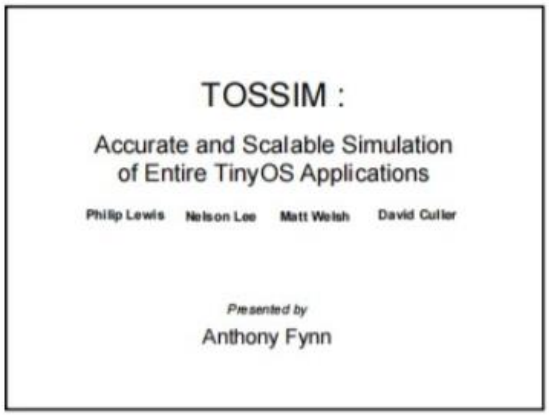
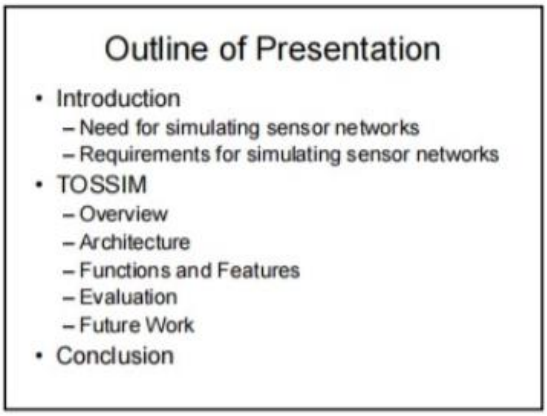
Practical 03

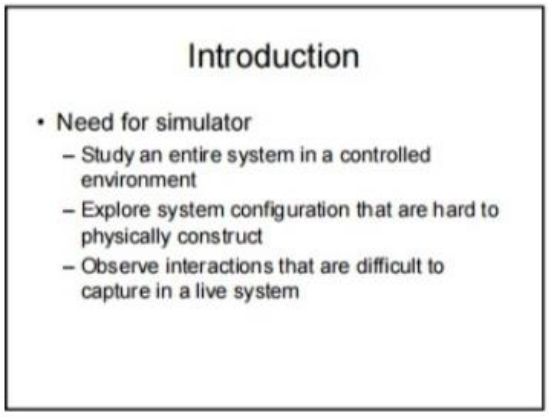
Understanding TOSSIM for

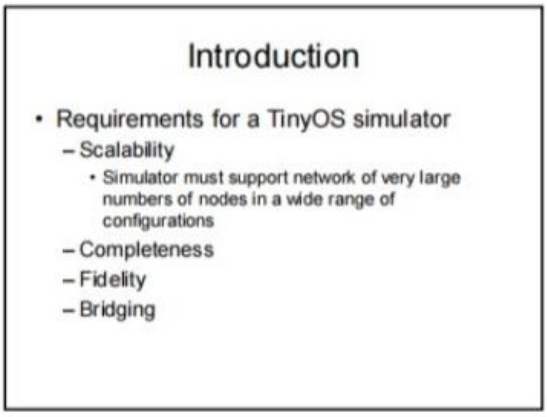
a. Mote-mote radio communication

b. Mote-PC serial communication









Introduction

- Requirements for a TinyOS simulator
  - Scalability
  - Completeness
    - Simulator must accurately capture all system interactions
  - Fidelity
  - Bridging

Introduction

- Requirements for a TinyOS simulator
  - Scalability
  - Completeness
  - Fidelity
    - Simulator must capture behavior of the network at a fine grain for evaluation and validation of code
  - Bridging

Introduction

- Requirements for a TinyOS simulator
  - Scalability
  - Completeness
  - Fidelity
  - Bridging
    - Simulator must bridge the gap between algorithm and implementation to allow developers to test code

Introduction

- Prior simulators for sensor networks
  - ns-2
    - A general network simulator
    - Does not meet the requirements of simulating an application such as TinyOS
  - EmStar
    - Simulate sensor network applications running on Linux
  - TOSSF
    - Simulates TinyOS code for a different simulation framework. Not evaluated
    - Authors claim techniques borrowed from TOSSIM

TinyOS Recap

- A TinyOS program is a graph of components
  - Components are mainly hardware resources such as ADC, sensors, computational entities
  - Components have 3 computational abstractions: commands, events and tasks
- TinyOS stack uses three network sampling rates – 40Kbps, 20Kbps and 10Kbps

TOSSIM - Overview

- Like TinyOS applications, TOSSIM is component based and event-driven
- Low-level components may be re-implemented to capture mote behavior at a fine grain
- Simulation is based on a virtual clock

TOSSIM - Overview

- Uses a directed graph to represent a network
  - Each vertex represents a node
  - Each edge has a bit error probability
  - Abstraction can capture most conditions

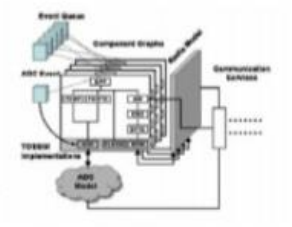
E.g.

Perfect transmission between *a* and *b*

Hidden terminal problem

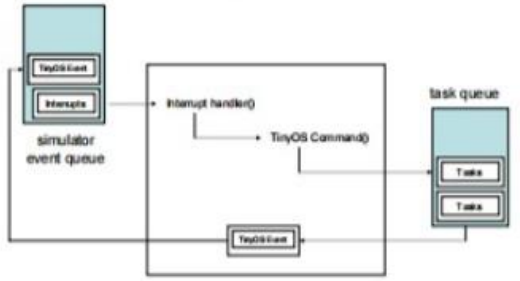
TOSSIM - Architecture

- Parts
  - Compiling component graphs into simulation infrastructure
  - Discrete event queue
  - Hardware abstraction of component
  - Radio and ADC models
  - Communication services



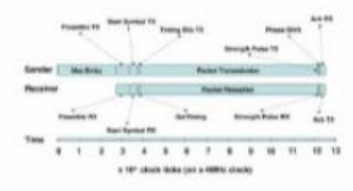
TOSSIM – Event Simulation

- Simulator event cycle



TOSSIM – Network Simulation

- Receiver radio clock need to be adjusted to synchronize with sender signal
- Adjustments to radio bit rates are made by changing the period between radio clock events
- Network stack is simulated at bit level

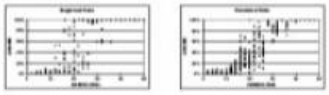


TOSSIM - Features

- Single compiler support for simulation and hardware
  - Added a compiler option to the nesC Compiler to compile an application for simulation instead of mote hardware
- Visualization tool (*TinyViz*)
  - Allows simulations to be visualized, controlled and analyzed through a GUI
  - Allows for *plugins* for users to interact with simulation

TOSSIM - Evaluation

- Fidelity
  - Radio noise
    - Radio loss graphs generated physical topologies based on empirical data from real-world network
    - Conclusion: Shows that TOSSIM's bit-error mechanism allows for implementation of complex models



(a) Empirical (b) Simulated

TOSSIM - Evaluation

- Fidelity
  - Packet-level interactions
    - Investigated effects of transmission rate and network density on reliability of packet transmission
    - Simulated a TinyOS application at three network densities
    - Conclusion: The denser the network and higher the transmission rate resulted in more packet loss respectively

TOSSIM - Evaluation

- Fidelity
  - Subtle race conditions
    - Race condition that occurred in the Media Access Control component was uncovered
    - Conclusion: Condition would not have been uncovered if TOSSIM did not have the ability to simulate at a very fine granularity

TOSSIM - Evaluation

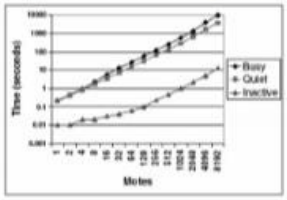
- Completeness
  - Surge
    - Wide range of anomalous behavior uncovered
      - Packet overflow due to interactions between transmission policy and Active Messaging level acknowledgements
  - TinyDb
    - SQL-based interface to query sensor network
    - Reported that TOSSIM provided a useful environment for prototyping new features
  - Evaluation mainly based on one application

TOSSIM - Evaluation

- Bridging
  - TOSSIM compiles directly from TinyOS code
  - Simulation code is almost identical to code running on hardware
  - Users can transition between simulation and real-world network
  - Users can step through simulation code with traditional debuggers without disrupting mote execution

TOSSIM - Evaluation

- Scalability
  - TOSSIM is scalable for large number of nodes and more complex application



Simulations of three application of varying complexities

TOSSIM – Future Work

- Researchers are exploring connecting real-world motes to the TOSSIM networking model
- Researchers are modifying TOSSIM so that multiple running copies can communicate through a network proxy
- Energy consumption need to be captured
- TOSSIM should be made less application specific

Conclusion

- TOSSIM captures a good deal of a real-work sensor network running a TinyOS application
- The effectiveness of TOSSIM cannot be generalized for all sensor networks
- It seems authors evaluated sensor network simulation based on the capabilities TOSSIM