

An LTE module for the ns-3 network simulator

Giuseppe Piro
DEE - Politecnico di Bari, Bari, Italy
g.piro@poliba.it

Nicola Baldo, Marco Miozzo
Centre Tecnològic de Telecomunicacions de
Catalunya, Barcelona, Spain
{nbaldo,mmiozzo}@cttc.es

ABSTRACT

In this paper, we present a novel module developed for the simulation of the LTE technology with the ns-3 simulator. It focuses mainly on modeling the E-UTRA part of the system, with a particular attention on the aspects related to the channel, PHY and MAC layers. First we discuss the overall modeling assumptions according to which the module was designed. Subsequently, we describe in detail the architecture of the module and its components. Finally, we discuss an example simulation scenario.

Categories and Subject Descriptors

I.6.5 [Simulation and Modeling]: Model Development—*Modeling methodologies*; I.6.7 [Simulation and Modeling]: Simulation Support Systems—*Environments*

General Terms

Simulation, Modeling, Algorithm, Design, Performance

Keywords

LTE, Channel, Propagation, PHY, MAC, RRM

1. INTRODUCTION

In the last years the 3rd Generation Partnership Project (3GPP) has been working on the definition of a new mobile communication technology called Long Term Evolution (LTE), which provides a significant improvement with respect to the currently widely-deployed Universal Mobile Telecommunications System (UMTS) and High-Speed Packet Access (HSPA) systems.

Thanks to features like enhanced data rates and lower power consumption, LTE represents an emerging and promising technology for providing ubiquitous broadband Internet access. For this reason, the research community is putting considerable effort into studying LTE with the objective of designing newer and innovative solutions that can maximize

its performance, both from a Radio Access Network (RAN) perspective and from the point of view of the Core Network. For instance, in order to improve the spectral efficiency, the Orthogonal Frequency Division Multiple Access (OFDMA) technique is introduced, together with Multiple-Input Multiple-Output (MIMO) schemes. At the same time, LTE completes the transition from Circuit Switched core network solutions (still heavily relied upon in HSPA) to an all-IP architecture, which had been long regarded as a much desirable feature, both for the increasing usage of data services by mobile users, and for the higher flexibility and potential lower cost of the Core Network equipment.

As the interest in LTE as a mobile communication technology increases, there is a growing need for instruments able to characterize the performance of LTE systems in order to tune and optimize them. This need is present at various stages: for the standardization of future version of the technology, as well as for the design and implementation of devices and network equipments and the roll-out of LTE network infrastructures. LTE is a rather complex technology, and for this reason traditional performance evaluation methods based on analytical models can be applied only to very small portions of it. As a consequence, researchers and engineers most often rely on simulators to assess the performance of LTE systems. In particular, open source network simulation tools are highly valued by many researchers, because of their usually wider acceptance within the research community. In the literature, several LTE simulators have been proposed for different purposes. Physical layer simulation tools such as [10] and [18] are quite limited in that they consider the performance at the physical layer only, normally focusing on a single transmitter-receiver pair. System level simulators, such as [14] and [13], go beyond the physical layer by normally introducing the MAC layer together with an abstract model of the higher layers. They are usually adopted in the research community for the evaluation of Radio Resource Management (RRM) algorithms, but due to their abstraction of the higher layers they cannot provide an accurate evaluation of the end-to-end performance and of the behavior of the network as a whole. The most promising solution to this problem is represented by network simulators, which feature an accurate implementation of all the protocol layers from the MAC up to the application.

A first step in this direction is done with [20], which is an open source LTE network simulator that supports single and multi-cell environments, QoS management, multi users environment, user mobility, handover procedures, scheduling, and frequency reuse techniques. Unfortunately, since

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

WNS-3 2011, March 21, Barcelona, Spain
Copyright © 2011 ICST 978-1-936968-00-8
DOI 10.4108/icst.simutools.2011.245571

this simulator aims at supporting LTE only, it lacks the many features of general purpose network simulator like ns-3, for example the availability of a full-fledged TCP/IP protocol stack, the support for other wireless technologies such as WiFi and WiMAX, the possibility of integration with testbeds and real application binaries, etc. The simulation module that we describe in this paper was explicitly designed in order to fill this gap. It introduces support for the simulation of the LTE technology in the ns-3 simulator, with a particular attention on modeling the most relevant aspects of the Evolved Universal Terrestrial Radio Access (E-UTRA), which is the radio interface of LTE [2].

2. THE LTE MODULE FOR NS-3

The development of the LTE module for ns-3 was carried out during the Google Summer of Code 2010. This module provides a basic implementation of LTE devices, including propagation models and PHY and MAC layers. Due to the intrinsic complexity of the LTE standard and the limited time of the GSoC framework, at the time of this writing it is not yet possible to simulate a complete LTE network. However, the proposed module already allows the simulation of several important aspects of LTE systems, such as for example downlink RRM and MAC scheduling and, moreover, it provides a very good basis for further extensions as well as the development of a complete tool in the future.

In summary, the most important features provided by the proposed module are (i) a basic implementation of both the User Equipment (UE) and the enhanced NodeB (eNB) devices, (ii) Radio Resource Control (RRC) entities for both the UE and the eNB, (iii) a state-of-the-art Adaptive Modulation and Coding (AMC) scheme for the downlink [9], (iv) the management of the data radio bearers (with their QoS parameters), the MAC queues and the RLC instances, (v) Channel Quality Indicator (CQI) management, (vi) support for both uplink and downlink packet scheduling, (vii) a PHY layer model with Resource Block level granularity and (viii) a channel model with the outdoor E-UTRAN propagation loss model.

The module is built completely in C++ and, at the time of this writing, comprises 89 classes and approximately 9000 lines of code. Figure 1 shows the UML diagram of the most important classes that compose the module. It is important to remark that the diagram only reports the most important data members and functions. Some details about the relationship among classes have been omitted due to space limitations.

2.1 LTE PHY and E-UTRAN channel models

It is our opinion that the most interesting and immediate use case for an LTE Network Simulator is to aid in the design and evaluation of RRM solutions. RRM in LTE is characterized by a high flexibility: thanks to the OFDMA and Single Carrier Frequency Division Multiple Access (SC-FDMA) access technologies, different radio resources in the time and frequency domain can be dynamically assigned to different users. The reason behind this design choice is to allow the RRM process to opportunistically exploit the variations of the channel conditions that happen both in time and frequency. For example, due to frequency selective fading it can happen that at a certain time slots a particular set of subcarriers is not suitable for communication with certain mobile users; then, by opportunistically assigning

to each user only those subcarriers which are known to be good for communication, the system can enjoy an enhanced capacity. Clearly, in order to properly evaluate such opportunistic RRM algorithm in an LTE network simulator, it is important that the simulator provides suitable means to model the time- and frequency-varying aspects of the communications in LTE system.

Ns3::Spectrum[11] was proposed to address this issue in ns3. It is a framework that provides support for spectrum-aware channel and PHY simulation, which is particularly suited for the modeling of the OFDMA and SC-FDMA technologies. Due to these characteristics, we chose to base our implementation of the LTE PHY and channel models on Ns3::Spectrum. In the definition of a particular wireless model based on the Spectrum framework, the first thing to do is to define (i) a set of frequencies/channels to use at the physical layer and (ii) the signal generated at the PHY and transported through the channel. To this aim, corresponding specialized versions of the *SpectrumModel* class have to be developed in order to define the portions of radio spectrum used by the considered technology and the granularity of their representation. In our case, we considered two separate *SpectrumModel* instances, one for the uplink (UL) and one for the downlink (DL), in order to model the paired spectrum in which LTE operates according to the Frequency Division Duplex (FDD) mode. At the present, the proposed module supports a spectrum model for the first E-UTRAN FDD operative band (i.e., [1929-1980] MHz for the UL and [2110-2170] MHz for the DL) [7]. In particular, our *SpectrumModel* is designed to describe a set of sub-channels used for transmission, defined as Resource Blocks (RBs) in 3GPP nomenclature. Since in LTE radio resources are divided in sub-channels of 180 kHz each, we chose this as the granularity of our spectrum model. Moreover, the *LteSpectrumValueHelper* class has been developed to make it easy to create the *SpectrumValue* corresponding to a given signal, taking into account in particular the set of sub-channels to be used for the transmission. During the simulation, the PHY layer uses this class to create the *SpectrumValue* instances that represent the Power Spectral Density (PSD) of both signal and noise. In [11] the channel/PHY interface has been defined by the abstract base classes *SpectrumChannel* and *SpectrumPhy*. The most important role of these classes is the transmission and receptions of signals. In particular, the *SpectrumChannel* class (i) handles the packet transmission process among attached PHY layers and (ii) it includes the channel propagation models. As said before, our LTE module supports the FDD model only, therefore in the implementation we use two separate *SpectrumChannel* objects in order to manage uplink and downlink independently; this choice is motivated by its lower simulation complexity.¹ In particular, the *SpectrumChannel* implementation that we use is *SingleModelSpectrumChannel*, which was already provided by Ns3::Spectrum. This is an acceptable choice, since the propagation model (for which we wanted to develop a version specific for LTE) is actually a separate model that

¹For example, let us consider a single cell FDD system with N users. If this system is simulated using a single *SpectrumChannel* object, the number of channel evaluations for each slot is $O(N^2)$, whereas it is just $O(N)$ if a separate channel object is used for the uplink and downlink. This is because in the uplink we are not actually interested in simulating the propagation among the mobile users.

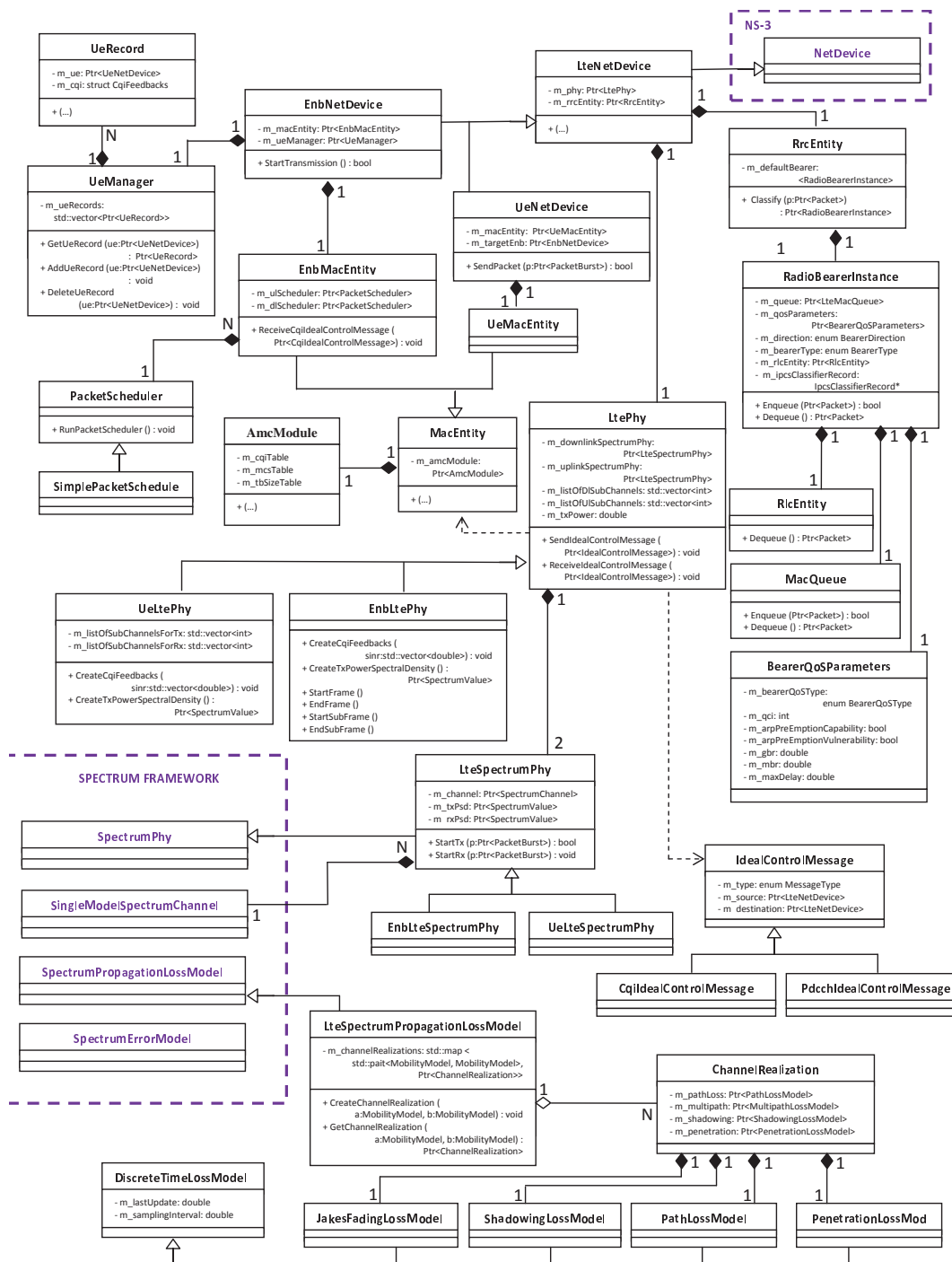


Figure 1: The class diagram.

is plugged into the channel object. Our LTE-specific propagation model will be discussed in detail in Section 2.1.3.

The *UeLtePhy* and the *EnbLtePhy* classes represent the physical layer of the UE and eNB devices, respectively. They inherit from the *LtePhy* class, which provides their common functionalities. In order to manage separately uplink and downlink, the LTE PHY layer has been conceived as a container of two different baseband subsystems. Each of these subsystems is modeled by the *LteSpectrumPhy* class, which is then specialized into the *UeLteSpectrumPhy* and the *EnbLteSpectrumPhy* classes, respectively. As we said before, each of these classes is attached to a different *SpectrumChannel* object. The FDD access mode defines a different set of sub-channels for both uplink and downlink. A list of available sub-channels for the uplink and the downlink are stored into the *m_ListOfUplinkSubchannel* and *m_ListOfDownlinkSubchannel* member variables of the *LtePhy* class, respectively. We note that these sets of subchannels must be defined at the beginning of the simulation. The *SendPacket()* method is the entry point for the PHY and handles the transmission of the packets on the channel. The UE and the eNB use different PHY interfaces to transmit packets.

Finally, the physical layer is also responsible for the transmission and the reception of control messages. With this respect, we note that the proposed module has the objective of modeling the transmission of user data only. As a consequence, in order to limit the complexity of the simulator and hence simulation time, an ideal control channel was implemented. This means that control messages exchanged among nodes are not subject to channel and propagation modeling, and therefore they are always received correctly. From an implementation point of view, the physical layer delivers the message to the destination directly, without passing through the channel object. A control message is sent using the *SendIdealControlMessage* method of the *LtePhy* class. The *ReceiveIdealControlMessage* method of the *LtePhy* class handles the reception of the control message. According to the message type, it delivers the message to a proper device entity for processing.

2.1.1 The eNB physical layer

The *EnbLtePhy* class models the eNB physical layer. The first task of this class is to handle the transmission and reception of signals by the eNB. With this respect, the transmission procedure is performed by the *StartTx()* member method of the downlink *EnbSpectrumPhy* object, while the reception procedure is performed by the *StartRx()* member method of the uplink *EnbSpectrumPhy* object.

At the end of the reception, the PHY interface (i) estimates the SINR of the received signal and (ii) delivers the received packets to the device. We note that no error model is currently implemented; however, considering that link adaptation is accurately modeled, we argue that the absence of an error model has a very limited impact on the overall accuracy of the model.

According to the LTE specification [3], in the time domain the radio resources are allocated every Transmission Time Interval (TTI), which has a duration of 1 ms; 10 consecutive TTIs form the LTE Frame. In our simulation model, the eNB physical layer handles the start and end of both frames and sub-frames, using the methods *StartFrame*, *StartSubFrame*, *StopFrame* and *StopSubFrame*. In particular, we note that the *StartSubFrame* function triggers both the up-

link and the downlink scheduling functions.

2.1.2 The UE physical layer

The *UeLtePhy* class models the UE physical layer. As we anticipated earlier, in an OFDMA/SC-FDMA system typically only a subset of the spectrum resources available both in DL and UL are normally assigned to an UE in a given slot. The allocation is performed by the MAC scheduler at the eNB on a per-TTI basis. To implement this per-user resource allocation, we use the *m_subChannelsForTransmission* and *m_subChannelsForReception* member variables of the *LtePhy* class. The first one describes a list of uplink sub-channels assigned by the uplink scheduler for transmitting packets through the uplink channel. The second one, instead, describes a list of downlink sub-channels assigned by the downlink packet scheduler for the downlink packet transmission. These variables are updated every TTI, according to the scheduling decisions.

When the UE device has to transmit a packet, it calls the *UeLtePhy::SendPacket()* method; this in turn delivers these packets to the *StartTx()* method of the uplink *LteSpectrumPhy* instance of the UE, which is in charge of two tasks. First, it creates the *SpectrumValue* instance, which models the signal to be transmitted by describing the PSD that the signal assumes for each sub-channel used for the transmission. To this aim, it uses the *LteSpectrumValueHelper* component, taking into account the information carried out by the *m_subChannelsForTransmission* variable, stored into the *UeLtePhy* class. Second, the UE calls the *SpectrumChannel::StartTx()* to actually simulate the transmission of the signal on the channel.

The reception procedure is performed by the *StartRx()* member method of the downlink *SpectrumPhy* object. At the end of the reception procedure, the PHY interface (i) computes the SINR of the received signal, (ii) transfers these values to the MAC layer for the creation of the CQI feedback, and finally (iii) delivers all the received packets to the device.

2.1.3 Propagation Loss Model

The propagation loss model proposed for the E-UTRAN interface is made up by four different components: path loss, shadowing, multipath and penetration loss [8], which are implemented respectively using the classes *PathLossModel*, *JakesFadingLossModel*, *ShadowingLossModel*, and *PenetrationLossModel*. All of these classes inherit from the *DiscreteTimeLossModel* class which provides methods and member variables common to all propagation loss models. Due to the time dependent channel nature, all models should be updated periodically. In particular, the *m_samplingPeriod* variable, defined into the *DiscreteTimeLossModel* class, describes the interval between two consecutive updates of the model. We note that this value should be set taking into consideration the coherence time of the channel.

The fast fading model has been implemented using the Jakes model [16] for Rayleigh fading, taking into account the frequency f , the user speed s , and the number m of multiple paths. In order to reduce the computational load during simulations, we chose to use pre-computed realizations of the Jakes model for $p \in \{6, 8, 10, 12\}$, $s \in \{0, 3, 30, 120\}$ km/h and f the central frequency of the j -th sub-channel defined according to the spectrum model that we described in Sec-

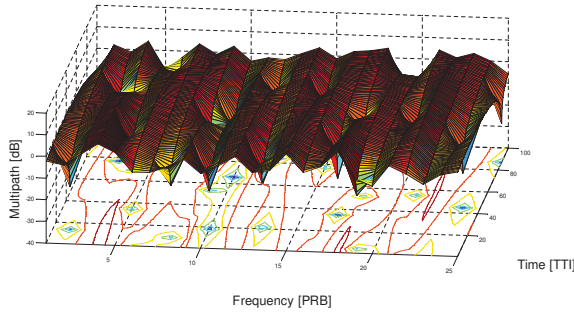


Figure 2: Time-Frequency Fast Fading Realization, UE speed = 3 km/h.

tion 2.1. During simulation, m is randomly drawn according to the uniform distribution [23], and s is inferred from the ns-3 mobility model. Figure 2 shows in time and frequency one of the pre-computed multipath realizations obtained for a user speed of 3 km/h and 6 paths. The path loss, PL , is calculated according to the outdoor urban propagation loss model in [4]; in particular, the loss expressed in dB is computed as $PL = 128.1 + 37.6 \log_{10} d$, where d is the distance in kilometers between the eNB and the UE. The large scale shadowing fading is modeled using a log-normal distribution for the loss coefficient; i.e., the loss value in dB is normally distributed with a zero mean and standard deviation of 8 dB. Finally, the penetration loss has been set to a fixed value of 10 dB according to [8]. The *ChannelRealization* class implements a realization of the propagation model for a given UE-eNB pair. This class instance acts as a wrapper for the instances of all the LossModel components to be used.

The *LteSpectrumPropagationLoss* class wraps all the propagation loss model functionality for a given *SpectrumChannel*. It stores all *ChannelRealization* objects into a `std::map` container, where each realization is identified by pointers to the mobility models of the pair of nodes it refers to. Before delivering the transmitted packets to all the attached receiving PHY instances, the *SpectrumChannel* uses the *CalculatePowerSpectralDensity* () member method of the *LteSpectrumPropagationLoss* object for calculating the power spectral density of the signal to be received. Let $P_{TX,i}$ MPL_i be the transmission power and the multipath loss for the i -th sub-channel; furthermore, let PL , PNL and SW be respectively the path loss, the penetration loss and the shadowing, $\forall i$. The power $P_{RX,i}$ of the received signal for the sub-channel i , expressed in dB, is computed as follows:

$$P_{RX,i} = P_{TX,i} - PL - MPL_i - PNL - SW \quad (1)$$

2.2 Network Devices

The LTE device, modeled by the *LteNetDevice* class, implements the abstract *NetDevice* class provided by the ns-3. In addition to performing all the common functionality of an ns-3 device, it contains and manages the main entities of the E-UTRAN protocol stack, i.e., the RRC, RLC, MAC and PHY. A couple of dedicated classes, *UeLteNetDevice* and *EnbLteNetDevice*, inherit from *LteNetDevice* and implement the functionality specific of the UE and the eNB devices, respectively.

The UeNetDevice

Since most of the functionalities are in charge of the the eNB, the implementation of the UE is simple. The UE stores information about which eNB it is registered to; for this purpose, a private member variable named *m_targetEnb* has been defined. This variable should be set at the beginning of the simulation, by using the *SetTargetEnb*() method. This is used, for example, to implement the ideal control channel that we described in Section 2.1.

The EnbNetDevice

The eNB plays a crucial role in the E-UTRAN. The most important task of the eNB is the Radio Resource Management (RRM), which is performed by uplink and downlink packet schedulers. A specific eNB MAC entity has been created for the eNB, where both UL and DL packet scheduler component are defined.

To keep track of the UEs registered with an eNB as well as their characteristics, a dedicated component called UE Manager is used (a separate UE Manager instance for each eNB). For each registered UE, an UE Record is created and stored into the UE manager component; this record is also used to store information about the latest CQI feedback sent by the UE, which need to be provided to the packet scheduler.

2.2.1 The Radio Resource Management Entity

Radio Bearers

In LTE, end-to-end user data traffic flows are grouped into logical entities called End-to-End Bearers, which are identified by their common end-to-end QoS requirements and the corresponding accounting and billing policies. At the radio level, End-to-End Bearers are mapped onto Radio Bearers [17], which represent the QoS requirements to be satisfied by the Radio Resource Management for each user data flow going from the eNB to the UE and vice versa. In our simulation module, the class *RadioBearerInstance* models the Radio Bearer established between UE and eNB. The direction of a radio bearer is stored into the *m_direction* variable. Each radio bearer is identified by a socket pair, i.e., the source and destination IP address, the type of transport protocol, and the source and destination port. In order to manage this information, we chose to reuse the *IpcsClassifierRecord* class which was defined for the WiMAX and is already present in ns-3 [15].

The *BearerQoSParameters* class has been designed to represent the QoS requirements associated to each bearer. It is a data structure that holds the parameters that characterize Bearers [2], in detail: (i) *m_bearerType*, which describes the type of a radio bearer (Guaranteed Bit Rate (GBR) or Non-Guaranteed Bit Rate (Non-GBR)); (ii) the QoS Class Identifier *m_qci* which identifies the different treatment to be adopted for each bearer (e.g., it will be translated by the RRM into specific scheduling weights, admission thresholds, queue management thresholds, link layer protocol configurations, etc.), (iii) the value of the guaranteed bit rate *m_gbr*, that represents the expected bit rate for a GBR bearer and (iv) *m_mbr* that represents the minimum bit rate to provide to the GBR bearer.

The RrcEntity

According to the LTE specifications [6], the RRC entity handles (i) connection establishment and management,

(ii) broadcast of system information, (iii) radio bearer establishment/reconfiguration and management, (iv) mobility management, and (v) paging. In our implementation, the RRC entity is implemented by the *RrcEntity* class, and provides only the Radio Bearer management functionality. The method *Classify* of the RRC entity performs the classification of the packets coming from the upper layer into the corresponding Radio Bearer. This classification is based on the information provided by the class *IpcsClassifierRecord* which we described earlier.

The class *MacQueue* implements the MAC queue where all the packets coming from the application layer and that belong to a particular Radio Bearer are stored. The *MacQueue* class implements a First In First Out (FIFO) management policy. This is acceptable since all the packets in the same queue instance always belong to the same traffic class.

The interaction between the MAC and the Radio Bearer is provided by the RLC entity, which is implemented by the *RlcEntity* class. In our simulation model, the RLC entity takes care of dequeuing the packets from the MAC queue and delivering them to the lower layer. The standard defines three different modes of operation for the RLC: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM) [5]. Currently, our simulation module provides only TM, i.e., the RLC delivers all packets to the MAC layer without performing neither segmentation nor concatenation of RLC PDU and without adding any RLC header.

2.2.2 The MAC Entity

The MAC Entity, implemented by the *MacEntity* class, provides an interface between the device and the physical layer for both the UE and the eNB devices. This interface has been designed for delivering packets coming from the upper layer to the physical layer and vice versa. In particular, it is responsible for the creation of the transport block to be sent to the PHY layer by composing packets and/or fragments provided by the RLC. Moreover, the MAC entity hosts the Adaptive Modulation and Coding (AMC) module [1], which will be described in the following.

The MAC functionalities differs between the UE and the eNB. For this reason, we defined two separate *UeMacEntity* and *EnbMacEntity* classes, in order to specialize the MAC functionalities for these different type of nodes. The *EnbMacEntity* is responsible of the radio resource allocation procedure, i.e., it host both the uplink and downlink packet scheduler components. On the other hand, an important feature provided by the *UeMacEntity* is the creation of CQI feedback, described below.

The Adaptive Modulation and Coding module

The AMC module implements the AMC functionality defined in the LTE standard, i.e., it takes care of adapting the modulation and coding scheme used in the physical layer to the channel conditions faced by each user, in order to maximize the channel capacity and keep the residual bit error rate below a maximum acceptable threshold. AMC is very important in the context of simulating RRM, because it causes the available channel capacity to vary according to the variations of the channel for each user. In particular, the MAC scheduler needs to consider the behavior of AMC in order to take effective scheduling decisions.

AMC works according to the following scheme: the UE periodically measures the quality of the signal received from the eNB, and converts it into a specific representation format which is called CQI. CQI is then forwarded to the eNB, where it is exploited by the AMC module to select, for each UE, the most suitable modulation and coding scheme [12].

The AMC component is implemented by the *MacEntity* class and it is used differently by UE and eNB. In general, the AMC module provides a unique mapping among the CQI, the MCS, the spectral efficiency, and the Transport Block (TB) Size. We chose to use the mapping function proposed in [9] and illustrated in Table 1. The spectral efficiency refers to the information rate that can be transmitted over a given sub-channel. The TB Size refers to the amount of bits at the physical layer, coming from the MAC layer during transmission (or passed to the MAC layer during reception) on transport channels; it depends on the MCS chosen by the AMC module, the number of antenna ports, the duration of the prefix code used at physical layer, and the number of symbols used by the control channel. As proposed in [9], our AMC module computes the TB size by using this configuration: normal prefix code, 2 antenna ports, 3 OFDM symbols for PDCCH (Physical Downlink Control Channel), no sync signals, and the absence of PBCH Physical Broadcast Channel.

Table 1: Mapping Function: from the spectral efficient to the TB size

Interval for $\eta_{i,j}$	CQI Index	Modulation Scheme	TB size [bit]
≤ 0.15	1	4-QAM	18
$0.15 \div 0.23$	2	4-QAM	28
$0.23 \div 0.38$	3	4-QAM	45
$0.38 \div 0.60$	4	4-QAM	72
$0.60 \div 0.88$	5	4-QAM	105
$0.88 \div 1.18$	6	4-QAM	141
$1.18 \div 1.48$	7	16-QAM	177
$1.48 \div 1.91$	8	16-QAM	230
$1.91 \div 2.40$	9	16-QAM	289
$2.40 \div 2.73$	10	64-QAM	308
$2.73 \div 3.32$	11	64-QAM	363
$3.32 \div 3.90$	12	64-QAM	433
$3.90 \div 4.52$	13	64-QAM	506
$4.52 \div 5.12$	14	64-QAM	578
≥ 5.12	15	64-QAM	640

CQI management

CQI feedback in LTE is determined according to the following procedure. The eNB sends reference symbols every TTI, spread over the whole operative downlink bandwidth. In order to estimate the channel quality, the UE uses the data collected from these reference symbols to compute the SINR of the received signal. In particular, a SINR value is obtained for each downlink sub-channel. These values are then converted into CQI values that are sent to the target eNB using the uplink control channel [12].

In the simulator, we have implemented a full-bandwidth periodic CQI reporting scheme: the UE creates a set of CQI feedbacks, one for each downlink sub-channel, every TTI. This operation is composed by three consecutive steps. First, the UE uses the PHY layer to compute a list of SINR values, one for each downlink sub-channel. Second, the

MAC entity converts these SINR values into a set of CQI indices, using some functionalities of the AMC module component. Third, an ideal control message is created and sent to the target eNB.

The exact CQI values are determined according to the procedure proposed in [19]. At the end of the physical reception procedure, the *UeSpectrumPhy* computes the SINR for the i -th sub-channel using the following equation:

$$SINR_i = \frac{SPD_{RX,i} \cdot B}{FN_0B} \quad (2)$$

where $PSD_{RX,i}$ is the PSD of the received signal for the i -th sub-channel (i.e., its value is taken from the *Spectrum-Value* instance associated with the received signal), B is the bandwidth of a resource block (in our case, $180kHz$), F is the noise figure (default value 2.5) and N_0 is the noise spectral density (default value $-174dBm/Hz$). After the SINR values have been computed, the UE PHY interface calls the *UeMacEntity::CreateCqiFeedbacks()* function to convert these SINR values to a set of CQI indices. This function first converts the SINR to the corresponding spectral efficiency η_i using Shannon's theorem [21] as follows:

$$\eta_i = \log_2 \left(1 + \frac{SINR_i}{\Gamma} \right), \quad (3)$$

where Γ is a coefficient introduced to model the difference between the theoretical bound and the performance of real modulation and coding schemes; such a coefficient depends on the target BER [22]: $\Gamma = -\ln(5 \cdot BER)/1.5$. The spectral efficiency value is then mapped to a corresponding CQI value, using the mapping function reported in Table 1.

A set of CQI values are used to create an ideal CQI feedback control messages that the UE sends to the eNB through the ideal control channel. As soon the eNB receives this control message from a given UE, it updates the channel quality report for this user, which is stored into the corresponding UE record.

Packet Scheduler

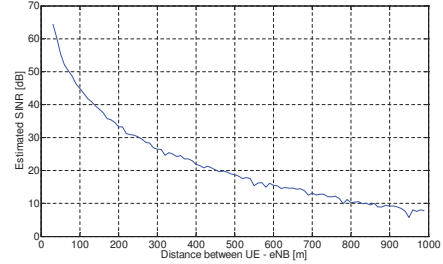
The proposed module supports both uplink and downlink schedulers. The virtual class *PacketScheduler* define the interface for the implementation of MAC schedulers (we note that a similar approach was followed for the ns-3 WiMAX module [15]). Therefore, custom scheduling strategies should be implemented by inheriting from this class. The implemented downlink and uplink schedulers are then plugged into the eNB, by setting the *m_downlikScheduler* and *m_uplinkScheduler* variables of the *EnbMacEntity* component, respectively. At the present, only a simple downlink scheduling strategy has been implemented in the class *SimplePacketScheduler*. This strategy consists of sending every TTI one packet from each active downlink Radio Bearer. These packets are included in the packet burst that is sent to the physical layer. Note that, in the case where there are no data packets to send, the eNB sends an empty packet burst which simulates the transmission of reference signals.

Finally, no uplink scheduling strategies have been developed. Still, UEs can send packets (routing messages, MAC control messages, TPC ACKs, application packets, etc.) over the uplink channel: a packet is forwarded to the physical layer and sent over the uplink channel as soon it is received by the device.

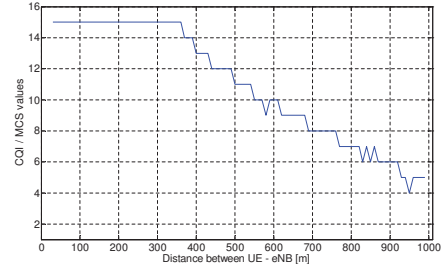
MAC Control Messages

The class *IdealControlMessage* provides support for the simulation of generic control messages to be sent over the ideal control channel. This class is to be extended in order to implement any specific type of control messages. For example, to send CQI feedback as well as PDCCH control information, two dedicated classes, namely *CqiControlMessage* and *PdcchControlMessage*, have been developed.

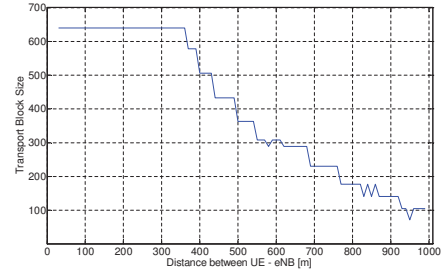
2.3 Example scenario



(a)



(b)



(c)

Figure 3: Estimated SINR (a), Chosen CQI (b), Selected TB size (c) vs distance between UE and eNB.

In this section we discuss an LTE simulation setup which shows how the implemented propagation loss model works². The scenario is composed by two nodes: a eNB and a single UE (registered to the eNB). The UE moves into the cell using the *ns3::ConstantVelocityMobilityModel*, along a radial direction. As for the available radio resources, we use a bandwidth of $10MHz$ for both the uplink and the downlink. The proposed example describes how the channel quality decreases as the distance between UE and eNB

²The considered scenario is implemented into the *lte-amc.cc* file present into the ns-3-lte branch.

increases. We note that we have analyzed what happens for the central sub-channel, but identical considerations hold for all the other sub-channels. Figure 3 shows how the estimated SINR, the chosen CQI value and the selected TB size decrease when the UE leaves from the eNB. In other words, the total bit rate (in bits per TTI) available to the UE decreases as the distance between nodes increases, as expected.

3. CONCLUSIONS

In this paper, we presented a novel module for the simulation of the LTE technology with the ns-3 simulator. Due to high complexity of the LTE technology, the module does not include all the functionalities defined in the standard. However, with the features implemented in this module (i.e., a basic implementation of LTE devices, RRC, RLC, and MAC entities, PHY layer, and propagation loss models) it is already possible to evaluate several aspects of the LTE networks, such as RRM and scheduling algorithms, link adaptation techniques, and downlink physical transmission. For the near future, we plan to add several enhancements, like uplink and downlink scheduling strategies, inter-cell interference and physical error models, RLC-level ARQ and MAC-level HARQ.

4. ACKNOWLEDGMENTS

This module has been developed as part of the Google Summer of Code (GSoC) 2010, to which the ns-3 project was participating as a mentoring organization.

5. REFERENCES

- [1] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification*, 3GPP TS 36.321.
- [2] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Overall description; Stage 2*, 3GPP TS 36.300.
- [3] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*, 3GPP TS 36.211.
- [4] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios*, 3GPP TS 36.942.
- [5] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification*, 3GPP TS 36.322.
- [6] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*, 3GPP TS 36.331.
- [7] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception*, 3GPP TS 36.101.
- [8] 3GPP. *Physical layer aspect for evolved Universal Terrestrial Radio Access (UTRA)*, 3GPP TS 25.814.
- [9] 3GPP. *Tech. Specif. Group Radio Access Network; Conveying MCS and TB size via PDCCH*, 3GPP TSG-RAN WG1 R1-081483.
- [10] S. Ascent. 3GPP LTE toolbox and blockset. [OnLine] Available: http://www.steepestascent.com/content/default.asp?page=s2_10.
- [11] N. Baldo and M. Miozzo. Spectrum-aware channel and phy layer modeling for ns3. In *Proc. of ICST VALUETOOLS*, Pisa, Italy, Oct. 2009.
- [12] E. Dahlman, S. Parkvall, J. Skold, and P. Beming. *3G Evolution HSPA and LTE for Mobile Broadband*. Academic Press, 2008.
- [13] D. Gonzalez, S. Ruiz, M. Garcia-Lozano, J. Olmos, and A. Serra. System level evaluation of LTE networks with semidistributed intercell interference coordination. In *Proc. of IEEE PIRMC 2009*, Tokyo, Japan, Sep. 2009.
- [14] J. C. Ikuno, M. Wrulich, and M. Rupp. System level simulation of LTE networks. In *Proc. of IEEE Veh. Technol. Conf., VTC Spring*, Taipei, Taiwan, May. 2010.
- [15] M. A. Ismail, G. Piro, L. A. Grieco, and T. Turletti. An improved IEEE 802.16 WiMAX module for the ns-3 simulator. In *Proc. of SIMUTools 2010*, Mar. 2010.
- [16] W. C. Jackes. *Microwave Mobile Communications* New York. Wiley, 1975.
- [17] F. Khang. *LTE for 4G Mobile Broadband, Air Interface Technologies and Performance*. Cambridge University Press, 2009.
- [18] mimoOn. mi!Mobile. [OnLine] Available: <http://www.mimoon.de/pages/Products/miMobile/>.
- [19] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda. A Two-level Scheduling Algorithm for QoS Support in the Downlink of LTE Cellular Networks. In *Proc. of European Wireless, EW2010*, Apr. 2010.
- [20] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda. Simulating LTE cellular systems: an open source framework. *IEEE Trans. Veh. Technol.*, 2010. to be published.
- [21] J. Proakis. *Digital Communications, IV Edition*. McGraw-Hill, 2002.
- [22] H. Seo and B. G. Lee. A proportional-fair power allocation scheme for fair and efficient multiuser OFDM systems. In *Proc. of IEEE GLOBECOM*, Dallas, TX, USA, Dec. 2004.
- [23] C. Wei, Z. Lili, and H. Zhiyi. Second-order statistics of improved Jakes' models for rayleigh fading channels wireless communications. In *Proc. of Int. Conf. on Networking and Mobile Computing, WiCom*, pages 1108 – 1111, Shanghai, China, Sep. 2007.