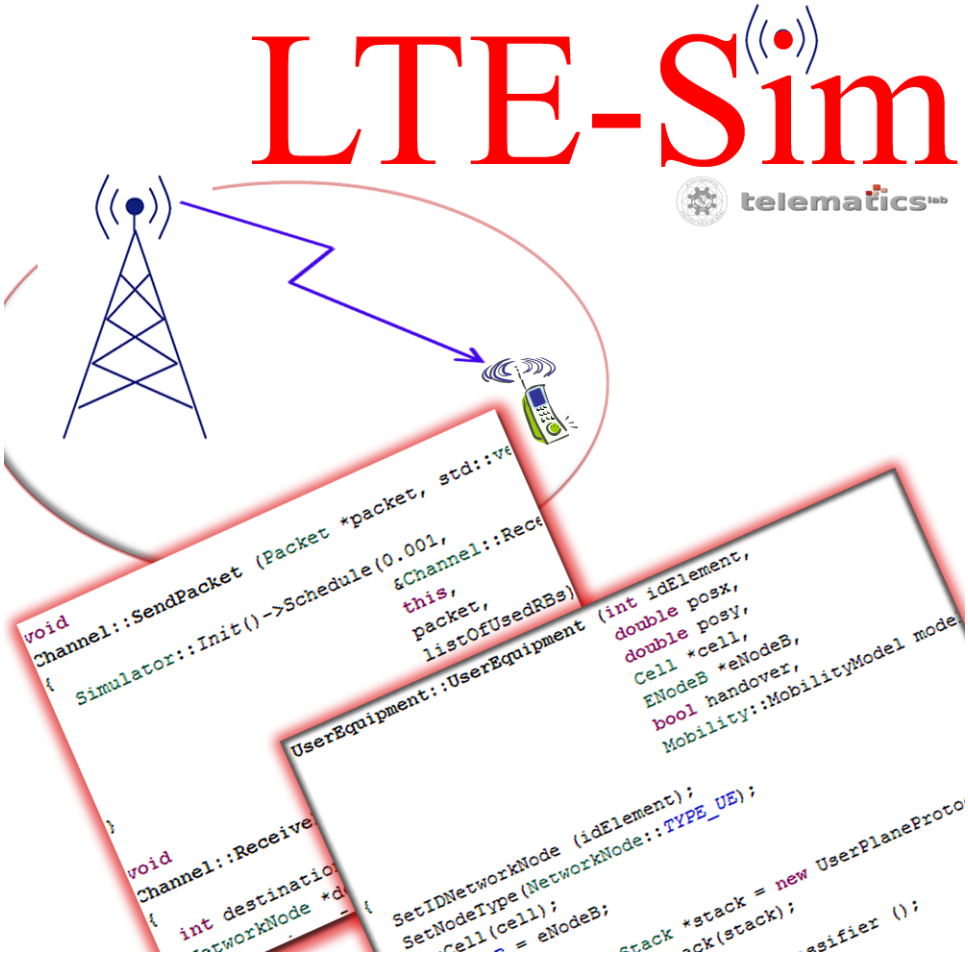

LTE-Sim is an open source framework to simulate LTE networks.

It encompasses several aspects of LTE networks, including both the Evolved Universal Terrestrial Radio Access (E-UTRAN) and the Evolved Packet System (EPS). It supports single and multi-cell environments, QoS management, multi users environment, user mobility, handover procedures, and frequency reuse techniques. Three kinds of network nodes are modeled: user equipment (UE), evolved Node B (eNB) and Mobility Management Entity/Gateway (MME/GW). Four different traffic generators at the application layer have been implemented and the management of data radio bearer is supported. Finally, well-known scheduling strategies (such as Proportional Fair, Modified Largest Weighted Delay First, and Exponential Proportional Fair), AMC scheme, Channel Quality Indicator feedback, frequency reuse techniques, and models for physical layer have been developed.

The purpose of this tutorial is to make it easier for new users to use LTE-Sim, to create their own simulation scenarios and to eventually add new functionality to LTE-Sim.

1 Getting LTE-Sim

LTE-sim is available via Subversion To obtain LTE-Sim, enter into the your preferred folder and write the following syntax:

```
$ svn co http://telematics.poliba.it/svn/LTE-Sim
```

To synchronize the project repository with the local copy, you can run update sub-command. The syntax is as follows:

```
$ svn update
```

2 Compiling LTE-Sim

On Linux systems, you can buil LTE-Sim with the following command:

```
$ make
```

To clear the project, you can use the following command:

```
$ make clean
```

For both Linux and Windows systems, you can import the LTE-Sim into the your preferred development platform (i.e., eclipse). To this aim, you can create a C++ project and copy/paste all files of LTE-Sim/src into the src folder of your project.

3 Running LTE-Sim

In this release several LTE scenarios have been developed as an example. To run a simple simulation, you can use the following command:

```
$ ./LTE-Sim Simple
```

Using

```
$ ./LTE-Sim -h
```

you can take a look to LTE scenarios we have developed as an example.

4 Software Design

5 Build a simple scenario

With LTE-Sim, a LTE scenario can be created as a static function in a C++ header file, that should be stored into the Simulation/Scenarios folder. A reference of this function should be added into

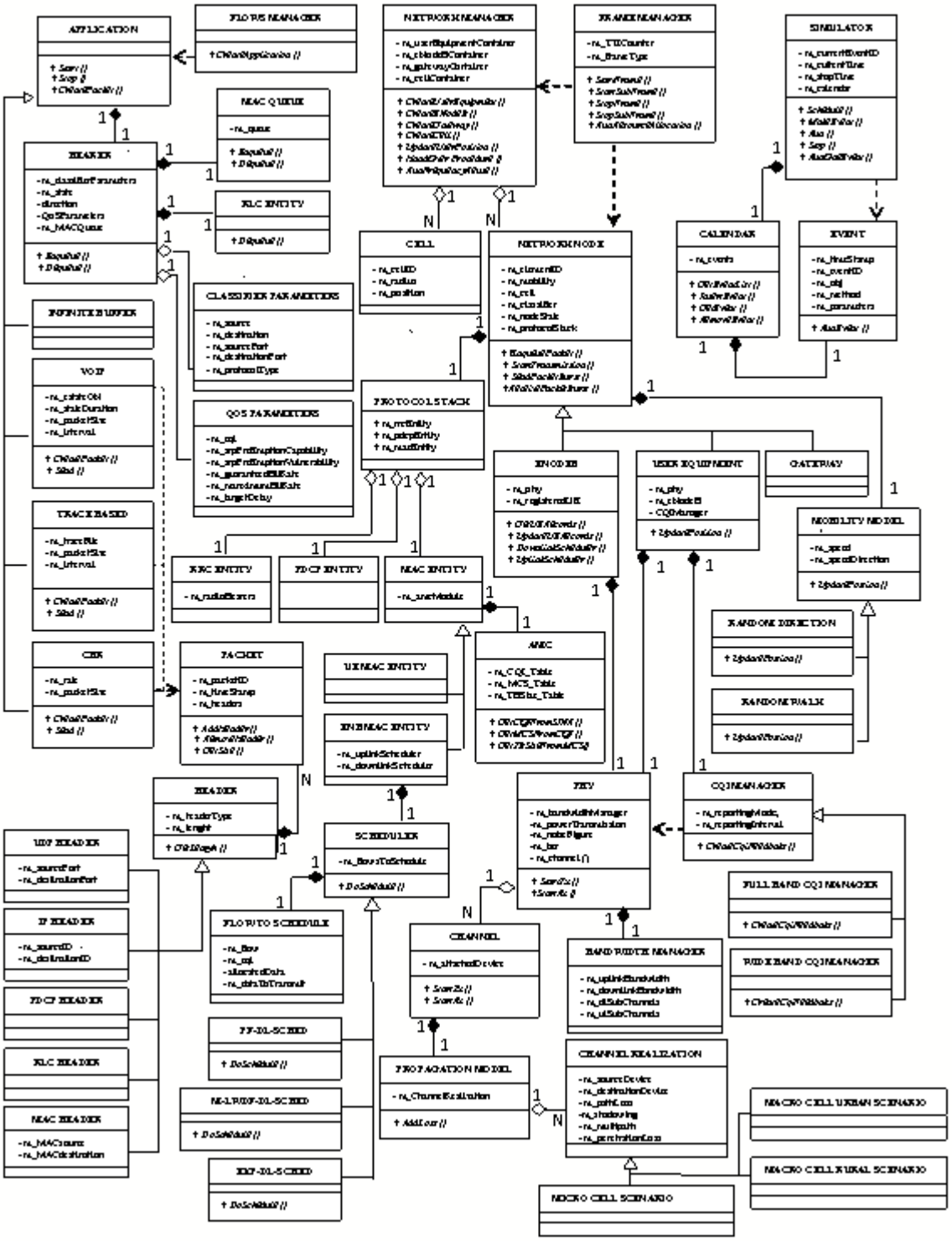

Figure 1: LTE-Sim[3]- the class diagram

the main program. In this way, the user is able to simulate a proper LTE scenario, selecting it directly from the main program.

A basic scenario can be created using the following guidelines:

- create an instance for *Simulator*, *NetworkManager*, *FlowsManager*, and *FrameManager* components.
- Create *Cell*, *ENodeB*, and *UE* objects using methods of the *NetworkManager* class. For each of these objects, several parameters can be assigned directly with the constructor of the class.
- Create applications, defining for each of them the data radio bearer type (GBR or non-GBR), IP classifier parameters, the start time, the stop time, and QoS parameters.
- Define the duration of the simulation and, finally, call the *Simulator::Run()* function.

In order to build a simple scenario, the first step is to define a new static function into the src/Simulation/SCENARIOS folder. For this tutorial, we call this function "SimpleScenario", and we define it into the SimpleScenario.h header file.

```
#include "../channel/LteChannel.h"
#include "../core/spectrum/bandwidth-manager.h"
#include "../networkTopology/Cell.h"
#include "../core/eventScheduler/simulator.h"
#include "../flows/application/InfiniteBuffer.h"
#include "../flows/QoS/QoSParameters.h"
#include "../componentManagers/FrameManager.h"
#include "../componentManagers/FlowsManager.h"

static void SimpleScenario ()
{
}
```

From this point, all instructions must be inserted into the "" of the previous declarated static function.

Create four basic LTE-Sim components (the *NetworkManager*, the *FramekManager*, the *FlowManager*, and the *Simulator*).

```
Simulator *simulator = Simulator::Init();
FrameManager *frameManager = FrameManager::Init();
NetworkManager* networkManager = NetworkManager::Init();
FlowsManager* flowsManager = FlowsManager::Init ();
```

Create Channels and Spectrum

```
LteChannel *dlCh = new LteChannel ();
LteChannel *ulCh = new LteChannel ();
BandwidthManager* spectrum = new BandwidthManager (5, 5, 0, 0);
```

Create an LTE cell.

```
// CREATE CELL
int idCell = 0;
int radius = 1; //km
int minDistance = 0.0035; //km
int posX = 0;
int posY = 0;
Cell* cell = networkManager->CreateCell (
              idCell, radius, minDistance, posX, posY);
```

Create network elements (eNB, GW, and UE).

```
//Create ENodeB
int idEnb = 1;
ENodeB* enb = networkManager->CreateEnodeb (
              idEnb, cell, posX, posY, dlCh, ulCh, spectrum);
enb->SetDLScheduler (ENodeB::DLScheduler_TYPE_PROPORTIONAL_FAIR);

//Create GW
Gateway *gw = networkManager->CreateGateway ();

//Create UE
int idUe = 2;
int posX_ue = 40; //m
int posY_ue = 0;  //m
int speed = 3;    //km/h
double speeDirection = 0;
UserEquipment* ue = networkManager->CreateUserEquipment (
              idUe, posX_ue, posY_ue, speed, speeDirection, cell, enb);
```

Create an Infinete Buffer Application

```
//Create an Application
QoSParameters *qos = new QoSParameters ();
int applicationID = 0;
int srcPort = 0;
int dstPort = 100;
int stratTime = 10; //s
int stopTime = 30;  //s
Application* be = flowsManager->CreateApplication (
              applicationID,
              gw, ue,
              srcPort, dstPort, TransportProtocol::TRANSPORT_PROTOCOL_TYPE_UDP ,
              Application::APPLICATION_TYPE_INFINITE_BUFFER,
              qos,
              stratTime, stopTime);
```

Define the duration of the simulation.

```
simulator->SetStop(60);
```

Call the *Simulator::Run()* function to strat the simulation.

```
simulator->Run ();
```

Finally, a reference of SimpleScenario() should be added into the main program. To this aim, insert into the LTE-Sim.cpp main program the folowing code:

```
#include "Simulations/SCENARIOS/SimpleScenario.h"
int
main (int argc, char *argv[])
{

  ...

        if(strcmp(argv[1], "SimpleScenario")==0)
          {
            SimpleScenario();
          }
}
```