Simulation-based UMTS e-Learning Software

Florin SANDU
"Transilvania" University
Bd Eroilor nr 29A
Brasov – 500036 – RO
Tel. +40268478705
sandu@unitbv.ro

Szilárd CSEREY
Siemens PSE Romania
21 Kogalniceanu str.
Brasov – 500090 – RO
Tel. +40268409460
szilard.cserey@ siemens.com

Titus Constantin BALAN
Siemens PSE Romania
21 Kogalniceanu str.
Brasov – 500090 – RO
Tel. +40268409341
titus.balan@ siemens.com

Mihai ROMANCA
"Transilvania" University
Bd Eroilor nr 29A
Brasov – 500036 – RO
Tel. +40268478705
romanca@unitbv.ro

ABSTRACT
This paper describes a soft-switch-based mobile network simulator developed for the purpose of e-Learning. The main goal of this simulation-based e-Learning software is to show and illustrate the behavior of the specific mobile communications network elements. The case studies presented are oriented on UMTS 3GPP's Release 4 network architecture.

Categories and Subject Descriptors
C. Computer Systems Organization - C.2 Computer-Communication Networks ; D. Software - D.4 Operating Systems - D.4.4 Communications Management ; H. Information Systems - H.4.3 Communications Applications - *Information Browsers* / H.5 Information Interfaces and Presentation - H.5.1 Multimedia Information Systems - *Artificial, Augmented, and Virtual Reality* / H.5.2 User Interfaces ; I.6 Simulation and Modeling

General Terms
Measurement, Experimentation, Verification.

Keywords
3G mobile telecom, UMTS, discrete event simulation systems, protocol monitors and emulators, virtual network adapters, 3GPP Release 4

1. INTRODUCTION
The Release 4 architecture of the 3rd Generation Partnership Project in mobile telecom adds two new elements to traditional network architectures: the mobile switching center server (MSC server) and the media gateway (MGW). These network elements communicate through the so-called "Mc interface". [1][2][7][9][10]

This architecture gives better opportunities to mobile operators, as it splits the call control functionality from the call switching functionality of the mobile network. This way, the architecture becomes much more flexible and the network becomes more scalable. For the purpose of the e-Learning of such modern and complex telecom architectures, the authors integrated and completed a software environment for simulation and monitoring of specific call control and of switching, allowing visualization of messaging throughout the network and signaling between network entities.

Monitor windows allow the trainees to perform detailed studies in the same way as in the process of protocol analysis. [8]

PETRA'08, July 15-19, 2008, Athens, Greece.

2. THE SIMULATOR AS E-LEARNING SOFTWARE

2.1 Motivation
The e-Learning package was built on top the OMNeT++ discrete event simulator and it was written in C++ , using the OMNeT++ Application Programming Interfaces. [3]

In order to implement our e-Learning concept, we chose the UMTS R4 – an architecture that was not implemented in any other kind of mobile network simulator. The software package was integrated for the use of telecommunication trainees.

The simulator is not just showing the message flows between specific network nodes, but it also creates realistic packets that can be viewed using protocol monitoring tools like Wireshark / Ethereal.

A very important feature of the simulator, that brings great benefits for e-Learning, is that it can be run step-by-step, this way letting the trainee to analyze thoroughly - and the trainer to explain in more detail - every message that was sent and received. The simulation can be viewed from two perspectives: one is the architectural (that shows which type of message was exchanged between which type of network nodes) - this is specific to the OMNeT++ simulator; the other is the perspective of protocol monitoring (using the Wireshark / Ethereal or other similar tools), where every packet can be analyzed in a great detail, bit-by-bit, showing what kind of protocol is used on each layer and which are the specific parameters. [8]

2.2 How the simulation environment was created
The software was created using the OMNeT++ simulation engine and OMNeT++ APIs. [3] We have developed 7 new models for OMNeT++, each model implementing the behavior of a specific network element. In order to have a complete simulation of the UMTS R4 mobile network, we had to create models for the following mobile network elements: User Mobile Equipment, Node B, Radio Network Controller, Circuit Switched - Media Gateway, Mobile Switching Center – Server, Gateway Mobile Switching Center – Server and a generic model representing the PSTN network.

The simulation works only at the message exchange level, every node can receive specific messages and can respond to specific requests. We put emphasis on messages, message exchange and detailed call flow. [1][8][9]

This was required by our specific *"reverse engineering" approach:* we introduced into simulation realistic packets, monitored in a real, operational, state-of-the-art industrial 3G network. These can be "monitored" and analyzed by the very popular Wireshark / Ethereal protocol monitor. [5]

The way we enhanced our simulation was based on collecting some realistic messages (difficult to be synthesized off-line) - packets prerecorded by the Tektronix K1205 and K1297 protocol analyzers at the laboratories of Siemens Program and System Engineering Romania. Thus they were brought to the attention of trainees complex packets generated by real mobile communication equipment, real Circuit Switched Media Gateways and real Mobile Switching Center Servers. We created C++ software that can rebuild the messages from these real packets. The simulation environment runs in two planes, one is the OMNeT++ simulation and the other is the packet generator software that sends - in the same time with the simulation - real packets to a virtual loopback adapter which is monitored by Wireshark / Ethereal (see figure 1). The packet generator is controlled by the OMNeT++ simulation.

3. THE UMTS R4 ARCHITECTURE

With UMTS Release 4 (R4), the architecture of the core network circuit switched domain was revised radically. The circuit traffic is delivered over an internal packet-switched Internet protocol (IP) network with connections to external networks handled via media gateways (MGW). The architecture of a R4 network is given in figure 2. The architecture of the CS (Circuit-Switched) core is described by the 3GPP TS 23.205 specification named "Bearer-independent circuit switched core network", termed bearer-independent because the core network can use asynchronous transfer mode (ATM) or IP, with many different Layer 2 options. In this case, traffic entering or exiting the circuit switch domain is controlled by the MGW. This is responsible for switching the traffic within the core network domain and performing data translation between the packet-based format used within the core network and the circuit switched data transmitted on the PSTN or ISDN external network. The MGW is controlled by the mobile switching center (MSC) server, which sends control commands to the MGW, for example to establish bearers in order to carry calls across the core network. The user data (i.e. voice traffic) within the CS-CN (Circuit Switched – Core Network) domain can be carried within ATM cells (ATM adaptation layer 2; AAL2) or IP packets. [1][2][4][7][9]

4. IMPLEMENTATION OF THE SIMULATOR BASED ON OMNET++

As already mentioned in the previous paragraphs, the e-Learning package was built on top of the OMNeT++ simulator; the code was written in C++, using different APIs from OMNeT++ that allow the integration with the simulation engine. As already mentioned, OMNeT++ is an object-oriented modular discrete-event network simulator, which can be used for: traffic modeling of telecommunication networks, protocol modeling and other network related simulations. [3]

Our development can be considered mainly as "modeling", as we created models for each element of an R4 UMTS network, and implemented their behavior from the perspective of call flows. The models have been implemented and tailored to support two kind of test scenarios: a Mobile Originated Call scenario and a Mobile Terminated Call scenario - the most basic ones, that happen most frequently in a mobile communication network. [1]

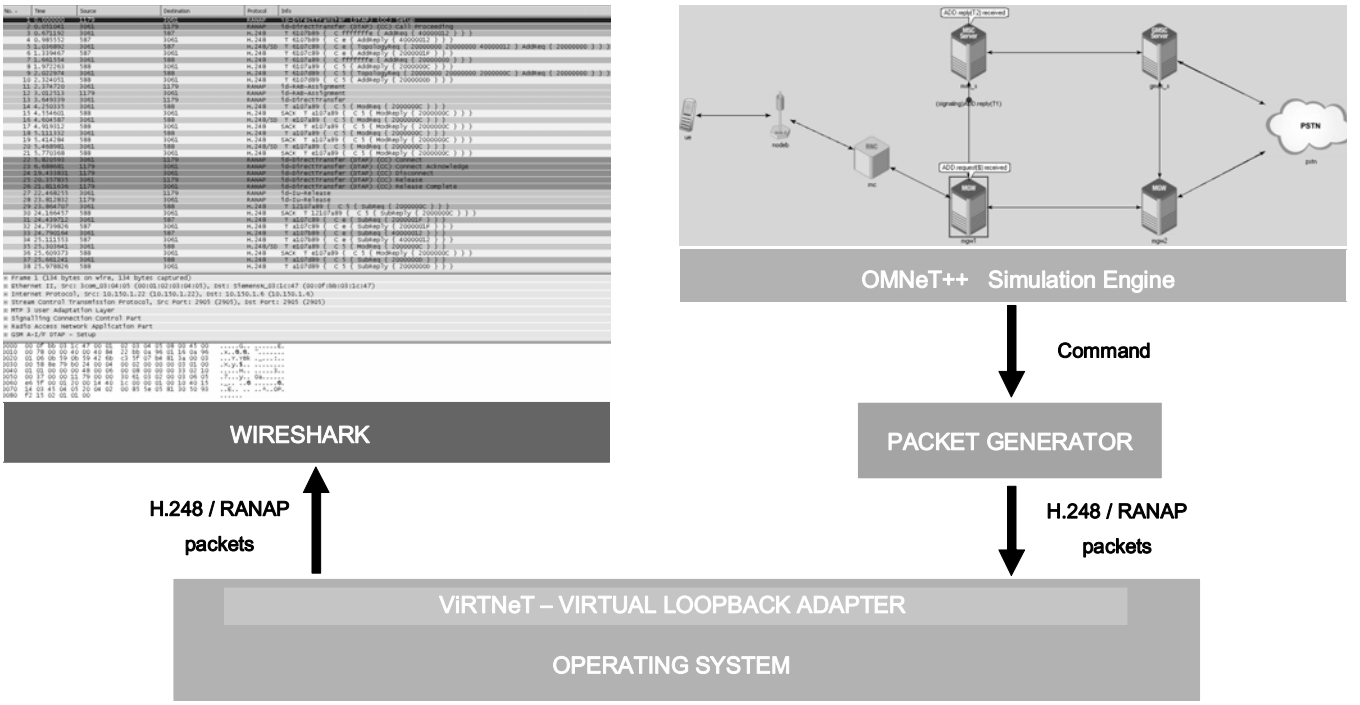

Figure 1. The software architecture of the e-Learning package

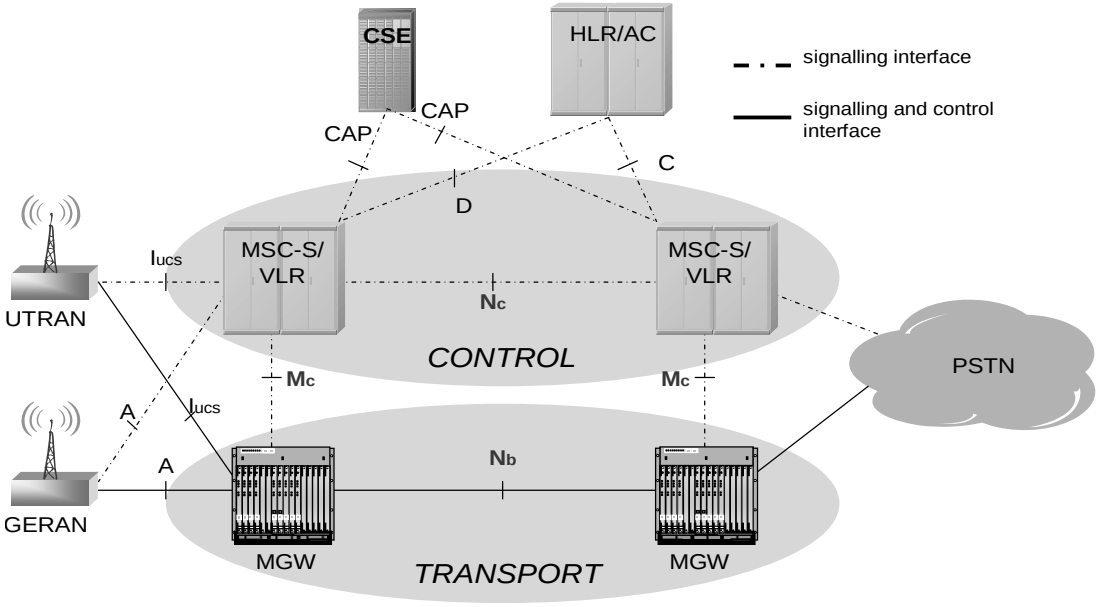

Figure 2. The architecture of the Release 4 UMTS mobile communication network

In the OMNeT++ simulator, every model has to be described at least by one class, derived from the cSimpleModule class (see fig.3). [3]

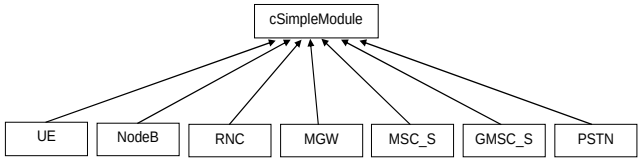

Figure 3. The class hierarchy of the models

Behavior of a model is associated to a generic state machine. This state machine describes the states of a specific network node (see fig.4).

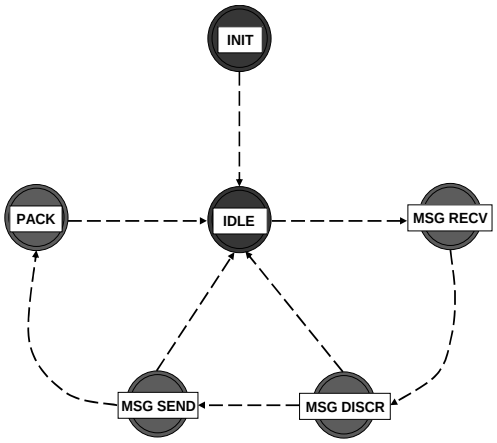

Fig. 4 The state machine associated to the network model

Every network node has an initialization state (INIT) where the variables are initiated; after this state the node enters into the waiting state (IDLE), where it waits for incoming messages. Nodes become active when they receive a message.

Every event that happens in the simulated network must be caused by the sending and the reception of a particular message. In order for the simulation to begin, a node in the initialization state must create and send a particular message to an appropriate node. There are two kinds of messages: one type of message is a message that is sent by a node to another node, or a message sent by a module to another module, and the other type of message is the self-message used for the implementation of counters and awake impulses. Because every message represents a specific event, these messages are introduced to a waiting queue, as events have to happen in a specific order. Another kind of state is the state of receiving a message (MSG RECV) – it could be message coming from another node or could be a self message. After this state the node enters into the message analyzing state, MSG DISCR (message discrimination). This interface can be monitored by the Wireshark / Ethereal network analyzer. RANAP messages are used by the MSC Server to communicate with the Radio Network Controller; these messages are forwarded by the Media Gateway to the RNC because the MSC Server is not directly interconnected with the RNC. RANAP is based on the SIGTRAN protocol stack – the Media Gateway contains an Signaling Gateway part whose role is to forward SIGTRAN messages. The SIGTRAN (Signaling Translation) protocol stack is an adaption of the SS7 protocol to the IP protocol – so that SS7 protocol messages could be transmitted through IP networks. [4][9]

The H.248 / MEGACO protocol messages are used for the communication: notification and control messages between the MSC-Server which is the *master/controller* and the Media Gateway which is the *slave*. [2]

The functionality of the nodes can be described by using SDL diagrams. SDL is a Specification and Description Language used to describe the behavior of communication systems. The SDL standard was created by ITU in the Z.100 specification.

Below, it is illustrated the SDL diagram (figure 5) and the code (figure 6) which describes the functionality of the Radio Network Controller (RNC).

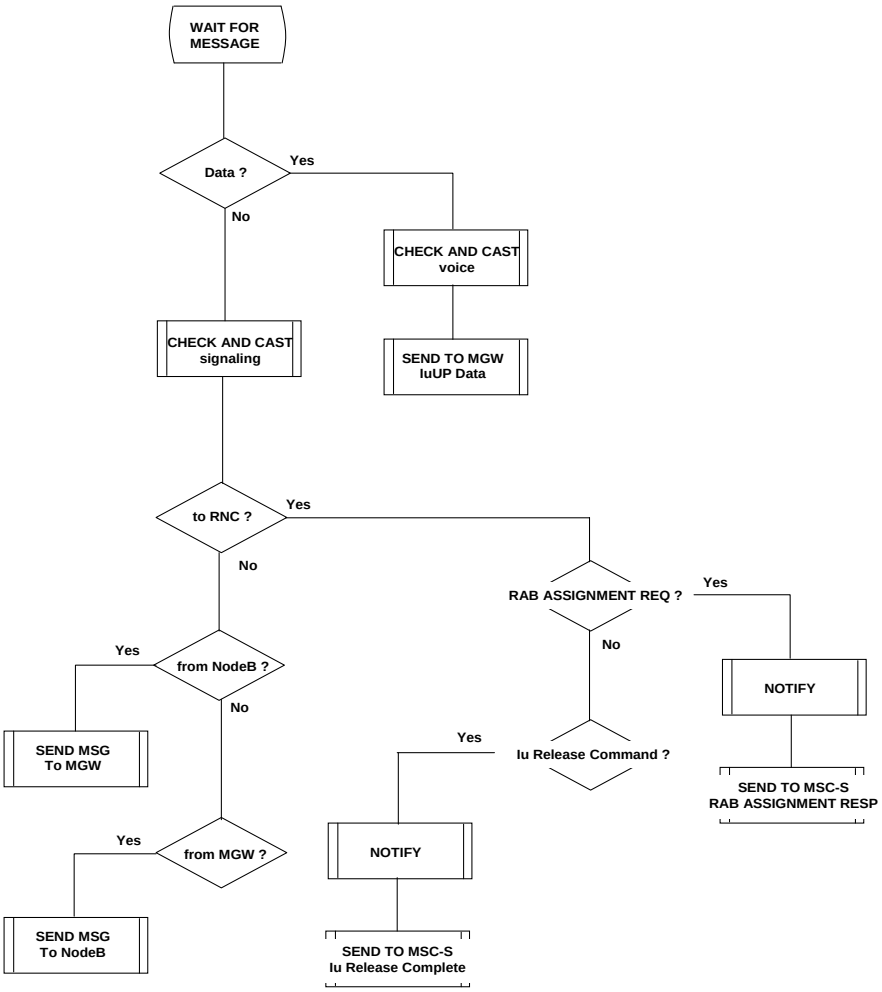

Figure 5. The SDL diagram which describes the functionality of RNC

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <omnetpp.h>
#include "signaling_m.h"
#include "voice_m.h"

class RNC: public cSimpleModule
{
      private:

      protected:

            virtual void initialize();
            virtual void handleMessage(cMessage *msg);
};

Define_Module(RNC);

void RNC::initialize()
{
}

void RNC::handleMessage(cMessage *msg)
{
      signaling *sigmsg;
      voice *vmsg;

      if(!strcmp(msg->name()," Data"))
            vmsg = check_and_cast<voice *>(msg);
      else
            sigmsg = check_and_cast<signaling *>(msg);

      cGate *arrival_gate = sigmsg -> arrivalGate();

      if (!strcmp(sigmsg->getDestination(),"RNC"))
      {
            if (!strcmp(sigmsg->name(),"RAB ASSIGNMENT REQUEST"))
            {
              delete sigmsg;
              bubble("RAB ASSIGNMENT REQ received");
              sigmsg = new signaling("RAB ASSIGNMENT RESPONSE");
              sigmsg->setSource("RNC");
              sigmsg->setDestination("MSC-S");
              send(sigmsg, "to_MGW", 0);
            }
            else if (!strcmp(sigmsg->name()," Iu Release Command"))
            {
              delete sigmsg;
              bubble("Iu Release Command received");

              sigmsg = new signaling(" Iu Release Complete");
              sigmsg->setSource("RNC");
              sigmsg->setDestination("MSC-S");
              send(sigmsg, "to_MGW", 0);
            }
      }
      else if(!strcmp(vmsg->name()," Data"))
      {
            delete vmsg;

            vmsg = new voice(" IuUP Data");
            vmsg->setSource("RNC");
            vmsg->setDestination("MGW");
            send(vmsg, "to_MGW", 0);
      }
      else if( !strcmp((arrival_gate -> fullName()) , "from_NodeB[0]"))
      {
            send(sigmsg, "to_MGW", 0);
      }
      else if ( !strcmp((arrival_gate -> fullName()) , "from_MGW[0]"))
      {
            send(sigmsg, "to_NodeB", 0);
      }
}
```

Figure 6. The code which describes the functionality of the RNC

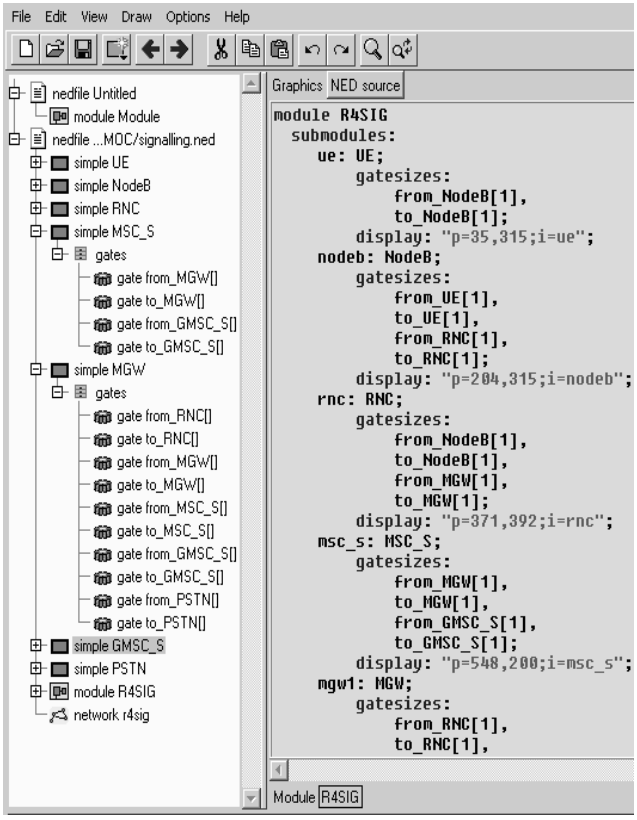

Figure 6. The Network Editor

The network nodes are behaving and communicating with each other as it is described in the 3GPP TS 23.205 specification where it can be found the call flows for MOC (Mobile Originated Call) calls and MTC (Mobile Terminated Call) calls. [1]

OMNeT++ also has a NED (Network Editor – see figure 7) which is used to describe the network topology.

The figure 7 below shows a snapshot of the OMNet++ running simulation. In figure 8 it can be seen how packets are captured and decoded by the Wireshark / Ethereal software. [5][6]

5. CONCLUSIONS

Our approach didn't cover all the message types that can occur in an UMTS network. Our aim was to assemble a basic and consistent pool of messages that are exchanged between the MSC Server and the CS-Media Gateway – these are the new network elements that make the difference between the new UMTS R4 network and the traditional GSM/UMTS network, introducing the "soft-switching" technology into mobile communications.

The specific of our implementation was the collection and reuse of prerecorded packets that are re-introduced in simulations, creating this way a new kind of realistic e-Learning software packages that could be very useful not only for university students but also for company employees, for the purpose of training (primary and updating – the so-called "delta training").

The practical work can be documented and cataloged as SCO (Shareable Content Objects) and listed in LMS (Learning Management Systems) directories for further use by teachers (that can include it in educational profiles) and/or students that can either search and discover it for vocational use or can be officially assigned to it in their learning program.

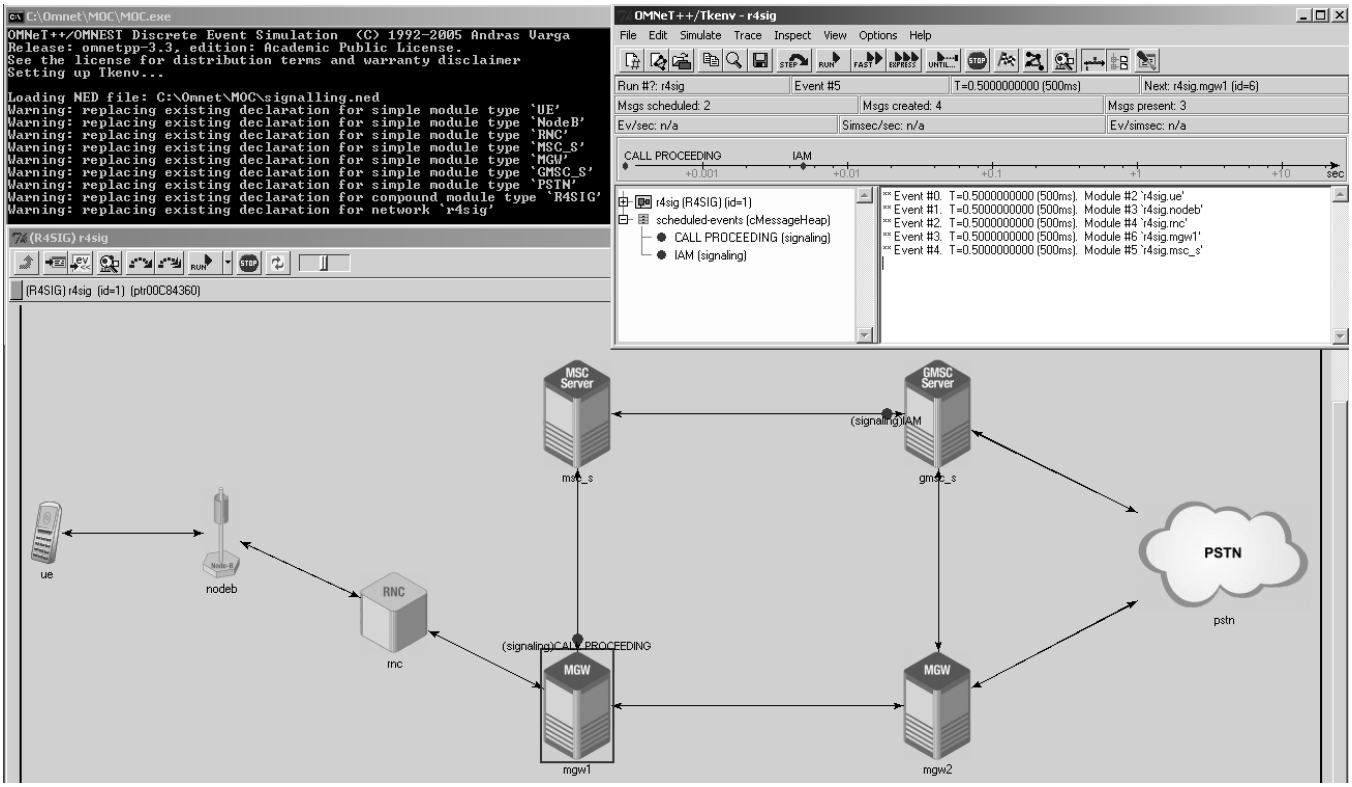

Figure 7. Snapshot of the OMNet++ simulation

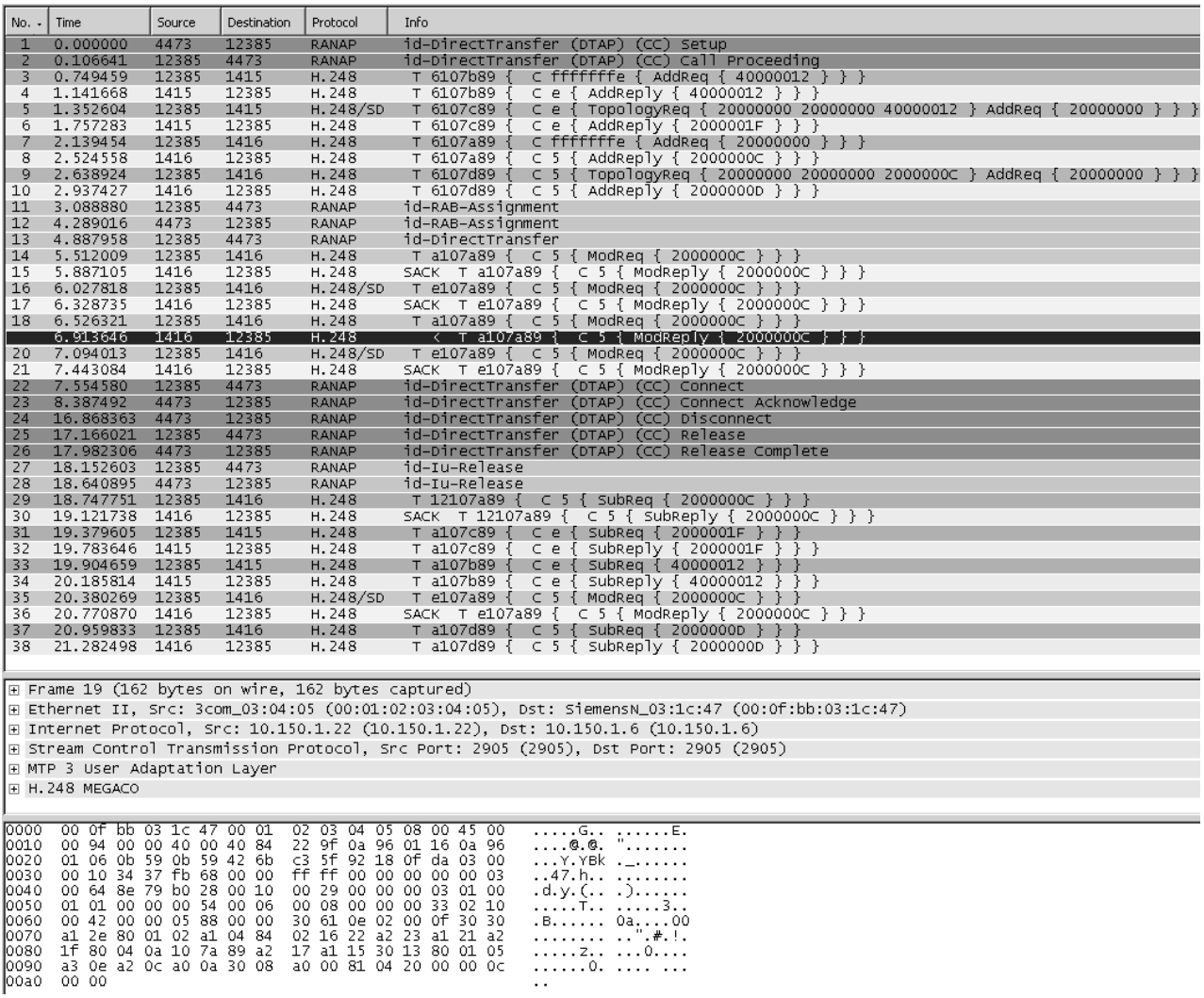

Figure 8. Packets captured and decoded by Wireshark / Ethereal

6. REFERENCES

[1] 3GPP TS 23.205 version 4.11.0 Release 4, “Universal Mobile Telecommunications System (UMTS); Bearer-independent circuit-switched core network; Stage 2” http://webapp.etsi.org /exchangefolder/ts_123205v041100p.pdf ; www.3gpp.org

[2] ITU-T H.248.1, “Gateway control protocol: Version 3” www.itu.int

[3] A.Varga: OMNeT++ Discrete Event Simulation System Version 3.2 User Manual, 2005, www.omnetpp.org

[4] J. Bannister, P. Mather and S. Coope: Convergence Technologies for 3G Networks IP, UMTS, EGPRS and ATM, John Wiley & Sons, 2004, ISBN 0-470-86091-X

[5] U. Lamping: Wireshark Developer’s Guide, 2007 http://www.wireshark.org/

[6] Virtual Network Adapter VirtNet 1.0 http://www.ntkernel.com/w&p.php?id=32

[7] J. Korhonen: Introduction to 3G Mobile Communications, Second Edition, Artech House, 2003, ISBN 1-58053-507-0

[8] R.Kreher, T.Rüdebusch: UMTS Signaling: UMTS Interfaces, Protocols, Message Flows and Procedures Analyzed and Explained, John Wiley & Sons, 2007, ISBN 978-0-470-06533-4

[9] S. Znaty: Next Generation Network (NGN) dans les réseaux mobiles, 2005, www.efort.com

[10] S. Znaty, J. L. Dauphin: Architecture NGN: Du NGN Téléphonie au NGN Multimédia, 2005, www.efort.com