An Agent Programming Model for Distributed Image Tracking Sensor Networks

Leo Szumel

11/16/2006

At the 13th Annual Signal and Imaging Sciences Workshop

Outline

- Introduction
- Motivating example
- Background
  - My work on programming and communication in WSN's
  - DISCERN overview
- Current work and outlook
- Conclusion
- Discussion

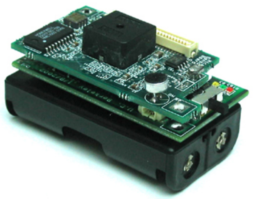
Introduction

What do I mean by wireless sensor network?

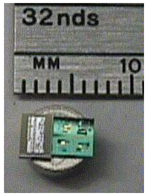
What make them interesting?

- Novelty
- Low cost
- Scalability
- System robustness

6 m



6 cm



6 mm

What makes them hard?

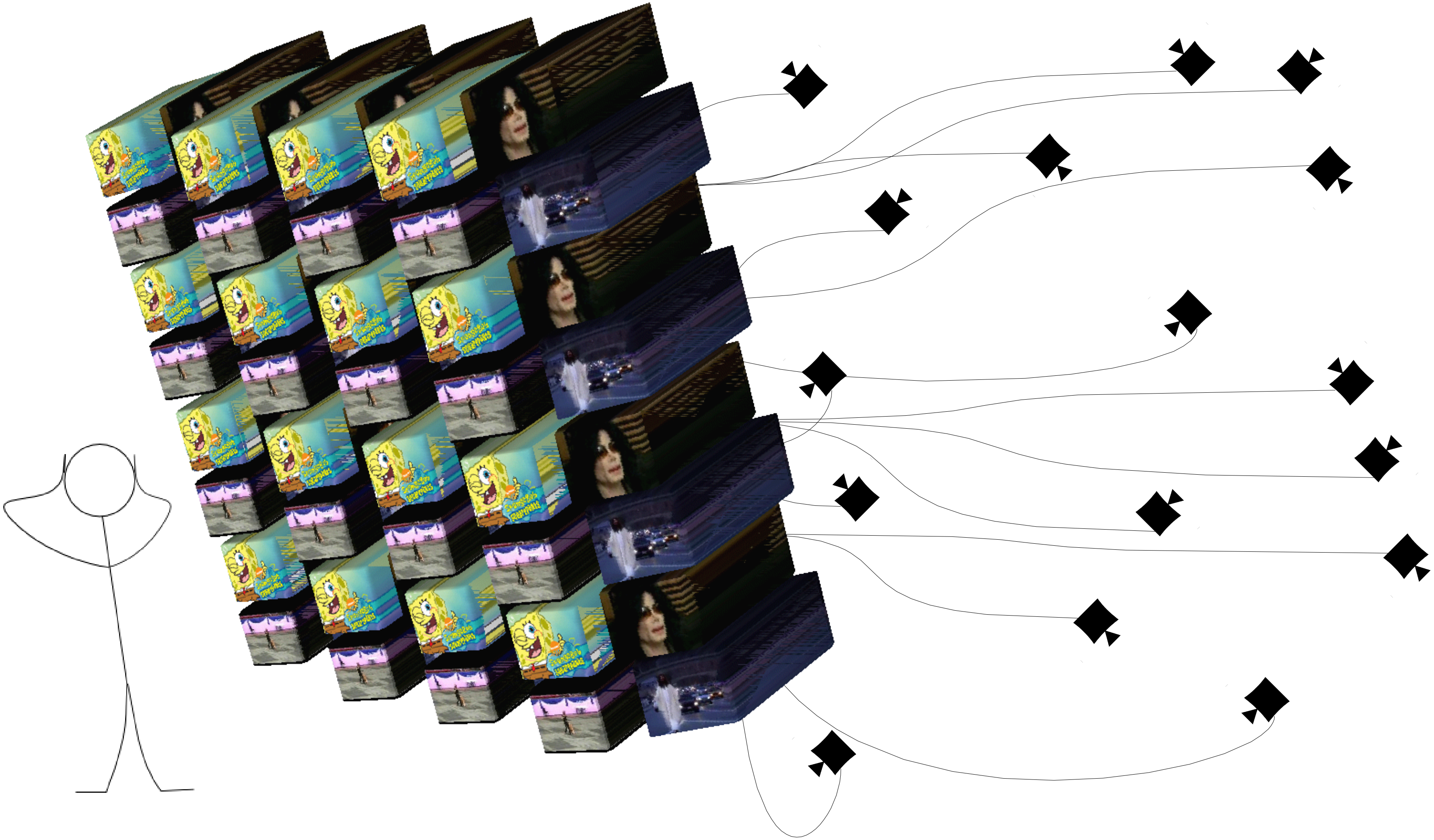
- Unreliable components, Distributed Programming, Resource Constraints

Motivation: Image Gathering

Security

Env. monitoring

Motivation: Non-Scalable!

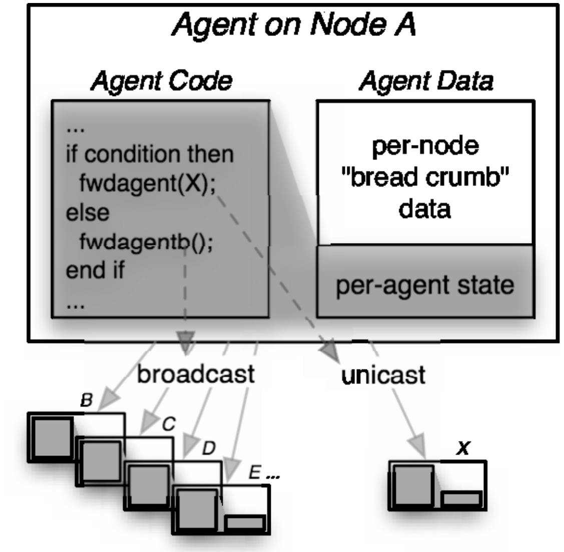


Previous Work

Mobile Agent Framework

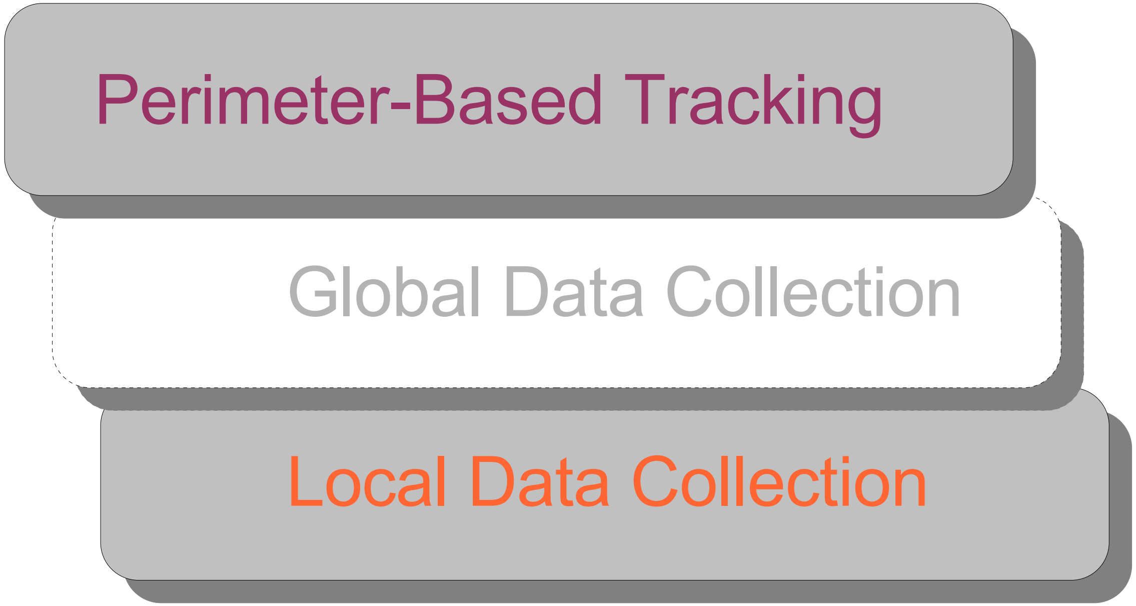
Framework for (re)programming networks of nodes that are

- arbitrarily large
- resource-constrained
    - energy, memory, cpu, bandwidth
- unreliable
- potentially heterogeneous
- potentially mobile



```
...
if condition then
  fwdagent(X);
else
  fwdagentb();
end if
...
```

Test Applications



Perimeter-Based Tracking

Global Data Collection

Local Data Collection

Virtual Pheromone Model

Pheromone source: creates pheromones

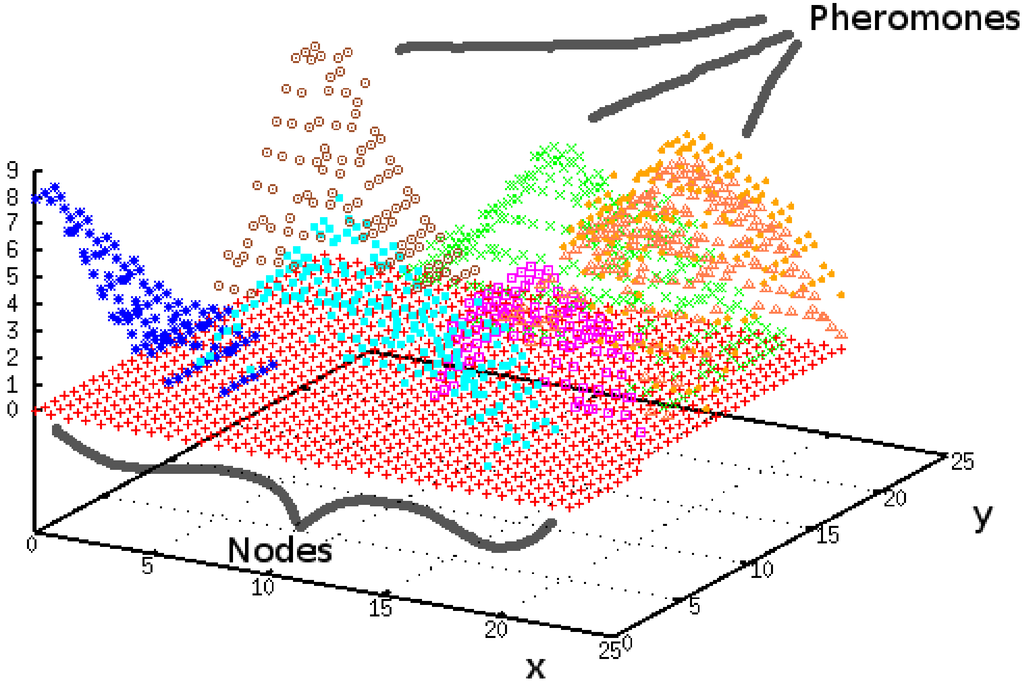
Neighbor propagation

Transfer function specifies values

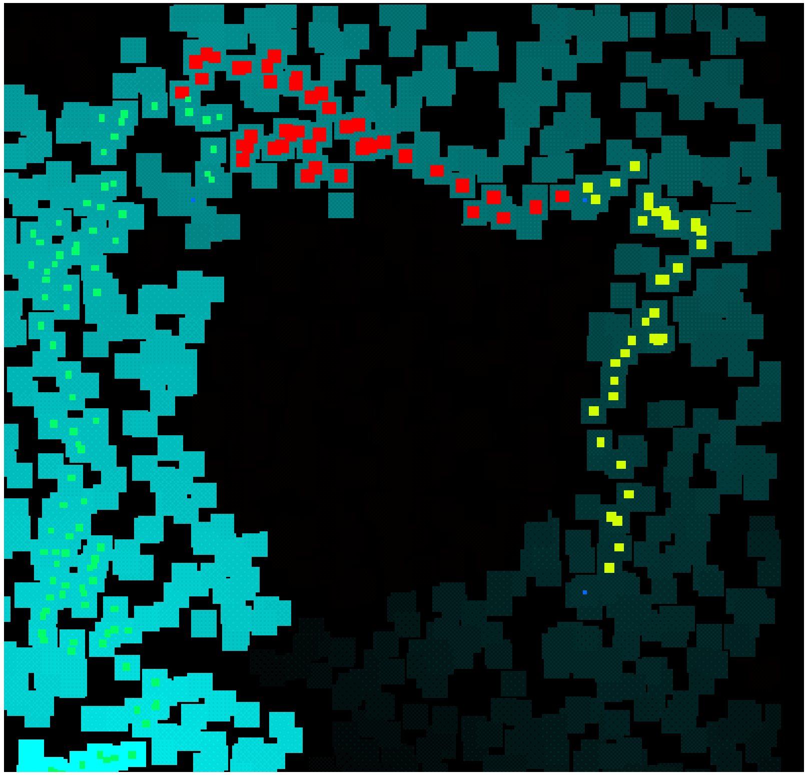
Decays in space

Can decay in time

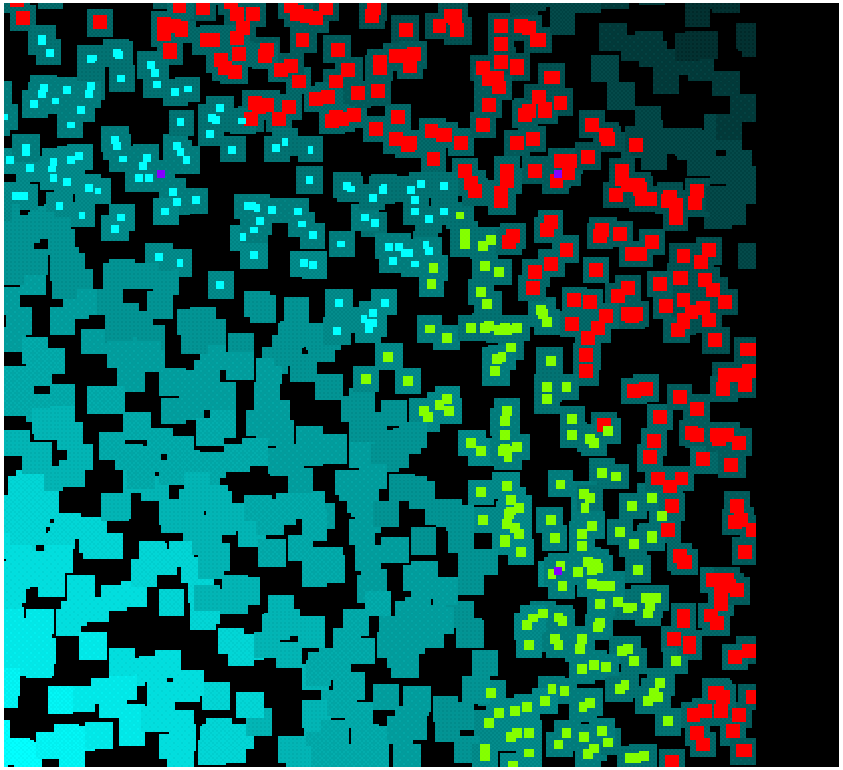
Additive



Pheromone Examples



Gradient Routing



Spatial Organization and Clustering

DISCERN

DISCERN

Assemblage of three platforms

Stargate PXA255

- ~200 mA

Cyclops ATMega128L

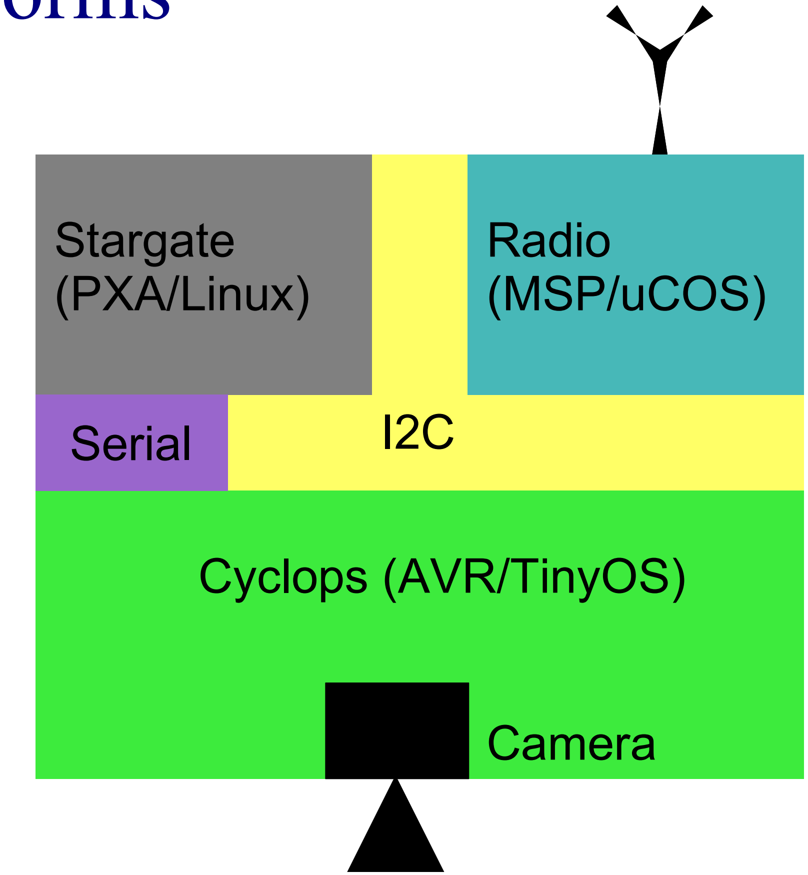
- ~20 mA

Radio CC2420

- ~20 mA

Camera ADCM-1700

- ~5 mA*



DISCERN “Link Neighbors”

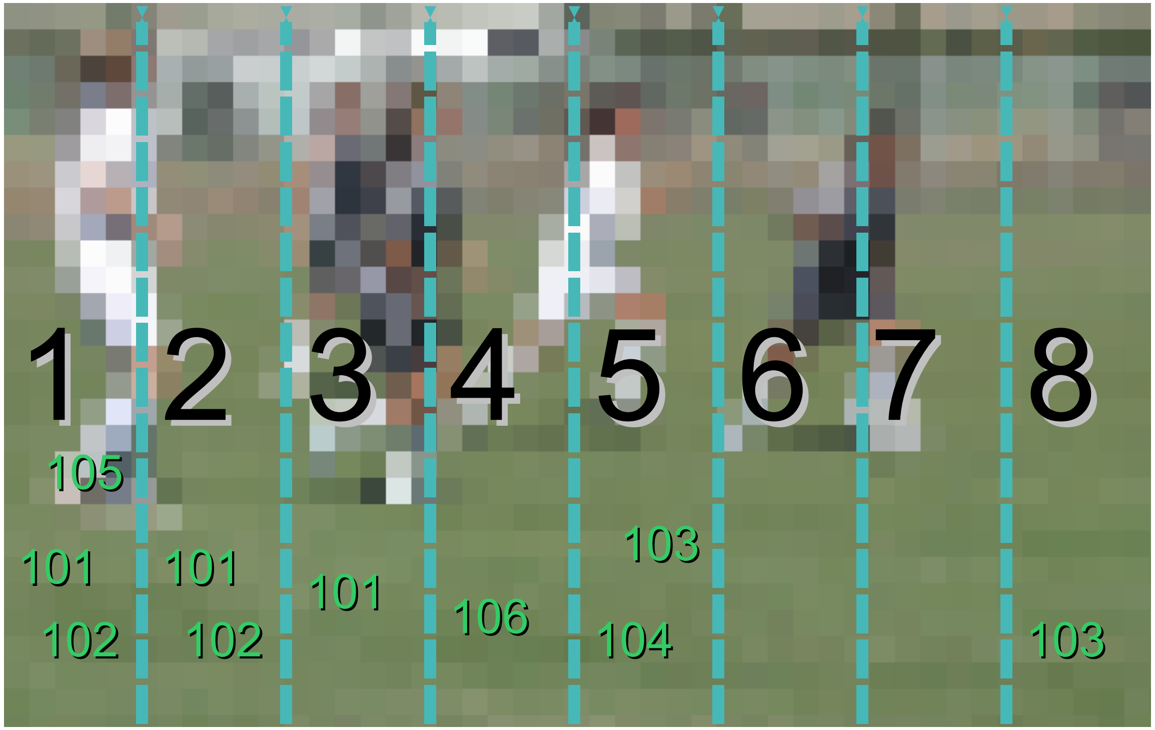
Visual field divided into vertical bins

Neighbor list maintained for each bin

- 1:{105,101,102}
- 2:{101,102}
- ...

Dynamic update

Single-hop



Work in Progress

Porting to Cyclops

Experiment 1: Notification agent

- Track object until statistical threshold satisfied (MA)
  - Send notification and image to BS
- Advantage: automatic alerting with reduced false positives

Experiment 2: Coverage agent

- Feature clustering, leader election (VP)
- Advantage: non-participating nodes can idle

Preliminary Results

Experiment 1: Tracking agent

- ~100 bytes for most basic tracking
  - expect 40% just for branching logic

Experiment 2: Coverage agent

- ~30 bytes for negotiation
  - special opcodes: pheromone search, feature bin

Expected Results (Pending)

Experiment 1:

- Enables decreased notification latency unachievable without distributed processing
- Analysis of tradeoff between inter-agent communication (bandwidth, energy) and performance (latency)

Experiment 2:

- Energy savings by allowing redundant nodes to sleep and "ignore" an event
- Analysis of tradeoff between robustness and energy savings

Conclusions

We can effectively add a *scripting layer* on top of an optimized application (here, DISCERN). This technique allows for:

- Post-deployment tweaks and optimizations
- Multiple users/apps of the same network
- More flexible designs
- Fault tolerant algorithms

Discussion Points

Where should the boundary be?

- Lower static layer is optimized and efficient, but inflexible (very expensive to change)
- High dynamic layer is cheap to change and experiment with

How should the scripting be exposed to the programmer?

Can anyone be expected to program using the agent methodology?

Backup Slides

AHLPS

Agent High-Level Pythonic Simulator

Python + SimPy + SciPy + ...

TOSSIM radio link model

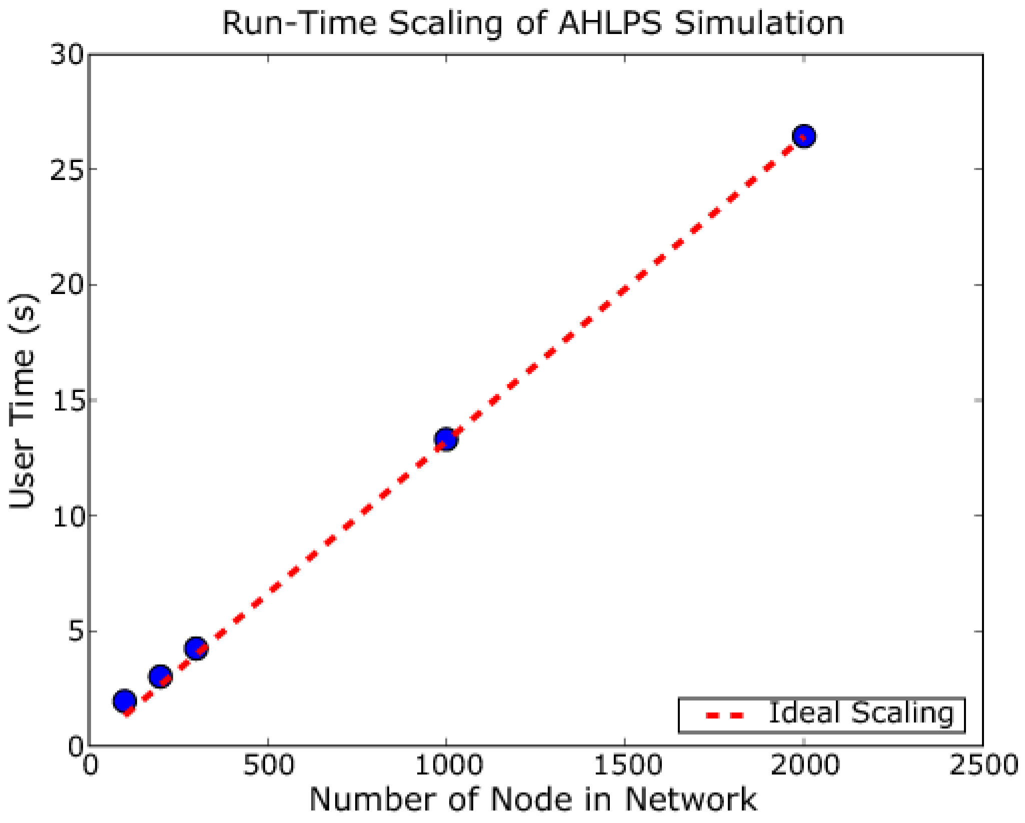
Simulate thousands of nodes

Easily extensible (OO)

Command-line and GUI modes

AHLPS Performance

Scaling performance for 10 s simulation


Run-Time Scaling of AHLPS Simulation
User Time (s)
Number of Node in Network
Ideal Scaling