An implementation of UTRAN for the WINEGLASS access network emulator

F. Badini[1], E. Scarrone[1], E. Kuehn[2], F. Casadevall[3], R. Agusti[3]

[1]CSELT Centro Studi E Laboratori Telecomunicazioni, Via Reiss Romoli 274, Turin, Italy
email: fabrizio.badini@cselt.it, enrico.scarrone@cselt.it
[2]Alcatel SEL, Lorenzstraße 10, Stuttgart, Germany
email: ekuehn@rcs.sel.de
[3]UPC Universitat Politècnica de Catalunya, c/ Jordi Girona 31, Barcelona, Spain
email: ferran@xaloc.upc.es, ramon@tsc.upc.es

Abstract*: 3G systems are designed for support of a brand new set of services (like location based services) thus, the capability to evaluate their provision in the laboratory is mandatory for their deployment. The IST WINEGLASS project aims to exploit IP-based techniques to support mobility and soft-guaranteed QoS, in a wireless Internet architecture incorporating a 3GPP-UMTS access network (UTRAN) and a WLAN access networks into the same "framework". The target is to explore the potential of this framework in enabling location- and QoS-aware application services for wireless mobile users. The UTRAN access network is an emulated W-CDMA FDD-based system. An emulation approach was chosen instead of a simulation approach, to reproduce the real behavior with more accuracy than what would be with simulation (more suited for global systems performance analysis). The WINEGLASS approach in developing the UTRAN emulator is to implement a subset of functionality appropriate for emulation of the critical aspects related to QoS and location awareness rather then to realize a "one by one" representation of UTRAN specs. The goal of test-bed is to demonstrate packet transport "end to end", both connection-less and connection-oriented. For this reason, only Iu-PS interface towards the core network is implemented. In this paper implementation of all protocol layers is described from a system point of view.*

Introduction

In future mobile communication systems, a set of brand-new services is envisaged in order to exploit the potential, in terms of bandwidth and flexibility, of modern mobile access networks. UMTS system not only supports voice but also high quality images and video together with non-real-time background traffic. Managing the problems derived from such a mixture of traffic classes revealing different bit rates, services and quality requirements, is mandatory. Optimized UTRAN protocol layer functionality ought to be investigated to allow an efficient support of the packet data (which is harder to manage with a guaranteed QoS). Another driving rule for third generation systems is provisioning of services with new “philosophy”, among these, location dependent services will be a major innovation with respect to 2G systems.

The IST WINEGLASS project aims to exploit IP-based techniques to support mobility and soft-guaranteed QoS, in a wireless Internet architecture incorporating UMTS (3GPP) and WLAN access networks in the same "framework". The main purpose is to explore the potential of this framework in enabling location- and QoS-aware application services for wireless mobile users. By its completion, the project will have developed a wireless Internet test-bed incorporating an IP-backbone connecting two kind of access networks: an UTRAN access, which is an emulated W-CDMA FDD-based access, and a WLAN access for intranet environment.

Such an integrated wireless Internet architecture allows mobile users to roam between the public (UMTS) and business (WLAN) contexts for access to their desired services in a heterogeneous network architecture, exploiting location-awareness by providing similar kind of services via above-mentioned access techniques. Regarding QoS, WINEGLASS approach is to realize “soft guaranteed QoS”, meaning that no special techniques (e.g. protocols) will be developed, instead, special approaches (regarding dimensioning and use of standard techniques) will be envisaged. Major part of investigation, will be focused onto core network structure and dimensioning. This is due to the fact that access networks, like UTRAN, are well standardized while core network actually not, allowing much more flexibility in terms of architectural reviewing.

In WINEGLASS, to assess UTRAN access, an emulation approach was chosen instead of a simulation one. The purpose of emulation is to reproduce the real behavior of the single "Service" with more accuracy than what would be with simulation (more suited for global systems performance analysis).
The philosophy used, in the process of development of WINEGLASS UTRAN emulator, is to implement a “subset” of functionality, appropriate for emulation of the critical aspects related to QoS and location awareness, rather then to realize a "one by one" representation of UTRAN specs. This approach leads to a lighter implementation, suitable to assess new services easily, not considering parameters that do not influence too much services developed. So, from an external point of view, the UTRAN emulator described in this paper appears as a "black box" that reacts to external stimulation (i.e. signaling messages) like the "real UTRAN". The goal of test-bed is to demonstrate "packet transport" end to end, both connection-less and connection-oriented. For this reason, only interface towards the core network packet domain (Iu-PS) is implemented. The W-CDMA FDD access technique was chosen for the implementation because the 3GPP (Release99) specs have been, at begin of project, more defined and more promising than TD-CDMA TDD ones.

The UTRAN emulator implements both User Plane and Control Plane as they are described in the reference model from 3GPP, as well as the concept of Access Stratum and Non Access Stratum procedures. The WINEGLASS UTRAN emulator is realized both in hardware and in software. It consists of some workstation plus a Real Time Emulator [1] implementing lower layers (below RLC layer included). The choice of splitting lower layers from upper layers was done because different timing requirement are necessary in order to take into account what a specific layer ought to do. Lower layers must provide not only a strict timing but it is also necessary to reproduce real "radio interface" conditions like interference, traffic load etc. Upper layers, instead, reproduce behavior of protocols needed to set up calls and to interwork with core network specific functionality, like call control and mobility management procedures.

UTRAN Protocol architecture

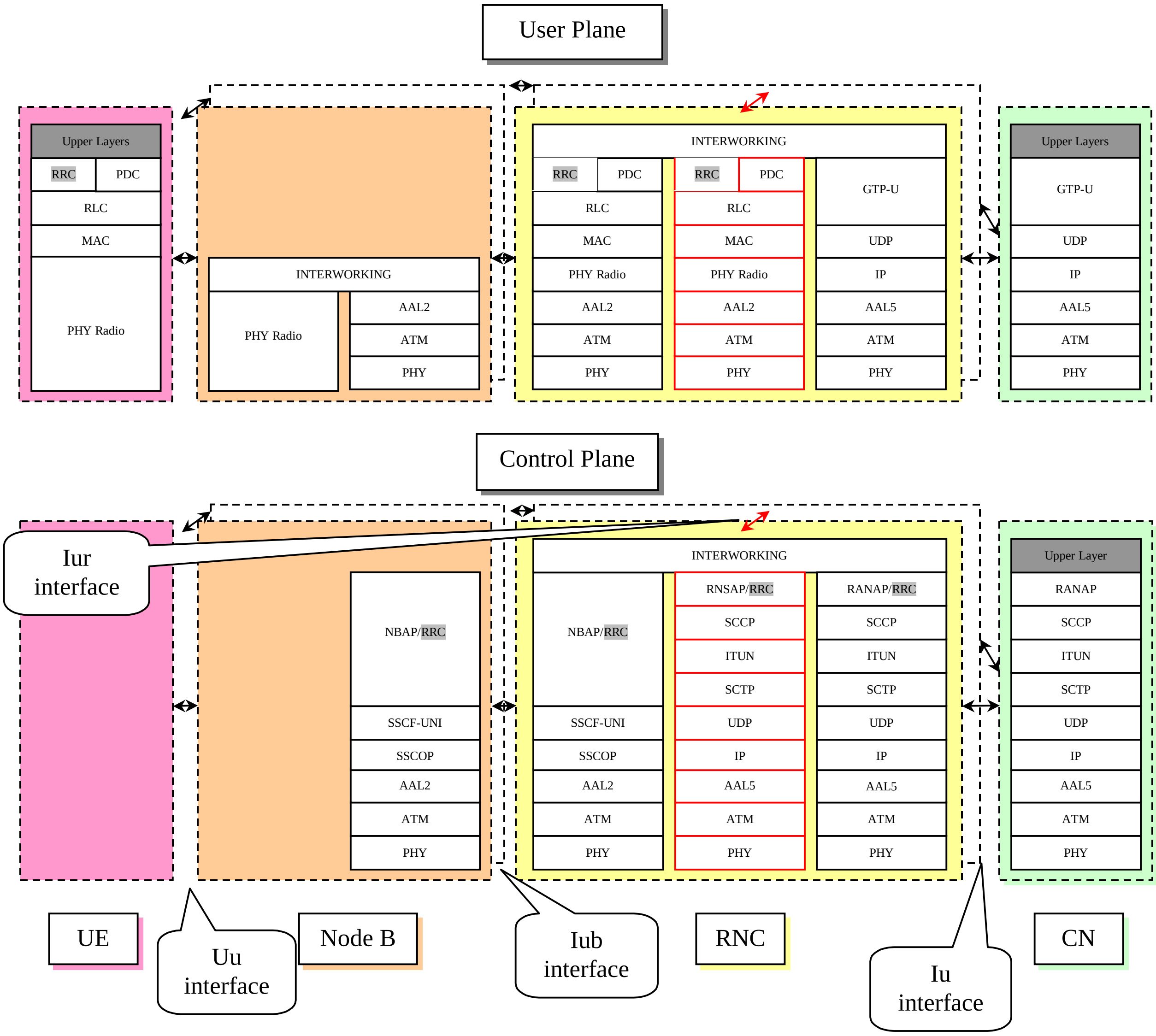
UTRAN specifications are structured in many control an transport layers, leading to a very complicated set of protocol stacks, some related to radio specific control, some to transport and so on. This is due to the approach chosen for describing the architecture.
In the common view of UTRAN specs there exists at least two domains from a protocol point of view:

- Transport on fixed interfaces inside UTRAN (Iu, Iub and Iur interface).
- Transport over Radio interface (Uu interface).

This approach is complicated by splitting, from philosophical point of view, of User Plane and control plane inside UTRAN. The first is used to transport user packets while the latter is used for control the user plane. Besides, on the fixed interfaces, there exists a transport of control plane + user plane above mentioned but, this transport has to be controlled too with an its own control plane.
This leads to the following structure of protocols:

- Protocols used to convey Control Plane informations on the fixed interfaces like RANAP, RNSAP, NBAP, SAPB and protocols used to convey Control Plane informations on the radio interfaces like RRC, BMC.
- Protocols to transport both User Plane and Control Plane packets over fixed interfaces like AAL2, ATM, GTP, UDP, IP.
- Protocols used to “control” the previous one for fixed interfaces, like Q.2630.1 for AAL2
- Protocols used for transport over Uu interface, both for User and Control planes, like PDCP, RLC, MAC and radio PHY (which is W-CDMA this case).

This crowded set of protocols can be resort and simplified to obtain a significant protocol stack to be



emulated. The simplified version is reported in Figure 1 (valid for Iu-PS only).

Figure 1 Simplified UTRAN protocol stack.

It is possible to see that two RNC and two Nodes B are reported to highlight Iur interface used for handover and SRNS relocation procedures. Both User plane and Control Plane are reported. RRC is highlighted in gray because is located in RNC and in UE but, some kind of interworking has to be performed with RANAP, RNSAP and NBAP in other network nodes. As a matter of fact, RLC, MAC and PHY radio layers are "spread" over RNC and Node B, depending of what radio channel is provided. RRC which control via SAP every radio protocols, must use e.g. NBAP protocol to convey lower layer parameter's, like power, synchronism to the involved Node B.

System Architectural description

The key issue regarding WINEGLASS UTRAN emulator is what is mandatory to implement and what is only "featuring" but not kernel functionality. Main approach of WINEGLASS team is to strip as much as possible "control" parts, which are useful in a "real" public network (to avoid, for example, exhausting of resources by one user), but would be redundant for an emulator which wants to evaluate applications (related to location awareness).

Radio lower layers (below RLC) are implemented via a Real Time Emulator which emulates the behavior of a single terminal among many others, introducing e.g. interference, then errors (see [1] for

details). The RTE is implemented both in hardware and in software and is connected to other parts of emulator via an Ethernet transport controlled by a workstation. These lower layers (PHY/MAC/RLC), are implemented in a significant subset of their functionality taking into account request from application demonstration.
At this level protocols implementation is not so evident because a finite state machine emulates channel behavior and many protocol aspects are merged together.
Around RTE, two workstations realize UTRAN protocols and related interworking towards NAS primitives (see Figure 2).

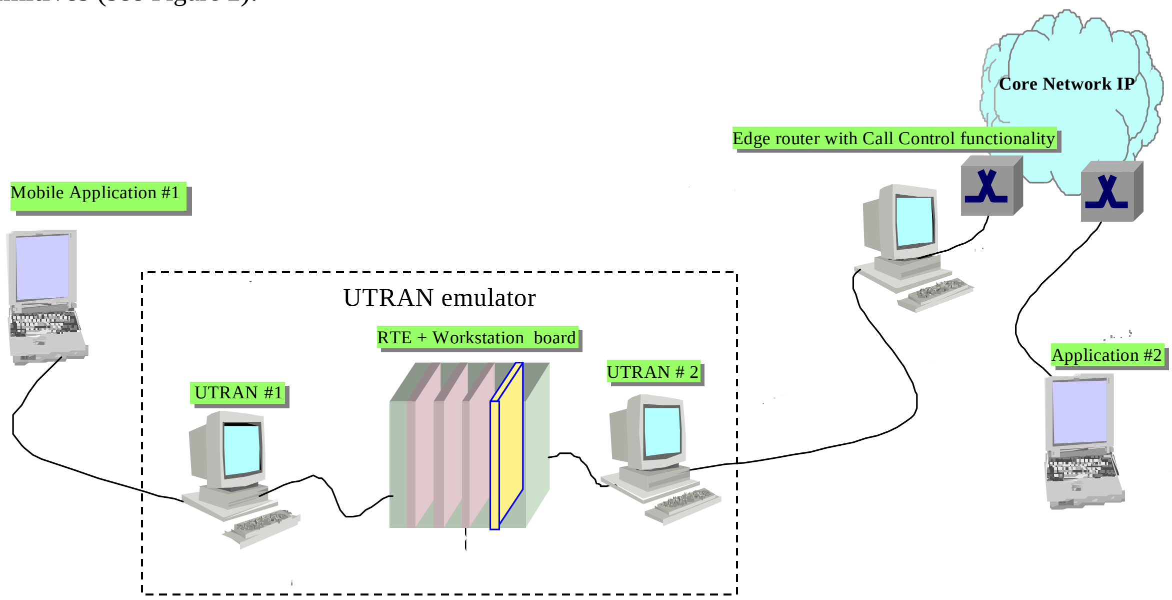


Figure 2 Overview of WINEGLASS architecture with UTRAN

From a protocol point of view only Iu interface is an open interface, while the other are merged all together inside the emulator.

Regarding the Control Plane in lower layers, it is important to notice that the transport of signaling has the same effects, whether or not they are carried via a real radio channel. Therefore, in WINEGLASS emulator, signals are tunneled from a "border" of RTE to other one, not crossing emulated radio channel because is not a WINEGLASS purpose to assess effectiveness of error rejection for control plane but only for user plane performance. We assume a reliable delivering of control messages.

Protocol emulation

The UTRAN emulator implements upper protocol layers in software, over a set of Linux/Intel workstation. A sketch of protocol interconnections is available in Figure 3. The main characteristics of each protocol are reported in the following subsections. In the picture is possible to see how Access Stratum-Non Access Stratum paradigm is realized using two software layer (labeled NAS interworking function daemon) which translates request primitives from, for example, Call Control entity into RANAP messages. In the figure, gray boxes represents software task for Control Plane, white for User Plane and dashed for software pieces which perform both Control and transport of User plane packets. Ellipses represent IP socket, which emulates SAP, both control and transport one. All transport is based on Ethernet/IP/UDP stack.

It is worth to mention that radio protocols below PDCP, which in the picture are labeled RLC/MAC/PHY, are emulated with support of statistical off-line emulation inside the previously mentioned RTE which provide normal IPv4 sockets. For details on implementation see [1].

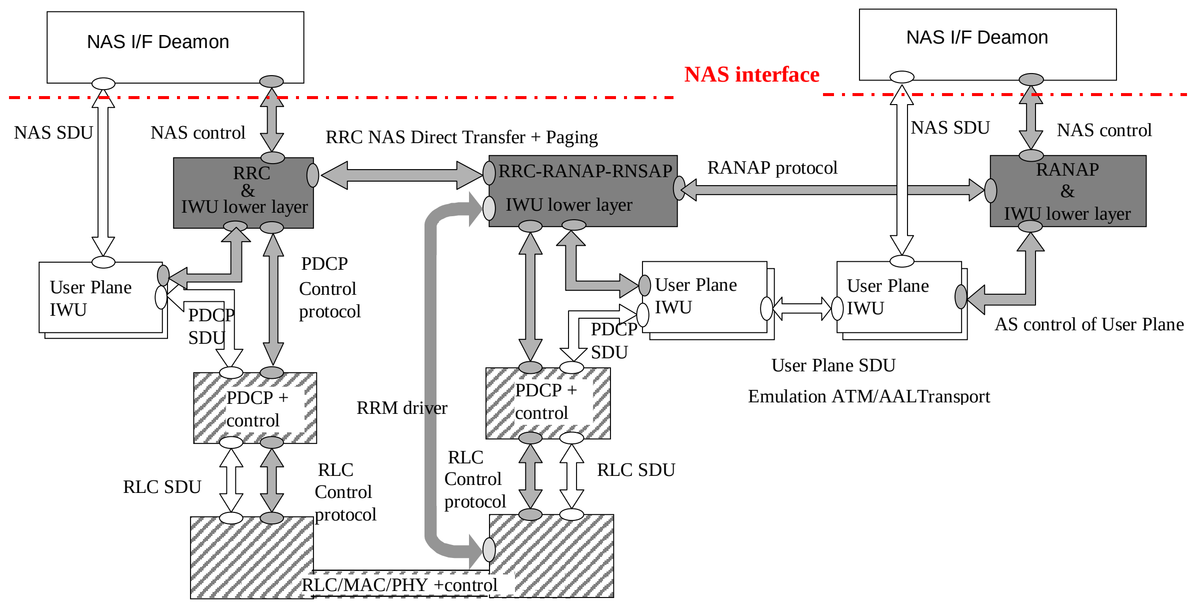


Figure 3 Protocol scheme for UTRAN emulator

Transport emulation (ATM, AAL, SCCP)

Transport inside UTRAN emulator is based upon Ethernet/IP/UDP protocol stack. This choice is based due to the properties of IP transport. IP is easier to configure than ATM transport, because of its connection less nature. Obviously, IP/UDP stack does not provide feature like a flexible control of quality of services, but this is not mandatory for WINEGLASS purposes regarding UTRAN, because WINEGLASS investigates QoS in core network and in a “soft” way (meaning with no ad hoc mechanism like RSVP). Besides, though UE perceives to stay in a “multi-user” environment (due to the scenario emulated by the RTE, which provides some BLER), over UTRAN interfaces (e.g. Iu) only one mobile is able to make a call. Taking into account that WINEGLASS scenario embodies only three type of services (one background, one interactive and one streaming) per call, the amount of traffic over the interfaces can be carried with no problems by a hundred megabit Ethernet interface. This avoids dimensioning issues, allowing WINEGLASS to concentrate on protocol investigations and using of that functionality (like location reporting).
Regarding AAL, WINEGLASS maps abstraction layer connections with UDP ports, avoiding emulation of Q.2630.1 because a set of preconfigured ports, registered in a file, allows connecting Core network Edge router with RNC emulator. In the
Figure 3, is possible to see User plane Interworking Unit (IWU) which performs mainly packet header processing mapping emulated terminal endpoint (GTP-TEI) to normal UDP ports.
At the end, an important aspect is emulation of SCCP signaling connections. WINEGLASS uses an “ad-hoc” field in the header of the signaling message container, which resembles the SCCP identifier of the connection (note that in UTRAN, SCCP is used in a connection-oriented way). This allows core network to distinguish among the two emulated RNCs, which are located on the same IP address. This is due to the fact that, for synchronization purposes, WINEGLASS has avoided to implement two distinct software tasks for the two RNC control plane subparts. Because this software, again to avoid synchronization issues, listens on a unique UDP port, the only way to distinguish among emulated RNC, is using the emulated SCCP id, which is realized inside the UDP packets.

Control Plane (RANAP, RNSAP, NBAP)

WINEGLASS chooses to split User Plane tasks from Control Plane to guarantee that “asynchronous events” (e.g. signaling) will not interfere with “streaming events” like User Plane ones. User Plane jobs are “light” (mainly packet header processing) but, at the same time, frequent to be carried out. At the same time, Control Plane jobs are harder to perform than User Plan ones, because they involve resource allocation and signaling transfer. WINEGLASS chooses multitask approach for User Plane tasks, because a process can't stop and fair OS scheduling provides equal chance for all calls to be served. Each task has its own "queue" where it receives data. Anyway, avoiding synchronization among calls is not possible because some procedures like RNSAP ones need it. A monotask approach is used for control plane task, with an internal state machine for every call, taking care of different status. There is only one queue where the control process "listen" and when a signal arrives to the queue, process wakes up performing actions related to a specific call (which has been previously "de-frosted"). After, the process freezes itself, waiting for next message on the queue (that can be arrived while process is still "running" on a specific call). RANAP procedures are realized in a meaningful subset, comprehending RAB assignment and release, Direct Transfer and Initial UE message. For a second phase, when location awareness is investigated, Location Reporting procedure is added. In **Figure 3** it is possible to see software blocks, where IWU (which performs User plane task) has multiple instances, while RANAP only one software task.

A "light" RNSAP inside Control Plane task is realized, mainly for coordination during SRNS relocation emulation. In UTRAN, RNSAP is mainly used for synchronization during handover, but WINEGLASS does not realize two "real" RNC with its own branch Node B, instead only protocol termination is implemented. RNSAP messages are exchanged via shared structured where the process , which emulates RNC, writes common data. NBAP is not emulated because does not exist, in UTRAN emulator, a software which emulates Node B (this functionality is merged inside RTE).

Radio Protocols (RRC, PDCP)

The radio protocols realized are RRC and PDCP, while RLC and below is implemented inside RTE [1]. RRC is not implemented in a single piece of software, but WINEGLASS chooses to share RRC duties among all sublayers (from interworking layer down to RTE) to avoid implementation of a single, complex, software block. Not all RRC procedures are implemented because it is out of WINEGLASS aims; mainly, connection management, radio bearer setup and location procedures are realized. Realization is carried out in a shared way among all layers involved; a procedures starts from a layer (e.g. RRC-RANAP interworking) with some parameters "dummy" and is enriched, with meaningful parameters, form layer below (e.g. PDCP sets its own useful parameter). This avoids multiple SAP inside RRC. PDCP layer is used for header compression and stripping. End to end QoS can be influenced from this process. Due to the relationship, of this layer, with end to end QoS, WINEGLASS implements this layer using the latest specification in IETF (like ROCCO) to perform efficient transport of IP packets over the radio interface (Uu). Simulations will compare robustness and efficiency of different header compression techniques with respect to the highly unreliable air interface, in order to choose the most reliable technique.

Conclusions

WINEGLASS UTRAN emulator, is designed to set an important step towards the assessment of models and protocols of 3G access networks in a Wireless Internet 4G context. The final target is to have a complete access "behavior" to evaluate the use and the enhancement of the UTRAN control and transport procedures for QoS and location aware services provision. Measurement, even if not the primary purpose, will be pursued, to identify the UTRAN parameters which are more sensitive for a better quality, and derive indications about their tuning.

References

[1] F. J. Casadevall, A. Gelonch, X. Revés, A. Umbert, R. Ferrús, "A Real Time Emulator of the UMTS lowers layers for the WINEGLASS Demonstrator", IST Mobile Summit 2000, Galway, October 2000.

[2] 3rd Generation Partnership Project - Technical Specification Group - Radio Access Network, "UTRAN Overall Description" TS 25.401.