INTEGRATED RECONFIGURATION MANAGEMENT FOR THE SUPPORT OF END TO END RECONFIGURATION

Aristotelis Glentis, Nancy Alonistioti
Communication Networks Laboratory, Department of Informatics & Telecommunications, National and Kapodistrian University of Athens, 157 84, Athens, Greece

Abstract: The development, delivery and management of mobile services are the subject of many research activities in both the academia and industry. The ultimate goal of these efforts is a dynamic environment that enables the delivery of situation-aware, personalized multimedia services over heterogeneous, ubiquitous infrastructures, commonly termed as systems beyond 3rd generation (3G). Reconfigurability and adaptability are key aspects of the mobile systems beyond 3G. Reconfigurable mobile systems and networks introduce additional requirements and complexity. Using the existing network control plane is inadequate for the realization of the reconfigurability process. Introducing a Reconfiguration Management Plane is very important for the deployment of network/system wide reconfigurability. In this paper we intend to discuss the basic functionality of RMP and respective interrelations.

Key words: reconfiguration management, mobile systems beyond 3G.

1. INTRODUCTION

1.1 Towards reconfigurability

The evolution of technology has led to the introduction of the Open Business Model in the world of mobile telecommunications. The Open Business model is the model that the Internet is based on: the ISP provides the connectivity and the user access the application/service provider using open APIs and protocols that reside on top of the IP protocol. Mobile telecommunications are based on different business models, since in the mobile world the user is confined to the provider's network and cannot access services outside this domain. The mobile provider is the one responsible for the deployment and maintenance of value added services. With the arrival of 3G networks and UMTS, which offer an all IP network, this fact has changed and the opportunity of the adoption of the Open Business Model in mobile communications is possible, and can be beneficial both to the telecom operator and the application/service providers[1,2]. The telecom operators can benefit since their users can have access to a larger range of applications and services, which they don't have the burden to deploy, manage, maintain themselves. The application/service providers benefit from the larger user base that can access their products. The users can benefit from the plethora of new services and from the competition between telecom providers and achieve optimum ratio of quality of service per price.

In order to achieve this goal the need for end to end reconfigurability rises. The users want to access the applications/services that have registered to, discover new service or applications that are offered, update their software and don't be tied to a certain underlying network infrastructure but can choose the one preferred from the networks that are available in the environment according to their preferences. The users could be able to change environments (i.e. from UMTS to GSM and 802.11b) without loosing the service, if possible. The service, on the other hand, must be able to adapt to the change of the network characteristics, or to the request of the user for having better or worse quality of the service, etc. However, in the mobile world there are two issues that have to be solved, the different capabilities of the mobile devices to execute applications or services, and the mobility of the user who comes across different networks with different characteristics.

1.2 Related work

The issue of reconfigurability on these two axes has been tackled in the past mainly in the two edges of the OSI layer model: the physical and the application. On the physical layer research has been carried out so that devices can detect the networks that are available and use them to communicate. However, the research was limited to the use of different physical layers to carry the information and no provision was made for the interoperability with application's requirements. Furthermore, several attempts have been made for the introduction of adaptive protocols and respective design[3]. Building on the knowledge from early software radio projects in the military domain[4,5], SDR Forum has pioneered in exploring reconfigurability concepts in the United States. However, being the vanguard of reconfigurability developments and the first to define a software radio architecture[6,7], seems to have come at the expense of a rather restricted view on reconfigurability that focuses primarily on the radio domain (RF processing, down-conversion, RF processing, A/D conversion, etc)[8]. On the application layer, research has been carried on the adaptation of the application or service according to the predefined profiles of the user and the service in the MOBIVAS platform[9,10]. The user can discover different instances of the service according to the profile and the terminal capabilities of his device. However no input from the underlying network is used in the service provision decision. The tackling of the problem in the two edge layers, physical and application, is not efficient and sufficient since it creates a lot of difficulties to the network devices, and to the application developers and providers.

Based on the above discussion, it is apparent that in the design of fully reconfigurable networks and systems, the introduction of advanced reconfiguration management functionality is necessary. In this paper we introduce a holistic solution for addressing reconfiguration management across all layers, namely, the Reconfiguration Management Plane (RMP). RMP enhances reconfigurability control in order to address end-to-end reconfiguration management aspects.

2. RECONFIGURATION MANAGEMENT ASPECTS

In order to address reconfiguration management it is important to tackle reconfiguration in two levels:

- The local level (addressing network node and mobile device reconfiguration)
- The system level (addressing network wide reconfiguration and service adaptation).

One simplistic approach (addressing only the local level of reconfiguration) would be to assume that each of the reconfigurable devices has a local reconfiguration manager (LRM), who is responsible for the reconfiguration plane of the local device. It keeps track of the state of the device and performs the necessary actions needed for reconfigurability. The actions vary, it can be downloading components and installing software to offer new protocol stacks, changing the QoS values of the protocol stack in use, ensuring that the requirements of the application running are met, choosing the optimum combination of protocol stacks and routing according to the policy that is defined by the user or operator, triggering reconfiguration on user or application request, etc. In order to achieve these goals the LRM should have a clear picture of the network topology and be able to contact different servers and services, as well as finding the optimal network path. Furthermore, the application can reside on a server that is not in the region controller by the telecom operator, so the LRM should be able to construct the path to the remote application server. This requires that the LRM uses a lot of CPU power and that each device has the routing info for all the networks that it participates. Provided that network topologies change often in reconfiguration environments, the LRM will be flooded with control messages. The processing and space complexity becomes a major issue considering the limitations on the mobile devices.

On the other hand, another simplistic approach would be to delegate the reconfiguration responsibility to the application developer and provider. In this case, the value added service developers should be able to easily access the network in order to provide the parameters needed for the service to operate smoothly. The parameters needed (for example QoS settings) should be propagated in all the devices that are in the network path from the server that provides the VAS and the user terminal. As a result the service should have knowledge of the network topology and the application developer should cope with different network infrastructures and provide the methods to communicate with them. The application also should be able to access all the internal nodes of the network, something that is not acceptable from the telecom providers view, since this might reveal the internal structure. One possible solution for this problem is to have the necessary functions packaged in a library. This solution is quite cumbersome since all applications and services should be linked with the library, and changes to the interfaces or addition of new network infrastructure would mean the need to upgrade both the library and the application or service. This introduces a lot of overhead to the application developer. The application developers want a clean interface between the application and the network, and shouldn't be forced to cope with specific network functions.

From the above the need for an entity that controls the reconfiguration process on the network level and provides a layer of abstraction both to the terminal and the service application comes to the surface.

3. RECONFIGURATION MANAGEMENT PLANE ARCHITECTURE

3.1 General architecture

The entity identified in the previous context, is the Reconfiguration Management Plane (RMP), which is introduced and described in the current section. The RMP can be viewed as another control plane that is operating on all OSI layers and runs along with the network control plane. Its main task is to provide layer abstractions to the applications and services on the one hand and to the terminals and network devices on the other. Furthermore the RMP is responsible to coordinate the reconfiguration process and provide the required resources in order to be completed. The RMP is comprises different components that are illustrated in figure 1.

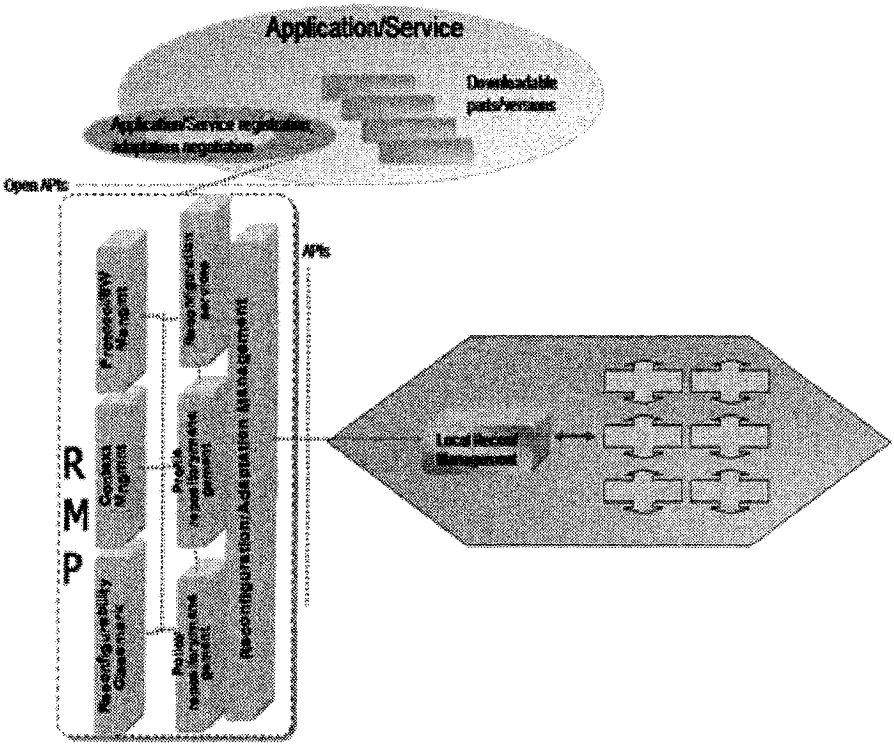
Application/Service
Downloadable parts/versions
Application/Service negotiation, adaptation negotiation
Open APIs
APIs
R
M
P
Reconfiguration/Adaptation Management
Local Reconf Management

Figure 1. The RMP entity

3.2 Architectural components

The components are:

1) The protocols/SW management. The reconfiguration procedure is based on the ability of the mobile devices, network nodes to download and install software that makes possible the support of new protocol stacks. The reconfiguration procedure encompasses the triggering of certain protocol and software to be downloaded on the mobile terminal or other network nodes/ devices in order to support efficient user connectivity and optimal service provision (e.g., downloading of certain protocols that are not installed in the mobile device). Therefore this functional entity is responsible to identify, locate and trigger the suitable protocol or SW for download.

2) the Context management. The context management is responsible to manage the context information of the network. In the RMP domain, context doesn't concern the applications that run over the network, but the network data themselves, i.e. the nodes, the state of the nodes, the congestion information etc. The context manager is responsible to have a picture of the network. This is necessary in the decision of feasibility of a reconfiguration event, or the choice of an alternate path between the application server and the user terminal.

3) the Reconfigurability Classmark. This component has a duplex role. On one side it keeps track of the different nodes of the network and their state regarding reconfigurability, i.e. the protocol versions that are installed. On the other side it keeps a database of the capabilities of the different network nodes that exist. For example it keeps track of the software that can be downloaded and run on each device, and the capabilities of the mobile device regarding reconfigurability and upgrading.

4) The Policy repository/management. The policy management component is the main decision making entity for reconfiguration. It provides the entry point for the reconfiguration related policies of the system. Furthermore, it undertakes the merging of different profiles from the profile management and combines them with the policies that are defined. The output is the final decision about the feasibility of a reconfiguration and respective actions to be triggered.

5) The profile repository/management. The profile management component is responsible to manage and combine the different profiles. The profiles come from different parts of the system since they refer to different entities of the system. The profiles can be classified in: the user profile, the network profile, the application profile, the terminal profile, charging profile, security profile etc.

6) The reconfiguration services. The reconfiguration services component is responsible for the communication between the RMP and the

application/service. It accepts and processes reconfiguration requests for the network in order to provide the necessary environment for an application and service to execute. It also provides feedback to the application for the feasibility of the request, or can also initiate a reconfiguration on the application in case for example of network configuration changes or selection of different settings by the users, mobility etc..

7) The reconfiguration Adaptation/Management. The reconfiguration adaptation management component is responsible for initiating the triggering of service adaptation based on network capability restrictions or reconfiguration policies. It triggers and coordinates the reconfiguration actions, exchanges messages with the LRM of the devices and coordinates the reconfiguration process. Internally, it accesses the components of the RMP and provides the necessary tools and information for the reconfiguration to take place.

Although there is no central security management component, security should be considered and tackled in all the RMP components. The authorization, authentication and integrity of communications among the RMP components should be assured by using proper security schemas. The selection of the security schema depends on the network infrastructure and the communication channels that are used between the components. However special consideration should be taken for the communication of the RMP and the external entities (end user terminals, network nodes, applications), most notably with the protocols/SW management component, since the need of a security schema that supports public key cryptography is needed for authentication purposes. Furthermore, solutions that use digital signatures could be considered to protect the network nodes and user terminal from downloading unauthorized software.

3.3 Communication between RMP and external entities

The communication between the RMP and the external entities is based on open APIs. The RMP communicates primarily with the local reconfiguration managers and with the Application/Service. The open APIs provide the infrastructure for applications from different vendors to communicate with the networks of different providers. Since the providers are not willing to reveal the inner structure of the network to the external applications, they are able to provide context and adaptation related information to the application/services through the use of open APIs in a controlled way. The open APIs can guarantee the construction of the communication path between the application and the reconfigurable network, so that the applications can provide the requirements and trigger reconfiguration when needed, and the network can give proper feedback without revealing internal structural information.

The local reconfiguration managers are internal components of the devices, and as a result they can have proprietary characteristics. However the need to communicate with different nodes and with the RMP makes the use of open APIs essential. The RMP has to communicate with the LRMs in order to trigger a reconfiguration procedure, to collect essential data (like protocols installed, node capabilities, etc), query about the status of the node, etc. The LRMs need to communicate with the RMP in order to trigger a reconfiguration procedure, to download software and answer to requests from the RMP. The use of open APIs is essential to provide the communication path between the devices of different manufacturers and enhancing interoperability.

3.4 Case studies

In order to clarify further the use and functionality of the RMP the following sequence diagrams that depict an overview of the main functionalities and operations performed by the RMP in the event of a reconfiguration, are also presented. The two cases that are illustrated are: the first case (illustrated in figure 2) is when the user terminal initiates the reconfiguration procedure, and the second (illustrated in figure 3) is when the reconfiguration procedure is initiated due to the provision requirements of a Value added service after a user selection for application download. In either case the RMP is the one that steers the reconfiguration process. It triggers the reconfiguration in the network nodes that are among the path from the Value Added Service Provider (VASP) to the user terminal, and provides the software for protocol stack reconfiguration in the network nodes and the user terminal. The RMP communicates with the LRM of the network nodes and the user terminal on the network side and with the application/service on the application side. However, the user terminal might have to communicate directly with the VASP in order to download extra software components (for example it might be essential to download codecs that are needed for the service the user is currently downloading).

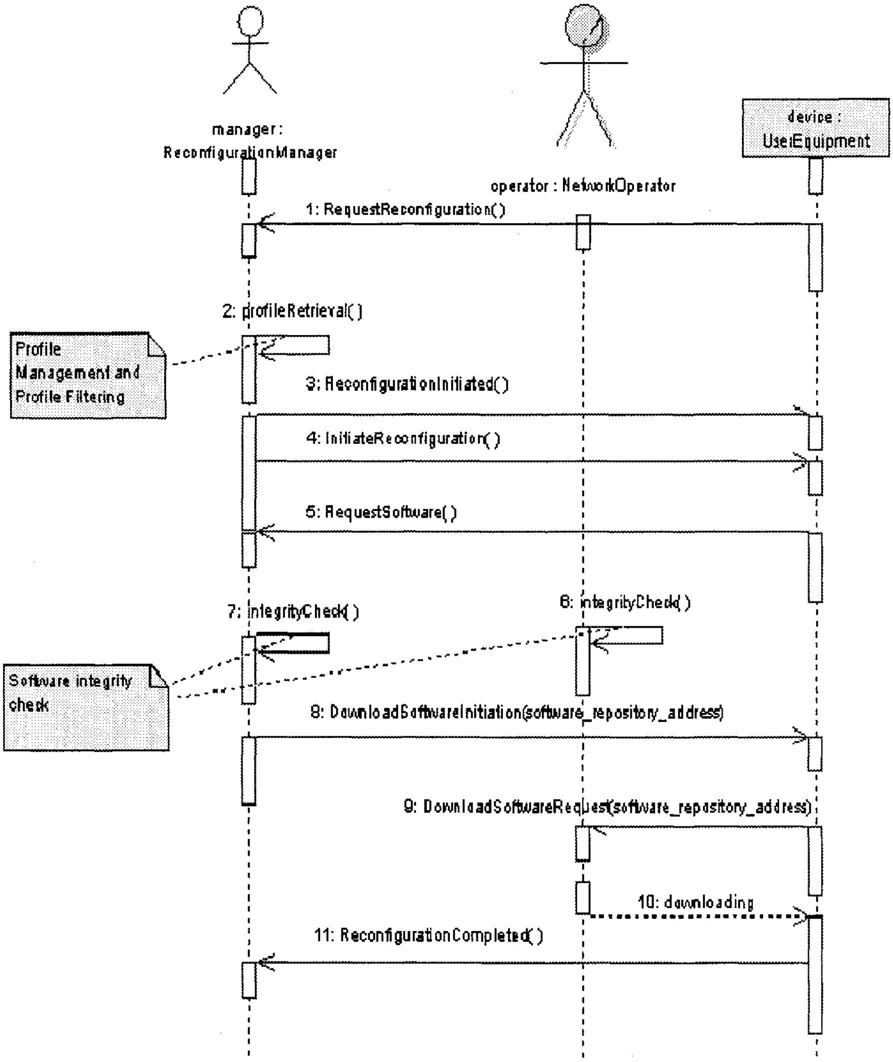
manager : ReconfigurationManager
operator : NetworkOperator
device : UserEquipment
1: RequestReconfiguration()
2: profileRetrieval()
Profile Management and Profile Filtering
3: ReconfigurationInitiated()
4: InitiateReconfiguration()
5: RequestSoftware()
6: integrityCheck()
7: integrityCheck()
Software integrity check
8: DownloadSoftwareInitiation(software_repository_address)
9: DownloadSoftwareRequest(software_repository_address)
10: downloading
11: ReconfigurationCompleted()

Figure 2. Terminal Initiated

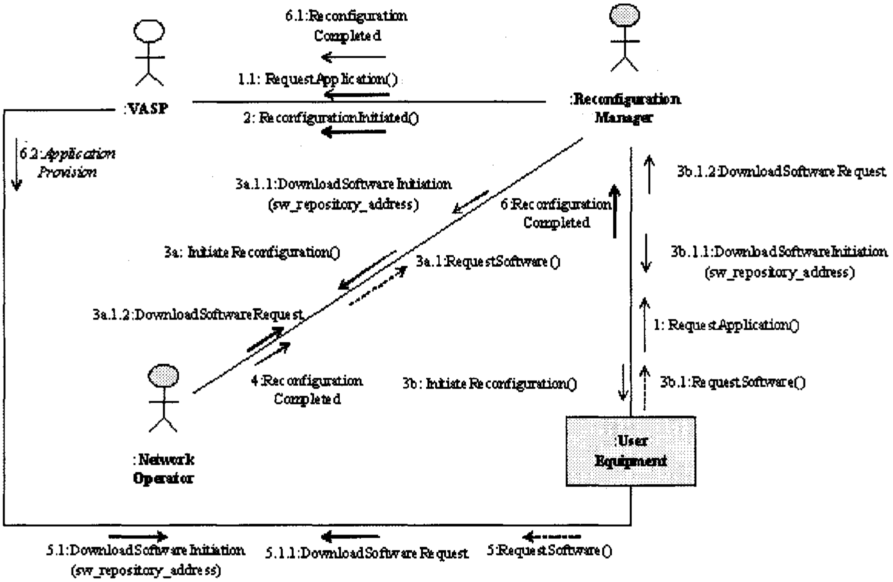
6.1:Reconfiguration Completed
1.1: RequestApplication()
:VASP
2: ReconfigurationInitiated()
:Reconfiguration Manager
6.2:Application Provision
3a.1.1:DownloadSoftware Initiation (sw_repository_address)
6 Reconfiguration Completed
3b.1.2:DownloadSoftware Request
3a: Initiate Reconfiguration()
3a.1:RequestSoftware ()
3b.1.1:Download Software Initiation (sw_repository_address)
3a.1.2:DownloadSoftwareRequest
1: RequestApplication()
4:Reconfiguration Completed
3b: Initiate Reconfiguration()
3b.1:RequestSoftware()
:Network Operator
:User Equipment
5.1:DownloadSoftware Initiation (sw_repository_address)
5.1.1:DownloadSoftware Request
5:RequestSoftware()

Figure 3. Network Initiated

4. CONCLUSIONS

The evolution of reconfigurability notion has been heralded as main concept for 4G mobile communications. In order to reach its full potential, a consistent framework that deals with reconfigurability challenges and control has to be introduced.

In this paper we have introduced a generic framework to cope with the complexity of reconfigurability management. This work will provide the basis for the evolution of End-to-End Reconfigurability notions that will be addressed and enhanced inside the E2R architecture design. The proposed architecture for reconfiguration management addresses the effective policy based reconfiguration triggering towards the network nodes and the combination of adaptation triggering towards the end-user services in order to achieve the optimal service provision and perception to the end user in a transparent way.

ACKNOWLEDGEMENTS

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

REFERENCES

1. UMTS Forum Report No. 9, "The UMTS third generation market - structuring the service revenues opportunities"; http://www.umts-forum.org/ .
2. J. Pereira, "Beyond third generation", Wireless Personal Mobile Communications (WPMC), 22 September 1999, Amsterdam, The Netherlands.
3. M. Dillinger, K. Madani, N. Alonistioti, 2003, "Software Defined Radio, Architectures, Systems and Functions", John Wiley, England.
4. M. C. Cox, "Joint tactical radio system (JTRS)", presentation available from http://www.jtrs.sarda.army.mil/.
5. The GloMo project, "Global mobile information systems (GloMo)", DARPA ITO; http://www.janet.ucla.edu/glomo/.
6. P. G. Cook, "Software architecture in software defined radios", contribution to SDR Forum meeting, February 24, 1999; http://www.sdrforum.org/.
7. J. Bickle et al., "Software radio architecture (SRA) 2.0 technical overview", presentation to OMG TC meeting, December 11, 2000, Orlando, Florida.
8. S. M. Blust et al., "SDR definitions", contribution to SDR Forum Plenary & Technical Committee meeting, September 1, 2000; http://www.sdrforum.org/ .
9. The MOBIVAS project; http://mobivas.cnl.di.uoa.gr .
10. N. Houssos, V. Gazis, A. Alonistioti, " Application-transparent adaptation in wireless systems beyond 3G", 2nd International Conference on Mobile Business (M-Business 2003), Vienna, Austria, 23-24 June 2003.