AROMA IST-4-027567

D16

Integrated Testbed

Contractual Date of Delivery to the CEC: 30-06-2007

Actual Date of Delivery to the CEC: 6-07-2007

Editor: Dev Pramil Audsin (KCL)

Author(s): See list

Participant(s): UPC, KCL, PTIN

Workpackage: WP4

Est. person months: 49

Security: Public

Nature: Prototype/Report

Version: 001

Total number of pages: 56

Abstract:

This deliverable describes the integrated test-bed developed by the AROMA project. The HW/SW tools and the integrated protocol suite for mobility, radio resource and QoS management proposed in the project are presented. In addition a user guide has also been included.

Keyword list: Integrated test-bed, Integration methodology, AROMA Demonstrator

DISCLAIMER

DOCUMENT HISTORY

Date	Version	Status	Comments
24-05-2007	001	Int	First draft of ToC for comments
24-05-2007	002	Int	PTIN comments to ToC
02-06-2007	003	Int	UPC comments to ToC and first contribution
04-06-2007	004	Int	PTIN first contribution
11-06-2007	005	Int	UPC second contribution
15-06-2007	006	Int	PTIN contribution on MPLS tunnel creation section and ToC change
21-06-2007	007	Int	KCL contribution to Introduction , section 5
27-06-2007	008	Int	UPC third contribution
29-06-2007	009	Int	KCL second contribution to Executive summary, Introduction, section 5 and 6 , conclusion , appended section 7 provided by UPC
01-07-2007	010	Int	UPC comments
01-07-2007	011	Int	KCL answers
02-07-2007	012	Int	PTIN Answers
02-07-2007	013	Int	KCL Answers
02-07-2007	014	Int	UPC input
02-07-2007	015	Int	Version for PCC review
06-07-2007	001	Apr	Approved version to be delivered to the U.E.

Authors List

Dev Pramil Audsin (KCL)
Anna Umbert Juliana (UPC)
Francisco Bernardo (UPC)
Łukasz Budzisz (UPC)
Nemanja Vucevic (UPC)
Miguel López Benitez (UPC)
Ricardo Azevedo (PTIN)
André Oliveira (PTIN)

EXECUTIVE SUMMARY

The scope of this deliverable is to provide details of the integrated AROMA testbed suite, in particular the protocols involved and the inter-process communication procedures between the various entities involved in the testbed are described.

The testbed definition and presentation is not the intent of this deliverable. The information presented in deliverable “D07 - Testbed Specification” should be used as the reference to understand the testbed architecture and available functionalities. However this document describes the testbed functionalites that were not enough defined in D07 and therefore it can be considered as an extension to D07.

The integrated AROMA testbed can be used as a demonstrator to test the protocols compliant with the 3GPP, IEEE and IETF specifications. After reading the user manual provided in this document, interested organizations or persons can use the AROMA testbed protocol stack out of the box. Also, the architecture of the testbed is modular and hence extensions or modifications to one element can be made easily with very little modifications to other elements of the stack. This document provides the architecture of the message structure passed between the elements of the protocols for the researchers to further enhance or replace the existing protocols with their own protocols and test the operations with respect to other elements of the protocol stack.

In summary, this document describes the integrated AROMA testbed and the supported procedures. Also a section is dedicated for including the testbed user guide.

Table of Contents

1 INTRODUCTION

The purpose of this document is to describe the integrated testbed framework developed in the AROMA project and provide clear information of its capabilities by means of the final version of the implemented procedures, which were defined as a proof of concept of the relevant aspects identified, and a useful User Guide giving additional details of its possibilities. Therefore, this deliverable provides a complete view of the integrated AROMA testbed understood as a flexible HW/SW tool able to test in a realistic but easy way the different radio resource, mobility management and QoS Management algorithms.

The AROMA testbed framework tries to reproduce in a realistic way a B3G heterogeneous network, that includes RATs, (UTRAN, GERAN, WLAN), interfacing a common Core Network based on Diffserv and policy-enabled networking (PBN) with improved mobility aspects and a new framework for the E2E QoS Management. In addition to all these elements the AROMA testbed incorporates the capacity to evaluate the QoS experienced by the user when using real applications under controlled conditions of the used RAT and the CN.

The document is organised as follows. Section 2 is devoted to the description of the AROMA testbed architecture, clearly identifying the two major components, the Radio Access Domain and the Core Network Domain and their interactions. This information is relevant here to easily understand the testbed framework as a first step before identifying the elements details, already described in previous documents and the remaining ones that must be detailed here, basically all of them related with the final version of interfaces and signalling messages. Section 3 describes the framework related to the CRRM emulation and focuses on the implementation details of the CRRM and the CRRM-WQB interactions. Section 4 describes the details of the implemented End-to-end QoS signalling and the interfaces involved. Section 5 describes the interactions between the mobility manager and the BB in the CN. This section also covers the message formats involved in the interactions of the mentioned entities. In section 6, all the supported procedures are explained in detail. Chapter 7 is the AROMA testbed User Guide followed by conclusion.

2 TESTBED OVERVIEW

As described in D07, the testbed architecture can be divided in two main parts: the heterogeneous Radio Access Network (RAN) and the Core Network (CN). The first part of this section details the emulated heterogeneous RANs considered in the AROMA project, and also the concerned Radio Resource Management (RRM) and the Common Radio Resource Management (CRRM). The second part of this section describes the CN architecture with the Diffserv/ MPLS QoS enabled traffic engineering. The remaining section details the Communications Manager (CM) and the other coordinating software modules for integrating all the protocols running in both the RAN and the CN to form an integrated testbed suite.

2.1 Radio Access Networks (RANs)

The AROMA testbed considers the following radio access networks (RANs):

- UMTS Terrestrial Radio Access Network (UTRAN),
- GSM/EDGE Radio Access Network (GERAN), and
- Wireless Local Area Network (WLAN).

These RANs are emulated in the same way than in EVEREST testbed, a full description of the inherited EVEREST testbed can be found in EVEREST deliverables D06 [1], D12 [2], and D18 [3]. The innovative aspects implemented in AROMA are the inclusion of High Speed Downlink Packet Access (HSDPA) in UTRAN, and the IP transport. Both implementations are described following.

2.1.1 High Speed Downlink Packet Access in UTRAN

The UTRAN emulation modules of the legacy IST-EVEREST testbed have been upgraded with the HSDPA FDD technology in accordance with 3GPP Release 6 specifications [4]-[8]. These modules were initially designed to cope with the following goals, which have been preserved in the final implementation:

- **Support for live users as well as full emulated users.** The emulator was designed to reproduce in real time the behaviour of a relative large amount of active users (around several thousands of users depending on scenario and traffic generation) within a given UTRAN service area. Traffic generation for most of these users was internally accounted in the emulation by means of traffic modelling, and live traffic of the user under test was processed by the emulator as well. Although emulated and real users have also been modelled in the current HSDPA implementation, an approximated real-time emulation approach has been adopted, as explained later in this section.

- **Emulation of the transmission chain between the user equipment (UE) and the radio network controller (RNC).** The emulator accounts for the main characteristics of the radio interface as well as radio resource management functions. The different functions performed at each level of the protocol stack, including those related to HSDPA, have been faithfully modelled in accordance to 3GPP specifications. Physical layer emulation has been addressed by means of tables obtained from extensive off-line link level simulations in order to reduce computational requirements while preserving realistic behaviour.

- **Execution of radio resource management (RRM) functions and support for common radio resource management (CRRM) capabilities**. RRM functions implemented in the testbed include essential functions like admission control, congestion control, radio resource allocation, handover management, outer and inner loop power control and transmission parameters management. Some of these functions have been modified in order to support the HSDPA operation. Although only a single user under test is running real applications, RRM algorithms are applied indistinctly over all the traffic generated by the rest of users emulated in the demonstrator. On the same way, support for CRRM functions is achieved by allowing the communication between the UTRAN emulator and the Wireless QoS Broker.

- **Support for different propagation scenarios and mobility models**. The definition of the scenarios takes into account the cell site deployment, radio environment and mobile distribution

and movement. The UTRAN emulator allows changing the considered scenario by means of the concept of the propagation matrix explained in [2]. These features have been maintained for HSDPA.

To support the HSDPA technology, changes are required neither in the existing hardware infrastructure (described in section 3.1 of [9]) nor the current software environment (described in section 3.2 of [9]). Only the UTRAN downlink module has been modified in order to include HSDPA transmissions in the testbed, and some minor modifications have been required in the CRRM module in order to account for the new UTRAN downlink capabilities in some CRRM procedures. In the rest of this section, the main features of HSDPA that have been taken into account in the emulation model as well as specific implementation aspects are described.

2.1.1.1 New channels required for HSDPA operation

Several new channels are introduced in Release 5 3GPP specifications for HSDPA operation [4]. For user data the high speed downlink shared channel (HS-DSCH) and the corresponding physical channel (HS-PDSCH) are defined. The HS-DSCH is a code- and time-shared channel operating at a fixed spreading factor equal to 16. Multicode operation is allowed, meaning that several codes can be simultaneously assigned to a single user, depending on resource availability, service priority, and user's terminal capabilities (12 different categories are defined in [10], allowing the utilisation of up to 5, 10, and 15 simultaneous physical codes). Transmissions over the HS-DSCH are performed following a 2 ms transmission time interval (TTI). This time interval is defined as the inter-arrival time of transport block sets (TBS), and is equal to the periodicity at which a TBS[1] is transferred by the physical layer on the radio interface [12]. This time interval also determines for instance the frequency of the scheduling or adaptive modulation and coding (AMC) decisions. For the HS-DSCH channel, fast power control and soft handover are not supported. All these features have been taken into account and are fully implemented in the UTRAN emulator.

For the signalling associated to HSDPA operation, two new channels are introduced in Release 5: the high speed shared control channel (HS-SCCH) in the downlink and the high speed dedicated physical control channel (HS-DPCCH) in the uplink.

The HS-SCCH is transmitted with an offset of two slots (4/3 ms) with respect to the HS-DSCH, as shown in Figure 1, in order to enable the HS-SCCH to carry time-critical signalling information (the codes assigned by the scheduler and the modulation employed) required by the UE to demodulate the correct codes in the HS-DSCH. If there is no data on the HS-DSCH, then there is no need to transmit the HS-SCCH. When there is a need to have code multiplexing, i.e. two or more users transmitting in parallel on the HS-DSCH in a given TTI, then more than one HS-SCCH needs to be transmitted. From the network point of view, there may be a high number of HS-SCCHs, but a single user terminal may consider at most four HS-SCCHs. Each HS-SCCH consumes a code with a constant spreading factor equal to 128.

The HS-DPCCH is used to carry uplink feedback information. This channel informs the Node-B whether a packet has been successfully decoded or not. This acknowledgement is always sent when there has been a correctly decoded HS-SCCH received in the downlink. A channel quality indicator (CQI) is also reported by the UE in the HS-DPCCH in order to inform the scheduler in Node-B about the maximum data rate it expects to be able to receive at a given point in time. CQI information is sent with every ACK or NACK of HS-DSCH data, in addition to periodic reporting whose period is determined through higher layer signalling by the system parameter *k* (see for example [13]). The HS-DPCCH is a dedicated channel. Therefore, each user accessing the HS-DSCH sends one HS-DPCCH in the uplink direction. Each HS-DPCCH consumes a code with a constant spreading factor equal to 256.

In addition to these two signalling channels, the fractional dedicated physical channel (F-DPCH) was introduced in Release 6 specifications to cover for operation when all downlink traffic is carried on the HS-DSCH channel. In Release 5, HSDPA operation requires a DCH channel associated to each user

[1] The transport block set size (TBSS) for transmissions over the HS-DSCH is always equal to one transport block (TB) [11]. The adjustable parameter in this case is the TB size, i.e. the number of bits transmitted in each TB.

receiving information through the HS-DSCH, which consumes a significant amount of code space and introduces a considerable amount of overhead, especially at lower data rates. Basically, the F-DPCH is a stripped-down version of the DPCH, where only the transmission power control (TPC) field is kept. An F-DPCH uses a code with a constant spreading factor equal to 256, and can be shared among several users, up to ten different users with different frame timing, thus reducing code space utilisation for the associated DCH for users with all services mapped to the HS-DSCH. The utilisation of the F-DPCH has some restrictions. For instance, it cannot be used with services requiring data to be mapped to the DCH, such as AMR speech call or circuit switched video.

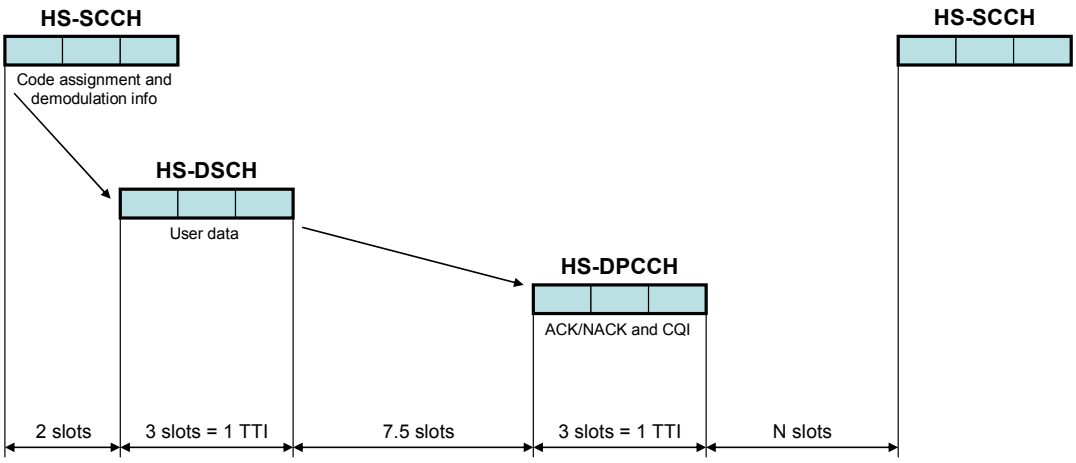


Figure 1. Timing relationship between HS-DSCH, HS-SCCH and HS-DPCCH.

The timing relationship between HS-DSCH, HS-SCCH and HS-DPCCH is shown in Figure 1. Only transmissions over the HS-DSCH channel are fully emulated in the AROMA testbed. Transmissions over HS-SCCH, HS-DPCCH and F-DPCH channels are not emulated. However, the functionalities of these channels are preserved in the emulation model and the existence of such channels is taken into account in terms of power and code space consumption.

2.1.1.2 New techniques introduced in HSDPA

The HSDPA concept introduces new adaptation and control mechanisms to enhance downlink peak data rates and spectral efficiency. The HSDPA concept is mainly characterised by the introduction of three key techniques for the HS-DSCH: adaptive modulation and coding (AMC), hybrid automatic repeat request (HARQ), and fast scheduling. These three principles rely on fast adaptation to changing radio conditions and the corresponding functionality is therefore handled by Nodes-B instead of RNCs, which allows for lower response times. These features are supported, with minimal impact on the previous existing radio-interface protocol architecture [14], by the introduction of a new MAC sublayer, known as MAC-hs, located at the Node-B. The MAC-d [15], RLC [16] and PDCP [17] layers remain unchanged with respect to previous releases. These three techniques are briefly summarised in the following subsections, highlighting some specific implementation aspects.

2.1.1.2.1 Adaptive Modulation and Coding (AMC)

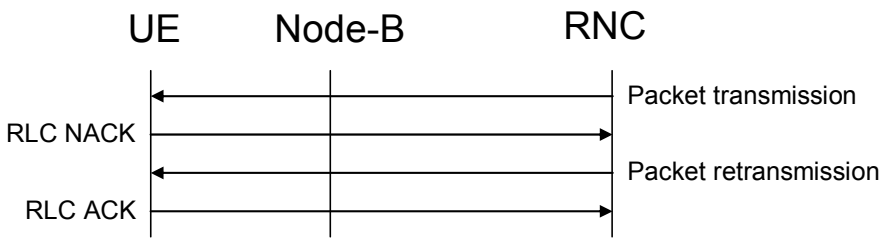
In a cellular radio communication system, the radio channel conditions experienced by each user can vary significantly in time due to many factors. In Release 99 DCH, a fast power control mechanism is used in order to compensate for these variations in the radio channel conditions. In HSDPA, concretely in the HS-DSCH, a different approach has been adopted in order to achieve the required energy per bit. Variations in channel quality conditions are controlled by adjusting data rate while keeping transmission power constant. A terminal experiencing good link conditions will be served with a higher data rate than a terminal in a less favourable situation. To support the different data rates, a wide range of combinations of channel coding rates, modulation formats (QPSK and 16QAM), and simultaneous parallel codes is supported. These transmission parameters can be adapted at a fast pace, once every 2 ms HS-DSCH TTI. This relatively short TTI allows the AMC mechanism to reasonably track rapid variations on the downlink channel quality. To this end, the CQI value reported

by the UE can be used by the MAC-hs to decide the optimum transmission parameters. Each CQI value is related to a given combination of transport block size, modulation and number of parallel channelisation codes. The full combinations can be found in [7], where different tables for the different existing UE categories are provided. Some categories of UE may not support large amounts of channelisation codes, or high order modulation (16-QAM). The CQI reported by the UE is the highest supported value of CQI that can result in a BLER not exceeding a target value equal to 10%. The CQI value is computed based on instantaneous measurements of the CPICH during a predefined 3-slot interval that ends one slot before CQI transmission. The measured value is then compared with a set of tables relating BLER and C/I for each possible CQI value. These tables have been obtained by means of extensive off-line link-level simulations. The highest supported value of CQI that can result in a BLER not exceeding a target value equal to 10% is then reported by the UE to the Node-B.

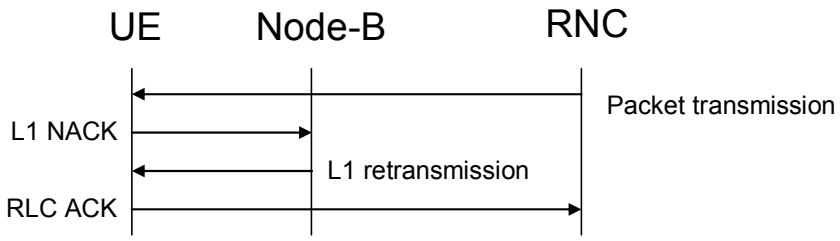
The CQI should be considered as a recommendation from the UE about the block size or data rate the MAC-hs in Node-B could allocate to that UE. This means that Node-B is not obliged to act basing its decisions on this information, although it could take such information into account when deciding. In the current implementation, CQI values reported by the UE are mapped in MAC-hs to appropriate transmission parameters settings following the tables provided in [7]. It is worth mentioning that a three-TTI delay exists between the actual C/I conditions on which the CQI is based and the time instant in which transmission parameters are updated according to the reported CQI (one TTI for measuring the experienced C/I, another TTI for reporting the CQI value to Node-B, and a third TTI for receiving and processing the reported value in Node-B). This aspect has been modelled as a simple delay.

2.1.1.2.2 Hybrid Automatic Repeat reQuest (HARQ)

The retransmission functionality in Release 99 is implemented as a conventional ARQ scheme at RLC level. When a data packet is not correctly received by the UE, it is discarded and a negative RLC acknowledgement is sent requesting a retransmission, as shown in Figure 2a. Then, a retransmission must be sent again from the RNC. The retransmitted packet is identical with the first transmission. With HSDPA, the packet is first stored in the buffer in the Node-B and kept even if it has been sent to the UE. In case of packet decoding failure, the UE stores the received data and requests a L1 retransmission, which automatically takes place from Node-B as shown in Figure 2b. The received retransmission is soft combined with previous transmissions of the same information bits. If this combined signal is unsuccessfully decoded again, further retransmissions and soft combinations occur until a successful decoding process takes place or the maximum number of allowed L1 retransmissions is reached, in which case an RLC retransmission is requested. This approach reduces the need of RLC retransmissions from RNC, since RLC retransmission are required only in case the maximum number of L1 retransmissions allowed is reached or due to terminal mobility. Thus, retransmission latency is significantly reduced in HSDPA with respect to Release 99.



(a) Release 99 retransmission.



(a) Release 5 HSDPA retransmission.

Figure 2. Retransmission procedure in (a) Release 99 and (b) Release 5 HSDPA.

The retransmission strategy implemented is based on the stop-and-wait (SAW) protocol. SAW is one of the simplest forms of ARQ requiring very little overhead. In SAW, the transmitter operates on the current block until the block has been received successfully. Every TTI, the receiver attempts to decode the received block. To decide whether the block is received in error or not, a set of specific curves obtained by means of extensive off-line link-level simulations has been employed. The concrete curve applied depends on the CQI used for transmitting the block. For each one of the possible CQI values, a curve is used for the first transmission and different curves are used for different retransmissions, as shown in Figure 3 for the case of CQI 0 (similar curves are used for the other CQIs). As it can be appreciated, each retransmission could tolerate a radio quality lower than that of the previous transmission and still be able to achieve the desired target error probability. The aim of this type of modelling is to reflect the impact on the decoding result of the soft combining process, which increases the probability of a correct reception after each retransmission. If the block is decided to be received in error, the transmitter sends the information again. This process is repeated until the block is successfully decoded, and then the transmitter sends the next information block. Protocol correctness is ensured with a simple one-bit sequence number that identifies the current or the next block. As a result, the control overhead is minimal. Acknowledgement overhead is also minimal, as the indication of a successful or unsuccessful decoding may be signalled concisely with a single bit. Furthermore, because only a single block is in transit at a time, memory requirements at the UE are also minimised. Therefore, HARQ using a SAW mechanism offers significant improvements by reducing the overall bandwidth required for signalling and the UE memory. However, one major drawback exists: acknowledgements are not instantaneous and therefore after every transmission the transmitter must wait to receive the acknowledgement prior to transmitting the next block. This is a well-known problem with SAW ARQ. In a slotted system, at least every other slot must go idle even on an error free channel. In the interim, the channel remains idle and the feedback delay will waste at least half the system capacity while the transmitter is waiting for acknowledgments. *N* channel SAW HARQ offers a solution by parallelising the SAW protocol and in effect running a separate instantiation of the HARQ protocol when the channel is idle. As a result, no system capacity goes wasted since one instance of the algorithm communicates a data block on the forward link at the same time that other communicates an acknowledgement on the reverse link. However, the receiver has to store *N* blocks for this scheme. The optimum number of parallel SAW channels should be set to the minimum number of TTIs required by acknowledgements messages to be propagated through the radio link and be processed by the receiver, in order to assure a continuous flow of information while minimising

buffering requirements in the UE. The number of parallel SAW channels may maximally be 8, but in practice it will be around 4 to 6.

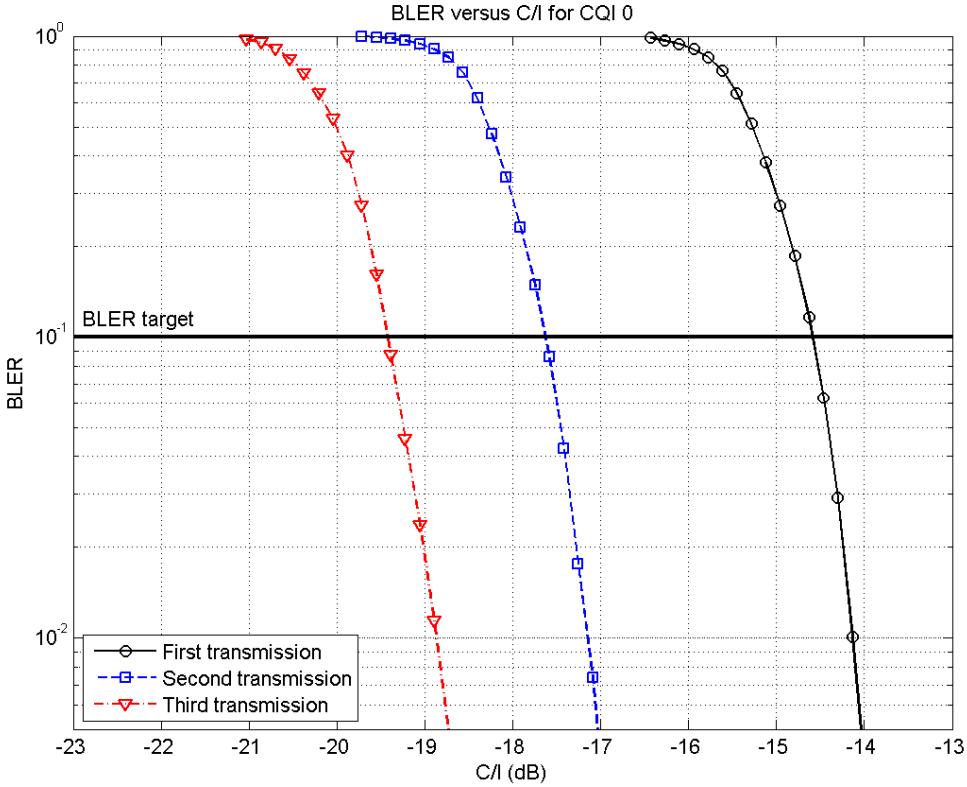


Figure 3. BLER versus C/I for CQI 0.

2.1.1.2.3 Fast scheduling

As the HS-DSCH is a shared channel, a scheduling entity is required in order to decide which user the channel is allocated to, and how the available codes are distributed among users in case of user multiplexing. This scheduling entity is placed in the Node-B for the HS-DSCH, not in the RNC. This feature, jointly with the reduced 2 ms TTI, allows a significant reduction in delay and retransmission latency.

The most basic scheduling strategy is known as round robin (RR). With RR, resources are assigned to users on a sequential and cyclic basis. Any knowledge of the experienced channel quality is not made use of, and user's QoS requirements are also neglected. This strategy offers the most fair throughput distribution among users. On the other hand, the Maximum C/I criterion allocates resources to those links experiencing the best instantaneous channel quality conditions. This approach maximises the cell throughput, but results in an unfair throughput distribution. Users located at cell border may be not served at all while users experiencing good transmission conditions, even with relatively little data to transmit, may monopolise the resources. This two scheduling strategies could be regarded as basic reference schemes. A practical scheduler should be in a middle point between these two approaches, taking into account the varying channel quality for each user and providing some degree of fairness. The current implementation has been designed to be able to incorporate new scheduling strategies.

2.1.1.3 Mobility aspects

As mentioned above, the HS-DSCH channel does not support soft handover. Therefore, although the associated DCH could be in soft handover, the HS-DSCH channel for a given user is always transmitted from one unique cell decided by UTRAN referred to as serving HS-DSCH cell [11].

If dedicated channels are operating in soft handover while HSDPA is in use, a variation could occur in the relative strength of the pilot signal received from the cells in the active set with no modification in the active set. This circumstance does not trigger actions for the DCH but may trigger a modification in the serving HS-DSCH cell, as show in Figure 4. In order to enable the UE to notify this situation to the RNC the measurement event 1D is defined in [18].

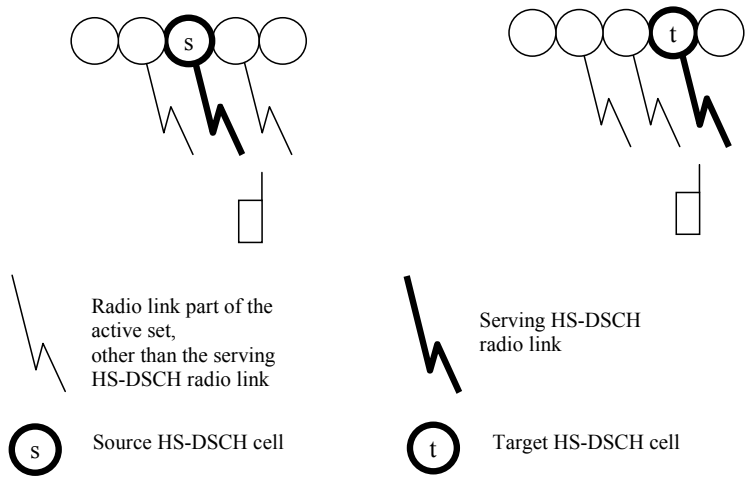


Figure 4. Serving HS-DSCH cell change [11].

In contrast to the example shown in Figure 4, a change in the serving HS-DSCH cell may require the active set to be updated if the target serving HS-DSCH cell is not in the current active set. In this case, some differences can be observed between the procedures defined in Releases 5 and 6. In Release 5, the cell first needs to be added to the active set before the serving HS-DSCH cell change can be performed. In Release 6, the situation has been changed to allow an active set update procedure to also modify the serving HS-DSCH cell.

Active set updates are decided and triggered by the CRRM module, and communicated to the UTRAN modules. Within the UTRAN downlink module, the serving HS-DSCH cell is assumed to be the cell with the highest received pilot among the cells of the active set.

2.1.1.4 Real-time emulation of the HSDPA functionalities

The testbed modules have been designed to operate using the Communications Manager (CM) software layer. CM may be understood as an abstraction layer that is in-between the hardware and/or operating system (if any) and the application (software modules). It hides to the application any aspect related to hardware or operating system of the possible heterogeneous platform compound of multiple machines. CM is in charge of other tasks such as enabling the execution of soft real-time applications through timing control of software modules, or gathering multiple forms of data provided by modules to be observed in real-time or post-processed after application has finished. For a comprehensive description of CM see Annex B of reference [2].

Each software module follows an internal 10 ms execution loop. The CM guarantees that the execution of the loop is synchronised in all the machines and, whenever tasks to be carried in a single execution of the loop last less than 10 ms, real time operation is maintained. CM also verifies that the execution loop of all modules does not exceed the 10 ms time interval. This execution period determines the maximum time resolution of the software modules. As it can be appreciated, this value is greater than the 2 ms TTI defined for HSDPA FDD in 3GPP specifications. A perfect real-time emulation of the HSDPA-related functionalities within the UTRAN downlink module is therefore not possible with a 10 ms execution period. A possible solution would consist on a reduction of the execution period of the testbed to 2 ms. However, this option has some problematic aspects. First of

all, it is worth mentioning that the software platform of the testbed is based on Linux, as described in section 3.2 of [9], and the short 10 ms execution period of the testbed is not completely independent of the Linux kernel process-scheduling time (see section B.6 in Annex B of reference [2]). However, the real problem is that this solution would imply that the computational load is increased around five times since the code of all software modules is forced to be executed five times every 10 ms, instead of once every 10 ms. Due to the high level of complexity and detail of the implemented testbed, the relatively high number of active users to be included in the demonstrations in order to force specific load conditions where the performance of the studied strategies could be evaluated, jointly with the fact that the considered scenarios may consist of a high number of cell sites, the feasibility of this solution is relatively uncertain if the system is to be emulated in real time.

A different approach has been adopted. The main idea behind this approach is illustrated in Figure 5. As it can be appreciated, the actual execution loop is maintained in 10 ms. In this loop, UTRAN-related functionalities are performed, including message processing from other modules, admission and congestion control, handover decision, traffic generation for emulated users, packet scheduling, TFC selection, inner and outer loop power control, and so on. However, a “virtual” execution loop of 2 ms, where all specific HSDPA-related operations are performed, is repeated five times within the 10 ms period. HSDPA-related specific functions include, for instance, HS-DSCH scheduling, emulation of transmissions over the HS-DSCH, AMC decisions, or ACK/NACK and CQI report generation. This virtual 2 ms execution loop is intended to emulate the 2 ms TTI defined for HSDPA FDD. In each iteration, all HSDPA-related specific functions are performed and the state of the system is updated before the next iteration. Five iterations of the loop correspond to five HSDPA TTIs with a total duration of 10 ms. At the end of the 10 ms execution loop, transmission statistics are updated in order be displayed in the Advanced Graphic Management Tool (AGMT). The transmission results, i.e. which packets have been successfully transmitted or not, and the experienced delay, are also updated at the end of the 10 ms execution loop. Note that although the HSDPA code is executed five times every 10 ms, the rest of the code is executed only once every 10 ms, keeping the total computational load at an acceptable level for the real-time emulation. This approach allows an approximated real-time emulation of the HSDPA-related functionalities, with practically negligible effects over the real-time performance of the transmitted traffic, while leaving the real-time emulation of the rest of UTRAN-related functionalities unaffected.

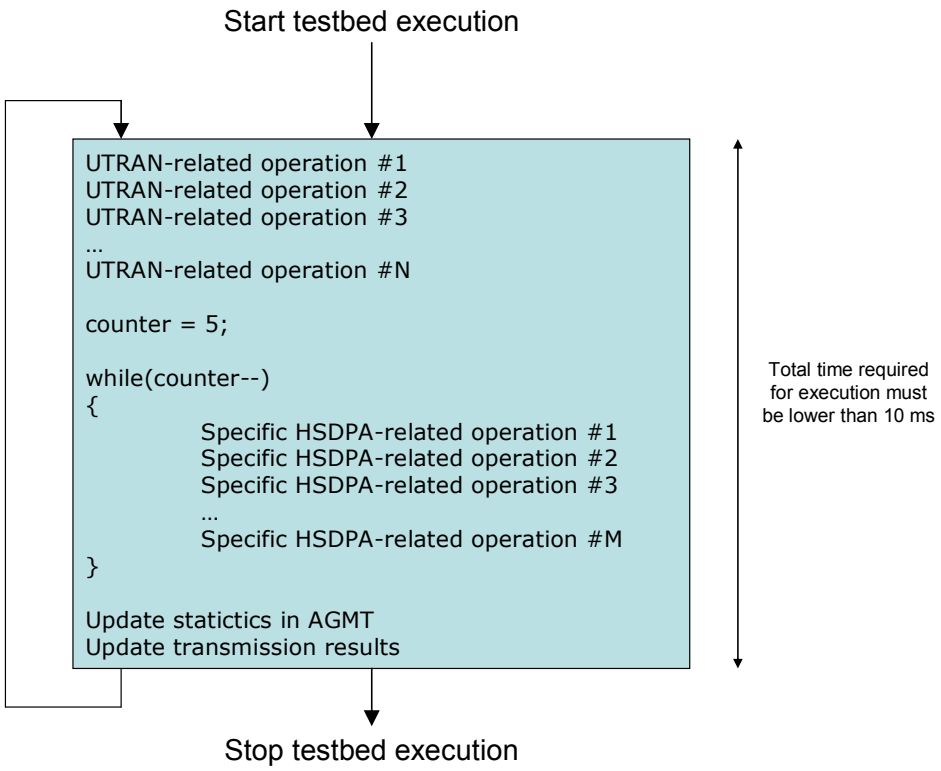


Figure 5. Approach adopted for approximated real-time emulation of HSDPA-related functionalities.

2.1.2 IP transport in the RANs

As it was mentioned before, presented testbed targets to keep up with the idea of all-IP networks. According to 3GPP specifications, an IP transport option is currently defined for I_{ub} in UTRAN [19]. Whereas TDM over IP (TDMoIP) solutions that are out of the scope of 3GPP, should be used to support the layer one (L1) interface (based on ITU Recommendations) defined in [20] for the A_{bis} interface in GERAN. The support of these interfaces implies a set of strong constraints over the IP-RAN transport so that QoS and traffic engineering (TE) solutions become mandatory. Therefore, the envisaged IP-RAN emulation model for the presented testbed accounts for delays and losses in transport network, obtained from off-line simulations, as shown in Figure 6. Existing I_{ub} interfaces for UTRAN (and A_{bis} in case of GERAN) are kept between base stations and radio network controllers (RNCs), but they are supported over an IP-based packet-switched network.

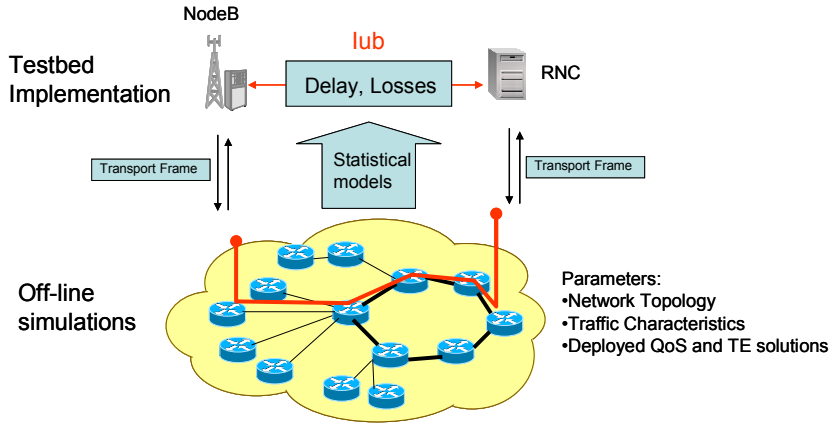


Figure 6. IP-RAN scenario considered in the testbed

As a consequence of such approach, a transport block (TB) can be lost at Node B due to radio conditions or because of transport network losses or excessive delays. In particular, a TB will be discarded at RNC if it arrives later than a defined delay (Max_Delay). In order to assure the validity of the proposed emulation model, the Max_Delay considered must be lower than the ACK delay at radio link control (RLC) layer minus the transmission time interval.
The statistical distribution for each base-station, and each DiffServ class would change depending on the traffic and user mobility pattern, the IP RAN topology chosen, the dimensioning of the network as well as the QoS and IP mobility architecture chosen (over-provisioning, pure DiffServ, or QoS routing).

2.2 Core Network

The Core Network (CN) of the AROMA testbed implements the required functionalities associated with the Mobile Operator Core Network. Thus it connects the UE from the Radio Access Network to the Server in a Backbone Network.

2.2.1 Core Network topology

The CN of the AROMA's project is not emulated; it is a real implementation of different nodes working as routers. Those nodes are Linux machines that work as routers with DiffServ and MPLS support.

The CN is based on a DiffServ domain with MPLS forwarding that determines and makes the flow to follow a pre-established LSP. With this, different type of traffic may experience different QoS. Packets are marked in the Ingress Routers, with the correct MPLS label corresponding FEC they belong to.

In order to test this different path, 3 new nodes have been included in the CN, obtaining with this a fish model topology slightly unbalanced, see Figure 7.

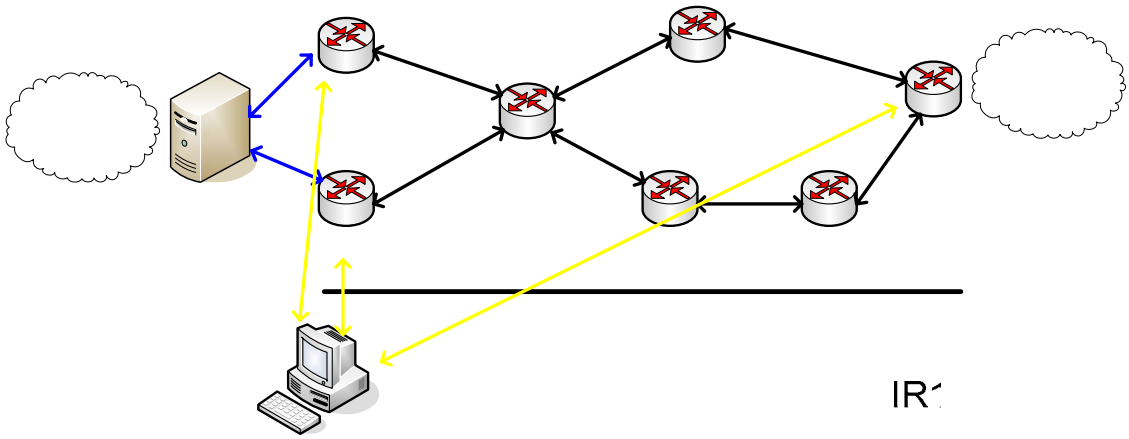


Figure 7. Core Network topology

For a more detailed description of the Core Network, see deliverable *D07 - Testbed Specification* [9].

2.2.2 DiffServ/MPLS architecture

Deliverable *D07 - Testbed Specification* [9], presents two different possibilities for MPLS and DiffServ association – L-LSP and E-LSP. For the AROMA's testbed the choice was the L-LSP approach, once it give us the needed funcionalitites with a better control over the MPLS suite software [21] (we faced some crucial problems when using E-LSP method of the MPLS open source package for Linux).

Despite the fact that DiffServ and MPLS are complementary methods (that can be put to work together), there are some points where functionalities overlapping exists.
DiffServ is based on the re-definition of the Type of Service (ToS) field of IP packets' header, to mark packets on one of a set of defined transport classes in one DiffServ domain.
In L-LSP approach, the LER (Label Edge Router) selects a label value not only by the packet destination address but also according to the DSCP (Differentiated Services Code Point) of the IP header (the corresponding FEC (Forwarding Equivalence Class). In the MPLS domain, packets may follow different paths according to their priority; as an example, low priority packets may follow a longer path than the high priority ones, making traffic engineering possible.
Next figure illustrates the explanation presented above.

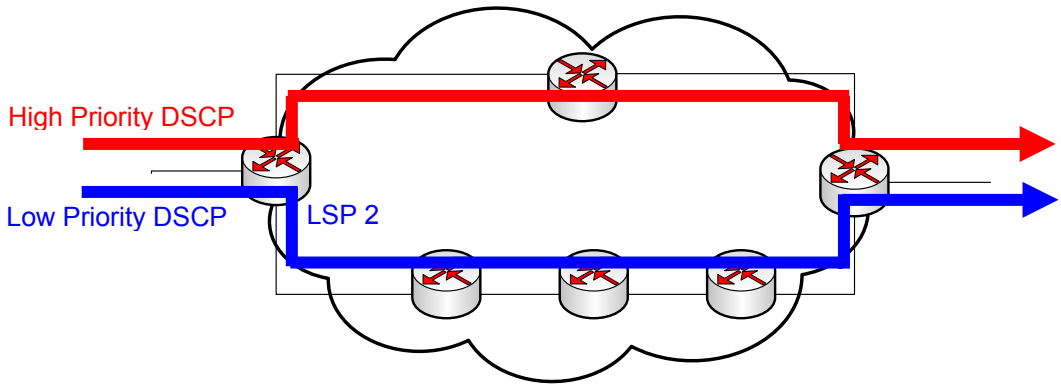


Figure 8 – DSCP marking

The integration of the previous scenario with the Bandwidth Broker (BB) is done each time a new request arrives the BB. At that time, BB should verify if the network capacity can receive the new requested flow, and if the answer is positive, should put a DiffServ filter in the LER, in order to correctly mark the new flow with the correct Label.

2.2.3 Core Network traffic coordinated generation

For the core network part, there is no emulation. The traffic passing through it is real IP traffic, and for the emulated users is generated by a modified iperf traffic generator [22]. Generation of traffic in IP network should be coordinated with traffic emulated in RANs. For this purpose, an aggregated traffic model has been used.
In each of the RANs, mean and variance of emulated traffic is calculated. After a predefined update interval this information is passed to traffic generator that controls up to 18 real flows entering the CN (Figure 9). For the easier control of traffic differentiation per class, as well as control of the attachment point (IR) of a certain RAT, separate flows are generated for different services in each RAT. The downlink flows are entering in ER and are directed towards corresponding IR. The uplink flows are entering one of the IR (each RAN is connected to one of the two IRs) and going toward ER. The IP packet sizes are predefined and fixed for certain class. These values may be changed as well as the update interval for the traffic generation.

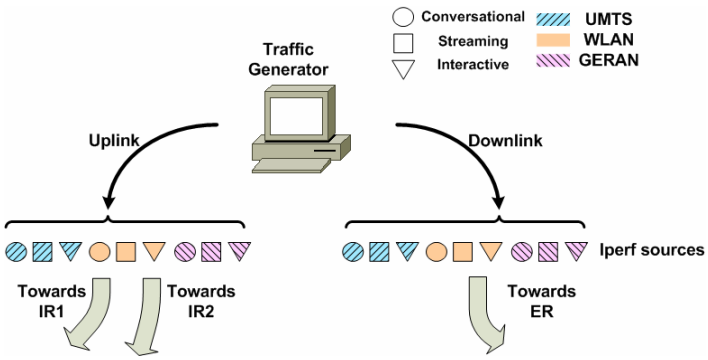


Figure 9. Traffic generator and flows

2.3 Software Environment

2.3.1 Operating System

The operating system selected for all the PCs in the testbed is Linux with a kernel 2.6.x. Any Linux distribution is suitable since required features mostly rely on kernel and not on installed software for each distribution.

2.3.2 Communications Manager

Communications Manager (CM) is a software tool mainly devoted to integrate software from different developers and manage its execution on a networked cluster of PCs with a Linux operating system. An application under CM is made of software modules running in parallel that are joined through interfaces adequately matched. It also offers means to such software to interact with the controlling entity of the system by means of dynamically modified parameters and statistics. Finally, CM controls the execution of the software in a slotted temporal framework to provide to the application the required timing. A detailed description of the CM can be found in Annex B of EVEREST D12 [2].

2.4 Application in the testbed

Table 1 contains a list of different applications that have been installed in the testbed for making the trials and tests. All of them are used under Microsoft Windows XP operating system.

Table 1: Applications used for the QoS tests

Purpose	Name	Version
Capturing		
Video Capturing:	**Camtasia Studio** [23]	v3.1.2
Audio Capturing:	**Microsoft Sound Recorder**	v5.1
Audio/Video Streaming		
Streaming Server:	**DSS** (Darwin Streaming Server) [24]	v5.5.1
	VLC (Video LAN) media player [25]	v0.8.5
Streaming Client:	**QuickTime Player PRO** [26]	v7.1
	VLC (Video LAN) media player	v0.8.5
Videoconference		
Video & Audio:	**Windows Net Meeting** [27]	v3.01
Video:	**VIC** (Videoconference Tool) [28]	v2.8ucl 1.1.6
Audio:	**RAT** (Robust Audio Tool) [29]	v3.0.35

In the Table 2 it can be seen how the different applications have been combined to provide the different services under test. These applications are used in testbed to measure QoS.

The support for web browsing is also an option that is enabled in the testbed. The most common applications for web browsing as Mozilla Firefox and Internet Explorer are used with Apache Web Server to make tests for the web browsing case.

Table 2: Services under test and applications

End to End Service	End to End Application		Capturing Application
	Server	Client	
Video Streaming	Darwin Streaming Server	QuickTime Pro	Camtasia Studio Recorder
	VLC	VLC	
Audioconference	RAT	RAT	Microsoft Sound Recorder
	NetMeeting	NetMeeting	
Videoconference	VIC	VIC	Camtasia Studio Recorder
	NetMeeting	NetMeeting	

2.5 Testbed Management Capabilities

The following management capabilities are supported in the AROMA testbed, inherited from the EVEREST testbed (see section 8 of EVEREST deliverable D12 [2]):

- Control the execution flow of the testbed (init, run, pause, restart options through CM facilities) and selection of the scenario to be demonstrated. The scenario here is understood as the needed software modules and their specific configuration involved in the demonstration of certain procedures or capabilities.
- Configuration of all the initialisation parameters required in the modules running in the test bed. Parameters needed to initialise the different software modules can be set-up centrally from the management platform.
- Collect and correlate logged data from the different modules of the demonstrator. Traces generated by the different modules share the same format in order to make easy the integration of

the data into a single file. Furthermore, a tool is available to post-process such traces allowing replaying the logged events in a dynamic way.
- Observe statistics during the execution of a demonstration (on-line representation/visualization).
- Change some configuration parameter during the execution of a trial to force a given situation (i.e. increase the number of users dynamically to analyse consequences over radio bearers established by the user under test).

The management platform developed during the ARROWS project and referred to as Arrows Graphical Management Tool (AGMT) was upgraded in order to include aforementioned functionalities in the EVEREST testbed. Under the testbed architecture the AGMT software should be installed in a separated PC, which must be connected to the public LAN of the whole testbed. The software of the AGMT has been developed in JAVA due to its facilities in programming graphical and networked applications and to allow the portability to different Operating System platforms. The acronym AGMT for the management tool has been left unchanged in EVEREST and AROMA although it now stands for Advanced Graphical Management Tool to account for the modifications carried out within those projects.

2.5.1 Software Modules managed by the AGMT

The AGMT framework in AROMA will be used to manage the following software modules:

- UTRAN Emulation modules
- WLAN Emulation modules
- GERAN Emulation Modules
- RAT Switching
- Wireless QoS Broker
- QoS Client
- Bandwidth Broker
- Edge Routers (PEP client and AR MM processes)
- Gateway (PEP Client and ANP MM processes)
- CN Traffic Generation
- Mobile Node (MM process)

All the software modules are linked to the Communications Manager Library in order to support the required functions that allow them to be managed through the AGMT.

2.5.2 Remote testbed Capabilities

The legacy EVEREST testbed was designed for remote operation (see section 9 of EVEREST deliverable D12 [2]). The AROMA testbed preserves this feature. In particular, the two main capabilities under consideration are the following:

- Remote testbed management and monitoring. This capability implies the remote operation of the testbed by accessing to the testbed management and monitoring tools. In this way, the testbed can be configured and the emulation of a given scenario controlled remotely from a machine outside the UPC premises. Nevertheless, applications of the user under test are locally executed in the testbed machines.

- Remote Execution of Applications. Applications tested over the testbed are executed in external machines other than the UE and Server machines in the UPC premises. So, traffic from a remote location is routed towards the testbed, passed through the testbed data path and then forwarded to its destination that could be located remotely as well. This usage is intended to test applications developed over specific platforms that can not be ported easily to the testbed machines.

Both usages can be supported simultaneously.

3 COMMON RADIO RESOURCES MANAGEMENT FRAMEWORK

Common radio resources management (CRRM) in AROMA follows the same approach as the EVEREST project. However two aspects have been improved:

- New RAT selection algorithms were added
- WQB-CRRM interaction was modified

3.1 RAT selection

RAT selection strategies choose the most suitable access network that each user should be connected to. Such decision can be taken at the session initiation (so called initial admission), as well as during the entire session lifetime (leading to a VHO). Both cases will be discussed in details in the following subsections.

3.1.1 Common Admission Control and Initial RAT Selection support

The admission control procedure currently implemented in the testbed allows distributing the users among the available RATs according to network usage, user preferences and operator preferences. AROMA project added also new RAT selection algorithms proposed in WP3 to improve CRRM capabilities offered by the testbed A functional organisation of testbed's admission control is depicted in Figure 10.

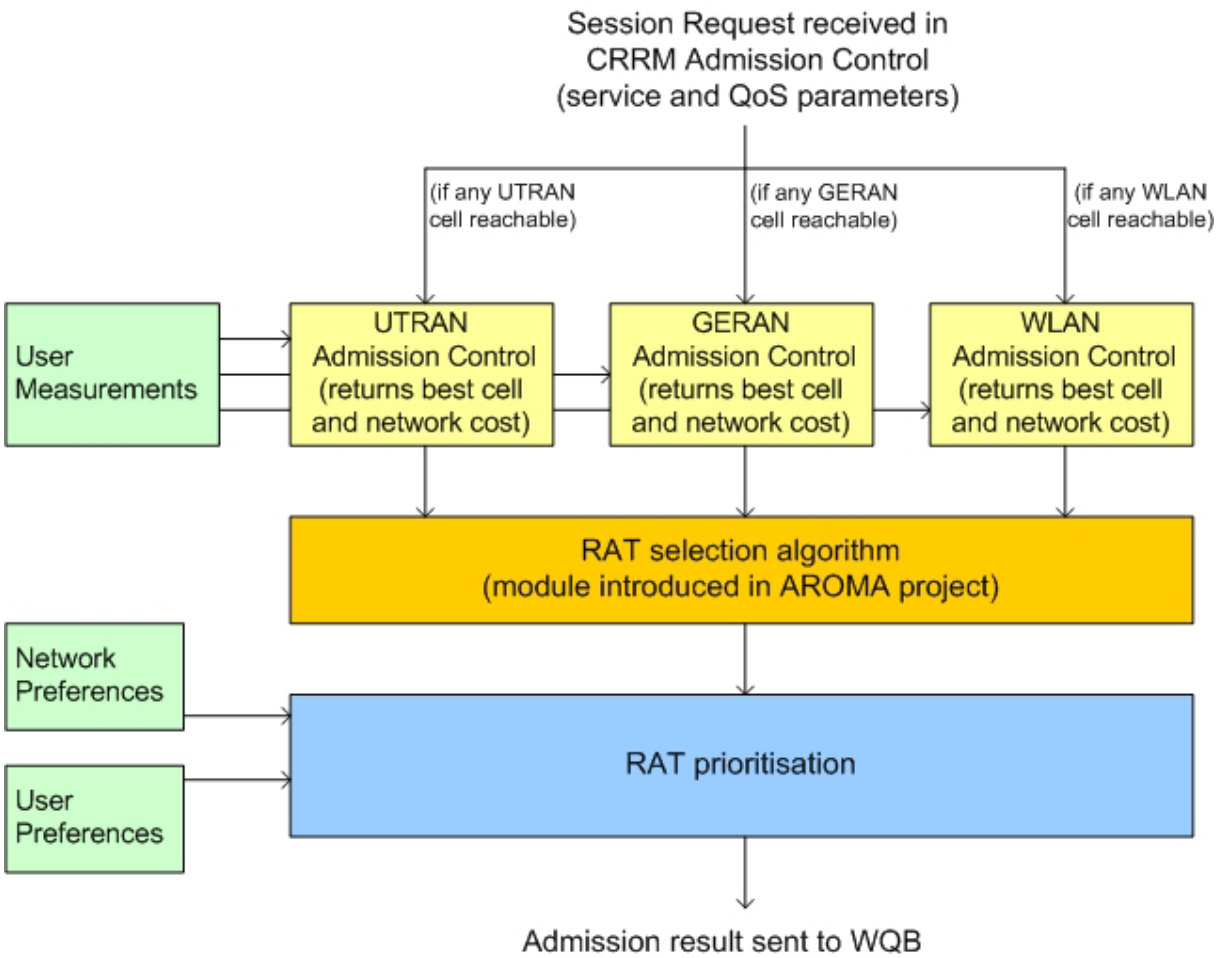


Figure 10. Admission Control algorithm with RAT Selection capabilities.

When a session request arrives, either it comes from the UUT or from an emulated user, local admission is checked in each RAN to which the user could potentially be connected. Local admission control results in a prioritisation of the candidate cells of a given RAT and also in an indication of the network cost, in terms of resource consumption, associated to such decision. Once admission information from local admission control is passed, new RAT selection module is launched. RAT selection process recalculates availability of each RAT (most of the introduced models cover UTRAN and GERAN only) in terms of conditions/restrictions introduced by the selected RAT selection algorithm (see subsection 3.1.3 for the details of newly introduced algorithms). Active RATs that does not meet specified conditions are discarded. Finally, a prioritisation is done among those cells where the user sessions could be established (after the results of RAT selection update). The priority value is obtained for each available RAT by using the following expression:

$$\text{Priority}(\text{RAT}, \text{service}) =$$
$$W_{NP} * \text{NP}(\text{RAT}, \text{service}) + W_{UP} * \text{UP}(\text{RAT}, \text{service}) + W_{RC} * (100 - \text{RC}(\text{RAT}, \text{service}))$$

where:

- UP(RAT, service): User preferences policies to use a given RAT for a given service. Its value ranges from 0 to 100%.
- NP(RAT, service): Network preferences policies to use a given RAT for a given service. Its value ranges from 0 to 100%.
- RC(RAT, service): Resource cost when using a given RAT for a given service. Its value ranges from 0 to 100%.

and W_{NP}, W_{UP} and W_{RC} are the weights (also from 0% to 100%) used to account for network preferences, user preferences and resource consumption respectively.

Resource cost is each RAT is calculated according to criteria provided in Table 3.

Table 3. Criteria for cost resource computation in each RAT

UTRAN	Maximum among uplink and downlink load factor of the candidate cells
GERAN	Ratio of used slots over available slots in the candidate BTS. For GPRS resource computation, data traffic served is measured in kilobits per second and compared to maximum allowed capacity.
WLAN	Ratio of used capacity in terms of data traffic load over total channel capacity.

RATs are put in the descending order of the priority value. Accordingly, RAT with the highest priority value is selected as the first candidate RAT. However, as a result of this admission control, a list of prioritised RATs is passed to the WQB, so that the status of the Core Networks routers can also determine the final RAT decision.

3.1.2 Vertical Handover support

Vertical handover support in the CRRM entity admits several approaches in terms of how tight is the coupling between CRRM and RRM, i.e. how frequent is the interaction between CRRM and RRM entities. Figure 11 shows the execution flow for the vertical and horizontal handover functions in the WQB. As illustrated in the figure, vertical handover (VHO) decision can be configured to be invoked more or less frequently based on the following conditions:

- Check VHO conditions whenever new measurements are available.
- Check VHO conditions only after a horizontal handover (HHO) is decided to be needed.
- Check VHO conditions only when the current RAT can no longer support that call/session.

Furthermore, VHO decision process encompasses RAT selection mechanisms similar to those explained in previous section for admission control. However, main improvements introduced in the

AROMA project to the VHO functionality scoped on the VHO triggering conditions, restricted by the new CRRM algorithms introduced.

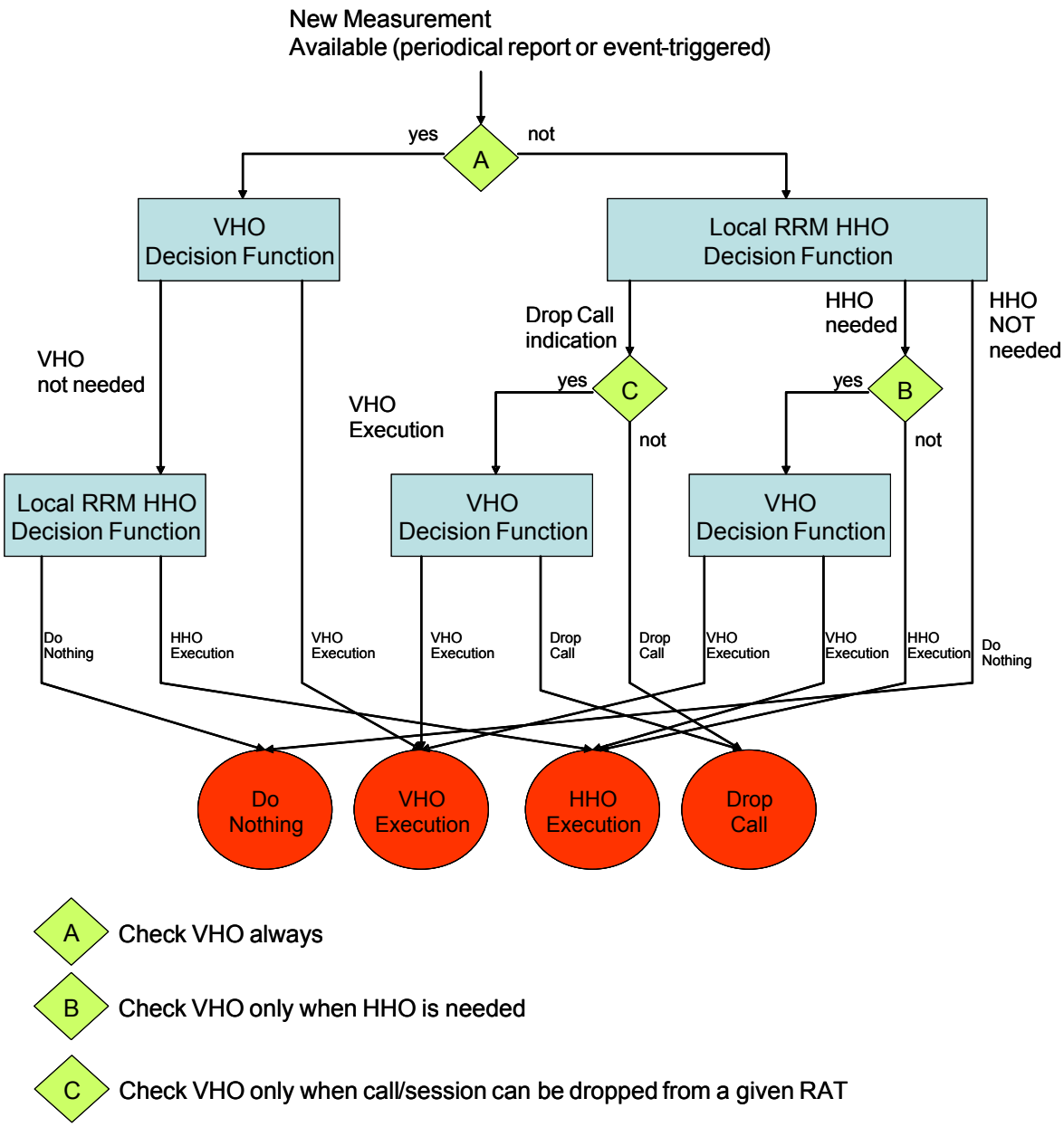


Figure 11. Flow execution for Vertical and Horizontal Handover functions in the WQB

3.1.3 New RAT selection algorithms

Currently, after AROMA update, testbed incorporates six different RAT selection algorithms, namely:

- Default policy, algorithm inherited from the EVEREST project. RAT selection process is based only on the priority value calculation; no special RAT selection is applied.
- Network-controlled Cell Breathing (NCCB), new algorithm incorporated in AROMA project, described in more details further in this section.
- NCCB_voice, new algorithm incorporated in AROMA project. Basically, a variant of the NCCB algorithm that is applied only for the voice users.

- VGNCCB (Voice GERAN NCCB), new algorithm incorporated in AROMA project. A variant of the NCCB algorithm where voice users are assigned to the GERAN RAT, if available and interactive users are managed with NCCB algorithm.
- Fittingness factor policy, new algorithm incorporated in AROMA project, described in more details further in this section.
- Indoor users GERAN, option inherited from the EVEREST project that was converted into a separate policy.

Since some of the newly introduced algorithms are just different variants of the same, basic algorithm the detailed description will be reduced to the following two: Network-Controlled Cell-Breathing (NCCB) and fittingness factor that will be presented here in more details, accordingly.

3.1.3.1 NCCB algorithm

The main idea of a NCCB algorithm, as presented in [30] and [31], is to take the advantage of the coverage overlap that several RATs may provide in a certain service area in order to improve the overall interference pattern generated in the scenario for the CDMA-based systems and, consequently, to improve the capacity of the overall heterogeneous scenario.
During the initial admission the RAT selection decision is taken according to the path loss measurements in the best UTRAN cell (PL_{UTRAN}), provided by the terminal in the establishment phase. If the PL_{UTRAN} is below the path loss threshold value (PL_{th}) the user may be admitted to UTRAN, otherwise it will be admitted to GERAN. The algorithm is addressed to cellular systems, therefore WLAN RAT is not taken into account.

When a VHO is considered, the NCCB algorithm acts according to a procedure presented in Figure 12 . The idea remains the same: keep the users with high path loss connected to GERAN and users with low path loss to UTRAN depending on how the propagation conditions change along the session lifetime. VHO is triggered upon the relation of path loss measurements (PLUTRAN) and the path loss threshold value (PL_{th}) with a certain hysteresis margin (Δ), during M_{up}/M_{down} successive samples to avoid the ping-pong effect (i.e., triggering consecutive back-and-forth VHOs).
Current testbed implementation allows performing the tests related to the NCCB algorithm, in particular, to evaluate the initial RAT selection process as well as the RAT selection process during an on-going VHO in a heterogeneous scenario.

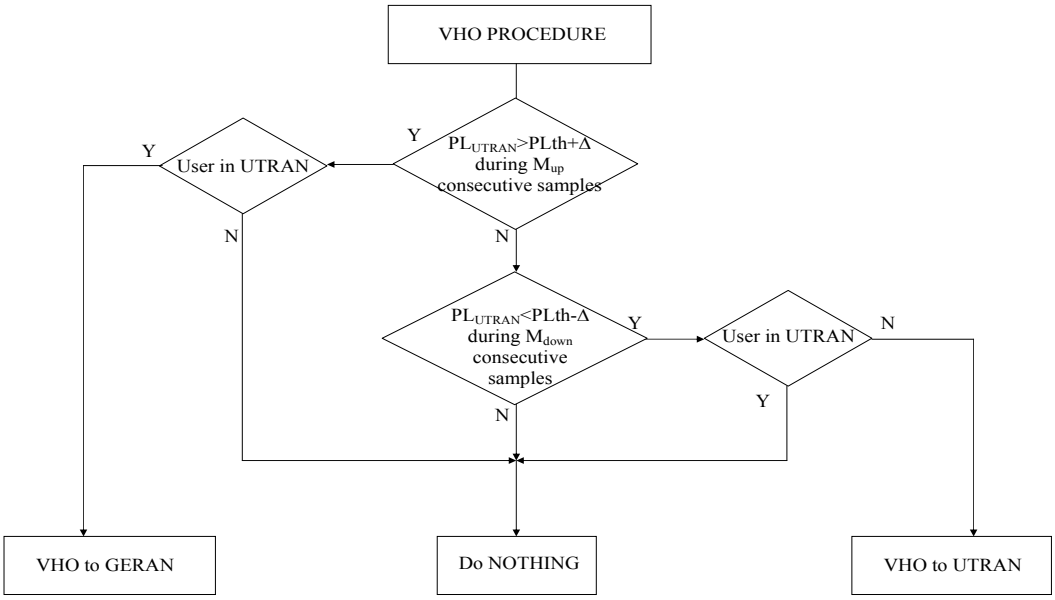


Figure 12. VHO procedure with NCCB algorithm

3.1.3.2 Fittingness factor algorithm

The fittingness factor [32] is a new, generic CRRM metric that facilitates the implementation of cell-by-cell RRM strategies by reducing signalling exchanges and aims at capturing the multidimensional heterogeneity of beyond 3G scenarios within a single metric.

Fittingness factor (Ψ) implemented in the testbed reflects two main aspects of such multidimensional heterogeneity: the capabilities of both terminal to support a particular RAT (i.e. depending on whether terminal is single or multimode), and the RAT to support a particular type of service (e.g. videophone is not supported in 2G networks), denoted here as C, as well as the suitability factor (Q), indicating the match between the user requirements in terms of QoS and the capabilities offered by the RAT (e.g. GERAN may be feasible for the economic users, whereas bit rates required by the business users can be facilitated by the HSDPA). Consequently, the fittingness factor for j-th RAT to support s-th service requested by the i-th user with a p-th customer profile ($\Psi_{i,p,s,j}$) is calculated as a product of corresponding $C_{i,p,s,j}$ and $Q_{i,p,s,j}$ as shown in the formula below:

$$\Psi_{i,p,s,j} = C_{i,p,s,j} * Q_{i,p,s,j}$$

RAT selection process based on the fittingness factor can be characterized by a following scheme:

Initial RAT selection phase:

- Step 1 – calculate $\Psi_{i,p,s,j}$ for each candidate cell of each detected RAT. Calculation is made separately for uplink and downlink; both results are weighted to obtain the final score.
- Step 2 – select RAT with highest $\Psi_{i,p,s,j}$
- Step 3 – Try admission in j-th RAT
- Step 4 - If admission failed, pick the next RAT in decreasing order of $\Psi_{i,p,s,j}$

VHO procedure:

- Step 1 – monitor $\Psi_{i,p,s,j}$ for each candidate cell and RAT. Measurements are averaged in period of T seconds.
- Step 2 – if $\Psi_{i,p,s,j}$(candidate cell) > $\Psi_{i,p,s,j}$(serving cell)+Δ_{VHO} for a period of T_{VHO}, trigger a VHO if there are available resources for this user in that RAT and cell. Δ_{VHO} denotes a hysteresis margin.

Current testbed implementation allows performing the tests related to the fittingness factor algorithm during the initial RAT selection process, as well as during an on-going VHO in a heterogeneous scenario.

3.1.4 Implementation aspects

Basically, the most important change related to the incorporation of new RAT selection algorithms is the introduction of a new global variable to handle the CRRM issues. This global variable called CRRM_policy indicates selected RAT selection algorithm. Provided variable not only encompasses the newly introduced RAT selection algorithms described in previous subsections, but also incorporates the already existing solutions (namely the default policy, and the indoor user geran policy). CRRM_policy variable indicates the selected CRRM policy according to the following code:

```
// CRRM policies introduced
#define DEFAULT 0
#define NCCB 1
#define NCCB_voice 2
#define VG_NCCB 3
#define FITTINGNESS_FACTOR 4
#define INDOOR_USERS_GERAN 5
```

Each RAT selection algorithm provides its own set of parameters, defined as a separate structure, as well as according set of tracing support that is active only if the according policy is selected. Here, the details of structures for newly incorporated algorithms are presented:

- Family of NCCB algorithms (CRRM_policy value is set to 1,2 or 3)

```
typedef struct
{
  double Lp_threshold;                    // UTRAN path loss threshold
  double hysteresis;                      // hysteresis margin in dB to avoid ping-pong effect
                                          when triggering VHO
  int measurement_interval;               // measurement interval in units of 10ms cycles
  int M_up;                               // number of consecutive samples that must be
                                          fulfilled for upper bound
  int M_down;                             // number of consecutive samples that must be
                                          fulfilled for lower bound
}
NCCB_parameters;
```

- Fittingness factor algorithm (CRRM_policy value is set to 4)

```
typedef struct
{
  double value[MAX_RATs];                 // fittingness factor value for each RAT
  double suitability_DL[MAX_RATs];        // suitability factor for DL in each RAT
  double suitability_UL[MAX_RATs];        // suitability factor for UL in each RAT
  double alpha;                           // weight factor UL/DL (0-only DL,1-only UL)
  double hysteresis;                      // hysteresis margin in dB to avoid ping-pong effect
when triggering VHO
  int measurement_interval;               // measurement interval in units of 10ms cycles
  int T_VHO;                              // number of consecutive samples that must be
fulfilled to trigger VHO

  // voice parameters
  double Lp_voice_GERAN_UL_threshold;     // GERAN uplink path loss threshold
  double Lp_voice_GERAN_DL_threshold;     // GERAN downlink path loss threshold
  double S_min;                           // sensitivity level (same for UL and DL)
  double Lp_voice_UTRAN_UL_threshold;     // UTRAN uplink path loss threshold
  double delta_P_max_DL;                  // maximum power available for the user
  double Pt_est_DL;                       // estimated required power for the user DL

 // interactive parameters
```

```
  double R_MCS;                    // max bit rate among all possible MCS schemes
  double R_b_max;                  // max theoretical bit rate
  double fi_p;                     // multiplexing factor
  int M;                           // multislot capability
  double f_R;                      // function that adjust bit rate R* to the closest
                                   transport format
}
fittingness_factor_parameters;
```

For the two remaining two CRRM policies, the modifications were as follows:

- Default CRRM policy (CRRM_policy value is set to 0)

This policy preserves the existing EVEREST implementation of the admission control algorithm. The RAT selection decision is taken only upon the priority value calculation, thus the network cost indications form the user measurments are directly passed to the RAT prioritisation module (transparently to the RAT selection algorithm module, so the calculation is performed for all active RATs) and together with user and network preferences serve for the RAT selection process.

- Indoor users GERAN policy (CRRM_policy value is set to 5)

This policy, previously existing as a standalone option in EVEREST project, was incorporated in general CRRM handling and received a separate value of the CRRM_policy value. Basically, if this option is activated indoor users (recognised by the value of the mobile user speed) are assigned to GERAN.

3.2 WQB-CRRM interaction

In the implemented testbed, the WQB presents an entity responsible for leading the end-to-end QoS negotiation, and for master decisions. The entity itself is tightly bounded with CRRM and has shared buffers as interface to communicate with it. The type of messages is legacy of EVEREST project, though the WQB module is completely new module. In exchange of information five different types of message pairs may appear. Those are Admission, Activation, Deactivation, Modification and Notification, which all appear in Request and Reply version. In the following, description of all of these messages is given. The use of the messages will be included in section 6.

3.2.1 Structures in use

In this section two structures used by the messages exchanged in WQB-CRRM interface are described. All the parameters are explained in short on the right side of structure.

The structure QoSParameters presents the group of parameters that may be used to define the SLS at any level of negotiation. These parameters were taken from [33].

```
typedef struct {
    int traffic_class;        // type of application for which the UMTS bearer service is
                              optimised
    int order;                // indicates whether the UMTS bearer shall provide in-
                              sequence SDU delivery or not
    int erroneous_sdu;        // Indicates whether SDUs detected as erroneous shall be
                              delivered or discarded.
    int max_sdu_size;         // the maximum SDU size for which the network shall satisfy
                              the negotiated QoS
    int max_bit_rate_UL;      // Maximum bitrate in Uplink
    int max_bit_rate_DL;      // Maximum bitrate in Downlink
    float residual_BER;       // Indicates the undetected bit error ratio in the delivered
                              SDUs. If no error detection is requested, Residual bit error
                              ratio indicates the bit error ratio in the delivered SDUs.
```

```
    float sdu_error_ratio;          // Indicates the fraction of SDUs lost or detected as
                                    erroneous. SDU error ratio is defined only for conforming
                                    traffic
    int delay;                      // Indicates maximum delay for 95th percentile of the
                                    distribution of delay for all delivered SDUs during the lifetime
                                    of a bearer service, where delay for an SDU is defined as the
                                    time from a request to transfer an SDU at one SAP to its
                                    delivery at the other SAP.
    int priority;                   // specifies the relative importance for handling of all SDUs
                                    belonging to the UMTS bearer compared to the SDUs of
                                    other bearers
    int guaranteed_bit_rate_UL;     // Guaranteed bitrate in Uplink
    int guaranteed_bit_rate_DL;     // Guaranteed bitrate in Downlink
} QoSParameters;
```

AdmissionResult is a structure that contains the results of admission in CRRM. It is actually a set of parameters that present result of RAT selection. The admission result is used only in messages with direction CRRM->WQB (Admission Reply and Notification Request). All the structure parameters are given for all the RATs in testbed (used ones). The RAT's code is predefined and is 0 for UMTS, 1 for GERAN and 2 for WLAN.

```
typedef struct {
    int ran_weigth[MAX_RATs];           // Weight of each RAT after the RAT selection, O
                                        indicates the user is not admitted
    int ran_er[MAX_RATs];               // Edge router (IR) to which corresponding RAT is
                                        connected
    QoSParameters ran_qos[MAX_RATs];    // This field can be used to propose a QoS settings
                                        for admission in case the user is not admitted
} AdmissionResult;
```

3.2.2 Admission message

AdmissionRequestTransaction is a structure type for Admission Request and Reply messages. These messages are always initiated by the WQB: during the session activation, or some modifications (SLS change or VHO).

The Admission Request bares user ID, service type, and SLS (QoS Parameters). The Admission Reply will respond with AdmissionResults where the weights and parameters available for each RAT will be filled. Optionaly bearer ID may be passed from the user.

```
struct AdmissionRequestTransaction {
    int user;                   // User ID
    int service;                // Service in use
    int bearer;                 // Bearer ID
    QoSParameters qos;          // QoS parameters
    AdmissionResult res;        // Admission Results
};
```

3.2.3 Activation message

ActivationRequestTransaction is a structure type for Activation Request and Reply messages. These messages are always initiated by the WQB. The message is used to activate a specific RAT for the user (selected_RAT) where additional parameters may be included. The action supposes previous Admission Request/reply negotiation.

The Activation Reply will return the successfulness of the action execution (result) to WQB.

```
struct ActivationRequestTransaction {
    int user;                       // User ID
    int service;                    // Service in use
    int bearer;                     // Bearer ID
    QoSParameters qos;              // QoS parameters
    int selected_RAT;               // Selected RAT
    char TFT_Uplink[128];           // TFSW filter, Uplink
    char TFT_Downlink[128];         // TFSW filter, Downlink
    int result;                     // Result on action execution
};
```

3.2.4 Deactivation message

DeactivationRequestTransaction is a structure type for Deactivation Request and Reply messages. These messages are always initiated by the WQB. The message is used to deactivate the resources of the RAT the UUT is using.

The Deactivation Reply will return the successfulness of the action execution (result) to WQB.

```
struct DeactivationRequestTransaction {
    int user;           // User ID
    int bearer;         // Bearer ID
    int result;         // Result on action execution
};
```

3.2.5 Modification Message

ModificationRequestTransaction is a structure type for Modification Request and Reply messages. These messages are always initiated by the WQB. The message is used to modify current UUT's connection. This modification may consider SLS changes or RAT change (VHO).

The Modification Reply will return the successfulness of the action execution (result) to WQB.

```
struct ModificationRequestTransaction {
    int user;                       // User ID
    int service;                    // Service ID
    int bearer;                     // Bearer ID
    QoSParameters qos;              // SLS parameters
    int selected_RAT;               // Selected RAT for VHO
    char TFT_Uplink[128];           // TFSW filter, Uplink
    char TFT_Downlink[128];         // TFSW filter, Downlink
    int result;                     // Result on action execution
};
```

3.2.6 Notification Message

NotificationRequestTransaction is a structure type for Notification Request and Reply messages. These messages are always initiated by the CRRM. The CRRM does not lead the QoS negotiation, and is therefore using this message to inform the WQB that there are modification proposals form the radio part.

The Negotiation Request message is used to inform in changes that are needed in UUT's connection. This modification may consider SLS changes or RAT change (VHO). The message will fill identification fields of user and session and the results of the RAT selection algorithm (Admission Results) that has been executed previous t notification message sending.

The Notification Reply from the WQB (ack) has just a purpose to confirm the reception of the Notification Request. However, this won't trigger any further actions in CRRM. This is subject to

definition that the WQB acts as a master PDP and will give priority to actions (QoS negotiations) that are already in progress. As it will be seen later in the document (section 6) that means the WQB will confirm reception of a notification, but will not necessarily start the corresponding renegotiation.

```
struct NotificationTransaction {
    int user;                   // User ID
    int service;                // Service ID
    int bearer;                 // Bearer ID
    AdmissionResult res;        // Admission results
    int ack;                    // Acknowledgement
};
```

4 END-TO-END QOS SIGNALLING FRAMEWORK

4.1 Discussion

This section provides implementation details of the end-to-end QoS signalling selected framework and some discussion about it. This framework is composed of the QoS Client, the WQB and the BB, which accomplish QoS functionalities and the mobility management entities that accomplish IP mobility management functionalities. The end-to-end QoS signalling is carried out by means of a proprietary interface that follows the kind of interactions present in a specific E2E QoS signalling framework used as a reference.

There are two candidates for such a reference framework for AROMA project: Diameter [34] and COPS-SLS [35]. COPS is a hierarchical protocol – it defines a PDP (Policy Decision Point) and a PEP (Policy Enforcement Point). It is easily extensible although it does not support failover, capability exchange, or security (IPSEC can solve this last issue). This was the framework used in EVEREST.

Diameter is different as there is no notion of server or client (at least until entities start their communications), in fact there are “peers”, where a peer can be a client or a server (Peer-to-Peer architecture). Diameter intrinsically supports authentication and authorization (access control), failover, security, and audit ability. It is also easily extensible.

Regarding the AROMA architecture both protocol frameworks fit the necessary requirements. However, it is only expected to replicate the interactions and the way communication is done between elements instead of the whole framework. Then, following COPS-SLS framework seems easier and more reasonable, taking into account the experience acquired with EVEREST project.

It is important to remark that in EVEREST Testbed COPS protocol is present for E2E QoS Negotiation (COPS-SLS) and for Policy Enforcement (COPS-PR). In the following we address to the E2E QoS negotiation procedures in AROMA copying COPS-SLS protocol procedure. However, COPS-PR protocol has also been changed in AROMA testbed for internal ad-hoc protocols between each PDP and its PEPs.

4.2 COPS-SLS protocol background

The interaction between two entities (PDP and PEP) in COPS-SLS consists of two phases: the initial connection establishment phase and communication phase. The first one consists of the COPS messages: OPN (client-open) and CAT (client-accept). This initial phase could be used for example to specify client specific feature negotiation. In the AROMA testbed this feature is unused. The second one can occur between the PDP and the PEP and is a three handshake protocol (REQ: REQuest, DEC: DECision, RPT: RePorT). This protocol structure has been reproduced in AROMA.

The message object transported by COPS-SLS is the SLS template. This template object is transported in the Client Specific Information Object (ClientSI) of the COPS protocol. This is illustrated in Figure 13, which shows the format of a ClientSI object, corresponding to a C-num=9 and C-type=1. And inside the ClientSI COPS object can be found the SLS template object. This object is made of two

parts: the Encoded Provisioning Identifier (EPRID) and the Encoded Provisioning Instance (EPRI) data. The PRID and PRI are defined in the PIB (Policy Information Base).

length of the object	C-num = 9 (ClientSI)	C-type=1 (Signaled)
Object Content (SLS template) = (EPRID + EPRI)		

Figure 13. COPS-SLS Template

Thus, the implemented interface has a structure that emulates a SLS template with the following functional fields attributes.

- Flow id attribute =
  - source IP address
  - destination IP address
  - destination port
  - source port
- Performance attribute =
  - One way delay
  - Inter-packet delay variation
  - Packet loss rate
  - Throughput
  - DSCP (optional)
- Traffic conformance attribute =
  - Rate
  - Burst

Therefore, these SLS attributes are negotiated between the entities. In the following the term SLS is used referring to this structure. It is worth mentioning here that the information included in the SLS is later mapped to the specific QoS parameters of the radio and core network part.

4.3 Message Structure Specification

Any message exchange between entities is composed of a header and a body. The header carries information about the type of message included in the body, the body length in bytes and an id field that contains the session ID. This session id should be created by the QoS Client (the one that initiates a session QoS request) and should also be used for session modification and release.

The body of the message carries the specific QoS message that can be REQ for a QoS Request, DEC for a Decision and RPT for a Report message. As it has been said this is a three handshake protocol where REQ->DEC->RPT are exchanged between QoS entities in that order.

In the following, the specific structure implemented for the header and each type of message is described.

4.3.1 QoS Message Header

The QoS message header is composed of the following structures:

```
typedef enum type_of_qos_msg {
        REQ = 0,
        DEC = 1,
        RPT = 2
```

```
};

typedef struct qos_header {
        unsigned int        id;          //unique session id
        type_of_qos_msg     msg_type;    //can be REQ, DEC or RPT
        unsigned int        len;         //length in bytes of the body
};
```

4.3.2 QoS Request Message (REQ)

The structure for QoS Request will be like the one described in the structure **qos_req_msg_t.** The rest of structures implement the SLS structure given above.

```
typedef struct flow_attributes_t {
        struct sockaddr_in      src;        //source and destination IP addresses of the QoS session
        struct sockaddr_in      dst;
        unsigned int            protocol;   //Transport protocol (TCP, UDP)
        unsigned int            ir_id;      //Ingress Router ID
};

typedef struct performance_att_t {
        unsigned int    one_way_delay;
        unsigned int    inter_packet_delay_variation;
        unsigned int    pkt_loss_rate;
        unsigned int    throughput;
        unsigned int    DSCP;
};

typedef traffic_conf_att_t {
        unsigned int rate;
        unsigned int burst;
};

typedef enum type_of_req_t {
        QOS_REQUEST         = 0,    //used for session establishment
        QOS_RELEASE         = 1,    //used for session deactivation
        QOS_MODIFICATION    = 2     //used for session modification
};

typedef struct qos_req_msg_t {
        type_of_req_t           type;
        flow_attributes_t       flow_attr;
        performance_att_t       performance_att;
        traffic_conf_att_t      conformance_att;
};
```

The QoS Request message carries the type of request (session establishment, session deactivation or session modification). Also, the requested QoS is carried in the performance_att_t and in the traffic_conf_att_t structs. Finally, the flow_attributes_t struct, carries all the information related to the flow specification, that is, the source and destination IP addresses, the transport protocol and the Ingress Router ID. This ID is used in the communication between the BB and the WQB to specify the Ingress Router where the request is addressed to. In AROMA testbed there are only 2 IRs, so this field can be equal to 1 or 2 representing IR1 or IR2 machines in testbed's architecture respectively. In addition, ir_id can be filled with 0, meaning that the request is addressed to both IRs (used if WQB want to know the capabilities of the CN in both IRs).

4.3.3 QoS Decision Message (DEC)

The QoS Decision message should contain information about the decision done by a QoS entity.

```
typedef enum type_of_dec_msg_t {
        REJECTED                    = 0,
        ACCEPTED_IR1                = 1,
        ACCEPTED_IR2                = 2,
        ACCEPTED_BOTH               = 3
}

typedef struct option_t {
        unsigned int    option_id
        unsigned int    pkt_loss_rate;
        unsigned int    throughput
};
```

```
typedef struct qos_dec_msg_t {
        type_of_dec_msg_t    answer;
        options_t            options_for_IR1;
        options_t            options_for_IR2;
};
```

The answer can be REJECTED if request cannot be accepted in any IR, or accepted in the corresponding IR(s). Additionally, information about the status of the IR is always provided in the options_t structs. Each Ingress Router option is identified by the field option_id.

4.3.4 QoS Report Message (RPT)

The three handshake of the framework should be closed with the Report message sent from the PEP to the PDP. This message should inform the PDP if the PEP was (or not) able to install the required policies.

```
typedef enum type_of_report_msg {
        ACK     = 0,
        NACK    = 1
};
```

```
typedef  struct qos_report_msg_t {
        type_of_report_msg    report;
        unsigned int          option_id;
};
```

The option_id field is filled with the same value provided in the DEC messaged. It is used for informing about the final option (IR) that has been accepted.

5 CN MOBILITY MANAGEMENT, MPLS AND QOS INTERACTIONS

5.1 Mobility Manager and Bandwidth Broker interaction

The objective of this section is to describe the interactions between the MPLS based mobility management with the Bandwidth Broker. The interactions happen during the initial login phase, handover phase and the logout phase. When the user terminal does handover, not only the IP point of attachment is changed but also the QoS configurations along the new path have to be recalculated. Thus, here the goal is to perform a QoS-aware IP handover.

During the login phase, a signalling packet is sent to the AR mobility agent located in the IRs, which is then forwarded to the ANP mobility agent located in the ER of the domain. This signalling packet is necessary in order to setup a filter at the ER for the interception of data packets, destined to the MN's address.
During the login phase this filter is set up, and a care-of-address is provided by the ANP to the MN. In addition, the ANP notifies the BB about the details of the UE and the IP point of attachment. When the UE starts a new session, the BB computes the QoS path in the CN. When a new session is started, the BB calculates the necessary Diffserv QoS reservations. The source routing is then configured at the edge in order to encapsulate the packets which are filtered by the ER. The source routing mechanism is set up at the ANP, which is needed for downlink packets. Furthermore, as the final current DiffServ class for the session has been determined, the BB enforces the DiffServ configuration at the edges of the access network: at the ER for downlink packets and the current IR for uplink packets.

The IP handover (Figure 14) is triggered by the serving RNC (RAN emulator), based on the available radio link measurements. Once the layer 2 handover has been carried out, this triggers an IP handover, which consists of a handoff message sent to the new IR. The new IR forwards this message to the ANP. The ANP process the handoff message and sends an acknowledge message to both the new access router and the old access routers. In addition to this, the ANP also sends the details about the UE and its new IP point of attachment to the BB. When handover happens, the mobility management protocol notifies the BB with the details about the new point of attachment. The ANP performs a QoS reservation request to the BB. If there is any active session during the handover, the BB re-calculates the QoS along the new path, If the DiffServ class is not modified during the handover due to a lack of resources on the new path, then there is no configuration of the edge routers but it is taken care by QoS context transfer between the old and new AR. It can be noticed that if the MN has several sessions, then the process described above is carried out several times. On receipt of the new details of the UE, the BB and the MPLS does the context transfer from the old access router to the new access router. The content of the context transfer includes the DiffServ configuration for the UUT flows. Once the handover and context transfer is done, a Label Switch Path (LSP) is set up between the mobility management entities (ANP/AR for a handover, or also old AR/new AR for a fast handover), and a MPLS tunnel towards the new access router is configured. A detailed description of MPL S tunnel setup is presented in section 5.2.

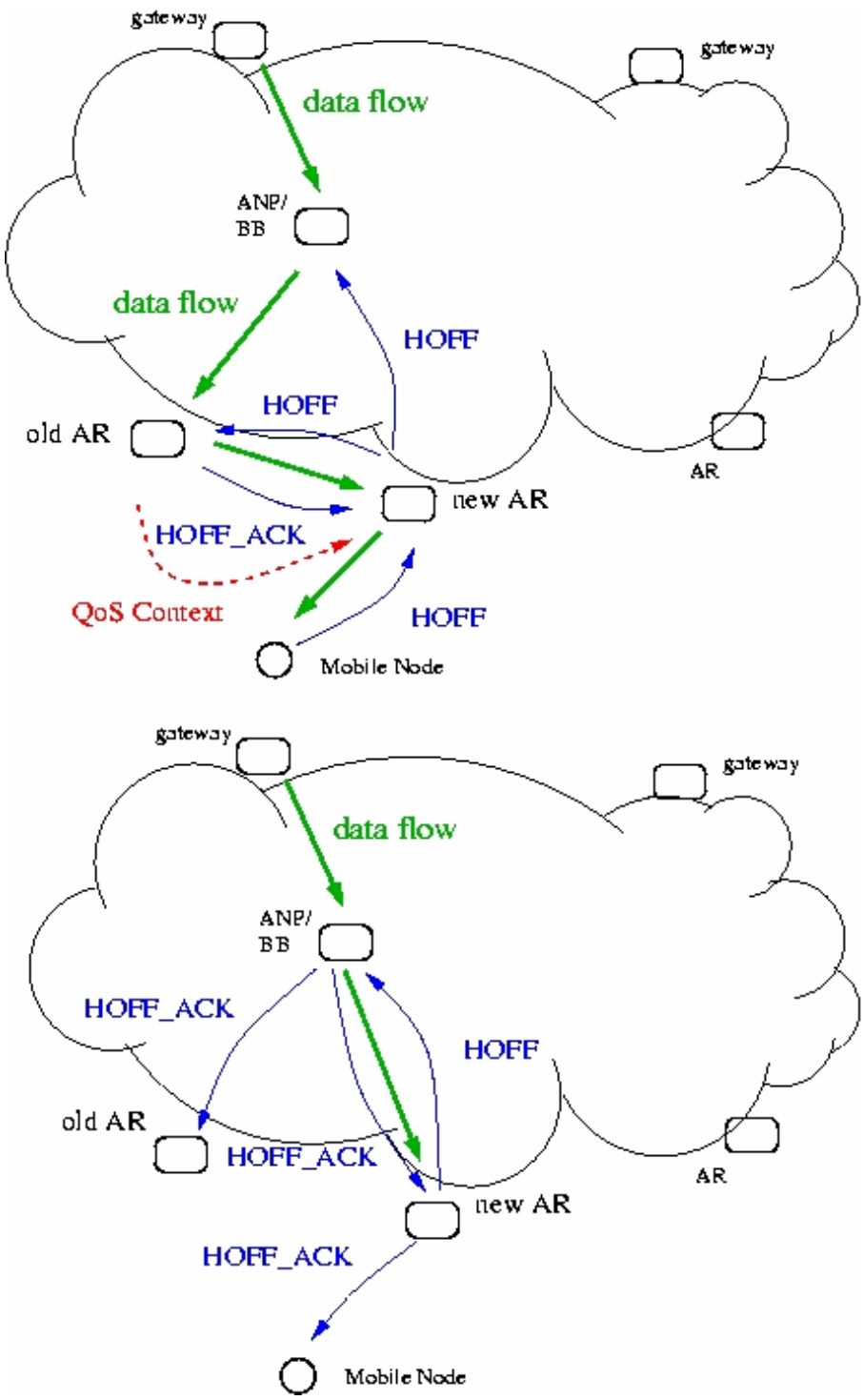


Figure 14. IP Handover

An enhancement to the handover phase is the handover preparation phase. The handover preparation phase aims to reduce the disconnectivity time during an IP handover execution. Thus, the handover preparation phase occurs just before the handover execution phase. Once the target AR is chosen, the MN sends the old AR a handover preparation message with the target AR IP address. A tunnel is set between the old access router and the new access router. This tunnel is removed once a handover execution has been performed afterwards.

The following are the messages for the interaction between the mobility management protocol and the BB.

```
struct BBreq{
        int             BBreq_mesgid;
        int             BBreq_sid;
        int             BBreq_type;
        int             BBreq_code;
        int             BBreq_len;
        int             BBreq_reserved[4];
        char            BBreq_oip[32];
        char            BBreq_nip[32];
        char            BBreq_oar[32];
        char            BBreq_nar[32];
};
```

The request message sent from the mobility management protocol to the BB is in the struct format shown above and has the following fields:

- BBreq_mesgid: contains the type of the generic request message sent by the ANP. In this case, it indicates the message being sent to QoS entities.
- BBreq_sid: Unique identifier to identify the UE
- BBreq_type : contains the state of the message request depending on the whether the MN is in login , handover or logout phase
- BBreq_code: contains the specific request message sent by the ANP. In this case, it specifically indicates that the message is being sent to BB
- BBreq_len; contains the length of the request message sent to the BB
- BBreq_reserved: reserved fields set to zero but can be used for future enhancements of the request message
- BBreq_oip: based on the BBreq_type, it either contains zero or the old IP address of the mobile node. If the MN is in login phase, it is set to zero or if its in handover phase, contains the IP address of the MN before the handover. This field is essential for the BB to identify the MN and their corresponding flows and map the old flow with the new flow
- BBreq_nip: contains the newly assigned address to the MN. This field is essential for the allocation of new path
- BBreq_oar: depending on the BBreq_type, it either contains zero or the old IP address of the mobile node. If the MN is in login phase, it is set to zero or if its in handover phase, contains the IP address of the access router to which the MN was attached before the handover
- BBreq_nar: contains the IP address of the AR to which the MN is or going to be attached after the handover

The mobility management protocol expects a reply from the BB in order to know the results of the BBreq message.

```
struct qrep{
        int             BBrep_mesgid;
        int             BBrep_sid;
        int             BBrep_type;
        int             BBrep_code;
        };
```

- BBrep_mesgid: contains the type of the generic reply message sent by the BB.
- BBrep_sid: Unique identifier to identify the UE
- BBrep_type : contains the state of the message reply depending on the whether the MN is in login , handover or logout phase. This is useful to validate the reply messages
- BBrep_code: contains the specific request message sent by the BB to the ANP. It might contain the result of the BB operations

5.2 MPLS tunnel creation

When the BB is requested to admit a flow from a source to a target that has no previously established path, the BB has the functionality of creating the needed path using a MPLS tunnel and map traffic flows to the created tunnel. In order to do this, the BB requires that all the CN's routers have a component, the BB Agent, that plays the role of a PEP (Policy Enforcement Point), receiving the instructions from the BB with the objective of configuring the router with the needed MPLS/forwarding instructions.

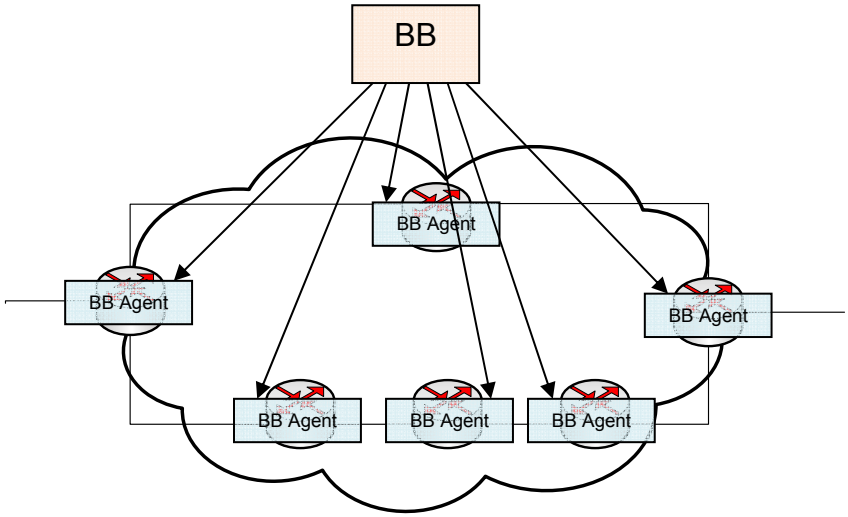


Figure 15. BB and PEP inter-working

When the BB decides to create a MPLS path (LSP), it has to send a message to each of the BB Agents in the involved routers containing the command and the instructions needed to configure the router. In order to establish a normal bidirectional path between a user and a server, two LSPs are required to carry the traffic in the MPLS domain: one for upload between the user and the server, and one for the download between the server and the user.
After the LSPs are created, traffic flows may be mapped to travel inside the established paths. In order to do it, the BB need to send FEC creation messages to both ingress routers (to map uplink traffic to a LSP) and egress routers (to map downlink traffic to a LSP). These messages contain all the needed information to identify a specific flow to a specific LSP.

The following list describes the commands that the BB may send to the BB Agents:

- Commands to manipulate LSPs:
    - Create LSP – Creates an ingress label, egress label or a cross-connection between labels in a router. In order to create a end-to-end LSP, the BB must send this command to all the path's routers.
    - Delete LSP – Removes an ingress label, egress label or a cross-connection between labels in a router. It is used to delete a LSP from the network. In order to delete an end-to-end LSP, the BB must send this command to all the path's routers.
- Commands to manipulate FECs:
    - Create FEC – Creates a filter that enables a flow to follow a certain LSP in a router. BB sends this command only to ingress/egress routers in order to map the network flows in the desired LSPs
    - Delete FEC – Removes a filter that enables a flow to follow a certain LSP. BB sends this command only to ingress/egress routers in order to disable the mapping of a network flow from a LSP.

All the messages share a common first field, which specifies the message type and command. This field is defined in the following enumerated type:

```
typedef enum RequestTypes
{
    CreateFEC = 0,
    DeleteFEC,
    CreateLSP,
    DeleteLSP,
} RequestType_t;
```

The LSP manipulation messages are composed by the following fields:

- Request type – specifies the message type (FEC or LSP manipulation) and command (create or delete)
- Incomming LSP – Specifies the ingress label for the rule
- Incomming Interface – Species in which interface the ingress label is arriving to the router
- Outgoing LSP – Specifies the outgoing label for the rule
- Outgoing LSP – Specifies to which interface the outgoing label is exiting from the router
- Next Hop – Specifies the IP address of the next hop for the LSP

The described fields are depicted in the structure LSP_Info:

```
struct LSP_Info
{
    RequestType_t request_type;
    unsigned int incomming_LSP;
    char incomming_interface[16];
    unsigned int outgoing_LSP;
    char outgoing_interface[16];
    char next_hop[16];
};
```

FEC manipulation messages are composed by the following fields:

- Request type – specifies the message type (FEC or LSP manipulation) and command (create or delete)
- Src – Source address of the flow in dotted string format (eg. “192.168.1.1”)
- Dst – Destination address of the flow in dotted string format
- srcPort – Source port of the flow
- dstPort – Destination port of the flow
- Protocol – Protocol of the flow (eg. tcp or udp), as defined in the linux system header file netinet/in.h.
- DSCP – Defines the traffic class of the flow
- Target LSP – specifies the matching between the described flow and a LSP in the MPLS domain.

The described fields are depicted in the structure FEC_Info:

```
struct FEC_Info
{
    RequestType_t request_type;
    char src[16];
    char dst[16];
    unsigned short srcPort;
    unsigned short dstPort;
    unsigned int protocol;
    unsigned int DSCP;
    unsigned int target_LSP;
};
```

6 SUPORTED PROCEDURES

In this section supported procedures that are involved in the end-to-end QoS negotiation are described. Before start describing the procedures it should be noted that the WQB as the master decision point will as implemented give priority to currently running process before starting any other new process. This means that the request of any form will automatically respond with rejection if another process is running.

6.1 Session Activation

6.1.1 Objective

The objective here is to show that a coordination between the admission control done in the radio part and in the CN can improve the overall communication service by avoiding cases where the QoS chosen by the radio part does not match the available QoS in the CN and vice versa. Furthermore, it can also been shown that depending on the policy used at the WQB and at the BB (operator defined policy, for example priority to a certain type of user or to a certain RAT) the outcome of the admission control in terms of the negotiated SLS and attachment point to the access network, will be different.

6.1.2 Involved entities, interfaces and protocols

- Control plane:
  - WQB: It bares master PDP functionalities. Is responsible for renegotiation of QoS. It communicates with QoS Client, BB and CRRM.
  - CRRM functionalities: monitor the status of the RATs and has knowledge of the radio QoS corresponding to each RNC. Moreover, it also has PDP functionalities so that based on its internal policy, it can take an admission control decision and enforce this decision at the RNC (and possibly also the NodeB) level.
  - BB: performs policy-based admission control for the IP access network. Thus, based on its policy and the information gathered in its database, it takes an admission control decision. This decision is then enforced to the edge entities of the IP access network through the E2E QoS signalling interface.
  - ANP: is a centralized mobility management entity. It has a mobility management database, which maps the user identification to the current IP point of attachment of the user. It acts similarly to the SGSN with the GTP-C protocol.

- Data plane:
  - RNC: in addition to its radio functionalities, it has DiffServ functionalities of marking and shaping traffic.
  - Gateway: separates the CN and the external networks. It has the usual gateway functionalities. The ANP resides at the gateway. Thus, it communicates with the BB to pass the information about the UE and its IP point of attachment for it to negotiate QOS requirements and establish the MPLS tunnel.

Note that in the testbed various modules form part of one entity. Different modules are responsible of different functionalities of the same entity and are made/controlled by different partners involved in the project. Those will appear later in the figures of this section. For example MN, AR, ANP are responsible for mobility management, while PEPs and QoS Client are involved in QoS provisioning.

6.1.3 Procedure description

1. It is supposed that a link layer connection set up has already been carried out.
2. The UUT receives router advertisements, which trigger a login request to the ANP.
3. Once the login request has been received and a new IP address has been configured, it is passed to QoS Client from Mobility Manager.
4. The QoS Client, than triggers the QoS request of the UUT.

5. There is a QoS negotiation between the user, the WQB, the BB. The implemented protocol is used as the communication protocol, and the object of the negotiation between the different entities is the SLS implemented structure. In this case, the UUT initiates the negotiation by sending a REQ message (whose type is QOS_REQUEST) to WQB.
6. Then WQB enforces the CRRM to execute the admission control in RANs (Admission Request).
7. The CRRM executes a RAT selection algorithm and decides on the Admission Request by creating a list of the RATs priority or just one RAT elected (depending on the RAT selection algorithm and MN's state). The result is sent to the WQB in Admission Reply.
8. On the reception of the Admission Reply message, the WQB takes a decision of the target RNC based on its CRRM information. It creates a SLS request, which contains corresponding IRs (used by potential target RATs) and forwards it to the BB (again with sub-type QOS_REQUEST).
9. The BB receives the requested SLS and a set of target IRs. The BB checks if the requested SLS could be provided through the possible IRs, which are attachment points to the IP access network. Finally the decision of the BB is forwarded to the WQB by sending the DEC message. It contains the evaluation of the available QoS on all the requested routers.
10. The WQB makes a final check of consistency of the chosen SLS with the available resources depending on the answer given by BB for one or both IRs. It makes final decision where the selected combination of RAT<->IR pair should provide requested QoS level and meat user preferences if possible. The final decision is forwarded to the user. In case the QoS has to be degraded this will also be included in the DEC message as SLS degradation, or in the worst case the failure to establish a connection may be a final decision.
11. The user accepts/rejects the available SLS offered by the network by sending a RPT message. The user sends RPT back to WQB including final decision.
12. The RPT is interpreted in WQB and corresponding equivalent is forwarded to BB.
13. The QoS negotiated at the WQB and the BB is enforced in the PEPs using an ad-hoc protocol internal to WQB or BB and their respective PEPs.
14. Based on the previous interpretation of the RPT received from QoS Client, WQB may send an Activation Request message to CRRM.
15. The CRRM replies with Activation Reply, informing on the successfulness of the executed action.

Note that during the session activation, an attachment point (IR) may be changed (not the same as the one from preadmission) where mobility management would deal with it. However it is explained in renegotiation section with handover appearance, and as does not influence QoS negotiation during session establishment is not included here.

6.1.3.1 Reference scenario

For the considered scenario, the following parameters have to be determined:

- The user under test's trajectory
- The traffic load in the CN (setup of the traffic generator)
- The different services requested by the user under test.
- Predefined policies at the BB, CRRM and WQB. For example load balancing, preferred RAT.

6.1.3.2 Message Chart

The message chart describes session activation steps in QoS negotiation. The REQ/DEC/RPT of QoS Client <-> WQB and WQB<->BB are marked with different colours for easier distinguishing.

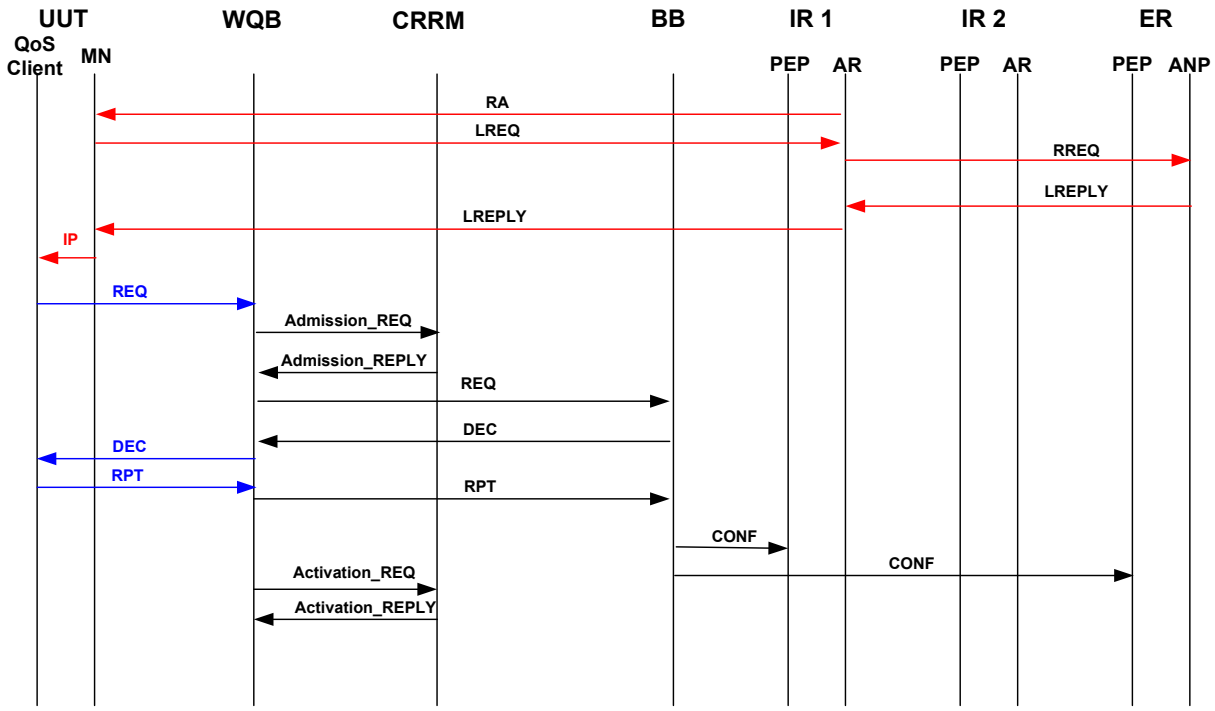


Figure 16. Session Activation Flowchart

6.2 Session Deactivation

6.2.1 Objective

The objective of this procedure is to define the deactivation process that QoS entities should follow in order to release resources once the session is finished.

6.2.2 Involved entities, interfaces and protocols

Those already detailed in 6.1.2.

6.2.3 Procedure description

1. It is supposed that a layer connection exists due to the previous session running.
2. When the session is finished, the UUT sends a REQ message with a QOS_RELASE indication.
3. The WQB now forwards the request for deactivation to the BB adding the IR id field of the used IR.
4. The BB receives the requested deactivation and sends a correspondent DEC ACCEPTED message to WQB.
5. The WQB forwards the deactivation confirmation (DEC message) to the user.
6. Finally, the resources are to be released by the reception of the RPT ACK message in WQB.
7. On receiving the RPT in WQB a corresponding equivalent is forwarded to BB.
8. Upon receiving the RPT message, the BB forces release of the resources.
9. Than, the WQB sends Deactivation Request message to CRRM.
10. After deactivation is executed, the CRRM returns the confirmation to WQB (Deactivation Reply).

6.2.3.1 Reference scenario

For the considered scenario, the following parameters have to be determined:

- The user under test's trajectory

- The traffic load in the CN (setup of the traffic generator)
- The different services requested by the user under test.
- Predefined policies at the BB, the WQB. For example load balancing, preferred RAT.

6.2.3.2 Message Chart

The message chart describes session deactivation steps in QoS negotiation. The REQ/DEC/RPT of QoS Client <-> WQB and WQB<->BB are marked with different colours for easier distinguishing.

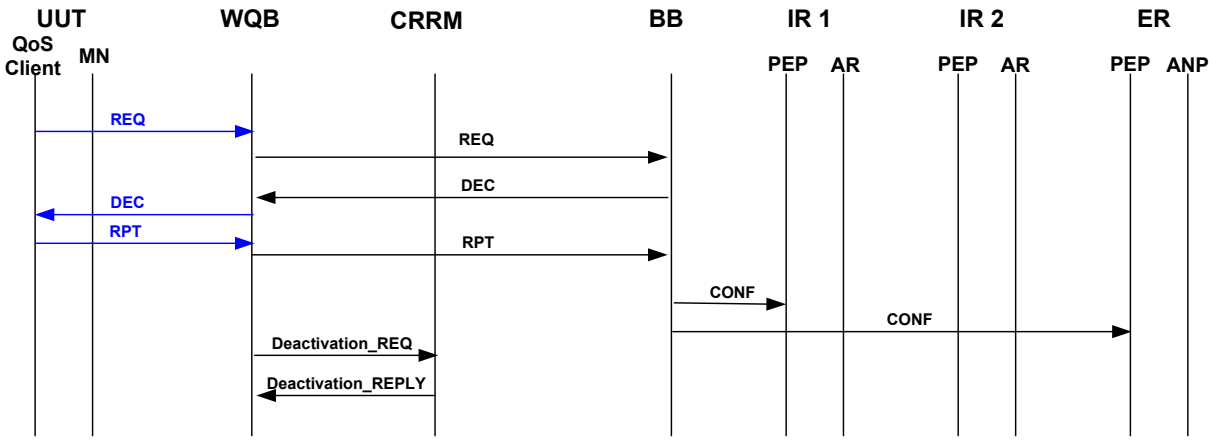


Figure 17. Session Deactivation Flowchart

6.3 Session Modification

6.3.1 Objective

The objective is to show that QoS modification can be successfully supported if it is triggered by the UUT in the middle of a session. This modification can result into an SLS downgrade or upgrade.

6.3.2 Involved entities, interfaces and protocols

See subsection 6.1.2

6.3.3 Procedure description

It is similar to that procedure in section 6.1.3 but with the initial login phase already done.

1. The UUT triggers a new QoS renegotiation by sending a REQ QOS_MODIFICATION message to the WQB. There is a QoS negotiation between the UUT, the WQB and the BB.
2. Then WQB enforces the CRRM to execute the (new) admission control in RANs (Admission Request). This admission will only test the possible new resource availability and will not make changes in currently established connection.
3. The CRRM executes a RAT selection algorithm and decides on the Admission Request by creating a list of the RATs priority or just one RAT elected (depending on the RAT selection algorithm and MN's state). The result is sent to the WQB in Admission Reply.
4. On the reception of the Admission Reply message, the WQB takes a decision of the target RNC based on its CRRM information. It creates a SLS request, which contains corresponding IRs (used by potential target RATs) and forwards it to the BB (again with sub-type QOS_REQUEST).
5. The BB receives the requested SLS and a set of target IRs. The BB checks if the requested SLS could be provided through the possible IRs, which are attachment points to the IP access network. Finally the decision of the BB is forwarded to the WQB by sending the DEC message. It contains the evaluation of the available QoS on all the requested routers.

6. The WQB makes a final check of consistency of the chosen SLS with the available resources depending on the answer given by BB for one or both IRs. It makes final decision where the selected combination of RAT<->IR pair should provide requested QoS level for newly defined session. The final decision is forwarded to the user. In case the QoS has to be degraded this will also be included in the DEC message as SLS degradation, or in the worst case the failure to establish a connection may be a final decision.
7. The user accepts/rejects the available SLS offered by the network by sending a RPT message. The user sends RPT back to WQB including final decision.
8. The RPT is interpreted in WQB and corresponding equivalent is forwarded to BB.
9. The QoS negotiated at the WQB and the BB is enforced in the PEPs using an ad-hoc protocol internal to WQB or BB and their respective PEPs.
10. Based on the previous interpretation of the RPT received from QoS Client, WQB may send a Modification Request message to CRRM. Only now new SLS specifications will be implemented in RAN.
11. The CRRM replies with Modification Reply, informing on the successfulness of the executed action.

Note that during the session modification (similar to the session activation), an attachment point (IR) may be changed where mobility management would deal with it. However it is explained in renegotiation section with handover appearance, and as does not influence QoS negotiation during session modification is not included here.

6.3.3.1 Reference scenario

In addition to the parameters considered in subsection 6.1.3.1, a secondary QoS level has to be selected for the user SLS in the case of QoS renegotiation due to the non-availability of the requested resources.

6.3.3.2 Message Chart

The message chart describes session modification steps in QoS negotiation. The REQ/DEC/RPT of QoS Client <-> WQB and WQB<->BB are marked with different colours for easier distinguishing.

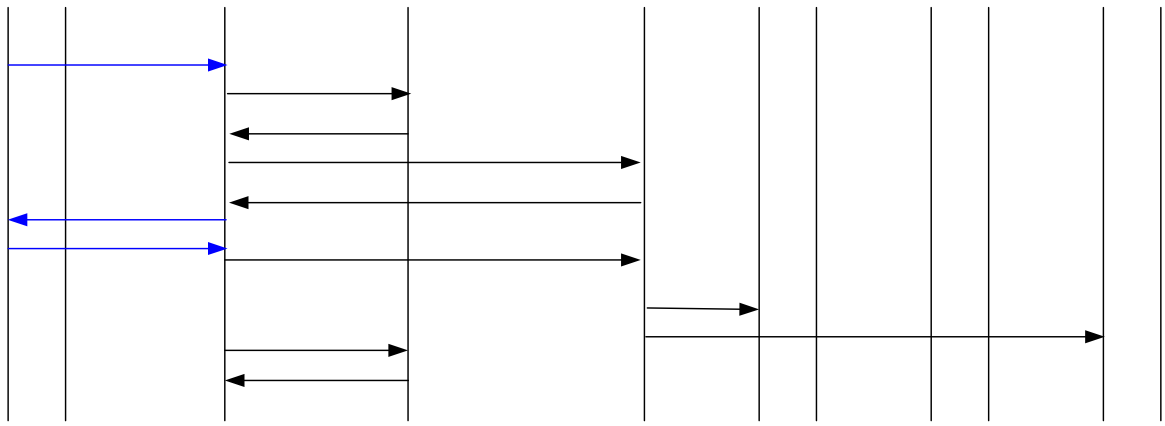


Figure 18. Session Modification Flowchart

6.4 QoS Re-negotiation

6.4.1 Objective

The objective is to show that QoS re-negotiation may be needed due to the changing QoS status of the CN and the RANs. This re-negotiation can be triggered by the RAN or the CN and may or may not result into a SLS modification.

For example after an IP handover to a new attachment point which does not support the previously negotiated QoS, the WQB should start a QoS re-negotiation and the SLS may be downgraded. In consequence, the user interaction is required for approval. In other case, the original SLS can be maintained while BB and/or WQB are re-configuring. In this situation, QoS re-negotiation is needed between those entities without involving the final user.

Finally, other situations not triggered by handovers can also be studied. For example, a situation where the load increases in the CN and it causes a degradation of the guaranteed BW (DiffServ class degradation). In this case we are dealing with a CN initiated (BB) QoS re-negotiation. Again, it may or may not involve UUT in the re-negotiation.

In the following all these situations are presented.

6.4.2 Involved entities, interfaces and protocols

See subsection 6.1.2

6.4.3 Procedures description

- RAN initiated QoS re-negotiation **with/without** SLS modification (Figure 19):

1. Based on the periodic measurement of the user, the current attached RNC may decide for a handover to a cell not under its control (Other RAT attached on another IR). Thus, it contacts the WQB by sending a Notification Request.
2. The WQB if available will receive the notification request, and replies on it with Notification Reply.
3. In order to decide for the SLS and the target attachment point, a QoS negotiation between WQB and BB has to be performed. The WQB initiates the negotiation by sending a REQ QOS_MODIFICATION request message to the BB for new attachment point. Note that in this case, WQB will request only for one IR.
4. The BB then checks if the requested SLS can be supported in the new attachment point and forwards its decision DEC regarding the requested IR to the WQB.
5. Now, the WQB checks if the previously negotiated SLS can be guaranteed. In the case it cannot, the newly negotiated SLS has to be accepted by the user. The UUT is involved in the procedure where a new SLS is within a REQ QOS_MODIFICATION message sent to the UUT.
6. The UUT replies by accepting or not the new SLS (DEC message).
7. The connection is closed by sending RPT message from WQB to UUT.
8. The WQB makes final decision and closes the negotiation with BB by sending RPT message.
9. Once the newly SLS has been accepted, the decisions are enforced by the PDPs at the PEPs. The WQB will use the Modification Request/Reply pair to communicate CRRM (or Deactivation in case of session rejection), while BB will use internal protocol.
10. On receipt of the router advertisement from the access router, the MN (UUT) sends a handoff request (HOFF) message to the ANP located in the ER. Also, if the optional handover preparation state is included a tunnel is established between the IR to which the UUT is currently attached and the new IR to which the UUT is going to attach after handover.
11. ANP process the handoff request message and sends the handoff acknoweldgement or reply (HOFF_ACK) message back to the UUT. The acknoweldgement message contains the IP care-of-address assigned to the UUT by the ANP. In addition the ANP notifies the BB about the UUT details and its IP point of attachment to the RAN.
12. On receipt of the handoff acknoweldgement message, the MN (UUT) notifies the QoS client about its new IP address.
13. In the meantime, the BB does QoS negotiations in the CN and notifies the MPLS to establish a MPLS tunnel between the ER and the IR to which the UUT is currently attached to.
14. So the packets arriving for the UUT at the ER are tunnel to the IR and then reaches the UUT

Note that communication in steps 5-7 is optional as the degradation of the service quality may not appear and the QoS need not be involved in re-negotiation.

Another option supported by the testbed is the case when degradation of the SLS appears in the CRRM so the WQB is called for re-negotiation with no option to switch to another RAT connected to other IR. Then the procedure would stay the same just the communication in between WQB and BB will be skipped and QoS Client will be contacted immediately.

- CN initiated QoS re-negotiation **with/without** SLS modification (Figure 20):

1. Based on the periodic measurements from the edge routers, the BB may decide that the current UUT's flow should be moved to another attachment point or that the QoS parameters in the SLS should be downgraded. Then, BB sends a REQ QOS_MODIFICATION message to WQB informing of the attachment point (IR) and the QoS parameters. Note that in this case, BB will request only for one IR, but the expected reply from WQB is a DEC ACCEPTED_IRx or REJECTED message only.
2. The WQB analyses the information and triggers admission control in the radio part by dending Admission Request to CRRM. Note that again the Admission will not be changing current session parameters by itself.
3. CRRM will start the RAT selection again and return the Admission Reply like in session establishment.
4. Now, the WQB checks if the previously negotiated SLS can be guaranteed. In the case it cannot, the newly negotiated SLS has to be accepted by the user. The UUT is involved in the procedure where a new SLS is within a REQ QOS_MODIFICATION message sent to the UUT.
5. The UUT replies by accepting or not the new SLS (DEC message).
6. The connection is closed by sending RPT message from WQB to UUT.
7. The WQB processes the final decision and sends a DEC message to BB.
8. BB closes the negotiation with WQB by sending RPT message.
9. Once the newly SLS has been accepted, the decisions are enforced by the PDPs at the PEPs. The WQB will use the Modification Request/Reply pair to communicate CRRM (or Deactivation in case of session rejection), while BB will use internal protocol.
10. On receipt of the router advertisement from the access router, the MN (UUT) sends a handoff request (HOFF)message to the ANP located in the ER. Also, if the optional handover preparation state is included a tunnel is established between the IR to the which the UUT is currently attached and the new IR to which the UUT is going to attach after handover.
11. ANP process the handoff request message and sends the handoff acknoweldgement or reply (HOFF_ACK) message back to the MN (UUT). The acknoweldgement message contains the IP care-of-address assigned to the UUT by the ANP. In addition the ANP notifies the BB about the UUT details and its IP point of attachment to the RAN.
12. On receipt of the handoff acknoweldgement message, the MN (UUT) notifies the QoS client about its new IP address
13. In the meantime, the BB does QoS negotiations in the CN and notifies the MPLS to establish a MPLS tunnel between the ER and the IR to which the UUT is currently attached to.
14. So the packets arriving for the UUT at the ER are tunnel to the IR and then reaches the UUT

Note that communication in steps 4-6 is optional as the degradation of the service quality may not appear and the QoS need not be involved in re-negotiation.

Another option supported by the testbed is the case when degradation of the SLS appears in the BB so the WQB is called for re-negotiation with no option to switch to another IR. Then the procedure would stay the same just the communication in between WQB and CRRM (steps 2-3) will be skipped and QoS Client will be contacted immediately.

6.4.3.1 Reference scenario

In addition to the parameters considered in subsection 6.1.3.1, a secondary QoS level has to be selected for the user SLS in the case of QoS renegotiation due to the non-availability of the requested resources.

6.4.3.2 Message Charts

The message chart describes session deactivation steps in QoS negotiation. The REQ/DEC/RPT of QoS Client <-> WQB and WQB<->BB are marked with different colours for easier distinguishing. At the same time for the QoS Client <-> WQB pair, the signals are drawn with dashed lines as are optional (appear in case of SLS modifications).

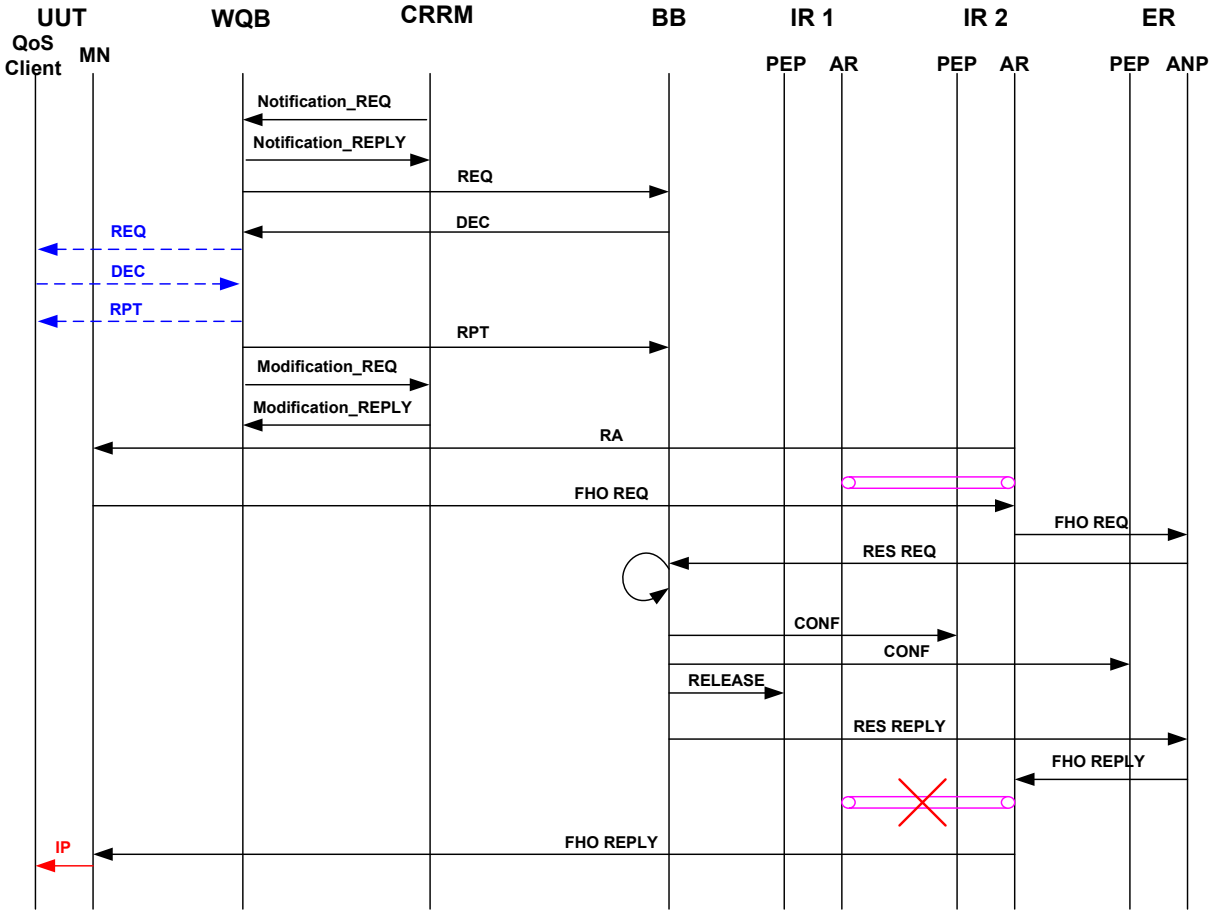


Figure 19. QoS Re-negotiation triggered by the CRRM

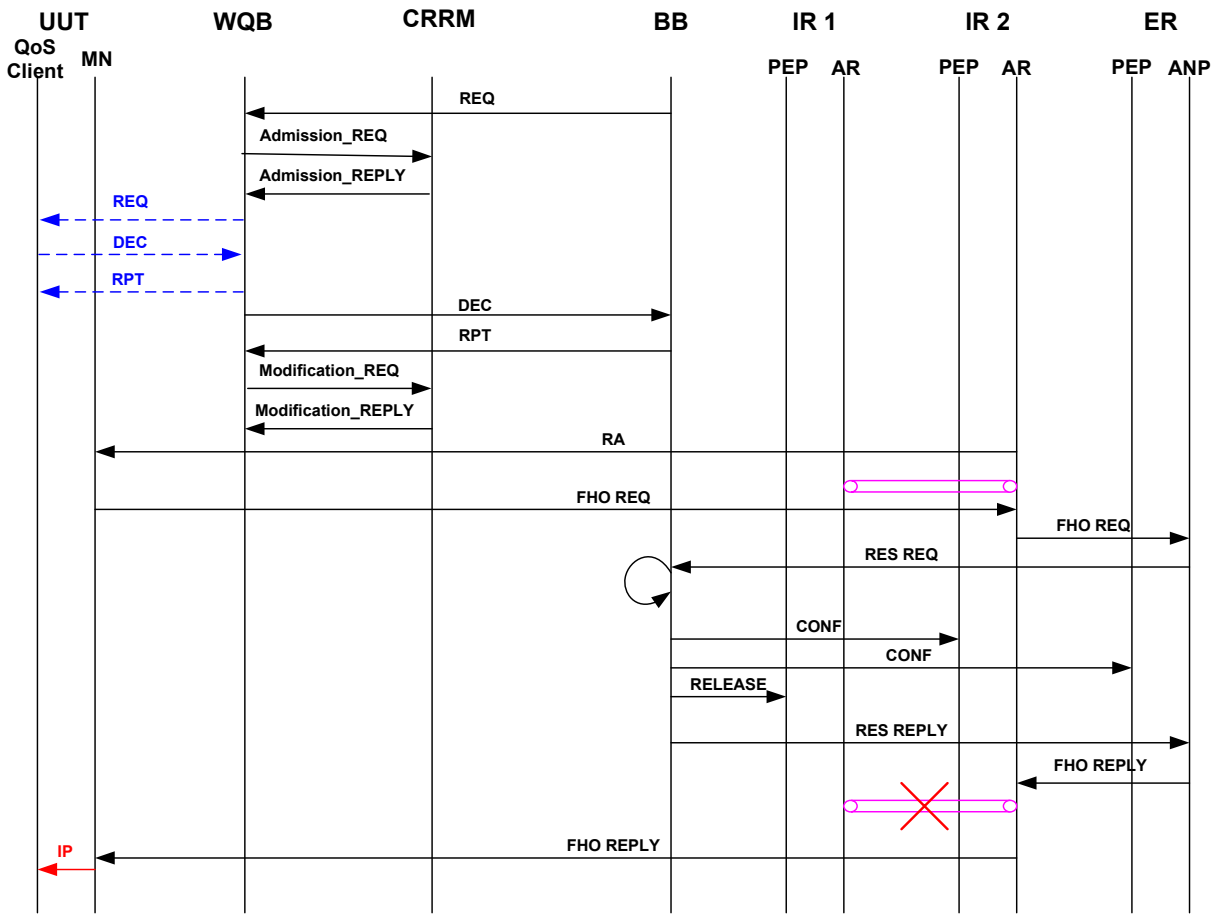


Figure 20. QoS Re-negotiation triggered by the CN

7 AROMA TESTBED USER GUIDE

The basis of testbed switching and control originates from the EVEREST project, though all the modules in the testbed have been upgraded or completely replaced in the AROMA project. Some more processes are automated now. The differences will be pointed in this document, while for additional details, that have origins in EVEREST, the reader is directed to Deliverable 18 of that project [3], where in Apendix B the EVEREST User Guide is given.

7.1 Switching on the AROMA testbed

The initialisation and start of the testbed is an automatic process. Just a service needs to be running on each testbed machine. This service is configured to start in the booting process (similarly as an *ftp* daemon or any other service is started) and can be reached from the AGMT to unleash all the programs on the different machines.

7.1.1 Starting Communications Manager

The processes (modules) suppose previous start-up of communications manager on the corresponding machine of execution. The Communications Manager is started in two steps. The first step consists of launching the daemon (RUNTB) that is able to start the different tools (processes) required for the correct behaviour of the testbed. One different daemon must be running on each testbed machine. The second step consists of starting a means to control what these daemons do. Execution for the trials that will be done in AROMA project is through the scripts that enable easier setup of parameters and command execution. One port and statistics have to be enabled in each run. For more details on starting CM and a description of the manual procedure, please see [3].

7.1.2 Starting AROMA Modules

Once CM is running, the modules can be started. It can be done manually opening different terminal windows on different machines to start each one of the modules. This procedure is cumbersome and not appropriate to get flexibility. Then an automatic procedure has been developed. It is based on the RUNTB and SHELL processes and on the AGMT. The start-up of modules is realized through configuration files of AGMT (wholetestbed.ini) and is integrated in the 'setup' fase of the AGMT. In case it is necessary, any module can be disabled and started (controlled) outside AGMT. This approach is the same as in [3], just with different modules involved, and additional machines used.

7.2 Setting up the Advanced Graphical Management Tool

In this section the solution adopted to manage the AROMA testbed in a centralised way from a graphical tool is described.

The functionalities supported by the Advanced Graphical Management Tool (AGMT) are the following:

1. Control the execution flow of the test bed (init, run, pause, restart options) and selection of the scenario to be demonstrated through the automatic use of SHELL.
2. Configuration of all the initialisation parameters required in the modules running in the test bed.
3. Collect and correlate logged data from the different modules of the demonstrator. This tool is to be used for post-processing purposes. It allows following the logged events in a dynamic way.
4. View statistics during the execution of a demonstration (on-line representation/visualisation).
5. Change some configuration parameter during the execution of a trial to force a given situation (i.e. increase the number of users dynamically to analyse consequences over radio bearers established by the user under test).

This software has not been modified, but rather retuned for the AROMA project, in order to support new modifications in testbed structure and new modules and their needs.

The AGMT software is installed in a separated PC that is connected to the public LAN of the test bed. The software has been developed in JAVA due basically to its facilities in programming graphical and networked applications and also to allow the portability to different Operating System platforms. The additional CRs that AROMA project has, are introduced in the network architecture. Apart of those, in AROMA project, the two new PCs are installed for running the applications under WINDOWS OS, one on client's and other on server's side.

The Figure 21 presents the connections of the machines in AROMA testbed.

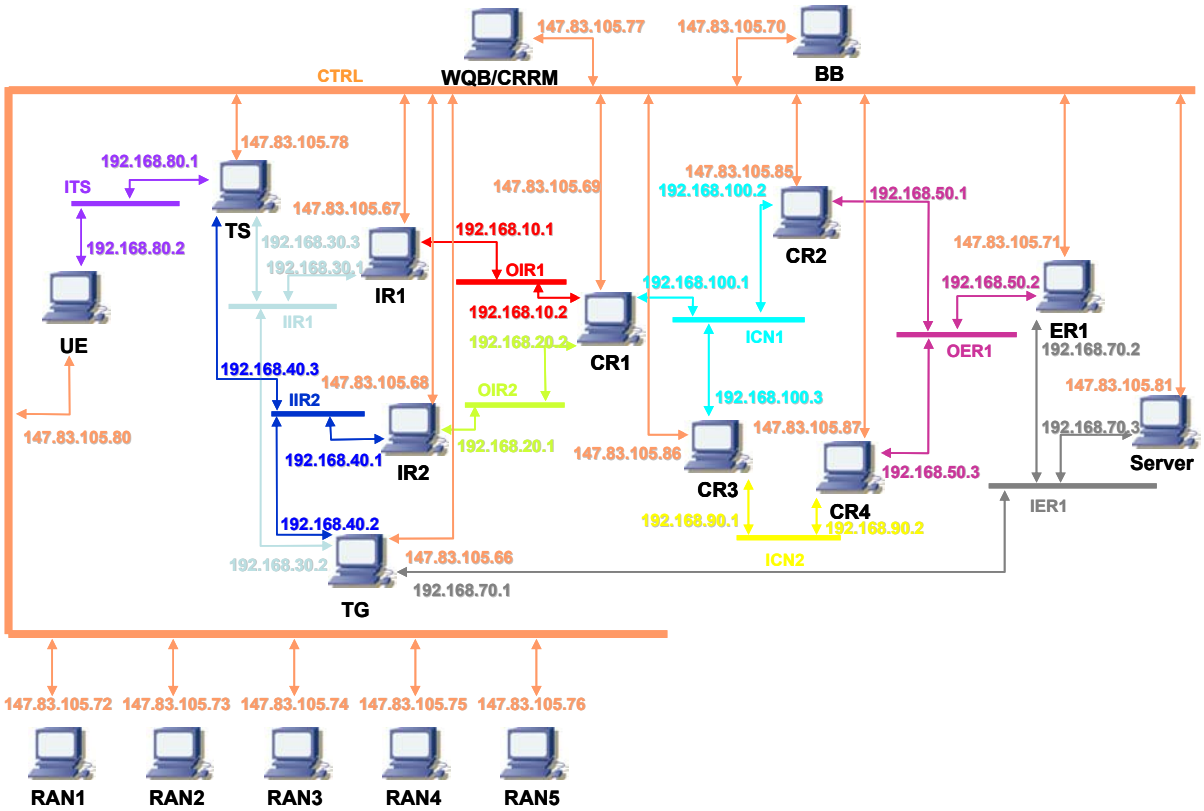


Figure 21: Aroma testbed connections.

New modules and structure have been translated to AGMT (Figure 22).

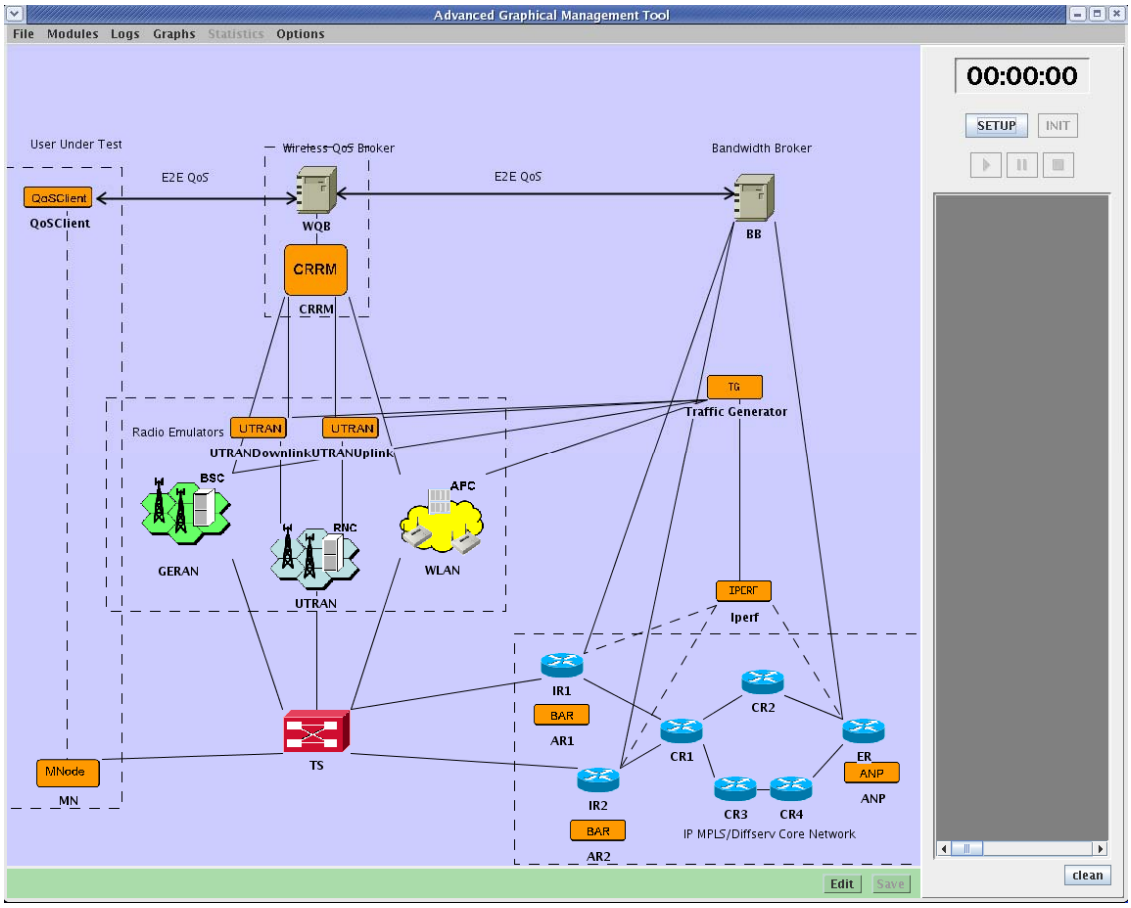


Figure 22: AGMT Main Window

As before, the flow management protocol still consists of the same steps (Figure 23) as in [3], and the configuration interfaces have stayed the same.

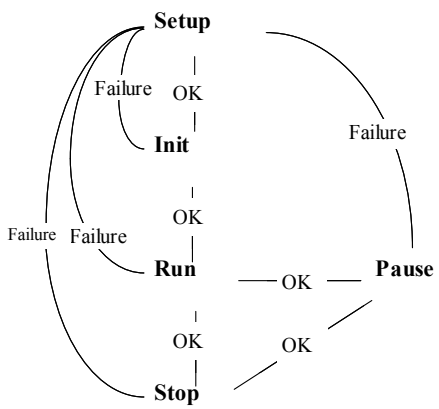


Figure 23: Execution flow management protocol

7.3 Configuration of the scenarios and data collection

Setting of the initialization files and formatting them follows the same ideology as given in [3].

As in the previous project the data collection relies on the logging of data and on the statistics collection. More on how this data is tracked, collected, analyzed, and some examples can be found in [3].

8 CONCLUSIONS

This deliverable describes the integrated testbed developed in the AROMA project. Details about the protocols involved and the inter-process communication procedures between the various entities involved in the testbed are described.The document also incorporates some additional details or refinements to the testbed specifications given in D07. Also, a user guide is provided for potential users to operate the AROMA testbed.

Therefore, this deliverable provides a complete view of the integrated AROMA testbed understood as a flexible HW/SW tool able to test in a realistic but easy way the different radio resource, mobility management and QoS Management algorithms considered in a B3G heterogeneous network, that includes different RATs, (UTRAN, GERAN, WLAN), interfacing a common Core Network based on Diffserv and policy-enabled networking (PBN) with improved mobility aspects and a new framework for the E2E QoS Management. In addition to all these elements the AROMA testbed incorporates the capacity to evaluate the QoS experienced by the user when using real applications under controlled conditions of the used RAT and the CN.

REFERENCES

[1] Felipe Cabral Pinto (Ed), "*D06 Testbed Specification*", EVEREST IST-2002-001858, Deliverable 06, 13-04-2004.

[2] Antoni Gelonch and Ramon Ferrús (Ed), "*D12 Implemented Testbed: Subsystems and Modules*", EVEREST IST-2002-001858, Deliverable 12, 14-01-2005.

[3] Ramon Ferrús and Antoni Gelonch (Ed), "*D18 Integrated Testbed*", EVEREST IST-2002-001858, Deliverable 18, 08-08-2005.

[4] 3GPP TS 25.211, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical channels and mapping of transport channels onto physical channels (FDD) (Release 6)*", v6.7.0.

[5] 3GPP TS 25.212, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD) (Release 6)*", v6.10.0.

[6] 3GPP TS 25.213, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Spreading and modulation (FDD) (Release 6)*", v6.5.0.

[7] 3GPP TS 25.214, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical layer procedures (FDD) (Release 6)*", v6.11.0.

[8] 3GPP TS 25.215, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical layer - Measurements (FDD) (Release 6)*", v6.4.0.

[9] Anna Umbert (Ed), "*D07 Testbed Specification*", AROMA IST-4-027567, Deliverable 07, 03-07-2006.

[10] 3GPP TS 25.306, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UE Radio Access capabilities (Release 6)*", v6.10.0.

[11] 3GPP TS 25.308, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 6)*", v6.4.0.

[12] 3GPP TS 25.302, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Services provided by the physical layer (Release 6)*", v6.8.0.

[13] 3GPP TR 25.899, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; High Speed Download Packet Access (HSDPA) enhancements (Release 6)*", v6.1.0.

[14] 3GPP TS 25.301, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Interface Protocol Architecture (Release 6)*", v6.4.0.

[15] 3GPP TS 25.321, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Medium Access Control (MAC) protocol specification (Release 6)*", v6.12.0.

[16] 3GPP TS 25.322, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Link Control (RLC) protocol specification (Release 6)*", v6.9.0.

[17] 3GPP TS 25.323, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Packet Data Convergence Protocol (PDCP) specification (Release 6)*", v6.9.0.

[18] 3GPP TS 25.331, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification (Release 6)", v6.13.0.

[19] 3GPP TS 25.434, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Iub Interface Data Transport and Transport Signalling for Common Transport Channel Data Streams*".

[20] 3GPP TS 48.054, "*3rd Generation Partnership Project; Technical Specification Group Radio Access Network;* Base Station Controller - Base Transceiver Station: (BSC-BTS) interface; Layer 1 structure of physical circuits".

[21] "MPLS for Linux", http://mpls-linux.sourceforge.net/

[22] Iperf Project. Available: http://dast.nlanr.nat/Project/Iperf

[23] Camtasia Studio, web site: http://www.techsmith.com/camtasia.asp

[24] Apple Darwin Streaming Server, web site: http://developer.apple.com/opensource/server/streaming/index.html

[25] VideoLan Streaming Server and Player, web site: http://www.videolan.org/vlc/

[26] QuickTime Pro 7, web site: http://www.apple.com/quicktime/win.html

[27] Microsoft Windows NetMeeting, web site: http://www.microsoft.com/windows/netmeeting/

[28] Video Conferencing Tool, web site: http://www-nrg.ee.lbl.gov/vic/

[29] Robust Audio Tool, web site: http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[30] Jordi Pérez-Romero, Oriol Sallent, Ramon Agustí, Lin Wang, Hagmid Aghvami, "A Novel Algorithm for Radio Access Technology Selection in Heterogeneous B3G networks", IEEE VTC 2006-Spring, Melbourne, May, 2006.

[31] Jordi Pérez-Romero, Oriol Sallent, Ramon Agustí, "Network Controlled Cell Breathing in Multi-service heterogeneous CDMA/TDMA scenarios", IEEE VTC 2006-Fall, Montreal, September, 2006.

[32] Jordi Pérez-Romero, Oriol Sallent, Ramon Agustí, "A Novel Metric for Context-Aware RAT Selection in Wireless Multiple Access Systems", The IEEE International Conference on Communications (ICC 2007), Glasgow, June 2007.

[33] 3GPP TS 23.107, "*3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 6)*", v6.4.0.

[34] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, "RFC3588 Diameter Base Protocol" http://www.rfc-editor.org/rfc/rfc3588.txt

[35] Thi Mai Trang Nguyen Boukhatem, N. Doudane, Y.G. Pujolle, G., "COPS-SLS: a service level negotiation protocol for the Internet", IEEE Communications Magazine, May 2002 Vol. 40, Issue 5O. pp. 158-165

ACRONYMS

3GPP	3rd Generation Partnership Project
ACK	Acknowledge
AGMT	Advanced Graphical Management Tool
AMC	Adaptive Modulation and Coding
AMR	Adaptive Multi-Rate
ANP	Anchor Point
AR	Access Router
ARQ	Automatic Repeat reQuest
B3G	Beyond 3rd Generation
BB	Bandwidth Broker
BLER	Block Error Rate
CM	Communications Manager
CN	Core Network
COPS	Common Open Policy Service
CQI	Channel Quality Indicator
CR	Core Router
CRRM	Common RRM
DCH	Dedicated Channel
DiffServ	Differentiated Services
DPCH	Dedicated Physical CHannel
DSCP	Differentiated Services Code Point
ER	Egress Router
FDD	Frequency Division Duplex
F-DPCH	Fractional Dedicated Physical CHannel
FEC	Forwarding Equivalence Class
GERAN	GSM/EDGE Radio Access Network
HARQ	Hybrid Automatic Repeat request
HHO	Horizontal Handover
HS-SCCH	High Speed Shared Control CHannel
HSDPA	High Speed Downlink Packet Access
HS-DPCCH	High Speed Dedicated Physical Control CHannel
HS-DSCH	High Speed Downlink Shared CHannel
HS-PDSCH	High Speed Physical Downlink Shared CHannel
IETF	Internet Engineering Task Force
IP	Internet Protocol
IR	Ingress Router
LER	Label Edge Routers
LSP	Label Switched Path
LSR	label-switching routers
MN	Mobile Node
MPLS	Multiprotocol Label Switchingç
NCCB	Network-controlled Cell Breathing
OS	Operating System
PC	Personal Computer
PDP	Packet Data Protocol
PEP	Policy Enforcement Point
PIB	Policy Information Based
PR	Policy Repository
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technique
RLC	Radio Link Control
RNC	Radio Network Controller
RR	Round Robin
RRM	Radio Resource Management
SAW	Stop-And-Wait
SGSN	Service GPRS Support Node
SLA	Service Level Agreement

SLS	Service Level Specification
SNR	Signal to Noise Ratio
TB	Transport Block
TBS	Transport Block Set
TBSS	Transport Block Set Size
TCP	Transmission Control Protocol
TDMoIP	TDM over IP
TE	Traffic Engineering
TFT	Traffic Flow Templates
TFRC	Transport Format and Resource Combination
TPC	Transmission Power Control
ToS	Type of Service
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
UUT	User Under Test
VHO	Vertical Handover
VGNCCB	Voice GERAN Network-controlled Cell Breathing
WLAN	Wireless Local Area Network
WQB	Wireless QoS Broker