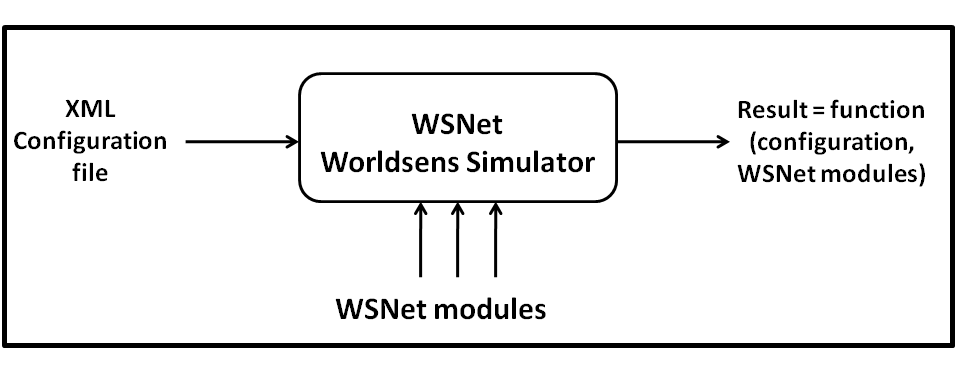
## XML files

As you can see on the following scheme, you just have to supply the simulator a XML configuration file and modules to run simulations



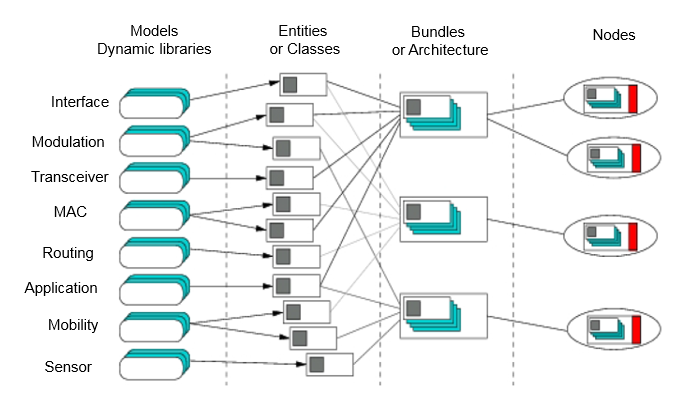
**Figure 45: XML configuration file in WSNet simulator.**

In this tutorial we will learn the basic steps necessary for setting up a configuration file. The WSNet simulator uses an xml file to configure a simulation. This file describes the simulation setup and specifies, for example, the number of nodes to simulate, the libraries used to model the radio medium and the nodes (e.g., for propagation, routing, ...). In this section, we analyze in details the architecture of a configuration file.

### Architecture

The simulation configuration file is specified using the command line option -c. Example: wsnet –c config.xml. This configuration file is an xml file describing the simulation setup. It is composed of five main sections:

* **Entities or classes** are an instantiation of dynamic libraries.
* **Mediums** define the different radio medium of the simulation.
* **Environment** defines the physical environment of the simulation.
* **Bundles or node architectures** correspond to the node architecture abstraction.
* **Global parameters or simulations** allow defining a simulation with a specific syntax.



**Figure 46:** Simulation from dynamic libraries

#### Entities

Entities or classes are an instantiation of dynamic libraries. These libraries can be instantiated several times in different entities. Each of these entities could have different global-parameter values and node-parameter default values. The parameters are name of entity, name of the dynamic library, global parameters (init) and node parameters (default).

In the XML file example in annex, we have defined the following entities:

* Medium classes
  + Pathloss, shadowing and fading for the propagation models

<pathloss class="pathloss\_xml\_name">

<c>

<param key="library" value="pathloss\_library\_name"/>

</c>

<class\_parameters>

<param key="pathloss" value="freespace"/>

<param key="other\_global\_pathloss\_parameter" value="1"/>

</class\_parameters>

</pathloss>

<shadowing class="shadowing\_xml\_name">

<c>

<param key="library" value="shadowing\_library\_name"/>

</c>

<class\_parameters>

<param key="shadowing" value="lognormal"/>

<param key="other\_global\_shadowing\_parameter" value="3.9"/>

</class\_parameters>

</shadowing>

<fading class="fading\_xml\_name">

<c>

<param key="library" value="fading\_library\_name"/>

</c>

<class\_parameters>

<param key="fading" value="rice"/>

<param key="other\_global\_parameter" value="3.9"/>

</class\_parameters>

</fading>

* + Interference for managing the interference

<interferences class="interferences\_xml\_name">

<c>

<param key="library" value="interferences\_library\_name"/>

</c>

</interferences>

* + Intermodulation for managing the intermodulation

<intermodulation class="intermodulation\_xml\_name">

<c>

<param key="library" value="intermodulation\_library\_name"/>

</c>

</intermodulation>

* + Noise for managing the level of noise on the selected band

<noise class="noise\_xml\_name">

<c>

<param key="library" value="noise\_library\_name"/>

</c>

<class\_parameters>

<param key="white-noise-dBm" value="-107dBm"/>

</class\_parameters>

</noise>

* + Modulation defines the modulation used for the calculation of the BER as a function of the SINR.

<modulation class="modulation\_xml\_name">

<c>

<param key="library" value="modulations\_library\_name"/>

</c>

</modulation>

Wsnet 4.0 shall include the following model

* + Spectrum defines the spectrum model used for the medium. When it appears is le XML file, WSNet 4.0 is automatically used.

<spectrum class="spectrum\_xml\_name ">

<cxx>

<param key="library" value="spectrum\_library\_name " />

</cxx>

</spectrum>>

* Environment classes
  + Map is useful to integrate map into the simulator or to define small walls/rooms

<map class="map\_xml\_name">

<c>

<param key="library" value="map\_library\_name"/>

</c>

<class\_parameters>

<param key="room\_width" value="5"/>

<param key="room\_length" value="5"/>

</class\_parameters>

</map>

* + Link is useful to define the type of link for propagation models according to the map

<link class="link\_xml\_name ">

<c>

<param key="library" value="link\_library\_name " />

</c>

<class\_parameters>

<param key="room\_width" value="5" />

<param key="room\_length" value="5" />

</class\_parameters>

</link>

* Node architectures classes
  + Application defines the application layer of the node.

<application class="application\_xml\_name">

<c>

<param key="library" value="application\_library\_name"/>

</c>

</application>

* + Routing defines the network layer of the node

<routing class="routing\_xml\_name">

<c>

<param key="library" value="routing\_library\_name"/>

</c>

</routing>

* + MAC defines the medium access control of the node

<mac class="mac\_xml\_name">

<c>

<param key="library" value="mac\_library\_name"/>

</c>

</mac>

* + Transceiver defines the radio used by the node. This class should be associated to a modulation entity

<transceiver class="radio\_xml\_name">

<c>

<param key="library" value="transceiver\_library\_name"/>

</c>

<class\_parameters>

<param key="modulation" value="modulation\_xml\_name"/>

</class\_parameters>

</transceiver>

* + Shared is a “shared memory” used for cross layer optimization.

<application class="shared\_xml\_name ">

<c>

<param key="library" value="shared\_info"/>

</c>

</application>

* + Interface defines the antenna of the node

<interface class="antenna\_xml\_name">

<c>

<param key="library" value="interface\_library\_name"/>

</c>

</interface>

* + Mobility defines the mobility of the node

<mobility class="mobility\_xml\_name">

<c>

<param key="library" value="mobility\_library\_name"/>

</c>

</mobility>

* + Energy defines the battery model of the node

<energy class="energy\_xml\_name ">

<c>

<param key="library" value="energy\_library\_name " />

</c>

</energy>

When Wsnet 4.0 is used some additional models must be defined to link to spectrum:

* + PHY model converts the packet into signal

<phy class="phy\_xml\_name ">

<cxx>

<param key="library" value="phy\_library\_name " />

</cxx>

</phy>

* + Signal tracker defines the capture effect of the transceiver

<signal\_tracker class="signal\_tracker\_xml\_name ">

<cxx>

<param key="library" value="signal\_tracker\_library\_name " />

</cxx>

</signal\_tracker>

* + Error could introduce error into the packet (not used yet)

<error class="error\_xml\_name ">

<cxx>

<param key="library" value="error\_library\_name " />

</cxx>

</error>

* + Interference is the interference model associated to spectrum

<interference class="interference\_xml\_name ">

<cxx>

<param key="library" value="interference\_library\_name " />

</cxx>

</interference>

* + Coding could model the coding of the packet (not used yet)

<coding class="coding\_xml\_name ">

<cxx>

<param key="library" value="coding\_library\_name " />

</cxx>

</coding>

* Simulation classes
  + Global map defines the area of the simulation. This includes x, y and z respectively the width, the length and the height of the simulation area.

<global\_map class="global\_map">

<c>

<param key="library" value="global\_map\_basic"/>

</c>

<class\_parameters>

<param key="x" value="20"/>

<param key="y" value="20"/>

<param key="z" value="0"/>

</class\_parameters>

</global\_map>

* + Monitor defines the physical events occurring in the network

<monitor class="birth\_mon">

<c>

<param key="library" value="monitor\_dummy\_monitor"/>

</c>

</monitor>

#### Mediums

**Mediums** define the different radio medium of the simulation. The parameters are the name of propagation (pathloss, shadowing and fading), the name of interferences, the name of intermodulation, the name of noise and the name of one modulation entity.

<medium name="air\_xml\_name">

<pathloss name="pathloss\_xml\_name">

</pathloss>

<shadowing name="shadowing\_xml\_name">

</shadowing>

<fading name="fading\_xml\_name">

</fading>

<interferences name="interferences\_xml\_name">

</interferences>

<intermodulation name="intermodulation\_xml\_name">

</intermodulation>

<noise name="noise\_xml\_name">

</noise>

<modulation name="modulation\_xml\_name">

</modulation>

</medium>

The definition of the medium in Wsnet 4.0 is lightly different.

<medium name="air\_xml\_name">

<spectrum name="spectrum\_xml\_name ">

</spectrum>

<pathloss name="pathloss\_xml\_name">

</pathloss>

<shadowing name="shadowing\_xml\_name">

</shadowing>

<fading name="fading\_xml\_name">

</fading>

<interferences name="interferences\_xml\_name">

</interferences>

<intermodulation name="intermodulation\_xml\_name">

</intermodulation>

<noise name="noise\_xml\_name">

</noise>

<modulation name="modulation\_xml\_name">

</modulation>

<link name="link\_xml\_name ">

</link>

</medium>

The medium must be defined after the definition of the entities.

#### Environments

**Environment** defines the physical environment of the simulation. The parameters are the name of a monitoring entity and the name of one environment entity.

<environment name="indoor">

<map name="map\_xml\_name">

</map>

</environment>

The environment must be defined after the definition of the entities.

#### Node Architectures

**Bundles or node architecture** correspond to the node architecture abstraction. After the environment and the medium, we define the bundles or the node architecture. The parameters are the name of bundle/architecture, default (to specify if this bundle is the default one or not, values true or false) and birth (optional, default value = 0).

<node\_architecture name="sensor\_node" birth="0" default="true">

…

</node\_architecture>

As default is set to true, all nodes will be defined automatically with a sensor\_node architecture.

In the implementation part, each entity used in the node architecture has to be connected from higher layer to lower layer.

In the following example, we have chained up and down entities: application to MAC, to transceiver, to interface, to medium.

For example, the transceiver layer named “radio\_xml\_name” is linked up to MAC and linked down to interface. It has also to be associated to a modulation.

The interface layer is linked up to radio but has to be associated to a medium.

<implementation>

<application name="application\_xml\_name">

<down name="mac\_xml\_name"/>

<parameters>

<param key="active" value="1"/>

</parameters>

</application>

<mac name="mac\_xml\_name">

<up name="application\_xml\_name"/>

<down name="radio\_xml\_name"/>

<parameters>

<param key="slot\_duration" value="25ms"/>

</parameters>

</mac>

<transceiver name="radio\_xml\_name">

<up name="mac\_xml\_name"/>

<down name="antenna\_xml\_name"/>

<parameters>

<param key="modulation" value="modulation\_xml\_name"/>

<param key="dBm" value="10"/>

</parameters>

</transceiver>

<interface name="antenna\_xml\_name">

<up name="radio\_xml\_name"/>

<parameters>

<param key="gain-tx" value="0"/>

<param key="gain-rx" value="0"/>

<param key="medium" value="air\_xml\_name"/>

</parameters>

</interface>

</implementation>

For WSNet 4.0, the interface layer is linked up to radio but is linked down to PHY.

The PHY layer is linked up to interface and linked down to signal tracker, interference, coding and error.

<implementation>

<application name="application\_xml\_name">

<down name="mac\_xml\_name"/>

<parameters>

<param key="active" value="1"/>

</parameters>

</application>

<mac name="mac\_xml\_name">

<up name="application\_xml\_name"/>

<down name="radio\_xml\_name"/>

<parameters>

<param key="slot\_duration" value="25ms"/>

</parameters>

</mac>

<transceiver name="radio\_xml\_name">

<up name="mac\_xml\_name"/>

<down name="antenna\_xml\_name"/>

<parameters>

<param key="modulation" value="modulation\_xml\_name"/>

<param key="dBm" value="10"/>

</parameters>

</transceiver>

<interface name="antenna\_xml\_name">

<up name="radio\_xml\_name"/>

<down name="phy\_xml\_name " />

<parameters>

<param key="gain-tx" value="0"/>

<param key="gain-rx" value="0"/>

<param key="medium" value="air\_xml\_name"/>

</parameters>

</interface>

<phy name="phy\_xml\_name ">

<up name="antenna\_xml\_name " />

<down name="signal\_tracker\_xml\_name " />

<down name="interference\_xml\_name " />

<down name="error\_xml\_name " />

<down name="coding\_xml\_name " />

</phy>

<error name="error\_xml\_name ">

<up name="phy\_xml\_name " />

</error>

<coding name="coding\_xml\_name ">

<up name="phy\_xml\_name " />

</coding>

<interference name="interference\_xml\_name ">

<up name="phy\_xml\_name " />

</interference>

<signal\_tracker name="signal\_tracker\_xml\_name ">

<up name="phy\_xml\_name " />

</signal\_tracker>

</implementation>

In this example, no routing layer have been defined.

For each layer, some local parameters can be defined.

#### Simulation

**Global parameters or simulation** allow defining a simulation with a specific syntax. These parameters are number of nodes and duration of the simulation. This simulation has also to be associated to a monitor and a global map.

<simulation nodes="10" duration="100s">

<global\_map name="global\_map">

</global\_map>

<monitor name="birth\_mon">

</monitor>

<node id="0" architecture="coordinator">

</node>

</simulation>

At the end, we define each node with their corresponding architecture.By default, nodes instantiate the node architecture set to true. If a node need to instantiates a specific architecture, the parameters are the node id and the architecture name and this will override parameters of an entity defined in the bundle.

In the example, node 0 will have the coordinator architecture.

#### Group

**Bundles or group architecture** correspond to the group architecture abstraction. It’s an optional feature. After the node architecture, we can define group architectures. Groups are used to assemble node architectures into groups, which can be especially useful when working with nodes that follow a group behavior for mobility, propagation, etc. The parameters are the name of group architecture, default (to specify if this group is the default one or not, values true or false) and type. Each group must have one leader and may contain several other members. We can set the number of nodes from each archictecture that will compose the group.

As an example, a Body Area Network may have nodes that will assemble to form a group. We may have a group formed by

<group\_architecture name="BAN" default="true" type="BAN">

<leader architecture="BAN\_coordinator"/>

<member architecture="BAN\_hand" nbr\_nodes="1"/>

<member architecture="BAN\_ear" nbr\_nodes="1"/>

</group\_architecture>

As default is set to true, all nodes of the simulation will be defined automatically assigned to a different group “BAN”. Thus, if the simulation has 18 nodes, 6 groups of type BAN will be formed. In the BAN example, a group would be formed by one BAN coordinator and also one node placed in the hand and other in the ear of the body. The number of nodes in the simulation and the number of groups must mach, meaning that defining 19 nodes for the example above would produce an error, as it is not possible to distribute the nodes in groups of 3.

#### Others information about XML file

All XML files for WSNET start and finish with worldsens instance

<?xml version='1.0' encoding='UTF-8'?>

<worldsens xmlns="http://www.cea.fr">

…

</worldsens>

All XML files accept different units or type of parameters:

* **time**: the time unit is the nano-second. However, a time parameter can be passed with a unit modifier as "s", "ms", "us" and "ns". Ex: duration="200s", birth="10s".
* **distance**: distances must be given in meters unless specified differently by the model.
* **size**: packet size must be given in Byte unless specified differently by the model.
* **Random**: Most of the parameters, i.e. distance/duration/time/destination/x/y/z, accept the "random" value. In most cases, a random value is drawn in a range chosen by the simulator as a function of the parameter type.

### Template of XML File

#### WSNET 3.1

<?xml version='1.0' encoding='UTF-8'?>

<worldsens xmlns="http://www.cea.fr">

<!-- == Classes ===================================================== -->

<!-- mediums classes -->

<pathloss class="pathloss\_xml\_name">

<c>

<param key="library" value="pathloss\_library\_name"/>

</c>

<class\_parameters>

<param key="pathloss" value="freespace"/>

<param key="other\_global\_pathloss\_parameter" value="1"/>

</class\_parameters>

</pathloss>

<shadowing class="shadowing\_xml\_name">

<c>

<param key="library" value="shadowing\_library\_name"/>

</c>

<class\_parameters>

<param key="shadowing" value="lognormal"/>

<param key="other\_global\_shadowing\_parameter" value="3.9"/>

</class\_parameters>

</shadowing>

<fading class="fading\_xml\_name">

<c>

<param key="library" value="fading\_library\_name"/>

</c>

<class\_parameters>

<param key="fading" value="rice"/>

<param key="other\_global\_parameter" value="3.9"/>

</class\_parameters>

</fading>

<interferences class="interferences\_xml\_name">

<c>

<param key="library" value="interferences\_library\_name"/>

</c>

</interferences>

<intermodulation class="intermodulation\_xml\_name">

<c>

<param key="library" value="intermodulation\_library\_name"/>

</c>

</intermodulation>

<noise class="noise\_xml\_name">

<c>

<param key="library" value="noise\_library\_name"/>

</c>

<class\_parameters>

<param key="white-noise-dBm" value="-107dBm"/>

</class\_parameters>

</noise>

<modulation class="modulation\_xml\_name">

<c>

<param key="library" value="modulations\_library\_name"/>

</c>

</modulation>

<!-- environments classes -->

<map class="map\_xml\_name">

<c>

<param key="library" value="map\_library\_name"/>

</c>

<class\_parameters>

<param key="room\_width" value="5"/>

<param key="room\_length" value="5"/>

</class\_parameters>

</map>

<!-- node architectures classes -->

<application class="application\_xml\_name">

<c>

<param key="library" value="application\_library\_name"/>

</c>

</application>

<application class="shared">

<c>

<param key="library" value="shared\_info"/>

</c>

</application>

<mac class="mac\_xml\_name">

<c>

<param key="library" value="mac\_library\_name"/>

</c>

</mac>

<transceiver class="radio\_xml\_name">

<c>

<param key="library" value="transceiver\_library\_name"/>

</c>

<class\_parameters>

<param key="modulation" value="modulation\_xml\_name"/>

</class\_parameters>

</transceiver>

<mobility class="mobility1\_xml\_name">

<c>

<param key="library" value="mobility\_library\_name"/>

</c>

</mobility>

<mobility class="mobility2\_xml\_name">

<c>

<param key="library" value="mobility\_library\_name"/>

</c>

</mobility>

<interface class="antenna\_xml\_name">

<c>

<param key="library" value="interface\_library\_name"/>

</c>

</interface>

<!-- simulation classes -->

<global\_map class="global\_map">

<c>

<param key="library" value="global\_map\_basic"/>

</c>

<class\_parameters>

<param key="x" value="20"/>

<param key="y" value="20"/>

<param key="z" value="0"/>

</class\_parameters>

</global\_map>

<monitor class="birth\_mon">

<c>

<param key="library" value="monitor\_dummy\_monitor"/>

</c>

</monitor>

<!-- == Mediums ===================================================== -->

<medium name="air\_xml\_name">

<pathloss name="pathloss\_xml\_name">

</pathloss>

<shadowing name="shadowing\_xml\_name">

</shadowing>

<fading name="fading\_xml\_name">

</fading>

<interferences name="interferences\_xml\_name">

</interferences>

<intermodulation name="intermodulation\_xml\_name">

</intermodulation>

<noise name="noise\_xml\_name">

</noise>

<modulation name="modulation\_xml\_name">

</modulation>

</medium>

<!-- == Environments ===================================================== -->

<environment name="indoor">

<map name="map\_xml\_name">

</map>

</environment>

<!-- == Node Architectures ===================================================== -->

<!-- == Sensor node architecture == -->

<node\_architecture name="sensor\_node" birth="0" default="true">

<implementation>

<application name="application\_xml\_name">

<down name="mac\_xml\_name"/>

<parameters>

<param key="active" value="1"/>

</parameters>

</application>

<mac name="mac\_xml\_name">

<up name="application\_xml\_name"/>

<down name="radio\_xml\_name"/>

<parameters>

<param key="slot\_duration" value="25ms"/>

</parameters>

</mac>

<transceiver name="radio\_xml\_name">

<up name="mac\_xml\_name"/>

<down name="antenna\_xml\_name"/>

<parameters>

<param key="modulation" value="modulation\_xml\_name"/>

<param key="dBm" value="10"/>

</parameters>

</transceiver>

<interface name="antenna\_xml\_name">

<up name="radio\_xml\_name"/>

<parameters>

<param key="gain-tx" value="0"/>

<param key="gain-rx" value="0"/>

<param key="medium" value="air\_xml\_name"/>

</parameters>

</interface>

</implementation>

<mobility name="mobility1\_xml\_name">

</mobility>

</node\_architecture>

<!-- == coordinator architecture == -->

<node\_architecture name="coordinator" birth="0" default="false">

<implementation>

<application name="application\_xml\_name">

<down name="mac\_xml\_name"/>

<parameters>

<param key="active" value="0"/>

</parameters>

</application>

<mac name="mac\_xml\_name">

<up name="application\_xml\_name"/>

<down name="radio\_xml\_name"/>

<parameters>

<param key="device\_type" value="0"/>

</parameters>

</mac>

<transceiver name="radio\_xml\_name">

<up name="mac\_xml\_name"/>

<down name="antenna\_xml\_name"/>

<parameters>

<param key="modulation" value="modulation\_xml\_name"/>

<param key="dBm" value="10"/>

<param key="sensibility" value="-100"/>

</parameters>

</transceiver>

<interface name="antenna\_xml\_name">

<up name="radio\_xml\_name"/>

<parameters>

<param key="gain-tx" value="0"/>

<param key="gain-rx" value="0"/>

<param key="medium" value="air\_xml\_name"/>

</parameters>

</interface>

</implementation>

<mobility name="mobility2\_xml\_name">

<parameters>

<param key="start\_time" value="10s"/>

</parameters>

</mobility>

</node\_architecture>

<!-- == Simulation ===================================================== -->

<simulation nodes="10" duration="100s">

<global\_map name="global\_map">

</global\_map>

<monitor name="birth\_mon">

</monitor>

<node id="0" architecture="coordinator">

</node>

</simulation>

</worldsens>

#### WSNET 4.0

<?xml version='1.0' encoding='UTF-8'?>

<worldsens xmlns="http://www.cea.fr">

<!-- == Classes ===================================================== -->

<!-- mediums classes -->

<spectrum class="spectrum\_xml\_name ">

<cxx>

<param key="library" value="spectrum\_library\_name " />

</cxx>

</spectrum>

<pathloss class="pathloss\_xml\_name">

<c>

<param key="library" value="pathloss\_library\_name"/>

</c>

</pathloss>

<shadowing class="shadowing\_xml\_name">

<c>

<param key="library" value="shadowing\_library\_name"/>

</c>

</shadowing>

<fading class="fading\_xml\_name">

<c>

<param key="library" value="fading\_library\_name"/>

</c>

</fading>

<interferences class="interferences\_xml\_name">

<c>

<param key="library" value="interferences\_library\_name"/>

</c>

</interferences>

<intermodulation class="intermodulation\_xml\_name">

<c>

<param key="library" value="intermodulation\_library\_name"/>

</c>

</intermodulation>

<noise class="noise\_xml\_name">

<c>

<param key="library" value="noise\_library\_name"/>

</c>

</noise>

<modulation class="modulation\_xml\_name">

<c>

<param key="library" value="modulations\_library\_name"/>

</c>

</modulation>

<!-- environments classes -->

<map class="map\_xml\_name">

<c>

<param key="library" value="map\_library\_name"/>

</c>

<class\_parameters>

<param key="room\_width" value="5"/>

<param key="room\_length" value="5"/>

</class\_parameters>

</map>

<link class="link\_xml\_name ">

<c>

<param key="library" value="link\_library\_name " />

</c>

<class\_parameters>

<param key="room\_width" value="5"/>

<param key="room\_length" value="5"/>

</class\_parameters>

</link>

<!-- node architectures classes -->

<phy class="phy\_xml\_name ">

<cxx>

<param key="library" value="phy\_library\_name " />

</cxx>

</phy>

<signal\_tracker class="signal\_tracker\_xml\_name ">

<cxx>

<param key="library" value="signal\_tracker\_library\_name " />

</cxx>

</signal\_tracker>

<error class="error\_xml\_name ">

<cxx>

<param key="library" value="error\_library\_name " />

</cxx>

</error>

<interference class="interference\_xml\_name ">

<cxx>

<param key="library" value="interference\_library\_name" />

</cxx>

</interference>

<coding class="coding\_xml\_name ">

<cxx>

<param key="library" value="coding\_library\_name" />

</cxx>

</coding>

<energy class="energy\_xml\_name ">

<c>

<param key="library" value="energy\_library\_name" />

</c>

</energy>

<application class="application\_xml\_name ">

<c>

<param key="library" value="application\_library\_name " />

</c>

</application>

<mac class="mac\_xml\_name ">

<c>

<param key="library" value="mac\_library\_name " />

</c>

</mac>

<transceiver class="radio\_xml\_name ">

<c>

<param key="library" value="transceiver\_library\_name " />

</c>

</transceiver>

<mobility class="mobility\_xml\_name ">

<c>

<param key="library" value="mobility\_library\_name " />

</c>

</mobility>

<interface class="antenna\_xml\_name ">

<c>

<param key="library" value="interface\_library\_name " />

</c>

</interface>

<!-- simulation classes -->

<global\_map class="global\_map">

<c>

<param key="library" value="global\_map\_basic" />

</c>

<class\_parameters>

<param key="x" value="180" />

<param key="y" value="180" />

<param key="z" value="0" />

</class\_parameters>

</global\_map>

<monitor class="birth\_mon">

<c>

<param key="library" value="monitor\_dummy\_monitor" />

</c>

</monitor>

<!-- == Mediums ===================================================== -->

<medium name="air\_xml\_name ">

<spectrum name="spectrum\_xml\_name ">

</spectrum>

<pathloss name="pathloss\_xml\_name">

</pathloss>

<shadowing name="shadowing\_xml\_name">

</shadowing>

<fading name="fading\_xml\_name">

</fading>

<interferences name="interferences\_xml\_name">

</interferences>

<intermodulation name="intermodulation\_xml\_name">

</intermodulation>

<noise name="noise\_xml\_name">

</noise>

<modulation name="modulation\_xml\_name">

</modulation>

<link name="link\_xml\_name ">

</link>

</medium>

<!-- == Environments =================================================== -->

<environment name="indoor">

<map name="map\_xml\_name ">

</map>

</environment>

<!-- == Coordinator == -->

<node\_architecture name="coordinator" birth="0" default="false">

<implementation>

<application name="application\_xml\_name ">

<down name="mac\_xml\_name " />

</application>

<mac name="mac\_xml\_name ">

<up name="application\_xml\_name " />

<down name="radio\_xml\_name " />

</mac>

<transceiver name="radio\_xml\_name ">

<up name="mac\_xml\_name " />

<down name="antenna\_xml\_name " />

</transceiver>

<interface name="antenna\_xml\_name ">

<up name="radio\_xml\_name " />

<down name="phy\_xml\_name " />

</interface>

<phy name="phy\_xml\_name ">

<up name="antenna\_xml\_name " />

<down name="signal\_tracker\_xml\_name " />

<down name="interference\_xml\_name " />

<down name="error\_xml\_name " />

<down name="coding\_xml\_name " />

</phy>

<error name="error\_xml\_name ">

<up name="phy\_xml\_name " />

</error>

<coding name="coding\_xml\_name ">

<up name="phy\_xml\_name " />

</coding>

<interference name="interference\_xml\_name ">

<up name="phy\_xml\_name " />

</interference>

<signal\_tracker name="signal\_tracker\_xml\_name ">

<up name="phy\_xml\_name " />

</signal\_tracker>

</implementation>

<energy name="energy\_xml\_name ">

</energy>

<mobility name="mobility\_xml\_name ">

</mobility>

</node\_architecture>

<node\_architecture name="node" birth="0" default="true">

<implementation>

<application name="application\_xml\_name ">

<down name="mac\_xml\_name " />

</application>

<mac name="mac\_xml\_name ">

<up name="application\_xml\_name " />

<down name="radio\_xml\_name " />

</mac>

<transceiver name="radio\_xml\_name ">

<up name="mac\_xml\_name " />

<down name="antenna\_xml\_name " />

</transceiver>

<interface name="antenna\_xml\_name ">

<up name="radio\_xml\_name " />

<down name="phy\_xml\_name " />

</interface>

<phy name="phy\_xml\_name ">

<up name="antenna\_xml\_name " />

<down name="signal\_tracker\_xml\_name " />

<down name="interference\_xml\_name " />

<down name="error\_xml\_name " />

<down name="coding\_xml\_name " />

</phy>

<error name="error\_xml\_name ">

<up name="phy\_xml\_name " />

</error>

<coding name="coding\_xml\_name ">

<up name="phy\_xml\_name " />

</coding>

<interference name="interference\_xml\_name ">

<up name="phy\_xml\_name " />

</interference>

<signal\_tracker name="signal\_tracker\_xml\_name ">

<up name="phy\_xml\_name " />

</signal\_tracker>

</implementation>

<energy name="energy\_xml\_name ">

</energy>

<mobility name="mobility\_xml\_name ">

</mobility>

</node\_architecture>

<!-- == Simulation ===================================================== -->

<simulation nodes="128" duration="37s">

<global\_map name="global\_map">

</global\_map>

<monitor name="birth\_mon">

</monitor>

<node id="0" architecture="coordinator">

</node>

</simulation>

</worldsens>

### Example of Core XML Scenario for WSNet 3.1

There are some examples of XML files to show the proper usage of XML files. This files are under the examples folder and are briefly explained below. Users are encouraged to go through the XML files to see how the models and its parameters were initialized and used.

#### bmac-2nodes

Simulation of 2 nodes with architecture “sensor1” exchanging data while using BMAC protocol along with the transceiver radio\_half1d.

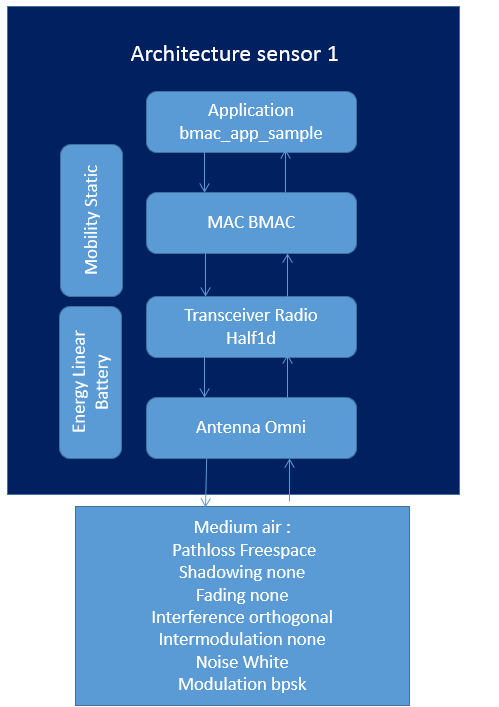
****

Figure 47: Architecture of the bmac-2node simulation

#### xmac-2nodes

Simulation of 2 nodes with architecture “sensor1” exchanging data while using XMAC protocol along with the transceiver radio\_half1d.

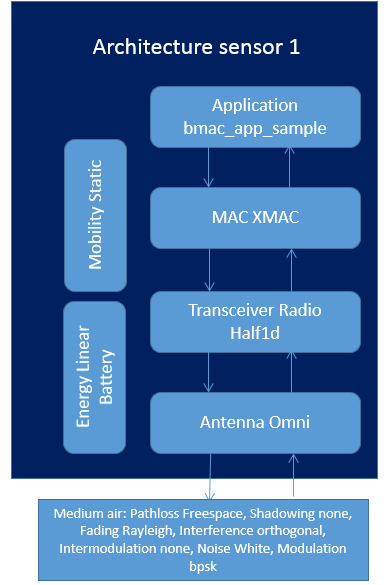


Figure 48: Architecture of the xmac-2nodes simulation

#### cbr\_v2

Simulation of 50 nodes with architecture “sensor” using the CBR\_v2 application combined with the routing\_greedy model over a DCF 802.11 MAC and Radio Half1d WiFi.

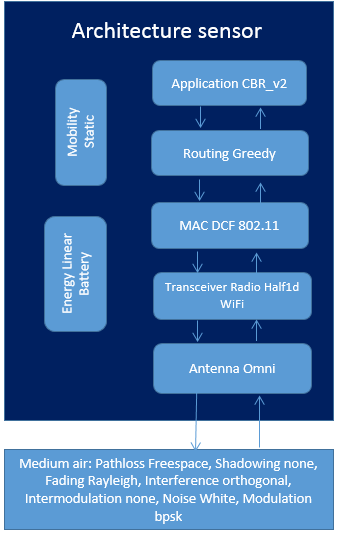
****

Figure 49: Architecture of the cbr\_v2 simulation

#### cbr

Simulation of 50 nodes using the CBR application combined with the routing\_greedy model over an WIFI MAC and radio.

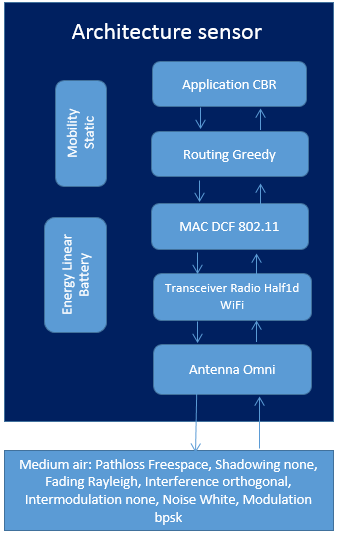
****

Figure 50: Architecture of the cbr simulation

#### demo

Simulation with two different mediums, namely, air and water. 3 nodes on each medium and one gateway between this two mediums.

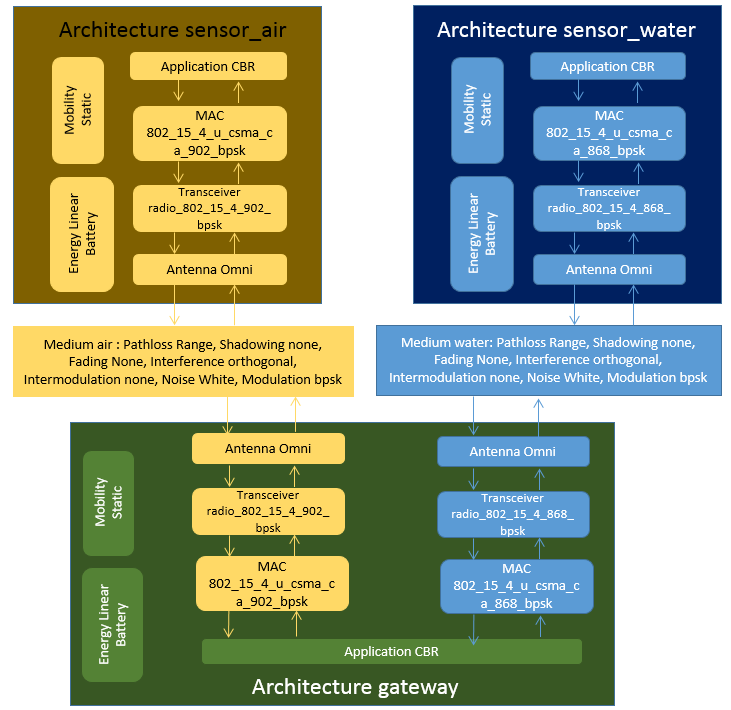


Figure 51: Architecture of the demo simulation

#### epidemic

Simulation of 19 nodes with architecture sensor and 1 node with architecture source to illustrate the use of the Epidemic Application module.

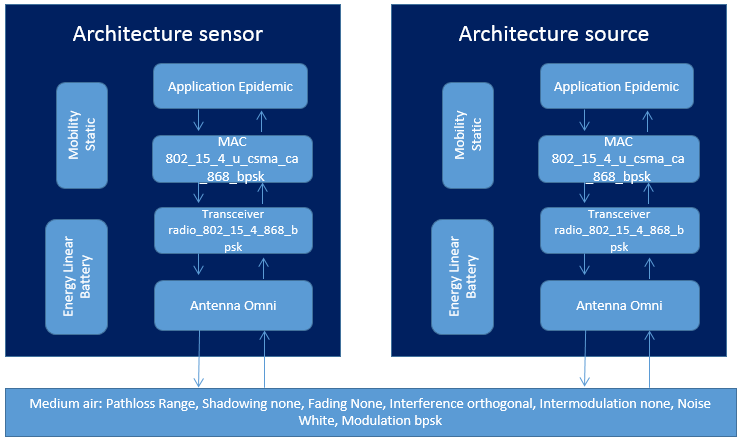


Figure 52: Architecture of the epidemic simulation

#### hello

Simuation with 50 nodes using node architecture sensor to exemplify the use of the Hello application along with mobility torus plane.

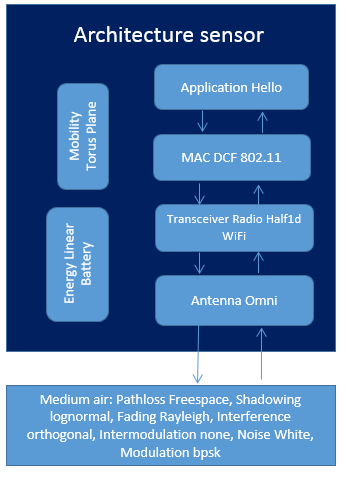


Figure 53: Architecture of the hello simulation

#### hello\_mixed.xml

Simulation with 50 nodes using node architecture sensor to exemplify the use of the Hello application in an indoor scenario. The indoor map has 25 rooms of 10m x 10m and is used together with the indoor link model. The position\_inside\_room mobility is configured to place 2 nodes per room. We also use the mixed model for pathloss, fading and shadowing.

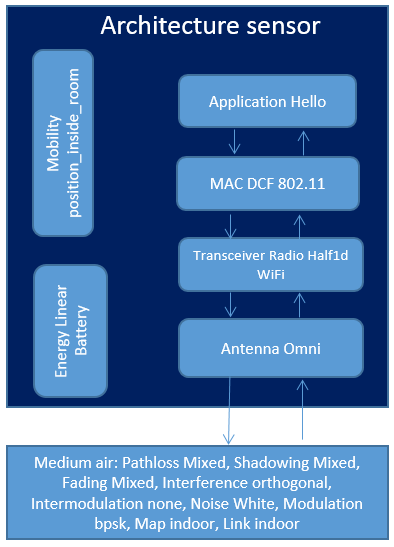


Figure 54: Architecture of the hello simulation with mixed propagation models

### Example of Core XML Scenario for WSNet 4.0

#### simple\_example\_spectrum

Simulation with 3 nodes giving a simple example of how to use the PHY, Interference, SignalTracker and Spectrum models. The simulation performs the data transfer of a single packet each node. We can observe the reception of each packet by the other 2 nodes.

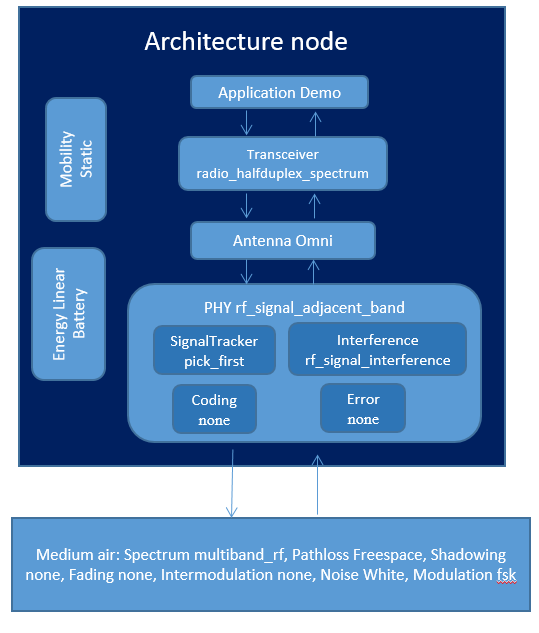
****

Figure 55: Architecture of the simple\_example\_spectrum simulation