TOSSIM

Răzvan Musăloiu-E.

TinyOS

What is TOSSIM?

Discrete event simulator

ns2

TinyOS

Alternatives

Cycle-accurate simulators

Avrora, MSPSim

Two directions

Port

make PC a supported platform

TOSSIM
in tinyos-1.x

Virtualize

simulate one of the supported platforms

TOSSIM
in tinyos-2.x

Features

- Simulates a MicaZ mote
  - ATmega128L (128KB ROM, 4KB RAM)
  - CC2420

- Uses CPM to model the radio noise

- Supports two programming interfaces:
  - Python
  - C++

TinyOS

Anatomy

TOSSIM

```
tos/lib/tossim
tos/chips/atm128/sim
tos/chips/atm128/pins/sim
tos/chips/atm128/timer/sim
tos/chips/atm128/spi/sim
tos/platforms/mica/sim
tos/platforms/micaz/sim
tos/platforms/micaz/chips/cc2420/sim
```
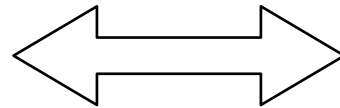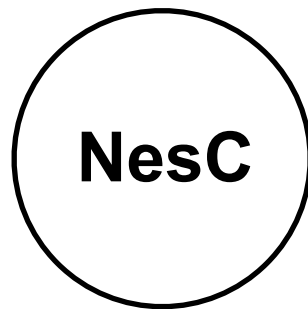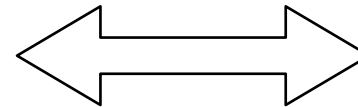
Application

```
Makefile
*.nc
*.h
```

Simulation Driver

```
*.py | *.cc
```

Quick Overview

Application   Simulation

Glue

NesC ⟷ **Python**

C++
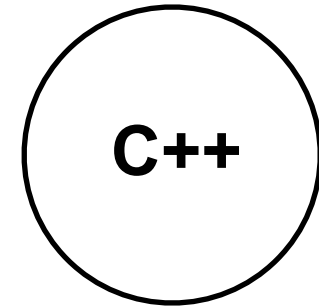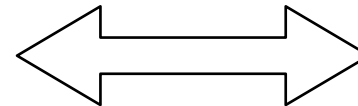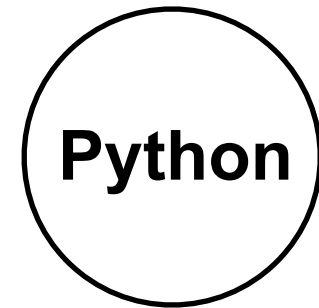
The Building Process

```
$ make micaz sim
```

1. Generate an XML schema   *app.xml*

2. Compile the application   *sim.o*

3. Compile the Python support   *pytossim.o*
                                                                      tossim.o
                                                                  c-support.o

4. Build a share object   *_TOSSIMmodule.o*

5. Copying the Python support

                                                                  TOSSIM.py

```
$ ./sim.py
```

TOSSIM.py

Tossim

Radio

Mote

Packet

Mac

TOSSIM.Tossim

.getNode() → TOSSIM.Mote

.radio() → TOSSIM.Radio

.newPacket() → TOSSIM.Packet

.mac() → TOSSIM.Mac

.runNextEvent()

.ticksPerSecond()

.time()

10 seconds

```
from TOSSIM import *

t = Tossim([])

 ...

while t.time() < 10*t.ticksPerSecond():

    t.runNextEvent()
```

dbg

Syntax

dbg(*tag, format, arg1, arg2, ...*);

Example

dbg("Trickle", "Starting time with time %u.\n", timerVal);

Python

t = Tossim([])
t.addChannel("Trickle", sys.stdout)

Useful Functions

char*   sim_time_string()

sim_time_t   sim_time()

int   sim_random()

sim_time_t   sim_ticks_per_sec()

typedef *long long int sim_time_t*;

Radio Model

Closest-fit Pattern Matching (CPM)
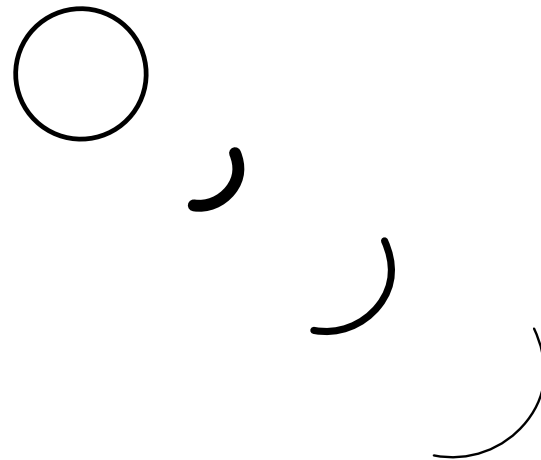
Improving Wireless Simulation Through Noise Modeling

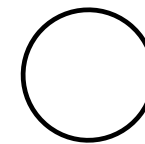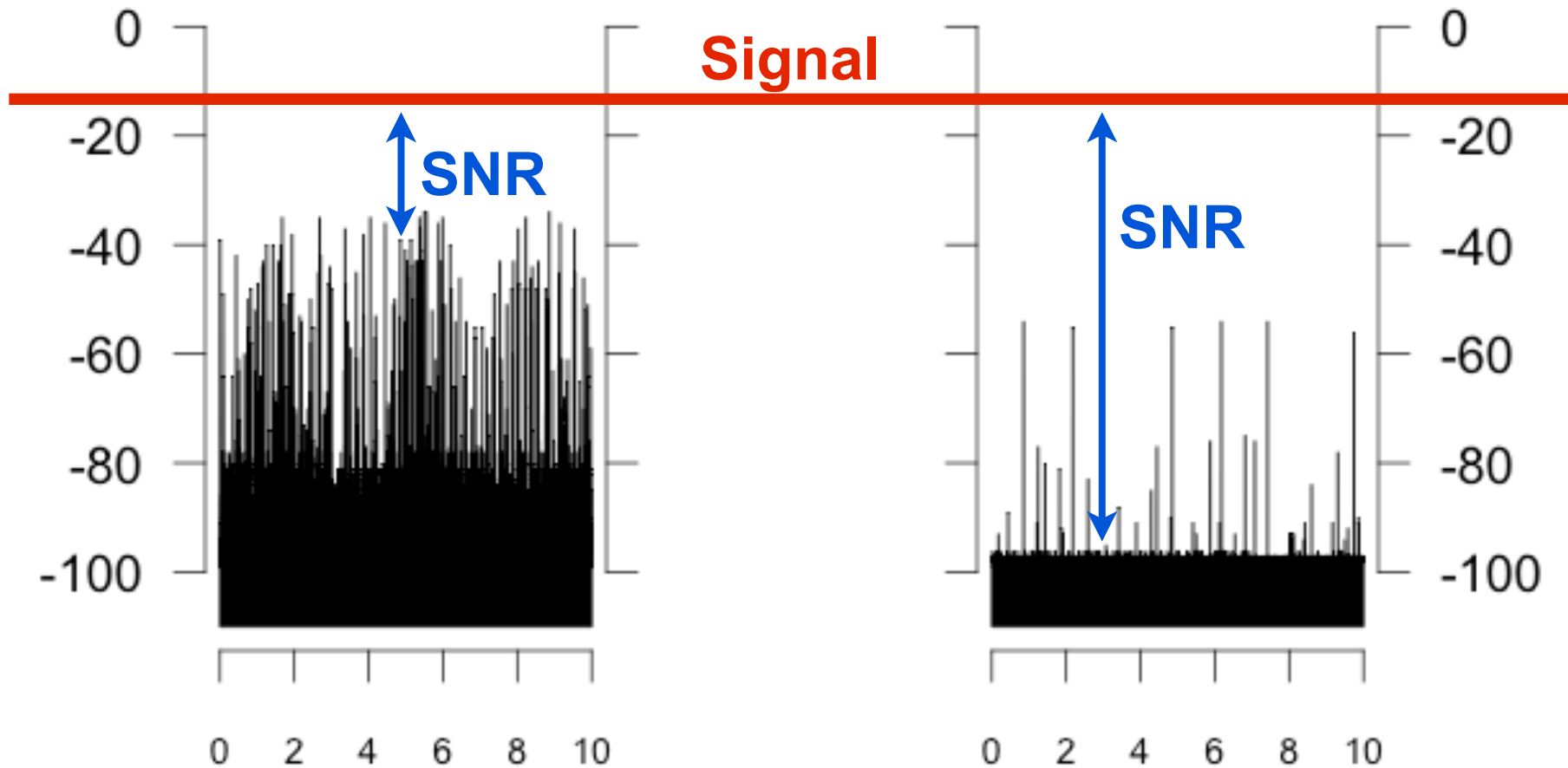HyungJune Lee, Alberto Cerpa, and Philip Levis

IPSN 2007

TinyOS

14

Radio Model

Sender

Receiver

Noise Level
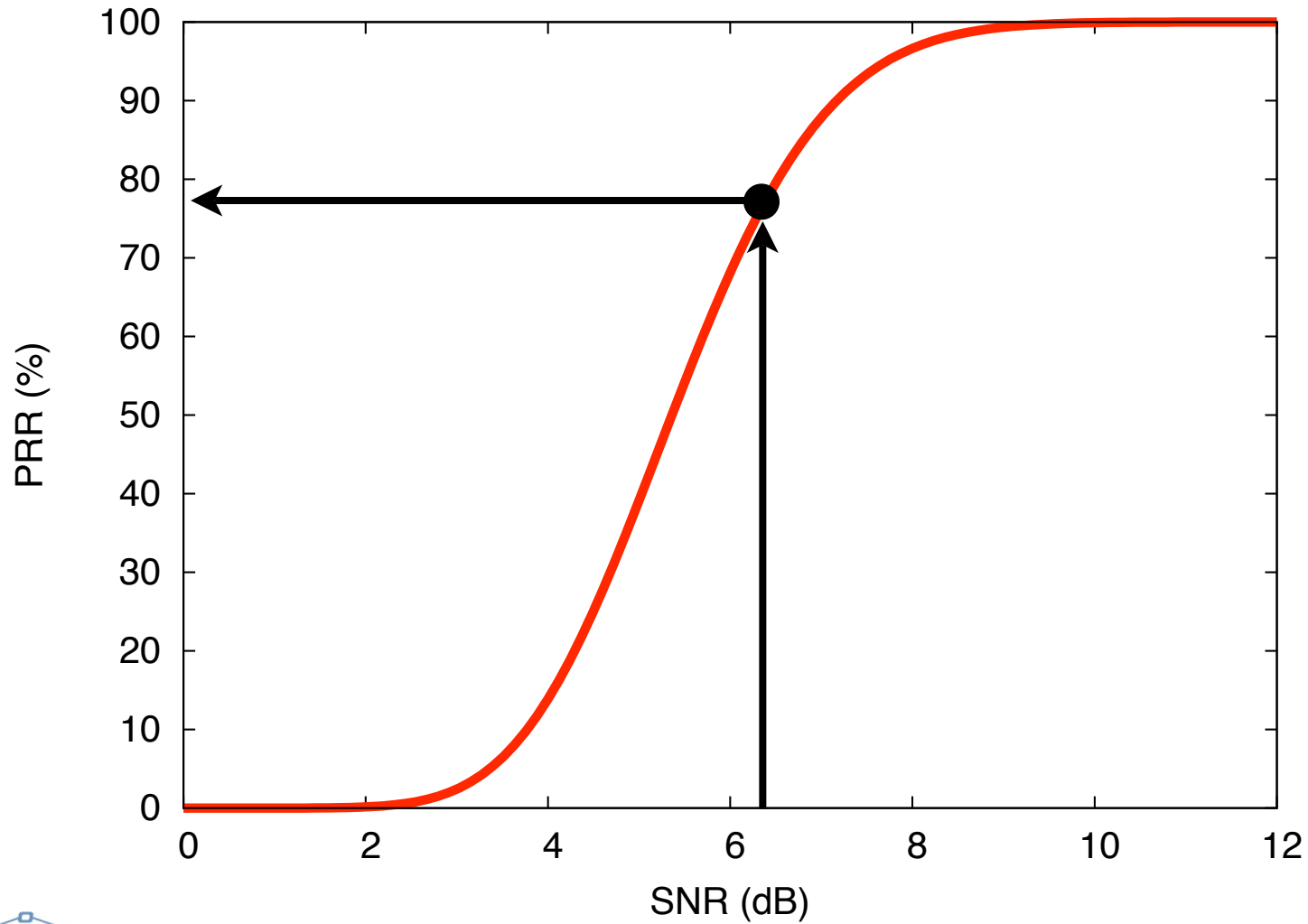

Meyer Heavy   Casino Lab

Signal

SNR   SNR

CC2420 SNR/PRR

TOSSIM.Radio

.add(*source, destination, gain*)

.connected(*source, destination*) → True/False

.gain(*source, destination*)

TOSSIM.Mote

.bootAtTime(*time*)

.addNoiseTraceReading(*noise*)
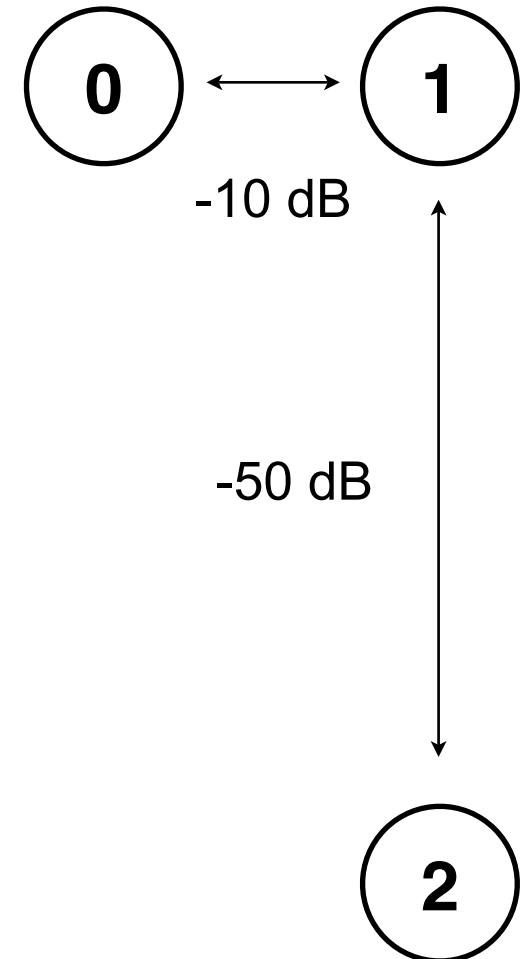
.createNoiseModel()

.isOn() → True/False

.turnOn()/.turnOff()

Example

```
from TOSSIM import *

t = Tossim([])

r = t.Radio()

mote0 = t.getNode(0)

mote1 = t.getNode(1)

mote2 = t.getNode(2)

r.add(0, 1, -10)

r.add(1, 0, -10)

r.add(1, 2, -50)

r.add(2, 1, -50)
```
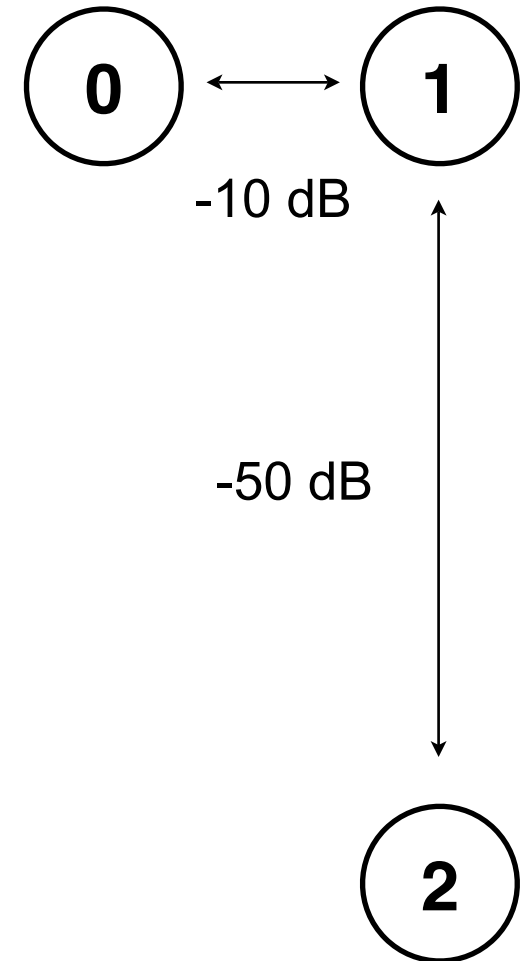
0 ←→ **1**

-10 dB

-50 dB

2

Example (cont)

```
noise = file("meyer-short.txt")
lines = noise.readlines()
for line in lines:
  str = line.strip()
  if (str != ""):
    val = int(str)
    for m in [mote0, mote1, mote2]:
      m.addNoiseTraceReading(val)

for m in [mote0, mote1, mote2]:
    m.createNoiseModel()
```

0 ⟷ 1

-10 dB

-50 dB

2

Other Features

- Injecting packets

- Inspecting internal variables

- C++ interface

- Debuging using gdb

Improvements

- **TossimLive**
  - SerialActiveMessageC

- **CC2420sim**
  - Multiple channels
  - PacketLink
  - CC2420Packet: .getRSSI(), .getLQI()
  - ReadRssi()
  - Flash support

Future

Parametrized the PRR/SNR curve based on packet size *(in progress)*

Support for multiple binary images *(harder)*

Next
Safe TinyOS