ARROWS UMTS Test-bed Description

Ramon Ferrús[*], Anna Umbert[*], Xavier Revés[*], Ferran Casadevall[*], Ronan Skehill[+], Ian Rice[+]

*Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona (Spain);
e-mail: ferrus@xaloc.upc.es
+University of Limerick, Plassey Technological Park, Limerick (Ireland)
e-mail: Ronan.Skehill@ul.ie

ABSTRACT

The analysis of radio resource management (RRM) strategies in a UMTS network and the impact that these strategies have on the end-to-end behaviour of multimedia applications with Quality of Service (QoS) requirements is one of the topical communication fields. Within the IST ARROWS[1] (Advanced Radio Resource Management On Wireless Services) project a real-time UMTS demonstrator is being developed to perform this analysis. The demonstrator platform is built over a network of personal computers and copes with the real-time restrictions imposed by the emulation of UMTS transport capabilities. This paper describes the global architecture of the ARROWS test-bed and provides some implementation-specific issues like real-time operation handling.

I. INTRODUCTION

One of the key objectives in the ARROWS [1] project is the development of a suitable software/hardware platform to test RRM strategies in a UMTS network. The envisaged platform has to be able to emulate advanced UMTS capabilities in a real-time framework. That means the possibility to test multimedia IP-based applications (videoconference, streaming services, WEB browsing, etc.) over an emulated UMTS network with enhanced RRM features.

The ARROWS demonstrator aims to provide a number of features in order to analyse and characterise the behaviour of a UMTS network that are not easily achievable by means of conceptual studies or system simulations [2]. Among such features we can emphasize the possibility to test end-to-end Quality-of-Service (QoS) performance and to assess, in real time, the effects that RRM algorithms have on the QoS perceived by the user. Moreover, the impact that RRM algorithms have on system performance could be demonstrated under different scenarios. Such scenarios include different traffic load conditions, propagation characteristics, user mobility patterns and several service configurations for the mobile users.

This paper describes the overall architecture of the ARROWS test-bed. The test-bed is currently being implemented in a network of Personal Computers (PCs). When designing the demonstrator architecture over such a platform, the emulation of the UMTS Bearer capabilities requires to cope with important time restrictions. These time constraints are mainly due to the 10 ms frame period defined in the UMTS air interface and, as the purpose of the test-bed is to demonstrate QoS performance, it is very important to have a complete determinism in all the processes running on the networked PCs to assure that process latencies, or even operating system latencies, could not mask or alter the results pursued by the demonstrator.

The paper is organised as follows: next section gives an overview on UMTS network architecture in order to, in section III, describe the relevant functionalities included in the ARROWS test-bed approach. Section IV deals with the software/hardware solution adopted to build the test-bed. On that basis, section V provides some implementation aspects considered within the test-bed to meet ARROWS goals. Finally, section VI concludes the paper.

II. UMTS NETWORK ARROWS APPROACH

Based on functionality aspects, the UMTS Network Architecture consists of three parts: UMTS Terrestrial Radio Access Network (UTRAN), UMTS Core Network (UMTS CN) and User Equipment (UE). The UTRAN part deals with radio-related issues while the UMTS CN is responsible for managing session and mobility information at the same time as switching and routing data calls. Figure 1 depicts the generic UMTS architecture[3].

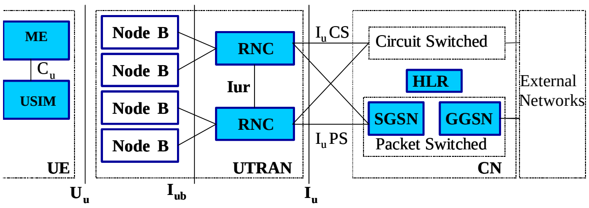

Figure 1. **Generic UMTS Architecture**

The UE connects to UTRAN wirelessly through U_u interface (air interface). Inside the access network,

---

[1] ARROWS project is partially funded by the European Commission under the IST framework (IST 2000-25133) and by the Spanish Science & Technology Ministry under grant TIC2000-2813-CE.

two distinct elements are specified: Node-B and Radio Network Controller (RNC). The function of the Node B is mainly to perform the air interface processing but also some basic RRM operation as the inner loop power control. However, RNC is where really radio resources are managed. RNCs and Node-B are linked by means of an standardised interface named Iub basically in charge of extending the transport channel capabilities provided by the radio physical layer up to the RNC. One RNC is responsible for the load and congestion control of its own cells, and also executes the admission control and code allocation for new radio links to be established in those cells.

RNC interfaces the CN through the Iu interface. Iu is an open interface that divides the system into radio-specific UTRAN and CN. In fact, the definition of the Iu interface discriminates between the connection of UTRAN towards a circuit switched network (Iu CS) and towards a packet switched infrastructure (Iu PS). Through these interfaces radio services provided by UTRAN (referred as Radio Access Bearers) are managed (setup, modification, clearing) from the correspondent node inside the CN.

Apart form RRC Signalling, RNCs perform Layer 2 (L2) processing of the data to/from the radio interface. Basic RRM operations, such as the mapping of RAB parameters into air interface transport channel parameters, the handover decision, and outer loop power control, are executed in the RNC.

On the core network side, the packet switched UMTS CN consists of two types of GPRS Support Nodes (GSNs): the Gateway GSN (GGSN) and the Serving GSN (SGSN). The GGSN provides the interface towards external packet data networks such as ones based on Internet Protocol (IP) while the SGSN is the node in charge of managing the correspondent UTRAN resources. Both nodes deal with session management (SM) and mobility management (MM) procedures inside UMTS CN.

An important concept used inside UMTS is the differentiation between Access Stratum (AS) and Non-Access Stratum (NAS) inside the UMTS system. Particularly the AS refers to the radio-dependent part of the system and comprises all the functionalities associated with the radio interface and RRM strategies. Instead, the NAS part deals with all the functionalities that are not directly dependent on the radio management, such as MM and SM.

As stated in previous section, ARROWS test-bed is focused on testing RRM strategies for packet services and the consequences derived from such management in the offered end-to-end QoS. To meet this goal, the demonstrator needs to cope mainly with functionalities around Uu and Iu PS interfaces. On the contrary, Cu, Iu CS and Iub are not implemented since they are not relevant to the pursued objectives. In the same way, Iub interface is not included since it does not impose any relevant restriction to RRM apart from providing a fixed transport capability among RNC and Node-B

The Iu PS interface differentiates between the Transport Network Layer and the Radio Network Layer (RNL)[4]. The former is conceived as an UMTS-independent transport platform between RNC and CN while the latter copes with UMTS specific signalling and user data. Particularly RANAP (Radio Access Network Application Part) is the control plane protocol within the RNL. From the view point of the implementation of the ARROWS demonstrator only the Radio Network Layer is relevant because all the important functionalities associated with the RRM are allocated in this plane.

Concerning to the U_u interface, the radio interface is layered into three protocol layers: Physical layer (L1), Data link layer (L2), Network layer (L3). Layer 3 contains the Radio Resource Control (RRC) that interfaces with layer 2 and provides UTRAN Services to the NAS layers located in the CN.

III. ARROWS TEST-BED DESCRIPTION

The approach taken in ARROWS demonstrator design depends on the possibility to implement and analyse RRM algorithms performance over a real-time UMTS user. The main elements and functions of an UMTS network included in the test-bed are described in this section and shown in Figure 2.

The test-bed comprises four main blocks: User Equipment (UE), UMTS Terrestrial Radio Access Network (UTRAN), Core Network (CN) and Application Server. Next each one is explained in detail.

A. User Equipment (UE)

The UE consists of six modules: the Client Application; the TCP/IP Traffic Control; the QoS management; the RSVP (Reservation Protocol) signalling module; the NAS driver and the RRC module.

The Client Application element refers to the QoS enabled application and copes with the four traffic classes identified in UMTS. Particularly, audio, videoconference, streaming, WEB browsing and e-mail applications could be tested within the demonstrator.

The QoS management of the reference user comprises the IP Bearer Service manager which deals with QoS aspects and is the responsible of translating QoS requirement from the application point of view into UMTS QoS parameters and IP QoS parameters. The UMTS QoS parameters in their turn are managed by the UMTS Bearer Service entity included in the NAS Driver.

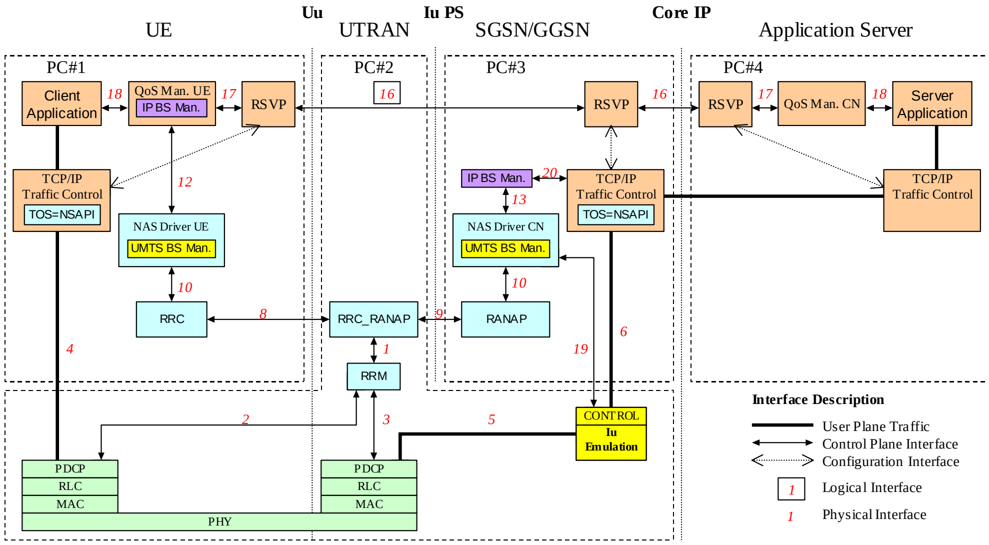

Figure 2. **UMTS functionalities included in the ARROWS test-bed.**

The TCP/IP Traffic Control manages QoS at IP level (queuing disciplines, scheduling policies, etc.) and, in the ARROWS framework, is also in charge of tagging the user data packets with the NSAPI (Network Service Access Point Identifier) which is put in the Type of Service (TOS) field inside the IPv4 header. This tagging allows to differentiate between IP packets with different QoS requirements and transfer them to the corresponding Radio Access Bearer (RAB) in UTRAN.

The NAS Driver module implements the session and mobility management needed in order to establish data sessions (PDP profiles) throughout UMTS. The NAS driver in the UE makes use of the RRC signalling module to manage radio resources across the air interface.

B. UMTS Terrestrial Radio Access Network

Three main elements can be distinguished in the UTRAN part of the ARROWS UMTS network: the radio resource control and management element; the lower layers and radio channel element; and the Iu interface emulation element.

As far as lower layers and radio channel are concerned, the complete protocol stack below network layer has been built. Thus, data link layer (L2) and physical layer (L1) have been implemented. Only three of four data link layer sublayers have been considered within ARROWS demonstrator: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC). The fourth sublayer Broadcast and Multicast Control (BMC) has been omitted since it is not involved in dedicated channels. The emulation of the physical layer and the radio channel is achieved by means of histograms of the bit error rate obtained from off-line link layer simulations also carried out within ARROWS project. These link layer simulations take into account the environment parameters such as cell type (macro/micro/pico), mobile speed, transport channel type, radio channel quality, etc.

Figure 2 depicts the complete protocol stack implemented for the reference user. To reduce the implementation complexity of the radio interface, the part corresponding to the lower layer stack implementation in the UE has been moved into the UTRAN. This decision does not impose any significant limitation to the ARROWS test-bed purposes since functionalities within these UE lower layers are addressed as if they were located at the UE side (i.e. distributed medium access control algorithms).

On the control plane there are two modules devoted to manage and implement radio resource algorithms. In fact, the approach followed in the ARROWS project has been to separate the functions of the Radio Resource Controller into the control itself, RR Control module, and the tools needed to perform it, RR Management Module. The former contains RRC and RANAP signalling protocols to communicate with the RRC module in the UE side and with the RANAP module in the CN side, and the latter implements RRM algorithms like admission and congestion control, outer loop power control, handover management and transmission parameters management (Transport Format allocation).

The effect of the rest of the users has been included inside the RRM Module (RRM) in the UTRAN. This emulation comprises traffic generation and radio propagation conditions for all the users in a given service area formed by several Node Bs. However, RRM strategies are actually executed inside the RRM for all the users in the system as if all users were real like the reference user.

Finally, apart from the radio interface lower layer implementation, the user plane in the Iu interface has also been emulated. Figure 2 shows where Iu emulation is done within the Test-bed.

C. Core Network (CN)

The CN must contain at least one Service GPRS Support Node (SGSN) and one Gateway GPRS Support Node (GGSN), thus in Figure 2 the CN part is called SGSN/GGSN. Within ARROWS framework the CN functionalities are performed throughout five main elements: the RSVP (Reservation Protocol) signalling module; the TCP/IP Traffic Control; the IP Bearer Service management; the NAS driver and the RANAP signalling module.

The user plane inside UMTS ends at the IP Bearer Service module in the CN who is responsible of connecting UMTS towards an external IP network.

The RSVP signalling module and the TCP/IP Traffic Control have the same functionalities detailed in the UE part.

In its turn the NAS driver is in charge of the Iu interface emulation control apart from session and mobility management, and makes use of the RANAP signalling module to manage UTRAN resources.

D. Server Application

Finally, an application server is made accessible from the UMTS CN through an IP intranet with QoS capabilities. Four elements make up the application server: the RSVP signalling module; the QoS Management; the TCP/IP Traffic Control; and the Application Server. Among the services to be tested in ARROWS we have considered the following possibilities: Mozilla as the HTTP client and Apache as the server for WEB; Mozilla as e-mail client and Qmail as server for e-mail; VIC for video, VAT or RAT for audio, and SDR for session management in the videoconference; and finally MPEG4IP for video-streaming.

IV. HARDWARE PLATFORM AND OPERATING SYSTEM ISSUES

To implement the hardware platform of the ARROWS test-bed a set of Personal Computers (PCs) with Linux operating system (Red Hat distribution) has been used to achieve a relatively low time resolution and high computational power. The Figure 3 shows a simplified structure of the demonstrator physical architecture. It is very important to assure that the platform on which the trial is performed will not mask or alter the measurements of algorithms. Then, it is basic to provide a virtually zero delay communications among machines and also enough processing power to cope with the huge number of operations to perform every few milliseconds. Communication among PCs is basically solved through fast dedicated Ethernet connections. If some considerations are taken into account when developing the source code, the high computational power available on current PC machines is adequate for the purposes of the project although partitioning of the most complex algorithms throughout several PCs becomes necessary. All the hardware performance do not avoid more or less random latencies introduced by the operating system, since it has not been specifically designed to meet such goals. Then, avoiding uncontrolled latencies is a challenge of the implementation.

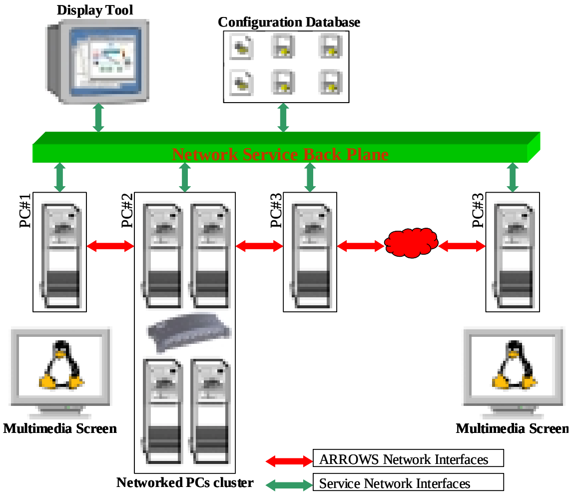

Figure 3 **Test-Bed architectural Description**

Moreover of the adequate performance, PCs have been selected because of other reasons. Maybe one of the most important is its flexibility. Also, PCs are common and programmers do not require special training to perform non-critical tasks. Another option would be working with high performance platforms devoted to critical operations such as multi-DSP platforms or costly hard real-time workstations. Of course in this test-bed some critical actions are carried out, but these are clearly localised on some concrete parts (RRM and Lower Layers) and can be masked, through appropriate actions, to algorithm developers.

PC's flexibility and performance involves also the operating system (O.S.) used to manage the hardware. It has been found that the freely available Linux [5] O.S. offers that expected flexibility for this project, although some aspects, such as process scheduling policy management, must be controlled or even modified.

The test-bed must deal with a time resolution of 10ms, the UTRAN frame duration. Every 10ms the Physical Layer must move data from one end of the radio interface to the other. Besides, RRM algorithms (including all the additional information required to emulate a set of radio cells with hundreds of users inside) must update transport information for the next frame to allow a right behaviour of the physical layer. If one of these parts fails into completing its tasks within the specified 10ms, the test-bed would not be operating in real time and the results from the point of view of the QoS capable applications could be disastrous.

V. OTHER IMPLEMENTATION ASPECTS

A. Abstraction Layer

Ensuring that all the processes running on the platform receive the resources they need at the required time is not obvious. It has been said that to emulate UMTS behaviour a 10ms resolution is acceptable. Some times one could think of the possibility of increasing platform processing power to solve some time misalignments. This solution may not be adequate because a bad usage of hardware resources would make useless the power increase. Moreover, in a test-bed like the one being presented there are not only real-time tasks but also other tasks devoted to monitor or to extract relevant information from the test-bed. Controlling the behaviour of these parts is crucial to avoid fatal real-time faults. Assuming that the best possible use of the hardware platform provides the capacity to operate in real-time, the problem is only to adequately use the resources provided by the hardware platform while preserving the advantages of PCs.

In the test-bed, moreover of processing lots of floating point numbers, one aspect to solve is the communication between processes, extracting information from them in order to debug the whole application and keeping them within the expected time and resource allocation bounds. All these issues have been solved through the introduction of an abstraction software layer placed between the Linux kernel and the test-bed software. This abstraction layer has been called Communications Manager. The basic idea behind this Communication Manager is to hide the operating system mechanisms (communication procedures, timing, etc.) to make the design of software modules as much independent as possible from operating system issues.

Following there are a list of functions implemented within the Communications Manager:

- Safe transfer of messages (packets) between modules throughout the test-bed.
- Timing control functions.
- Module task dispatching according to established timing policies.
- Access to configuration files.
- Generation of log files.
- Collect Statistics to allow their real-time visualisation.
- Test-bed control mechanisms such as start, stop, run, etc.

B. Signalling Procedures

UMTS upper layer signalling procedures within the ARROWS test-bed such as the RANAP protocol or the RRC protocol were designed using the International Telecommunication Union (ITU) Recommendation Z.100, Specification and Description Language (SDL). With SDL, we may specify different levels of abstraction, beginning from an overview and going on to very specific details. The entire test-bed is first implemented as one SDL system, by adopting this approach with no signals going to/from the environment the full functionality of the system can be simulated and debugged without the added complexity of a distributed system. Using this approach the entire system is analysed and simulated within SDL. In the next phase, the components representing the UE, UTRAN & CN are migrated to separate SDL systems. The targeting expert is then used to generate C code and environmental files. The generation chosen is Cmicro light integration for the Sun GNU gcc complier, this is chosen since Telelogic Tau does not directly support Linux. The generated C code is modified in order to interface with the Communications Manager and use its API functions. The entire code is then cross-compiled with the Communications Manager using the Linux gcc compiler.

VI. CONCLUSIONS

The architecture of the ARROWS test-bed has been shown. A complete UMTS stack of protocols has been constructed for a reference user. This user has the possibility to use QoS enabled applications over the UMTS emulated capabilities. The test-bed is focused to study the impact of RRM strategies on provided services from a qualitative and quantitative point of view. The effect of other users is included, through position and traffic statistical generation, even though radio resource algorithms are really executed for all of them.

Some implementation related aspects concerning the construction of a real-time test-bed for the ARROWS project have been outlined. Despite the high computational requirements and timing restrictions of the emulated system, a network of standard PCs with a Linux Operating System have been selected as hardware platform. An abstraction layer between the O.S. kernel and the test-bed software has been constructed to address real-time constraints and test-bed management.

VII. REFERENCES

[1] ARROWS Project WEB site, http://www.arrows-ist.upc.es/

[2] N.P.Magnani et al., “Overview of the ARROWS Project”, IST Summit 2001, Sitges-Barcelona, Spain, September 9-12.

[3] 3GPP Technical Specification 25.401, “UTRAN Overall Description”.

[4] Harri Holma and Antti Toskala, “WCDMA for UMTS”, John Wiley & Sons, 2000.

[5] Linux WEB site. http://www.linux.org/