Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based Rel'99 UTRAN Testbed

Van Peteghem, Hugues; Schumacher, Laurent

Published in:
the 2nd Management Committee Meeting of COST2100 - Lisbon (Portugal), 26-28 February 2007

Publication date:
2007

Document Version
Early version, also known as pre-print

EUROPEAN COOPERATION
IN THE FIELD OF SCIENTIFIC
AND TECHNICAL RESEARCH

EURO-COST

COST 2100 TD(07)011
Lisbon, Portugal
2007/Febr/26-28

SOURCE: Pôle "Réseaux et Sécurité"
FUNDP - The University of Namur
Namur, Belgium

Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based *Rel'99* UTRAN Testbed

Van Peteghem H. and Schumacher L.
FUNDP - Institut d'Informatique
Pôle "Réseaux et Sécurité"
Rue Grandgagnage 21
5000 Namur
BELGIUM

Phone: +32 81 72 {4999,4980}
Fax: +32 81 72 4967
Email: {hvp,lsc}@info.fundp.ac.be

Investigation of the Sensitivity of UMTS Traffic Classes to Time-Correlated Errors on an IPv6, Linux-based *Rel'99* UTRAN Testbed

Van Peteghem H. and Schumacher L.

February 19, 2007

Abstract

Wireless networks encounter typically more errors than wired ones, and these errors are time-correlated. This paper presents how our UTRAN testbed emulates this behaviour based on a modified Gilbert-Elliot model taking into account the nature of the transport channel (shared/dedicated). The paper also presents the impact of this error characteristic on the four different traffic classes in a *Rel'99* UMTS network.

1 Introduction

The evolution of mobile communications and the Internet has lead to the third generation of cellular networks (3G), known as Universal Mobile Telecommunications System (UMTS). With the launch of this new generation of mobile communication, users are now truly able to access the Internet from mobile terminals, allowing them to enjoy a large variety of added value services.

In this context, UMTS Terrestrial Radio Access Network (UTRAN) modelling and traffic characterisation are important steps of the research. The goal of our project is to provide a representative Linux-based testbed which emulates a real UTRAN segment supporting typical IPv6 traffic samples based on significant statistical properties.

The rest of this paper is organised as follows. Section 2 exposes the motivations we had to launch this project. Section 3 presents the testbed architecture. Section 4 describes the supported traffic classes. Section 5 depicts our world representation, explains how we emulate mobility above a wired network infrastructure and exposes the testbed handover management. Section 6 shows the architecture designed to emulate the UTRAN. Section 7 presents our results about the emulation of the variable and time-correlated Bit Error Rate (BER) communications undergo over an UMTS radio link. Finally conclusions are drawn in Section 8.

2 Motivations

Time- and event-driven network simulators implemented in software are valuable tools for researchers to develop, test and diagnose network protocols, architecture, Quality of Service (QoS) management, etc. They can figure out a number of issues to be solved before even thinking about a real deployment. But network simulators also have their limitations. As the network to simulate becomes very large or very complex, they begin to be really difficult to implement and they require more and more computing time.

In this perspective, a testbed can be a complementary tool for researchers. Indeed, testbeds are cheaper than a real deployment while approaching even more the real network behaviour than a simulator would do in certain cases such as hardware limitation for example. They are also capable to reveal issues hard to notice using simulators. The testbeds are then seen as a good tradeoff between computer simulations and tests over a real life network.

Our aim is to design and exploit a modular Linux-based testbed mimicking a *Rel′*99 UTRAN. The modularity is conceived in a way that evolution of the testbed to *Rel′*5 High Speed Downlink Packet Access (HSDPA), further releases and Long Term Evolution (LTE) would be straightforward.

3 Testbed

As shown on the upper part of Fig. 1, we emulate a UTRAN segment consisting of one Radio Network Controller (RNC) managing four NodeBs each of them serving a population of User Equipments (UEs). The NodeBs are placed on a hexagonal grid but, for representation simplicity, the four cells are represented on a rectangle map.

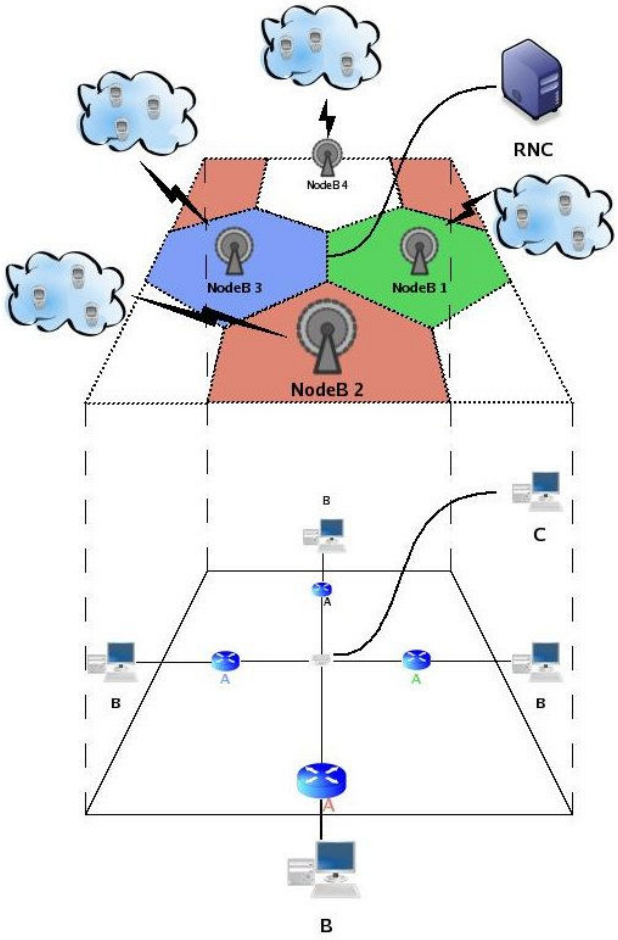


Figure 1: UTRAN testbed representation.

In order to emulate this UTRAN segment, we have built a testbed which is composed of nine Linux personal computers interconnected with 100 Mb/s Ethernet links, as shown on the lower part of Fig. 1. It is formed by three groups of computers, namely A, B and C.

The first one, the 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 128 MB of RAM. Each one is connected to one of the computers of the A group. Finally, the C group, which is in fact composed of a single computer, has the same hardware specification as the A group computers 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 Red-Hat based Fedora Core 6, kernel 2.6.18.

The A group is placed in the centre of the network. These computers are used as routers running the routing freeware *Quagga* [1], whereas the computers of the B and the C groups are placed in periphery and used as traffic sources/sinks. To serve that purpose, we have installed the open-source application *TG* [2] which is a traffic generator program able to create one-way UDP or TCP streams between two computers. This traffic is detailed in Section 4 in terms of inter-arrival times and packet lengths with the help of statistical distributions.

Using this testbed, the idea is to mimic the working of a real UTRAN segment:

- The A group computers act as the NodeBs. Their role is to take care of the traffic coming from the UE (resp. RNC) and to forward it to the RNC (resp. UE). Most of the QoS management functions will be performed at these spots since they are the bottlenecks of the UTRAN (border between the wireless and the wired networks). Actually, in the early UMTS standardisation the NodeBs were seen as simple access points linking the UEs to the UTRAN and the Core Network (CN). But, since the High Speed Downlink Packet Access (HSDPA) introduction, they have much more responsibilities in terms of QoS management and packet scheduling.
- Each computer of the B group represents a population of UEs attached to their serving NodeB (emulated on the attached A group computer). Each of them will generate several uplink flows (from the UE to the RNC). In order to simultaneously emulate several autonomous UEs, we implemented a control panel based on virtual network interface management. The role of this control panel is to facilitate the UE management by delegating them a single IP address.
- The C group computer plays the role of the RNC. It will be the sink of the traffic coming from the four NodeBs and will generate the downlink traffic (from the RNC to the UEs). As in a real UTRAN, the complexity is gathered in the RNC. It consequently controls the entire emulation.

4 Traffic Classes

In [3], the 3^{rd} Generation Partnership Project (3GPP) has defined four classes of services that need to be supported in UMTS. In order to characterise them, it might be useful to focus ourselves on four representative applications (Table 1).

Several studies have already been done about UMTS traffic modelling [4], but none of them characterised all four traffic classes simultaneously. Most of them consider only two classes, namely conversational and interactive. Although these traffic classes will certainly be the most popular, our aim is to be able to emulate any UMTS traffic mix.

To characterise the composite traffic of a single user, we fit each representative application in a three-level model shown on Fig. 2 [5]. Here is a quick run-down of their meaning:

Table 1: Traffic classes and applications.

3GPP Traffic Classes	Representative Applications
Conversational	VoIP
Interactive	Web browsing
Streaming	Video streaming
Background	E-mail

– Session Level: it lasts as long as the application is running. Its statistics are mainly influenced by user behaviour.

– Connection Level: it describes the connection behaviour of a single session. The pattern chosen for our sources is based on an On-Off (High-Low) model in which it could be possible to generate traffic on both states.

– Packet Level: it represents the packet inter-arrival and size distribution for each state of the Connection Level.

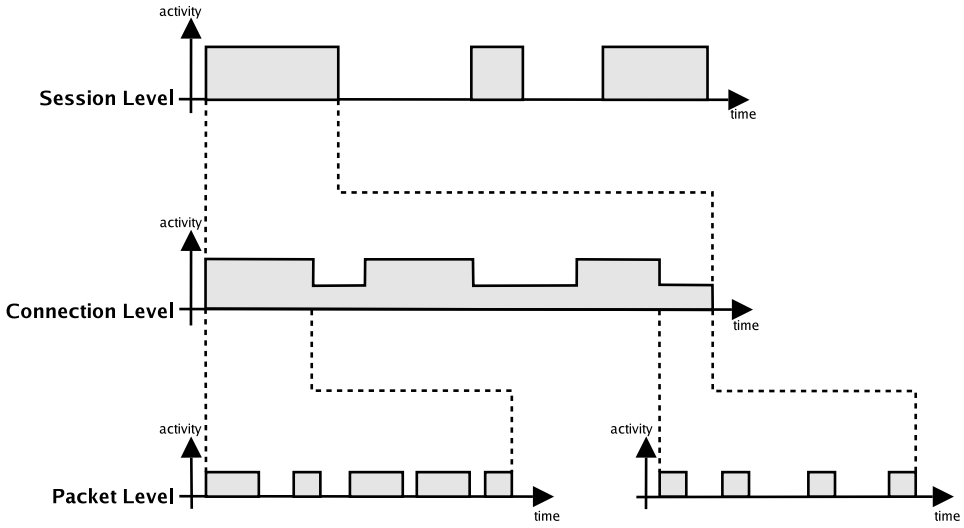


Figure 2: Three-level traffic model.

As an example to illustrate this three-level model, let us consider the conversational class. In this case, the session level describes the moments when a user is engaged in a phone call and the moments when s/he is not. When the user is engaged in a call, the connection level represents the time intervals during which the user is speaking and the time intervals during which s/he is simply listening. Finally, when the user is speaking, the packet level gives the size of each packet sent and the time interval separating two successive packets.

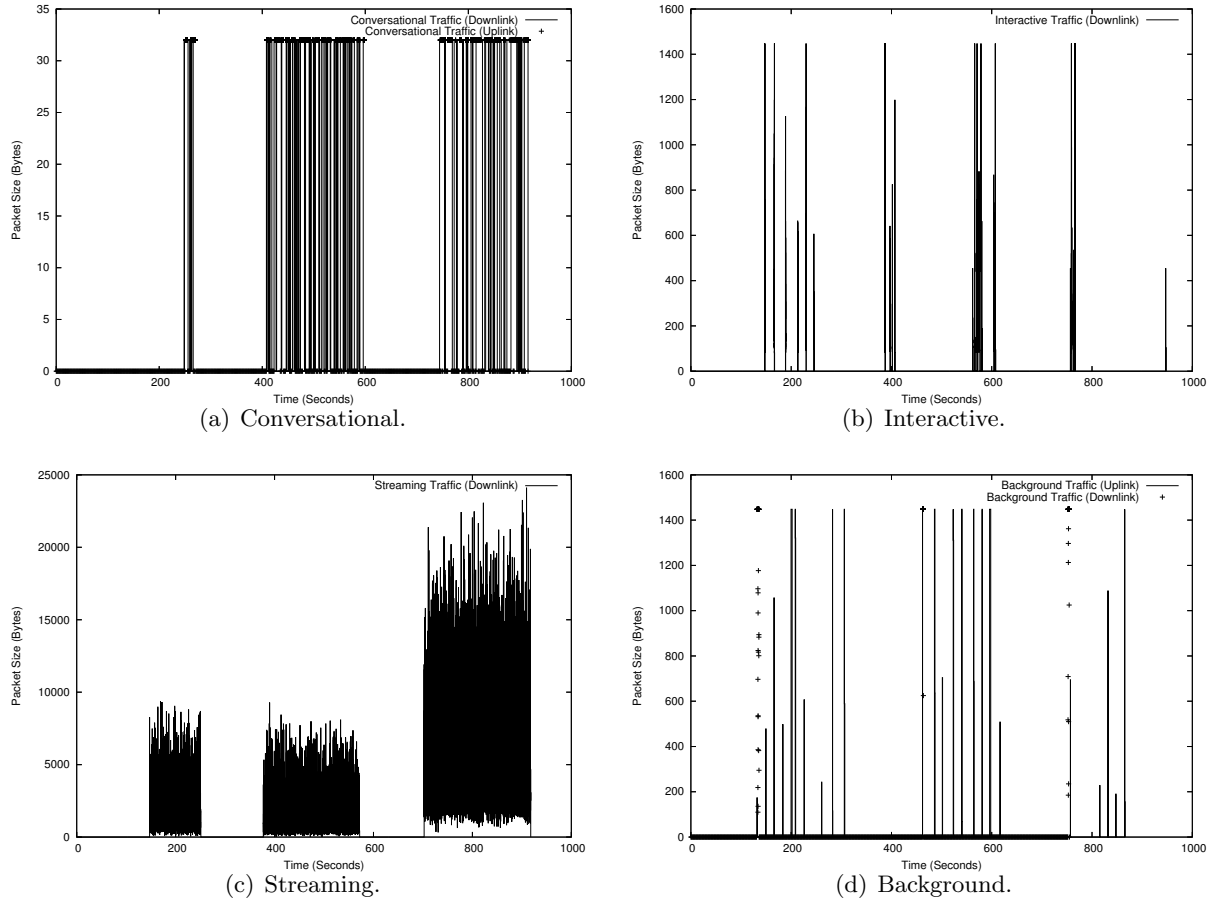
Based on this three-level division, the four traffic classes are described in terms of statistical distributions, in line with literature [6–9] and 3GPP standards [10–12] when applicable. These descriptions are summarised in Table 2:

Table 2: Traffic classes statistical distributions.

Classes	**Session-Level**	**Connection-Level**	**Packet-Level**
Conversational	Poisson	Two States Markovian	Constant
Interactive	Poisson	Geometric & Exponential	Pareto & Exponential
Streaming	Poisson	–	Pareto
Background	Poisson	Lognormal & Pareto	Lognormal

This probabilistic model of the traffic is used in the testbed to initiate sessions of any of the four traffic classes at any time in accordance to Poisson processes. It means for example that an emulated UE may very well initiate a conversational session while already having an ongoing interactive session. This would be translated in a real UTRAN by the fact that an UMTS user gets called as s/he was browsing the Internet on his/her 3G terminal.

As a matter of validation, we simulated a single UE connected to a NodeB, generating and receiving all kinds of traffic during $1,000$ seconds. Fig. 3 depicts the activity graphs of each of the four classes.



(a) Conversational. (b) Interactive.

(c) Streaming. (d) Background.

Figure 3: UE activity graphs.

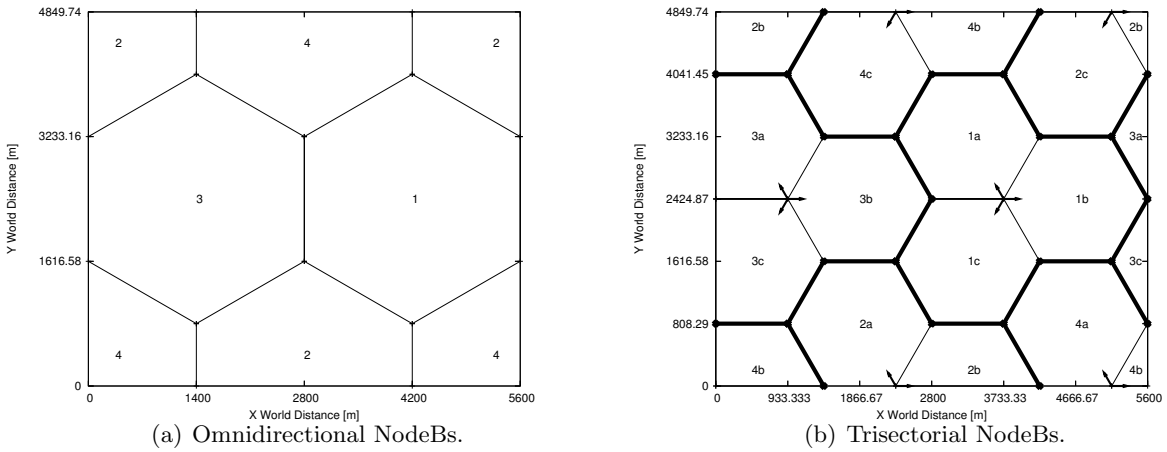
Note that the emulated traffic is simplex for applications emitting a significant amount of data, e.g. Streaming class, and that the Interactive return traffic, mainly constituted of TCP acknowledgements, is not emulated. Consequently, the return traffic is only emulated for the Conversational and Background classes, as one can see on Fig. 3(a) and 3(d).

There is a need of synchronisation for these two bidirectional traffic classes. Indeed, in a real conversation both of the participants are speaking throughout the session. Therefore, the emulated downlink and uplink traffics have to be synchronised and mixed. In the background case, we can think of a user downloading his/her mails and answering some of them one by one after their download. In this case, the emulation begins with a downlink traffic followed by multiple smaller uploads.

5 Mobility Management

5.1 Emulated World

As previously mentioned, we emulate a UTRAN segment consisting of four cells. We decided to place the NodeBs at 2,800 meters away from each other on an hexagonal grid which is the standard macrocellular set-up [13, 14]. This means that the cell radius equals 1,400 meters and the total surface of the four emulated cells is about 27 km^2. But in a real world the NodeBs are not omnidirectional, they are composed by three sectors emitting in different directions. This leads us to adapt our world representation. Fig. 4 depicts the world evolution from the omnidirectional to the trisectorial NodeBs. On Fig. 4(b), the arrows represent the sector's main emitting directions.



(a) Omnidirectional NodeBs. (b) Trisectorial NodeBs.

Figure 4: Emulated world representations.

Note that UEs are not constrained within this 27 km^2 area thanks to a wrap around technique. When a UE is about to leave the emulated world by crossing one edge, it reappears at the opposite edge (still within the same cell if necessary), as if it was somehow moving on a torus (Fig. 5).

Due to virtual network interface management limitations, the testbed supports up to 30 users by sector (up to 360 users in the entire emulated world). This number is fixed at the beginning of an emulation run and does not vary during it. We assume that the different UEs are switched on during the entire emulation. This means that we do not have any UE birth-death mechanism. However, their activity is controlled by the stochastic models of Section 4. As a result, during an

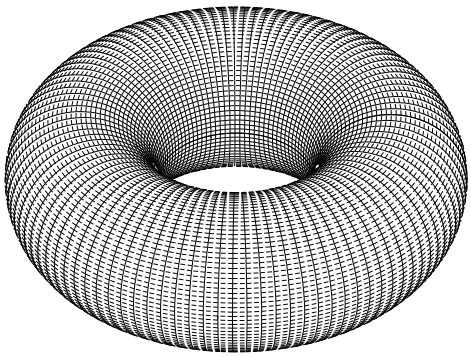


Figure 5: Torus world.

emulation with say 12 users, we may encounter periods of inactivity as well as full load periods when all 12 users are active.

This is illustrated on Fig. 6 which shows the aggregate downlink traffic of a given NodeB during an emulation of 1.200 seconds with 12 users. Up to three users are simultaneously streaming around the 600^{th} second whereas there was no traffic at all a few seconds before.

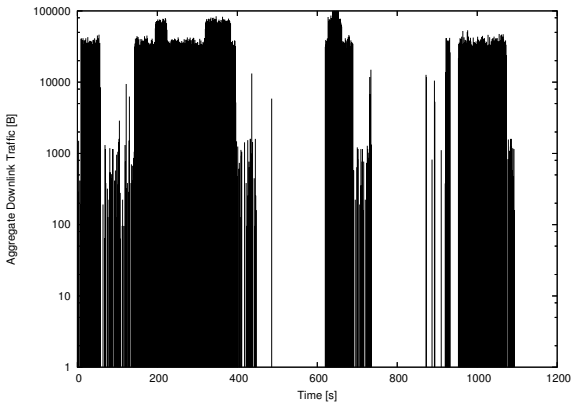


Figure 6: NodeB downlink traffic.

5.2 Random Walk

To enable terminal mobility inside the testbed, we decided to focus on four specific UE speeds: 3 km/h representing a pedestrian walk, 30 and 70 km/h which represents a UE on board of a vehicle inside a city and 120 km/h which finally represents a UE embarked on a train or on a car on the speedway.

To describe the UE mobility, we opted for the Gauss-Markov mobility model [15]: every second, the distance the UE will cover and the direction it will follow during the next second are drawn randomly. Keeping in mind that a vehicle moving at 120 km/h could not turn as abruptly as a pedestrian, the range of possible directions is then reduced with increasing UE speed. Fig. 7 illustrates the course of two different UEs respectively moving at 120 and 30 km/h during 600 seconds.

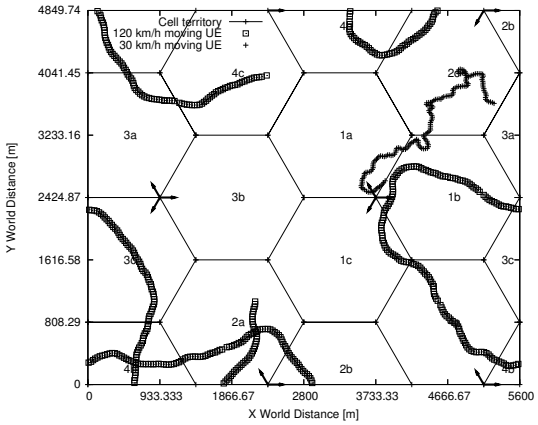


Figure 7: Random walk at 2 different speeds.

5.3 Handovers

Now that our UEs are able to move, we have to establish a handover procedure which will decide where and when a handover has to be performed. This procedure is based on the Signal to Noise plus Interference Ratio (SNIR):

$$SNIR = \frac{Received\ Power}{Interferences + Noise} \tag{1}$$

The computation of the received power is based on the signal power received on a specific UMTS physical channel known as Common PIlot CHannel (CPICH). Our hypotheses concerning the CPICH as well as all the other testbed cellular parameters are summarised in Table 3.

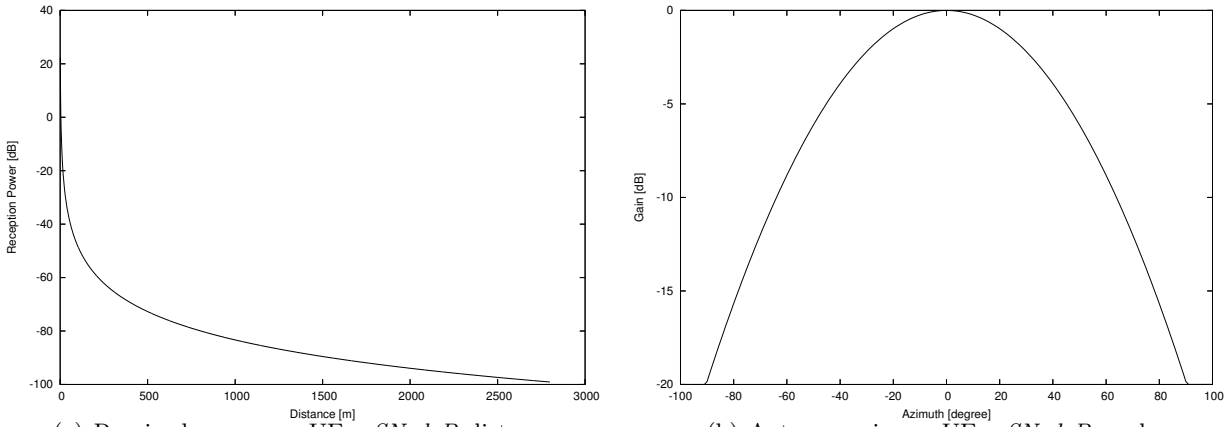
Table 3: Testbed cellular parameters.

Parameter	**Setting**
Transmission power [dBm]	43
Sectorial gain [dBi]	14
UE Noise figure [dB]	9
Thermal noise density [dBm/Hz]	-174
Emitting frequency [GHz]	2.14
NodeBs height [m]	30
UEs height [m]	1.5
CPICH spreading factor length [chips]	256
CPICH power [% of the total power]	10
SNIR handover threshold [dB]	4

The reception level is then emulated by a calculation mainly influenced by the distance separating the UE and its serving NodeB (*SNodeB*, the NodeB the UE is connected to), and by the angle created between the UE position and the *SNodeB*'s main emitting direction:

– The received power decreases as the distance between the UE and its *SNodeB* increases. This relation is known as path-loss and it is not a linear one (Fig. 8(a)). To model this, we use the Okumura-Hata wireless propagation equations and their UMTS adaptation [16].

– As shown on Fig. 8(b), if a UE is connected to a trisectorial NodeB, its received power will decrease as it will move away from the sector main lobe [17].



(a) Received power vs. UE – *SNodeB* distance. (b) Antenna gain vs. UE – *SNodeB* angle.

Figure 8: UE reception influencing factors.

Another concept, embedded in the WCDMA, influencing the UE reception level is the signal spreading [18]. It is able to amplify a signal and to attenuate the interferences. The higher the spreading factor is, the more amplified the signal will be (resp. the more attenuated the interferences will be) but the slower the sending of the data on the wireless interface will be. The spreading factor's choice is then a real challenge the RNC has to overcome.

In the testbed, the interferences are simply considered as the sum of the received power from the NodeBs but the UE *SNodeB*, multiplied by two. Indeed, we only have four NodeBs in our emulated world, but to compute a representative interference figure we have to take into account all the NodeBs lying around the *SNodeB*. In our hexagonal grid world representation, this means we have to consider six instead of three surrounding NodeBs.

As the shadow fading is not taken into account and the noise level is assumed to be constant during an entire emulation, (1) finally becomes:

$$SNIR = \frac{SNodeB\ Rec.\ Pow.\ *\ SF_{CPICH}}{2 * (\sum Other\ NodeBs\ Rec.\ Pow.) + Noise} \tag{2}$$

Derived from (2), Fig. 9 represents the SNIR level map of our emulated world.

A handover occurs when the difference between the power of the *SNodeB*'s signal and the power of another NodeB's signal is greater or equal to the handover threshold (Table 3). Based on this continuous SNIR evaluation the RNC knows exactly when to handoff a UE.

Fig. 10 illustrates the results from the working of the testbed. It represents the journey of one UE during 450 seconds generating and receiving conversational traffic only. Thanks to the mobility graph of Fig. 10(a), we figure out that the UE undergoes a handover passing from the NodeB #1 (sector a) to the NodeB #3 (sector b) at about the middle of the emulation.

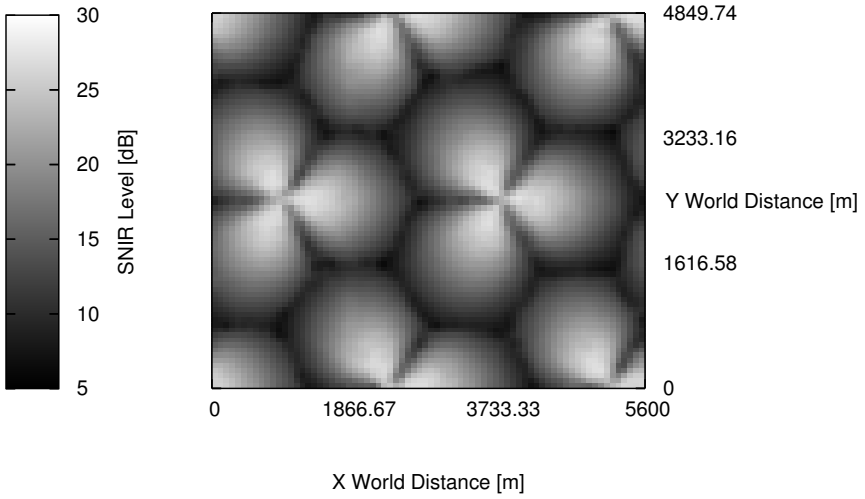
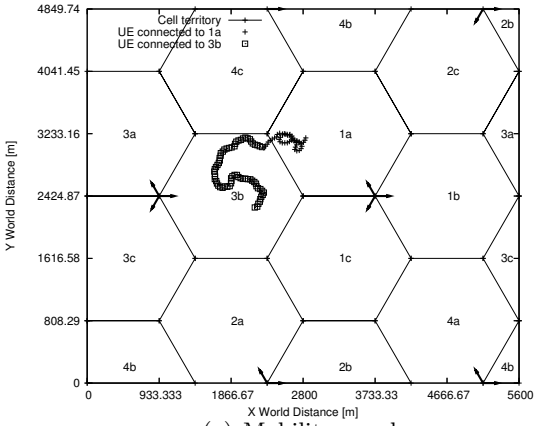
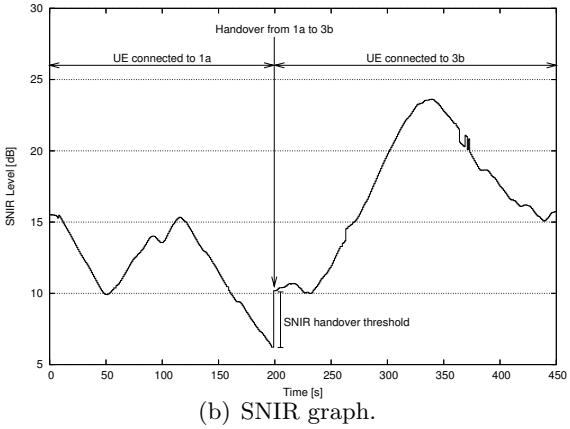


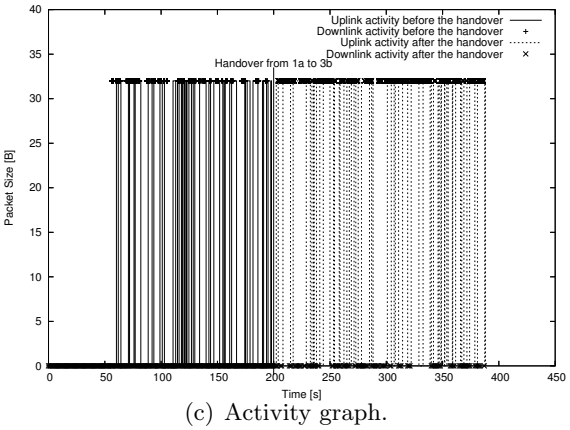
Figure 9: SNIR map.



(a) Mobility graph.



(b) SNIR graph.



(c) Activity graph.

Figure 10: Handover impacts.

As we look at the SNIR evolution of the UE (Fig. 10(b)), we observe that the handover exactly occurs when the power of the signal received by the NodeB #3 (sector b) exceeds by 4 dB the signal received by the UE's $SNodeB$ power. It is indeed what we expected regarding the handover SNIR threshold value (Table 3).

Finally, if we look at the activity graph of Fig. 10(c), we discover that the UE was active (sending/receiving data) when the handover occurred. We can clearly distinguish the end of the first connection and the beginning of the second connection, both totally synchronised in uplink and downlink directions. Note that when the handover has occurred, the service has been stopped during a short duration representing the transport channel allocation time.

From the testbed point of view, this handover is translated by a process migration from a computer to another (Fig. 11): first of all, the traffic emitting processes have to be stopped on the source computers (the C computer and the eastern computer of the B group), then new processes corresponding to the traffic generation after the handover can be started on the C computer and new destination computer (the southern computer of the B group). As a result, the packets belonging to the same activity session do not use the same physical route after this handover. Consequently, we can say that our UMTS testbed supports UE mobility above a wired infrastructure.

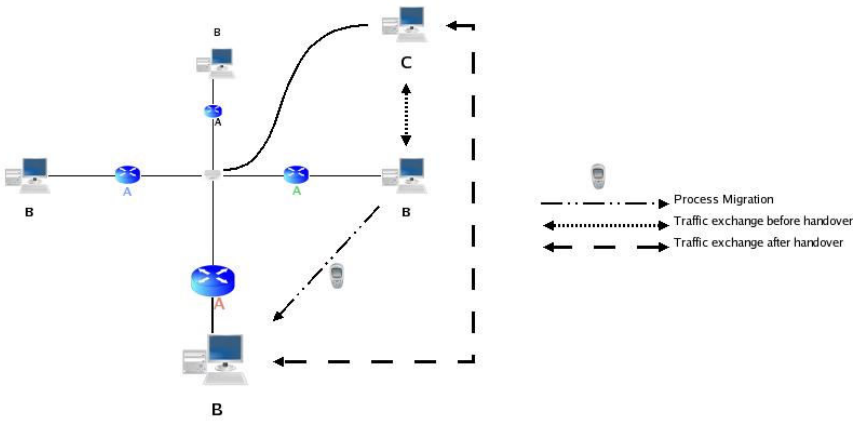


Figure 11: Handover testbed's point of view.

6 Air Interface

6.1 Transport Channels

As described in Section 4, traffic whose distribution complies with the characteristics of the 4 traffic classes is generated from the RNC to the PC emulating a given UE (downlink) using a synthetic traffic generator. Each of these data flows is mapped to one of the three available UMTS transport channels (Table 4) following the guidelines given in [18]:

- The Dedicated CHannel (DCH): this is the only dedicated transport channel in the UMTS standard, it carries the service data of a single user at variable bit rate,
- The Dedicated Shared CHannel (DSCH): this transport channel carries signalling and service data of several users at variable bit rate, or
- The Forward Access CHannel (FACH): this last transport channel carries also the service data of several users but at fixed bit rate.

We consider that each sector in the emulated world may only have one FACH, one DSCH and a number of DCHs depending on the RNC's Spreading Factor (SF) management. Those sole FACH and DSCH however collect traffic from several active users. Since FACH and DCH carry user data flow with a fixed bit rate depending on their SF, the bandwidth of the DSCH is variable since its SF occupies the free room left in the Orthogonal Variable Spreading Factor (OVSF) tree after the DCHs SF allocation.

The Real Time (RT) sessions (Conversational and Streaming) are directly allocated to a DCH. This is natural since this kind of traffic can not tolerate the significant packet delay or the delay variation likely to appear when using shared channels. Since there is no bandwidth reservation mechanism into shared channels, RT sessions are guided to dedicated channels only.

In the opposite way, Non Real Time (NRT) sessions (Interactive and Background) may support a much longer delay. So, these traffics accept to be guided to either dedicated or shared channels since the packet scheduling of the shared channels does not affect the NRT sessions as it does to the RT ones.

6.2 Received Power

The quality of a transmission between a NodeB and a UE can be evaluated by the received power the UE is getting. This power is a direct function of the NodeB emitted power and is also influenced by a number of factors. At the receiver side, the carrier to interference ratio is calculated as followed [19]:

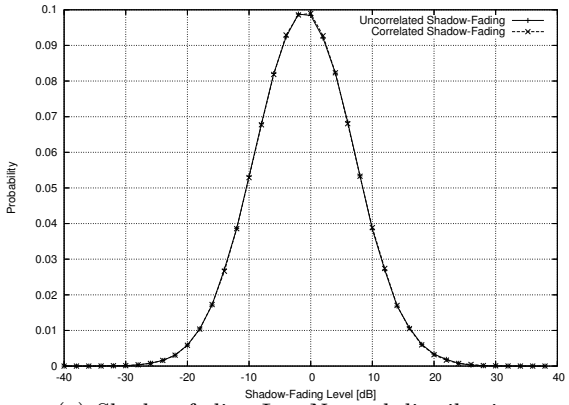
$$\left(\frac{C}{I}\right)_{CH} = \frac{Sh \;.\; Pl \;.\; (P_t)_{CH}}{\alpha I_{intra} + I_{inter} + P_N} \tag{3}$$

where Sh is shadow fading affecting a communication at a precise location in the emulated world. Pl is the path loss the communication undergoes at the same position. It has already been introduced at Section 5.3. $(P_t)_{CH}$ is the transmitted power the NodeB allocates to the transport channel. I_{intra} is the intracell interference which is generated by those users that are connected to the same base station that the observed user. I_{inter} is the intercell interference which is generated from the other cells. α is the orthogonality factor which takes into account the fact that all the downlink communications are not perfectly orthogonal due to multipath propagation. An orthogonality factor of 0 corresponds to perfectly orthogonal intracell users while with the value of 1 the intracell interference has the same effect as intercell interference. Assumed value for the orthogonality factor is 0.4 in the macrocellular case. P_N is thermal noise power which is assumed to be equal to -99 dBm.

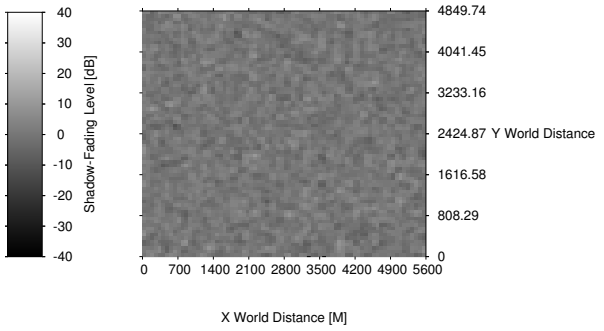
The shadow fading is a long-term (Log-Normal) fading characterised by a Gaussian distribution with zero mean and a standard deviation chosen to be 8 dB [20] as shown on Fig. 12(a). Due to the slow fading process versus distance Δx, adjacent fading values are correlated. Its normalised autocorrelation function $R(\Delta x)$ can be described with sufficient accuracy by an exponential function:

$$R(\Delta x) = e^{-\frac{|\Delta x|}{d_{cor}} ln2} \tag{4}$$

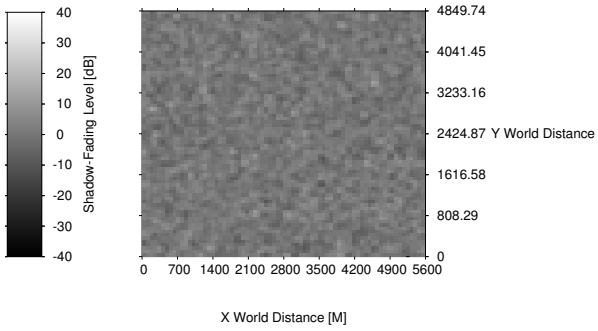
with the decorrelation length d_{cor}, which is dependent on the environment. This concept can be applied in the vehicular test environment with a decorrelation length of 20 metres [10]. Fig 12(b) and 12(c) respectively shows the shadow fading emulated on the testbed before and after the correlation process.



(a) Shadow fading Log-Normal distribution.



(b) Uncorrelated shadow fading.



(c) Correlated shadow fading.

Figure 12: Emulated shadow fading.

As a proof of validation, we calculated the Gfactor [21] on every possible position in our emulated world. The Gfactor is the ratio of intracell/intercell interference, but following the OWNP2 approach presented in [22], the intercell interference is calculated based on the received power from the two other sectors of our *SNodeB* plus the received power from 2 other NodeBs generating the most interferences. Fig. 13 shows the comparison between our results and those obtained by their simulator.

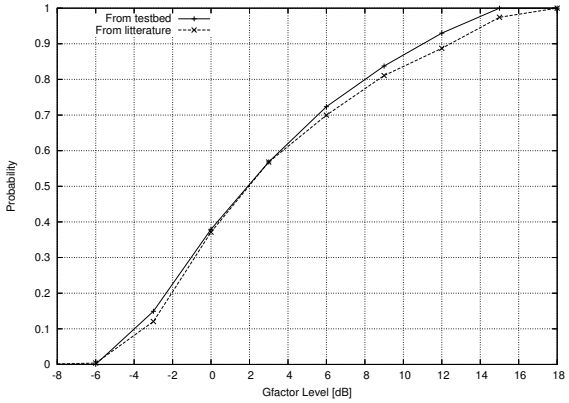


Figure 13: Gfactor distribution in the emulated world.

From the $(C/I)_{CH}$ evaluated as in (3), the normalised energy per information bit is obtained as followed:

$$\left(\frac{E_b}{N_0}\right)_{CH} = \frac{1}{2R_{CH}} \cdot SF_{CH} \cdot \left(\frac{C}{I}\right)_{CH} \tag{5}$$

where R_{CH} is the coding rate of the channel and SF_{CH} its spreading factor.

Thanks to (5) it is now possible to calculate the power received in each transport channel presented in Table 4. From the DCH point of view, due to its fast power control, the E_b/N_0 ratio is considered as constant and fixed during the whole emulation.

The FACH transmission power is constant and defined as -5 dB relative to the CPICH transmission power [23]. From Table 3, we determine the CPICH transmission power to 33 dBm meaning that the FACH is emitted at 28 dBm. The E_b/N_0 of the UE receiving a communication through the FACH is then evolving during the emulation based on its position.

The DSCH transmission power is controlled using the fast power control of its associated DCH with a power offset. Considering that this offset is a manner to counter-balance the difference between the SF of these two channels, we may fix it to SF_{DCH}/SF_{DSCH} as in [24] giving us the following equations:

$$(P_t)_{DSCH} = (P_t)_{DCH} \cdot \frac{SF_{DCH}}{SF_{DSCH}} \tag{6}$$

From (3) and (5), we have:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}} \cdot SF_{DSCH} \cdot \frac{Sh \,.\, Pl \,.\, (P_t)_{DSCH}}{\alpha I_{intra} + I_{inter} + P_N} \tag{7}$$

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}} \cdot SF_{DSCH} \cdot \frac{Sh \,.\, Pl \,.\, (P_t)_{DCH}}{\alpha I_{intra} + I_{inter} + P_N} \cdot \frac{SF_{DCH}}{SF_{DSCH}} \tag{8}$$

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}} \cdot SF_{DCH} \cdot \frac{Sh \,.\, Pl \,.\, (P_t)_{DCH}}{\alpha I_{intra} + I_{inter} + P_N} \tag{9}$$

Considering that $(\alpha I_{intra} + I_{inter} + P_N)$ has approximately the same value in both cases, we have:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DSCH}} \cdot SF_{DCH} \cdot \left(\frac{C}{I}\right)_{DCH} \tag{10}$$

Supposing now that the coding rate of these channels are equal, we finally have:

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \frac{1}{2R_{DCH}} \cdot SF_{DCH} \cdot \left(\frac{C}{I}\right)_{DCH} \tag{11}$$

$$\left(\frac{E_b}{N_0}\right)_{DSCH} = \left(\frac{E_b}{N_0}\right)_{DCH} \tag{12}$$

We may then consider that a UE using the DSCH enjoys the same E_b/N_0 ratio than through its associated DCH.

6.3 Time-Correlated Errors

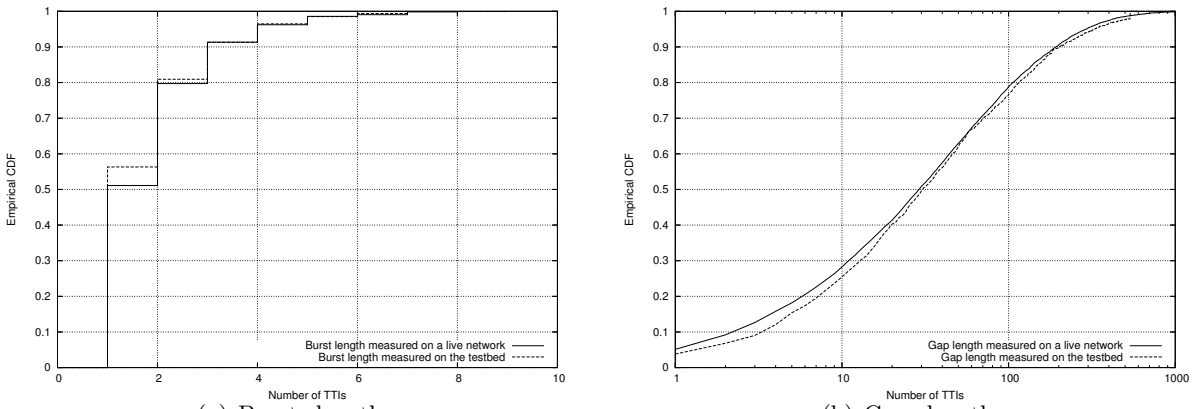
Wireless networks encounter typically more errors than wired ones, so the main objective of our testbed is to emulate the lower layers of the UMTS air interface over an Ethernet (wired) link. The emulation of the Radio Link Control (RLC) and Medium Access Control (MAC) layers is achieved using the Linux Traffic Controller (TC) which copes with packet shaping, scheduling, policing and dropping. More precisely, we use the TC NETwork EMulation component (NETEM) [25] which is a waiting queue able to delay, drop, corrupt, duplicate or even reorder packets.

The NETEM packet corruption functionality is the most appropriate to approach what packets really undergo over the UMTS radio link. Indeed, it introduces a single bit error at a random position in the packet. Instead of being dropped, this corrupted packet is transmitted to the UE, hence consuming a part of the bandwidth allocated to the flow it belongs to. Eventually, this corrupted packet is dropped, because of the CRC error detected by Ethernet.

It has been proven that the typical bit errors over a radio link are not totally decorrelated and tend to appear in bursts. One of the analytical models trying to describe this property is the Gilbert-Elliot model [26, 27]. It uses a two-state (error and error-free) time-homogeneous discrete time Markov chain to describe the channel variations. This model is a popular one for wireless bit errors since it is complex enough to capture burstiness and simple enough to be treated analytically.

Moreover, recent results have shown that an enhanced version of the Gilbert-Elliot model approaches even more the UMTS radio channel [28]. The error state in this new model is the same than the one in the simple two-state Markov model. However, the error-free state of the new model has been modified. The duration of the stay at that state is now taken from a Weibull distributed random variable which correctly fits the measurements over real networks.

We have then modified the NETEM module to respect to these specifications. We will consider the two states of the Markovian chain at the Transmit Time Interval (TTI) level: if we are in the error-free (resp. error) state, all the transmit RLC Packet Data Units (PDUs) transiting over the air interface during the TTI are considered as well received (resp. corrupted). We may then define a burst as a number of subsequent erroneous TTIs (Fig. 14(a)) whereas a gap is a number of subsequent error-free TTIs (Fig. 14(b)).



(a) Bursts length. (b) Gaps length.

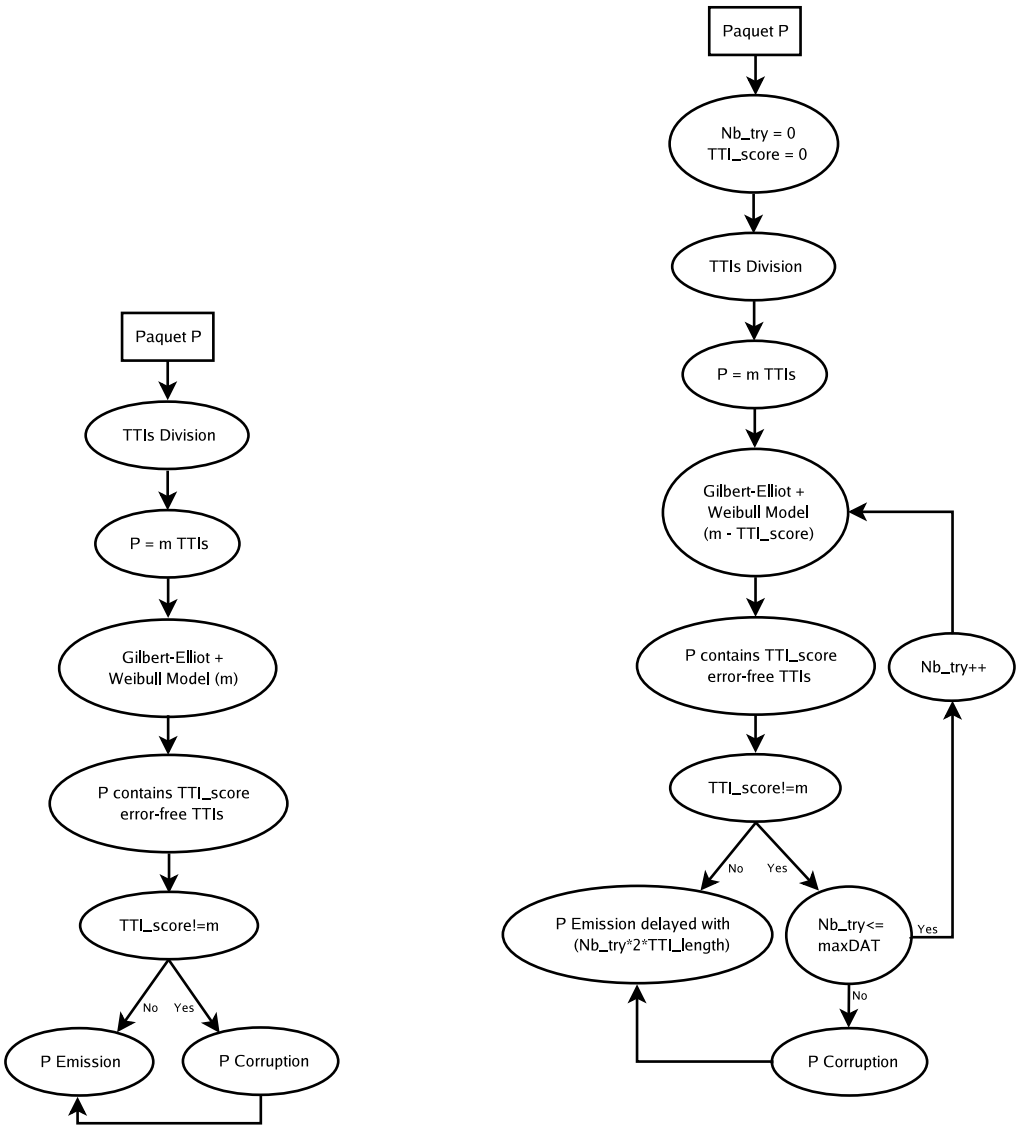
Figure 14: Enhanced NETEM module behaviour.

6.4 Typical Radio Link BER

The RLC layer is able to transmit RLC PDUs in three different modes: Acknowledged Mode (AM), Unacknowledged Mode (UM) and Transparent Mode (TM). Note that the Automatic Repeat Request (ARQ) mechanism is only available in the AM.

We decided to focus on two of them, namely the TM to transmit the RT traffic to handle their intolerance to great packet delay. On the other hand, we use the AM for NRT traffic to lower the packet error rate. The ARQ mechanism is able to retransmit an erroneous PDU a fixed number of times before declaring this PDU as lost. The number of retransmissions is known as *max*DAT and is fixed at 3 in the emulation following the proposition made in [29, 30].

Those two modes have been adapted to an Ethernet link and included in the enhanced version of the NETEM module as described on Fig. 15:



(a) TM. (b) AM.

Figure 15: RLC TM and AM implemented in NETEM.

Focusing on the Fig. 15(a), when a packet P arrives, NETEM computes the numbers of TTIs needed to transmit P over the air interface. This number is referred to as m. This division is mainly based on P's length and on the SF allocated to the flow P belongs to. All these m TTIs are then injected into the two-states Markovian chain modified with the Weibull distribution presented in Section 6.3 to calculate the number of error-free TTIs referred to as TTI_score. If P does not have any erroneous TTI (meaning m is equal to TTI_score), P is correctly transmitted. However, if P has at least one erroneous TTI, P is corrupted before being transmitted to its final destination.

From the AM point of view (Fig. 15(b)), the computation is just like in the TM case but if P contains erroneous TTIs, we reinject these $(m - TTI\_score)$ TTIs into the Markovian chain until all the TTIs needed to transmit P are error-free or until the number of attempts (referred to as Nb_try) exceeds maxDAT. If all TTIs are error-free, P is transmitted with a $Nb\_try \times 2 \times TTI\_length$ delay including the ARQ delay due to the retransmissions. On the other hand, if there is still at least one erroneous TTI at the end of the computation, P is corrupted, then transmitted with a $(maxDAT + 1) \times 2 \times TTI\_length$ delay.

Table 4: Traffic Classes and Channels Mapping

Traffic Classes	**Authorised Transport Channel(s)**	**RLC Mode**	**Target BER**
Conversational	DCH	TM	10^{-4}
Interactive	DCH, DSCH or FACH	AM	10^{-5}
Streaming	DCH	TM	10^{-4}
Background	DCH, DSCH or FACH	AM	10^{-5}

Now, for each traffic classes, we are able to know in which UMTS transport channel it has to be transmitted thanks to Table 4. Based on this channel and following the equations of Section 6.2 and the SF allocation map presented in [31], we can derive the UE E_b/N_0 ratio. Considering a coding rate of 1/2, for each E_b/N_0 value, the curve in Fig. 16 (taken from [32]) gives the corresponding BER that communication undergoes.

On the DCH, the only transport channel supporting a dedicated fast power control, the target E_b/N_0 setpoint is permanently adjusted and aims at a constant quality, usually defined as a certain target BER, depending on the traffic class. Following specifications presented in [3], we have fixed the BER for the RT traffic at 10^{-4} and 10^{-5} for NRT traffic.

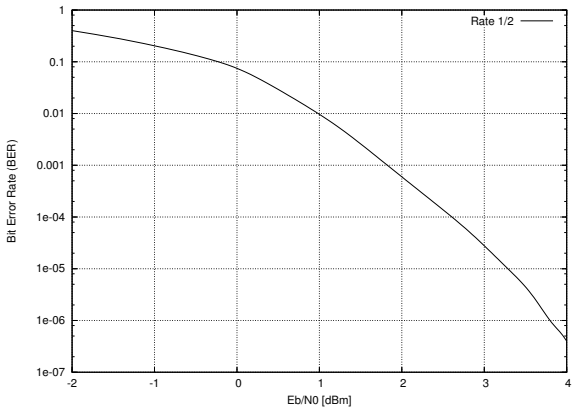


Figure 16: BER adopted in UMTS as function of the bit normalised energy.

Finally, assuming independent bit errors despite correlated errors at TTI level for simplification purposes, this BER enables to approximate the BLock Error Rate (BLER) as follows:

$$BLER \;=\; 1-(1-BER)^{l} \tag{13}$$

where l is the PDU length which has been fixed to 336 bits [33]. This BLER is then injected into our modified version of NETEM with two other informations: the allocated SF and the RLC mode used.

6.5 Implementation

Using TC and its Token Bucket (TB) module to mimic the different available transport channels, and its NETEM module to emulate the packet loss over the UMTS air interface, we can now build a structure to manage all the packets going from a serving NodeB to one of its UEs. Fig. 17 illustrates the implementation of one of our NodeBs:

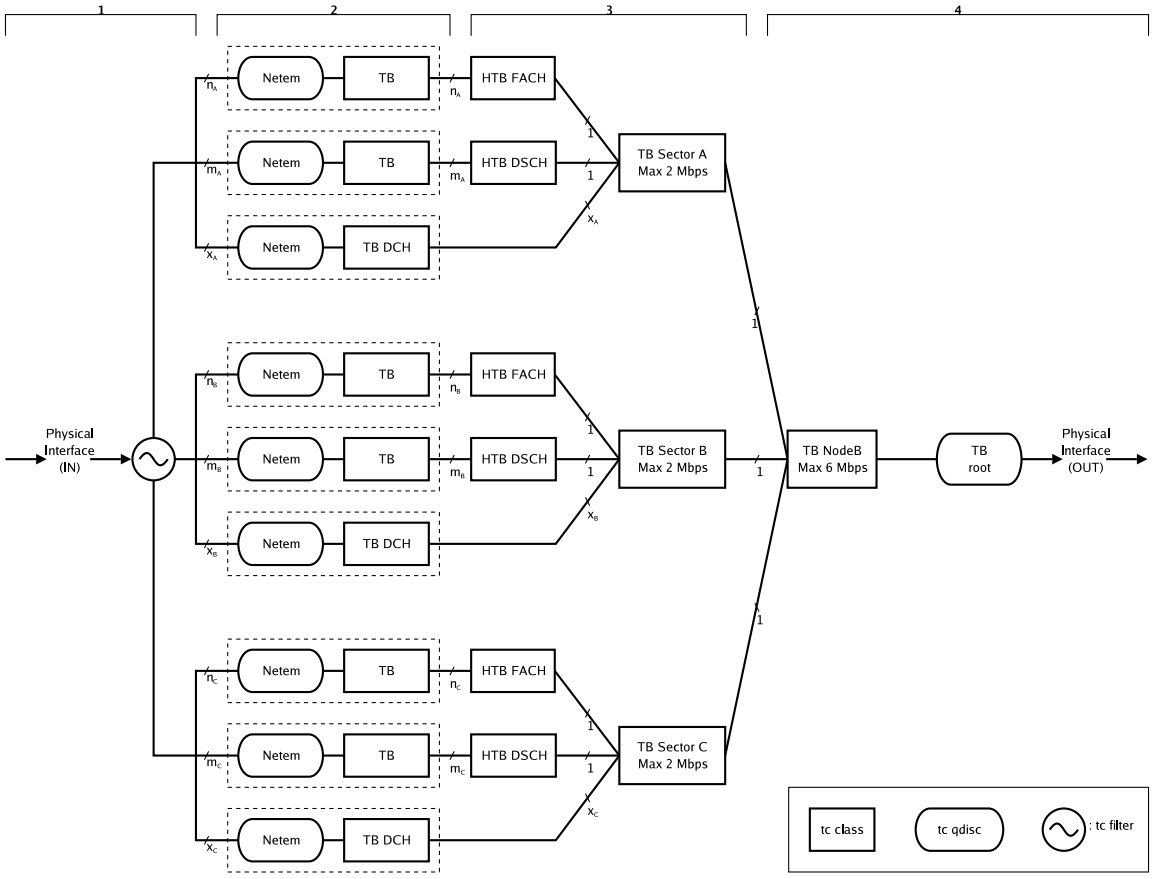


Figure 17: Air Interface Emulation

1 First of all, a new downlink data flow is filtered based on the destination IPv6 address and the "Traffic Class" field of the IPv6 header. Note that the filtering could also easily be based on the Differentiated Services Code Point (DSCP) of a arriving packet. Following these header fields we know exactly in which sector the UE receiving this data flow is located and which transport channel this flow will use over the air interface.

2 A new instance of a NETEM queue connected to a TB is then created. We corrupt some of the packets following the principles exposed in the previous sections. The TB is only responsible of the bandwidth consumption of the flow ensuring that it never exceeds the maximum bandwidth it has received via the SF allocation. In this architecture, there is a NETEM-TB instance for each single data flow. That means there are $\mathtt{n}_i$ flows sharing the FACH, $\mathtt{m}_i$ flows sharing the DSCH and $\mathtt{x}_i$ flows each of them occupying a single DCH of the sector `i` (`i`=`A`, `B` or `C`).

3 Notice that if the flow is using a common channel (FACH or DSCH), it has to share the channel bandwidth with other flows. All NETEM-TB instances belonging to the same common channel are then supervised by a Hierarchical TB (HTB) which has the ability to borrow some unused tokens from a given TB to give them to another TB waiting for tokens. If the flow is using a DCH, it is directly scheduled at the TB responsible of a sector. This last TB has to equitably manage $(\mathtt{x}_i+2)$ entries: the $\mathtt{x}_i$ DCHs + 1 FACH + 1 DSCH of the sector `i` (`i`=`A`, `B` or `C`). It also has to guarantee that the total bandwidth consumption of the sector never exceeds 2 $Mbps$. Each entries of the (H)TB is managed one after the other in a round-robin scheme, but this scheduling is sufficiently fast so that the impossibility to simultaneously transmit packets over Ethernet does not influence the end-to-end performances.

4 Finally, the three TBs managing the three sectors are controlled by the last TB representing the NodeB itself.

The modularity of the testbed appears in Fig. 17. To support a new transport channel like the High Speed-DSCH (HS-DSCH), one only needs to add a new NETEM-TB-HTB branch, where the NETEM behaviour would be driven by the current modulation and coding scheme, and users would be scheduled in that channel by the HTB. The same reasoning applies to LTE.

7 Results

Using the testbed and the presented TC structure, we have launched an emulation loading each sector with 7 UEs (total of 84 UEs in the emulated world) generating the different kinds of traffic during $5,400$ s and moving around into an urban environment. In the urban area scenario, the UE speed distribution is the following: 30% at 3 km/h, 40% at 30 km/h, 20% at 70 km/h and 10% at 120 km/h. The results of the emulation are summarised in Table 5.

On the one hand, the NRT traffics undergo a greater mean packet delay with the NETEM module included into the TC structure. This is obvious since Interactive and Background flows use TCP as transport protocol for the sake of reliability. With TCP, each corrupted packet has to be retransmitted but it also triggers the congestion control. As a result, following packets will get a lower bandwidth and will then reach their destination with a greater delay. The ARQ mechanism plays also a role in this greater delay since it introduces retransmission at the RLC layer. The jitter standard deviation increases also in the same proportions.

On the other hand, the RT traffic may support some packet losses while the focus is mainly set on delivery time. Conversational and Streaming flows use UDP as transport protocol. This leads to the same mean packet delay with or without NETEM since no retransmission is supported using UDP. As for the mean packet delay, the jitter standard deviation of this kind of traffic tends to be stable. But the number of lost packets raises considerably.

Table 5: Traffic Classes Delay

	without NETEM			with NETEM		
	Pck. Delay [ms]	**Jit.** [ms]	**Pck. Lost** [%]	**Pck. Delay** [ms]	**Jit.** [ms]	**Pck. Lost** [%]
Conversational	46	39	0.02	36	38	16.99
Interactive	1,439	623	0.00	1,622	644	0.00
Streaming	4	10	0.07	9	19	17.66
Background	5,089	988	0.00	12,499	1,613	0.00

8 Conclusion

We have shown that, with a minimal investment, it is possible to build a testbed mimicking a real UMTS network. This paper has presented our testbed and how it emulates the UMTS air interface over a wired network, the different transport channels and the higher BER. Such a testbed is able to evaluate the quality of user experience of services delivered over a cellular network. This way of doing reproduces the real network in a different way than simulators do. It is really important since such an approach may reveal unsuspected problems.

Acknowledgements

The authors would like to thank Yuji Sekigushi for the helpful and valuable contribution he delivered during his traineeship at FUNDP.

References

[1] Quagga routing software suite. `http://www.quagga.net`, last visited: 15 February 2007.

[2] Traffic generator tool. `http://www.postel.org/tg/tg.htm`, last visited: 15 February 2007.

[3] Technical Specification Group Radio Access Network. TS 23.107 V6.4.0, Quality of Service (QoS) concept and architecture. Technical report, 3^{rd} Generation Partnership Project (3GPP), March 2006.

[4] J.A. Aguiar and L.M. Correia. Traffic Source Models for the Simulation of Next Generation Mobile Networks. In *Proceedings of the* 7^{th} *Wireless Personal Multimedia Communications (WPMC)*, Abano Terme (Italy), September 2004.

[5] A.B. García, E. García, M. Álvarez-Campana, J. Berrocal, and E. Vázquez. A Simulation Tool for Dimensioning and Performance Evaluation of the UMTS Terrestrial Radio Access Network. *Lecture Notes In Computer Science*, 2515:49–60, 2005.

[6] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. Traffic modeling and characterization for UMTS networks. In *Proceedings of the* 44^{th} *Globecom conference*, San Antonio (TX - USA), November 2001.

[7] Y. Li Frank and Stol Norvald. QoS Provisioning using Traffic Shaping and Policing in 3rd-generation Wireless Networks. In *Proceedings of the* 3^{rd} *Wireless Communications and Networking Conference (WCNC)*, Orlando (FL - USA), March 2002.

[8] Soam Acharya and Brian Smith. An Experiment to Characterize Videos Stored on the Web. In *Proceedings of the* 5^{th} *Multimedia Computing and Networking (MMCN)*, USA, January 1998.

[9] A. Reyes, E. Gonzalez-Parada, E. Casilari, J.C. Casasola, and A. Díaz-Estrella. A page-oriented WWW traffic model for wireless system simulations. *Teletraffic Engineering in a Competitive World (ITC16)*, pages 1271–1280, 1999.

[10] Technical Specification Group Radio Access Network. TS 30.03 V3.2.0, Selection procedures for the choice of radio transmission technologies of the UMTS. Technical report, 3^{rd} Generation Partnership Project (3GPP), March 1998.

[11] Technical Specification Group Radio Access Network. TR 25.896 V6.0.0, Feasibility Study for Enhanced Uplink for UTRA FDD (Release 6). Technical report, 3^{rd} Generation Partnership Project (3GPP), April 2004.

[12] Technical Specification Group Radio Access Network. TR 25.933 V5.4.0, IP transport in UTRAN (Release 5). Technical report, 3^{rd} Generation Partnership Project (3GPP), January 2004.

[13] Technical Specification Group Radio Access Network. TR 25.876 V1.8.0, Multiple Input Multiple Output in UTRA. Technical report, 3^{rd} Generation Partnership Project (3GPP), December 2005.

[14] Technical Specification Group Radio Access Network. TR 25.892 V6.0.0, Feasibility study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement. Technical report, 3^{rd} Generation Partnership Project (3GPP), June 2004.

[15] Alexandrosz Burulitisz, Sándor Imre, and Sándor Szabó. On the Accurancy of Mobility Modelling in Wireless Networks. In *Proceedings of the* 39^{st} *International Conference on Communications (ICC)*, Paris (France), June 2004.

[16] COST 231 Final Report. Digital Mobile Radio Towards Future Generation Systems. Technical report, COST 231, November 1999.

[17] Spatial Channel Model AdHoc Group. Spatial Channel Model Text Description. Technical report, 3GPP & 3GPP2, April 2003.

[18] Harri Holma and Antti Toskala. *WCDMA for UMTS*. Wiley inter-science, third edition, October 2004.

[19] Technical Specification Group Radio Access Network. TR 25.942 V6.4.0, Radio Frequency (RF) system scenarios (Release 6). Technical report, 3^{rd} Generation Partnership Project (3GPP), March 2005.

[20] Jay Kumar Sundararajan and Harish Viswanathan. Comparison of Schemes for Streaming Multicast in Cellular Networks with Relays. In *Proceedings of the* 7^{th} *Wireless Communications and Networking Conference (WCNC)*, Las Vegas (NV - USA), March 2006.

[21] Javier Rodríguez Fonollosa, Markku Heikkilä, Xavier Mestre, Alba Pagès, Adam Pollard, Laurent Schumacher, Lars Torsten Berger, Ami Wiesel, and Juha Ylitalo. Adaptive Modulation Schemes for MIMO HSDPA. In *Proceedings of the* 11^{th} *IST Mobile & Wireless Telecommunications Summit*, pages 78–82, Thessaloniki (Greece), June 2002.

[22] Lars Torsten Berger. *Performance of Multi-Antenna Enhanced HSDPA*. PhD thesis, Aalborg University, April 2005.

[23] Harri Holma and Antti Toskala. *HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. Wiley inter-science, April 2006.

[24] Paolo Giacomazzi, Luigi Musumeci, and Giacomo Verticale. Performance of Web-browsing services over the WCDMA-FDD downlink shared channel. In *Proceedings of the* 44^{th} *Globecom conference*, San Antonio (TX - USA), November 2001.

[25] Stephen Hemminger. Network Emulation with NetEm. In *Proceedings of the* 6^{th} *Australia's National Linux Conference (LCA)*, Canberra (Australia), April 2005.

[26] E. N. Gilbert. Capacity of a Burst-Noise Channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960.

[27] E. O. Elliot. Estimates of Error Rates for Codes on Burst-Noise Channels. *Bell Systems Technical Journal*, 42:1977–1997, September 1963.

[28] Wolfgang Karner. A UMTS DL DCH BLER Model Based on Measurements in Live Networks. Technical report, COST 273, January 2005.

[29] Oumer Mohammed Teyeb. *Quality of Packet Services in UMTS and Heterogeneous Networks*. PhD thesis, Aalborg University, October 2006.

[30] Olivier De Mey, Laurent Schumacher, and Xavier Dubois. Optimum Number of RLC Retransmissions for Best TCP Performance in UTRAN. In *Proceedings of the* 16^{th} *Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin (Germany), September 2005.

[31] Hugues Van Peteghem and Laurent Schumacher. Emulation of a Downlink Spreading Factor Allocation Strategy for Rel'99 UMTS. In *Proceedings of the* 64^{th} *Vehicular Technology Conference (VTC-fall)*, Montreal (Canada), September 2006.

[32] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. Packet service in UMTS: delay-throughput performance of the downlink shared channel. *Computer Networks*, 38(1):43–59, January 2002.

[33] Special Article on IMT-2000 Services. *NTT DoCoMo Technical Journal*, 3(2), September 2001.