PhD Thesis Proposal
Supporting QoS Delivery in the UTRAN

PhD student: Hugues VAN PETEGHEM*
PhD supervisor: Laurent SCHUMACHER†

Abstract

The evolution of mobile communications and the Internet has lead to the third generation of cellular networks, known as UMTS. In this evolution, new applications/services have been designed requiring QoS guarantees with respect to data transfer reliability, available bandwidth and delay/jitter bounds. The goal of this PhD is twofold. First we will build a testbed simulating the UTRAN of a UMTS network using a dozen of personal computers and open source softwares. The second step of our work will be to improve the best-effort network algorithms to adapt them to the wireless/narrow-band links (typically the air interface of the UTRAN), in order to fulfil the QoS requirements of upcoming applications and services.

Key words **- UMTS, 3G, QoS, UTRAN, traffic model, simulation, testbed.**

1 Introduction

The last few decades have seen two great revolutions in communication: Internet and mobile communication.

Primarily, Internet was used by scientists for exchanging information among themselves and for networking research up to beginning of '90s. Remote access, file transfer, and e-mails were then the most common applications. But the World Wide Web has fundamentally changed the Internet. Nowadays, Internet is much more than just e-mail and web browsing. It is really a complete data technology revolution which transforms our daily life with new services such as e-commerce, Internet telephony, videophony, instant messaging or video/audio streaming to name a few. Some of these new applications require a certain minimum throughput to operate, while others require a minimum latency. Jitter, which is the difference in latency between consecutive packets, has also to be carefully constrained for many time sensitive applications. Some applications do not tolerate dropped packets while others contain time-sensitive information whose delivery is better cancelled than delayed. With all these new flow characteristics, QoS has become the major concern of the Internet operators and their challenge is to provide an effective QoS while staying unperceivable from the user's side.

On the other hand, mobile communications have also evolved thanks to wireless standards evolution (802.11b/g, Bluetooth, GPRS, etc.) and hardware improvement (laptops, PDA's, mobile phones, etc.). But current mobile phones still mainly offer a single service: voice ; from this point of view, they are still closer to the PSTN than IP networks. However, with the launch of so-called third generation of mobile communication (3G), users are now able to truly access the Internet from mobile terminals, allowing mobile phones to deliver other services than voice. The convergence of mobile communication and Internet technologies is considered to be the next great communication evolution.

This proposal is organised as follows. Section 2 briefly presents the third generation of mobile communication on which we will base our work. Section 3 defines in detail the term "QoS" which we will use

*hvp@info.fundp.ac.be
†lsc@info.fundp.ac.be

in the rest of the document. Section 4 explains why QoS provisioning over a wireless link is a real challenge. Section 5 exposes the methodology to reach the thesis goal: the creation of new scheduling strategies, mainly located in the NodeB, to efficiently support QoS over an UMTS network. Section 6 depicts the project organisation for the next four years and finally, Section 7 concludes the paper.

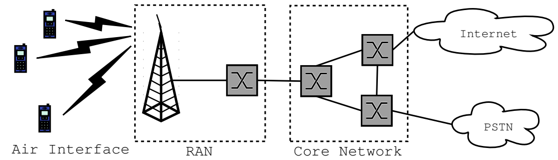

Figure 1: Third generation network overview.

2 UMTS and 3G

What exactly are 3G networks ? It is in fact the result of the mobile system evolution. Here is a quick run-down [1]:

- 1G is the first generation systems. They were analogue and offered only a voice service. 1G systems were very insecure against eavesdroppers, and offered no roaming possibilities.
- 2G heralded a digital voice and messaging service and offered encrypted transmissions. GSM has become the dominant 2G standard and roaming is now possible between overall 150 countries where GSM is deployed.
- 3G systems are being rolled out across the globe since 2003. They ultimately offer true broadband data: multimedia messaging, video on demand, videophones and high bandwidth games are available. Third generation systems differ from the second generation in terms of both the bandwidth and data capabilities that they provide.

3G systems can be divided very crudely into three (network) parts: the air interface, the RAN, and the Core Network (Figure 1). The air interface is the technology of the radio hop located between the terminal (the cellular phone, UE) and the UMTS Base Station (NodeB). The Core Network links the operator switches and routers together and extends to a gateway connecting to the wider Internet or PSTN. Finally, the RAN is the 'glue' that links the Core Network to the NodeB's and deals with most of the consequences of the terminal's mobility.

During its development, two flavors of the 3G have emerged. They are known as UMTS (developed and promoted by Europe and Japan) and cdma2000 (developed and promoted by North America). Both are tightly integrated systems that specify the entire architecture (from the air interface to the services offered). Although each has a different air interface and network design, they offer users broadly the same services of voice, video and fast Internet access.

As already mentioned, third generation wireless communication systems are providing wide-band multimedia services beside the conventional voice service. In order to support these various services, 3G networks bring a fundamental change since they are expected to be primarily packet-switched instead of circuit-switched. These new services need different QoS management. To support it over the UMTS packet-switched system, the 3GPP has established, in its releases 5 and 6, an architecture for the convergence of data, speech and mobile networks known as IMS [2, 3]. It is based on a wide range of protocols (SIP, RTP, GPRS, etc.) and combines them to allow real time services over the UMTS. Via the IMS, UE negotiates its capabilities and expresses its QoS requirements (media type, bit rate, bandwidth adaptation, etc.) during a SIP session. After negotiating the parameters at the application level, UE reserves suitable resources from the access network. Once end-to-end QoS is created, the UE encodes and packetizes its media data's and sends these media packets to the access and transport network by using an appropriate transport layer protocol (e.g., TCP or UDP) over IP.

The 3GPP has also fixed in [4, 5] four classes of services that need to be provided in UMTS. In terms of QoS, that means it will be possible to support several QoS classes linking them to specific applications (Table 1) or end-users profiles. We can easily imagine two different user profiles:

- Standard users: they pay for a basic use of their mobile terminal. That means they have access to all applications previously presented, but in case of bandwidth starvation in the cell, they accept a degradation or even a cut-off if needed.
- Premium users: they subscribe for a more ex-

Class Name	QoS Description	Typical application
Conversational	Delay/jitter high sensitivity Error medium sensitivity	VoIP
Streaming	Jitter high sensitivity Delay/error medium sensitivity	Video Streaming
Interactive	Error high sensitivity Delay/jitter low sensitivity	Web browsing
Background	Error high sensitivity No delay/jitter sensitivity	Email transfer

Table 1: UMTS QoS classes and applications.

pensive contract than the Standard users, but they are almost sure that their communications will not be degraded or cut-off at any time. That would however be the case if all the Standard users in the cell had been cut-off and the demand in bandwidth would still larger than 2 Mbps.

3 Quality of Service

3.1 What is QoS ?

The success of the Internet has brought us fresh new challenges. Many of the new applications have very different requirements from those for which the Internet was originally designed. One issue is *performance assurance*. The IPv4 implementation of the datagram model, on which the Internet is based, has few resource allocation capabilities inside the network and these have almost never been used. Therefore IPv4 does not provide any resource guarantee to users. However, most real-time applications, such as video conferencing, require some minimal level of resources to operate effectively. As the Internet becomes indispensable in our life and work, the lack of predictable performance is certainly an issue we have to address.

Another issue is *service differentiation*. Because the original Internet treats all packets identically, it can only offer a single level of service. The application, however, have diverse requirements. Interactive applications such as Internet telephony are very sensitive to latency and packet losses. In contrast, a file transfer can tolerate a fair amount of delay and losses without much degradation of perceived performance. But it is also to be said that *service differentiation* is not only application dependent, the user has also a role to play in this decision. Indeed users requirements can vary, depending on what he has paid for, influencing the requirements attached to their packets.

In a word, the capability to provide *resource assurance* and *service differentiation* in a network is nowadays often referred to as QoS [6].

3.2 Where enabling QoS ?

QoS can be defined on two different levels (Figure 2). The first one, the macroscopic level, is an end-to-end definition which can provide QoS from the initial transmitter to the final receiver. The other level, the microscopic one, is a QoS management inside the routers. This one can protect predefined flows from being delayed or dropped allowing their packets to pass rapidly through the router.

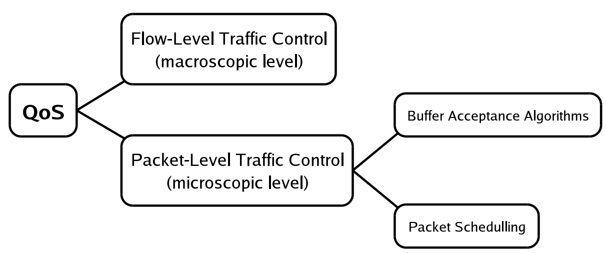

Figure 2: QoS levels.

On the former point of view, the flow-level traffic control, four technologies have emerged in the last few years as the core building blocks for enabling QoS in the Internet: IntServ/RSVP, DiffServ, MPLS and traffic engineering. Without them, it would be difficult to support QoS on a large scale and at reasonable cost.

On the packet-level traffic control point of view, many algorithms have been created to take into account the QoS during the packet routing to effectively protect sensitive traffic in congested situations. We can easily divide them in two categories. The first one gathers buffer acceptance algorithms which decide whether an incoming packet may enter the router or not (policing) or must be delayed (shaping). The most popular implementations are known as RED or Token Bucket (TB) [7, 8, 9]. The second category takes place at the egress side of the router. Its role is to decide which packet may leave the router at a given instant as all the packets can not use the egress link at the same time. Several implementations exist and one directly thinks about Processor Sharing (PS) which is the ideal, mathematical solution but cannot be implemented in real life because PS assumes that each active queue is served by the scheduler as if it contained a fluid flow. Besides this inapplicable algorithm, a largely used scheduling implementation is the Round Robin (RR) which serves the active queues one after the other. This algorithm tries to get close to the PS but in an implementable way [7].

By using one of these QoS algorithms, packets are sorted into a series of flows linking them to a priority level, a drop probability or a part of the available bandwidth.

3.3 Why enabling QoS ?

The Internet Protocol was originally designed as a mean to deliver QoS when it was released in September 1981 [10]. However, its features have not been extensively used until recently. New applications such as video conferencing, Web searching, on-line gaming, discussion boards or Internet telephony are now coming out and require some degree of resource assurance in order to operate effectively. As we enter the twenty-first century, the Internet is destined to become the ubiquitous global communication infrastructure. The need for QoS capabilities in the Internet stems from the fact that best-effort service and pure datagram routing do not meet the needs of these new applications.

To clarify the view of the different Internet applications, we can divide them in two great families (Figure 3):

- The "time critical" family includes applications imposing high delay and jitter requirements. But certain of these applications, such as streaming, can tolerate some errors whereas others are extremely sensitive to corrupted or lost packets.
- The "elastic" family gathers applications setting less stringent delay/jitter requirements. We can classify them according to their interactivity level which corresponds also to their delay sensitivity.

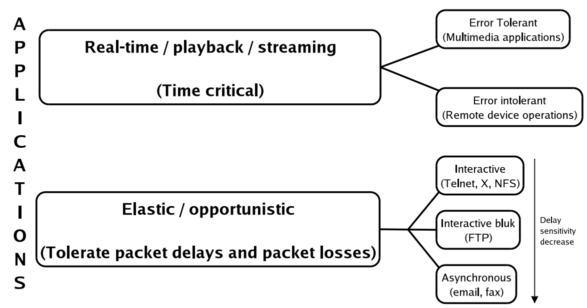

Figure 3: Applications families.

Each of the new applications now available over the Internet can be classified into one of these two families in terms of QoS requirements.

4 Difficulties to provide QoS in mobile systems

It has been years that researchers are working on QoS, however it has never been widely used over the Internet. Indeed, at the ISP point of view, it is easier and faster to over-provision resources, hence ensuring the bandwidth availability for all time-sensitive applications, rather than to introduce complicated QoS algorithms. But, in the narrow-band/wireless network, the over-provisioning is impossible making QoS provisioning more intricate than providing it in a wired network [11]. Three main issues can be tackled out in that respect. They are exposed in the next Subsections (4.1, 4.2 and 4.3):

4.1 Physical layer

The first major challenge to address in order to support QoS over a wireless network is the physical layer, e.g., the radio channel. Compared to a wired one, a wireless link has high noise and low bandwidth. In typical copper cable BER is usually between 10^{-6}

and 10^{-7} while in optical fibre between 10^{-12} and 10^{-14}. In wireless channel it can be as high as 10^{-1}. This involves transmission problems. A higher BER inevitably increases packet loss and naturally causes a reduction of the link throughput.

Wireless links are not fitted to effectively support QoS as we consider it over a wired network. An adaptation has to be made to take into account the high loss rate and the low bandwidth of such links.

4.2 Network layer

The network layer also needs additional functionality in mobile networks when compared to the PSTN or fixed computer networks. Mobility, known as the possibility of communicating in different locations and while 'on-the-move', is one of the key characteristics of wireless networks. Moving around within a few meters range is clearly not a big problem. More interesting is what happens if the terminals move over larger distances, when users drive around in their cars or even when they step out of an aircraft and switch on their terminal far from home, possibly on a different continent.

Two types of problems arise in this context. The first is tracking of terminals in dormant mode (alternative terms are idle and standby). This is usually referred to as the paging or locating network functionality. The paging process regularly "wakes up" a dormant mobile by sending it a special packet named a page. Once a dormant mobile receives this page, it moves into active mode and informs the network of its exact location (the NodeB of which it depends) before returning in dormant mode. The mobile must also inform the network when it moves into a new paging area (when it changes of serving NodeB). All these location informations are stored and maintained in central data bases called Location Registers (LR).

The other problem is that a moving, active terminal will face the risk that it will leave the area where its currently serving NodeB is capable to provide sufficient QoS. In order to get a continuous or even seamless service, the connection has to be handed over to another NodeB which is better suited to fulfil the QoS commitment. This research is currently in progress. The 'handover' or 'hand-off' can be divided into three phases or sub-problems tackled in [12]:

- Handover decision/detection - when and where to hand over ?
- Handover resource assignment - are there resources available in the receiving access area ?
- Handover execution - protocols for the reliable exchange of handover data.

Note that these phases can occur during an handover between two 3G access points and in this case the literature speaks about an horizontal handover, or between an UMTS and an other wireless technology access points, this is known as vertical handover. This last type of handover is out of the scope of the present thesis.

4.3 Transport layer

Finally, the third issue to keep in mind is located in the transport layer. Typical transport protocols, such as TCP, are designed on the assumption that they are running over a wired network and that errors are negligible. Over these low loss connections, any packet loss is assumed to be due to congestion and causes the TCP protocol to reduce the transmission rate [13, 14]. But wireless links have a very high and variable error rate, limited and variable bandwidth and a dynamic network topology compared to fixed connections. All these link characteristics make the transport protocols totally unadapted to wireless links and they perform very poorly when one is included into the path.

4.4 QoS over wireless

All in all, the extreme variability of mobile traffic, the need for handover support, the adaptability of TCP to wireless environment and the fact that the radio link is a bandwidth bottleneck means that mobile systems have to manage resources very carefully and it is very worthwhile to develop resource admission control and handover procedures.

Conscious of these problems, R&D teams have already elaborated some processes to solve them and to enable QoS over a wireless link. Research is mainly focused on TCP optimisation techniques to enhance the TCP performance for various wireless environments. The different protocols that have been proposed to improve TCP can be classified in two separate groups.

The first one is the "End-to-End Protocols". In this approach the TCP sender attempt to handle the losses in a way that improves the performance over regular TCP, using various end-to-end protocols. As example, TCP Reno [1, 11, 15] introduces the fast recovery phase forcing the transfer mechanism not to return to slow-start during the fast retransmit phase. This considerably improves the sender throughput. TCP New Reno [16] introduces the idea of "partial ack" to accelerate the packet loss recovery process of TCP Reno. We may also think about TCP Sack (Selective acknowledgement) [17, 18, 19] which uses TCP options to provide more detailed information about recently received packets to the sender. The TCP Fack (Forward acknowledgement) [20] can be seen as an extension of TCP Sack. This protocol uses the additional information provided by TCP Sack option to keep an explicit measure of the total number of bytes of data outstanding in the network. The Eifel detection algorithm [21] uses the TCP timestamps option to detect (a posteriori) whether it has entered loss recovery unnecessarily. This algorithm discards almost all useless retransmission. The TCP Vegas protocol brings light changes at the sender side. These modifications take place in the retransmission, the congestion avoidance and the slow-start mechanisms and tend to increase the TCP throughput while reducing retransmitted packets. In ECN (Explicit Congestion Notification) approach [22], intermediate hosts provide incipient congestion information. EBSN (Explicit Bad State Notification) schemes are similar to the ECN based approach in that the intermediate hosts swiftly forward the bad state information back to the sender so that the sender's timer is reset. The Snoop protocol[1] [23] introduces a module called the snoop agent at the NodeB. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments that have not yet been acknowledged. If a packet loss is detected the snoop agent retransmits the lost packet and suppresses the duplicate acknowledgements.

The second group of protocols is known as "Split-Connection Protocols". These are intended to isolate mobility and wireless related problems from the existing network protocols. This is done by splitting the TCP connection between the mobile host and the fixed host into two separate connections: a wired connection between the fixed host and the NodeB, and a wireless connection between the NodeB and the mobile host. In this way the wired connection does not require any change in existing software on the fixed hosts, and the wireless connection can use a mobile protocol designed to improve performance. Such split connection protocols include I-TCP (Indirect-TCP) [24] which utilises the resources of Mobility Support Routers (the RNC for the UMTS) to provide transport layer communication between mobile hosts and hosts on the fixed network. With I-TCP, the problems related to mobility and unreliability of the wireless link are handled entirely within the wireless link. The TCP protocol on the fixed hosts is not modified. M-TCP (Mobile-TCP) [25] is another split connection protocol. In the case of disconnections, the sender is forced into 'persist' state by receiving persist packets from M-TCP. While in persist state, the sender will not suffer from retransmit timeout, it will not exponentially back off its retransmission timer, and it will preserve the size of its congestion window. Hence, when the connection resumes following the reception of a notification from M-TCP, the sender will be able to transmit at previous speed. Finally, METP (Mobile-End Transport Protocol) [26] replaces TCP/IP protocol over the wireless link by a simpler one with smaller headers. By doing so, it shifts functions needed to communicate with an Internet host using TCP/IP from the mobile host to the NodeB, so that the distinct wireless link is hidden from the outside Internet.

Nevertheless, other research axes are going on to increase the bandwidth available over the UTRAN wireless interface. The HSDPA [27, 28] is the first major result of those studies. It improves system capacity and increases user data rates in downlink direction (from the RAN to the UE). This improved performance is based on:

- Adaptive modulation and coding: link adaptation in HSDPA is the ability to adapt the modulation scheme and coding according to the quality of the radio link.
- Fast scheduling: besides the PS and the RR scheduling algorithms (Section 3.2) which are essentially based on the QoS requested by the flow, the HSDPA scheduling algorithm is based on several other informations (the channel quality, terminal capability and power/code availability). This scheduling of transmission of data packets over the air interface is performed in the NodeB.
- Fast retransmissions: in current WCDMA, retransmission requests (due to errors) are pro-

[1] Some authors consider Snoop as a Split-Connection protocol.

cessed by the RNC. In HSDPA, these requests are processed in the NodeB, hence providing a faster response. In addition, incremental redundancy is also used. This technique selects correctly transmitted bits from both the original transmission and its retransmitted version to minimise the need for further repeat requests.

Thanks to these improvements, HSDPA offers maximum peak rates of up to 10 Mbps. However, more important than the peak rate is the packet data throughput capacity, which is improved significantly. Another important characteristic of HSDPA is the reduced variance in downlink transmission delay. A guaranteed short response time/delay time is important for many applications such as video streaming.

5 Methodology

Our work takes place in the third track of a project carried out by the *Pôle Réseaux et Sécurité* (PRS in the following) of University of Namur. This project is proposed in [29] and has the ambition of tackle the topics of interest by following three main, parallel tracks (Figure 4), enabling the PRS to use the results obtained following each track as a benchmark against which the team could check the validity of the results obtained through the two other tracks.

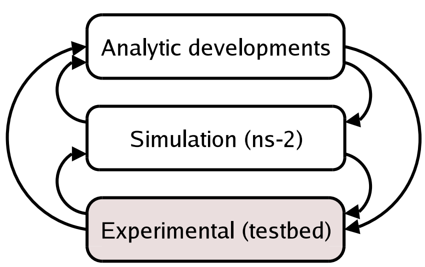

Figure 4: 3-tracks project overview.

1. The first track involves **analytical developments**. With the help of the tools of Network Calculus, statistical modelling and finite state machines, the PRS intend to model the behaviour of computer networks relying on wireless access networks and multiplexing several connections on a given wireless segment.

2. The second track will be of **simulation** type. The complexity of the communication networks has become so intricate that no one can truly claim to have apprehended it completely with analytical studies only. For instance, protocols designed to cope with congestion in wired networks are likely to react in an unexpected way when coping with phenomena triggered by wireless communications. In a move to assess the exhaustiveness of the analytical approach mentioned in the previous paragraph, computer simulations will be performed.

3. The third and last track involves the implementation of the schemes under study on a simplified Linux-based **testbed**. The present proposal entirely belongs to this final track. It will be designed so as to take into account the impact of UMTS access networks.

By comparing the results obtained through these three tracks, analytical studies, computer simulations and Linux-based testbed, the ambition of the PRS is to produce results whose validity will have been thoroughly assessed.

In this context, the thesis main goals will be to reproduce the real behaviours of the UMTS users and to establish an acceptable QoS level over the network (especially in the UTRAN).

To study these problems, and in order to test the new scheduling strategies over exactly the same network, we will build a testbed to mimic a real UTRAN. The purpose of creating an UTRAN testbed is to reproduce the user's behaviour (in terms of application usage, mobility, QoS requirements, traffic distribution, etc.) with less theoretical hypotheses and more accuracy, scalability and flexibility than what would be obtained by the other tracks of the project (ns-2 simulation and analytical developments), but with less costs and implementation complexity than a real system would demand. The testbed solution is a good compromise between costs efficiency and reality, that is the reason why we choose this research option.

But this will of course generate a set of issues which we will address one by one. We have identified five main stages presented in the following Sections (5.1, 5.2, 5.3, 5.4 and 5.5):

5.1 Experimental testbed preparation

The idea of using a testbed to validate the former simulation experiments is not a novel one. Some other projects are already using this strategy for

Testbeds	Main Research topics						
	QoS classes				Others		
	Conversational	Streaming	Interactive	Background	Vertical Handover	Horizontal Handover	Security
ARROWS		✓	✓			✓	
NOVEMBRE	✓						
NET INSTITUTE						✓	✓
MOBIQ	✓	✓					
WHYNET	✓				✓	✓	
Our testbed	✓	✓	✓	✓		✓	

Table 2: Testbeds comparison.

UMTS studies and are getting good results out of it. We will here after present some of the most well-known experiences before introducing our own testbed, its main characteristics and its innovation points.

5.1.1 Testbed state of the art

The first testbed emulating an UMTS network that we address is the IST project ARROWS [30, 31] which emulates a complete UMTS network comprising four main blocks: UE, UTRAN, CN and an application server. This testbed is focused on studying the impact of RRM strategies on delivered real-time services from the QoS point of view.

Besides, the NOVEMBRE testbed joins together several wireless studies. One of them is working on the UMTS and more particularly on VoIP traffic over the UTRAN. It tries to limit as much as possible the VoIP end-to-end delay [32] by using DiffServ or by multiplexing small VoIP packets in larger IP packets. This aggregation takes place in the NodeB taking care of the stringent delay and jitter requirements of VoIP traffic.

Other projects gather telecommunication partners who ensure them an access to their wireless infrastructure. In this case, the testbed is in fact a simple reduction of a real UMTS networks. The NET INSTITUTE [33] project has this great opportunity with T-Mobile and Nokia as partners. The main research axis of this project resides in security improvements over the air interface of the UMTS.

The MOBIQ project [34] focuses on the VoIP and audio/video streaming. Its key components are compression of RTP/UDP/IP headers and QoS control utilising feedback information from a network agent called “RTP monitoring agent” in the face of both network congestion and radio link errors. These components enhance quality of multimedia delivery services in 3G networks.

Finally, we can cite the large WHYNET project [35]. This one is an hybrid testbed which gathers many wireless standards and works on their interoperability. It is no use to say that the number one concern of this project lays in the vertical handover and the deployment of the Always Best Connected service (ABC, [36]).

5.1.2 Our testbed

Our testbed differs from the previously presented ones both in scope and approach. Our objective is to establish a suitable open-source software/hardware platform to deeply analyse and improve QoS. As shown on the Table 2, most of the presented testbeds are focused on one or two types of the typical UMTS traffic flows, defined by the 3GPP (Table 1) such as Streaming or Conversational. Instead, we will not constrain our research on a single service but will consider all typical UMTS applications together and work on an acceptable QoS level for all these services while running together. By running together, we consider for example a population of users within the same cell, some of them having standard conversations (Conversational) while others are browsing the Internet (Interactive) or retrieving their Emails (Background). Meanwhile, a few of them may arrive and initiate an audio streaming session (Streaming).

In order to emulate the UMTS network, we have installed a testbed which is composed of interconnected Personal Computers (PC’s) to create a little network isolated from the Internet (Figure 5). It is formed by three computer groups. The first one, the

Figure 5: The A group (left), the overall testbed (centre) and the B group (right).

A group, is made of Pentiums III 750 MHz with 512 MB of RAM. They all have four network interfaces connected to various sub-networks. The B group is composed of Celerons 300 MHz with 64 MB of RAM. Each one is connected to one of the computer of the A group as shown on figure 6. Finally, the C group, which is in fact composed of a single computer, has the same hardware specification of those from the A group and is directly connected to all the A group machines.

For the sake of homogeneity, we have installed the same operating system on each machine: a Linux RedHat based Fedora Core 1, kernel 2.4.22-1.2115.nptl.

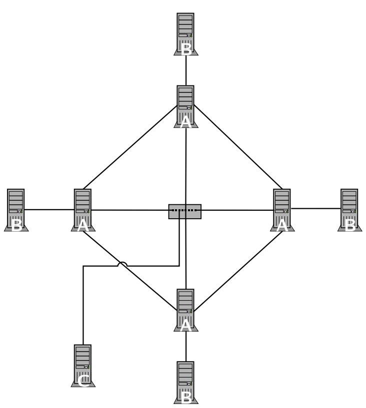

Figure 6: Laboratory testbed overview.

The A group is placed in the centre of the network because these computers are more powerful and then more able to deal with a large amount of traffic. Typically, the A group is used as routers using the routing freeware *Quagga* [37], whereas the computers of the B and the C groups are placed in periphery and used as traffic generators. In this way, we have installed the freeware *TG* [38] which is a traffic generator program able to create one-way UDP or TCP streams between a source and a sink. This traffic is described in terms of inter-arrival times and packet lengths using statistical distributions.

Using this testbed, the idea is to mimic the UTRAN. We plan to emulate one RNC managing four NodeB's with each of them supporting a population of mobile phones:

- A group, NodeB: Computers of the A group will act as NodeB's. Their role will be to take care of the traffic coming from the UE (resp. RNC) and to forward it to the RNC (resp. UE). Most QoS management functions will be performed in A group nodes since they are at the bottleneck of the UTRAN (connection between the wireless and the wired network). Indeed, in the early UMTS standardisation the NodeB's were seen as simple access points linking the UE's to the UTRAN and the CN. But, since the HSDPA introduction, the 3GPP gives them much more responsibility in term of QoS management and scheduling [27, 28].
- B group, UE: Each one of these computers will represent a population of UE attached to a single NodeB. Each of them will emulate several uplink

flows (from the UE to the RNC) to approach the reality. To create this kind of traffic, we mainly followed the specifications given in Appendices A and D of [39].

- C group, RNC: This computer will play the part of the RNC. It will receive the traffic from the four NodeB's and, in counter part, will generate the downlink traffic (from the RNC to the UE's). The specifications of this traffic have been derived from the recommendations given in Appendix A of [40].

The goal of this testbed is to approach as much as possible a real UTRAN (Figure 7), to generate a realistic traffic exchange and to deliver the requested QoS to the different services/users.

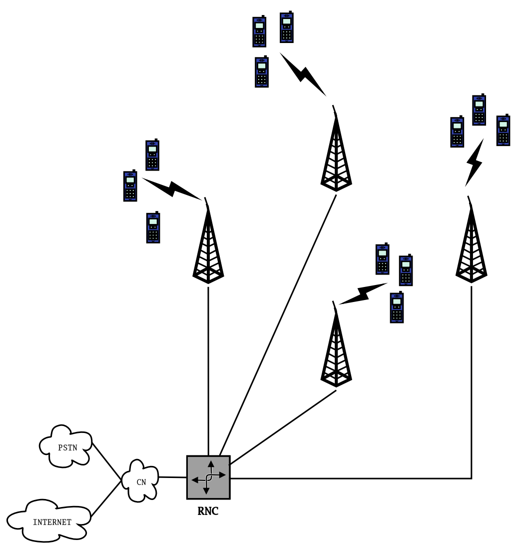

Figure 7: Real UTRAN.

5.2 UMTS traffic generation

We have to create an UMTS traffic as realistic as possible. In this way, we know the four QoS traffic classes for UMTS defined by the 3GPP standardisation [4]. In order to construct models for them, it might be useful to focus ourself on four representative applications: VoIP (Conversational), video streaming (Streaming), Web browsing (Interactive) and Emails (Background). But instead of using real applications to simulate user's behaviour, we decided to create traffic using statistical distributions and packets including fake/dummy data. We take this decision for the sake of simplicity. Actually, it's much more simple to configure some probabilistic scripts and launch them at the beginning of a simulation than creating as many as application sessions than users during the simulation.

But introducing one real application session during the simulation could be interesting as well. For example, including a real VoIP session while the simulation is running would give us a good hint of what a UMTS user would experience in such circumstances.

Here is a quick presentation of our traffic generation choices:

5.2.1 Conversational traffic

To generate a standard VoIP traffic, the first thing to decide is the used codec. As specified in [39], we choose the AMR vocoder with a rate of 12.2 kbps, which will generate 32 Bytes packets every 20 ms in case of activity.

The voice activity average of 32% is given by the following two-states *Makovian* model:

```
IF PrevState=0
  THEN IF RAND()<0.01
    THEN NewState=1; /* Go to voice activity state */
    ELSE NewState=0; /* Remain in voice inactivity state */
  ELSE IF RAND()<0.9875
    THEN NewState=1; /* Remain in voice activity state */
    ELSE NewState=0; /* Go to voice inactivity state */
```

5.2.2 Interactive traffic

The Web browsing traffic consists of a sequence of file downloads named "Packet Call", each of them modelled as a sequence of packet arrivals. All the parameters are listed in [40], but here is a quick summary of it:

- Number of packet per Packet Call: *Geometric* distribution with a mean of 25 packets.
- Packet inter-arrival time: *Exponential* distribution with a mean of 8.3 ms.
- Packet Call inter-arrival time : *Exponential* distribution with a mean of 12 ms.
- Packet size: *Pareto* distribution with the parameters $\alpha = 1.1$ and $\beta = 81.5$.

	IEEE 802.11b	IEEE 802.11g	UMTS
Frequency band [GHz]	2.4	2.4	2
Typical bit rate [Mbps]	11	54	2
Typical range [m]	100	100	>1000
Multiple access	CSMA/CA	CSMA/CA	WCDMA
Supporting Handover	No	No	Yes

Table 3: Differences between the 802.11b/g IEEE standards and the UMTS.

5.2.3 Streaming and Background traffic

For the moment, we are still looking for a suitable stochastic model to fit the Streaming or the Background traffic. We expect to identify it in the next few weeks.

5.3 Wireless link emulation

The implementation of the wireless segments of the testbed is one of the key issues of the proposal. Computing perfectly simulated UMTS traffic over a totally erroneous wireless link emulation would lead to useless results. On the other hand, one can not expect to get a (free) NodeB to introduce it into the testbed. So, the only way to go through this thesis will be to emulate the wireless link connecting the UE to the NodeB (the U_u interface). We may depict the UMTS Data Link layer as shown on Figure 8 [41]. To emulate this, we thought about two options.

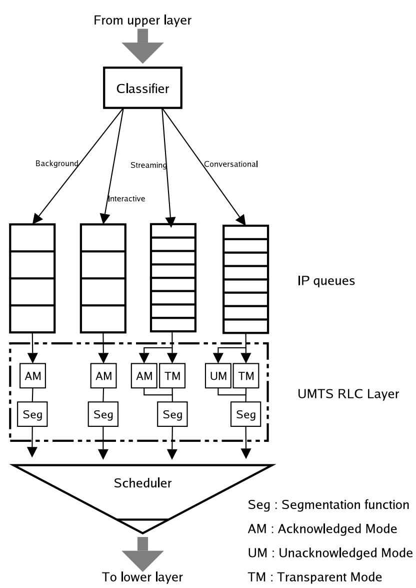

Figure 8: UMTS layer 2.

5.3.1 UMTS over IEEE 802.11b/g

The first option is to start from another standardised wireless AP such as the IEEE 802.11b (or g). We know that the UMTS and IEEE standards are not similar, especially in terms of frequency band and multiple access method (Table 3), but the ambition is to modify the AP to approach as much as possible a real U_u interface. To do so, many things have to be taken into account:

- Packet segmentation: we will implement a segmentation/reassembly module. On one hand, it will divide the data packets in smaller RLC PDU's [42] before sending them on the wireless link. On the other hand, this module will reconstruct a data packet using severals RLC PDU's.
- Data transfer services: the RLC sub-layers organisation includes three data transfer services [42]: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM). We will include a mechanism which will take care of the specifications of each transmission mode.
- Bandwidth limitation: we will decrease the available bandwidth to 2 Mbps (or 10 Mbps in downlink direction in the case of HSDPA emulation).

The advantage of this option is to benefit from a real wireless interface which will create typical wireless issues like bandwidth variations or high error rate.

Obviously, despite the fact that both IEEE 802.11 and UMTS rely on CDMA at physical layer, they proceed in very different ways. That does not jeopardise this emulation approach, however. Our vision of the IEEE 802.11-enabled testbed enables us to perform real radio transmissions, and therefore, to suffer from typical impairments like reflection or obstruction. But as far as the interference is concerned, CDMA is used in UMTS to solve the issue, while IEEE 802.11 aims at avoiding interference from other users by directly handling contention of the medium through CSMA/CA. We would therefore need to emulate the incidence of interference in our IEEE 802.11-enabled testbed in order to be able to claim that it truly mimics UMTS communications.

5.3.2 UMTS over Ethernet

The other option would be to emulate the air interface over a wired network. The idea is to create a "simulation box" which will act as the lower layers (Data Link and Physical) of the UMTS protocol stack. It will simulate the RLC segmentation of the IP packets, data transfer services and wireless issues using mathematical equations and probabilistic distribution. The good point of this option is the implementation assurance. Using the free operating system Linux, it will be certainly possible to code our "simulation box". But the disadvantage is the fact that we have emulate some other things compare to the first option. We can seen on the figure 9 that each option sets the border between emulation and implementation at different level.

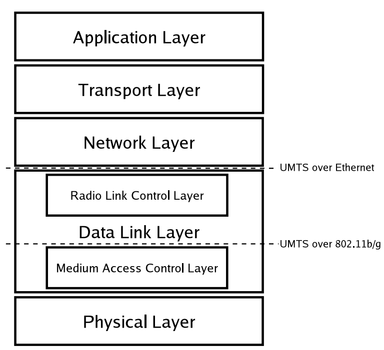

Figure 9: UMTS protocol stack and emulation limits.

These two options are attractive and are certainly not exclusive. The idea of starting a research in both way in parallel to constantly compare the results obtained seems to be a good one.

5.4 QoS management

The ambition of the thesis is to deliver an acceptable QoS to all services while running together. That means many users will use many applications at the same time. To support an equitable QoS between all these, we have to elaborate an algorithm to decide which user/application may or may not use the available bandwidth. This algorithm could be based on the type of user. Considering we have only two types of users: the Standard users which pay a minimum but accept communications/connections degradation/cut-off and the Premium users which pay a more expensive contract, but are sure that their communications/connections will almost never be degraded/cut-off.

That is the macroscopic QoS point of view. On an other hand, we have the microscopic QoS point of view which is the packet scheduling inside the NodeB. How to identify/classify/delay/drop all these packets ? To answer those questions, an other information has to be exploited by the algorithm. This information is the type of application or, more precisely, the QoS class the application belong. The four classes have totally different QoS requirements (Table 1) and have to be treated differently by the algorithm.

Having four QoS classes and only two types of users, we could identify all the packets (in terms of QoS class and type of user) using only height values (i.e. 3 bits, Table 4). Finding three unused bits in the IP header is not a big problem. The first field we have found in the IPv4 header [10] is the "Type Of Service" field which is not really used for the moment, except for some DiffServ application. But, as we won't use DiffServ in our testbed, the TOS field is a good candidate to support our new packets identification. The IPv6 header [43] contains also a 8-bits field which could fit to our utilisation known as "Traffic Class" field.

Bits value	Description
1 1 1	Conversational flow/Premium user
1 1 0	Streaming flow/Premium user
1 0 1	Interactive flow/Premium user
1 0 0	Background flow/Premium user
0 1 1	Conversational flow/Normal user
0 1 0	Streaming flow/Normal user
0 0 1	Interactive flow/Normal user
0 0 0	Background flow/Normal user

Table 4: Packets identification.

Relying on the QoS class for the scheduling and on the type of users to perform or not communication/connection degradation, the algorithm will be

	Year 1			Year 2			Year 3			Year 4		
State of the art and thesis delimitation	✓	✓										
Stage 1: Experimental testbed preparation	✓	✓	X									
Stage 2: UMTS traffic generation		✓	X	X								
Stage 3: Wireless link emulation				X	X	X	X					
Stage 4: QoS management				X	X	X	X					
Stage 5 (1/2): Control Plane simulation							X	X	X			
Stage 5 (2/2): Mobility management									X	X	X	
Thesis report and presentation											X	X

Table 5: Thesis working plan.

able to balance the flows in order to satisfy as many users as possible.

5.5 Control Plane simulation and mobility management

As noticed in [41], the UTRAN protocol model is based on the principle that the planes are logically independent. The Control Plane is used for all UMTS-specific control signalling. It includes the Application Protocol (RANAP for example) and the Signalling Bearers for transporting the Application Protocol messages. The User Plane transports all information sent and received by the users, such as the coded voice in a voice call or the packets in an Internet connection.

In the above description (Sections 5.1 to 5.4), we have only considered the User Plane. Indeed, we have made the assumption that the UE stay into the same cell, in other words, these UE always stand within the serving range of the same NodeB. In this context, we do not need handover management and Control Plane signalling. But, as mentioned before, the ambition of the thesis is to support a global acceptable QoS policy for near-real traffic, even when an handover occurs. The movement of UE across cells will require RRM and access control. Within the framework of this thesis, we will only take care of the Control Plane part directly connected to the handover management. So, topics like IMS or spreading factor computation will not be addressed in this work.

Enabling terminal mobility inside the testbed is once more an issue to control. The way the mobility will be implemented will largely depend on the wireless link emulation (Section 5.3). If we decide to simulate the UMTS the air interface above another wireless standard, we can imagine to create mobility simply by moving the hardware from a "A" node to an other. But if we choose to use the wired network option to simulate the U_u interface, the mobility problem will be solved by software by killing the running application on the old "B" node to transfer its context to the new serving "B" node. This will simulate a migrating user from one cell to an other.

6 Working plan

We divided the thesis in five main stages and we will approach them separately. As each one of these stages represents a lot of work, we have planned to finish this work within four years. A detailed calendar is shown in the Table 5.

7 Conclusion

In this proposal, we have presented the ambition of our thesis. In this way, we described our experimental testbed which will copy an UTRAN. Upon this infrastructure, we explain the real challenges of setting up a testbed, creating UMTS traffic (based on the 3GPP specifications), emulating the UMTS air interface (over an other wireless standard or a wired network), enabling QoS (depending on the application and/or the type of user) and managing UE mobility (with a Control Plane simulation).

Currently, the testbed is prepared and ready to support traffic generation. We are working on Streaming and Background traffic simulation using statistical distribution and we are already thinking on the better way to simulate the U_u interface.

A Terms and abbreviations

1/2/3G	$1^{st}/2^{nd}/3^{rd}$ Generation
3GPP	3^{rd} Generation Partnership Project
ABC	Always Best Connected
AMR	Adaptive Multi Rate
AP	(wireless) Access Point
ARQ	Automatic Repeat reQuest
BER	Bit Error Rate
CAC	Call Access Control
CN	Core Network
DiffServ	Differentiated Services
ECN	Explicit Congestion Notification
EBSN	Explicit Bad State Notification
FQ	Fair Queueing
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HSDPA	High Speed Downlink Packet Access
I-TCP	Indirect-TCP
IEEE	Institute of Electrical and Electronics Engineers
IMS	IP Multimedia Subsystem
IntServ	Integrated Services
IP	Internet Protocol
IST	Information Society Technologies
LR	Location Register
M-TCP	Mobile-TCP
MAC	Medium Access Control
METP	Mobile-End Transport Protocol
MPLS	Multi-Protocol Label Switching
MTU	Maximum Transmission Unit
NodeB	UMTS base station
PDU	Protocol Data Unit
PS	Processor sharing
PSTN	Public Switched Telephony Network
QoS	Quality of Service
RAN	Radio Access Network
RANAP	Radio Access Network Application Part
RED	Random Early Detection
RLC	Radio Link Control
RNC	Radio Network Controller
RR	Round Robin
RRM	Radio Resource Management
RSVP	ReSerVation Protocol
RTP	Real-Time Protocol
SIP	Session Initiation Protocol
TB	Token Bucket
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial RAN
VoIP	Voice over IP
WCDMA	Wideband Code Division Multiple Access

References

[1] Dave Wisely, Philip Eardley, and Louise Burness. *IP for 3G, Networking Technologies for Mobile Communications.* Wiley, 2002.

[2] Technical Specification Group Radio Access Network. *TS 22.340 V6.1.0, IP Multimedia System (IMS) messaging; Stage 1 (Release 6).* 3GPP.

[3] Miikka Poikselkä, Georg Mayer, Hisham Khartabil, and Aki Niemi. *The IMS, IP Multimedia Concepts and Services in Mobile Domain.* Wiley, 2004.

[4] Technical Specification Group Radio Access Network. *TS 23.107 V6.1.0, Quality of Service (QoS) concept and architecture.* 3GPP.

[5] Technical Specification Group Radio Access Network. *TS 23.907 V1.2.0, Quality of Service (QoS) concept.* 3GPP.

[6] Zheng Wang. *Internet QoS, Architectures and Mechanisms for Quality of Service.* Morgan Kaufmann, 2001.

[7] Kevin Dooley and Ian J. Brown. *Cisco Cookbook.* O'Reilly, 2003.

[8] J. Heinanen and R. Guerin. *RFC 2697: A Single Rate Three Color Marker.* IETF, September 1999.

[9] J. Heinanen and R. Guerin. *RFC 2698: A Two Rate Three Color Marker.* IETF, September 1999.

[10] University of Southern California Information Sciences Institute. *RFC 791: Internet Protocol.* IETF, September 1981.

[11] Antony Oodan, Keith Ward, Catherine Savolaine, Mahmoud Daneshmand, and Peter Hoath. *Telecommunication Quality of Service Management, from Legacy to Emerging Services.* IEE Telecommunications Series 48. The Institution of Electrical Engineers, 2003.

[12] Jens Zander and Seong-Lyun Kim. *Radio Ressource Management for Wireless Networks.* Artech House, 2001.

[13] James F. Kurose and Keith W. Ross. *Computer Networking - A Top-Down Approach Featuring the Internet, Second Edition.* Addison Wesley, 2003.

[14] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols.* Addison Wesley, 1994.

[15] W. Stevens. *RFC 2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*. IETF, January 1997.

[16] S. Floyd, T. Henderson, and A. Gurtov. *RFC 3782: The NewReno Modification to TCP's Fast Recovery Algorithm*. IETF, April 2004.

[17] E. Blanton, M. Allman, K. Fall, and L. Wang. *RFC 3517: A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP*. IETF, April 2003.

[18] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. *RFC 2883: An Extension to the Selective Acknowledgement (SACK) Option for TCP*. IETF, July 2000.

[19] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *RFC 2018: TCP Selective Acknowledgment Options*. IETF, October 1996.

[20] M. Mathis and J. Mahdavi. Forward acknowledgment: Refining TCP congestion control. *ACM SIGCOMM*, pages 281 – 291, 1996.

[21] R. Ludwig and M. Meyer. *RFC 3522: The Eifel Detection Algorithm for TCP*. IETF, April 2003.

[22] S. Floyd. Tcp and explicit congestion notification. *ACM Computer Communications Review*, 24(5):10 – 23, 1994.

[23] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4):469 – 481, 1995.

[24] A. Bakre and B.R. Badrinath. *I-TCP: Indirect TCP for Mobile Hosts*. 15^{th} International Conference on Distributed Computing Systems (ICDCS), May 1995.

[25] K. Brown and S. Singh. M-TCP: TCP for mobile cellular networks. *ACM Computer Communications Review*, 27, 1997.

[26] K.-Y. Wang and S. K. Tripathi. Mobile-end transport protocol: An alternative to TCP/IP over wireless links. *IEEE INFOCOM*, 3:1046, 1998.

[27] Technical Specification Group Radio Access Network. *TR 25.899 V6.0.0, High Speed Downlink Packet Access (HSDPA) enhancements (Release 6)*. 3GPP.

[28] Technical Specification Group Radio Access Network. *TS 25.308 V6.1.0, High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 6)*. 3GPP.

[29] Laurent Schumacher. *Appel à Projets, Fonds Spécial de Recherche (FSR) 2003-2004*. FUNDP, June 2003.

[30] Ramon Ferrús, Anna Umbert, Xavier Revés, Ferran Casadevall, Ronan Skehill, and Ian Rice. *ARROWS UMTS Test-bed Description*. IST Mobile Summit - Greece, July 2002.

[31] Anna Umbert Xavier Revés, Ramon Ferrús, and Ferran Casadevall. *Implementation of a Real-time UMTS Testbed in a PC Network*. 3^{rd} ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing - Spain (Madrid), June 2002.

[32] A. Samhat and T. Chahed. *Transport in IP-based UMTS Radio Access Network : Analytical Study and Empirical Validation*. ICC'2004 - France (Paris), June 2004.

[33] Net institute: Networks, electronic commerce, and telecommunications. `http://www.netinst.org/`, last visited: 08 July 2004.

[34] T. Yoshimura, T. Ohya, H. Matsuoka, and M. Etoh. *Design and Implementation of Mobile QoS Testbed MOBIQ for Multimedia Delivery Services*. 12^{th} International Packetvideo Workshop - USA (Pittsburgh PA), April 2002.

[35] Whynet: Scalable testbed for next generation mobile wireless networking technologies. `http://pcl.cs.ucla.edu/projects/whynet/`, last visited: 08 July 2004.

[36] Hakima Chaouchi and Guy Pujolle. *Policy based management framework for Always Best Connected users*. 1^{st} International ANWIRE Workshop - France (Paris), April 2003.

[37] Quagga routing software suite. `http://www.quagga.net`, last visited: 08 July 2004.

[38] Traffic generator tool. `http://www.postel.org/tg/tg.htm`, last visited: 08 July 2004.

[39] Technical Specification Group Radio Access Network. *TR 25.896 V6.0.0, Feasibility Study for Enhanced Uplink for UTRA FDD (Release 6)*. 3GPP.

[40] Technical Specification Group Radio Access Network. *TR 25.933 V5.4.0, IP transport in UTRAN (Release 5)*. 3GPP.

[41] Harri Holma and Antti Toskala. *WCDMA for UMTS, Radio Access For Third Generation Mobile Communication, Second Edition*. Wiley, 2002.

[42] Technical Specification Group Radio Access Network. *TS 25.322 V6.1.0, Radio Link Control (RLC) protocol specification (Release 6)*. 3GPP.

[43] S. Deering and R. Hinden. *RFC 2460: Internet Protocol, Version 6 (IPv6) Specification*. IETF, December 1998.