

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



TAL RUSAK, Department of Computer Science, Cornell University (Class of 2009), Student Member SIGCSE
Advisors: Zygmunt Haas (Cornell University), Philip Levis (Stanford University)

TinyOS and Sensor Networks

- The TinyOS system is a suite of tools that are used in the implementation of wireless sensor networks.



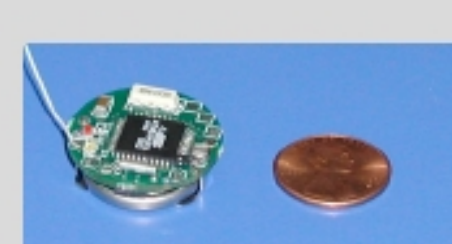
- Sensor Network Hardware:

Gateway:



Linux: MB of RAM
Active power: W
Sleep power: mW

Note:

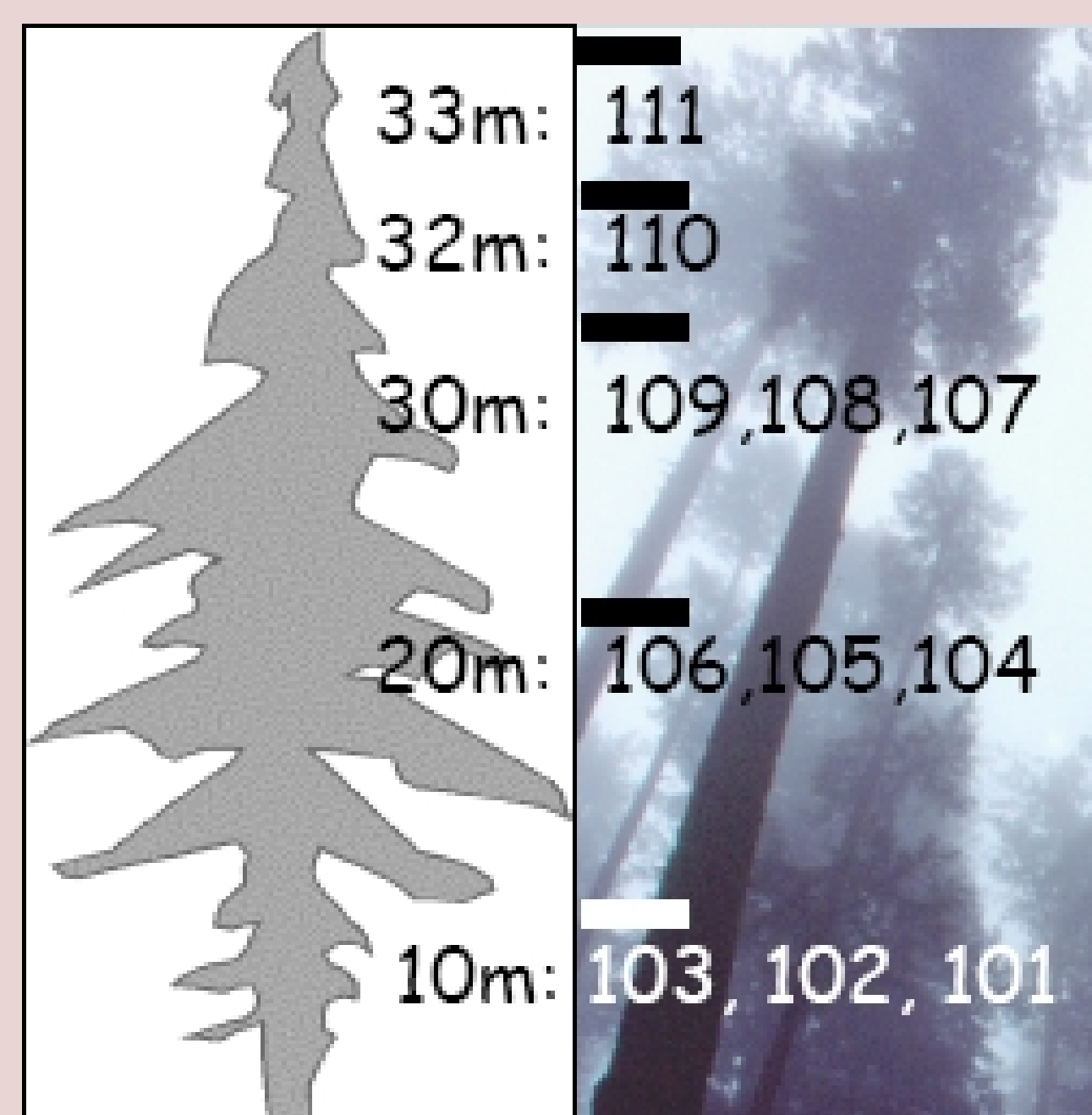


TinyOS: KB of RAM
Active power: mW
Sleep power: μ W

- Sensor Network Applications:

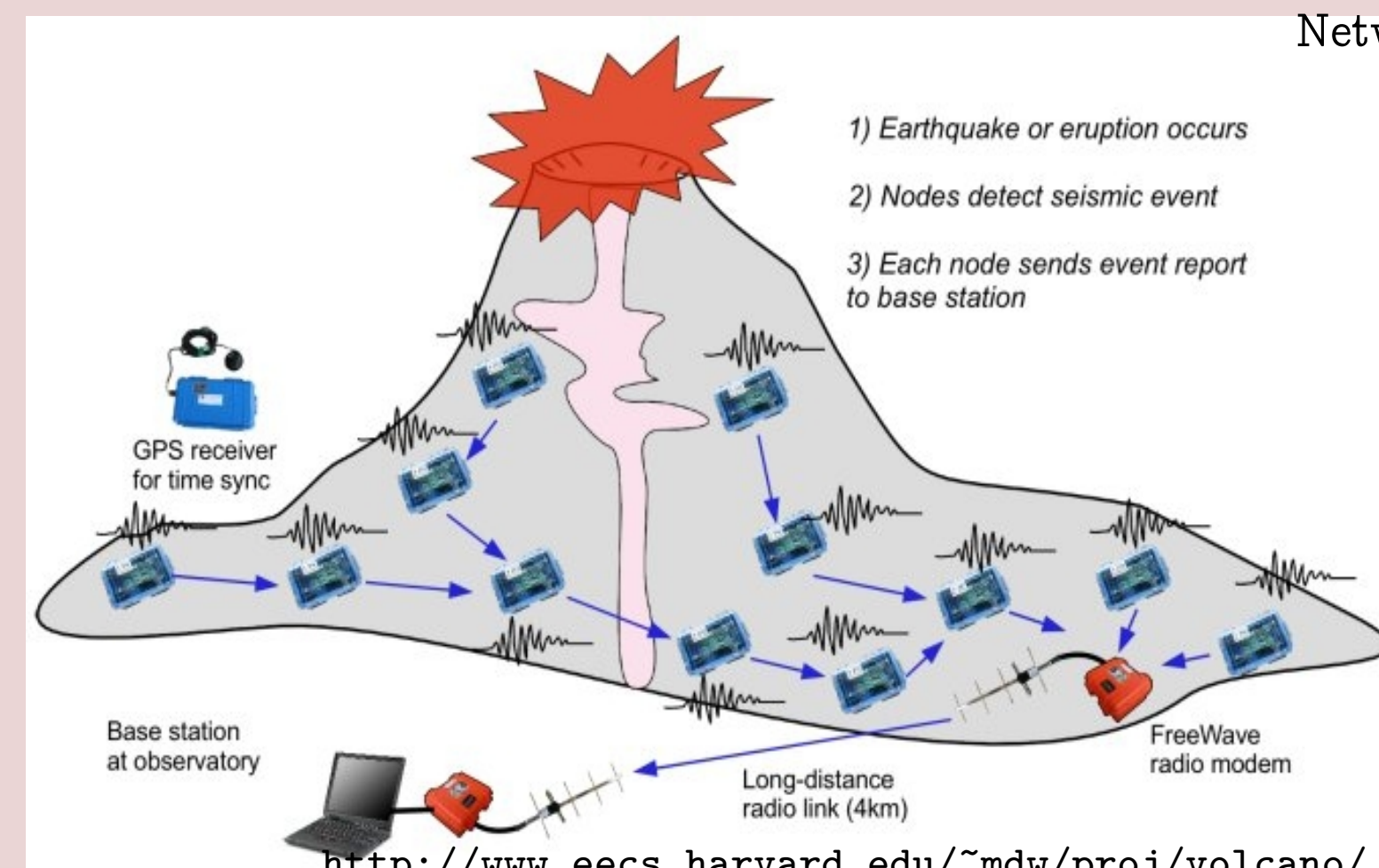


http://www.wired.com/wired/archive/11.12/network_pr.html
Great Duck Island, Maine: Monitoring of seabirds (2004), 58% packet yield



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

Redwood Microclimate Study (2005), 40% packet yield



Volcano Monitoring (2006), 68% packet yield

- TinyOS can be a link between information science, computer science, engineering, and biology.
- Reasons for low network yield:
 - Protocol conflicts/interference
 - Collisions and congestion induced loss
 - Cannot be known: not enough resources to log/debug in deployment application
- Robust simulation may lead to a better understanding of sensor network systems.
- We aim to develop theory from these results.

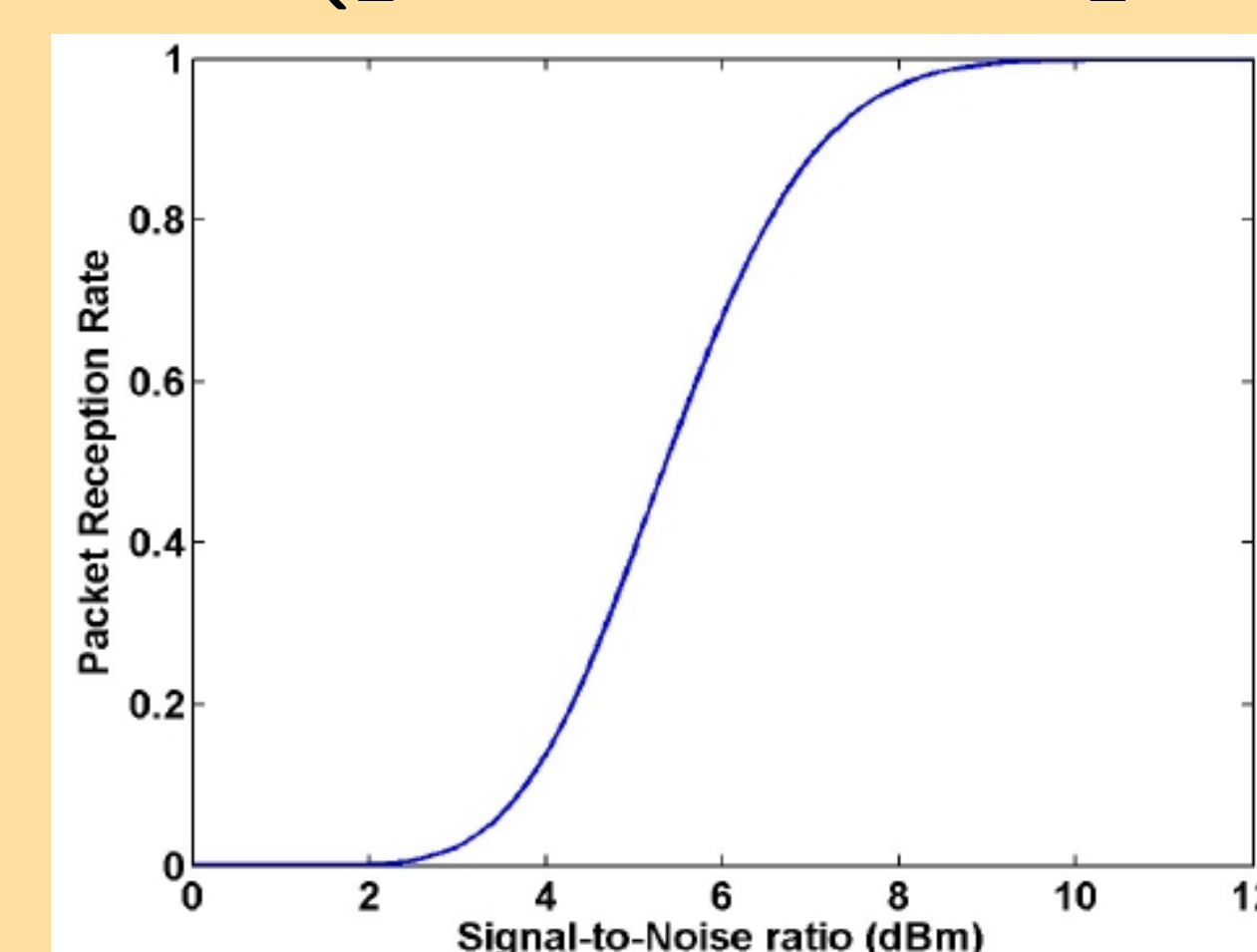
P. Levis, "Low-Power Sensor Networks: A Case Study in Seeking Distributed Dependability." Keynote at NSF HCCS-CPS Workshop, Nov. 30, 2006.

TOSSIM Simulator

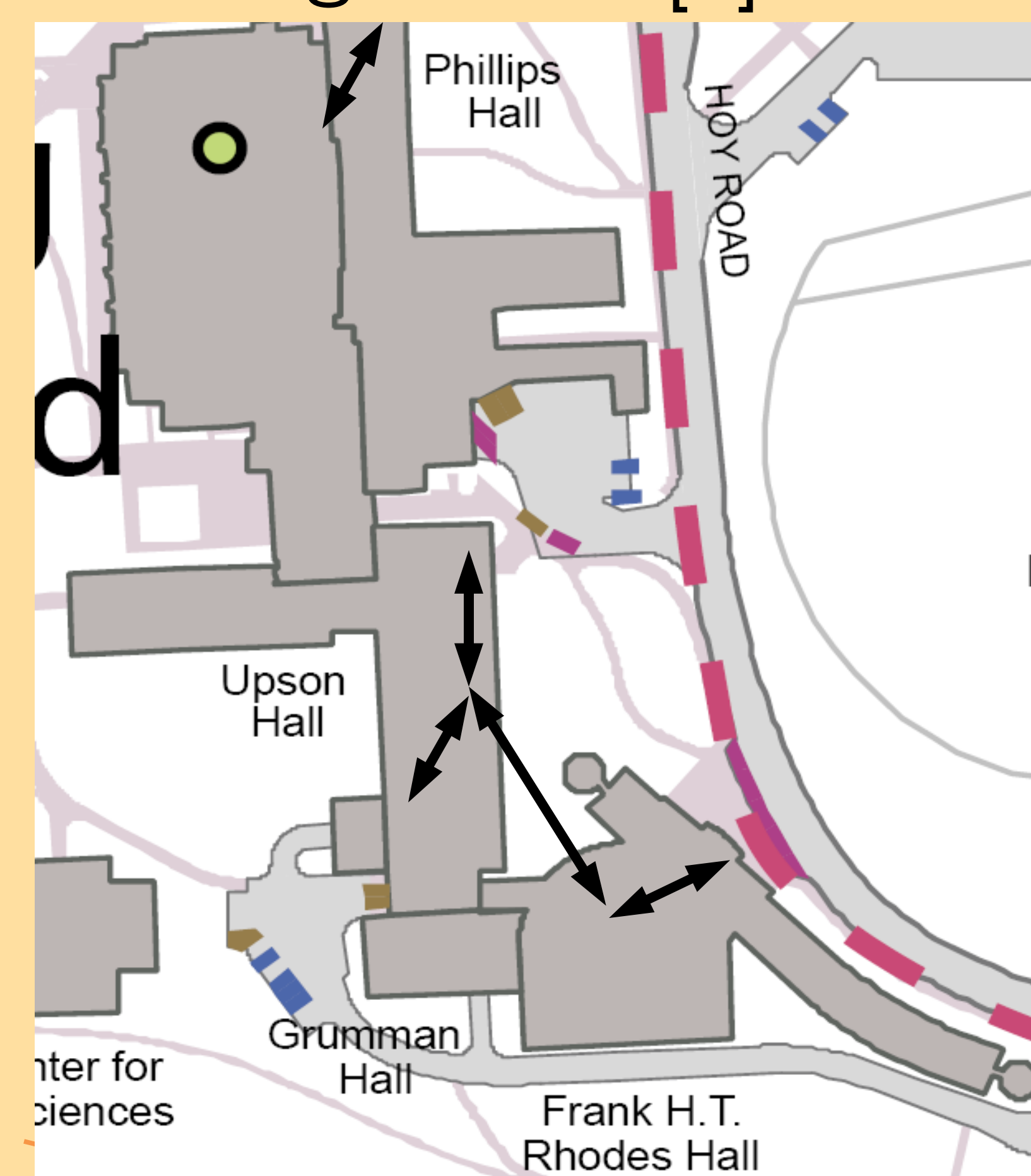
- TOSSIM is the TinyOS SIMulator.
- Physically-based simulation using an SNR-PRR curve: TOSSIM 2.0.1 calculate the probability of reception of each packet based on the following formula:

$$SNR = S - N$$

where SNR is the signal to noise ratio, S is the signal power (dBm), and N is the noise (dBm). The following curve [1] is used to determine PRR (packet reception rate):



- We collect data about S and N individually and to predict noise values using the CPM algorithm [1]. Locations:

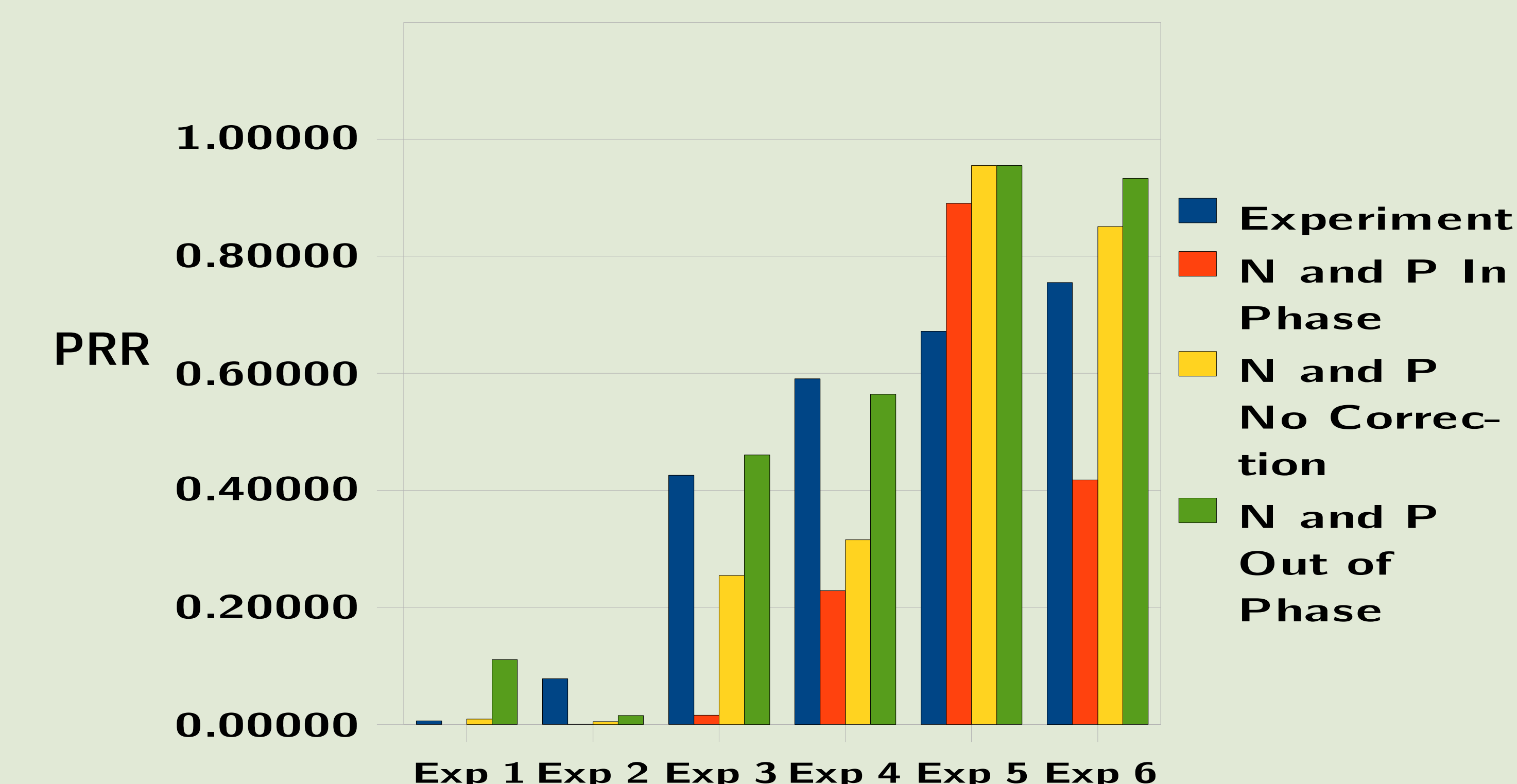


Computational Considerations

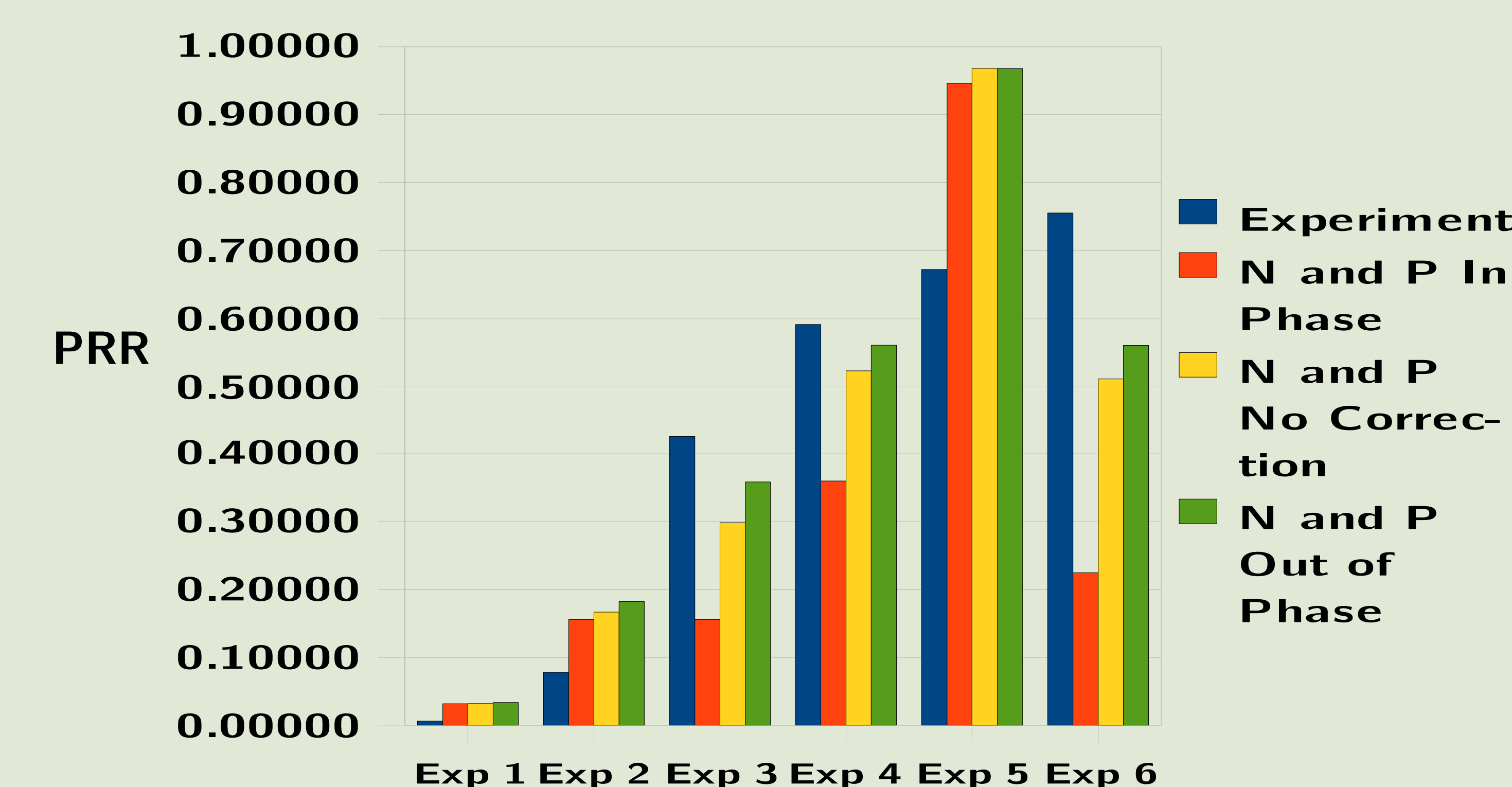
- To improve state space, careful quantization is introduced to input, but not to output, traces. A hash table is used for efficient storage of trace histories. Applications of this simulator are to develop more robust wireless protocols for applications with lost packets.

Results

- Power trace filled in from expected-value PMF:



- Power trace filled in from average-value of power:



- Different experiments are better approximated by different simulation methods.
- These are preliminary results.
- Future work:
 - Continue the analysis of the data measured using simulations following the CPM and constant average noise and power models.
 - Check correlation between simulation and measured data.
 - Audit simulator against application-level programs to test effectiveness.

[1] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks, pages 21–30, New York, NY, USA, 2007. ACM Press.