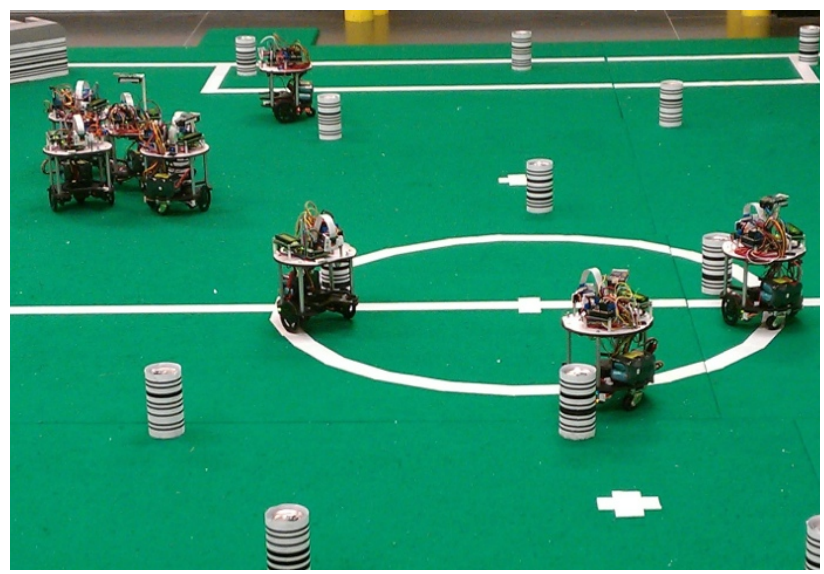
Supporting Mobile Swarm Robotics in Low Power and Lossy Sensor Networks

Kevin Andrea kandrea@gmu.edu
Robert Simon simon@gmu.edu
Sean Luke sean@cs.gmu.edu

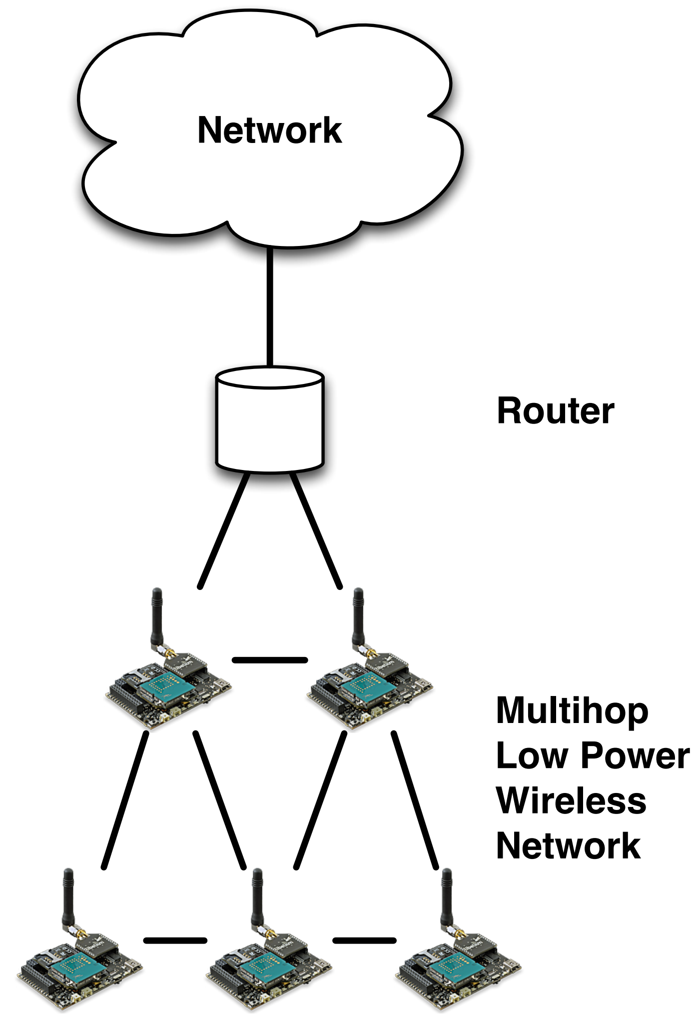
Department of Computer Science
George Mason University

Washington, DC



Supporting Self-organizing Sensor Swarm Systems

- **Wireless Low Power** and **Lossy Networks** (**LLNs**) are becoming ubiquitous
- They form a **key** portion of the Internet-of-Things landscape
- A tremendous source for providing situational awareness for sensor swarms
- ***Question:***
  How can we make LLN protocols work with mobile sensor swarms?



LLN Characteristics

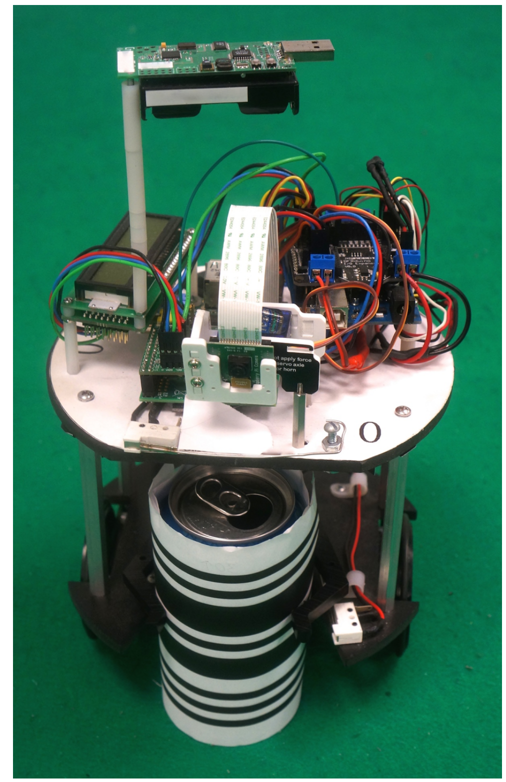
- Targeted class of nodes have a multi-point to point communication pattern
- Nodes sense and process data and communicate up and down a tree
- Meant for **very** resource constrained hardware
    - Tmote Sky class nodes are **ultra-low power** wireless hardware
    - 16 bit, 8 Mhz MSP430 microcontroller
    - 10K RAM, 48K Flash

MoRoMi: **Mo**bile **Ro**botic **M**ulti-s**i**nk

- Meant to support swarms of autonomous mobile robots.
- An intentionally thin wrapper layer over **RPL** within an open swarm robotics architecture
  - *RPL:* IPv6 Routing Protocol for Low-Power and Lossy Networks.
    IETF Standard (RFC 6550).
- **Goal:** to ***maintain*** compliance with evolving networking standards **while** providing support for complex, dynamically changing environments such as mobile swarm robotics.

The FlockBots

- Off-the-shelf open robot architecture
- **Controllers:**
  Arduino Uno or Mega, Raspberry Pi 2
- **Sensors:**
  Five Sharp IR infrared range sensors, two bump sensors, wheel encoders, tilt-servoed camera
- **Effectors:**
  Two wheels (differential drive), gripper, push bar, camera servo, display

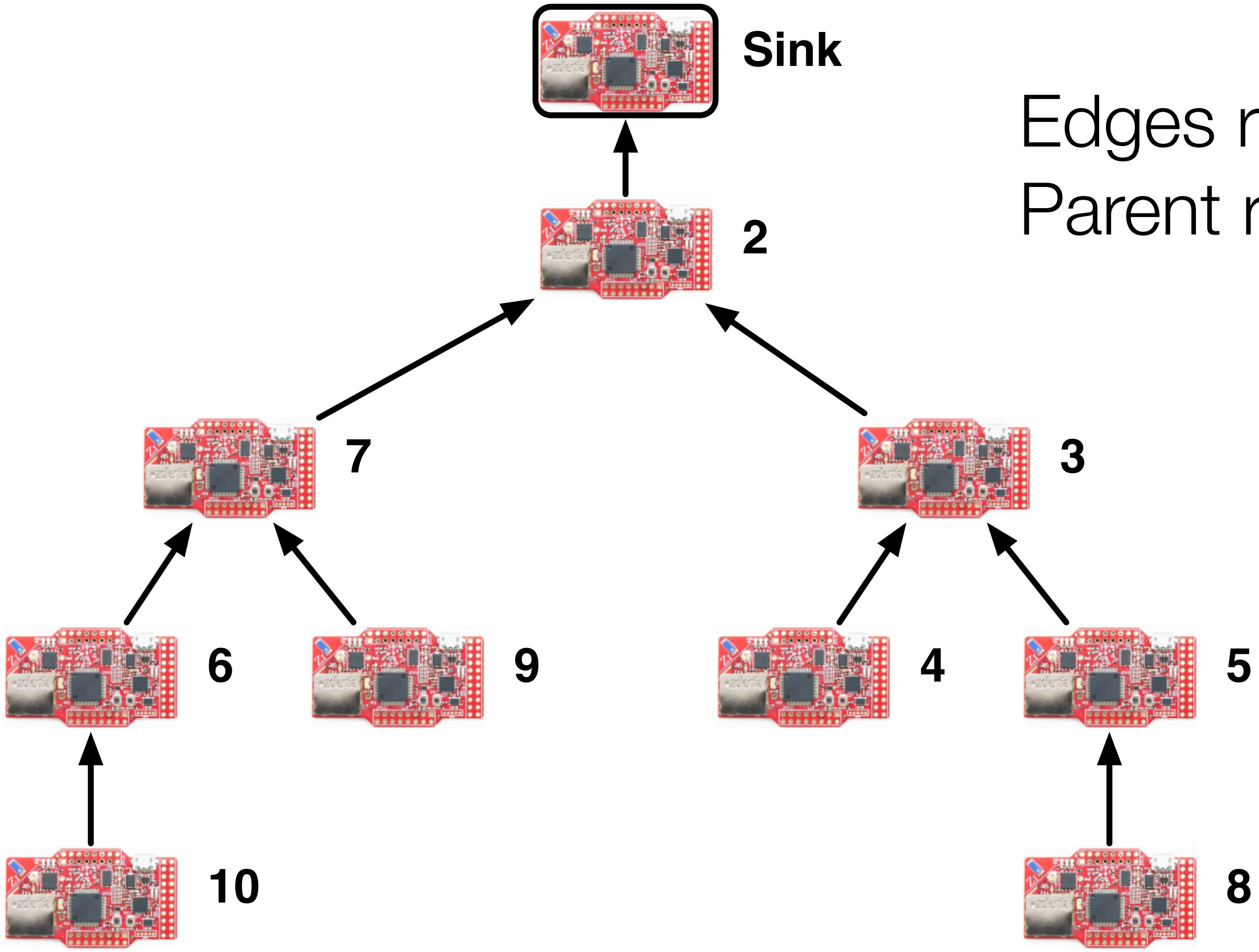


- **Uses an attached Tmote Sky wireless sensor mote for LLN interaction**

Integrating Robotic Swarms into RPL

- Wireless channel quality is highly variable
- Due to high packet loss rates routes break frequently
- LLN routing protocols constantly update their best path up a tree to the **sink** (the root)
- Uses sink-oriented gradient routing and Directed Acyclic Graph (DAG)
- Each DAG instance is specified by its sink node
- Sinks are advertised to the network via a Destination Oriented DAG Information Object (DIO)
- **Robotic swarms must interact with LLNs via a new type of DAG**

RPL Network Destination-Oriented Directed Acyclic Graph (DODAG)



Edges represent Parent relationship

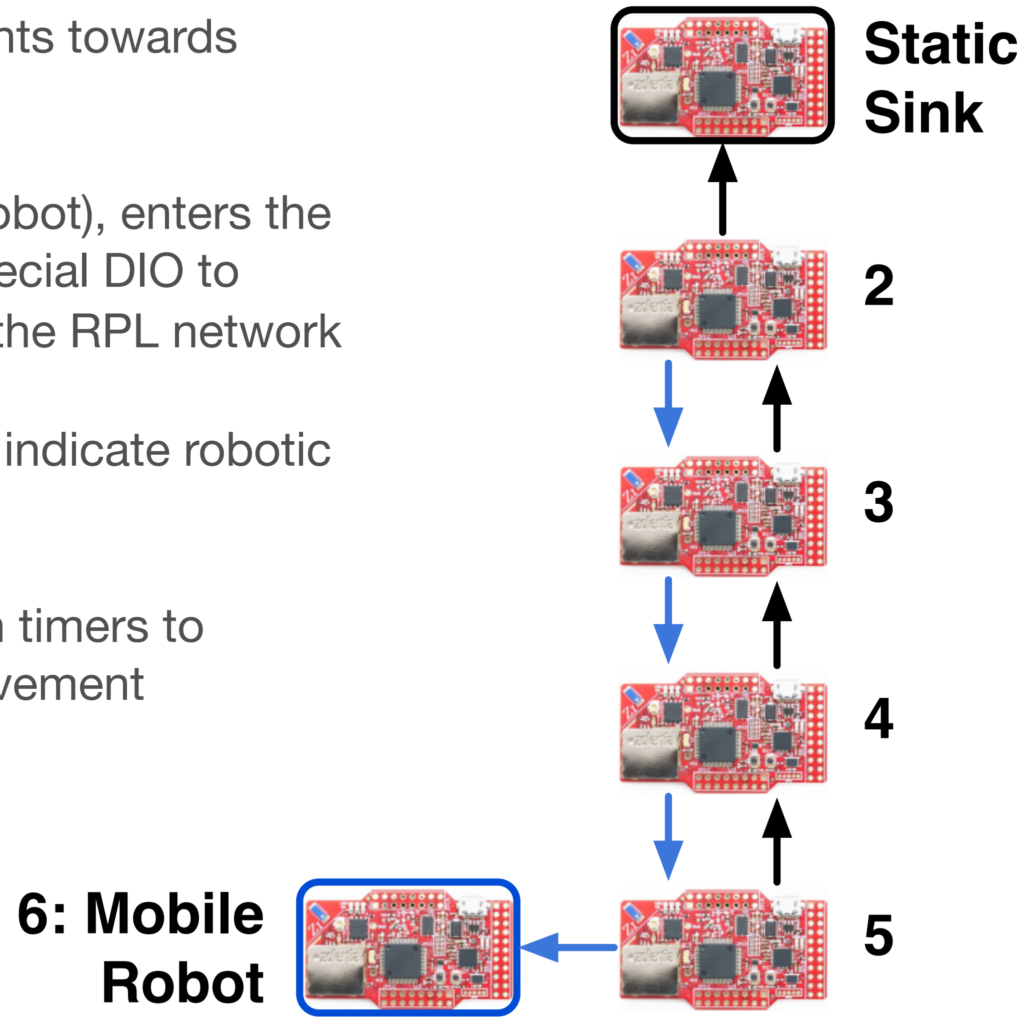
Senders

Basic Challenge

- RPL and similar protocols assume a single sink
  - We require multiple mobile sinks
- LLN protocols like RPL often experience long convergence times
  - Mobile nodes need to limit these times
- To ensure reliability RPL uses a **trickle timer** for automatic DIO retransmission
  - May pose dynamic convergence issues

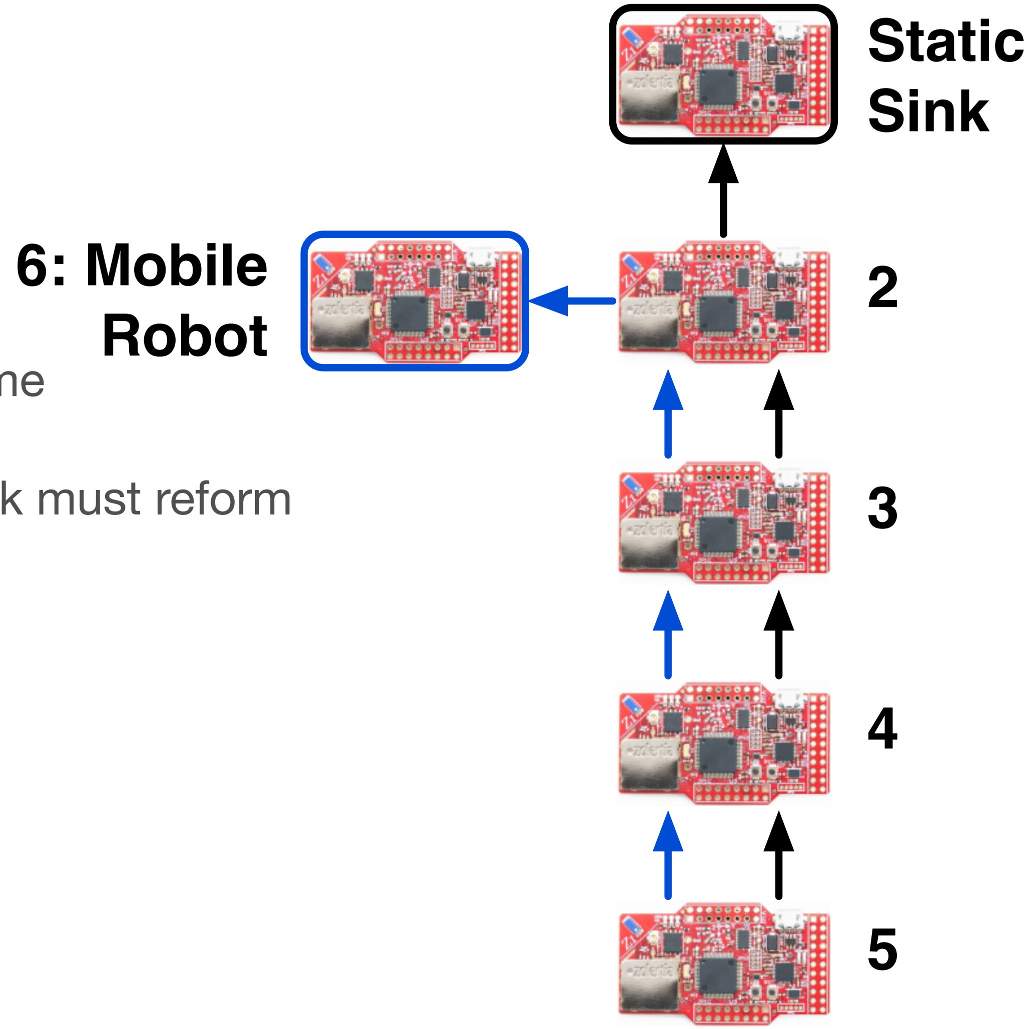
Routing in MoRoMi

- Establish, on-demand, gradients towards mobile sinks.
- As each new sink (e.g, each robot), enters the environment, it transmits a special DIO to announce a new **Instance** of the RPL network
- Frequent tree parent changes indicate robotic mobility
  - Modify the retransmission timers to correctly support this movement



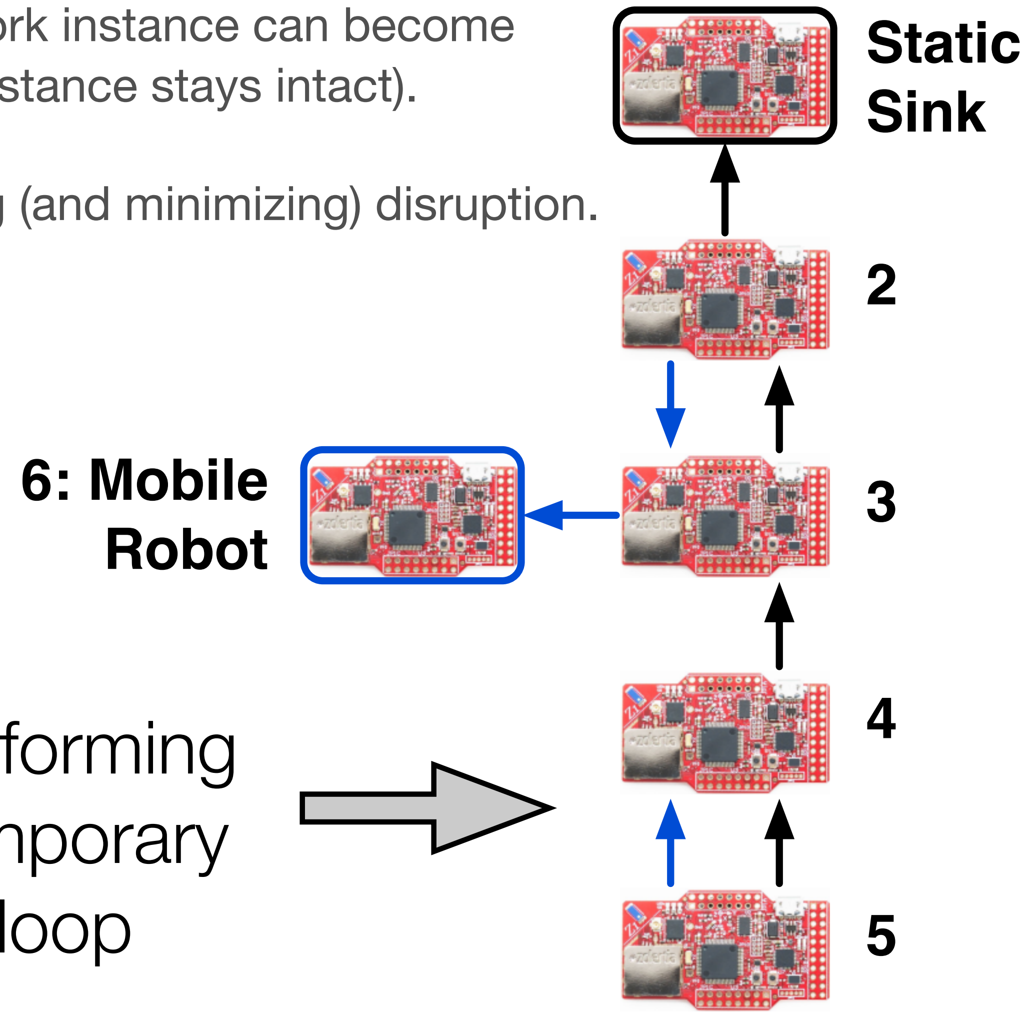
MOROMI Network Scenario

- Typical scenario:
  - Robot enters field
  - Robot sends DIO
  - Robot dwells for some time
  - Robot moves and network must reform



Disruption Modes

- As the robot moves, its network instance can become **disrupted** (the static sink's instance stays intact).
- We are interested in analyzing (and minimizing) disruption.



Network reforming causes temporary routing loop

Evaluation

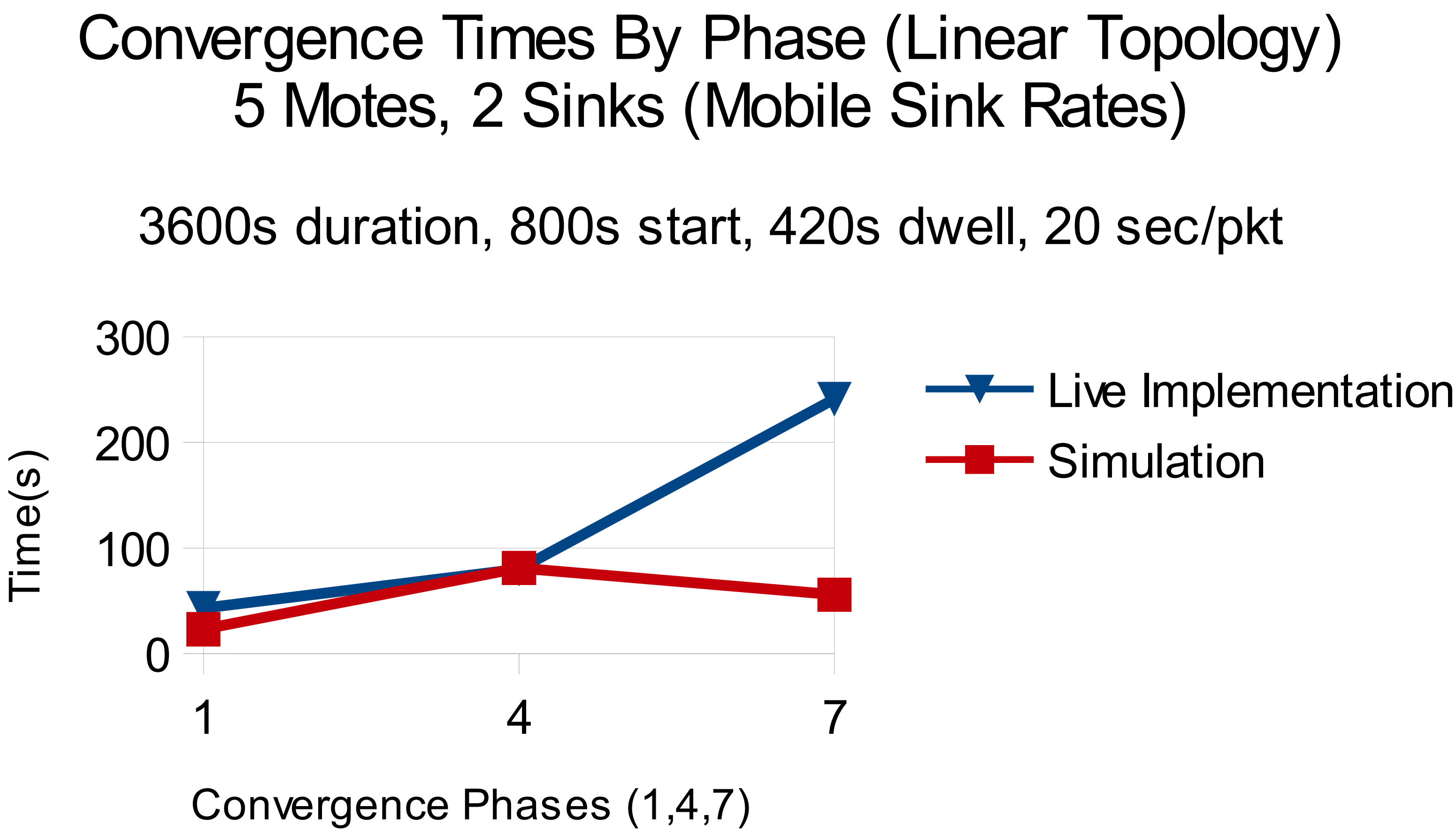
- Performance is judged by
  - Time for routing to stabilize
  - Packet Delivery Ratio (PDR)
- Two evaluations:
  Physical Implementation and Simulation
  - For simulation, used **Cooja** and wrote a new tool, called **Tamara**, to rapidly generate and evaluate swarm-LLN interactions
- Considered both a linear network and a star network

Linear Network Evaluation

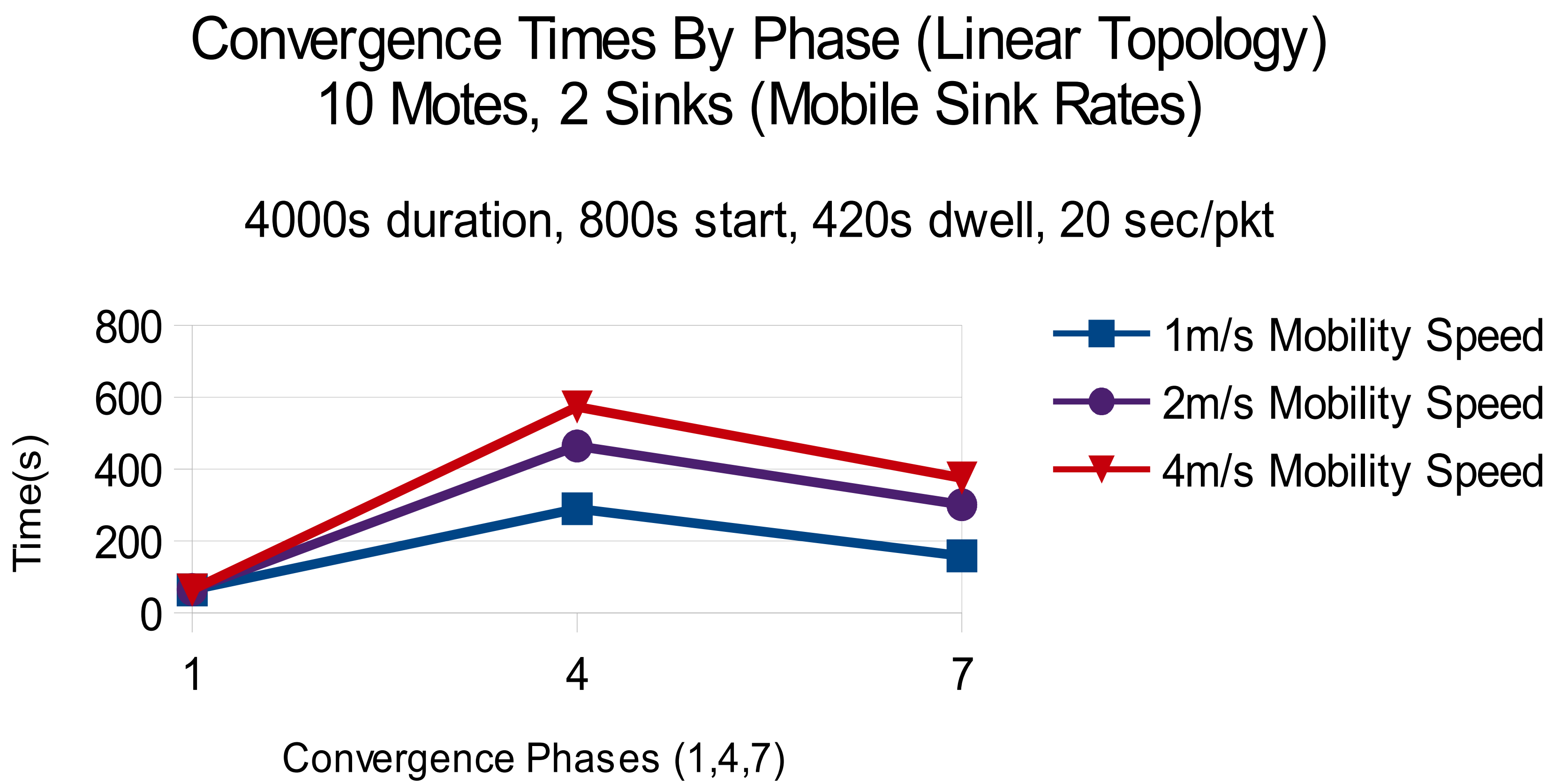
- Models applications such as intrusion detection, pipeline infrastructure maintenance or equipment heath status
- **8 Phases** of operations
- Phases 1, 4 and 7 require convergence

1. Initial time after activation for the network to form
2. Initial configuration operation
3. Movement to the dwell location
4. Time to reform at the dwell location
5. Post-convergence operation at the dwell location
6. Movement back to the initial position
7. Time to reform at the initial location
8. Post-convergence operation at the initial location.

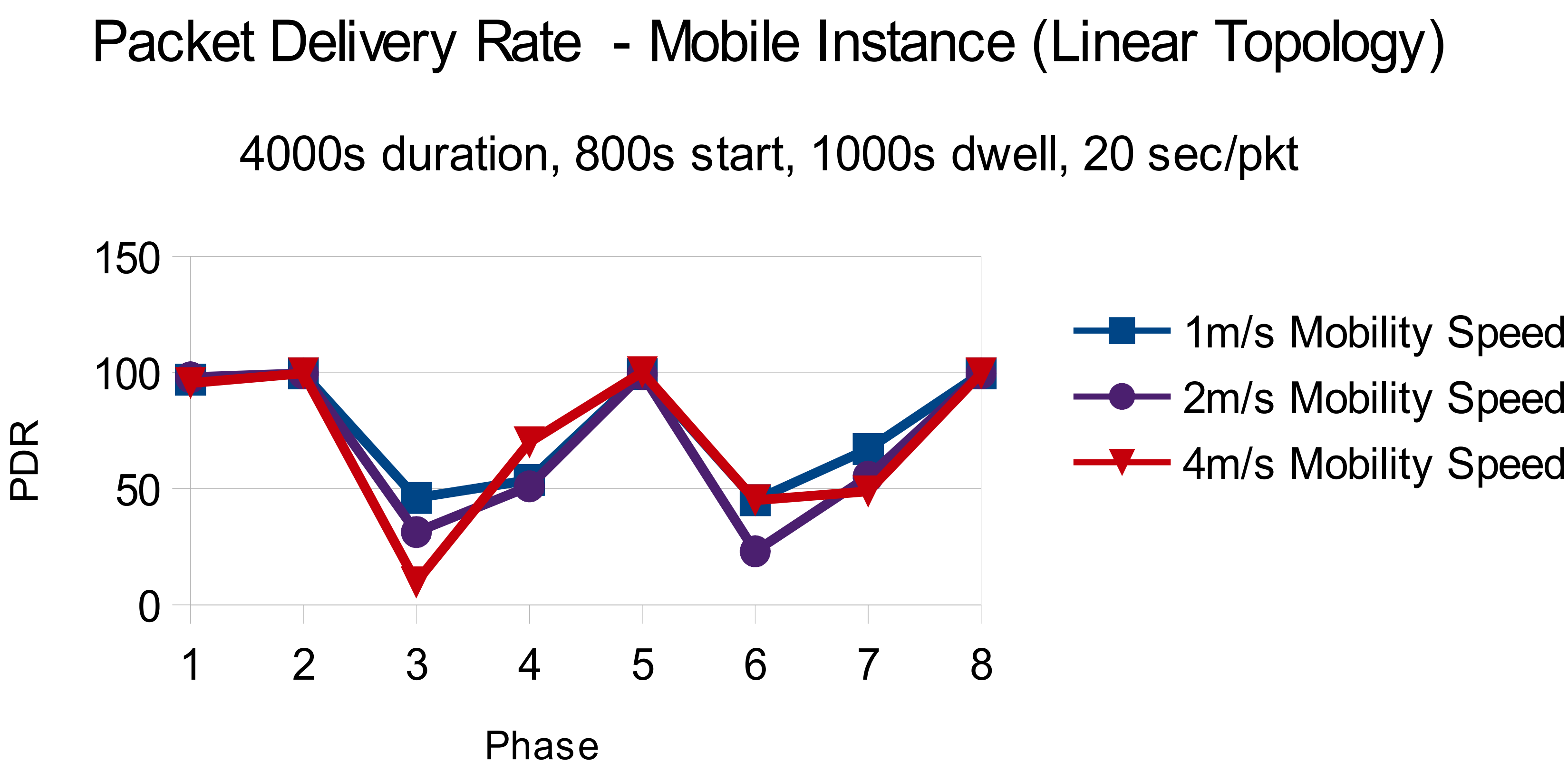
Convergence Times: Implementation vs. Simulation



Convergence Times by Mobility Speed



PDR by Mobility



Conclusions and Observations

- Able to successfully build a thin software layer for swarm-LLN communication without breaking standards
- Connectivity demonstrated
- Time to converge remains a challenge: partitions lasted up to ten minutes
- However, in all cases, network reconverged
- **Issues**
    - Network reformation must be faster and more robust to better support large numbers of robots
    - The network is a tree rather than a graph, so there is high load on the sink. How can we reduce this while staying within the memory/computational constraints of the motes?

Supporting Mobile Swarm Robotics in Low Power and Lossy Sensor Networks

Kevin Andrea kandrea@gmu.edu
Robert Simon simon@cs.gmu.edu
Sean Luke sean@cs.gmu.edu

Department of Computer Science
George Mason University

Washington, DC