Accurate, Scalable Simulation of TinyOS Sensor Networks using Physically-Based Noise and Power Models

Tal Rusak

tr76@cornell.edu

Department of Computer Science

Cornell University

Class of 2009

SIGSCE, ACM Student Member

Wireless Sensor Networks and TinyOS

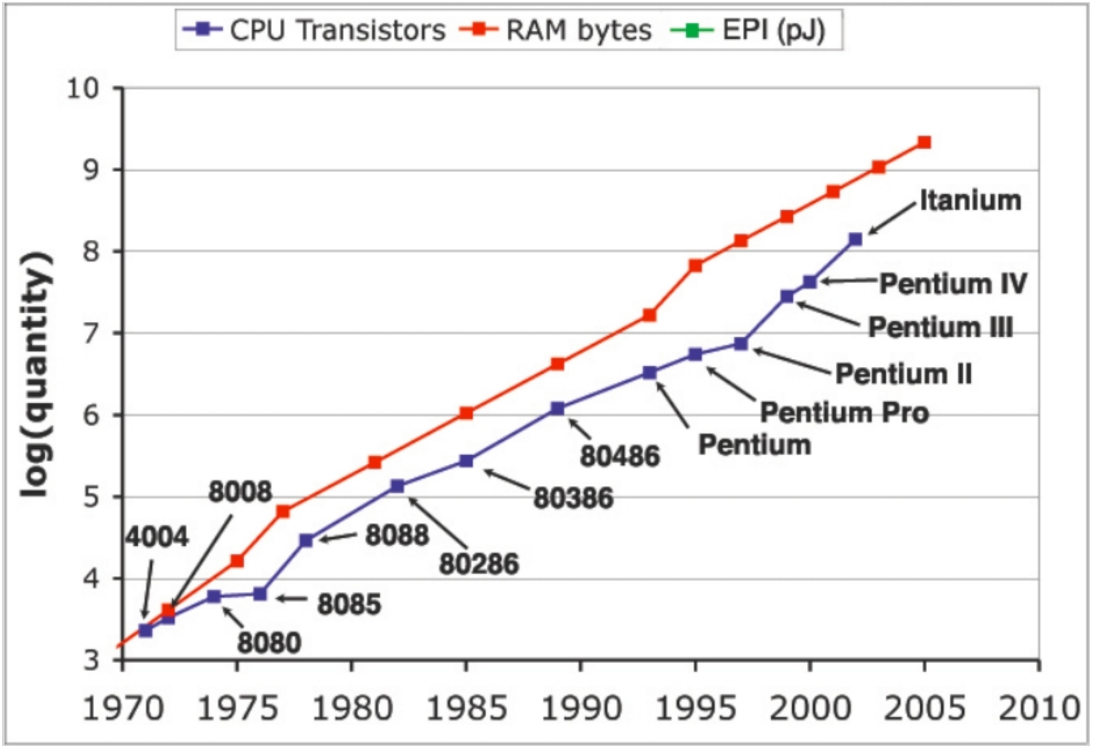
- ***Wireless sensor networks*** are networks of small, low capacity, slow but low-powered devices (*motes*) that ***collect information***, ***communicate wirelessly***, and ***make intelligent decisions*** over long periods (months to years) without human intervention.
- TinyOS is a software layer that provides frameworks, components, and interfaces for use with sensor network motes.

P. Levis, "TinyOS: An Open Platform for Wireless Sensor Networks." Invited Tutorial, IEEE MDM, May 10, 2006.

- Open-source code: `http://www.tinyos.net/`

Past 40 years ...

- Focus of research and development in computing was to satisfy Moore's Law for processor speed and memory capacity.

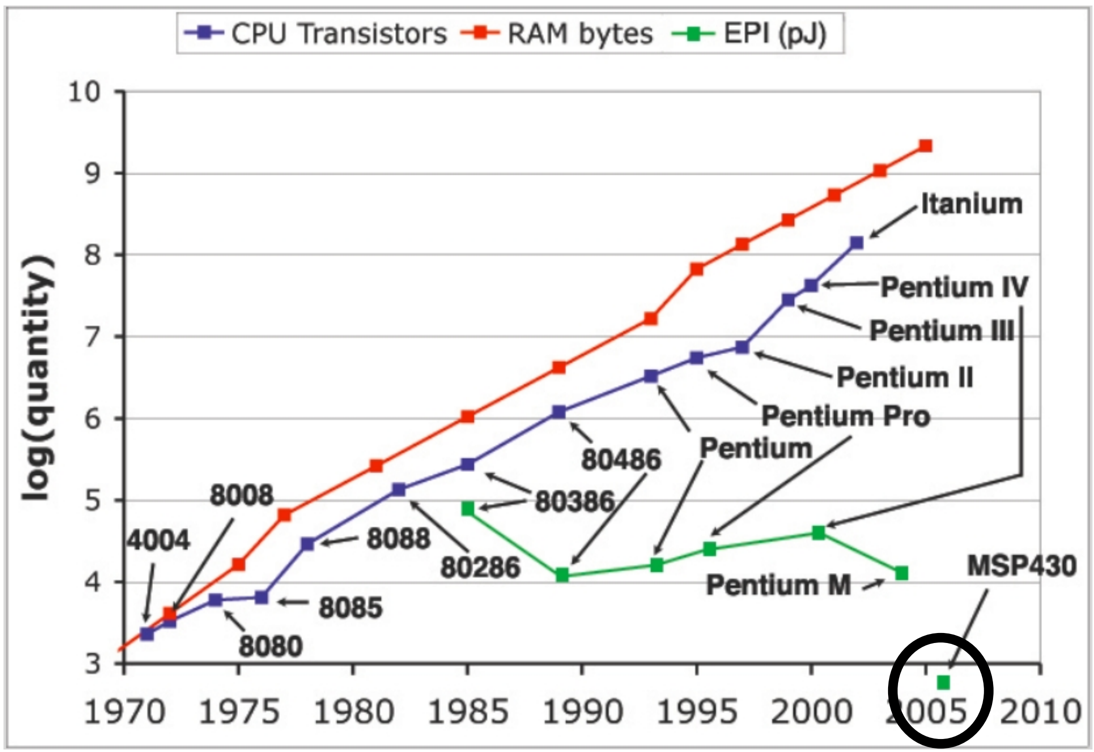


P. Levis, "TinyOS: An Open Platform for Wireless Sensor Networks." Invited Tutorial, IEEE MDM, May 10, 2006.

- Computers (usually) tied to power source to remain operational over long periods.

Moore's Law with Power Consumption

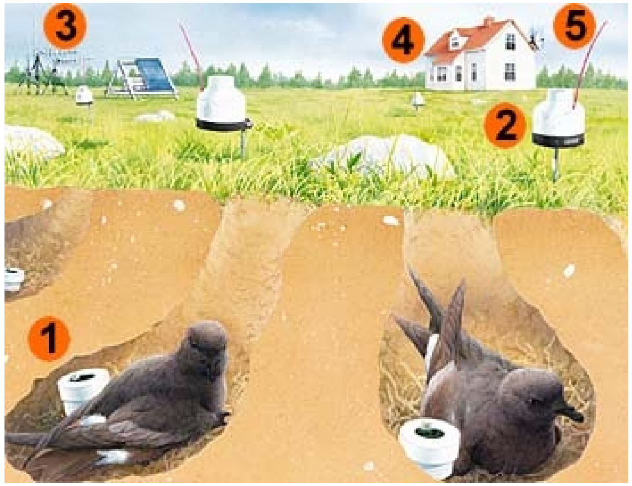
- Highly distributed computing applications require low-power, long-life computers.



P. Levis, "TinyOS: An Open Platform for Wireless Sensor Networks." Invited Tutorial, IEEE MDM, May 10, 2006.

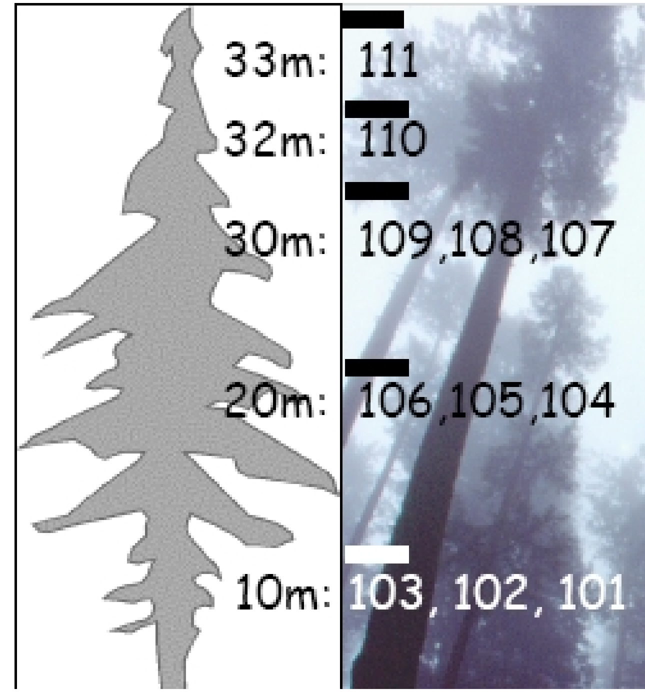
- Microcontrollers (e.g., MSP430) require an order of magnitude less power than typical processors.

Applications In Interdisciplinary Research ...



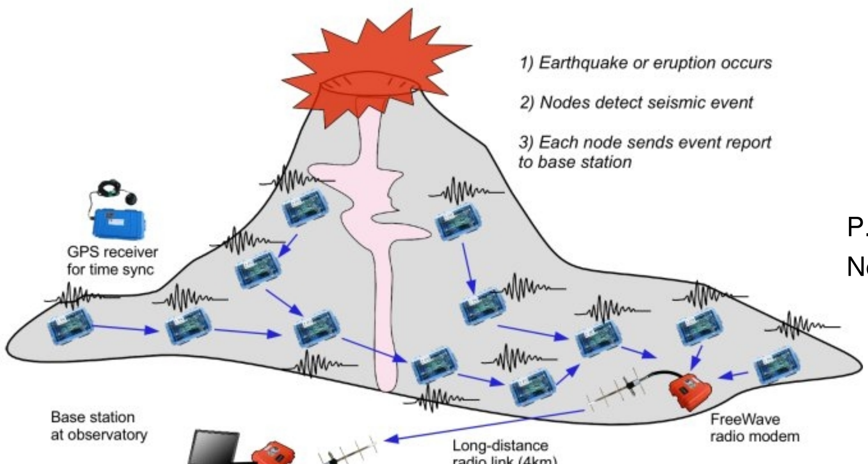
http://www.wired.com/wired/archive/11.12/network_pr.html

Biology: Great Duck Island: Monitoring of seabirds (2004)



Ecology: Redwood Microclimate Study (2005)

P. Levis, "TinyOS: An Open Platform for Wireless Sensor Networks." Invited Tutorial, IEEE MDM, May 10, 2006.



http://www.eecs.harvard.edu/~mdw/proj/volcano/

Geology: Volcano Monitoring (2006)

Simulation of TinyOS Sensor Networks

Tal Rusak

tr76@cornell.edu

Outline

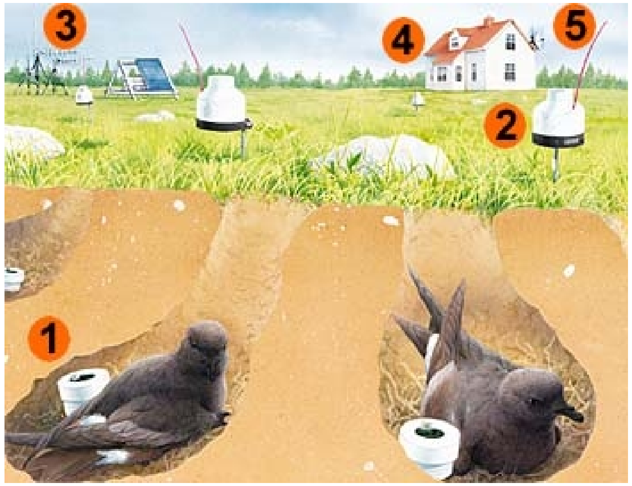
Introduction

Applications

Simulation and Experiments

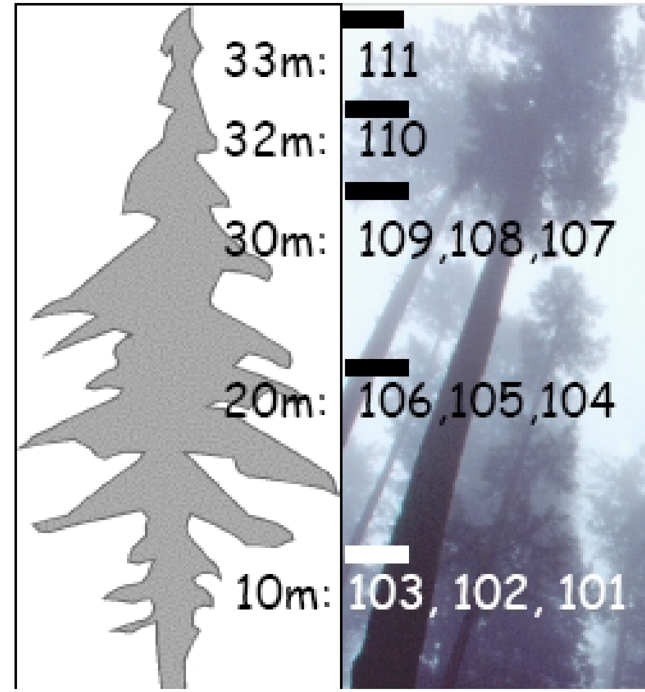
Results

... And Yield Success Rates

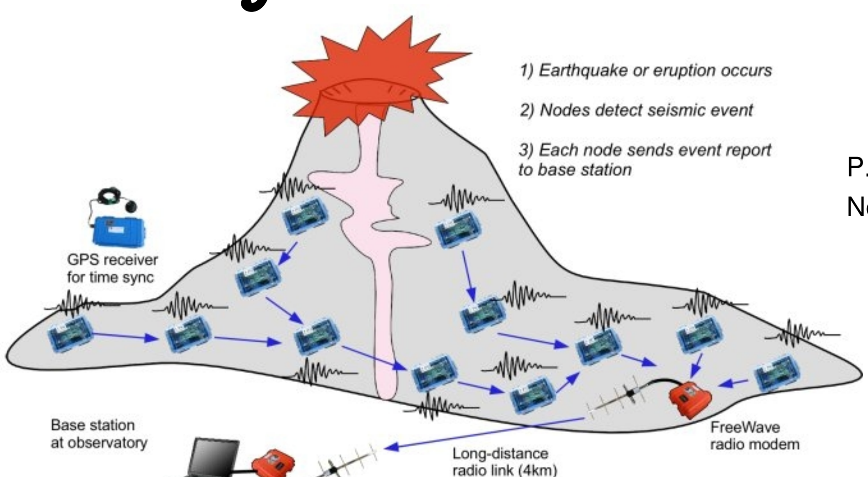


http://www.wired.com/wired/archive/11.12/network_pr.html

58% packet yield



40%



http://www.eecs.harvard.edu/~mdw/proj/volcano/

P. Levis, "TinyOS: An Open Platform for Wireless Sensor Networks." Invited Tutorial, IEEE MDM, May 10, 2006.

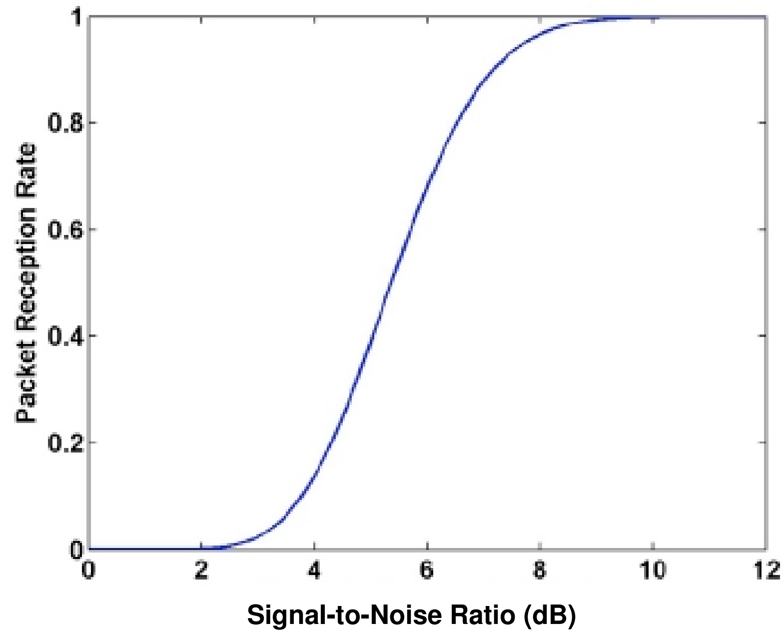
68%

Simulation: Enabling Layer for Successful Sensor Network Applications

- Debugging sensor networks is difficult:
  - (1) Expensive, time consuming deployments in remote regions.
  - (2) Constrained resources (3 LEDs = 3 bits of debugging information displayed on typical motes).
  - (3) Little memory to keep detailed logs.
- **TOSSIM** is the **T**iny**OS SIM**ulator.
- TOSSIM replaces several low level abstractions with PC-based equivalents, but otherwise uses the same code as TinyOS.
- The challenge is to effectively simulate radio links of wireless sensor networks.

Physically Based Simulation of Radio Links

- SNR curve for typical mote radio (CC2420):



HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

- SNR = Signal Power – Noise (logarithmic scale)
- Packet reception probability can be determined using the SNR value and the above curve.
- The challenge is to predict Signal Power (**S**) and Noise (**N**).

TOSSIM 2.0.1 (2007) Implementation

- TOSSIM 2.0.1 models *noise* by using the Concurrent Pattern Matching (CPM) algorithm:
  - (1) Input and pre-process an experimental trace:

0	2	1	2	0	2	2	0	0	1	1	1	1	2	0	0	0	2	9	0

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

k = History size = 2

  - (2) Take k traces from experiment; then sample PMF: 0 2 2 | 0 2 2 0 | 0 2 2 0 0

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

- TOSSIM 2.0.1 assumes *signal power* to be constant and user-defined.

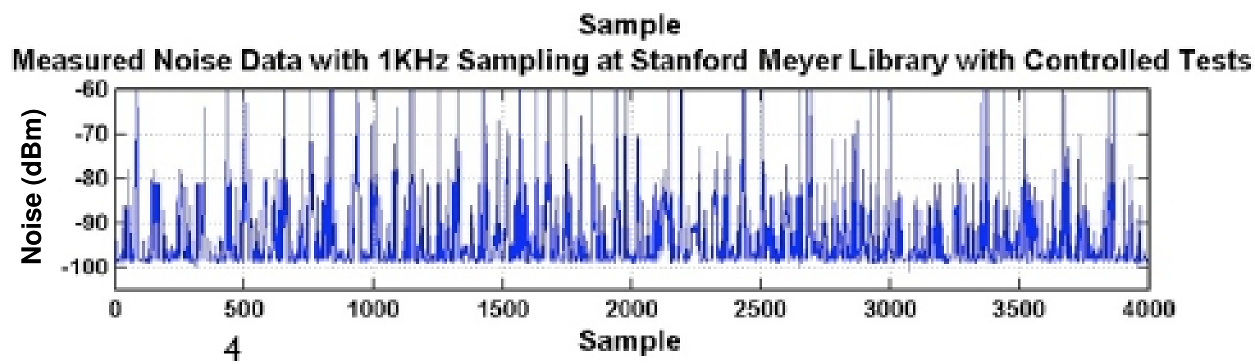
Simulation of TinyOS Sensor Networks

Tal Rusak

tr76@cornell.edu

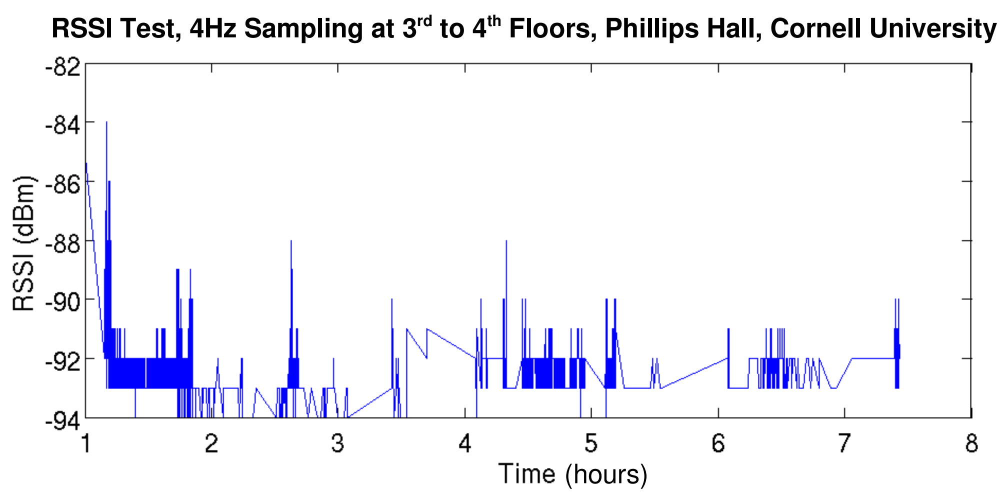
Power and Noise Variations

- Noise:



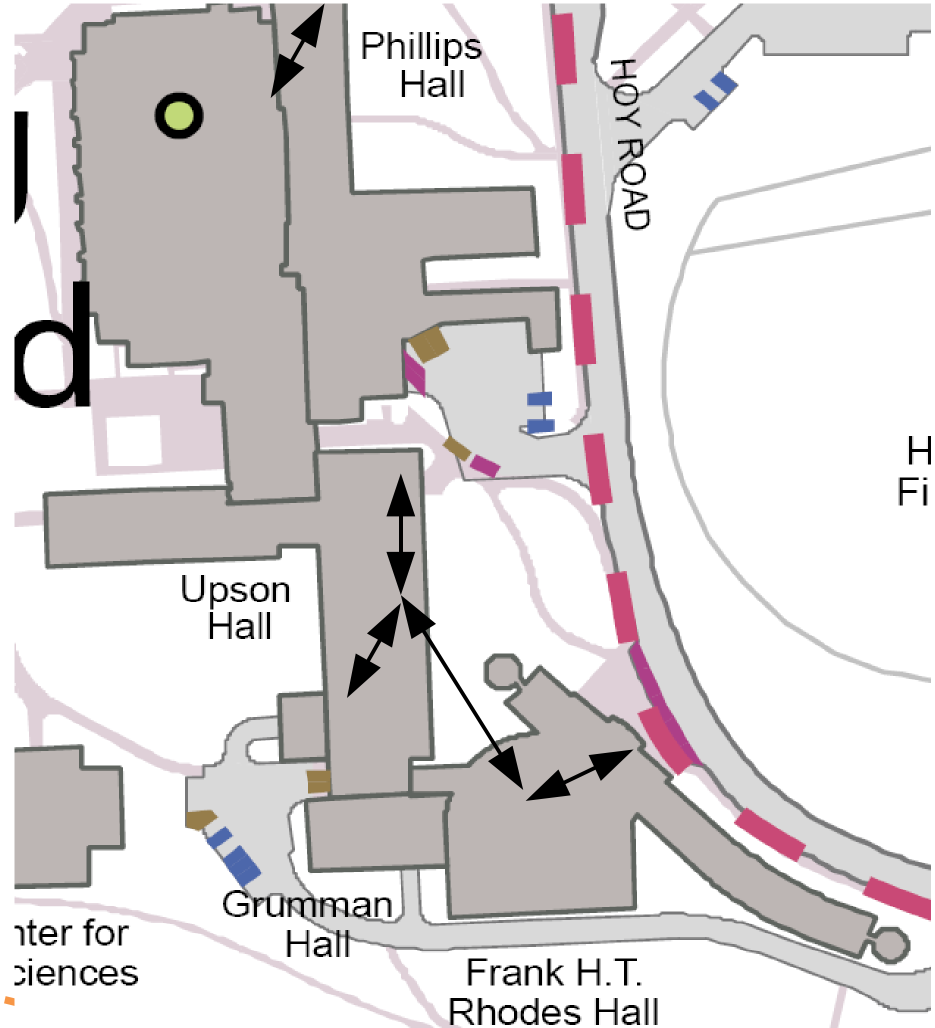
HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

- RSSI ≈ Signal Power:



Experiments Collected

- Signal power is a property of a link between two positions, not a property of the individual position.



Challenges of Collecting Power Traces

- ***Idea:*** Collect a signal power trace and use CPM to model signal power.
- Collecting power traces is more complex than collecting noise traces, since:
  - (1) Signal power can only be approximated by sampling the RSSI register:
    - Let **S** be the signal wave and **N** be the resultant wave from the sum of all noise waves in an environment. Then

$$RSSI \text{ (dBm)} = |\mathbf{S} + \mathbf{N}| \text{ and}$$

$$Signal\ power = |\mathbf{S}| = |RSSI - \mathbf{N}|$$

where $\mathbf{S} + \mathbf{N}$ and $RSSI - \mathbf{N}$ consider wave phases.
  - (2) If a packet is lost in transmission, then even the RSSI estimate is not possible.

Expected Value PMF Algorithm

- Average noise = −90 dBm

Power trace:	-82	_	_	-87	-85	_	-86	-82	_	-81	_
SNR:	8	?	?	3	5	?	4	8	?	9	?
PRR:	.99	?	?	.1	.4	?	.2	.99	?	1.0	?
Expected lost:	0	?	?	9	1.5	?	4	0	?	0	?

Actual lost packets: 5; Expected lost packets: 14.5

- Fill in the gaps by sampling from the unfilled packet distribution:

Power Value:	-81	-82	-85	-86	-87	
# Remaining to fill:	0	0	1.5	4	9	(total 14.5)
% of those remaining:	0%	0%	10.3%	27.6%	62.1%	

- Filled in trace (one possible variation):

-82 <u>-87</u> <u>-86</u> -87 -85 <u>-87</u> -86 -82 <u>-87</u> -81 <u>-85</u>

- Run CPM on this trace to determine the Power value at any time.

Average Power Algorithm

`Power trace:   -82 _ _ -87 -85 _ -86 -82 _ -81 _`

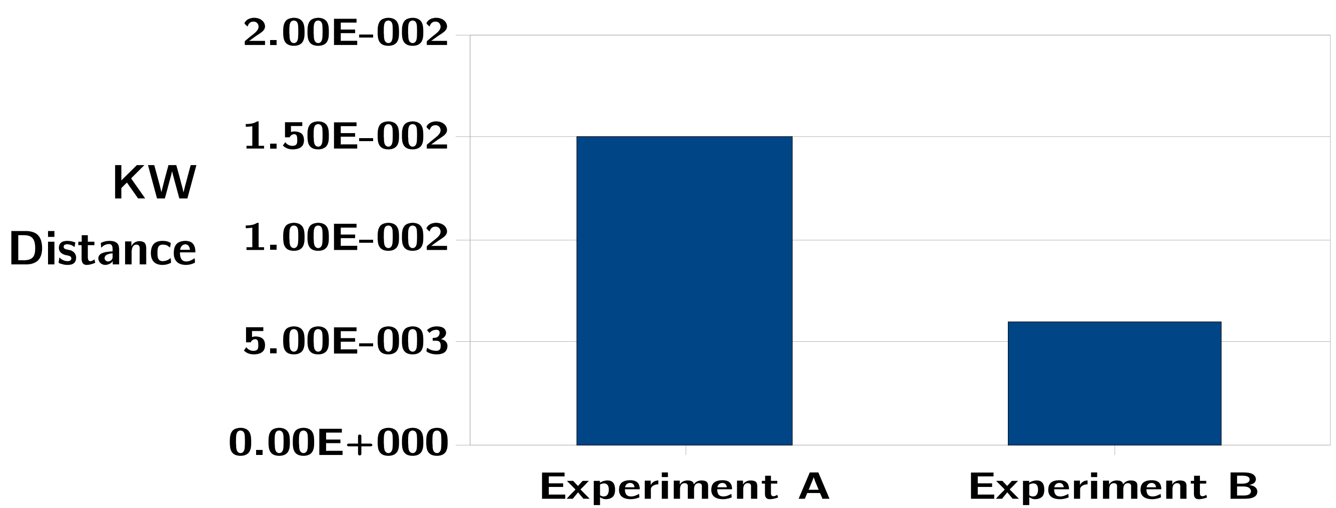
- Average power (rounded to integer) = −84 dBm
- Fill the power trace with average power value.
- Filled in trace:

-82 -84 -84 -87 -85 -84 -86 -82 -84 -81 -84

- Run CPM on this trace to determine the Power value at any time.

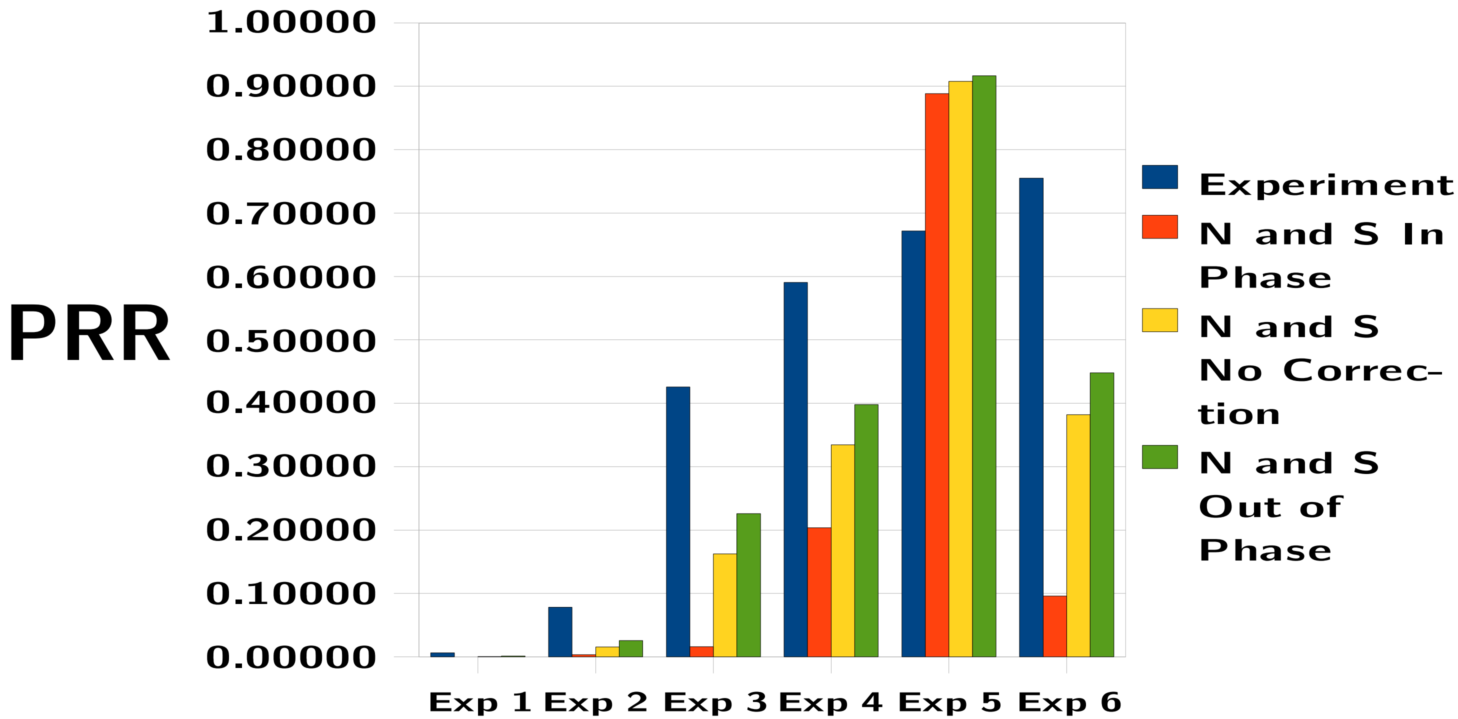
CPM is Effective for Modeling Power: KW Distance

- Kantorovich-Wasserstein (KW) distance is a formal quantification of the effectiveness of the CPM algorithm on a given experimental trace.
- KW Distances on the order of 10^{-2} or lower are excellent.
- Representative examples:



Impact of Phase Differences in Expected Value PMF Algorithm

- One of the first comparisons of TOSSIM simulations to real experiments.



Simulation of TinyOS Sensor Networks

Tal Rusak

tr76@cornell.edu

Outline

Introduction

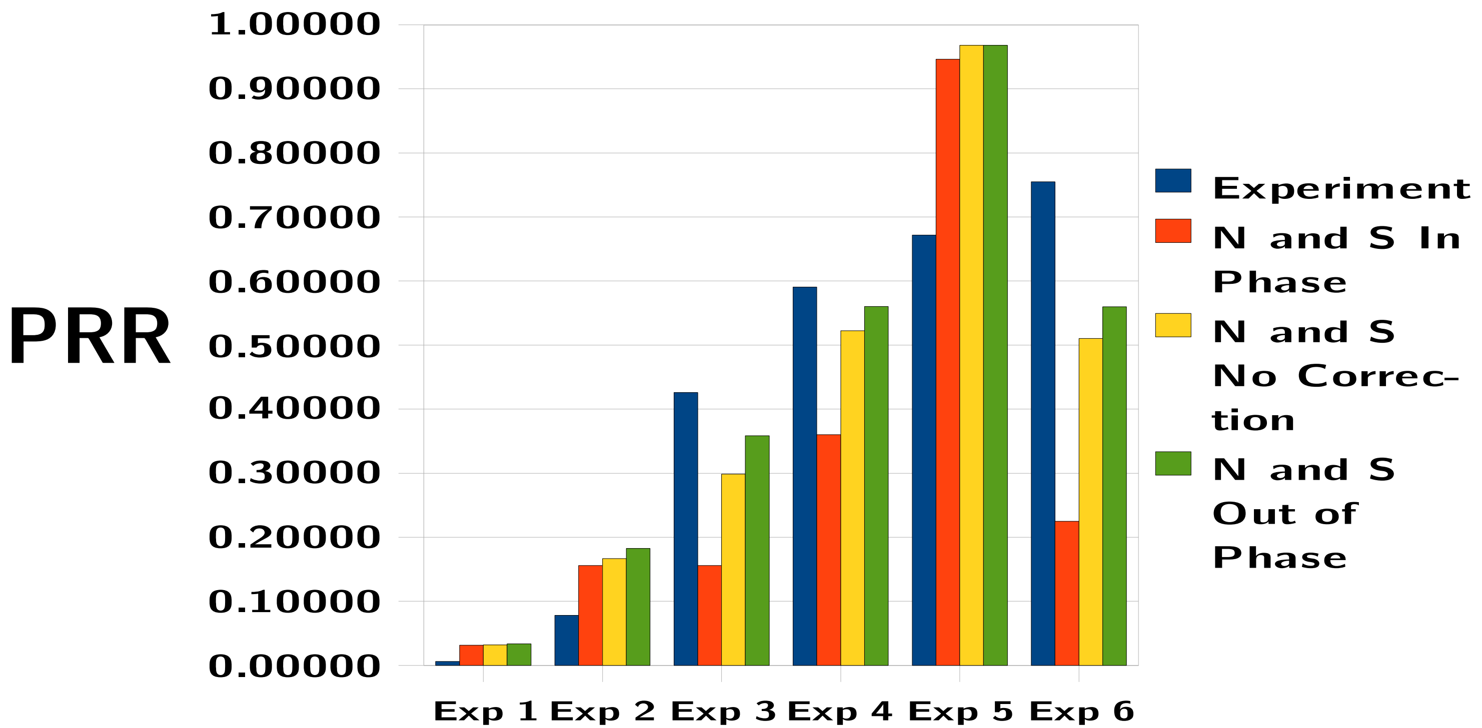
Applications

Simulation and Experiments

Results

Impact of Phase Differences in Average Power Algorithm

- One of the first comparisons of TOSSIM simulations to real experiments.



Tal Rusak

tr76@cornell.edu

Conclusions

- As expected, different assumptions work more effectively for different experiments.
- This observation corresponds to reality, since the phase of noise waves may differ in different environments and packets are lost for different reasons.
- ***Future work:*** Development of an automated optimization layer to predict the most reasonable assumptions for a given environment.
- ***Future work:*** I aim to develop theoretical models based on the information obtained from these studies in simulation.

Tal Rusak

tr76@cornell.edu

Thank you.

Questions?

Tal Rusak

tr76@cornell.edu

CPM Model for Trace Histories

- Scan noise trace, keeping a history of size k.
- For each signature of k prior noise readings, construct the probability distribution for the next reading.

0	2	1	2	0	2	2	0	0	1	1	1	1	2	0	0	0	2	9	0

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

CPM Model for Trace Histories

- Scan noise trace, keeping a history of size k.
- For each signature of k prior noise readings, construct the probability distribution for the next reading.

0	2	1	2	0	2	2	0	0	1	1	1	1	2	0	0	0	2	9	0

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

Tal Rusak

tr76@cornell.edu

CPM Sampling Demo

0	2				

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

30

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

Tal Rusak

tr76@cornell.edu

CPM Sampling Demo

0	2	2			

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

31

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

CPM Sampling Demo

0	2	2	0		

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

CPM Sampling Demo

0	2	2	0	0	

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

33

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

CPM Sampling Demo

0	2	2	0	0	1

signature	0	1	2	9
00	33%	33%	33%	0%
01	0%	100%	0%	0%
02	0%	33%	33%	33%
11	0%	66%	33%	0%
12	100%	0%	0%	0%
20	66%	0%	33%	0%
21	0%	0%	100%	0%
22	100%	0%	0%	0%

34

$k = 2$

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.

CPM Sampling Result

- Modeled trace is not the same as the experimental trace:

0	2	2	0	0	1

0	2	1	2	0	2	2	0	0	1	1	1	1	2	0	0	0	2	9	0

- This increases the randomness of simulation output and thus decreases the predictability of the simulation.
- This allows for substantial representative simulation.

HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling." In Proceedings of the IPSN, 2007.