



# Wireless World Research Forum (WWRF)

WG6: Reconfigurability



Title of the research item: *Cross Layer and End to End Reconfiguration Management*

## (a) Experts:

Eiman Mohyeldin<sup>1</sup>  
Markus Dillinger<sup>1</sup>  
Jijun Luo<sup>1</sup>  
Michael Fahrmaier<sup>2</sup>  
Peter Dornbush<sup>2</sup>  
muenchen.de  
Egon Schulz<sup>1</sup>

[eiman.mohyeldin@siemens.com](mailto:eiman.mohyeldin@siemens.com)  
[markus.dillinger@siemens.com](mailto:markus.dillinger@siemens.com)  
[Jesse.Luo@siemens.com](mailto:Jesse.Luo@siemens.com)  
[fahrmaier@informatik.tu-muenchen.de](mailto:fahrmaier@informatik.tu-muenchen.de)  
[dornbusc@informatik.tu-muenchen.de](mailto:dornbusc@informatik.tu-muenchen.de)  
[egon.schulz@siemens.com](mailto:egon.schulz@siemens.com)

<sup>1</sup>Siemens AG, St. Martin str. 76, D- 81541 Munich, Germany

<sup>2</sup>Technical University of Munich, Boltzmann str.3, D-85748 Garching, Germany

## (b) Subject Area: WG 6: Reconfigurability

System reconfiguration (terminal and networks) is the key issue in the introduction of world wide mobility and has been a core research issue for Software defined Radio (SDR). In the early stages of SDR, the focus of the research topics was on the hardware, dual mode, filters, wide range ADC converters, etc. However hardware technology has been improved and most of the shortcomings have been solved and amended. Recently other issues in the domain of software reconfiguration has been identified such as mobility awareness of application and their adaptations to different available radio access technologies (RAT's) [1].

There are several aspects and perspectives of the reconfiguration scenario:

- Type of the reconfiguration (total, partial, components/modules, layers)
- Entities involved in the reconfiguration (Networks entities, terminals)
- Reconfiguration time frame
- Type of the software to be downloaded and necessary to complete the reconfiguration (operating systems, drivers, radio software)

In this contribution, the reconfiguration support in wireless systems with emphasise on concurrent reconfiguration process in a given system is studied. The contribution exploits the capabilities of reconfiguration technologies for optimised dynamic radio access network reconfigurations. New perspectives in the specific areas of reconfigurable networks and their supporting functions are given.

## (c) Objectives of the required research

The architecture of wireless systems is designed according to a layered structure with a minimum interdependency, allowing for easy reuse and building of infrastructures, which can be shared among a number of different modules on the same layer (e.g. applications on the top layers can share a common platform of lower layer services). The communications between layers are accomplished either via Service Access Points (SAP) or open platform implementation, i.e. based on application programming interface (API's). Service Access Points (SAP's) provide access via set of primitives that enables access to a given layer by its neighbouring layer, i.e. the next layer up or down in the hierarchy. Application Programming



# Wireless World Research Forum (WWRF)

WG6: Reconfigurability



Interfaces (APIs) are used to allow interaction between two software modules or services inside a terminal, as well as with external entities.

Reconfiguration support in wireless systems means that there can be a number of concurrent reconfiguration processes in a given system at the same time. For example reconfiguring on the lower layers of the system (physical, data and network layers) implies a change of the network type (e.g. inter RAT handover) used for communication depending on the current demand of bandwidth from the applications on the higher layers. Similarly application could reconfigure its functionality or quality of service depending on the currently available bandwidth (not necessary inter RAT handover). It is obvious that parallel reconfigurations on different protocols layers are leading to conflicts hence it is required that several techniques should be developed aiming at optimising and providing:

- Efficient signalling of reconfiguration decisions
- Sophisticated reconfiguration mechanisms handling agreed reconfigurations
- Orthogonal reconfiguration management layer

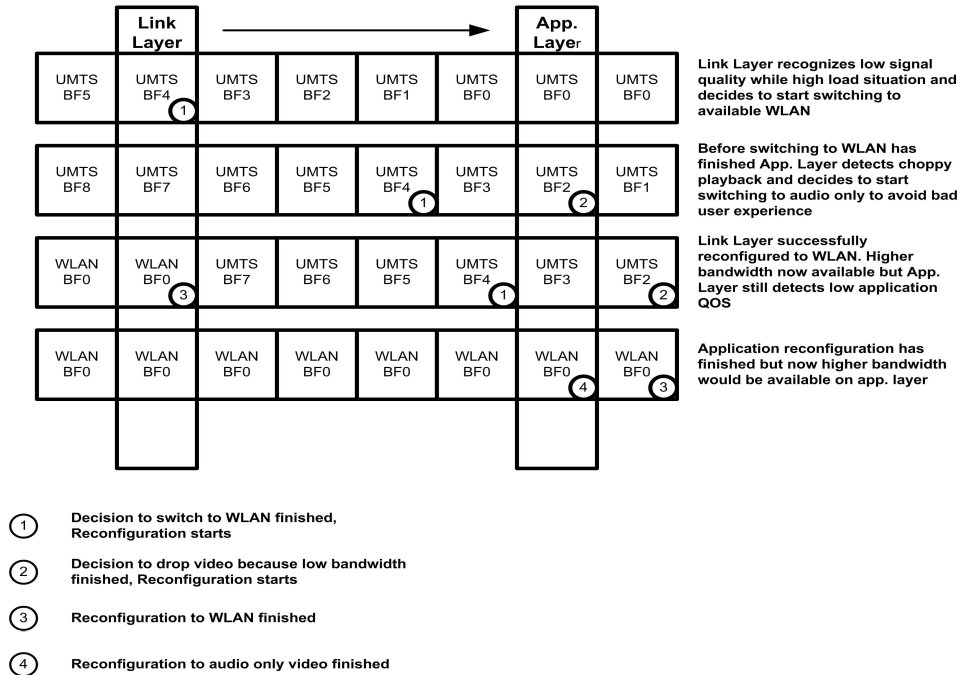
## **(d) State of the art in the area**

Two approaches for accomplishing communication between layers namely via Service Access Points (SAP) or open platform implementation, i.e. based on application programming interface (API's) are discussed. These SAP's provide access via set of primitives that enables access to a given layer by its neighbouring layer, i.e. the next layer up or down in the hierarchy. However SAP's lacks of flexibility, in other words they don't support changes, alteration or improvement within the protocol stack [1]. Application Programming Interfaces (APIs) are used to allow interaction between two software modules or services inside the terminal, as well as with external entities. Different APIs are employed to reconfigure different parts of the terminal [2]. Both approaches neither have nor consider the possibilities of concurrent reconfigurations on different layers of the protocol stacks.

## **(e) Possible approach**

The concepts of layered service architectures and reconfiguration/adaptations are independent of each other, moreover the reconfiguration logic is depending on information that is either passed from one of the direct neighbour layers or generated inside the layer under adaptations. This implies that reconfigurations taking place in non neighbouring layers are isolated from each other. They can only communicate indirectly by propagating results of their reconfiguration to the next upper or lower layer which again needs to propagate the information to the next layer and so forth until it reaches the target layer. Depending on the decision logic used in one layer there certain delay is generated between the reconfiguration/adaptation decision which is embedded in the data stream and the impact of this reconfiguration decision on the stream that is passed to the next layer. Several types of collision negative interaction effects between adaptations that take part on several layers simultaneously are resulting from non neighbouring layers isolation (so called reconfiguration resonance effects).

## Reconfiguration Propagation Problem



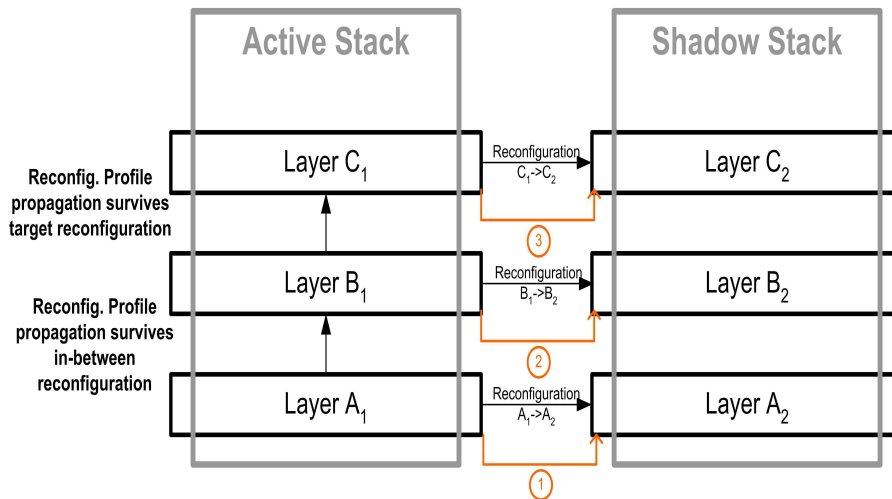
**Figure 1: Reconfiguration Propagation Problem**

Figure 1 shows an example for the Reconfiguration Propagation Problem (RPP). reconfigurations deployment spans some time (Time span 1-3 in Figure 1) and since non neighbouring layers have to communicate by propagating the information through all intermediary layers, a conflicting reconfiguration will occur before the results of the reconfiguration in certain layer (e.g. lower layers) is communicated to the target layer (e.g. application layers). Resulting in an unnecessary reduction or adaptation of the application QoS or an immediate second reconfiguration following the first one which could also lead to alternating reconfiguration resonance e.g. link layer start reconfiguring to function in WLAN mode while the application layer reduces its QoS and adapt the services. In the meantime the link layer recognizes low usage of the available Bandwidth/throughput and switches to another RAT (e.g. GPRS), before the other mode is totally deployed, the application layer recognizes that the link layer has switched to WLAN mode and activates and update the service to utilize this throughput (e.g. switch from audio only to enhanced video transmission).

The *Reconfiguration Propagation Phenomenon* arises also in scenarios with seamless reconfiguration in which the changes and new configuration are prepared in the shadow mode while continuing the current session in the active mode. Figure 2 depict implicit cross layer reconfigurations that utilize the principle of active/shadow chain reconfiguration. However the reconfiguration signalling information will not be lost while concurrently reconfiguring the target or in-between layers but blind spots in synchronization will results. In other words dependent layered can not be reconfigured at the same time unless a possibility for shadowing the reconfiguration is feasible, meaning the old reconfiguration synchronization logic can be active until the new layer configuration together with the new reconfiguration synchronization logic is in place and ready to take over. Even if this is the case, without the 2-phase-reconfiguration there is still a small blind spot while handing over the session information from the old to the new layer

configuration.

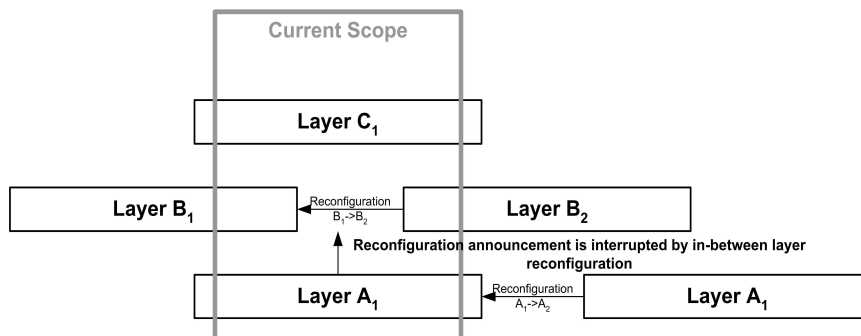
### Implicit Cross Layer Reconfiguration



**Figure 2: Active/Shadow Reconfiguration**

In hard reconfiguration scenarios the RPP effect is even worse, since each reconfiguration is also connected with an interruption of the current session.

These interruptions are affecting the propagation of reconfiguration decision and therefore interrupt the flows of the signals (see Figure 3, Reconfiguration Propagation Interruption (RPI)). This effects result from the fact that each of the in between layers or even the target layer are subject to reconfiguration while the reconfiguration of the source layer is signalled. Therefore the information could be lost or arrived at the target reconfigured layer while having the same reconfiguration signalling protocol with different semantic interpretation of the signalled content. While the latter can be prevented by standardization there needs to be extra effort to establish an indirect and buffered communication between two neighbouring layers to prevent the reconfiguration signalling information from getting lost in a reconfiguration that takes place in between the two layers that exchange the information.

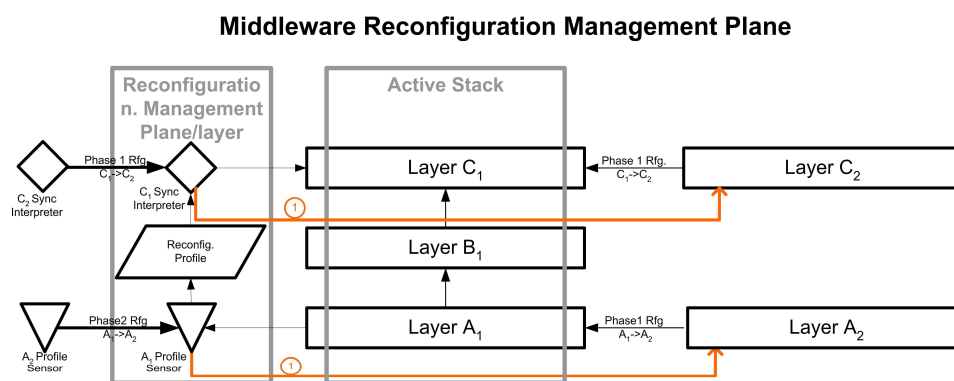


**Figure 3: Reconfiguration Propagation Interruption**

To further facilitate the usage of specialized reconfiguration mechanisms, an extension of the approach of an indirect communication mechanism for propagating reconfiguration data using a bypass channel on a middleware for cross layer reconfiguration management is proposed.



Existing approaches allow for indirect communication between layers which is robust against concurrent reconfigurations of the target or in-between layers but still suffer from blind spots in any part of the reconfiguration. However sophisticated synchronization mechanisms are not offered since they rely on generic reconfiguration logic in the middleware that is a common denominator without taking into consideration the application dependent reconfiguration logic. Therefore the solution for the RPI problem is by using a common meta model of reconfiguration on the same reconfiguration management plane that is used to decouple reconfiguration propagation between layers (see Figure 4 bottom).



**Figure 4: Middleware Reconfiguration Plane**

With this solution, the reconfiguration signalling and decisions on each layer are routed over the management plane. Layers can then be reconfigured using a two-phase-protocol. In the first phase the layer is switched over but the reconfiguration decision logic for this layer still remains active on the management plane. This is equivalent to have a shadowed reconfiguration of only the reconfiguration logic part of the current layer, thus gaining the benefit of shadowing also in situations where no complete shadow reconfiguration is possible. In the second Phase after successful deployment of the current layer reconfiguration, the new layer configuration can change the previous reconfiguration decision logic in this layer (every configuration is associated with certain reconfiguration logic). With these mechanisms the logic can incorporate the reconfiguration characteristics not only for a given layer but also for the current configuration of this layer.

**(f) Time frame to get the expected results**

The research described is part of the IST Integrated Project for End-to-End Reconfigurability (E<sup>2</sup>R) [6], the time frames for the research are cast into three phases, the initial phase will have duration of 2 years and it is expected.

**(g) Example**

To explain clearer the effect of the reconfiguration decision logic an example is given:

Comparing two different application scenarios, one for using a browser to surf the internet and another of using a video conference software, it is obvious that while being on the same layer the two applications (i.e. two possible configurations of the application layer) have different usage characteristics and therefore different requirements from the user quality of service perspective:



# Wireless World Research Forum (WWRF)

WG6: Reconfigurability



- The web browsing application has a discrete bandwidth spectrum usage characteristic. This means while the user is reading already downloaded web page, the current bandwidth and therefore any reconfigurations on lower layers are not important. There is no cross layer reconfiguration dependency and hence no tied reconfiguration management actions are necessary. On the other hand reconfiguration can only occur immediately before a new page is loaded and loading time is usually quite short. Therefore the decision can be based on the current state even if there is a high probability for a concurrent reconfiguration on the link layer. Synchronization of colliding reconfigurations can be just a simple timeout mechanism. If the page is not finished loading in a given time for whatever reason the page download is simply resumed or repeated with a lower application quality (e.g. without pictures)
- The video conference has a continuous bandwidth characteristic. The application layer and its associated reconfiguration logic need to take into consideration the reconfiguration profile of the lower layers. It is not advisable to handover/reconfigure to a WLAN network while driving at 120km/h just to transmit 1 sec of video and then again falling back to audio only. From the users quality point of view this could even be more annoying than having no video at all.

## **List of References**

- [1] W. Tuttlebee (Editor). Software Defined Radio. Origins, Drivers and International Perspectives, Wiley Series in Software Radio, John Wiley and Sons, Ltd., 2002
- [2] J.Sessler et al « Recommendations for API definition and management of core software in terminals”, Jan 2004, IST-2001-34091 SCOUT, <http://www.ist-scout.org>
- [3] End-to-End Reconfigurability (E<sup>2</sup>R), IST-2003-507995 E<sup>2</sup>R, <http://www.e2r.motlabs.com>