Realistic Performance Analysis of WSN Protocols Through Trace Based Simulation

Alan Marchiori, Lin Guo,
Josh Thomas, Qi Han

Toilers

Pervasive Sensing Systems

COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

Existing Approaches to Analyze WSN Performance

- Build a prototype system
  - Very good measure of performance, but costly, time consuming, and difficult to optimize
- NS-2, OMNeT++
  - Good "average" measure of performance under significant assumptions; steep learning curve
- TOSSIM, Avrora, and Cooja
  - Focus on validating functionality; not a good measure of performance

Our Vision: a Hybrid Approach

- Network connectivity information can be easily collected from a deployed WSN
  - This captures all real-world artifacts
  - Can be shared as well: http://wsn.eecs.berkeley.edu/connectivity/index.php
- Use these network profiles instead of synthetic models from an easy to use network simulator

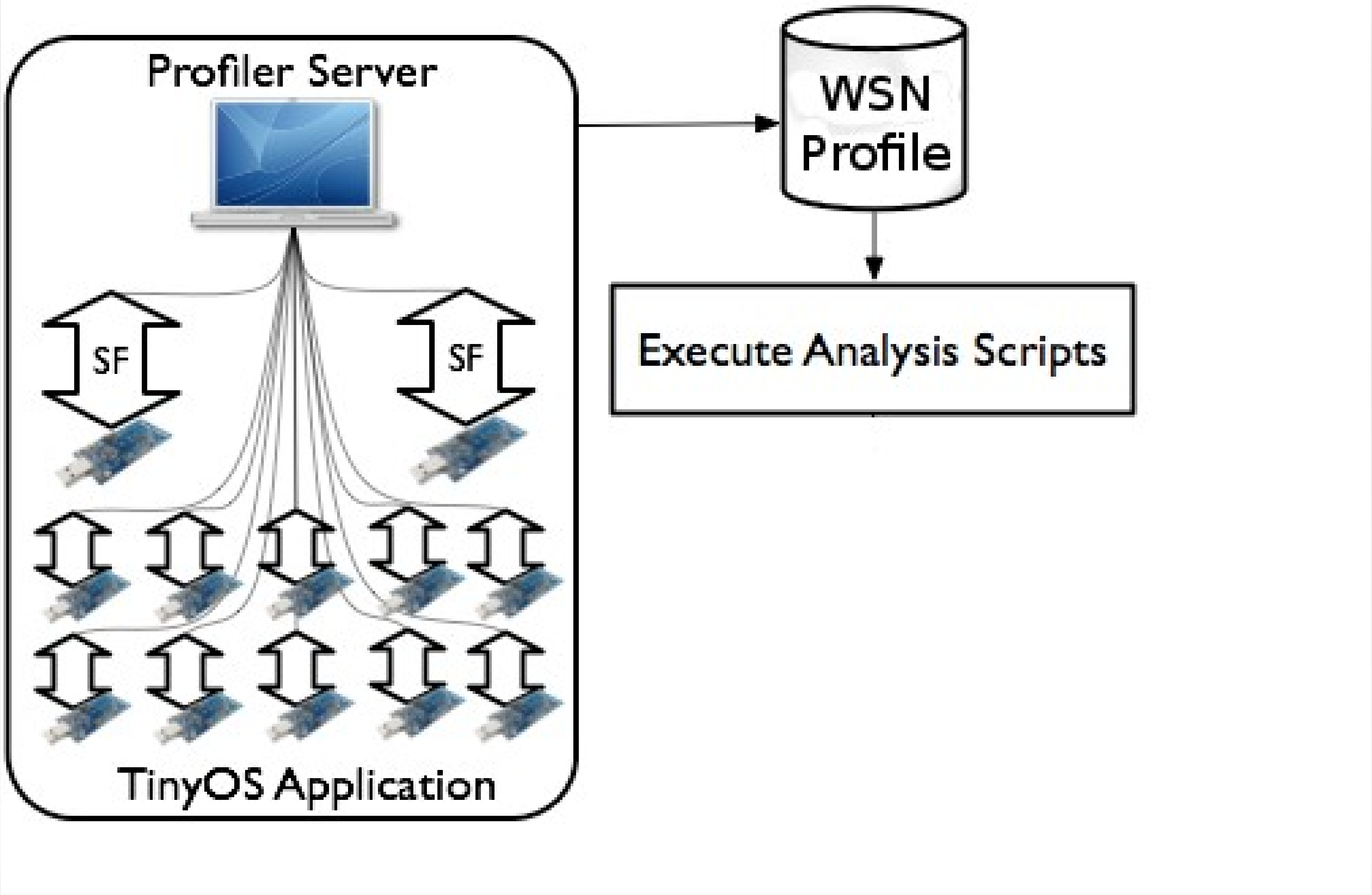
Our Solution

- Split the performance evaluation
  - Hardware
  - Software

Our Tools

- WSN Profiler
  - Automates the collection of network connectivity data
  - TinyOS application with a Java-based central server for coordination
- WSN SimPy
  - A network simulator that uses collected trace data as the basis for communication
  - Built on the discrete event simulator SimPy
    - http://simpy.sourceforge.net/

WSN Profiler: Architecture



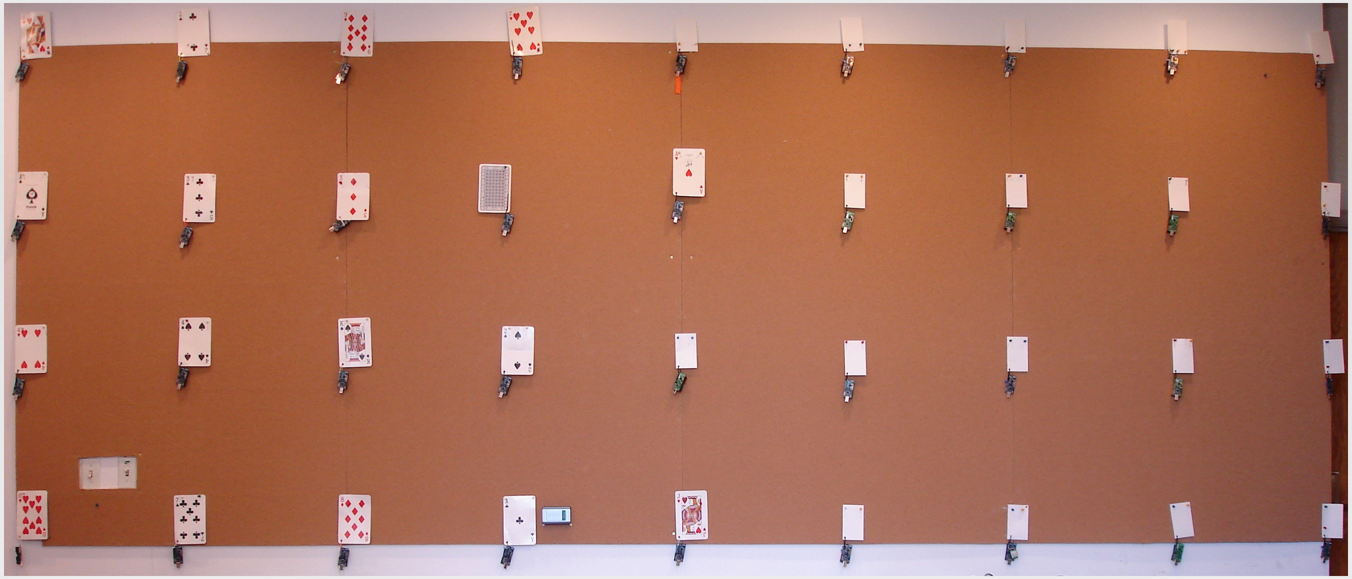
TinyOS Application

- Sender
  - Broadcasts a preset number of packets at a some frequency
- Receiver
  - Reports packet receptions to the profile server

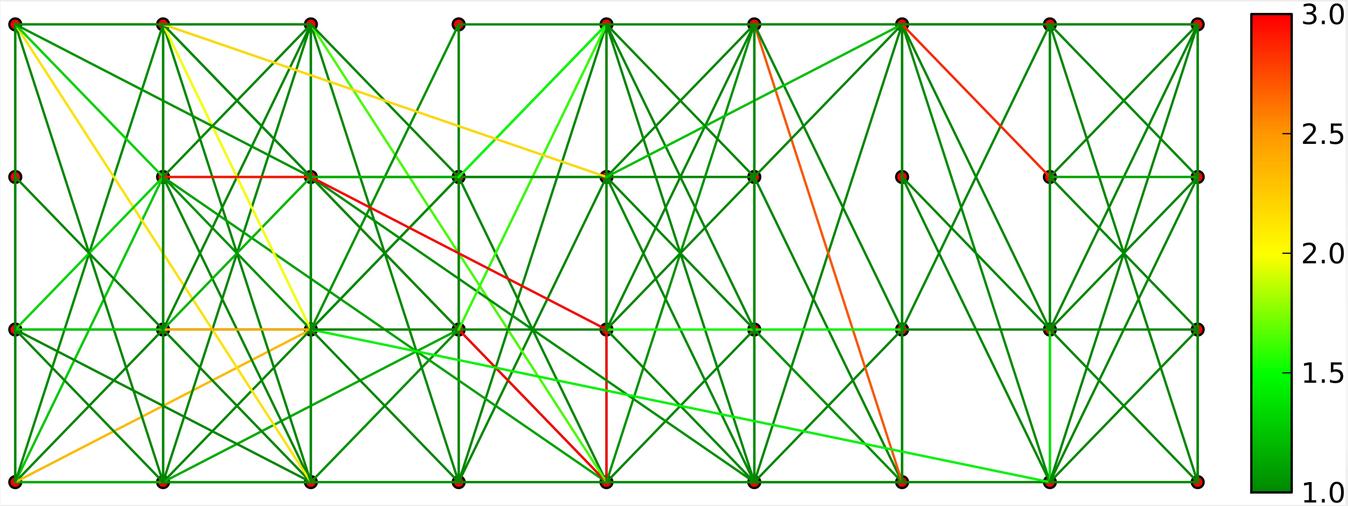
Profile Server

- Selects one node to be a sender at any time
- Records packet receptions to a log file
- Many configurable options:
  - Power level
  - Radio Channel
  - Number of transmissions
  - Packet transmission rate
  - Transmitted packet size

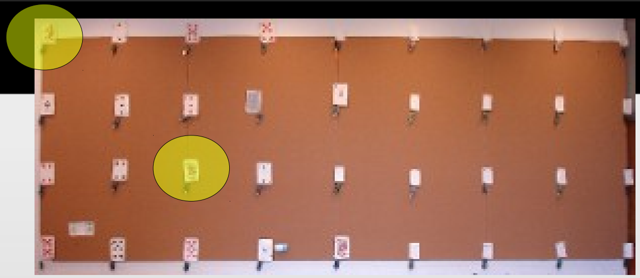
Network Visualization



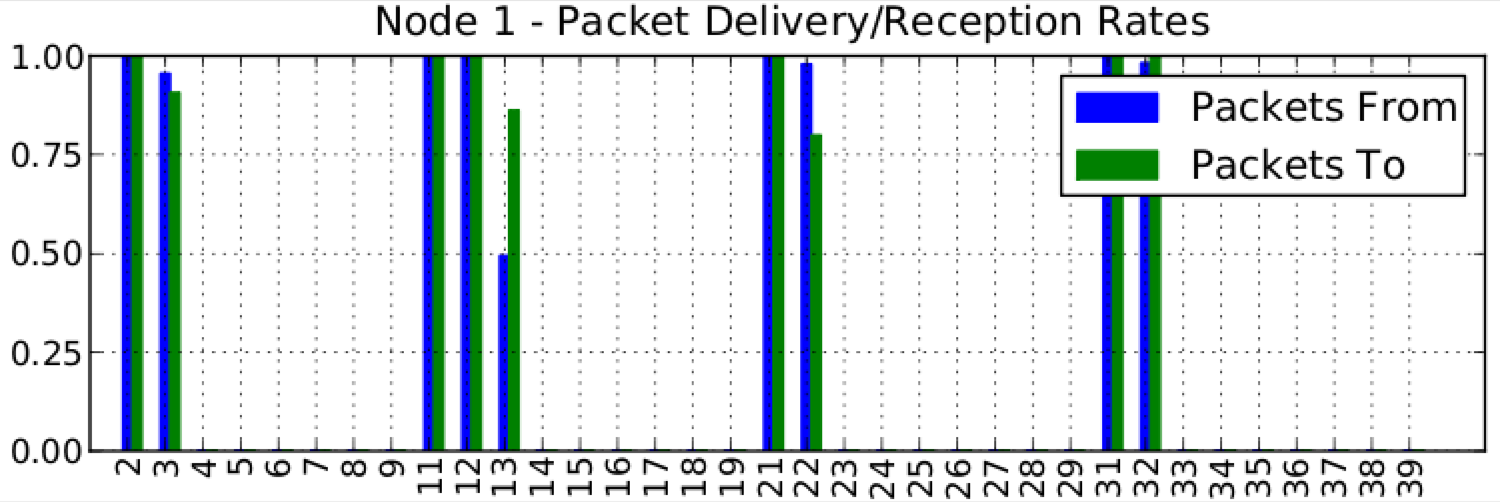
ETX Graph

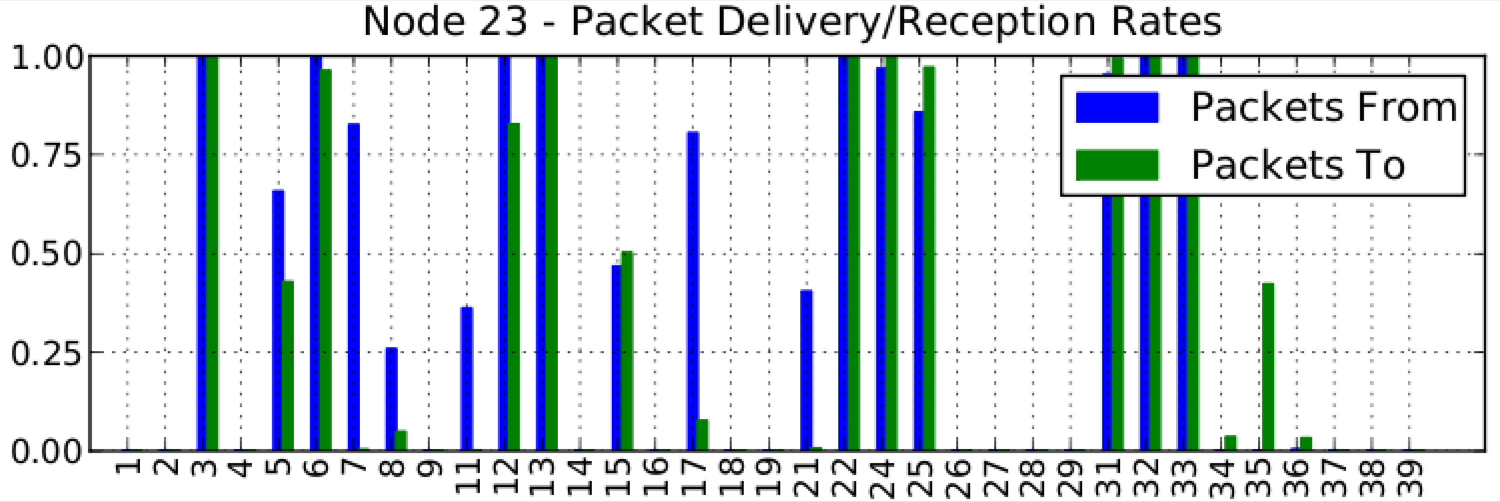


Network Visualization



- Per-node PDR

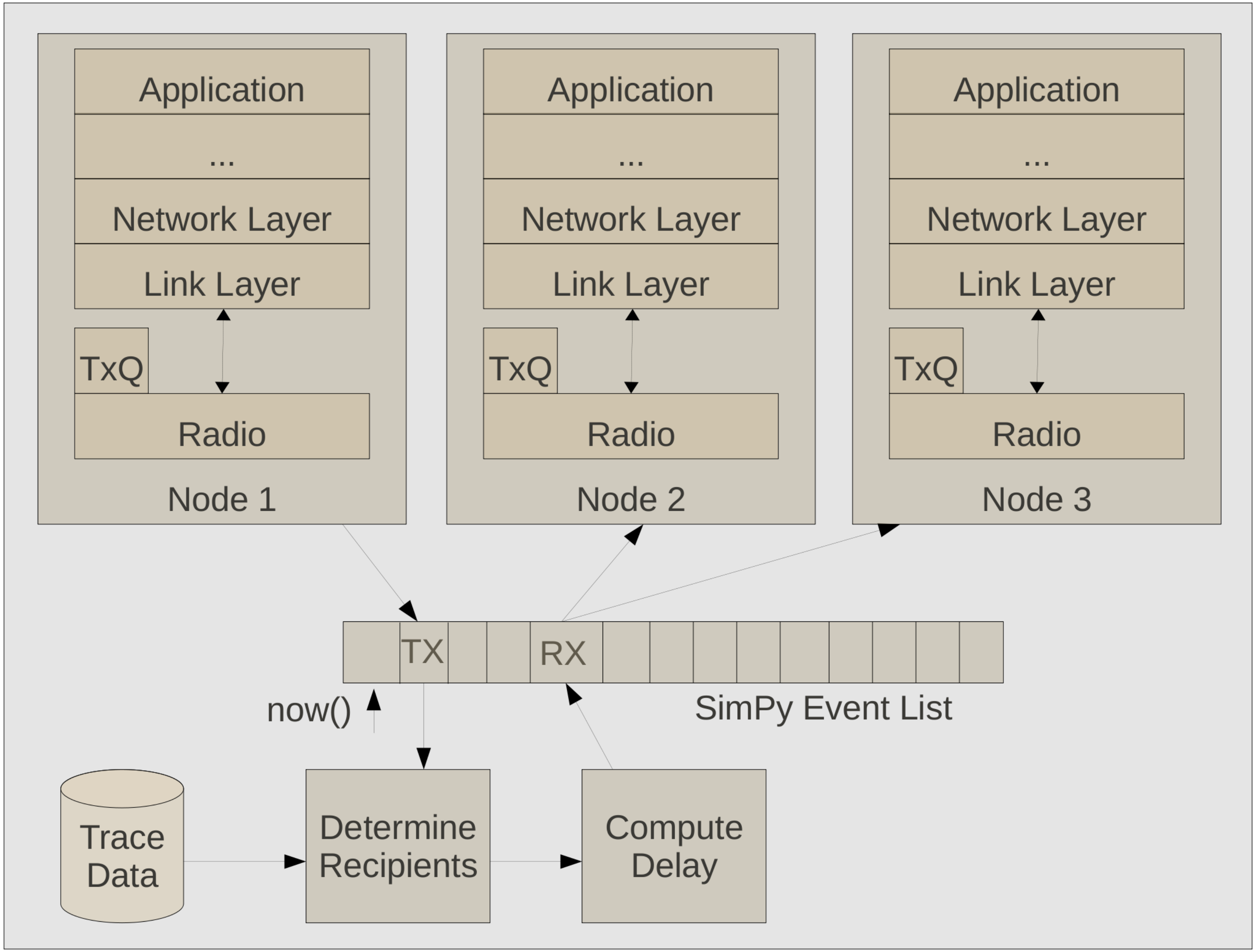




WSN SimPy

- SimPy
  - Discrete event simulator
  - Object oriented, process based
  - Standard Python source
- WSN extensions for SimPy
  - Single **network** object
  - Nodes are represented by individual objects

WSN SimPy Architecture



Selecting Recipients

- Each packet transmission from WSN Profiler is assigned a unique sequence number (included in the packet)
- Receptions can then be positively matched to transmissions
- To simulate a transmission an initially random sequence number is used to select recipients directly from the trace data
- Subsequent transmissions use sequential sequence numbers (wrapping to the beginning of the trace)

Packet Timing

- Assuming the IEEE 802.15.4 radio
  - 250 Kbit/sec (32 microseconds per byte) therefore transmission delay is computed from packet length
  - Ignore propagation delay: it is not significant over short distances
  - Processing delays are application specific; they can be simulated by the user

Radio Layer

- Half-duplex radio is simulated by the base node class i.e.:
  - A node cannot receive a packet while it is transmitting a packet
  - Packet transmission cannot start if the node is receiving a packet

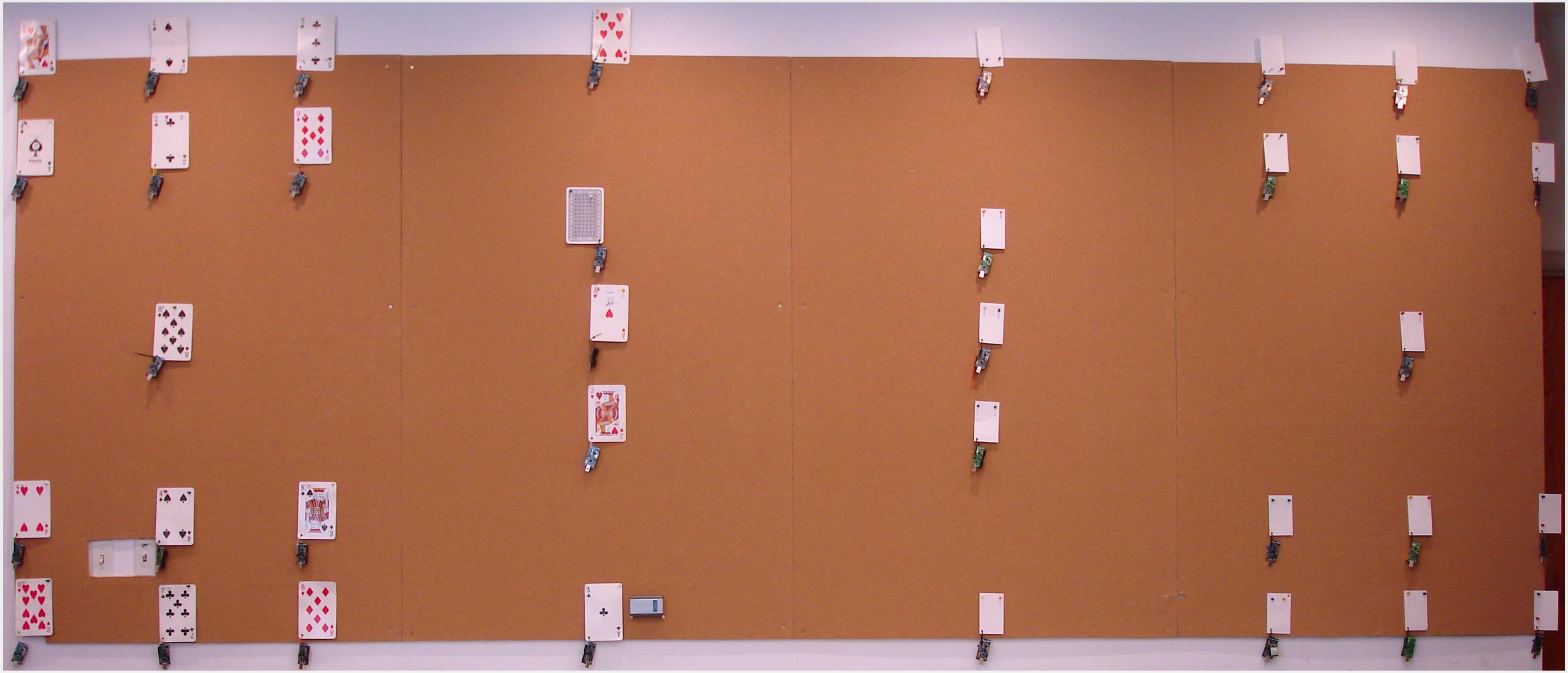
Collisions

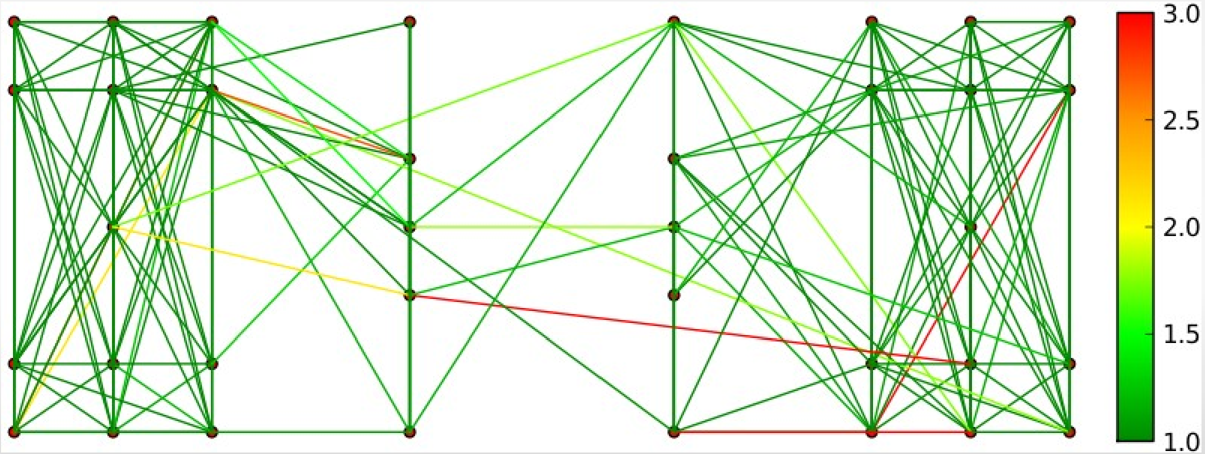
- Are also simulated by the base radio class of each node
- Currently assumes an idealized MAC layer
  - The network layer signals each node after the computed transmission delay
  - The radio layer inserts a 32 microsecond collision window on each packet reception

Sample Performance Evaluation

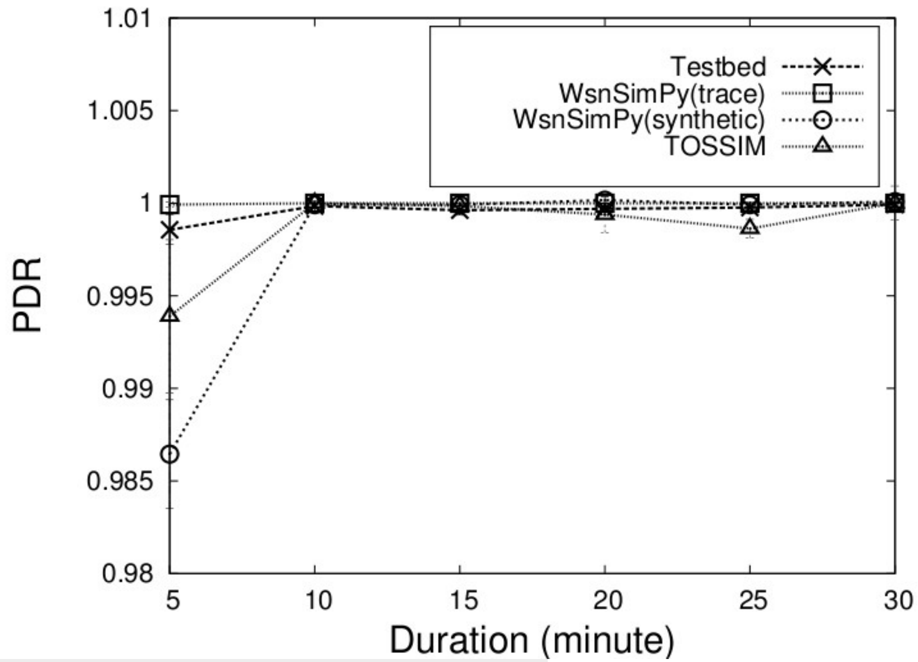
- Simple application using the collection tree protocol (CTP)
- Evaluated with
  - WSN SimPy
  - WSN SimPy(synthetic)
  - TOSSIM
  - Testbed
- Using two different topologies: grid and clustered

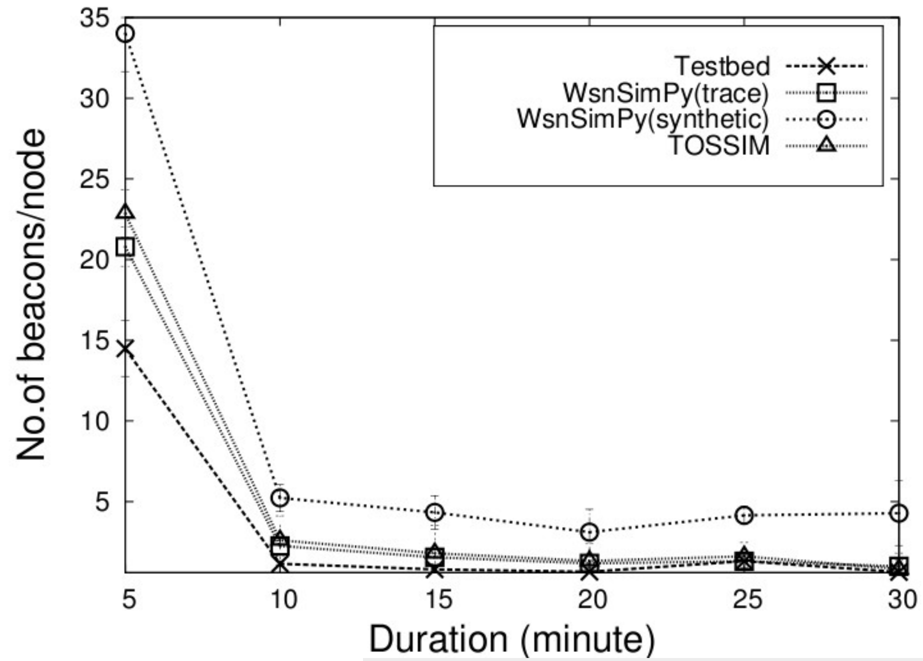
Clustered Topology

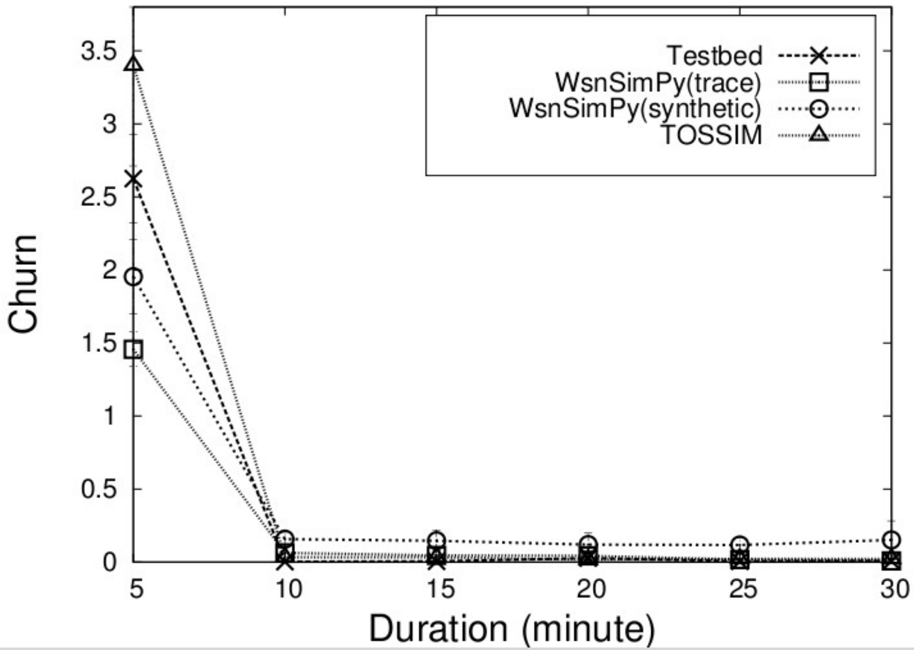




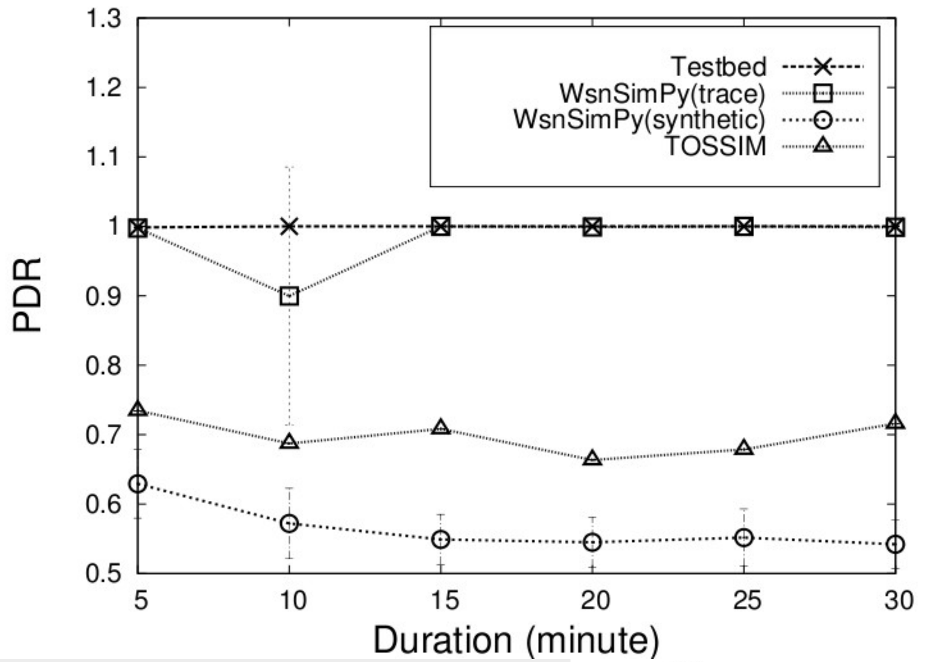
Grid: Performance

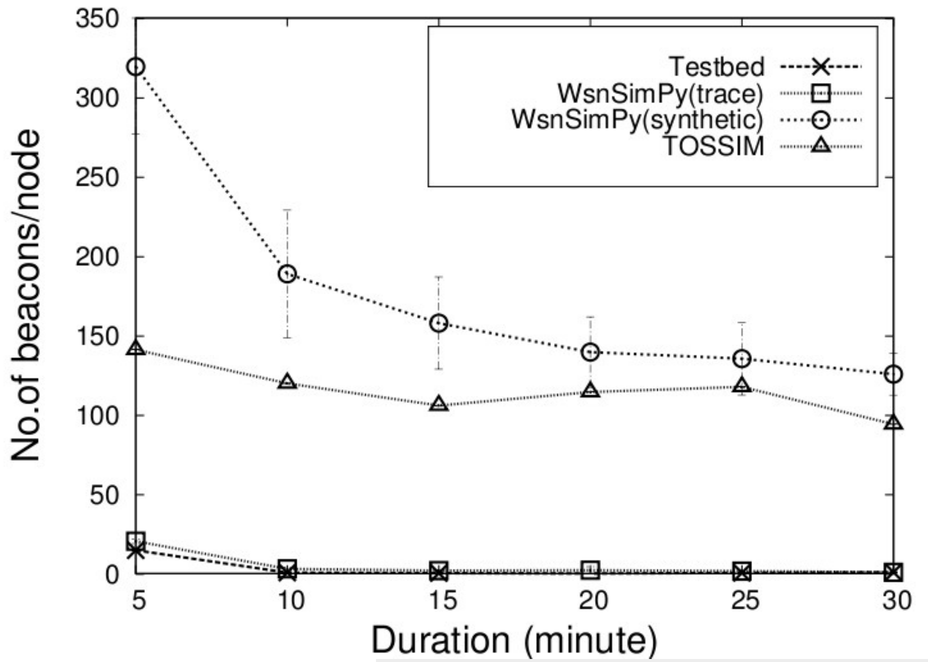


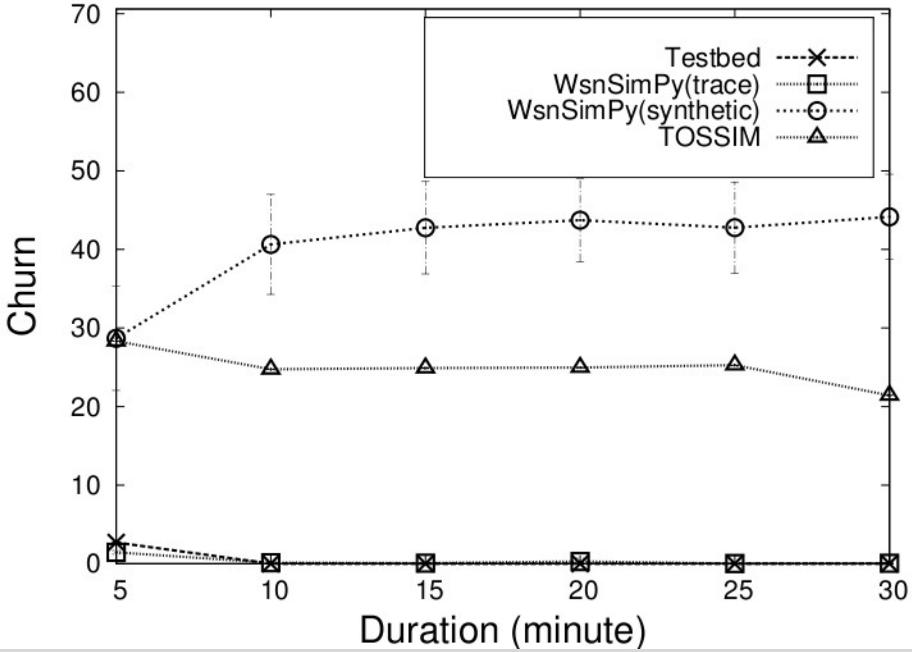




Clustered: Performance







Conclusion

- We present two tools
  - WSN Profiler – automates collecting connectivity information and visualizing the performance of deployed networks
  - WSN SimPy – an extension of SimPy to simulate WSNs using network profiles collected by WSN Profiler
- Simulated results of a sample application closely match real-world performance results from a testbed
- Both tools are available at http://thor.mines.edu/