MiXiM

A mixed simulator framework for Omnet++

Brief overview about PHY layer architecture and channel models in MiXiM

17.4.2009

Sergio Benco
ACTS lab
School of Engineering
University of Rome “Sapienza”
Outline:

Forthcoming project and requirements of its simulation environment

Presentation of MiXiM framework

Some details about PHY layer and Channel Models

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

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

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

All of them are now integrated into a single framework:

- General structure
- Connection management
- Mobility support
- Radio propagation models
- Protocol library

MiXiM

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

ChSim CHannel SIMulator
(Project homepage)

MAC Simulator, Positif Framework,
Mobility Framework
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...
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**: Calculates the attenuation (shadowing, fading and path loss) introduced by the channel.
- **Decider**: Evaluates the presence of useful signal.

Designed as pure C++ classes instead of Omnet++ modules.
- Clear interface
- Avoids memory overhead
The internal structure of the Physical layer is represented in the following class graph:

TX side parameters:
- Transmission power
- Bitrate
- Channel model
- Transmission antenna

RX side parameters:
- Path loss
- Shadowing
- Fading
- Receiving antenna

The PHY layer supports multiple analogue models.

It also keeps the Channel State up-to-date (idle or busy).

Physical layer messages
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 implementation 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
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:

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

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

If the received signal is under threshold an output “too weak” is shown and it is discarded 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
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:

\[ P_{rx} = f(t,f,s) + P_{tx} \] (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:

\[ P_{tx} = 50 \text{ mw} \]
Freq = 2.4 GHz
\[ \alpha = 2 \text{ (Path Loss)} \]
Threshold = -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
Available channel models are:

Example values:

<table>
<thead>
<tr>
<th>TX Signal (RadioState=TX)</th>
<th>0.21</th>
<th>6.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41</td>
<td>16.99</td>
<td>16.99</td>
</tr>
<tr>
<td>2.47</td>
<td>16.99</td>
<td>16.99</td>
</tr>
</tbody>
</table>

Multiplied with Attenuation(Pathloss):

| 0.00 | -81.06 |

Result:

| 2.41 | -64.07 | -64.07 |
| 2.47 | -64.07 | -64.07 |

Path Loss channel
Frequency-selective, time-invariant channel
Frequency-selective, time-variant channel

Example values:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>-2.67</td>
<td>-0.84</td>
<td>-8.16</td>
<td>-5.49</td>
<td>-4.57</td>
<td>-2.86</td>
<td>-0.86</td>
<td>-1.46</td>
<td>-1.47</td>
<td>-3.04</td>
<td>-2.74</td>
</tr>
<tr>
<td>2.41</td>
<td>-7.06</td>
<td>-7.51</td>
<td>-4.87</td>
<td>-3.36</td>
<td>-3.19</td>
<td>-3.05</td>
<td>-2.24</td>
<td>-5.56</td>
<td>-1.94</td>
<td>-0.47</td>
<td>-0.59</td>
</tr>
<tr>
<td>2.42</td>
<td>-0.56</td>
<td>-0.67</td>
<td>-3.31</td>
<td>-2.05</td>
<td>-1.42</td>
<td>-1.51</td>
<td>-4.44</td>
<td>-0.59</td>
<td>-2.87</td>
<td>-0.97</td>
<td>-1.57</td>
</tr>
<tr>
<td>2.43</td>
<td>-2.45</td>
<td>-0.08</td>
<td>-0.24</td>
<td>-0.06</td>
<td>-0.95</td>
<td>-1.81</td>
<td>-1.55</td>
<td>-5.24</td>
<td>-7.18</td>
<td>-0.99</td>
<td>-4.24</td>
</tr>
<tr>
<td>2.44</td>
<td>-3.80</td>
<td>-5.90</td>
<td>-1.22</td>
<td>-0.90</td>
<td>-1.22</td>
<td>-0.30</td>
<td>-3.45</td>
<td>-1.85</td>
<td>-0.93</td>
<td>-4.81</td>
<td>-2.93</td>
</tr>
<tr>
<td>2.45</td>
<td>-5.61</td>
<td>-0.68</td>
<td>-6.39</td>
<td>-0.55</td>
<td>-6.69</td>
<td>-9.93</td>
<td>-4.01</td>
<td>-0.39</td>
<td>-1.56</td>
<td>-0.62</td>
<td>-0.47</td>
</tr>
<tr>
<td>2.46</td>
<td>-1.92</td>
<td>-0.69</td>
<td>-1.55</td>
<td>-5.19</td>
<td>-2.58</td>
<td>-0.46</td>
<td>-4.04</td>
<td>-4.59</td>
<td>-5.60</td>
<td>-1.48</td>
<td>-2.97</td>
</tr>
</tbody>
</table>

Available channel models are:

Sergio Benco - sergio@newyork.ing.uniroma1.it
### Example values:

**TX Signal (RadioState=TX)**

<table>
<thead>
<tr>
<th>f/t</th>
<th>0.21</th>
<th>6.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41</td>
<td>16.99</td>
<td>16.99</td>
</tr>
<tr>
<td>2.47</td>
<td>16.99</td>
<td>16.99</td>
</tr>
</tbody>
</table>

**Multiplied with Attenuation(Pathloss):**

<table>
<thead>
<tr>
<th>f/t</th>
<th>0.21</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>-81.06</td>
</tr>
</tbody>
</table>

**Result:**

<table>
<thead>
<tr>
<th>f/t</th>
<th>0.21</th>
<th>6.03</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41</td>
<td>-64.07</td>
<td>-64.07</td>
</tr>
<tr>
<td>2.47</td>
<td>-64.07</td>
<td>-64.07</td>
</tr>
</tbody>
</table>

---

### Example values:

**Multiplied with Attenuation (Random time and freq attenuation):**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.41</td>
<td>-2.67</td>
<td>0.34</td>
<td>-8.16</td>
<td>-5.49</td>
<td>-4.57</td>
<td>-2.86</td>
<td>-0.86</td>
<td>-1.46</td>
<td>-1.47</td>
</tr>
<tr>
<td>2.42</td>
<td>-7.06</td>
<td>-7.51</td>
<td>-4.87</td>
<td>-3.36</td>
<td>-3.19</td>
<td>-3.05</td>
<td>-2.24</td>
<td>-5.56</td>
<td>-1.94</td>
</tr>
<tr>
<td>2.43</td>
<td>-0.56</td>
<td>0.67</td>
<td>-3.31</td>
<td>-2.05</td>
<td>-1.42</td>
<td>-1.51</td>
<td>-4.44</td>
<td>-0.59</td>
<td>-2.87</td>
</tr>
<tr>
<td>2.44</td>
<td>-2.45</td>
<td>0.08</td>
<td>-0.24</td>
<td>-0.06</td>
<td>-0.95</td>
<td>-1.81</td>
<td>-1.55</td>
<td>-5.24</td>
<td>-7.18</td>
</tr>
<tr>
<td>2.45</td>
<td>-3.80</td>
<td>-5.90</td>
<td>-1.22</td>
<td>-0.90</td>
<td>-1.22</td>
<td>-0.30</td>
<td>-3.45</td>
<td>-1.85</td>
<td>-0.93</td>
</tr>
<tr>
<td>2.46</td>
<td>-5.61</td>
<td>-0.68</td>
<td>-6.39</td>
<td>-0.55</td>
<td>-6.69</td>
<td>-9.93</td>
<td>-4.01</td>
<td>-0.39</td>
<td>-1.56</td>
</tr>
<tr>
<td>2.47</td>
<td>-1.92</td>
<td>-0.69</td>
<td>-1.55</td>
<td>-5.19</td>
<td>-2.58</td>
<td>-0.46</td>
<td>-4.04</td>
<td>-4.59</td>
<td>-5.60</td>
</tr>
</tbody>
</table>

**Refreshing interval = 580ms**

---

**Graph:**

- TX Signal
- Signal + Path Loss
- RX Signal (Selective Channel)

(plotted 18.29 seconds after simulation start)
How to start working with MiXiM:

✔ 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
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
References:


*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.