

Real-Time HSPA Emulator for End-to-Edge QoS Evaluation in All-IP Beyond 3G Heterogeneous Wireless Networks

Miguel López-Benítez, Francisco Bernardo, Nemanja Vučević, Anna Umbert
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya
Barcelona, Spain
[miguel.lopez, fbernardo, vucevic, annau]@tsc.upc.edu

ABSTRACT

This paper is aimed at presenting the real-time High Speed Packet Access (HSPA) emulator that has been developed in the framework of the AROMA project. Real-time emulators allow reproducing realistic scenarios to test algorithms, strategies, protocols and applications under realistic conditions. Therefore, real-time emulators constitute a powerful tool to evaluate the end-user's Quality of Experience (QoE), which could not be achieved by means of off-line simulations. The presented emulator is integrated in the AROMA real-time testbed, which has been developed to provide a framework for demonstrating the benefits of the common radio resource management algorithms as well as the proposed end-to-edge Quality of Service (QoS) management techniques developed for all-IP beyond 3G heterogeneous wireless networks in the context of the AROMA project. This paper presents a qualitative description of the developed tool, emphasizing some interesting implementation details that may result helpful in the development of similar emulation platforms. Some illustrative results, showing the capabilities of the developed tool, are also presented and analyzed.

Categories and Subject Descriptors

I.6.3 [Simulation and Modeling]: Applications; I.6.7 [Simulation and Modeling]: Simulation Support Systems—*environments*; J.2 [Computer Applications]: Physical Sciences and Engineering—*engineering*.

General Terms

Algorithms, Design, Experimentation, Measurement, Performance, Verification.

Keywords

All-IP networks, beyond 3G networks, end-to-edge quality of service, heterogeneous wireless networks, high speed packet access, real-time emulation, real-time testbed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

QoSIm 2008, March 3, 2008, Marseille, France.

Copyright 2008 ACM ISBN 978-963-9799-20-2 ...\$5.00.

1. INTRODUCTION

Due to the increasing user expectations and the operator's wish to be ahead of the competitors, mobile communication systems have been experiencing a constant evolution with the aim to support new services, provide Quality of Service (QoS) guarantees, and offer higher data-rates and enhanced capabilities. The recent High Speed Packet Access (HSPA) technology, conceived as a natural evolution of the already existing Universal Mobile Telecommunications System (UMTS), has emerged as one of the most promising paradigms for the achievement of the aforementioned goals in a cost-effective manner. The term HSPA is commonly used to refer jointly to two different technologies, namely the High Speed Downlink Packet Access (HSDPA) [5] for the downlink and the High Speed Uplink Packet Access (HSUPA)¹ [4] for the uplink. HSPA is standardized by the 3rd Generation Partnership Project (3GPP) and has been designed to provide higher data-rates (theoretical values up to 14.4 Mbit/s in the downlink and 5.76 Mbit/s in the uplink), reduced delays and improved spectral efficiency. These improvements ensure the competitiveness of UMTS networks in the short- and mid-term.

Although the HSDPA and HSUPA technologies are standardized, some aspects have not been defined in the specifications and have been left open to the mobile operator's choice. For instance, the scheduling policy to be applied or other radio resource management techniques are not defined. As a result, many efforts have been devoted to find appropriate and efficient solutions during the last years. The proposed solutions, however, usually have been evaluated by means of off-line simulations. The utilization of simulation tools is common within the research and industrial communities and can be useful for obtaining preliminary results. Nevertheless, to conduct meaningful and appropriate studies, and to accurately assess the performance of the proposed solutions for mobile communication systems before considering a prototype or full-scale deployment, the evaluation over realistic emulation platforms is becoming essential as a step forward toward the implementation in a real system. Real-time emulators allow reproducing realistic scenarios to test algorithms, strategies, protocols and applications under realistic conditions, constituting a powerful tool to evaluate the Quality of Experience (QoE) of the end-user, which could not be achieved by means of off-line simulations.

¹The official term employed in the specifications is Enhanced Uplink. However, the term HSUPA is widely used in the industrial and academic environments.

In this context, the aim of this paper is to present the real-time HSPA emulator that has been developed in the framework of the AROMA project [12]. The presented emulator is integrated in the AROMA real-time testbed [15], which has been developed to provide a framework for demonstrating the benefits of the (Common) Radio Resource Management (CRRM/RRM) algorithms as well as the proposed end-to-edge (e2e) QoS management techniques developed within the project for all-IP beyond 3G heterogeneous wireless networks. This paper presents a qualitative description of the developed tool, emphasizing some interesting implementation details that may result helpful in the development of similar complex emulation platforms. Due to the lack of space, this paper focuses on the HSDPA emulator implementation details. Nevertheless, most of the aspects described here are also applicable to the HSUPA emulator.

The paper is organized as follows. First, section 2 provides a general overview of the AROMA testbed where the presented HSPA emulator is integrated. Section 3 describes the implementation details of the HSDPA emulator, many of which are also applicable to the HSUPA emulator. Section 4 presents some illustrative results, showing the potentials and capabilities of the tool in evaluating the end-user perceived QoS and QoE. Section 5 discusses the applicability of the emulator. Finally, section 6 concludes the paper.

2. OVERVIEW OF THE AROMA TESTBED

2.1 General description

The AROMA testbed allows the real-time emulation of an all-IP beyond 3G heterogeneous wireless network composed of the UMTS Terrestrial Radio Access Network (UTRAN), GSM/EDGE Radio Access Network (GERAN), and Wireless Local Area Network (WLAN) as well as the corresponding common Core Network (CN) based on DiffServ technology [11] and Multi-Protocol Label Switching (MPLS) [14]. The evaluation platform emulates, in real-time, the conditions that the behavior of the all-IP heterogeneous network, including the effect of other users, produces on a given User Under Test (UUT) when making use of real multimedia IP-based applications such as video-conference, streaming, or web browsing. Such approach allows testing real applications on an e2e basis on a complete all-IP heterogeneous network with CRRM/RRM algorithms and e2e QoS management policies. The presented tool is therefore a powerful emulation platform that enables advanced CRRM/RRM strategies as well as e2e QoS mechanisms to be accurately evaluated in a realistic environment with different real user applications and mobility patterns, which could not be achieved by means of off-line simulations.

2.2 Software and hardware platform

The AROMA testbed is implemented with twenty off-the-shelf Personal Computers (PCs). Two of them run Windows operating system (the applications' PCs) and eighteen PCs run Linux operating system. This approach has been proven to be adequate for its capacity to assure appropriate levels of real-time management while guaranteeing a high degree of flexibility. The capacities provided by Linux operating system to interact at low level with the kernel offer the possibility to tune accurately the performance required by the testbed, especially in the issues related with the real-time execution and management. To implement real-time operation

a very high computational power is required. These computational requirements are out of the scope of today's off-the-shelf PCs. Then, a cluster of PCs has been constructed to distribute the computational load throughout different processors. To this end, a software tool named Communications Manager (CM) was designed and implemented to make this distribution completely transparent.

Figure 1 shows all the entities and connections of the AROMA testbed. Black connections correspond to user data interfaces, whereas red and blue connections correspond to control plane interfaces. The UUT has at his disposal two stand-alone PCs: one PC (application's client) is used to run the UUT's application, e.g. a commercial web browser or streaming video player, while the other PC is used to run the main functionalities associated to the User Equipment (UE). This second PC provides a graphical interface through which a session can be activated, deactivated or modified in real-time during the testbed execution, specifying different QoS parameters (e.g., conversational, streaming or interactive QoS class, or requested throughput in uplink and downlink). This interface enables the human interaction with the testbed in real-time. To test symmetric services as video-conference and to serve multimedia applications such as web-browsing or streaming, a correspondent node (application's server) is run in a stand-alone PC. The three mentioned Radio Access Networks (RANs) are emulated using three PCs for UTRAN (uplink, downlink, and common functions), one PC for GERAN and one PC for WLAN. The CN has been built using seven Linux PCs acting as routers: three PCs serve as edge routers, two Ingress Routers (IRs) and one Egress Router (ER), and four PCs, identified as Core Routers (CR), interconnect the edge routers. A Traffic Switch (TS) is mainly used to establish different connection configurations between RANs and the IRs in the CN. It captures the UUT's IP packets, passes them to the appropriate RAN (where the UUT is connected to) to make the real-time emulation and re-injects them in the interface of the IR where the RAN is supposed to be connected to. For the emulated users passing through the testbed there is a PC called Traffic Generator (TG) that is in charge of generating real IP traffic to load the core network in accordance with the traffic amount that active users generate in the system. Obviously, generation of this traffic is coordinated with the traffic emulated in the radio part. Finally, a graphical management and configuration tool called Advanced Graphical Management Tool (AGMT) has been developed to configure the initialization parameters, to control the execution flow, to collect logged data and to obtain statistics in real-time during the testbed execution. The yellow area in figure 1 includes all the machines controlled by the AGMT.

2.3 Quality of service framework

The end-user perceived QoS and QoE can be assessed once the real IP packets of the UUT have passed through the testbed and have been impaired according to the RAN and CN status. Moreover, this framework allows the testing of particular implementation of the QoS entities which may be important for operators before applying these implementations to real networks. The QoS management is provided by the Wireless QoS Broker (WQB) entity in the RAN part and by the Bandwidth Broker (BB) in the CN part. Additionally, other entities such as the QoSClient (i.e., the QoS negotiation interface for the UUT) and the CRRM module

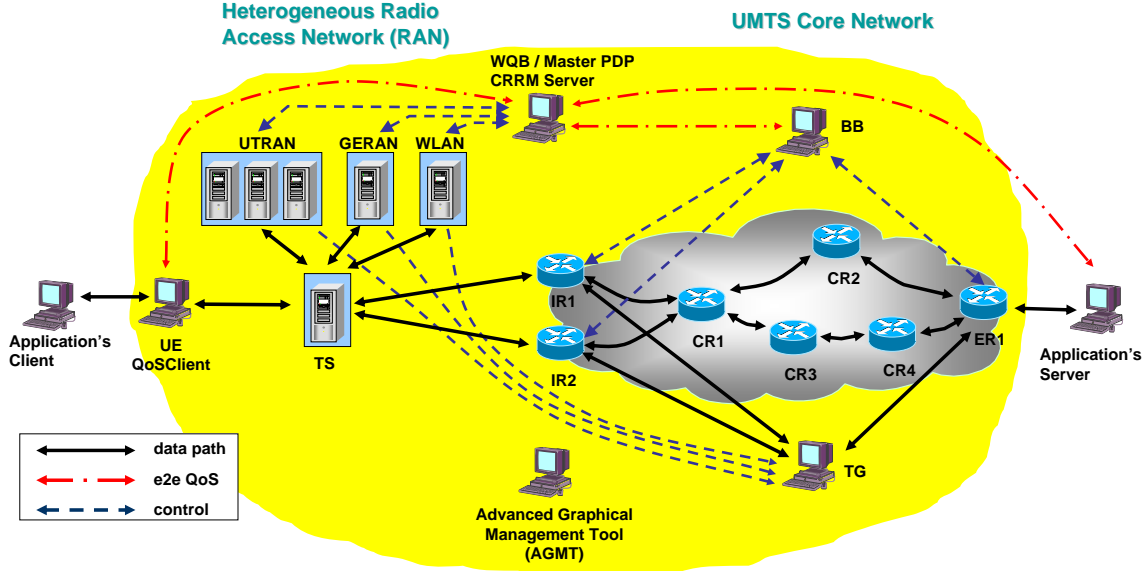


Figure 1: Entities and connections of the AROMA testbed.

(in charge of managing the radio resources and running the radio admission and congestion control algorithms) are also involved in the e2e QoS negotiation.

In the testbed, the e2e QoS support is enabled by the interaction between these entities. As it can be observed in figure 1, WQB leads the QoS negotiation procedures. When it receives a QoS request from the QoSClient, it consults CRRM and BB about their QoS capabilities for the acceptance of a new session. Then WQB finally takes decisions on session establishment (or modification) based on the information provided by the CRRM and BB. In fact, the information that CRRM generates during the e2e QoS negotiation actually bases on admission control and RAN selection algorithms, whereas the BB decision is based on its proper Connection Admission Control (CAC) algorithms. Thus, it can be seen that the WQB includes the functionalities of a Master Policy Decision Point (MPDP), since it acts as a master broker taking the final decision on the acceptance of a new user flow. A more detailed description of the implemented AROMA testbed can be found in [10].

3. HSPA REAL-TIME EMULATOR

3.1 Supported features

The presented HSPA emulator has been designed to cope with the following goals and capabilities:

- *Support for live users as well as fully emulated users.* The emulators are designed to reproduce in real-time the behavior of a relatively large amount of active users, around several thousands of users depending on scenario and traffic configuration. The traffic for both the UUT (real IP traffic) and the rest of users (internally generated by means of traffic modeling) is processed by the emulator. Therefore, the UUT behaves as any other user in the system where processing of its

data differs along time depending on the current RAN status, i.e. load conditions, interference level, etc.

- *Emulation of the transmission chain between the User Equipment (UE) and the Radio Network Controller (RNC).* The emulator accounts for the main features of the radio interface as well as RRM functions. The different functions performed at each level of the protocol stack have been accurately modeled in accordance with the specifications. Physical layer emulation has been addressed by means of curves obtained from extensive off-line link level simulations in order to reduce computational requirements while preserving realistic behavior. The functionalities related to higher layers in the protocol stack have been implemented in detail in order to ensure a realistic real-time behavior of the emulator under dynamically varying conditions.
- *Execution of RRM functions and support for CRRM capabilities.* RRM functions implemented in the emulator include essential functions such as admission control, radio resource allocation, handover management or transmission parameters management. Although only a single UUT is running real applications on the testbed, RRM algorithms are applied indistinctly over all the traffic generated, including the UUT and also the rest of emulated users. The support of the needed CRRM functions is achieved by means of a communication interface provided by the CM between the emulator and the WQB machine.
- *Support for different communication scenarios.* The definition of the scenarios takes into account the cell site deployment, radio environment, mobile distribution, user movement, etc. The considered scenarios are mainly based on the requirements and visions of the four operators that participate in the AROMA project.

3.2 Implemented aspects

As mentioned in section 1, this paper focuses on the implemented aspects of the HSDPA emulator. The main features of HSDPA Release 6 that have been taken into account in the emulator as well as specific implementation issues are described in this subsection.

3.2.1 Channels required for HSDPA operation

Several new channels are introduced in 3GPP specifications for HSDPA operation [6]. For user data the High Speed Downlink Shared CHannel (HS-DSCH) is defined. The HS-DSCH is a code- and time-shared channel operating at a fixed spreading factor of 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 [8], allowing the utilization 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 determines the periodicity at which Transport Blocks (TBs) are transferred by the physical layer on the radio interface [3]. The TTI also determines for instance the frequency of the scheduling or adaptive modulation and coding 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 HSPA emulator.

For the signaling associated to HSDPA operation, two new channels are introduced: 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 carries signaling information (the codes assigned by the scheduler and the modulation employed) required by the UE to demodulate the correct codes in the HS-DSCH. Each HS-SCCH consumes a code with a constant spreading factor of 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. 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 signaling by the system parameter k (see for example [1]). 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 signaling 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. 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 of 256, and can be shared among several users, up to ten different users with different frame timing, thus reducing code space utilization for the associated Dedicated Channels (DCHs).

Only transmissions over the HS-DSCH channel are fully emulated in the HSPA real-time emulator on a per-TTI basis. Transmissions over HS-SCCH, HS-DPCCH and F-DPCH channels are not emulated. However, the functional-

ities of these channels are fully preserved in the emulation model and the existence of such channels is carefully taken into account in terms of power and code space consumption.

3.2.2 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 characterized 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 [7], by the introduction of a new MAC sublayer, known as MAC-hs, located at the Node-B. These three techniques are briefly summarized in the following subsections, highlighting some specific implementation aspects.

Adaptive Modulation and Coding (AMC).

In Release 99 DCH, a fast power control mechanism is used in order to compensate the 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 favorable 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 are provided. 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 TB size, modulation and number of parallel channelization codes. The full set of combinations can be found in [2], where different tables for the different existing UE categories are provided. The CQI reported by the UE is the highest supported value of CQI that can result in a Block Error Rate (BLER) not exceeding a target value equal to 10%. The CQI value is computed based on instantaneous measurements of the Common Pilot CHannel (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 (carrier to interference ratio) for each possible CQI value. These tables have been obtained from the European Network of Excellence NEWCOM [13].

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 modeled as a simple delay.

Hybrid Automatic Repeat reQuest (HARQ).

The retransmission functionality in Release 99 is implemented at RLC level. When a data packet is not correctly received by the UE, it is discarded and a negative RLC acknowledgment is sent requesting a retransmission. 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 layer 1 (L1) retransmission, which automatically takes place from Node-B. 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 for 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.

The retransmission strategy implemented is based on the *N*-channel Stop-And-Wait (SAW) protocol. 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 [13] 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 2 for the case of CQI 0 (similar curves are used for the other CQIs). As it can be noted, 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 modeling 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. Since acknowledgments are not instantaneous, after every transmission the transmitter should wait to receive the acknowledgment prior to transmitting the next block. In a slotted system, at least every other slot must go idle even on an error free channel. By parallelizing *N* SAW HARQ processes no system capacity goes wasted since one instance of the HARQ protocol communicates a data block on the forward link at the same time that other communicates an acknowledgment on the reverse link. The optimum number of parallel SAW channels should be set so that a continuous flow of information is guaranteed at the same time the buffering requirements in the UE are minimized. The number of parallel SAW channels may maximally be 8, but in practice it will be around 4 to 6.

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

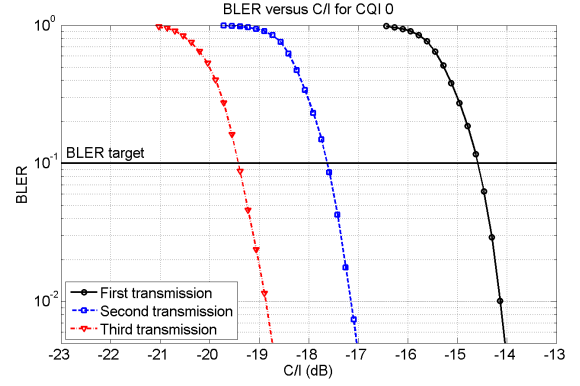


Figure 2: BLER versus C/I for CQI 0.

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 implemented scheduling approach is divided in two phases. In the first phase, the list of transmission requests is ordered according to the service's type: conversational services are of highest priority, then streaming, interactive and, finally, background services. In the second phase, a given scheduling policy is employed to establish priorities among requests for the same service. The HSPA emulator has been designed to be able to incorporate new scheduling strategies in a easy and modular way.

3.3 IP-RAN emulation model

The HSPA emulator presented in this paper is integrated in the AROMA real-time testbed, which comprises a complete all-IP beyond 3G heterogeneous wireless network. In all-IP networks, IP transport is employed not only in the CN part, but also in the RAN part. Existing Iub interfaces for UTRAN are kept between base stations and RNCs, but they are supported over an IP-based packet-switched network. As a consequence of such approach, a data block can be lost at Node-B not only because of unfavorable radio conditions but also due to transport network losses or excessive delays. Therefore, a model for emulating the effects and impairments of the IP-based transport in the RAN has been implemented. The envisaged IP-RAN emulation model takes into account losses in the transport network, obtained from non-real-time simulations, as shown in figure 3. In these off-line simulations, a data block is discarded at RNC if it arrives later than a maximum predefined delay δ_{max} . In order to assure the validity of such approach, the δ_{max} value considered must be lower than the acknowledgment delay at the Radio Link Control (RLC) layer minus the TTI. The loss statistics depend 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). These loss statistics obtained from off-line simulations are used to determine, for different scenarios, the probability that an IP packet is lost in the IP-RAN transport network, which is used to decide in real-time whether a packet is discarded due to IP transport impairments.

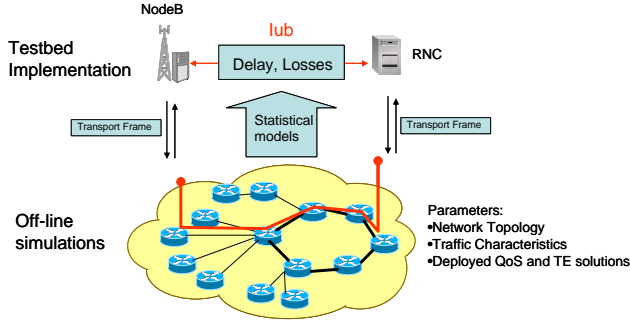


Figure 3: IP-RAN emulation model.

3.4 Interaction between the HSPA emulator and the rest of the AROMA testbed

The interaction between the HSPA emulator and the rest of the AROMA testbed is accomplished through two different planes: a control plane and a data plane.

The control plane supports all the functionalities needed for exchanging control messages between the HSPA emulator and other modules such as the CRRM and TG. To this end, the CM abstraction layer provides the logical concept of *flow*. During the initialization phase of the testbed, each module, including the HSPA emulator, creates a flow between itself and all modules to/from which control messages need to be sent/received. Thereafter, modules can write control packets in the flow addressed to the destination module. All this message exchanging process is managed by the CM in a completely transparent way. For more details, see [10].

As shown in Figure 1, the UTRAN emulation PCs manage two control interfaces, one with the CRRM PC and other with the TG PC. The control interface between UTRAN and TG is used by the UTRAN emulation PCs to communicate periodically to the TG the instantaneously experienced traffic load for each service. This information is used by the TG to load the CN with a traffic level according to that experienced in the RANs. On the other hand, the control interface between UTRAN and CRRM is mainly used for session management. A session activation message is sent from CRRM to UTRAN whenever a new session is established within UTRAN, see figure 4(a). This event may occur for the UUT when a new session is activated through the QoSClient graphical interface and for the rest of users according to the traffic generation model implemented for each service. For all users, a session activation message may also result from a vertical handover from any RAN to UTRAN. This message carries information about the user, the required service type and QoS requirements, the base station the user should be attached to and the geographical position of the mobile within the scenario. Similarly, a session modification message may be sent from CRRM to UTRAN whenever the QoS requirements change (due to an e2e QoS renegotiation event or because the UUT requests new QoS parameters), the mobile position has to be updated (the periodicity of such updating will depend on the mobile speed) or the active set changes. A session deactivation may be initiated by both CRRM and UTRAN. Deactivation from CRRM, illustrated in figure 4(b), occurs for the UUT when the session is deactivated through the QoSClient graphical interface. For the rest of users, the traffic generation mod-

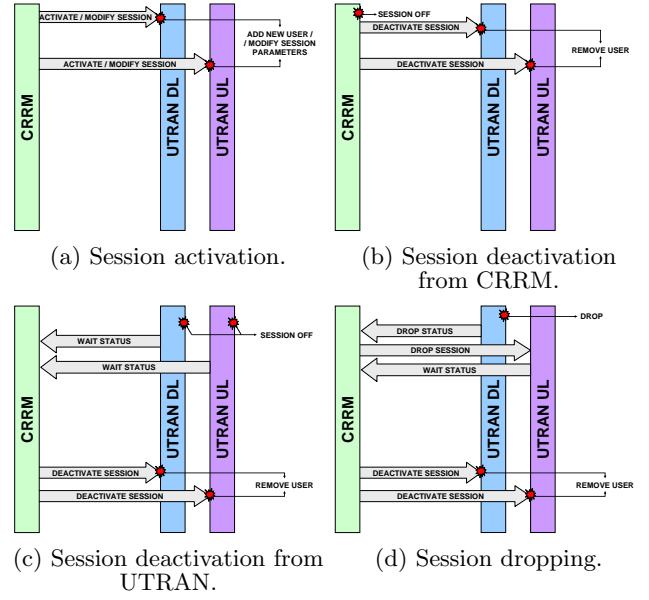


Figure 4: Interaction between emulator and CRRM.

els implemented in the RANs decide the end of each session and advertise the CRRM as shown in figure 4(c). Finally, UTRAN may decide to drop a specific established session for any user, e.g. if QoS requirements are not satisfied. The message exchanging in this case is illustrated in figure 4(d).

The data plane comprises all the functionalities needed to support the transmission of real IP packets for the UUT through the testbed. As shown in Figure 1, a data interface between the UTRAN emulation PCs (HSPA) and the TS module is defined. The interaction between these two modules is qualitatively illustrated in figure 5. In the downstream direction, i.e. from the CN to the UE, real IP packets of the UUT coming from the CN are captured by the TS and are stored in a data buffer. Some descriptive parameters regarding the packet (e.g., a packet identifier, the packet size, the service type, or the QoS requirements, among some others) are sent to the HSPA emulator by making use of the communication interface provided by the CM. The HSPA emulator maintains a data structure emulating buffers of all the users (UUT and emulated users). This data structure is updated upon the arrival of a new real IP packet (indicated by the TS message, only for the UUT) or whenever a new data packet is generated by the traffic generation model implemented in the RAN emulator (for emulated users). The radio transmission of each packet is emulated on a per-TB basis taking into account several system-level aspects such as the cell site deployment, number of emulated users, mobility patterns, propagation impairments, and so on. The result of an IP packet transmission is communicated back to the TS through the interface provided by the CM. The message sent to the TS indicates a packet identifier and the result of the emulation, i.e. correct or incorrect transmission. Then, the TS forwards the packet to the UE or discards the packet, depending on the transmission result obtained in the HSPA emulation of that packet. The described procedure also applies for IP packets in the upstream direction, i.e. from the UE to the CN. This procedure is completely managed in real-time. Notice that this emulation approach is able not

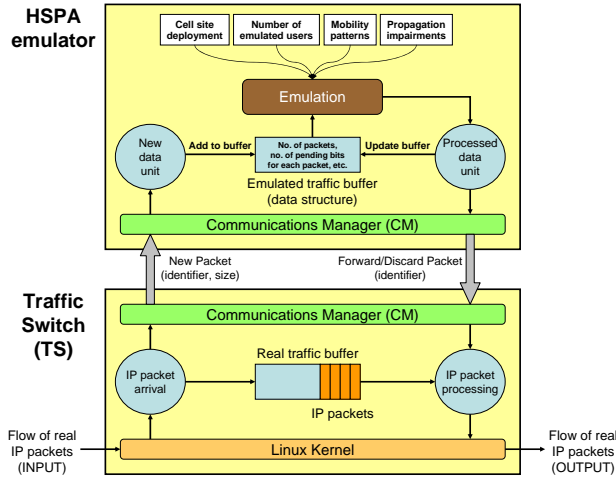


Figure 5: Interaction between emulator and TS.

only to reflect the loss of packets incorrectly transmitted (some packets arriving to the TS are not forwarded to their destination) but also the delay experienced by packets correctly transmitted (the TS does not forward a packet until the HSPA emulator indicates it, and the HSPA emulation is performed in real-time).

3.5 HSPA real-time execution

The AROMA testbed modules have five operating states: setup, init, run, pause and stop. The execution flow through these states is controlled from the AGMT (see section 2). During the setup state the AGMT verifies the reachability of all the testbed machines and establishes the communication with each one of them. During the init state the processes executed in each PC load configuration parameters and initialize resources (e.g., memory or communication flows with other modules). Once modules are initialized, they switch to the run state, which is the normal execution state and will be detailed below. The testbed execution may be paused to allow the statistics to be analyzed (during the pause state the processes in execution are frozen). Finally, during the stop state all the resources used by the processes are released and the testbed execution is finished.

The run state is the normal operation mode. During this state, several emulation PCs (including the UTRAN/HSPA modules) perform a set of specific functions that are periodically repeated every certain time interval. The execution loops of these machines run perfectly synchronized, meaning that at the beginning of every new time interval the modules execute their respective functions and then remain idle until the beginning of the next time interval. This synchronization is accomplished with the aid of a special *sync* process executed on all the machines that must be synchronized among them (see annex B of [9] for more details). The overall testbed operates with a periodic time interval of 10 ms, i.e. each software module follows an internal 10 ms execution loop that is synchronized with the execution loops of other machines. Although this value is appropriate for emulating in real-time the UTRAN TTIs (10, 20, 40 and 80 ms), even the 10 ms HSUPA TTI, this granularity is insufficient for emulating the 2 ms TTI defined for

HSDPA/HSUPA in 3GPP specifications. A possible solution would consist on a reduction of the execution period of the testbed to 2 ms. However, this option was found to show some important problematic aspects. First of all, it is worth mentioning that the software platform of the testbed is mainly based on Linux, and the short 10 ms execution period of the testbed is not completely independent of the Linux kernel process-scheduling time (see annex B of [9] for more details), which imposes some constraints to the minimum execution period that can be employed. Moreover, a reduction of the execution period from 10 ms to 2 ms would imply that the computational load would be significantly increased since the code of all the software modules would be 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 (required to force specific load conditions where the performance of the strategies under study could be evaluated), and the fact that the considered scenarios may consist of a high number of cell sites, this approach was found to be definitely unfeasible. A different approach was adopted instead. The main idea is to maintain the 10 ms execution loop and obtain a virtual 2 ms execution loop by repeating the HSPA-related functions five consecutive times within the 10 ms execution loop. Notice that although the HSPA 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 was proven to allow an adequate real-time emulation of HSPA, with 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.

For the case of the HSDPA emulator, the execution loop during the run state is illustrated in figure 6 (an analogous scheme is defined for the uplink counterpart, i.e. HSUPA, not analyzed here due to the lack of space). The functions shown in figure 6 are repeated periodically five times every 10 ms. As it can be appreciated, the first step in each iteration is to update mobiles' position within the scenario, according to the mobility model implemented. Afterwards, the propagation conditions for the new position of each mobile are updated. To this end, a radio propagation matrix is used. This propagation matrix (specific for each emulated scenario) is computed during the init state and contains the propagation losses experienced between a set of discrete geographical locations and each one of the base stations included in the emulation scenario. Every execution loop, the new position of each mobile is approximated to the nearest point of the propagation matrix, and the new propagation conditions are derived from the matrix. The distance between consecutive points of the propagation matrix is equal to the de-correlation distance of the fading model. Notice that all the possible propagation conditions are computed during the init state and read during the run state, instead of computing them in real-time. This approach reduces the real-time computational requirements and therefore results more appropriate for real-time emulation. Depending on the new propagation conditions, the active set might be updated and a handover between cells might be required.

The next step is to read and process messages coming from CRRM (see figure 4) and TS (new real IP packets for the UUT). The traffic for the rest of users (emulated users) is

then generated according to the traffic models implemented for each service. Later on, all users with non-empty buffers are added to a list of users requesting a transmission. Before scheduling them, the received CQI reports are processed (this information might be required by the scheduling algorithm) taking into account the 3 TTI delay mentioned in section 3.2. The implemented scheduling approach is divided in two phases. In the first phase, the list of transmission requests for each Node-B is ordered according to the service type: conversational services are of highest priority, then streaming, interactive and, finally, background services. In the second phase, a numerical metric is employed to establish priorities among requests for the same service. This numerical metric depends on the selected scheduling policy, e.g. a mobile has higher priority if it has not been served during longer time periods (round robin), has reported a higher CQI value (maximum C/I), provides a higher quotient between the maximum instantaneously supported data rate and average served data rate (proportional fairness), or experiences a higher difference between the average supported data rate and the guaranteed data rate (minimum guaranteed bit-rate). All these scheduling algorithms are currently implemented in the HSDPA emulator, and new strategies can be easily incorporated since the only difference among algorithms with the current implementation is the formula used to compute the numerical metric. Some users are then selected by the scheduler according to the capabilities of their UE HSDPA category (minimum inter-TTI, maximum number of HS-PDSCHs), available resources (HS-PDSCHs, power), and some other aspects. The traffic load produced by the selected users is then communicated to the TG. Afterwards, the radio transmission of one TB for each selected user is emulated in each iteration taking into account the cell site deployment, number of emulated users, mobility patterns and propagation impairments. All these parameters determine the actual CIR experienced during a radio transmission, which is used to decide whether a transmitted TB is received in error or not. To this end, a curve similar to that shown in figure 2 is used, depending on the actual CQI employed to transmit the TB and the number of previous transmissions of that TB. Whenever an IP packet is completely transmitted without errors, a forward message is sent to the TS. Similarly, if either some TBs are lost or the IP-RAN model indicates that an IP packet is discarded, then a discard message is sent to the TS. Finally, CQI reports, based on the experienced channel quality (measured over the CPICH) are sent. These reports are sent for active users every TTI, and for inactive users every k TTIs.

4. ILLUSTRATIVE RESULTS

This section provides some illustrative results aimed at showing the capabilities of the developed emulator to analyze in real-time the system performance and user perceived e2e QoS/QoE. To this end, a simple case study comparing the behavior of two basic scheduling policies is presented. In particular, the Maximum C/I (MaxC/I) and Round Robin (RR) scheduling algorithms are considered. The MaxC/I criterion allocates resources to those UEs experiencing the best instantaneous channel quality conditions. This approach achieves high cell throughput values at expense of the user throughput distribution. Users located at cell border may not be served at all while users experiencing good transmission conditions may monopolize the resources. On

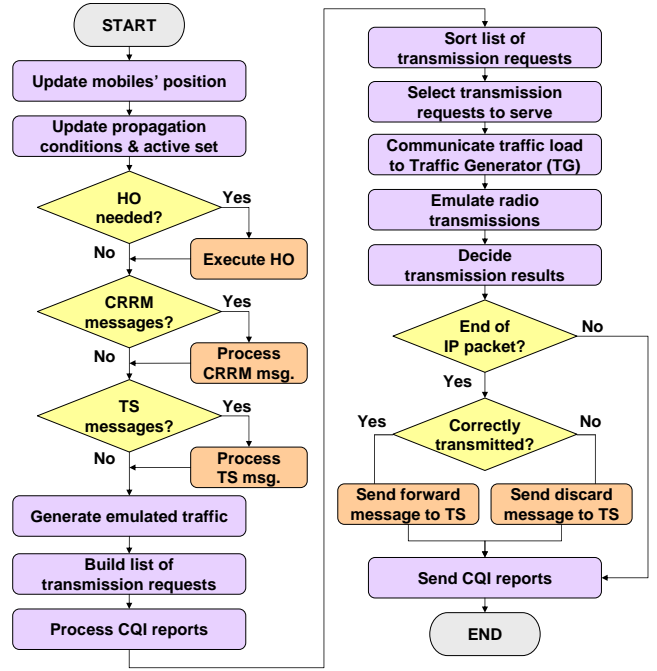


Figure 6: HSDPA real-time execution loop.

the other hand, with the RR criterion resources are assigned to users on a sequential and cyclic basis; knowledge of the experienced channel quality is not made use of. This policy offers a fairer throughput distribution among users at expense of the overall system throughput. The objective of this sample study is to show the aforementioned behavior in real-time and analyze the impact on the end-user perception.

The main configuration parameters are shown in table 1. It is worth mentioning that the maximum number of HS-PDSCHs that can be assigned to streaming users is limited, since the more constant traffic pattern of streaming services may result in the monopolization of the HS-DSCH resources.

Table 1: Main configuration parameters.

Parameter	Value
Service area	8 km × 4 km
Nodes-B / Cell radius	13 / 1500 m
CPICH power	30 dBm
Max. HS-DSCH power	30 dBm
No. of HS-PDSCHs	15
Max. no. of HS-PDSCHs for streaming	12
F-DPCH	Disabled
Streaming / Interactive emulated users	150 / 200 @ 400 kbit/s
UUT service	Streaming @ 400 kbit/s
Max. no. of transmissions	3 for streaming 5 for interactive
HARQ type	Chase combining
UUT HSDPA category	12 (5 HS-PDSCHs max.)
Emulated UEs HSDPA category	12 (5 HS-PDSCHs max.)
CQI update interval k	40 TTIs (80 ms)

The UUT periodically moves in straight line between two base stations, thus experiencing different channel quality conditions. After 25 seconds of emulation, the UUT requests a streaming session. The session is accepted and at time instant 30sec the video sequence starts reproducing. The obtained results are shown in figures 7 to 10. In figures 7 and 8, the graphs on the left show (from top to bottom) the abscissa coordinate of the UUT within the service area (between two base stations), the CIR measured for the pilot in dB, and the UUT's CQI report (the maximum CQI value for HSDPA category 12 is equal to 15 - higher CQI values correspond to the same transmission parameters [2]). As it can be observed, the measured pilot signal changes rapidly but in average it varies according to the UUT's position: the most favorable channel quality conditions are observed when the UUT is near to a base station (time instants 0, 130 and 260 sec) while the poorest channel quality is experienced at the cell border (time instants 65 and 195 sec). This behavior is also observed in the CQI report sent by the UUT, which follows the measured pilot strength.

The behavior of the two considered scheduling algorithms can be observed in the top-right side of figures 7 and 8. These graphs show the number of TTIs (measured over 10 TTI/20 ms periods) that the UUT requests transmission in (blue line) and the number of times that these requests are accepted by the scheduler (green line) or rejected (red line). As it is expected, the MaxC/I behavior is highly dependent on the experienced channel quality: under favorable channel quality conditions all the UUT's requests are accepted, while several requests are rejected when the UUT is at the cell border (time instants around 65 and 195 sec). On the other hand, the RR policy exhibits a more homogeneous and channel-independent pattern. In average, around 80% of the UUT's requests are accepted (20% rejected) under both favorable and unfavorable channel quality conditions. This behavior is also observed in the number of physical codes (HS-PDSCHs) that the scheduler assigns to the UUT, which is shown in the bottom-right side of figures 7 and 8 (measured over 10 TTI/20 ms periods). At the cell border, the number of HS-PDSCHs assigned to the UUT is limited by the maximum number of HS-PDSCHs corresponding to the reported CQI value (see [2]). As a result, a low number of HS-PDSCHs is assigned to the UUT in time instants around 65 and 195 sec with both MaxC/I and RR. On the other hand, under good channel quality conditions, data transmissions can be reliably performed using the maximum number of simultaneous codes allowed by the UE's HSDPA category (5 codes for category 12). In this case, the main aspect limiting the number of HS-PDSCHs assigned to the UUT is the scheduler's policy. As it can be observed for RR under good channel quality conditions, the UUT is hardly assigned up to 40 codes every 10 TTIs, i.e. 4 codes per TTI in average. On the other hand, the value of this parameter for MaxC/I under similar channel quality conditions usually reaches the average of 5 codes per TTI. This is due to the fact that users experiencing favorable channel conditions tend to monopolize resources when the MaxC/I algorithm is applied.

The impact of this behavior on the instantaneous throughput experienced by the UUT is shown on the top-right side of figures 9 and 10. At the cell border, a low throughput is observed for both scheduling algorithms due to the poor channel quality. However, under good conditions, the peak

and average data-rates observed for MaxC/I are significantly higher than those for RR. Another interesting statistic is shown in the top-left side of figures 9 and 10. These graphs show the histogram of the UUT's CQI usage. One significant parameter is the percentage of times that the highest CQI, i.e. CQI 15, is used. The value of this parameter is 39% for MaxC/I and 28% for RR, which also explains the higher data rates observed for MaxC/I. This is due to the fact that with the MaxC/I criterion, users are scheduled to transmit when the channel shows a good quality level. In these cases, the highest CQI is usually selected. Under the RR criterion, the time instant in which users are scheduled is channel-independent, and sometimes the instantaneous channel quality may forbid the use of high CQI values.

To analyze the impact of the instantaneously experienced throughput on the user perceived QoE, figures 7 and 8 show some screenshots of the transmitted video sequence during three different phases. The first phase corresponds to the first handover between cells (poor channel quality conditions) at time instants around 65 sec, whose consequences are observed some seconds later on. The second phase corresponds to time instants around 150 sec in which the experienced channel quality is favorable. Finally, the third phase corresponds to the second handover between cells at time instants around 195 sec. During the first phase, some degradation is observed in the image quality. This is due to the throughput reduction experienced as a result of the poor channel quality conditions. Although this behavior is observed for both scheduling algorithms, a more serious degradation is observed for MaxC/I since the number of transmission requests rejected during the first phase is higher in the MaxC/I case than in the RR case (top-right side of the figures). During the second phase, the video sequence is transmitted in both cases without noting any distortion since the channel quality conditions are favorable. The higher peak data-rates offered by the MaxC/I scheduler during this phase are not reflected in the instantaneous image quality. In fact, the image quality during the second phase for the RR scheduler is as good as for the MaxC/I scheduler. The difference becomes apparent, however, during the third phase, when the second handover occurs. As a result of the higher data-rate experienced during the second phase when the MaxC/I criterion is applied, the UUT's reception buffer is almost full by the time the handover occurs. Thus, although some transmission requests are rejected during the third phase, the sequence can be reproduced without loss of continuity and no degradation is observed. In the RR case, the data-rates experienced during the second phase are not as high as in the MaxC/I case, and the UUT's buffer is not so full by the time the second handover occurs. As a result, there exist some time instants in which the UUT's buffer gets empty, which indeed leads to the image degradation observed in the third screenshot of figure 8. It is worth highlighting that this kind of appreciations are not possible with off-line simulations.

Although this section has focused on the UUT performance, many global system performance aspects can also be analyzed with the developed tool. For instance, figures 9 and 10 show the histogram of the CQI usage at cell level (bottom-left) and the experienced cell throughput (bottom-right) for one base station. In this later case, it can be observed that the MaxC/I criterion achieves a higher cell throughput than the RR criterion.



Figure 7: Real-time performance of the Maximum C/I scheduling criterion.

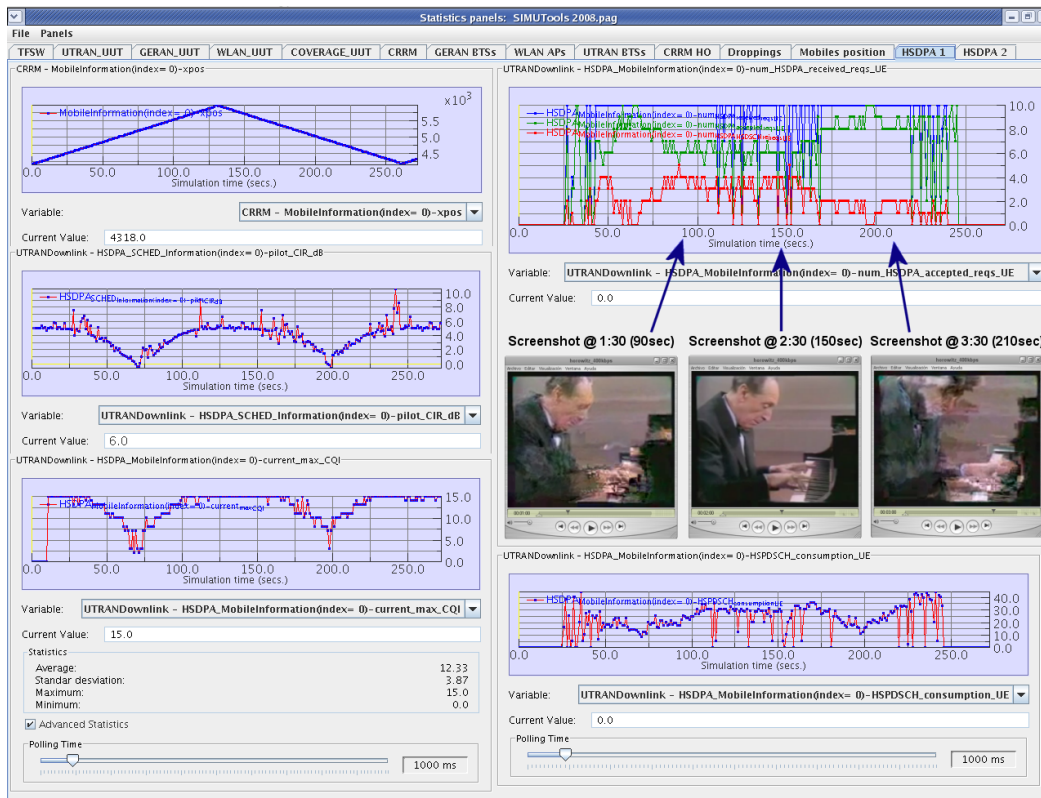


Figure 8: Real-time performance of the Round Robin scheduling criterion.

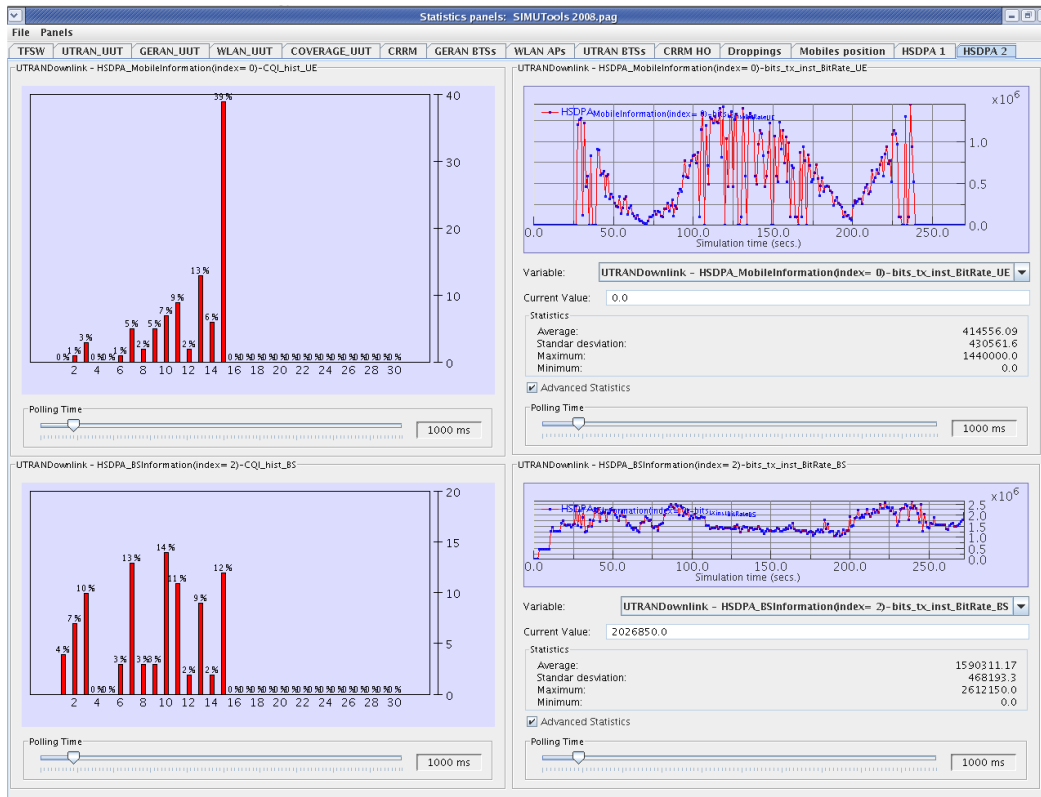


Figure 9: Real-time performance of the Maximum C/I scheduling criterion.



Figure 10: Real-time performance of the Round Robin scheduling criterion.

5. APPLICABILITY OF THE TOOL

The AROMA HSPA emulator comprises a multi-user and multi-cellular mobile environment, with physical layer emulation, standard-compliant protocol implementations, and comprehensive system-level scenarios that accounts for specific cell layouts, base station deployments, sectorized/omni-directional antennas and beam patterns, transmitter and receiver configurations, and large number of mobile terminals with accurate mobility and traffic models.

The emulator has been developed to satisfy the following main requirements: 1) *Real-time emulation*: Conventional off-line system simulators are not well suited for investigating real-time services and multimedia applications on dynamically varying radio environments. Certain effects such as end-user perception, e2e QoS, QoE or human interactions with the network (i.e., how the user reacts to the network performance), cannot be fully understood with off-line simulation results and/or analytical studies; 2) *Comprehensive and flexible tool*: The emulator models a comprehensive system-level network that can be easily configured to reflect a wide range of scenarios, thus allowing a system engineer to validate and optimize its design and algorithms before bringing them to a real system; and 3) *Standard compliancy*: The emulator models the behavior of the network in a realistic manner and without oversimplifications of essential components such that it truly reflects a real system.

Widely available commercial applications can be tested or employed to carry out real-time studies. For instance, the results shown in section 4 were obtained running Darwin Streaming Server in the application's server and QuickTime player in the application's client. Nevertheless, any IP based applications can be run over the testbed (video streaming and conferencing, voice over IP, gaming, web browsing, FTP, e-mail and so on). An application developer can make use of the platform to test the end-user perception and to optimize the application behavior for a particular scenario.

The AGMT graphical interface provided by the testbed simplifies these tasks. AGMT is used to easily control the testbed execution, configure modules, collect and observe statistics in real-time, and save trace files for further analysis. An extensive set of performance measurements or state information can be extracted from the emulator and visualized in real-time. Moreover, the effect of changing certain parameters can immediately be recognized allowing an in-depth system understanding.

6. SUMMARY

This paper has presented the real-time High Speed Packet Access (HSPA) emulator that has been developed in the framework of the AROMA project. The presented emulator is currently integrated in the AROMA real-time testbed, which was designed to assess in real-time the performance of common radio resource management algorithms as well as end-to-edge Quality of Service (QoS) management solutions for all-IP beyond 3G heterogeneous wireless networks. Real-time emulators allow reproducing realistic scenarios to test algorithms, strategies, protocols and applications under realistic conditions. Therefore, real-time emulators constitute a powerful tool to evaluate the end-user's Quality of Experience (QoE), which could not be achieved by means of off-line simulations. This paper has presented a qualitative description of the developed tool, emphasizing some inter-

esting implementation details that may result helpful in the development of similar platforms. Some illustrative results, showing the potentials and capabilities of the presented emulator for evaluating in real-time the system behavior and user perception, have also been presented and analyzed.

7. ACKNOWLEDGMENTS

This work has been performed in the framework of the IST AROMA project, which is partly funded by the European Community and by the Spanish Research Council (CICYT) under grant TEC2006-26873-E, and of the COSMOS project TEC2004-00518.

8. REFERENCES

- [1] 3GPP. *High Speed Downlink Packet Access (HSDPA) enhancements (Release 6)*, TR 25.899, v6.1.0, September 2004.
- [2] 3GPP. *Physical layer procedures (FDD) (Release 6)*, TS 25.214, v6.11.0, December 2006.
- [3] 3GPP. *Services provided by the physical layer (Release 6)*, TS 25.302, v6.8.0, October 2006.
- [4] 3GPP. *Enhanced uplink; Overall description; Stage 2 (Release 7)*, TS 25.319, v7.3.0, October 2007.
- [5] 3GPP. *High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 6)*, TS 25.308, v6.4.0, April 2007.
- [6] 3GPP. *Physical channels and mapping of transport channels onto physical channels (FDD) (Release 6)*, TS 25.211, v6.8.0, September 2007.
- [7] 3GPP. *Radio interface protocol architecture (Release 6)*, TS 25.301, v6.5.0, October 2007.
- [8] 3GPP. *UE Radio Access capabilities (Release 6)*, TS 25.306, v6.12.0, October 2007.
- [9] M. T. Aparicio et al. *Implemented testbed: subsystems & modules, Deliverable D12, IST EVEREST Project*, January 2005.
- [10] D. P. Audsin et al. *Integrated testbed, Deliverable D16, IST AROMA Project*, July 2007.
- [11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An Architecture for Differentiated Services, RFC 2475*, December 1998.
- [12] IST AROMA Project. (*Advanced Resource Management Solutions for Future All IP Heterogeneous Mobile Radio Environments*), 6th Framework Program of the European Community, <http://www.aroma-ist.upc.edu>.
- [13] NEWCOM Project. (*Network of Excellence in Wireless Communications*), 6th Framework Program of the European Community, <http://newcom.ismb.it>.
- [14] E. Rosen, A. Viswanathan, and R. Callon. *Multiprotocol Label Switching Architecture, RFC 3031*, January 2001.
- [15] A. Umbert, M. López-Benítez, F. Bernardo, N. Vučević, R. Azevedo, and A. Oliveira. The real-time AROMA testbed for all-IP heterogeneous wireless access networks. In *4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2008)*, March 2008.