MiXiM

Michael Swigulski, Karl Wessel, Daniel Willkomm

TKN OMNeT++ workshop
Berlin, September 21st 2009
Outline

- What is MiXiM?
- What can you do with MiXiM?
- Basic network in MiXiM
- Details on modeling of PHY/MAC
Outline

- What is MiXiM?
- What can you do with MiXiM?
- Basic network in MiXiM
- Details on modeling of PHY/MAC
What is MiXiM

- It's a simulation framework for OMNeT++ capable of simulating
  - Wireless networks
  - Mobile networks
  - Energy consumption
- Focus on lower layers (Layer 1 + 2)
Outline

- What is MiXiM?
- What can you do with MiXiM?
- Basic network in MiXiM
- Details on modeling of PHY/MAC
What can you do with MiXiM

- simulation of a very detailed PHY-layer and corresponding MAC-layer
  - A *Signal* class is used to represent power as well as channel effects
  - A signal as well as channel-effects (e.g. path-loss) affecting it can be represented in arbitrary dimensions (time, frequency, space)
  - Evaluation of a signal at detailed level (e.g. bit-level)
What can you do with MiXiM

- Supports energy consumption by ported Energy-Framework (written by Laura Marie Feeney)
- Mobility support from old Mobility-Framework
  - 2D and 3D
- Multiple (not interfering) channels in one network
What can you do with MiXiM

- Provides several ready-to-use modules
  - CSMA MAC-layer (ported from MF)
  - 802.11 protocol (ported from MF) with energy consumption support
  - Different channel-effects: LogNormalShadowing, SimplePathloss and Rayleigh-Fading using Jakes-model
  - Several mobility modules (ported from MF)
  - (Simple versions of ARP, Network-layer and Burst-Application-layer)
Outline

- What is MiXiM?
- What can you do with MiXiM?
- Basic network in MiXiM
- Details on modeling of PHY/MAC
The basic setup of a simulation in MiXiM

- Core components of the simulation: Hosts, BaseWorldUtility, ConnectionManager
BaseWorldUtility

- Provides utilities for the whole simulation, e.g. constants like speed-of-light as well as the playground-size
- Mandatory module
ConnectionManager

- Establishes and removes connections between hosts by evaluating their maximum interference distance
- There can be multiple ConnectionManagers (one for each non-interfering channel)
- Mandatory module
Hosts

- Consists of several components → compound module on its own
BaseHost

- Submodules: BaseUtility, ARP, mobility-module, Battery, network stack (application-layer, network-layer, NIC)
BaseUtility

- Provides utility methods for its host
- Main task is blackboard-functionality
- Mandatory module
ARP-module

- Used by Network- and MAC-layer for address-resolution
- **BaseArp** provides simple address-resolution by using host IDs
Battery-module

- Defines power-source for energy consuming devices
- SimpleBattery implements simple a linear power-consumption model
Mobility-module

- Defines current position and movement-pattern for the node
- `BaseMobility` can define static (non-moving) nodes
- Examples: `TractorMobility`, `ConstSpeedMobility`, `TurtleMobility`, ...
- Mandatory module
Network-stack

- Arbitrary setup, depends on users needs
- Listed modules are most common components
  - Application-layer: implements network use-cases
  - Network-layer
  - NIC: main focus of MiXiM, compound module on its own
BaseNIC

- Contains MAC- and PHY-layer
- Complex part of MiXiM, will be explained in detail later
- Can define the name of the ConnectionManager to be used in its parameter connectionManagerName
config.xml

- Defines the *AnalogueModels* and *Decider* for the PHY-layer to use

```xml
<root>
  <AnalogueModels>
    <AnalogueModel type="SimplePathlossModel">
      <parameter name="alpha" type="double" value="4.0"/>
      <parameter name="carrierFrequency" type="double" value="2.412e+9"/>
    </AnalogueModel>
    <AnalogueModel type="LogNormalShadowing">
      <parameter name="mean" type="double" value="0.5"/>
      <parameter name="stdDev" type="double" value="0.25"/>
      <parameter name="interval" type="double" value="0.001"/>
    </AnalogueModel>
  </AnalogueModels>

  <Decider type="Decider80211">
    <parameter name="threshold" type="double" value="0.12589254117942"/>
    <parameter name="centerFrequency" type="double" value="2.412e9"/>
  </Decider>
</root>
```

```python
baseSim.node[*].nic.phy.analogueModels = xml_doc("config.xml")
baseSim.node[*].nic.phy.decid = xml_doc("config.xml")
```
Outline

- What is MiXiM?
- What can you do with MiXiM?
- Basic network in MiXiM
- Details on modeling of PHY/MAC
Send/receive-process inside NIC
Send/receive-process: BaseMacLayer

- Packet from upper layer arrives at MAC-layer
- MAC defines TX-power and bit-rate for transmission
- Encapsulates packet from upper layer into a `MacPkt`
- Sends `MacPkt` down to PHY-layer
Send/receive-process: BasePhyLayer

- PHY-layer encapsulates *MacPkt* into *AirFrame*, together with an instance of class *Signal* which defines physical representation of the signal to transmit.
Send/receive-process: Signal

- Contains start and duration of transmission
- Contains TX-power and bit-rate as instances of class ConstMapping
- Later also contains the attenuation of the signal and therefore also the resulting RX-power
- Already initialized by MAC-layer with TX-power, bit-rate, start and duration
Excursion: Mapping

- Represents mathematical function of arbitrary domain using arbitrary resolution
- There are a lot of mapping-implementations that are all based on the same `ConstMapping`-interface
- Every mapping contains the time-dimension in its domain
Excursion: Mapping

- **Dimension**:
  - Defines a dimension by name
  - e.g. “time”, “frequency”

- **DimensionSet**:
  - A set of Dimensions used to define the domain of a mapping

- **Argument**:
  - Multi-dimensional value to define a position in a mapping's domain

- Every mapping can define an iterator which iterates over a set of points-of-interest of the mapping
Excursion: Mapping

- Mappings are evaluated using the `getValue()`-method which takes an `Argument` defining the position we are going to evaluate.
- How evaluation is done depends on the mapping's implementation of the `getValue()`-method.
Send/receive-process: BasePhyLayer

- Clone of *AirFrame* is sent to every host in range
- *AirFrame* arrives at receiving PHY-layer
- Is processed by *AnalogueModels*
Send/receive-process: AnalogueModels

- Represent channel-effects by adding attenuation as mapping to the Signal
- There can be multiple AnalogueModels in parallel (one for each channel-effect)
- RX-power is calculated by multiplying the attenuation values with TX-power
Send/receive-process: Decider

- After attenuation is added, *AirFrame* is handed to *Decider* for the first time.
- Has to decide if signal should and can be received.
- Has to decode signal and has to decide if decoding has been successful.
- To achieve this, *AirFrames* can be passed multiple times to *Decider* by PHY-layer during their transmission.
If decoding has been successful, *MacPkt* is decapsulated from *AirFrame* and sent up to the MAC-layer.
Conclusion

- MiXiM provides a very detailed NIC-simulation including channel-effects
- Provides all the things from MF (mobility patterns, 802.11 protocol, blackboard, etc.)
- Features energy consumption using the Energy-Framework from Laura Marie Feeney
Thank You

Thank you for your attention

Questions?

Comments?