Writing Programs that Work

Nupur Kothari
University of Southern California

Roy Shea
University of California, Los Angeles
Introduction

• Sensor network deployments are getting more advanced and complicated
  – Images, video feeds from cameras in sensor networks
• Writing programs for these deployments also getting more complicated
• Need for new technologies that simplify programming for distributed sensing
Studying lizards at the James Reserve
Deployment at James Reserve
• Resource efficiency
  – Energy efficient communication
  – Memory efficiency

• Reliability
  – Data should be consistently handled
  – Coordination among nodes needed

• Failure recovery
  – Detect and deal with unexpected failures
The Consequence of Failure

Most programmer errors ultimately result in node failures or unexpected application behavior. Data may be lost, and the application may crash.
How to avoid these errors

Current Technique

• Perform pre-deployment testing (trial and error) and fix the errors caught.
• Onus is on the programmer to catch all possible errors.

Our Proposals

• Use static program analysis techniques to generate warnings for code that may cause errors during execution.
• Develop new languages to avoid some of the programmer-generated errors.
Lighthouse: Static Analysis

Sensor Network Program → Lighthouse → Display Potential Program Errors to Developer
Lighthouse: Static Analysis

Sensor Network Program

Lighthouse

Display Potential Program Errors to Developer
Lighthouse: Static Analysis

Sensor Network Program → Lighthouse → Display Potential Program Errors to Developer
• Analysed program is not executed
• Analysis is independent of execution path
  – Not biased by developer expectations
  – Find all possible errors
Lighthouse: Programming Challenges Addressed

• Resource efficiency
  – Energy efficient communication
  – Memory efficiency

• Reliability
  – Data should be consistently handled
  – Coordination among nodes needed

• Failure recovery
  – Detect and deal with unexpected failures
Conventional Programming Approach

Conventional sensor net programming

Node-level program written in nesC

Compiled to executable
Pleiades: A new way to write programs

Programming using Pleiades

Central program that specifies application behavior

Node-level program written in nesC

Compiled to executable

Lets the programmer focus on application semantics
• Resource efficiency
  – Energy efficient communication
  – Memory efficiency

• Reliability
  – Data should be consistently handled
  – Coordination among nodes needed

• Failure recovery
  – Detect and deal with unexpected failures
Conclusions

• Programming sensor networks is inherently hard
• Employing techniques developed in the Programming Languages/Compilers community to sensor network programs can significantly reduce the burden of the programmer
• Lighthouse:
  https://projects.nesl.ucla.edu/public/lighthouse/
• Pleiades:
  http://kairos.usc.edu
Thank You!