our experience gained from a prototype implementation, evaluate the total signaling cost, and give directions for future work.

Keywords 4G · Metadata · Mobile · Protocol stack · Reconfiguration · Signaling

1 Introduction

At the dawn of the twenty first century, wireless mobile communication technologies show a remarkable development. Given that the observed proliferation of wireless technologies (e.g., GSM/GPRS/UMTS, HSDPA, IEEE 802.11a/b/g/n, IEEE 802.16a/d/e, Bluetooth, IEEE 802.15.3a/4, DVB-T/S, etc.) is likely to persist, thus requiring support for multiple dissimilar wireless standards in future devices, the industry's focus has included reconfiguration as a technological enabler of multi-standard wireless systems. Reconfiguration manifests in various forms, including the over-the-air download and activation of protocol software during runtime, the modification of operational parameters through an open Application Programming Interface (API), and profiles that describe the different operation modes of wireless devices (e.g., for switching to another Radio Access Technology—RAT) [1]. Use cases of reconfiguration include the dynamic modification of the protocol subsystem (e.g., the insertion of additional protocol stacks) in both stationary (e.g., radio base stations) and mobile (e.g., handheld user devices) operating network elements.

The work presented herein proposes and introduces a novel reconfiguration architecture based on the semantic description of the options available to the reconfiguration process with regard to the system's synthesis. The objective of this layer of semantic information is to support the discovery of reconfiguration options that realize the desired function in a consistency-preserving manner. In addition, we propose and evaluate an innovative reconfiguration support architecture and protocol that extend the 3GPP and ETSI next-generation network (NGN) framework. The protocol introduces novel signaling solutions for registration of reconfigurable devices to the network, negotiation of operational mode, and administration of the radio-software download process, with the architecture being modular and inline with the subsystem-oriented rationale of 3GPP/ETSI NGN IMS (IP Multimedia Subsystem).

The paper is organized as follows: Sect. 2 presents the status of reconfiguration-related activities from industry fora, standardization bodies, and research efforts, surveys existing standards for describing reconfiguration capabilities (i.e., metadata) of mobile network elements, and points out their shortcomings. Section 3 introduces a generic, modular logical architecture to support protocol stack reconfiguration of 4G mobile devices. Section 4 analyses the degree of runtime flexibility required by an adaptable communication subsystem and establishes the basic classes of protocol stack reconfiguration. To this end, it presents a generic object-oriented UML metadata model for the description of reconfigurable protocol stacks in multi-standard settings and discusses appropriate representation technologies. Section 5 illustrates signaling sequences for protocol stack reconfiguration and Sect. 6 summarizes our experience from a prototype implementation and evaluates the performance of the proposed signaling. Section 7 concludes the paper and provides directions for future research.

2 Standardization, Industry Initiatives, and Research Efforts

2.1 Activities Related to Architecture Development

Developments in the mobile equipment industry indicate that market stakeholders find realistic benefits in the vision of reconfigurable systems. The Open Base Station Architecture Initiative (OBSAI) launched by stakeholders in the cellular radio equipment industry aims to create an open market for cellular base stations by facilitating the flexible assembly of a functional base-station from OBSAI-compliant modules [2]. Having established collaboration agreements to relevant standardization bodies (e.g., 3GPP) and industry fora (e.g., SDR Forum), OBSAI has published a specification release for internal base-station interfaces, covering control, transport and base-band modules. The Common Public Radio Interface (CPRI) initiative is an industry effort that develops specifications for the internal interfaces of radio base stations between the radio equipment and radio equipment control. Similar initiatives are pursued by mobile network operators and radio equipment manufacturers; for instance, NTT DoCoMo has introduced a software renewal system that allows the users of i-mode^{TM} cellular handsets to download software upgrades for their handset model while mobile [3] and Vanu Inc. has installed software radio GSM systems in various North American locations [4].

By leveraging the work of the DARPA Joint Tactical Radio System (JTRS) software radio projects, SDR Forum addressed reconfiguration in wireless communications and pioneered in defining a Software Radio Architecture (SRA) [5] based on the CORBA Component Model (CCM) standard [6]. DARPA work continues in the neXt Generation (XG) program that develops enabling technologies and system concepts to dynamically redistribute available spectrum among wireless devices [7].

The OMG Software-Based Communication (SBC) Domain Task Force has published a specification for a Platform-Independent Model (PIM) and a Platform-Specific Model (PSM) for software radio components [8]. Based upon the SRA foundation but without its CORBA CCM model specificities, the PIM/PSM specification is aligned to OMG's Model-Driven Architecture (MDA) conventions and covers a subset of the SRA architecture through a platform/waveform approach. The platform provides a standard set of services that abstract hardware dependencies and support the development of portable waveform (i.e., software radio) and other (e.g., management) types of applications.

Jointly formed by IEEE's ComSoc and EMC societies, the IEEE P1900 Committee[1] is tasked with the development of IEEE standards for next generation radio systems capable of advanced spectrum management [9]. With regard to design matters and, more specifically, the definition of a logical architecture in support of reconfiguration, the P1900.4 working group develops architectural building blocks to support distributed decision making between the target (reconfigurable) device and the supporting network for optimized radio resource usage in heterogeneous wireless access networks. However, the work pursued therein has not (yet) included the description of protocol stack reconfiguration options along with their associated semantics. Finally, in January 2008, the ETSI Board approved the creation of the Technical Committee on Reconfigurable Radio Systems (RRS) chartered to the development of European standards in the areas of software/cognitive radio and reconfigurable systems. We note that the building blocks introduced by the IEEE 1900 standard [10] do not capture the functionality proposed by our architecture, as described in Sect. 3.

[1] In early 2007, the IEEE P1900 Committee was reorganized as the IEEE Standards Coordinating Committee (SCC) 41, Dynamic Spectrum Access Networks (DySPAN).

2.2 Activities Related to Capabilities Description

2.2.1 3GPP Standards

In the 3GPP management specifications [10], the structure of cellular mobile network architectures and its equipment's capabilities are addressed by the Network Resource Model (NRM) concept. The NRM is a technology-agnostic model describing information objects that represent 3GPP network resources (e.g., an RNC node). The generic NRM defines information object classes and generic interfaces that are independent of any protocol solution set and network domain, thus providing the largest subset of information object classes that are common to all NRM instances to be defined by 3GPP work (e.g., Core Network NRM, UTRAN NRM). The generic NRM defines logical interfaces that allow management agents to retrieve the attributes of a network element, to navigate any containment relations to information objects contained therein and to request notifications about particular events of interest (e.g., a change in an attribute's value). The UMTS NRM specification [12] builds on the generic NRM and complements it with additional information object classes for functional entities contained in UMTS network elements (e.g., RNC functions) along with their containment configuration (e.g., an RNC function may contain zero or more Iub functions).

To support the online update of a terminal device's firmware, the 3GPP Mobile station application Execution Environment (MExE) framework [13] defines a so-called core software download capacity which, however, considers device firmware a monolithic software package that can only be updated in its entirety. Clearly, in cases where only a small part of the firmware is affected by the update, this feature proves ineffective.

2.2.2 Open Mobile Alliance Standards

The 3GPP MExE standard for 3G mobile terminals reuses the Open Mobile Alliance (OMA) User Agent Profile (UAProf) specification [14]. UAProf deals with classes of device capability and user preference information for the purpose of adapting content delivery. UAProf extends the precedent WAP specification to enable end-to-end flow of user agent profile information termed Capability and Preference Information between the client device, the intermediate network elements (i.e., proxies and gateways) and the server providing the content. It aligns to existing standards for the distribution of Composite Capabilities/Preferences Profile (CC/PP) information over the Internet by leveraging the following World-Wide Web Consortium (W3C) standards:

- The Resource Description Framework (RDF) [15] for the definition of the UAProf information model.
- The Resource Description Framework Schema (RDF Schema) for the definition of the UAProf vocabulary.
- The CC/PP specification as a high-level structured framework for describing capability and preference information using RDF.

The capability information is represented as collections of properties which, collectively, form a vocabulary. Profile attributes may have composite and/or multiple values and their final value is determined according to the associated resolution rules defined in the UAProf standard.

The mechanisms for conveying UAProf capability data are the HTTP Extension Protocol (HTTPex), the CC/PP exchange protocol, and W-HTTP. Device capability information is embedded in HTTP/1.1 headers and sent to the service and/or content provisioning entity. Capability exchange between the latter and the terminal device is performed using either HTTP/1.1 or an HTTP/1.1 derived protocol (e.g., the Wireless Session Protocol (WSP) of the WAP protocol stack).

2.3 Research Efforts

Conduits+ [16] proposed a black-box framework for network protocol software whose design factors out common structure and behavior from the protocol specific parts in reusable software components, which, can be combined into different protocol implementations without requiring any source code modifications. DiPS/CuPS supports the dynamic adaptation of protocol stacks by treating an entire protocol layer as a component [17,18]. In the x-kernel framework, microprotocol objects form the functionality of the protocol when assembled, according to a graph definition [19]. The Cactus and Appia systems extended this concept, by proposing a hierarchical composition mechanism for composite protocols based on QoS requirements [20,21].

The common shortcoming of these approaches is that they either address protocol composition issues solely, or, in cases where protocol reconfiguration is considered, focus exclusively on the functionality required (e.g., for signaling). The problem of how can alternative valid protocol stacks be discovered and evaluated according to specific communication requirements is not addressed.

In Europe, the E^2R project has developed architectural designs for reconfigurable devices and supporting system functions to offer an extensive gamut of operational choices to the users, application and/or service providers, operators, and regulators in the context of heterogeneous mobile radio systems [16]. The design impact of flexible and efficient spectrum sharing on the radio interface technologies has also been addressed in the IST WINNER integrated project [23].

2.4 Problem Statement and Motivation for Research

Modern mobile networks are heterogeneous in the type of equipment they comprise and disparate in their architecture. Hence, without a coherent logical architecture that defines end-to-end reconfiguration procedures, the effectuation of reconfiguration schemes is all but straightforward.

Both the SDR Forum SRA and the OMG SBC PIM/PSM model support the reconfiguration of an SDR system by formalizing the structural elements of SDR software and specifying OMG Interface Definition Language (IDL) interfaces to support assembly/disassembly and bootstrap/shutdown procedures. Since only technological-independent interfaces are specified, these definitions are purely abstract. Deployment concerns pertaining to mobile network architectures are not addressed by the specification, primarily because they are considered an implementation choice. Consequently, although reconfiguration-related functionalities are specified, the actual network entities where these will reside have not been designated. This lack of deployment mappings undermines the advent of reconfiguration technologies in mobile networks.

In addition, SRA and the PIM/PSM specifications focus on the (software) radio front-end subsystem while the problem of combining individual protocol layers into a valid protocol stack is only partially addressed. PIM/PSM does not consider semantic dependencies that arise when a protocol layer is reused across multiple protocol stacks in different stratifications. Such dependencies are independent of a particular model and/or its implementation artifacts and describing those calls for a properly designed vocabulary. Unfortunately, the PIM/PSM standard provides no such modeling instrument. Consequently, the semantic integrity of PIM/PSM component assemblies is impossible to guarantee.

The 3GPP NRM also cannot be used to describe the feasible reconfiguration options of 3GPP network elements, for it lacks a definition of object classes and attribute types related to reconfiguration. However, the 3GPP NRM can potentially support the description of reconfiguration options, provided it is extended with appropriate concepts, such as the ones included in the metadata model presented in subsequent sections.

UAProf supports content adaptation by considering device capability information that does not change significantly during an access session. Network and link-layer properties (e.g., multi-RAT capability), however, are technology-specific and, therefore, are either unsupported or insufficiently addressed by the UAProf standard. Most importantly, however, the latter is bound by a flat component model, i.e., the hierarchical composition of its components is impossible. This is a serious design limitation that practically rules out the representation of inherently hierarchical structures such as a protocol graph.

Finally, the E^2R research efforts have addressed the reconfiguration by using a platform-independent model for reconfigurable equipment according to the MDA design paradigm. However, a definitive model for essential metadata regarding the dynamic reconfiguration of a device's communication subsystem and its enclosed protocol stacks is not yet available, although such descriptions are a prerequisite to reconfiguration.

3 A Novel Modular Reconfiguration Architecture

In previous work, we proposed architecture for the dynamic provisioning of network elements to support downloadable applications to mobile devices, focusing on service provisioning in beyond 3G settings [24]. The work presented herein addresses the deficits explained in the previous section by defining a generic functional architecture to support end-to-end reconfiguration procedures where, besides the reconfigurable mobile device, additional network entities also take an active role. The architecture is founded upon an object-oriented information model that describes the valid feasible combination of protocol layers and protocol stacks.

3.1 Functional Analysis and Decomposition

At a general level, our architecture is based on the following functionalities.

3.1.1 Capability Negotiation

Capability negotiation takes place between the reconfigurable device (the mobile terminal in most cases) and the functional entities in the network that support reconfiguration operations. Capability data are communicated to the entity responsible for discovering the feasible communication personalities and protocol stacks. Such data include information about the protocol software downloaded to the device, its current protocol graphs and the status of relevant resources, e.g., available capacity of the download cache.

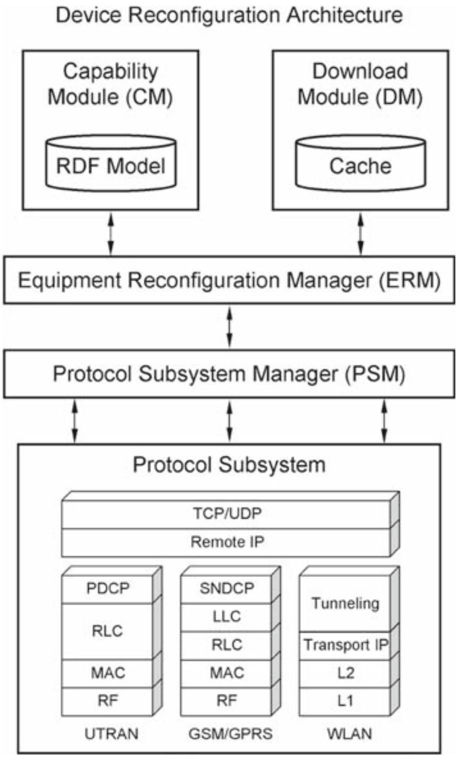


Fig. 1 The device reconfiguration architecture

3.1.2 Decision-Making

The decision-making functionality processes device capability information and global reconfiguration metadata about the communication modes, protocol stacks, and contributed implementations to select the ones that can be deployed on the reconfigurable device. The footprint of software bundles and the availability of device resources are considered in building the list of feasible protocol stacks.

3.1.3 Deployment of Protocol Software

Deployment concerns the over-the-air download of protocol software from online repositories to the target device as well as some post-download tasks. The download phase mainly concerns the error-free reception of software bundles. Post-installation tasks cater for the validation of the software's trustworthiness (i.e., that it has not been tampered with), its installation into the protocol execution environment, its incorporation into the device's protocol libraries and, optionally, an in-situ testing phase.

Our architecture features functional entities on the reconfigurable mobile device and the associated network segments, termed *Device Reconfiguration Architecture (DRA)* and network *Reconfiguration Support Subsystem (RSS)*, respectively.

3.2 Device Reconfiguration Architecture

The device reconfiguration architecture comprises the following modules (Fig. 1):

- The *Capability Module (CM)* is responsible for storing, retrieving, parsing, interpreting, querying, comparing and, updating the device's reconfiguration capabilities. The latter are represented in RDF using our RDF Schema with SPARQL Protocol and RDF Query Language (SPARQL) being employed as the query language.
- The *Download Module (DM)* deals with the download of protocol software from online repositories to the device's cache (i.e., download takes place in pull mode).
- The *Equipment Reconfiguration Manager (ERM)* exchanges reconfiguration-related signaling with supporting elements in the fixed network infrastructure for purposes of capability negotiation, over-the-air software download, and protocol stack reconfiguration.
- The *Protocol Subsystem Manager (PSM)* is responsible for consistency of protocol operation across reconfigurations. To this end, PSM uses a life cycle management interface to the protocol subsystem in order to steer affected protocol components towards a reconfiguration-safe state. Typically, that involves suspending selected computational tasks in the affected protocols, such as input/output processing, timeout expirations, and so on. Thus, protocol state is self-contained and insertion and/or removal of protocols are made possible.
- The *Protocol Subsystem (PS)* provides an execution environment suitable for protocol software (i.e., catering for memory management, multithreading, synchronization, task scheduling, etc.), along with the facilities needed for runtime reconfiguration (i.e., late binding, dynamic class loading). Several functional prototypes of SDR Forum's SCA have validated the suitability of CORBA technology for the development of an execution environment that meets the stringent performance requirements of a protocol stack subsystem [25].

Figure 1 illustrates the device reconfiguration architecture for a UMTS/WLAN mobile device for the 3GPP Release 6 system specification.

3.3 Network Architecture in Support of Reconfiguration

The Reconfiguration Support Subsystem provides a modular functional architecture to support and orchestrate the reconfiguration process. Typically, an RSS deployment resides in the non-access stratum of the 3GPP system (e.g., the 3GPP core network domain or a trusted third party domain) and is accessible over the IP protocol. The entire RSS architecture is based on the functional decomposition of reconfiguration procedures and their entailed prerequisite capacities.

3.3.1 RSS Design Principles and Rationale

Designed to integrate smoothly within existing cellular mobile network architectures, the RSS architecture comprises three distinct functional planes:

- The *Intermediary Reconfiguration Management Plane (IRMP)* that concerns management aspects of reconfiguration procedures such as access control, scheduling of vendor-initiated batch software upgrades, and remote provisioning of device policies.
- The *Data Switching Sub-Plane (DSSP)* that preserves the data bearer services of protocol layers affected by reconfiguration, in a seamless way, if possible.

- The *Mode-Independent Control Plane (MICP)* that supports IRMP operations through suitable control interfaces for its tasks such as reconfiguration session control, software download, and validation of a software package's integrity.

In the following sections, we focus on MICP and its functional entities. For details on IRMP and DSSP, the reader is referred to [26].

3.3.2 The Mode-Independent Control Plane (MICP)

The MICP resembles a meta-control plane, which mediates between the control plane and the IRMP. MICP functions carry out reconfiguration actions requested from the IRMP for network-initiated reconfigurations. Conversely, they comprise the network point-of-presence and initial contact point for device-initiated reconfigurations. The MICP functional entities initiate, setup, supervise and, ultimately, teardown sessions hereafter called reconfiguration sessions and download sessions:

- A *reconfiguration session* allows the RSS to (a) be informed about the option set of software or protocol configurations that can be supported by the terminal, (b) grant the terminal permission for the planned reconfiguration, and (c) schedule protocol installation. The initiation and coordination of capability exchange and negotiation procedures are integral parts of the reconfiguration session, as will be explained in Sect. 5.
- A *download session* determines the steps required to ensure the efficient transport of the selected protocol stack software. We introduce the notion of *download context*, which includes the necessary network resources and state information to support the download transfer (e.g., bearer QoS identifiers, <requester, server, provider> identifiers), is generated during a download session and distributed to the concerned network elements (e.g., GGSN, PDG), as will be illustrated in the signaling exchange of Sect. 5.

MICP entities support the following functionality:

- Reconfiguration strategies and policy rules processing. Reconfiguration strategies define high-level constraints dependent on non-technical restrictions such as time zone restrictions defined as part of the user subscription and/or roaming options defined by inter-operator agreements. Policy rules define the respective technical considerations, such as the required memory, processing, and energy resources for affected network elements.
- Signaling for operational mode negotiation and transition, as well as actual switching between different modes of operation (e.g., between different protocol stacks).
- Distribution of download context information to the concerned network elements.

3.3.3 MICP Functional Entities

The MICP realizes its reconfiguration-related functionality through distinct functional entities, thus providing a modular architecture with clear separation of reconfiguration control tasks. The MICP functional specification is inline with the best practices in the industry. Specifically, the functional decomposition bears the well-established philosophy of the IP Multimedia Subsystem (IMS) developed in 3GPP SA2 [27] and ETSI TISPAN NGN working groups [28]. NGN IMS offers a clear functional separation between entities that (a) act as central anchor point for user sessions, (b) undertake specific tasks upon redirection of messages from the anchor point, and (c) comprise dedicated functions for locating and accessing specialized registries, or for other specific capabilities.

Inline with the NGN model, the MICP design rationale defines four categories for its functional entities (Fig. 2):

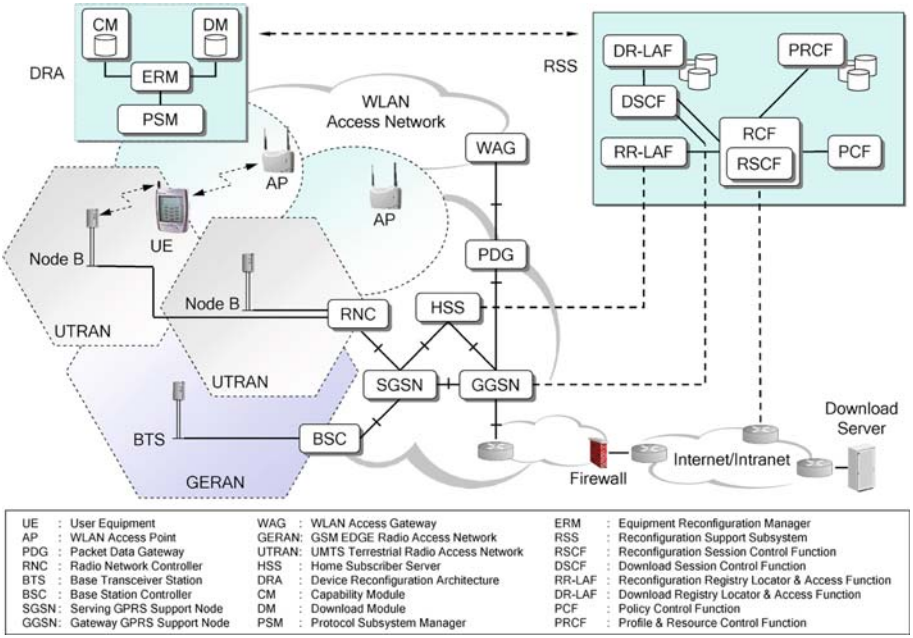


Fig. 2 The overall DRA/RSS architecture as an overlay on top of a 3GPP Release 6 (UMTS/WLAN) system

- The initial RSS contact point as addressed by the reconfigurable device, namely the *Reconfiguration Control Function (RCF)*.
- Session-related entities, namely the *Reconfiguration Session Control Function (RSCF)* and the *Download Session Control Function (DSCF)*.
- Locator and access functional entities, termed *Reconfiguration Registry Locator and Access Function (RR-LAF)* and *Download Registry Locator and Access Function (DR-LAF)*.
- Policy and profile/resource control entities, namely the *Policy Control Function (PCF)* and the *Profile and Resource Control Function (PRCF)*.

The RCF provides the initial contact point in case of terminal-initiated reconfiguration. The RCF hands over control of individual reconfiguration sessions to the RSCF, which handles session-specific state and progresses the reconfiguration procedure. Similarly, DSCF is responsible for individual download sessions.

The RSCF employs the RR-LAF to locate (in multi-HSS environments) and access the user's HSS for subscription data. Additionally, the RR-LAF discovers and retrieves reconfiguration profile data from reconfiguration registries. In a similar fashion, the DSCF interrogates the DR-LAF for which download servers to use for a particular software module.

The PCF receives reconfiguration policy rules and strategies from the IRMP and may interact with a Policy Enforcement Point (PEP) in the network infrastructure (e.g., the GGSN PEP). These policy rules and strategies persist in a dedicated policy and QoS registry. Finally, the PRCF handles user, service, terminal, network, and reconfiguration profile information. Such data are delivered to the PRCF by its peer management entity (called *Performance Management Function—PMF*[2]) via access to a dedicated *Profile and Resource Registry (PRR)*,

[2] IRMP and DSSP functional entities are not depicted in Fig. 2; space and scope limitation do not allow us to further detail them herein.

and are exploited during the *reconfiguration session establishment* and *reconfiguration setup* stages, as elaborated in Sect. 5. The PRCF also keeps track of resource consumption during the progress of a reconfiguration (e.g., drain of a mobile device's energy supply). Finally, by accessing the PRR, the PRCF receives offline reports on performance measures by the PMF.

Being a generic functional architecture, DRA and RSS can be deployed over a variety of network infrastructures, including 3GPP cellular wireless networks. Given that the most recent release (i.e., 8) of the 3GPP system is still in flux, we opted to consider the deployment of DRA and RSS in a prior stable release, specifically the 3GPP Release 6 system. The later supports multiple wireless technologies for its radio access network segment (i.e., UTRAN, EDGE, WLAN), thus providing a sufficiently rich set of protocol layer and protocol stack options to use in reconfiguration applications. Figure 2 depicts an enhanced 3GPP Release 6 system that includes DRA and RSS functional components.

4 A Generic UML Model for Reconfiguration Metadata

4.1 Describing Reconfigurable Protocol Stacks

Reconfiguration involves an exchange functionality that manifests itself at particular reference points. When a reconfiguration is merely about switching between different implementations of a particular protocol layer in the protocol stack, the exchange reference point is virtual. This means that, it exists between an implicit abstract specification of the affected protocol's functionality and all of its available implementations—as opposed to being an actual reference point defined in the protocol stack architecture. In a reconfiguration that involves changes to the stratification (i.e., the layering) of protocol layers in a protocol stack, the exchange reference point lies within the realm of the protocol stack architecture—at the boundaries of adjacent protocol layers. Hence, protocol reconfiguration within and across the boundaries of communication standards requires two distinct kinds of exchange functionality:

- Exchanging different implementations of a protocol (e.g., exchanging a TCP Tahoe implementation with a TCP Reno one).
- Exchanging different protocol stratifications as a whole, i.e., exchanging parts of the protocol stack. Such substitutions may involve an entire protocol stack as the substitute (e.g., using a GPRS protocol stack as a link-layer protocol in the TCP/IP protocol suite).

4.2 Requirements of Description

Currently, protocol specifications are specified in human language and, therefore, are not subject to automation. This practically rules out the possibility of having protocol specifications interpreted by a computational agent engaged in a protocol stack reconfiguration. However, to accomplish this task, such an agent does not need a description of each involved protocol's internal functions; the capacity to identify each distinct protocol in the protocol stack and to discover its associated implementation is sufficient. Consequently, an information model that describes reconfigurable protocol stacks must support the following functionalities:

- Identifying a protocol's specification;
- Identifying a protocol's implementations.

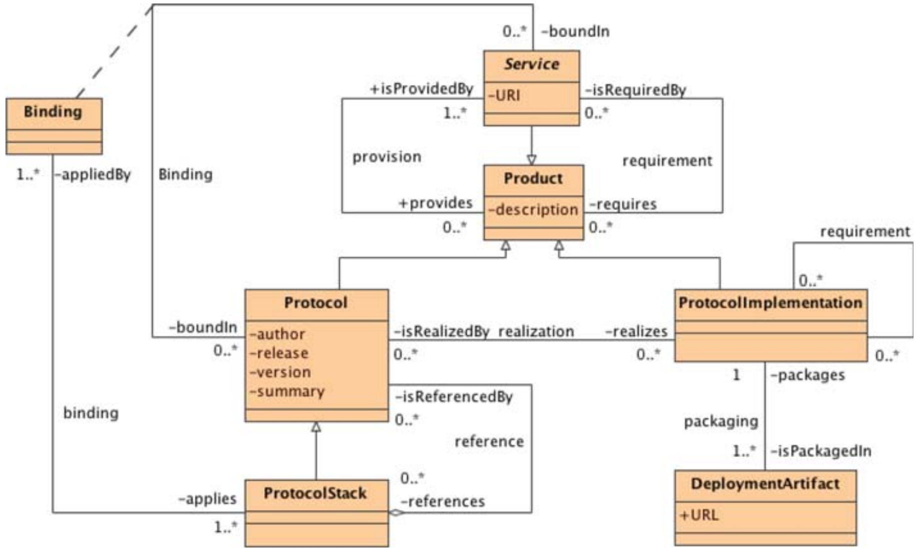


Fig. 3 Proposed metadata model for reconfigurable protocol stacks

In addition, to ensure that the reconfiguration of a protocol stack does not compromise its integrity, a way to identify the dependencies that arise between protocol layers due to their stratification pattern is necessary. Figure 3 below illustrates our model's classes and associations in UML notation.

4.3 Metadata Classes

In our model (Fig. 3), all the information classes extend the *Product* class, which represents objects that are vested with resource semantics and can be identified through a (textual) name.

The *Service* class represents a specific functionality which is associated to a textual description. Thus, it acts as a placeholder for the definition of a service and its textual descriptor. The descriptor may bear formal semantics (e.g., IDL). We stress that, it is not particularly important that a unique formal format is agreed upon for the service descriptor's notation, as adaptation mechanisms can be used to identify the proper handling procedure for each format.

Specification is a subclass of *Product* that serves as an abstract class for the established specification of protocol functionality (e.g., the specification of a mobility protocol's functionality). To this end, it provides a suitable set of attributes (author, release, version, and summary) to accommodate protocol specifications published by standardization bodies (e.g., the protocol specifications of 3GPP).

Implementation is a subclass of *Product* that refers to a real-life artifact that realizes (i.e., implements) functionality associated to one (or more) protocol specifications. It models a software instrumentation of one or more specifications but may also be used to represent an implementation of functionality unrelated to a protocol's specification (e.g., utility functionality). Protocol stacks having one or more common protocols in identical or different layers are represented ontologically via (shared) instances of the Implementation class. Considering that an implementation may be packaged in various deployment formats (e.g., Java archive, Microsoft CAB, etc.) its proper modeling must be capable of representing multiple different packaging artifacts. The *Deployment Artifact* class supports this packaging disparity and deployment concerns (i.e., download and installation of software bundles).

4.4 Metadata Associations

In general, a *Specification* instance may depend on the availability of multiple services much as it may render multiple services. In a similar manner, an *Implementation* instance, besides the set of services that its associated (via the *realization* association) *Specification* instances collectively require and provide, may depend on additional *Service* instances to function properly (e.g., utility functionalities) and may also provide additional *Service* instances. These options are represented through *requires/required By* and *provides/ provided By* associations at the closest common super-class (i.e., *Product*), respectively.

This high degree of modeling flexibility with regard to required and realized services does not vitiate the requirement to associate software implementations of a particular protocol's functionality to the latter's specification. Nonetheless, it is necessary to ensure that, at least at a modeling level, compliance to a specification does not smother innovation and that the different ways in which a particular protocol's implementation may be structured form an open set.

It is not mandatory for an *Implementation* instance to be associated to a *Specification* instance through a *realization* association; for it might be implementation of utility functionality not subject to standardization yet required by some other *Implementation* instance. The possibility of having an *Implementation* instance that is unrelated to a *Specification* instance does not obliterate the value of the *Specification* class as a modeling instrument. *Implementation* instances that are associated to a *Specification* instance will be identified as such through the *realization* association, while those without such an association will not. That is not to say that *Implementation* instances without an association to a *Specification* instance will not be identified as such; just that they will be classified as software implementations that do not realize a particular protocol's specification.

4.5 Metadata Encoding

The reconfiguration of a mobile device's protocol stack may acquire and process metadata from multiple sources (e.g., device manufacturer, mobile network operator). Hence, an encoding format that is suitable for circulating across technologically dissimilar administrative domains and, in addition, guarantees semantic univocality, is necessary.

For this task we chose RDF, primarily due to its powerful capabilities for expressing semantically rich information and its capacity for unambiguous representation (thanks to its RDF Model Theory [29] foundation). The vocabulary used in the RDF representation is a combination of the standard RDF vocabulary and a custom vocabulary derived through an isomorphic mapping [30] of the UML model's classes and associations to an RDF Schema document and identified by the prefix 'rcm'.

Figure 4 illustrates a simplified form of the RDF graph for reconfiguration metadata related to the protocol stacks of a 3GPP Release 6 mobile device that supports UMTS and WLAN modes of operation. The list of identified *Service* instances is tentative; alternative identification and naming of service entities (e.g., in further detail) is possible.

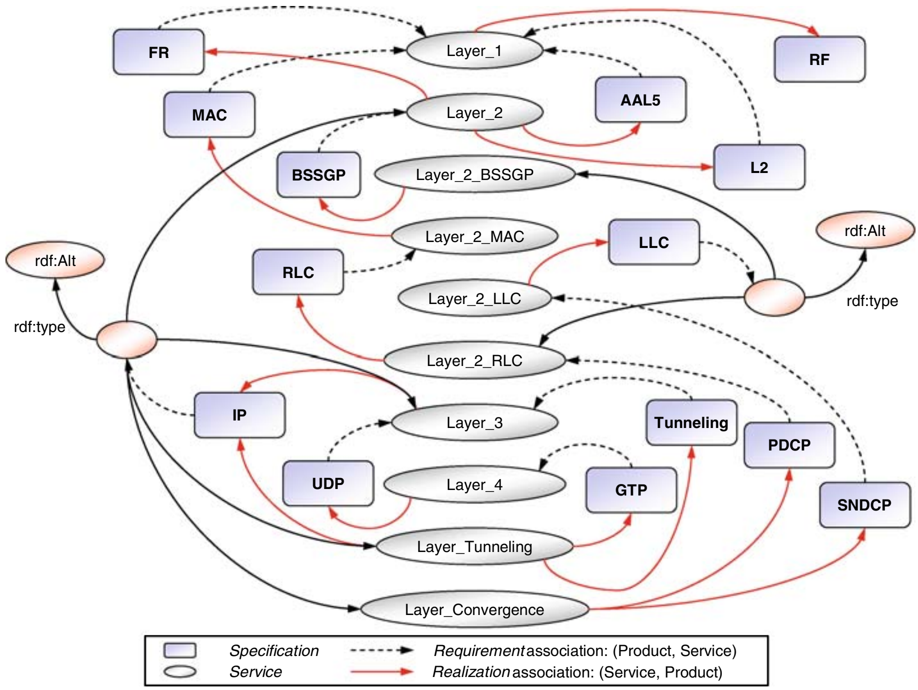


Fig. 4 Simplified RDF graph for reconfiguration metadata about the packet-switched domain protocol stacks of a 3GPP UMTS/WLAN mobile device (i.e., 3GPP Release 6)

4.6 Metadata Generation

Metadata generation basically involves the mapping the aforementioned protocol inter-dependencies associated to a protocol stack onto the appropriate model elements. In cases where common protocol layers may be reused in multiple protocol stacks, these inter-dependencies form a directed graph. The challenging task in metadata generation is to come up with a mapping that guarantees uniqueness of representation and preserves semantic consistency. In our approach, a (formally proven) transformation is applied to the original directed graph of protocol inter-dependencies to produce the associated model instances. However, due to space limitations, the presentation of this transformation is not included herein.

4.7 Metadata Processing

We call *reconfiguration option discovery* the process by which an intelligent agent discovers the combinations of known protocol specifications, standards, and associated protocol implementations that render an integral and usable system. Considering that a reconfiguration may often involve heterogeneous equipment with radically different characteristics, some sort of capability exchange and negotiation phase should take place prior to reconfiguration option discovery (e.g., through the aforementioned standard mechanisms). Thanks to the graph model theory foundation of RDF, one can query the RDF graph of reconfiguration metadata for entries with specific properties and get unambiguous results. The W3C

Table 1 Querying the definitive descriptions of feasible protocol stack combinations using SPARQL

SPARQL Query: retrieve all (non-blank RDF node) instances of RDF type rcm:Service with an association of RDF type rcm:providedBy to a (non-blank) instance of RDF type rcm:Specification.

```
BASE  <http://www.di.uoa.gr/~gazis/metadata/instances/>
PREFIXrdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIXrcm: <http://www.di.uoa.gr/~gazis/metadata/schemas/RCM#>
SELECTDISTINCT                          ?t
WHERE { ?t rdf:type                     rcm:Service;
          rcm: providedBy               ?s.
        ?s rdf:type                     rdf:Specification;
          rcm:provides                  ?t.
FILTER(!isBlank(?s) && !isBlank(?t)) }
```

SPARQL Query: retrieve all (non-blank RDF node) instances of RDF type rcm:Service with an association of RDF type rcm:requiredBy to the (non-blank) instance of RDF type rcm:Specification that has an association of type rcm:name to the RDF (string) literal "GTP"

```
BASE  <http://www.di.uoa.gr/~gazis/metadata/instances/>
PREFIXrdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIXrcm: <http://www.di.uoa.gr/~gazis/metadata/schemas/RCM#>
SELECTDISTINCT                          ?t
WHERE { ?t rdf:type                     rcm:Service;
          rcm:providedBy                ?s.
       ?s rdf:type                      rdf:Specification;
          rcm:provides                  ?t;
          rcm:name                      "GTP".
FILTER(!isBlank(?s) & & !isBlank(?t)) }
```

SPARQL [31] language establishes the definitive grammar for querying RDF. Table 1 provides two indicative applications of SPARQL.

By formulating appropriate SPARQL queries, an intelligent agent can navigate an RDF knowledge base of reconfiguration metadata and support reconfiguration option discovery.

5 Reconfiguration Signaling Between Terminal and Network Entities

In our architecture, a reconfiguration progresses through the following generic stages (Fig. 5).

5.1 Stage 1: Reconfiguration Registration

This stage allows a reconfigurable device to register and authenticate with the RSS, thus enabling the latter to retrieve subscription data from the user's HSS and to compile the *UE Reconfiguration Profile (URP)* option set. The URP includes parameters related to the initial applicable option set for the particular subscriber and mobile device. The URP is expressed in RDF through the RDF Schema vocabulary, whereas profile data are encoded in XML. The *network reconfiguration profile* includes RDF/XML descriptions of the available (radio)

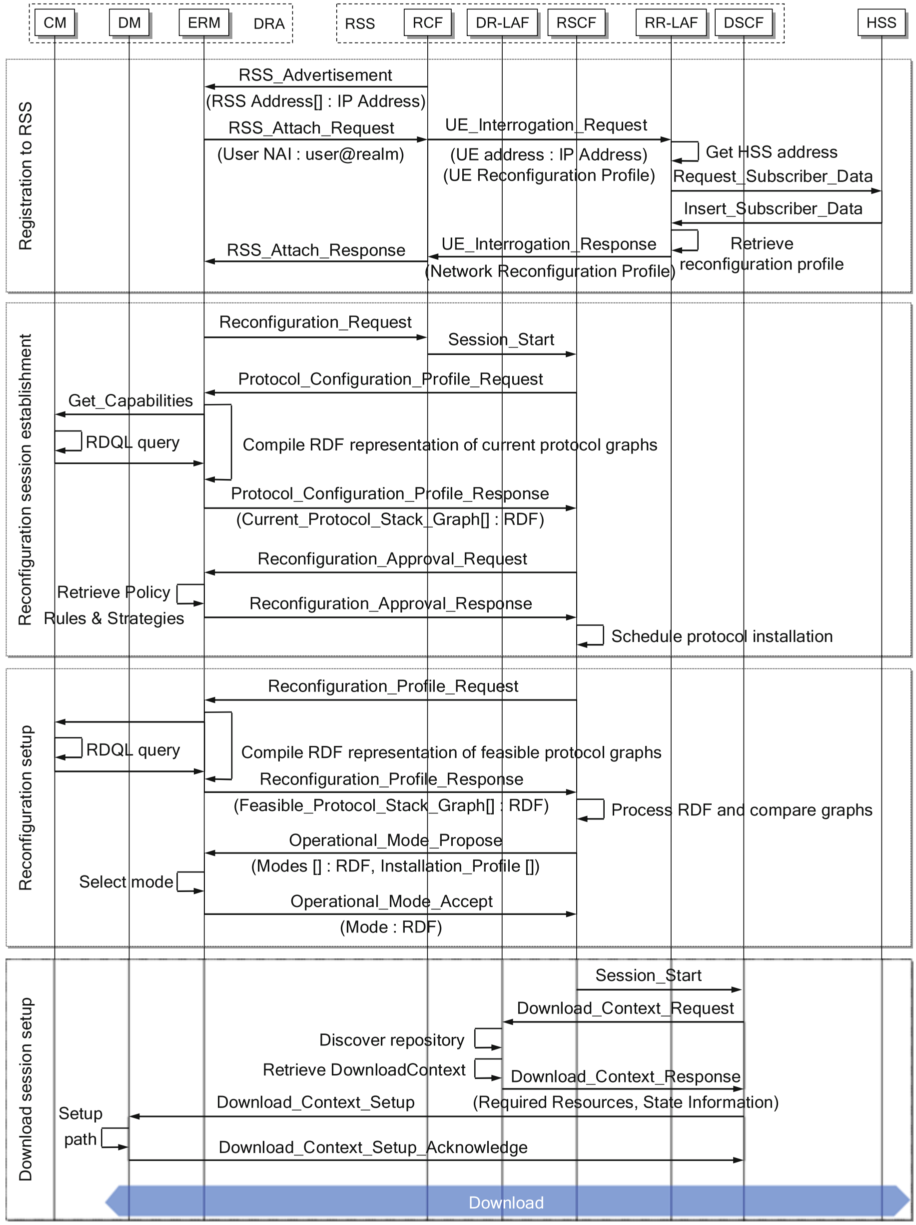


Fig. 5 Reconfiguration signaling between DRA and RSS as it progresses through the reconfiguration stages

access networks and protocol operation modes suitable for the particular user and feasible by his/her mobile device.

Reconfiguration registration is handled by the RCF, which employs the RR-LAF to locate the user's HSS and access subscription data therein.

5.2 Stage 2: Reconfiguration Session Establishment

When the DRA ERM detects a new RAT, it notifies the RSS RCF and requests an update on the available reconfiguration services. Such fresh information is important in order to cover self-management scenarios, where the device may opt to reconfigure autonomously, i.e., upgrade its RAT configuration without further involving the RSS. The RCF hands over the control of the reconfiguration session to RSCF, which requests the *protocol configuration profile* from the ERM. First, the device compiles a list of current protocol stack stratifications (i.e., protocol modules and their combinations into a working protocol stack), encodes it in the RDF Schema vocabulary and delivers it to the RSS. The use of RDF for capability data yields a small footprint for the encoded profile by delegating the resolution and retrieval of static profile components to the RSS via CC/PP mechanisms, thus economizing on wireless medium resources. Next, the RSS requests the approval of the reconfirmation decision by the ERM. To validate the response, the ERM retrieves and evaluates its local reconfiguration strategies. Besides, the ERM retrieves and evaluates local policy rules; this step is important in order to exclude the forbidden configuration options (for example, the device can juxtapose with a "black list" of configurations). If the ERM acknowledges positively, the RSCF schedules the protocol software installation phase that will follow.

5.3 Stage 3: Reconfiguration Setup

The reconfiguration setup stage involves a communication loop between the DRA, the RSS, and—possibly—other network nodes, in order to reach a common agreement regarding the mode of operation and the associated provisioning parameters. The ultimate decision depends on the anticipated impact of the selected modes to the network segment providing connectivity between the RSS and the reconfigurable device. For example, addition of a WLAN radio access capability to a mobile device will also impact its traffic distribution pattern through the availability of a high capacity radio interface. Hence, when a large number of mobile devices is involved in such reconfigurations, the aggregate impact on traffic distribution will be manifold and, therefore, should be considered, particularly in the case of evolved sophisticated scenarios (e.g., involving roaming between different mobile network operators; 3GPP network sharing scenarios; 3GPP/WLAN tight coupling integration, where traffic from the WLAN air interface will traverse the 3G core network). Such procedures and scenarios are not presented here due to space limitations.

Stage 3 involves the RSS RSCF acquiring (a) the RDF representations of *feasible* protocol stack graphs (compared to the *current* graph at stage 2), and (b) the agreed-upon protocol mode of operation. In addition, it compiles the RDF representation of the installation profile. In a similar fashion, the DRA ERM processes the received RDF data, and decides on the optimal mode. Once the ERM reports its decision to the RSCF, the latter fixes the reconfiguration session state information and hands over control to the DSCF.

5.4 Stage 4: Download Session Setup

During the download session setup stage, the DR-LAF locates the appropriate software repositories and retrieves the download context, which it delivers to the DSCF. Next, the DSCF triggers the download context setup, which aims to support the resource reservation phase.

5.5 Final Stages: Software Transfer, Protocol Stack Activation, and Transition of Operational Mode

The final stages of the reconfiguration process are rather straightforward and are listed below:

- Transfer of the protocol stack (software) bundles from the software repositories to the reconfigurable device.
- Instantiation of protocol modules and construction of the new protocol stack.
- Integration with existing protocol stacks in the protocol subsystem and activation of the new protocol stack.
- Switching from the old to the new protocol stack (if necessary).
- Transition to the new operational mode.
- Deactivation of the old protocol stack (if necessary, or simultaneous multi-stack operation otherwise).

Figure 5 illustrates the typical signaling sequence between DRA and RSS for the aforementioned reconfiguration stages.

6 Evaluation of DRA/RSS Functionalities

6.1 Prototype Design Objectives

The reconfiguration of a protocol stack involves several delay-incurring tasks:

(a) Device-to-network signaling for reconfiguration control.
(b) Information processing tasks (e.g., processing of protocol stack descriptions).
(c) Download of software bundles.
(d) Activation/deactivation of protocol instances.
(e) Insertion/removal of protocol instances into/from protocol stacks.

Delay factors (a) and (c) depend on the actual infrastructure, architecture design choices, as well as the network load, whereas factors (d) and (e) are determined by the control capacities offered by the device's communication subsystem. Delay factor (b), upon which we focus next, is a function of the associated information model's complexity.

In our modeling approach, delay factor (b) results from the processing of the associated RDF graphs. To assess it quantitatively, we have developed a prototype of the protocol stack discovery algorithm. To validate our information model, we developed RDF descriptions of the GPRS, UMTS and IMS protocol stacks for both the user and control planes using our information model. Figure 6 presents the spatial footprint of the respective RDF graphs in terms of the count of its elements, i.e., resources, literals, and statements.

6.2 Protocol Stack Discovery

The on-demand discovery of valid protocol stacks is required in nomadic settings where one cannot safely assume the set of protocol options available at a mobile device. Moreover, several of the 3GPP specifications detail only the service to be provided by a particular protocol layer, leaving the choice of the actual protocol as an implementation option. As a result, pre-computation and caching of known options is not possible, since the set of available protocol is essentially an open one.

In our prototype, the discovery of a feasible protocol stack translates to the traversal of a path in the RDF graph (see Fig. 4). This path originates at the node corresponding to the

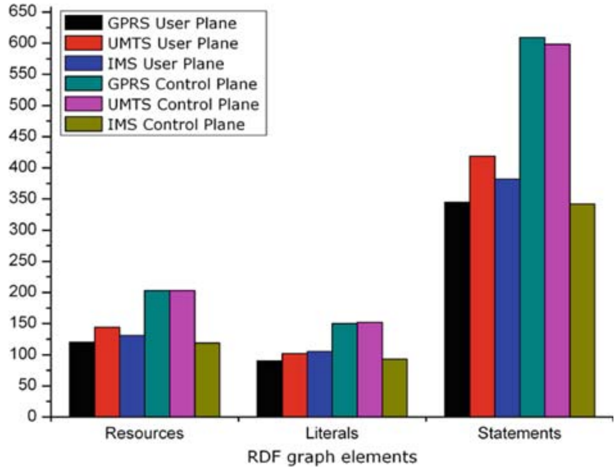


Fig. 6 Spatial footprint of the RDF graphs of 3GPP standards described in the trials

given *Service* instance *t* and terminates at a node that corresponds to a *Specification* instance *s* without requirement associations to a service instance (i.e., a protocol layer at the bottom of the protocol stack). The completion criterion of the discovery algorithm is the formation of a protocol stack that (a) provides a specific service instance *t* at its top specification instance *s* and, (b) provides no instance *t* more than once (i.e., duplication of functionality within the protocol stack is considered redundant).

The number of iterations carried out by the protocol stack discovery algorithm is proportional to the number of protocol layers in the protocol stack (i.e., its height). Hence, the higher the protocol stacks of an access standard are, the more iterations will be required to discover all their feasible combinations. Due to their GSM legacy, the control planes of the GPRS and UMTS standards exhibit higher protocol stacks than the IMS standard whose control plane is based on the shorter Internet protocol stacks. This is also reflected in their spatial footprints that quantify the size and complexity of the corresponding RDF graphs (Fig. 6).

6.3 Trial Instrumentation

Our prototype is based on the Jena Semantic Web framework [32], the ARQ SPARQL API [33], the Jakarta Velocity template engine [34] and the Jakarta JMeter framework [35]. Reflecting the design of our information model, the developed prototype is also generic in that it queries the given RDF graphs using the standard W3C SPARQL query language for RDF. Thus, independence from the employed encoding format for RDF (XML, N3, N-Triples, etc.) and storage pattern is achieved and the seamless integration of RDF graphs from multiple sources encoded in different formats is straightforward.

To investigate performance aspects pertaining to the discovery of all protocol stack combinations for common access standards, we trialed our prototype. The trial setup employed Sun's Java VM 1.5.0–10 for Debian/GNU Linux 4.0 installed in a Pentium M 1.6 GHz laptop with 768 MB RAM. To factor out temporal variability attributed to the dynamic compilation feature of the Java VM (i.e., to allow it to reach a steady-state performance with regard to the dynamic compilation of Java bytecode), the discovery of all feasible protocol stacks was

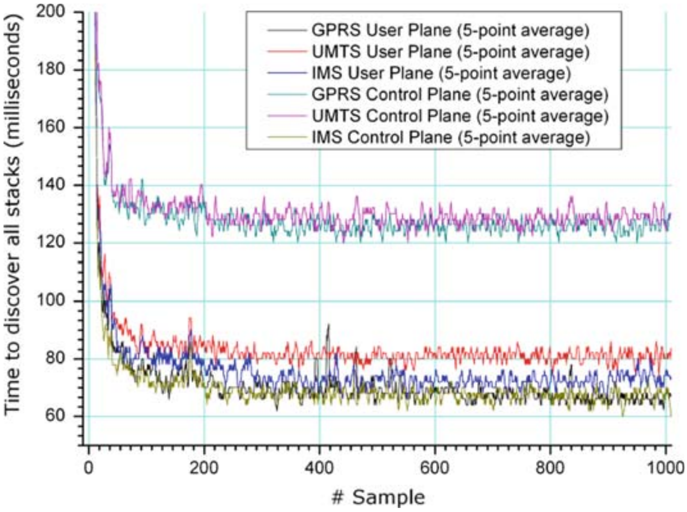


Fig. 7 Time required for the discovery of all the feasible protocol stacks for common 3GPP standards

carried out 1,000 times for each of the described 3GPP standards and five-point moving average values were taken (Fig. 7).

6.4 Assessment of Computational Cost

As Fig. 7 illustrates, the (steady-state) discovery time falls in the range of 60–90 ms for the GPRS, UMTS and IMS user plane protocol stacks. This is also the case for the IMS control plane, but not for the GPRS and UMTS control planes which require larger discovery times, in the range of 120–140 ms across all trials. This is due to the larger (in comparison to that of the other access standards) average height of their control plane protocol stacks which, in turn, results from their GSM/SS7 legacy (to a large extent, the GPRS and UMTS standards inherited the protocol stacks of the GSM control plane). IMS, on the other hand, is a standard independent of the employed transport technology (referred to as the “IP connectivity access network” in 3GPP standards), with its control plane protocol stacks employing SIP/SDP on top of the TCP/IP suite of protocols. In our trial, the link-layer protocols used by IMS were the GPRS and UMTS user plane protocol stacks. As a result, the average height of its control plane protocol stacks is comparable to that of the GPRS and UMTS user plane protocol stacks. In fact, the IMS control plane stacks only one or two additional protocol layers on top of whatever user plane protocol stacks it employs. Consequently, the respective delays were only marginally longer than those of the GPRS and UMTS user plane protocol stacks (as shown in Fig. 7).

Hence, reconfiguring the protocol stacks of a GPRS or UMTS device (i.e., a commercially available cellular handset) will take more time for the control plane protocol stacks than the user plane ones. If IMS protocol stacks are also included in the reconfiguration procedure, the delay overhead will be minimal.

Consequently, in discovering all feasible protocol stacks for the case of the GPRS, UMTS and IMS access standards, it is the GPRS and UMTS control planes protocol stacks that form the dominant delay factors. If, however, protocol stack reconfiguration is allowed to independently evolve user and control plane protocol stacks, then the overall minimum of reconfiguration delay can be achieved (as user plane switching comprises the time-critical operation). Based on these quantitative assessments, the formulation of reconfiguration strategies that produce the desired modes of operation with incurring minimal delays is made possible.

It should be clarified that an amount of variation in the measured delay is unavoidable, because the underlying execution platform (i.e., the Java virtual machine) does not support a predictable real-time performance.

6.5 Evaluation of Signaling Cost

To evaluate the performance of the proposed signaling, we have estimated the total signaling cost associated to the four stages depicted in Fig. 5, namely:

- registration to the RSS (stage $s1$),
- reconfiguration session establishment (stage $s2$),
- reconfiguration setup (stage $s3$), and
- download session setup (stage $s4$).

We assume that the cost of each signaling message is proportional to (a) the network distance (in terms of IP hops) between the communication endpoints and (b) a constant per-hop transmission and link cost C_t (e.g., in terms of delay or monetary cost). Message processing costs at each node (DRA, RSS, and HSS) may depend on many factors (e.g., node capabilities, overall load of the node). For the sake of simplicity, we do not consider processing costs in our analysis.

The following additional parameters are used:

d_r	average distance between the DRA and the RSS;
d_h	average distance between the RSS and the HSS;
μ	number of times a DRA changes RSS service area per unit of time;
λ	number of times a DRA triggers a reconfiguration procedure per time unit;
P_a	probability a DRA accepts a reconfiguration approval request by the RSS, given that a reconfiguration session establishment procedure has been initiated (i.e., probability of initiating the reconfiguration setup stage $s3$, given a reconfiguration trigger at stage $s2$);
k	number of mode negotiation iterations during the reconfiguration setup stage;
C_{s1}	signaling cost of registration to RSS (stage $s1$);
C_{s2}	signaling cost of reconfiguration session establishment (stage $s2$);
$C_{s3,4}$	signaling cost of reconfiguration setup and download session setup (stages $s3$ and $s4$).

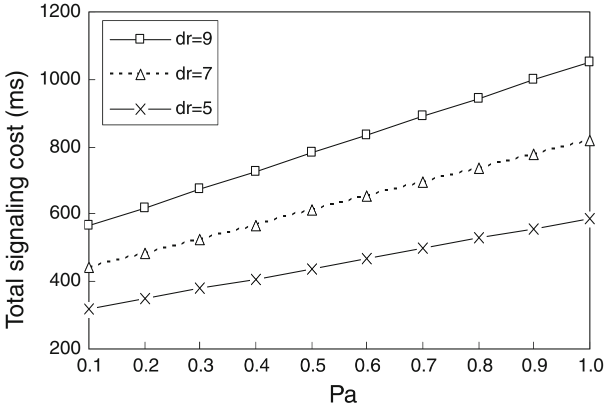
According to the signaling flows of Fig. 5, the total signaling cost C_{tot} can be computed as follows:

$$C_{tot} = \mu \cdot C_{s1} + \lambda \cdot (C_{s2} + P_a \cdot C_{s3,4}) \tag{1}$$

where:

$$C_{s1} = (3 \cdot d_r + 2 \cdot d_h) \cdot C_t$$
$$C_{s2} = 5 \cdot d_r \cdot C_t$$
$$C_{s3,4} = 2 \cdot (k + 2) \cdot d_r \cdot C_t$$

Fig. 8 Total signaling cost given $C_t = 10\,\text{ms}$, $d_h = 2$, $\lambda/\mu = 5$, and $k = 1$



The above formulas consider only the signaling messages between the DRA, RSS, and HSS elements. The signaling cost of messages exchanged between DRA-internal and RSS-internal functional entities is assumed zero. Figure 8 plots the total signaling cost as a function of P_a, and for various values of d_r. As expected, the total signaling cost increases when the distance between the DRA and the RSS increases.

As can be deduced from Fig. 5, the major component of signaling cost originates from stages *s2* and *s3*, which involve interactions between the DRA ERM and the RSS RSCF functional entities. Therefore, placing the RSCF entity closer to the radio interface will yield significant improvement of the total signaling cost (due to smaller value of d_r). It is worth noting that this gain will occur without increasing the cost of communication between the RSS and HSS (which involves only the RR-LAF entity and depends solely on the value of d_h). In other words, the modular design of the RSS allows the fine-grained introduction of reconfiguration capacities in 3GPP networks, thereby alleviating the shortcomings of a fully-centralized RSS server lying beyond the core network domain. Thus, by the appropriate distribution of the RSS logical entities over the domains of 3G networks (i.e., RAN, CN, etc.) and their subsequent allocation and/or co-location to standard 3GPP network elements, the performance impact associated to the introduction of reconfiguration capacities can be minimal.

Let $d_{r'}$ be the new distance between the DRA and the RSS RSCF ($d_{r'} < d_r$). The total signaling cost $C_{tot'}$ is expressed as

$$C_{tot'} = \mu \cdot C_{s1} + \lambda \cdot (C_{s2'} + P_a \cdot C_{s3,4'}) \tag{2}$$

where C_{s1} is independent of the RSCF location, and $C_{s2'}$ and $C_{s3,4'}$ are computed from the following formulas:

$$C_{s2'} = (2 \cdot d_r + 3 \cdot d_{r'}) \cdot C_t$$
$$C_{s3,4'} = [3 \cdot d_r + (2 \cdot k + 1) \cdot d_{r'}] \cdot C_t$$

Figure 9 shows the cost improvement factor $(C_{tot\neg} - C_{tot'})/C_{tot}$ as a function of the distance ratio $d_{r'}/d_r$, for various settings of λ / μ and $P_a = 0.8$, $k = 2$. This factor is independent of C_t. We call λ/ μ = *RMR* (*reconfiguration-to-mobility ratio*), which indicates the ratio of the number of reconfiguration triggers per unit of time to the number of changes of RSS service areas per unit of time for a mobile terminal. The other functional entities of the RSS element still reside at a distance d_r from the DRA. It can be seen that, the closer the placement of the

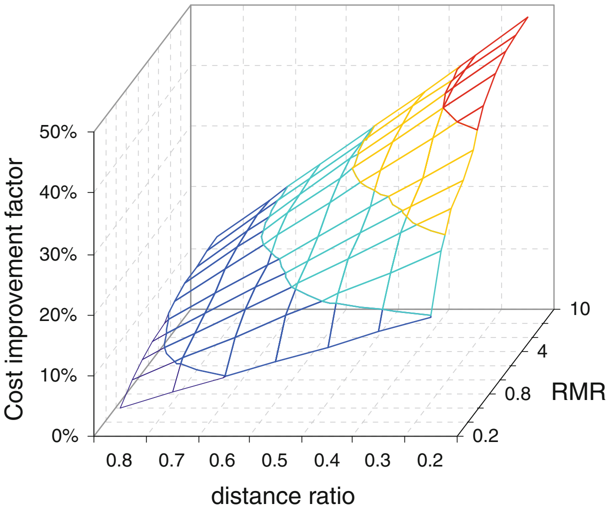


Fig. 9 Cost improvement when placing the RSCF closer to the radio interface, as a function of the distance ratio $d_{r'}/d_r$, for various settings of λ/μ ($P_a = 0.8$, $k = 2$)

RSCF towards the air interface (i.e., lower values of the distance ratio), the higher the cost improvement factor. In addition, the cost improvement factor increases for higher values of RMR. This is because, for large values of λ/μ, the mobile terminal triggers a reconfiguration procedure more often than it changes RSS areas. Since stages $s2$ to $s4$ involve many interactions with the RSCF (whereas *s1* does not), and generate more signaling messages compared to *s1*, the cost improvement increases for higher values of λ/μ.

Next, we obtain the cost ratio as a function of RMR:

$$\frac{C_{tot'}}{C_{tot}} = \frac{1 + \frac{\lambda}{\mu} \cdot \frac{1}{C_{s1}}(C_{s2'} + P_a \cdot C_{s3,4'})}{1 + \frac{\lambda}{\mu} \cdot \frac{1}{C_{s1}}(C_{s2} + P_a \cdot C_{s3,4})} \tag{3}$$

Figure 10 illustrates the behavior of the cost ratio for $P_a = 0.8$ and $k = 1$. The ratio is always smaller than one, since the new RSCF placement reduces the total signaling cost. The decreasing shape of the curves comes from the fact that, for low-speed users, the frequency of reconfiguration events dominates over the frequency of mobility events (i.e., registrations to RSS), thus yielding smaller values of the total signaling cost.

Finally, it is worth noting that the reduction of the total cost is higher when the DRA and the RSS are involved in mode negotiation loops ($k > 1$). This is again due to the optimized RSCF location, which results in faster signaling exchanges during stage $s3$.

7 Conclusions

We have presented a novel generic architecture that supports the reconfiguration of operating protocol stacks on mobile devices in a manageable modular fashion. The architecture addresses issues such as registration of reconfigurable devices to the network, establishment of reconfiguration sessions, mode negotiation, administration of download sessions, and local protocol deployment tasks. Our design takes into account established practices and guidelines in the relevant standardization bodies to come up with a functional architecture that can

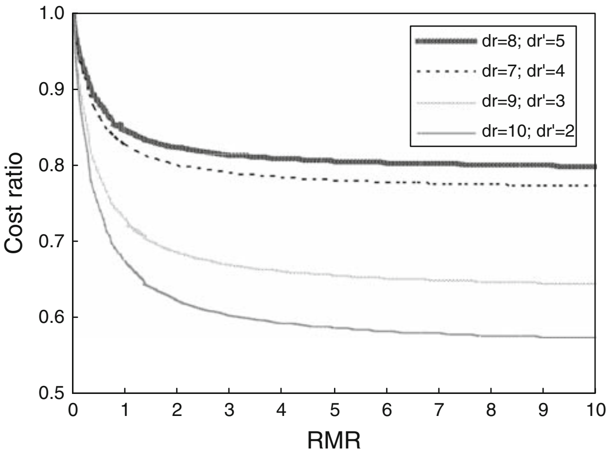


Fig. 10 Cost ratio for various reconfiguration-to-mobility ratios and $< d_r, d_{r'} >$ pairs ($P_a = 0.8$, $k = 1$)

be seamlessly integrated into existing mobile networks in a non-invasive manner. The clear identification of functional entities in support of reconfiguration paves the way for the piecemeal definition of reconfiguration capacities that augment functional entities in current user, control and management planes. The separation between user and control plane functions is inline with 3GPP design principles for its Release 8 system and supports the IETF interest in the utter and complete separation of control and forwarding elements [36].

The proper semantic description of reconfigurable protocol stacks forms the cornerstone of our design and is used in signaling between our architecture's functional entities. With regard to metadata description, RDF presents a somewhat steep learning curve that discourages its prospective adopters. However, once mastered, it offers powerful expressive capabilities which, in addition, are always semantically unambiguous. For instance, in XML, semantic univocality is impossible without an established agreement between all involved parties about the interpretation of XML documents. The intellectual effort associated with RDF can be efficiently alleviated by the use of modern visual modeling tools [37], thus significantly lowering the barriers to its adoption. With regard to scalability, RDF is a distributed framework designed to scale effortlessly to huge (e.g., Web) dimensions. By using SPARQL syntax, the navigation of RDF graphs is greatly simplified.

The prototype developed for validation purposes has been based solely on open standards and employs exclusively open source technologies. This proof-of-concept instrument is employed to quantitatively assess the impact of information processing tasks that occur in the context of reconfiguration procedures. We have also systematically computed the total signaling cost and proposed optimized placements of specific functional entities in a real-network setting, thus minimizing the impact of the introduced interactions on already deployed infrastructures.

Our future efforts include the extensive trials of cross-standard reconfigurations between protocol stacks that exhibit common parts (e.g., the protocol stack of a 3GPP Release6/Release7 mobile device). Additional performance evaluation studies will indicate the most suitable distribution of RSS functional entities at network elements of next-generation topologies such as the 3GPP System Architecture Evolution [38].

Acknowledgments The second author would like to acknowledge the Alexander S. Onassis Public Benefit Foundation for a doctoral fellowship.

References

1. Dillinger, M., Madami, K., & Alonistioti, N. (Eds.) (2003). *Software defined radio: architectures, systems and functions*. Hoboken, NJ: Wiley.
2. Open Base Station Architecture Initiative, http://www.obsai.org/.
3. Infoworld: NTT DoCoMo and NEC plan 3G/WLAN handset, http://www.infoworld.com/article/03/12/03/HN33gwlanhandset_1.html.
4. Vanu Inc., press release, available from http://www.vanu.com/.
5. Bickle, J. (2000). Software radio architecture (SRA) 2.0 technical overview, OMG TC meeting, Orlando, Florida, http://www.sdrforum.org/.
6. OMG CORBA Components, Version 3.0. (2002). document formal/02-06-65, http://www.omg.org/.
7. DARPA next Generation (XG) communication program, http://www.darpa.mil/sto/smallunitops/xg.html.
8. OMG Software-Based Communication (SBC) Domain task force, http://sbc.omg.org/.
9. IEEE P1900 Committee, http://grouper.ieee.org/groups/emc/emc/1900/index.html.
10. IEEE Std. 1900.4-2009. (2009). IEEE standard for architectural building blocks enabling network-device distributed decision making for optimized radio resource usage in heterogeneous wireless access networks.
11. 3GPP TS 32.106, 3GPP; Technical specification group services and system aspects; telecom management: Principles and high-level requirements telecommunication management; configuration management, http://www.3gpp.org/.
12. 3GPP TS 32.106-5, Configuration management (CM); part 5: Basic CM integration reference point (IRP): Information model (including network resource model (NRM), http://www.3gpp.org/.
13. 3GPP TS 22.057, Mobile execution environment (MExE) service description; stage 1, http://www.3gpp.org/.
14. Open Mobile Alliance, UAProf, http://www.openmobilealliance.org/.
15. W3C RDF: Resource description framework, http://www.w3.org/RDF/.
16. Hermann, H., Johnson, R., & Engel, R. A framework for network protocol software, Proceedings OOPSLA '95, ACM SIGPLAN Notices.
17. Janssens, N., Michiels, S., & Verbaeten, P. (2001). DiPS/CuPS: A framework for runtime customizable protocol stacks, technical report, DistriNet, K.U. Leuven.
18. Michiels, S., Mahieu, T., Matthijs F., & Verbaeten, P. (2001). Dynamic protocol stack composition: Protocol independent addressing. In 4th ECOOP workshop on object-orientation and operating systems (ECOOPOOOSWS' 2001).
19. Hutchinson, N. C., & Peterson, L. L. (1991). The x-kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering, 17*(1), 64–76.
20. Mena, S., Cuvellier, X., Gregoire, C., & Schiper, A. (2003). Appia vs. cactus: Comparing protocol composition frameworks. 22nd International symposium on reliable distributed systems (SRDS'03), Florence, Italy.
21. Kannan, M., Komp, E., Minden, G., & Evans, J. (2003). Design and implementation of composite protocols. Technical report ITTC-FY2003-TR-19740-05.
22. Muck, M., Bourse, D., Moessner, K., Alonistioti, N., Demestichas, P., Nicollect, E., et al. (2007). End-to-end reconfigurability in heterogeneous wireless systems–software and cognitive radio solutions enriched by policy-and context-based decision making. *Proc. 16th IST mobile and wireless communications summit* (pp. 1–5). IEEE: Budapest, Hungary.
23. Bennis, M., Wijting, C., Ojanen, P., Thilakawardana, S., Tafazolli, R., Aebdi, S., et al. (2007). Dynamic spectrum management for future radio networks. *Proc. WWRF#19*, Chennai, India.
24. Gazis, V., Alonistioti, N., Houssos, N., Koutsopoulou, M., Gessler, S., & Quittek, J., et al. (2006). Intelligent network provisioning for dynamically downloadable applications in beyond 3G mobile networks. *Journal of Network and Systems Management, 14*(2), 221–241.
25. Bertrand, J., Cruz, J., Majkrzak, B., & Rossano, T. (2002). CORBA delays in a software-defined radio. *IEEE Communications Magazine, 40*(2), 152–155.
26. Boufidis, Z., Alonistioti, N., & Merakos, L. (2006). RSS: The reconfiguration support subsystem for next-generation mobile networks. *Proc. IEEE International Conference on Communications (ICC)*, Istanbul, Turkey.

27. 3GPP TS 23.228: 3rd Generation partnership project; technical specification group services and system aspects; IP multimedia subsystem (IMS); Stage 2 (Release 8).
28. ETSI ES 282 001: Telecommunications and internet converged services and protocols for advanced networking (TISPAN); NGN functional architecture release 1.
29. Hayes, P. RDF semantics, http://www.w3.org/TR/rdf-mt/.
30. Chang, W. (1998). A discussion of the relationship between RDF-Schema and UML, W3C discussion note, http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804.
31. SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/.
32. Jena Semantic Web framework, http://jena.sourceforge.net/.
33. ARQ–A SPARQL processor for Jena, http://jena.sourceforge.net/ARQ/.
34. Jakarta Velocity, http://velocity.apache.org/.
35. Jakarta Jmeter, http://jakarta.apache.org/jmeter/index.html.
36. IETF Forwarding and Control Element Separation (forces) working group, http://www.ietf.org/html.charters/forces-charter.html.
37. The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.
38. 3GPP TR 23.882, 3rd Generation partnership project; technical specification group services and system aspects; 3GPP system architecture evolution: Report on technical options and conclusions, http://www.3gpp.org.

Author Biographies

Vangelis Gazis holds B.Sc., M.Sc. and Ph.D. degrees from the Department of Informatics and Telecommunications at the University of Athens, and an M.B.A. degree from the Athens University of Economics and Business). He is a senior researcher at the Department of Informatics and Telecommunications and has participated in national (OTE-DECT, GUNet, GEANT-2) and European projects (MOBIVAS, ANWIRE, SELFNET). He specializes in reconfigurable mobile systems and protocol stacks for beyond 3G mobile, ontology languages for autonomic systems, reflective and component middleware, adaptable services, and open API frameworks for telecommunications.

Zachos Boufidis received the B.Sc. and M.Sc. degrees (both with honors) from the Dept. of Informatics and Telecommunications, University of Athens, Athens, Greece, in 2000 and 2004, respectively, and the Ph.D. degree from the same department in 2009. Since 2000, he has acquired technical and management experience in EU-funded (E3, Self-NET, E^2R II, E^2R, MOBIVAS, ANWIRE) and national research projects (GUnet). He has articulated part of his work in (pre)standards bodies and research fora such as 3GPP, ETSI, SDR Forum, IEEE P1900, OMG, WWI, and WWRF. He received an Alexander S. Onassis doctoral fellowship, an Ericsson Hellas Award of Excellence in Telecommunications, and an award from the Hellenic Mathematical Society. His research interests are in protocol design and performance analysis of cognitive mobile networks, reconfigurable systems, and autonomic communications.

Nancy Alonistioti has B.Sc. and Ph.D. degrees in informatics and telecommunications from the University of Athens. She specializes in reconfigurable systems and networks for beyond 3G, adaptable services, pervasive computing, and context awareness. She has participated in national and European projects, (CTS, SS#7, ACTS RAINBOW, EURESCOM, MOBIVAS, ANWIRE, E2R, LIAISON, SELFNET) and is co-editor of Software Defined Radio, Architectures, Systems and Functions (Wiley Series on Software Radio). She has been elected Lecturer at the Department of Informatics & Telecommunications, University of Athens.

Lazaros Merakos received his Diploma in electrical and mechanical engineering from the National Technical University of Athens in 1978, and M.S. and Ph.D. degrees in electrical engineering from the State University of New York, Buffalo, in 1981 and 1984, respectively. From 1983 to 1986 he was on the faculty of the Electrical Engineering and Computer Science Department, University of Connecticut, Storrs. From 1986 to 1994 he was on the faculty of the Electrical and Computer Engineering Department, Northeastern University, Boston, Massachusetts. During the period 1993–1994, he served as director of the Communications and Digital Processing Research Center, Northeastern University. During the summers of 1990 and 1991 he was a visiting scientist at the IBM T. J. Watson Research Center, Yorktown Heights, New York. In 1994 he joined the faculty of the University of Athens, where he is presently a professor in the Department of Informatics and Telecommunications, and director of the Communication Networks Laboratory (UoA-CNL) Networks Operations and Management Center. Since 1995 he has led the research activities of UoA-CNL in the area of mobile communications, in the framework of the Advanced Communication Technologies and Services (ACTS) and Information Society Technologies (IST) programs funded by the European Union (RAINBOW, Magic WAND, WINE, MOBIVAS, POLOS, ANWIRE, E2R, LIAISON, GEANT-2, SCIER, SELFNET). His research interests are in the design and performance analysis of broadband networks, and wireless/mobile communication systems and services. He has authored more than 150 papers in the above areas. He is on the Board of the Greek Universities Network, the Greek Schools Network, and a member of the board of the Greek Research Network. In 1994 he received the Guanella Award for the Best Paper presented at the International Zurich Seminar on Mobile Communications.