

1 Overhaul of IEEE 802.11 Modeling and Simulation in NS-2 (802.11Ext)

Authors:

Qi Chen ⁺, Felix Schmidt-Eisenlohr ^{*}, Daniel Jiang ⁺,
Marc Torrent-Moreno ^{*}, Luca Delgrossi ⁺, Hannes Hartenstein ^{*}

^{*} University of Karlsruhe (TH)

⁺ Mercedes-Benz Research & Development North America

Last update: 02/01/2008

1.1 Overview

1.1.1 Introduction

This chapter describes the recently developed new modeling of IEEE 802.11 for NS-2, which introduces two new modules: **Mac802_11Ext** and **WirelessPhyExt**. The extensions are based on Mac802_11 and WirelessPhy, but did a major modification to the original code, aiming at a significantly higher level of simulation accuracy.

The new modeling includes the following key features:

- Structured design of MAC functionality modules: transmission, reception, transmission coordination, reception coordination, backoff manager and channel state monitor
- Cumulative SINR computation
- MAC frame capture capabilities
- Multiple modulation scheme support
- Packet drop tracing at PHY layer

By simply choosing these two modules and replacing MAC-802_11 and WirelessPhy used in a running TCL script, users are able to get the above benefits in wireless communication studies.

1.1.2 Evolution and Compatibility

MAC-802_11Ext and WirelessPhyExt are implemented and tested under NS-2.31, but is also installable under NS-2.32. They are introduced as new protocol modules, given the TCL class name **Mac/802_11Ext** and **Phy/WirelessPhyExt**. NS-2 users have the freedom to choose either the original or enhanced IEEE802.11 implementation to use in their study. Figure 1 shows the protocol stack of an NS mobile node in a NS2 simulation. It needs to be pointed out that Mac-802_11Ext and WirelessPhyExt have to be used together.

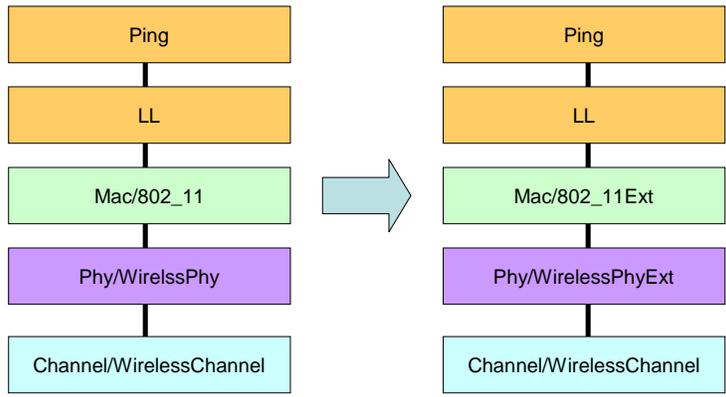


Figure 1: Node construction in TCL Script

Though Mac-802_11Ext and WirelessPhyExt are inherited classes from Mac-802_11 and WirelessPhy, they undergone major modifications in the design and coding. The first major change is the separation of functionalities of PHY and MAC. WirelessPhyExt implements network interface logic completely and also takes care of monitoring the received radio power. Mac-802_11Ext is in charge of processing of MAC frames, maintaining the logic view of channel states and managing the back off procedure.

The second most distinguished change is a structured design for the internal function modules, as shown in Figure 2. Each function module matches with IEEE standard's design and allows an easy extension or simplification. With carefully defined signaling calls between function modules, the implementation of every module is relatively independent of each other.

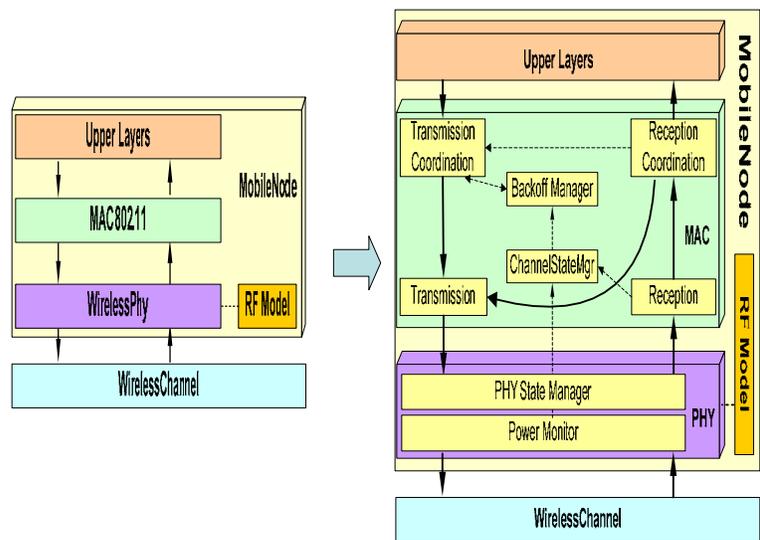


Figure 2: Extension of MAC and PHY modules

Details of the modeling are available in a paper published at MSWiM 2008: *Overhaul of IEEE 802.11 Modeling and Simulation in NS-2* that can be found here:

<http://portal.acm.org/citation.cfm?id=1298155&coll=ACM&dl=ACM&CFID=47104902&CFTOKEN=14737307>

1.2 Patch and Installation (based on NS 2.31)

1.2.1 Installation by Patch

The code contribution is hosted on the following webpage:

http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php

patch and code can be downloaded from there. To have our code, please follow the below steps:

1. Install a clean ns-2.31 distribution
2. Copy NS-2.31-MAC-PHY-Overhaul-2008-02-01.patch to ns-allinone-2.31/ns-2.31 folder and apply the patch by command

```
patch -p0 < NS-2.31-MAC-PHY-Overhaul-2008-02-01.patch
```

1.2.2 Configure and Make ns-2

configure and make ns2

run ./validation to test the installation.

Test scripts are available for both broadcast and unicast scenarios in the directory *examples*

ns broadcast_validation.tcl

In this scenario, two groups of nodes are configured to do periodic broadcast. Hidden terminals effects shown through PHY drops in the trace file.

ns unicast_validation.tcl

In this scenario, two nodes are configured to do unicast transmissions. The interactions, such as ACK, retry timeout, CTS/RTS, can be observed in the trace file.

The directory also contains the files *IEEE802-11a.tcl* and *IEEE802-11p.tcl* that set the PHY/MAC parameters according to the standard defined in the file name. **Note that currently for IEEE 802.11b/g the according values are not provided but must be set by the user.**

1.2.3 List of the touched source code files

| For Module | Path | File name | Touch Reasons |
|-------------------------------|---------|--------------------|---|
| | Mac | wireless-phyExt.h | Create as a child class of WirelessPhy; |
| | Mac | wireless-phyExt.cc | Introduce trace of PHY drops |
| | ./ | Makefile.in | Include mac-802_11Ext.o in the compilation |
| Phy/WirelessPhyExt | tcl/lib | ns-default.tcl | |
| | Trace | cmu-trace.h | Printout drop line with drop reasons |
| | Trace | cmu-trace.cc | |
| | tcl/lib | ns-lib.tcl | Put a switch to configure a node do phytrace |
| PHY tracing | tcl/lib | ns-moiblenode.tcl | support PHY tracing |
| (tracing packet drop reasons) | tcl/lib | ns-default.tcl | support PHY tracing by "Simulator set PhyTrace_ON/OFF" |
| | Mac | mac-802_11Ext.h | Create as a child class of Mac-802_11; |
| | Mac | mac-802_11Ext.cc | |
| | tcl/lib | ns-default.tcl | Mac/802_11 -> Mac/802_11Ext |
| Mac/802_11Ext | ./ | Makefile.in | Include mac-802_11Ext.o in the compilation |

| | | | |
|-------------------------|----------|---------------|---|
| | Mobile | nakagami.h | Implementation Nakagami RF model |
| | Mobile | nakagami.cc | Implementation Nakagami RF model |
| | Tools | ranvar.h | Implement Gamma Distribution to support Nakagami distribution |
| | Tools | ranvar.cc | |
| Propagation/Nakagami | ./ | Makefile.in | Include nakagami.o in the compilation |
| | Apps | pbcc.h | Create as a child class of Agent; |
| | Apps | pbcc.cc | |
| | ./ | Makefile.in | Include pbcc.o in the compilation |
| Agent/PBC | | | Add packet type PT_PBC; |
| (a Periodic Broadcast | Common | packet.h | Add modulation_scheme in the common header |
| Ping Agent message | ./ | Makefile.in | Include pbcc.o in the compilation |
| generation application) | tcl/lib/ | ns-packet.tcl | Include PBC_header in the header list |

Table 1: List of touched files

1.3 Using the Extended IEEE802.11 Models

1.3.1 Configuration Node in TCL

Configuration is simple; first just choose the right protocol module:

```
$ns_ node-config \
...
-macType Mac/802_11Ext
-phyType Phy/WirelessPhyExt \
```

Secondly chose the proper protocol parameter profile or set protocol parameters manually

```
source "IEEE802-11a.tcl"
```

Many protocol parameters are bound to TCL variables. Several TCL scripts have been prepared to collect the values defined in different protocol standards.

```
IEEE802-11a.tcl
IEEE802-11p.tcl
```

Users can simply include the script in their running script, **before constructing nodes**, such as:

```
source "IEEE802-11p.tcl"
```

Beyond this inclusion, users are expected to adjust the packet transmission power and packet size manually to meet their testing scenarios.

1.3.2 Capture capability

The capture capability is a very important feature in this model. It allows a receiver to distinguish the MAC frame header and body, using different criterion to processing them. MAC header is transmitted with defined BPSK modulation, while the MAC data can be coded in a much higher modulation scheme. A real IEEE802.11 product can pick up a frame with stronger header signal among multiple frames and complete its data reception. Such a technology, called capture, is wildly used and proved very effective in enhancing reception probabilities for vehicular communications. NS-2 users now can turn on a TCL switch to simulate this feature or turn off the switch to simulate original style implementation without capture.

Following lines shows the parameters and their default values.

```

Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1
Phy/WirelessPhyExt set DataCaptureSwitch_ 0
Phy/WirelessPhyExt set SINR_PreambleCapture_ 2.5118; ;# 4 dB
Phy/WirelessPhyExt set SINR_DataCapture_ 100.0; ;# 10 dB

```

For more information about the preamble capture and data capture, please check the publication mentioned above.

1.3.3 Multiple Modulation scheme Support

MAC802_11Ext and WirelessPhyExt are able to simulate MAC frame in different modulation schemes. A MAC frame modulated with an advanced modulation scheme has shorter transmission duration but requires a higher SINR to receive and decode it.

NS-2 users can define the desired modulation scheme for the entire simulation or even for every individual MAC frame. The definition is done by flagging a variable in the frame common header:

```
cmnhdr->mod_scheme_
```

it takes a integer value to specify the exact modulation scheme. The values are listed in the Table 2 Modulation Scheme. Please note that this table refers to IEEE 802.11a profile and not a comprehensive list of all modulation and coding rate combinations yet. When corresponding threshold values are obtained, this table will be expanded. With different IEEE 802.11 air links, this table needs to be adjusted accordingly.

Table 2 Modulation Scheme

| | Defined Modulation Schemes | | | |
|-----------------------------|----------------------------|------------------------|-------------------------|-------------------------|
| Modulation and coding rate | BPSK and ½ coding rate | QPSK and ½ coding rate | QAM16 and ½ coding rate | QAM64 and ¾ coding rate |
| Data rate for 20MHz channel | 6 mbps | 12 mbps | 24 mbps | 54 mbps |
| Reference ID | 0 | 1 | 2 | 3 |

In the patch, a PingAgent like message generator application PBCAgent is able to generate messages and set the modulation scheme flag.

1.3.4 PBCAgent

As a Ping_Agent like message generator, PBCAgent is added in the patch to allow users test the code. This PBCAgent can define message generation frequency, the frame data modulation scheme and MAC frame payload size.

The TCL name of the agent is Agent/PBC. Packet type is PT_PBC. The TCL commands to define the message generation behavior and some examples are here:

```

Agent/PBC set payloadSize 200 ; #(in bytes)
Agent/PBC set periodicBroadcastInterval 0.1 ;# (in seconds)
Agent/PBC set periodicBroadcastVariance 0.1 ;# (in seconds)
Agent/PBC set modulationScheme 0; # the default modulation scheme for data is BPSK

```

1.3.5 Nakagami RF model

As an addition to the existing radio propagation models, Nakagami RF model is developed and added in this patch. Nakagami is a mathematical general modeling of a radio channel with fading. Compared to the existing models (shadowing and two-ray ground), Nakagami RF model has more configurable parameters to allow a closer representation of the wireless communication channel. It is able to model from an perfect

free space channel, to a moderate fading channel on highway, even to a dramatically fading channel in urban communities.

Nakagami distribution is defined by the following probability density function:

$$f(x) = \frac{2m^m x^{2m-1}}{\Gamma(m)\Omega^m} \exp\left[-\frac{mx^2}{\Omega}\right], \quad x \geq 0, \quad \Omega > 0, \quad m \geq 1/2$$

The corresponding pdf of power (square of the signal amplitude) at the given distance can be obtained by a change of variables and is given by a gamma distribution of the following form:

$$p(x) = \left(\frac{m}{\Omega}\right)^m \frac{x^{m-1}}{\Gamma(m)} \exp\left[-\frac{mx}{\Omega}\right], \quad x \geq 0$$

Ω is the expected value of the distribution and can be interpreted as the average received power. m is the so called shape or fading parameter.

The values of the parameters m and Ω is functions of distance. So the Nakagami model is defined by two functions: $\Omega(d)$ and $m(d)$.

- Rayleigh distribution is a special case of Nakagami distribution where $m(d) = 1$ (for every d)
- Larger values of m give less severe fading.

The implementation of Nakagami RF model in NS-2 locates in nakagmi.cc. The TCL class name is Propagation/Nakagmi. It defines a set of parameters:

***gamma0, gamma1, gamm2
d0_gamma_,d1_gamma_***

gamma values define the radio signal average attenuation over distance, corresponding to the Ω function. $d0_gamma_$ and $d1_gamma_$ is the distance where gamma value discontinues.

***m0, m1, m2
d0_m, d1_m***

m values define the radio signal fading, corresponding to the m function. $d0_m$, $d1_m$ is the distance where gamma value discontinues.

use_nakagami_dist_

this TCL variable is the switch to turn on the random number generator. If it is set to false, reception power is deterministically calculated by transmission power and distance between the sender and the receiver. If it is set to true, Nakagami model returns each time a randomized reception power. The average of the random powers is defined by gamma function, and the variance of the random power is defined by m function.

Here is an example, a set of parameter values representing a radio propagation channel on highway 101 in bay area.

```
Propagation/Nakagami set gamma0_ 1.9  
Propagation/Nakagami set gamma1_ 3.8  
Propagation/Nakagami set gamma2_ 3.8  
  
Propagation/Nakagami set d0_gamma_ 200  
Propagation/Nakagami set d1_gamma_ 500  
  
Propagation/Nakagami set use_nakagami_dist_ false
```

Propagation/Nakagami set m0_ 1.5
Propagation/Nakagami set m1_ 0.75
Propagation/Nakagami set m2_ 0.75

Propagation/Nakagami set d0_m_ 80
Propagation/Nakagami set d1_m_ 200

1.3.6 PHY Tracing

In all previous NS-2 distributions, WirelessPhy delivers all the arriving packets whose power is higher than the carrier sensing threshold to the upper Mac802_11 layer. Mac802_11 decides which packet to receive, which to drop. The new WirelessPhyExt does more logical processing, and only delivers the successfully received packets to Mac802_11Ext. The tracing of dropped events, therefore, are implemented in WirelessPhyExt. By default, Phy tracing is not active and should be activated through TCL command if interested.

The new PHY tracing format follows the traditional cmu-tracing style. The cmu-trace.cc is modified to include the PHY tracing codes. The trace-name is defined as PHY in tracing outputs.

In the TCL running script, activate PHY trace by using command line:

```
$ns_ node-config \  
...  
-agentTrace ON \  
-routerTrace OFF \  
-macTrace OFF \  
-phyTrace ON
```

In the current implementation, a message drop event is tagged with drop reasons. All the drop reasons are categorized into two groups.

The first group of reasons is related to the deficit of receiving signal strength of the dropped message, even under non-interfering conditions.

- PND Reception power is either lower than the carrier sensing threshold or not enough for its preamble being received even without any interference
- DND Reception power is higher than the carrier sensing threshold but not enough to decode the data even without any interference

If the reception power of the dropped message is not only higher than the carrier sense threshold but also sufficient to decode the data if no interference is present, then such a drop is related to the scheduling of the message on the channel, i.e., interferences from messages transmitted during the same time. Those drops belong to group two and are named after the current state of the wireless interface.

- SXB a message is dropped when the PHY interface is IDLE, but busy searching for a valid preamble
- PXB a message is dropped when the PHY interface is in the progress of receiving a frame preamble
- TXB a message is dropped when the PHY interface is busy in transmitting a frame
- INT a message is dropped because of the interruption from the MAC, (MAC forces the abortion of the current reception, usually for transmitting a control frame of its own, like a ACK or CTS frame)

$PND+DND+SXB+PXB+RXB+TXB+INT+recvd\ message = 100\%$ of arriving events

Note: There is a TCL variable: **Phy/WirelessPhyExt trace_dist_**. If the distance from the receiver to the sender is larger than this distance, message drops due to PND or DND will not be logged in the tracing output file. This is a protection mechanism to prevent the output file from being extremely large. The default value of **trace_dist_** is set to infinity (1e6 m). Users can customize the value.

1.3.7 Inheritance and Extension

The structured design of IEEE802.11 is aimed to facilitate NS-2 users understanding the implementation and allow them to better focusing on their interested part of the code. Any modification inside one functionality module is isolated to the other modules to the maximal extent. Therefore, users can modify Transmission Coordination or Reception Coordination to change the MAC distributed coordination function to their design, or implement a more advanced backoff procedure than the current one.

For example, users who want to extend this implementation can create a child class of MAC-802_11Ext, call it MAC-802_11Ext_newRXprocessing and implement a more advanced RXC (reception coordination) to process some new MAC frames. He does not need to worry about the TXC, backoff , channel state, etc.

One thing needs to be pointed out is that some of the internal member variables or functions inside a function module are not publicly accessible by the child classes of other modules. In this case, users can define some public access function to solve the problem.

1.4 Appendix: Design SDL

The design of the new PHY/MAC is described in detail in “Overhaul of IEEE 802.11 Modeling and Simulation in NS-2” which was published at MSWiM 2007.

The following SDL helps to understand the logic flow of the code implementation.

