MiXiM

A **mix**ed s**im**ulator framework for Omnet++

Brief overview about PHY layer architecture and channel models in MiXiM

17.4.2009

Sergio Benco sergio@newyork.ing.uniroma1.it
ACTS lab
School of Engineering
University of Rome “Sapienza”

Sergio Benco - sergio@newyork.ing.uniroma1.it

Outline:

Forthcoming project and requirements of its simulation environment

Presentation of MiXiM framework

Some details about PHY layer and Channel Models

Conclusions

Sergio Benco - sergio@newyork.ing.uniroma1.it

We are going to start a project focusing on Time Reversal technique

Our actual preferred simulation environment is Omnet++ with its well know Mobility Framework

However MF lacks in details when we are looking for:

...simulating real world obstacles and environments (walls, doors, etc)

...applying specific channel models (fading, shadowing, etc)

...doing some trasformations on the signal being transmitted (TRM)

We will now evaluate a new framework designed to offer better control on phy-related issues and channel models

MiXiM

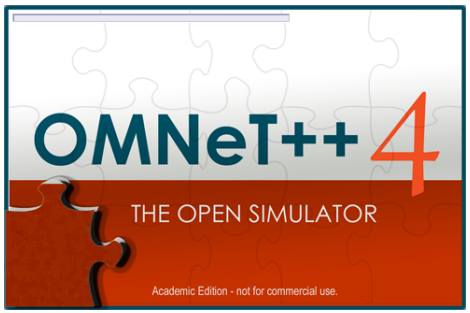
Sergio Benco - sergio@newyork.ing.uniroma1.it

MiXiM is a combination of several frameworks designed for OMNeT++ Discrete Event Simulation environment:

- Mobility Framework (MF)
- Channel Simulator (ChSim)
- MacSimulator
- Positif Framework

Project currently under development:
http://mixim.sourceforge.net

Omnet++ has reached version 4.0 (March 2009) with significative improvements since version 3.x, such as:



- An Eclipse based IDE
- Improved NED language
- Better documentation

MiXiM

Sergio Benco - sergio@newyork.ing.uniroma1.it

The features available in MiXiM are the result of experience gained developing wireless simulations in Omnet++ by many reasearch groups.

All them are now integrated into a single framework:

- General structure
- Connection management
- Mobility support

} **Mobility Framework**
http://mobility-fw.sourceforge.net/

- Radio propagation models

} **ChSim CHannel SIMulator**
(Project homepage)

- Protocol library

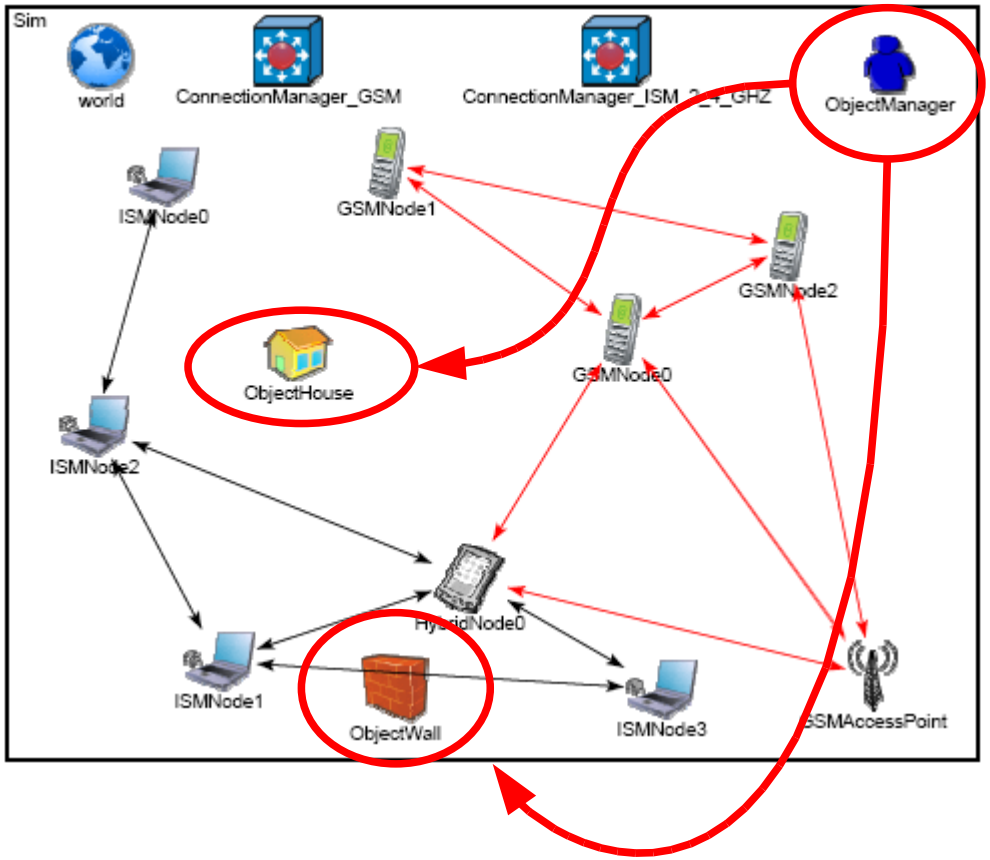
} **MAC Simulator, Positif Framework,**
Mobility Framework

MiXiM

Sergio Benco - sergio@newyork.ing.uniroma1.it

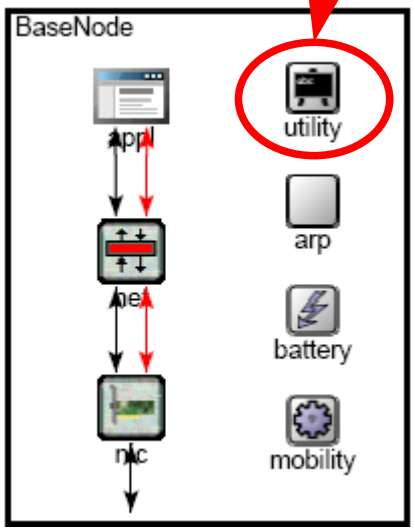
Some screenshot about MiXiM simulation GUI:

A playground example

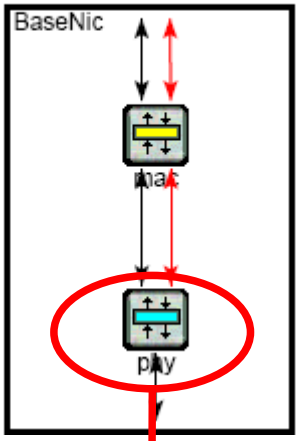


Similar to Mobility Framework Blackboard

A node



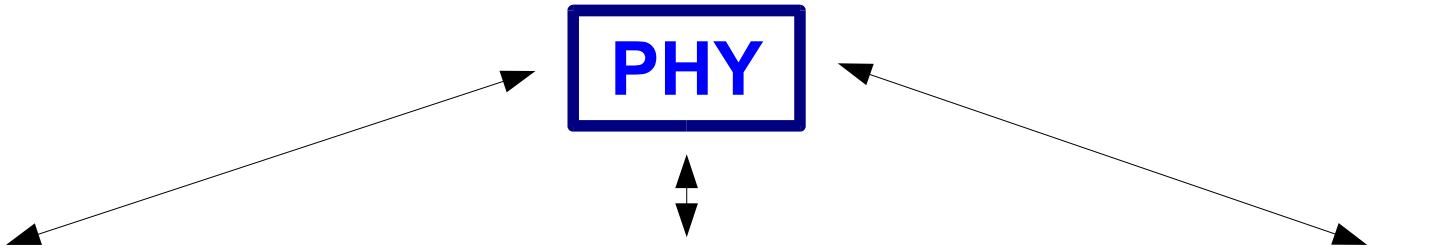
A Network Interface Card



We will now focus on PHY architecture...

Sergio Benco - sergio@newyork.ing.uniroma1.it

The physical layer in MiXiM is divided into 3 parts:



BasePhyLayer

It provides the interfaces towards its MAC and PHY layer of other nodes

AnalogueModels

Calcutates the attenuation (shadowing, fading and path loss) introduced by the channel

Decider

Evaluates the presence of useful signal

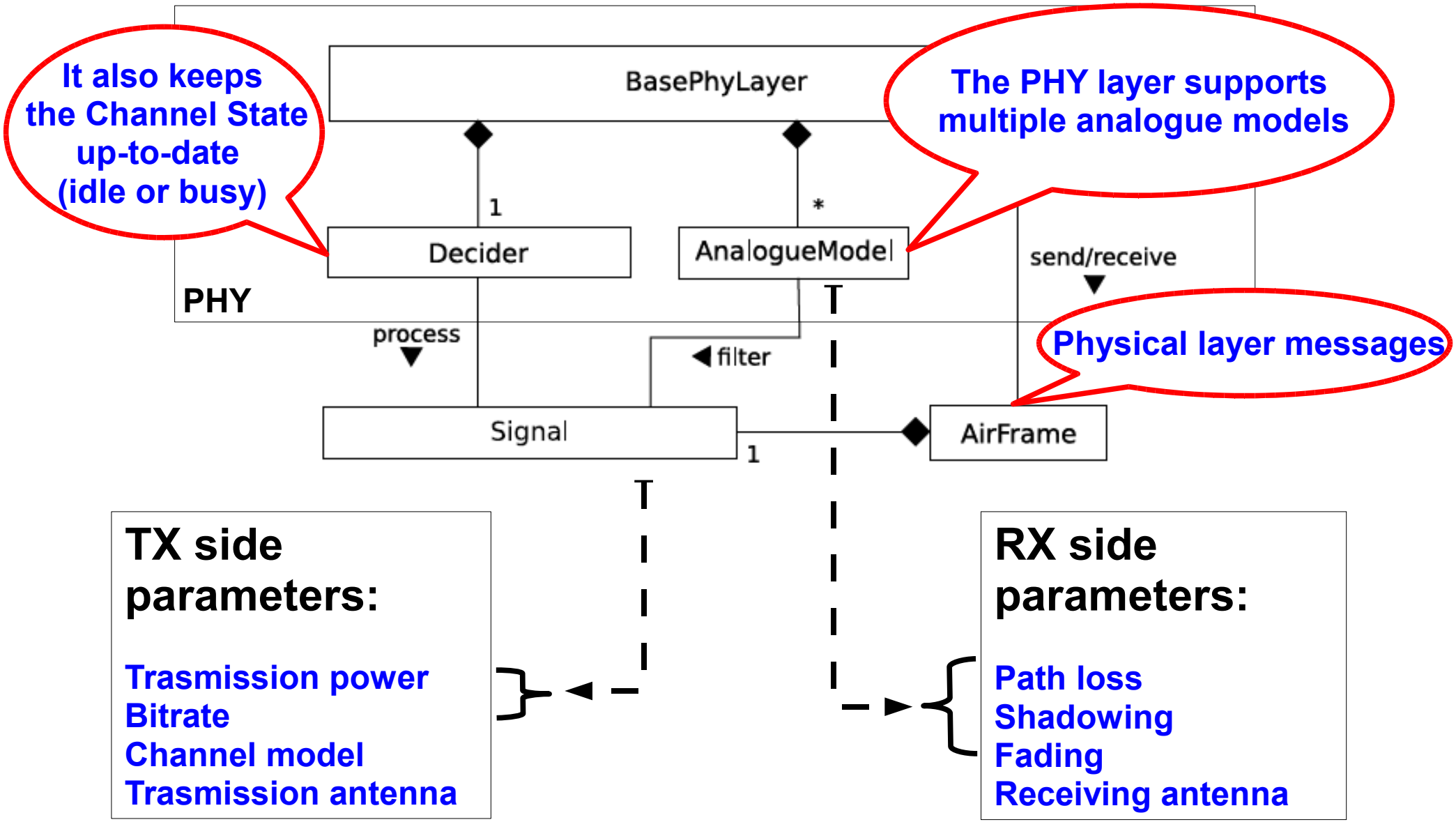
Demodulation of received messages

Designed as pure C++ classes instead of Omnet++ modules

- ✔**Clear interface**
- ✔**Avoids memory overhead**

Sergio Benco - sergio@newyork.ing.uniroma1.it

The internal structure of the Physical layer is represented in the following class graph:



Sergio Benco - sergio@newyork.ing.uniroma1.it

Inside PHY layer there are 2 other elements:

Channel Info, that keeps track of all AirFrames on the channel and offers data at the Decider to calculate SINR / RSSI

Radio State, that is responsible of switching between TX, RX, SLEEP, SWITCHING radio states

Actual PHY layer implentation interact with signals as objects and considering time, frequency and space as separate dimensions

This enables us to simulate the attenuation caused by various types of channel

Sergio Benco - sergio@newyork.ing.uniroma1.it

Analogue Models are used to reproduce the attenuation in a specific environment. These are applied to the signal by multiplication by an attenuation matrix.

The simplest example is path loss:

$$Att = \frac{\lambda^2}{(4*\pi)^2} * \left(\frac{1}{d^{\alpha}}\right)$$

```
double attenuation = 1.0;
// wavelength in metres
double wavelength = (BaseWorldUtility::speedOfLight/carrierFrequency);

if (sqrdistance > 1.0)
{
attenuation = (wavelength * wavelength) / (16.0 * M_PI * M_PI)* (pow(sqrdistance, -1.0*pathLossAlphaHalf));
}

return attenuation;
}
```

Analogue model parameters are defined in a XML file

No help from omnetpp.ini file

Analogue models are not Omnet++ models!

```
<!-- This is only a config template which shows the settings for this
AnalogueModel.
You have to define these settings in the config.xml of your PhyLayer.-->

<?xml version="1.0" encoding="UTF-8"?>
<root>
<AnalogueModels>
    <AnalogueModel type="SimplePathlossModel">
      <parameter name="alpha" type="double" value="2.0"/>
      <parameter name="carrierFrequency" type="double" value="2.412e+9"/>
    </AnalogueModel>
</AnalogueModels>
</root>
```

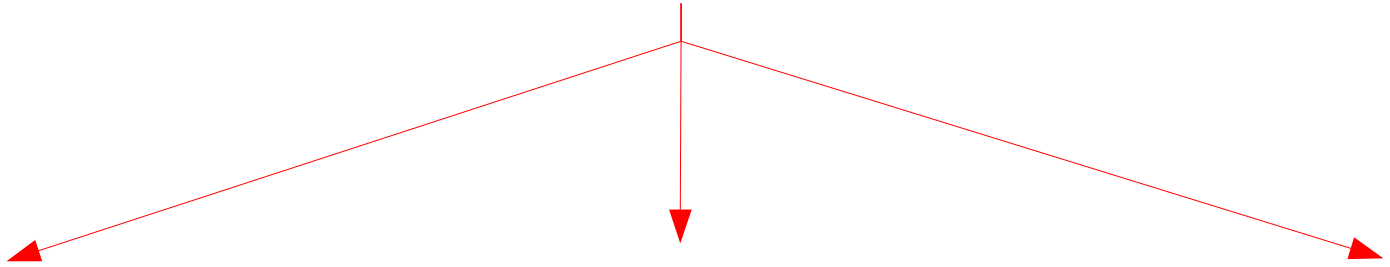
config.xml

Sergio Benco - sergio@newyork.ing.uniroma1.it

The **Decider** implemented in built-in examples of MiXiM is very simple.

If the received signal is under threshhold an output “too weak” is shown and it is discarted otherwise the output is “Strong enough” and it is processed.

A new custom Decider can be created independently from PHY layer as a C++ class



BER calculation **Signal detection in noisy channels** **Channel state register**

Sergio Benco - sergio@newyork.ing.uniroma1.it

Channel modelization issues

FEC, fast fading, slow fading, path loss and antenna gain can all be expressed as a mapping or a function like the following:

$$P_{rx} = f(t,f,s) + P_{tx} \qquad \text{(dBm)}$$

All effects on a signal are expressed as Gain

(Gain smaller than 1 for attenuation effects)

In the next slide it is represented an example of mapping. Other parameters are:

P_{tx} = 50 mw
Freq = 2.4 GHz
α = 2 (Path Loss)
Threshhold = -90 dBm
Distance TX-RX = 100 m

Channel fading is modellizable as a random process so it is reproduced by a RNG with a given seed

MiXiM

Sergio Benco - sergio@newyork.ing.uniroma1.it

Available **channel models** are:

Path Loss channel

Frequency-selective, time-invariant channel

Frequency-selective, time-variant channel

Example values:

TX Signal (RadioState=TX)

f\t	0.21	6.03
2.41	16.99	16.99
2.47	16.99	16.99

Multiplied with Attenuation(Pathloss):

f\t	0.21
0.00	-81.06

Result:

f\t	0.21	6.03
2.41	-64.07	-64.07
2.47	-64.07	-64.07

Example values:

Multiplied with Attenuation (Random time and freq attenuation):

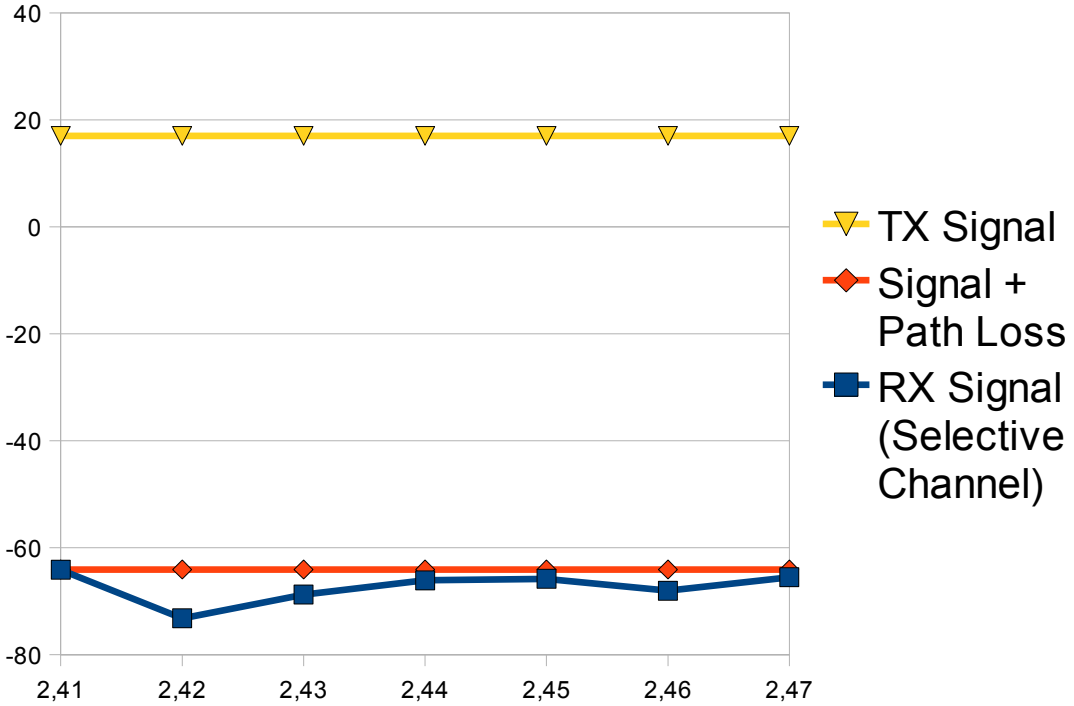
f\t	18.29	18.88	19.46	20.04	20.62	21.20	21.79	22.37	22.95	23.53	24.11
2.41	-2.67	-0.84	-8.16	-5.49	-4.57	-2.86	-0.86	-1.46	-1.47	-3.04	-2.74
2.42	-7.06	-7.51	-4.87	-3.36	-3.19	-3.05	-2.24	-5.56	-1.94	-0.47	-0.59
2.43	-0.56	-0.67	-3.31	-2.05	-1.42	-1.51	-4.44	-0.59	-2.87	-0.97	-1.57
2.44	-2.45	-0.08	-0.24	-0.06	-0.95	-1.81	-1.55	-5.24	-7.18	-0.99	-4.24
2.45	-3.80	-5.90	-1.22	-0.90	-1.22	-0.30	-3.45	-1.85	-0.93	-4.81	-2.93
2.46	-5.61	-0.68	-6.39	-0.55	-6.69	-9.93	-4.01	-0.39	-1.56	-0.62	-0.47
2.47	-1.92	-0.69	-1.55	-5.19	-2.58	-0.46	-4.04	-4.59	-5.60	-1.48	-2.97

MiXiM

Sergio Benco - sergio@newyork.ing.uniroma1.it

Channel models

(plotted 18.29 seconds after simulation start)



Example values:

TX Signal (RadioState=TX)

f\t	0.21	6.03
2.41	16.99	16.99
2.47	16.99	16.99

Multiplied with Attenuation(Pathloss):

f\t	0.21
0.00	-81.06

Result:

f\t	0.21	6.03
2.41	-64.07	-64.07
2.47	-64.07	-64.07

Example values:

Multiplied with Attenuation (Random time and freq attenuation):

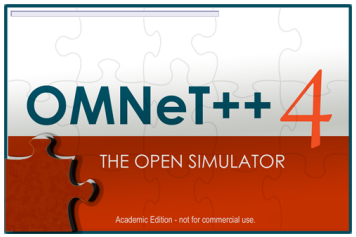
f\t	18.29	18.88	19.46	20.04	20.62	21.20	21.79	22.37	22.95	2.53	24.11
2.41	-2.67	-0.84	-8.16	-5.49	-4.57	-2.86	-0.86	-1.46	-1.47	-3.04	-2.74
2.42	-7.06	-7.51	-4.87	-3.36	-3.19	-3.05	-2.24	-5.56	-1.94	-0.47	-0.59
2.43	-0.56	-0.67	-3.31	-2.05	-1.42	-1.51	-4.44	-0.59	-2.87	-0.97	-1.57
2.44	-2.45	-0.08	-0.24	-0.06	-0.95	-1.81	-1.55	-5.24	-7.18	-0.99	-4.24
2.45	-3.80	-5.90	-1.22	-0.90	-1.22	-0.30	-3.45	-1.85	-0.93	-4.81	-2.93
2.46	-5.61	-0.68	-6.39	-0.55	-6.69	-9.93	-4.01	-0.39	-1.56	-0.62	-0.47
2.47	-1.92	-0.69	-1.55	-5.19	-2.58	-0.46	-4.04	-4.59	-5.60	-1.48	-2.97

Refreshing interval = 580ms

MiXiM

Sergio Benco - sergio@newyork.ing.uniroma1.it

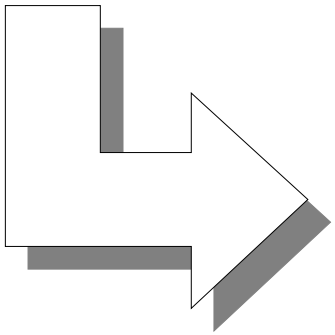
How to start working with MiXiM:

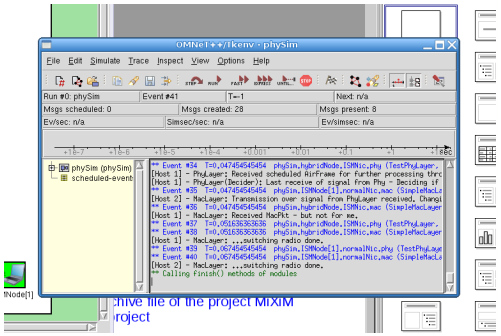


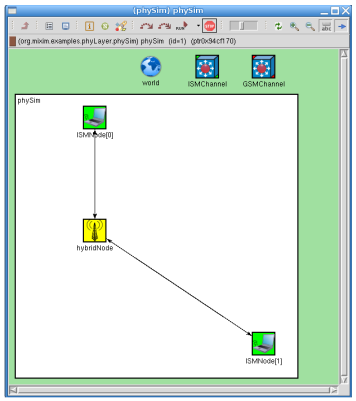
+

You have to import MiXiM archive or folder as an omnet++ project

- ✔ **Start Omnet++ IDE**
- ✔ **Click File → Import... then select “General” project → Existing Projets into Workspace**
- ✔ **Select root directory or archive file of the project MiXiM**
- ✔ **Click finish and build the project**
- ✔ **Eventually you can run an example to test your build**







Sergio Benco - sergio@newyork.ing.uniroma1.it

Conclusions:

The MiXiM Framework offers a better choice than Mobility Framework alone:

- **A more customizable PHY layer (useful to address TimeReversal requirements)**
- **Enables to explore all three dimensions of a signal (time, frequency and space)**
- **An improved channel modelization (fading, shadowing, path loss)**

Further improvements:

- **MiXiM is currently under development (a real documentation is not available yet!!)**
- **We need to work with channel impulse response but the built-in examples are too simple to verify that possibility**
- **Channel models available in Matlab have to be ported in MiXiM through Analogue Models**

Sergio Benco - sergio@newyork.ing.uniroma1.it

References:

A. Köpke, M. Swigulski, K. Wessel, D. Willkomm,
P. T. Klein Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, S. Valentin
Simulating wireless and mobile networks in OMNeT++ the MiXiM vision
Proceedings of the 1st international conference on Simulation tools and techniques
for communications, networks and systems & workshops, Marseille, France, 2008

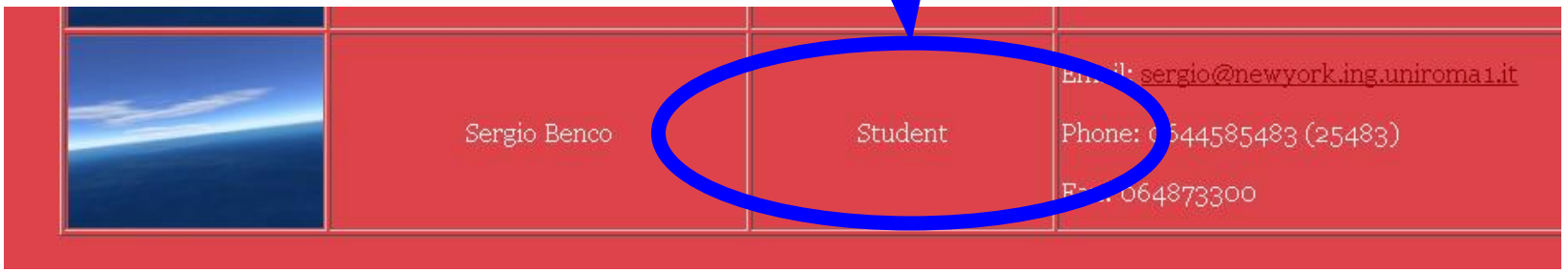
K. Wessel, M. Swigulski, A. Köpke, D. Willkomm
MiXiM The Physical Layer An Architecture Overview
2nd Int. Workshop on Omnet++, Rome, March 6th 2009

Karl Wessel, Michael Swigulski
MiXiM Physical Layer
20 November 2007

Omnet++ Discrete Event Simulation System, version 4.0, User Manual

Any questions?

(please remember I'm a simple student!)



:-) ...Thanks for your attention.