Efficient Clustering of Sensor Networks

Jason Hemann
Et al.

Outline

1. Why do we cluster?
2. Clustering thus far
3. Measurement & Success
4. YACA - Yet Another Clustering Algorithm
5. “How I spent my summer vacation”
6. Prospects and Future Work

Why do we cluster?

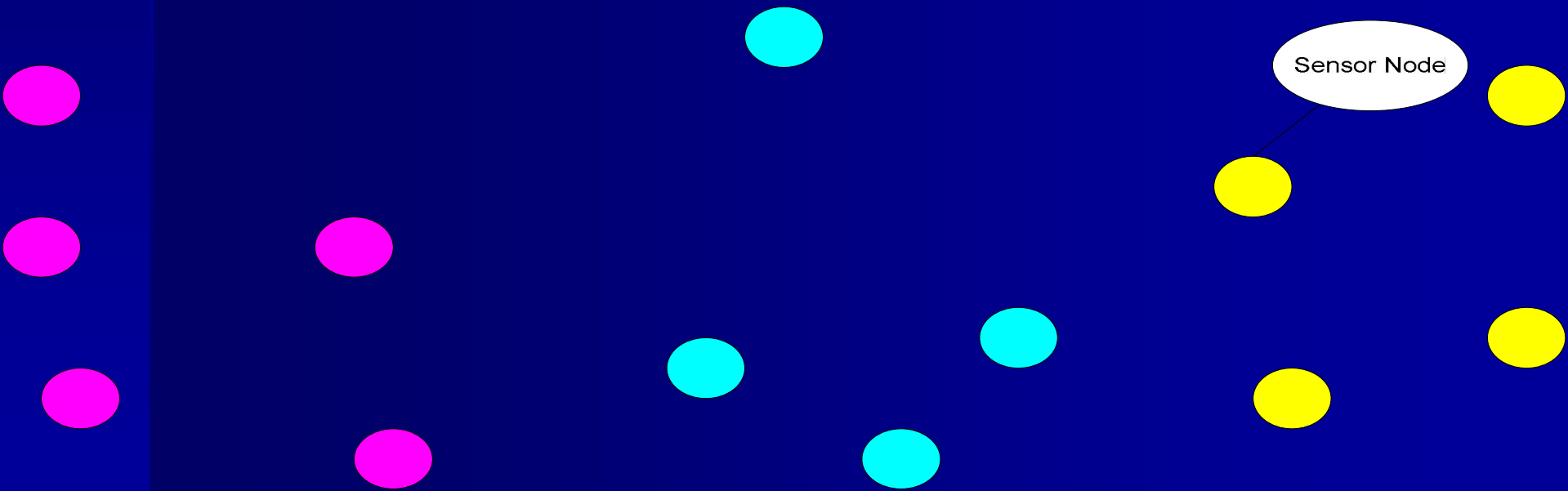
- Consist of limited ability nodes
- Networks used for a variety of purposes
- Power is non-renewable and irreplaceable
- Our goal is to stretch the life of the network while collecting accurate data.
- Clustering is necessary to achieve this end

1-all communication

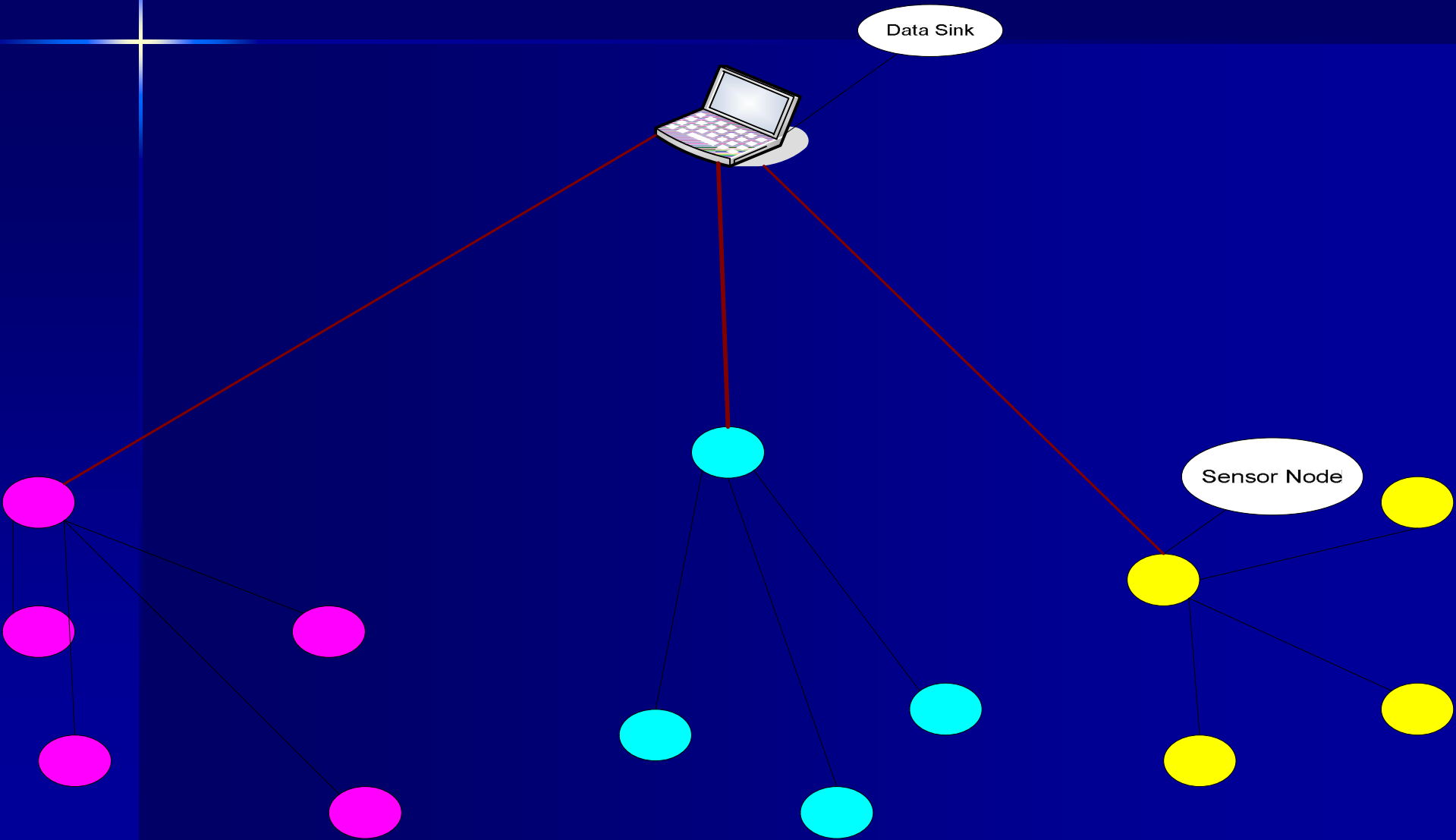
Problems of 1-all communication

- Transmission cost proportional to distance.
- Nodes will have their power depleted and will die quickly.
- Nodes need to *collaborate* to save power and lengthen life.

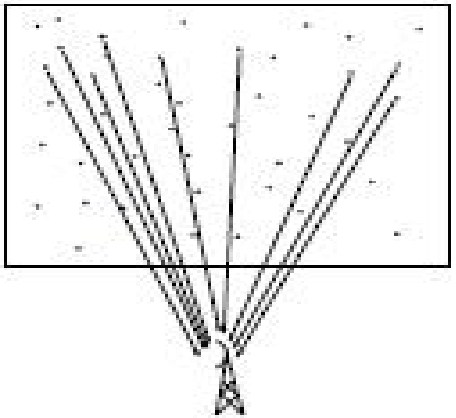
Collaboration through clustering



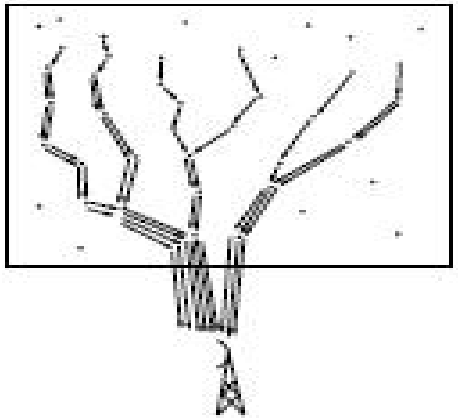
2-layer structure



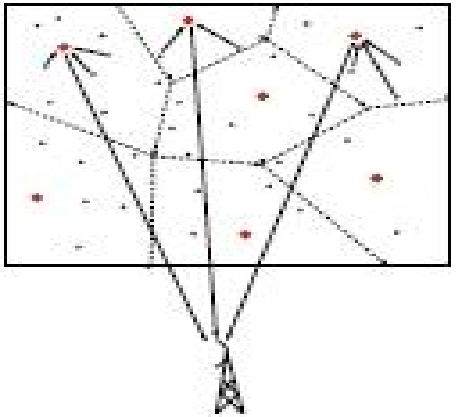
Clustering and Routing



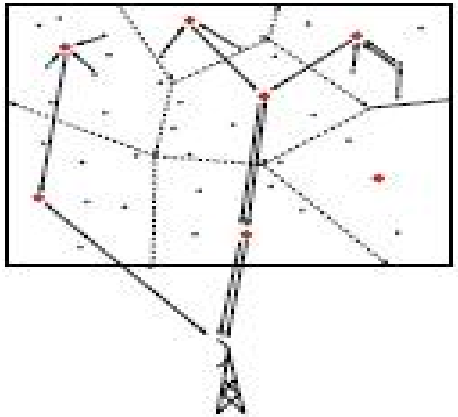
(a) Single hop without clustering



(b) Multi-hop without clustering



(c) Single hop with clustering



(d) Multi-hop with clustering

2. Clustering thus far

The naïve approach:

- Get full information about the network, and devise the *optimal* organization.
- Uses up too much of the resource it is meant to save.

A better way: Each node makes a decision based on local information (distributed).

Clustering Algorithm Template

Each Node:

Exploratory Step	Should I self nominate?
Primaries	Announcements. Compete with other cluster head candidates.
Voting	Non cluster heads select a cluster head and join.

Exploratory Step

LEACH	Given input parameters l (network density), and cluster number N, compute probability to self-nominate. **No communication with network is needed to make the decision.**
HEED	Compute power required to communicate with each neighbor. Total the power, broadcast it to all neighbors. Check residual power, compute probability to self nominate.

Announcement Step

LEACH	Broadcast a message declaring candidacy.
HEED	Broadcast possible candidacy and total cost. Listen for other candidates. Desist or increase probability to remain candidate depending on what is heard from other candidates.

Election Step

LEACH	If node is cluster head, join own cluster. If node is not cluster head, select the "closest" (based on signal strength) cluster head.
HEED	If node is cluster head, join own cluster. If node is not cluster head, select a cluster head based on criteria (e.g. smallest for load balancing)

3. Measurement & Success

- Cost of clustering
  - Measured with in units of power.
  - The most power hungry operation is communication.
  - Total cost is roughly proportional to the number of messages exchanged.
- Quality of clustering
  - Heuristic approximations:
    - Number of clusters
    - Balance between clusters
    - Cluster heads are nodes with highest power
  - Ultimately, better clusters are those which result in the longer lived network.

Performance, Assumptions

- Most algorithms have similar clustering costs:
  - total number of messages on O(N) (a few messages per node).
- Most algorithms make similar assumptions
  - Nodes are location un-aware
  - Nodes are not mobile
  - Nodes have similar capabilities
  - Nodes are distributed in some specified way
  - Nodes are synchronised

Assumption: Synchronisation

For clustering algorithms to work properly, we need

- a common time frame,
- shared schedule
- Shared deadline

This assumption is overall reasonable but hard to accomplish.

- Even when nodes are identical and start with their clocks synchronised "clock drift" occurs.

4. YACA

- Distributed decision.
- Minimizes amount of communication between nodes, while
- Providing a good distribution of clusters.
- Cluster heads not crowded.
- Makes no assumptions about node distribution.
- Synchronization not required.

The idea

1. All nodes introduce themselves.
2. As they listen, each node counts its neighbors.
3. Once introductions over, node waits a time inversely proportional to its power and number of neighbors heard.
4. Then –if not too late- announces candidacy.

How do we address Synchronization?

- Potential problem: How do we know when introductions are over?
- Solution: Once a node hears silence for a time t, it starts the countdown to announce its candidacy.
- Problem: Nodes with fewer neighbors will start countdown earlier.
- Solution: First node to start countdown announces it so that its neighbors do the same.

Benefits

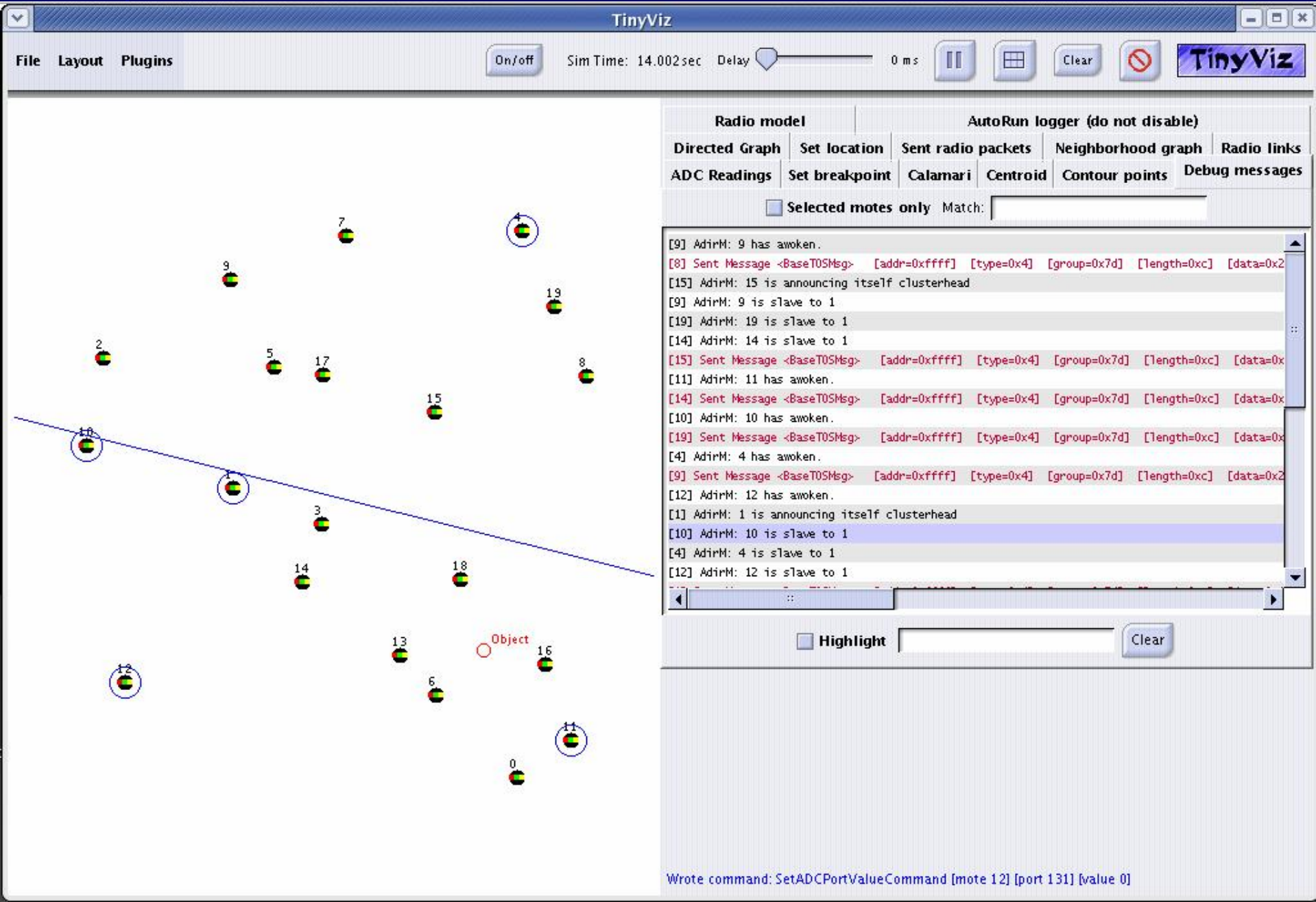
- No prior information about distribution required.
- Cluster heads decided based on hard data rather than probabilities.
- No two cluster heads are within transmission range of each other.
- Number of communications required per node <3 (1 for introduction, 1 to announce countdown and 1 to announce being a cluster head).

Performance

- Some of its properties can be proven analytically.
- Others are best established through simulation.
- Comparison with other algorithms is also best done through simulation.
- How well does it **REALLY** fare?

5. "What I did on my Summer Vacation"

- Read Background Material
- Learned the nesC Language
- Installed/Configured TinyOS & TOSSIM
  - Software Incredibly difficult to use
  - Most time was spent installing calibrating
  - Multiple platforms each with their own quirks
  - Multiple Ph. Ds and entire IT staff
  - Documenation limited to nonexistent.
  - Software Destroyed my computer
- Developed Basic Simulation
- Contended with Assault, Grand Theft, and a Mutiny


TinyViz
File
Layout
Plugins
On/off
Sim Time: 14.002 sec
Delay
0 ms
Clear
TinyViz
Radio model
AutoRun logger (do not disable)
Directed Graph
Set location
Sent radio packets
Neighborhood graph
Radio links
ADC Readings
Set breakpoint
Calamari
Centroid
Contour points
Debug messages
Selected motes only
Match:
[9] AdirM: 9 has awoken.
[8] Sent Message <BaseTOSMsg> [addr=0xffff] [type=0x4] [group=0x7d] [length=0xc] [data=0x2
[15] AdirM: 15 is announcing itself clusterhead
[9] AdirM: 9 is slave to 1
[19] AdirM: 19 is slave to 1
[14] AdirM: 14 is slave to 1
[15] Sent Message <BaseTOSMsg> [addr=0xffff] [type=0x4] [group=0x7d] [length=0xc] [data=0x
[11] AdirM: 11 has awoken.
[14] Sent Message <BaseTOSMsg> [addr=0xffff] [type=0x4] [group=0x7d] [length=0xc] [data=0x
[10] AdirM: 10 has awoken.
[19] Sent Message <BaseTOSMsg> [addr=0xffff] [type=0x4] [group=0x7d] [length=0xc] [data=0x
[4] AdirM: 4 has awoken.
[9] Sent Message <BaseTOSMsg> [addr=0xffff] [type=0x4] [group=0x7d] [length=0xc] [data=0x2
[12] AdirM: 12 has awoken.
[1] AdirM: 1 is announcing itself clusterhead
[10] AdirM: 10 is slave to 1
[4] AdirM: 4 is slave to 1
[12] AdirM: 12 is slave to 1
Highlight
Clear
Wrote command: SetADCPortValueCommand [mote 12] [port 131] [value 0]
Object

6. Conclusions, Plans

- Turn the basic simulation into full version of YACA, HEED and LEACH
- Compare them under different environments
- Determine whether we actually outperform other algorithms in simulation.

References

- O. Younis and S. Fahmy. HEED: A hybrid energy-efficient distributed clustering for adhoc sensor networks. *INFOCOM*, 2004.
- W. Heinzelman, et al. Energy Efficient Communication Protocol for Wireless Sensor Networks. *Proceedings of the Hawaii International Conference on System Sciences*, 2000.
- S. Bandyopadhyay and E. Coyle. An Energy Efficient Heirarchical Clustering Algorithm for Wireless Sensor Networks. *INFOCOM*, 2003