An Architecture for Sensor Networks: Directed Diffusion

Deborah Estrin

USC CS Dept and ISI

In collaboration with

Co-PIs: *Ramesh Govindan, John Heidemann*

Diffusion: *Chalermak Intanagowat, Amit Kumar*

Localized algorithms: *Jeremy Elson,Satish Kumar, Ya Xu, Jerry Zhao*

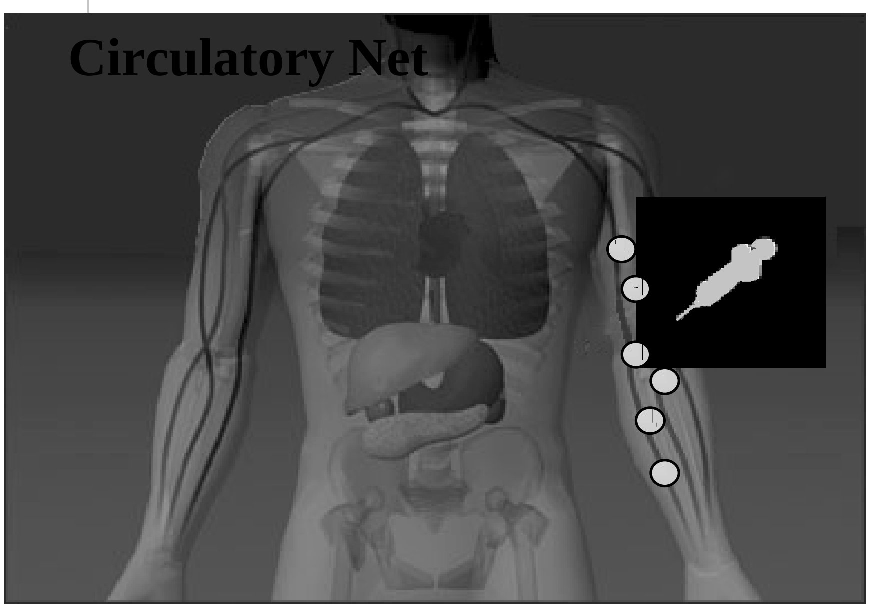
Localization: *Lew Girod, Nirupama Bulusu*

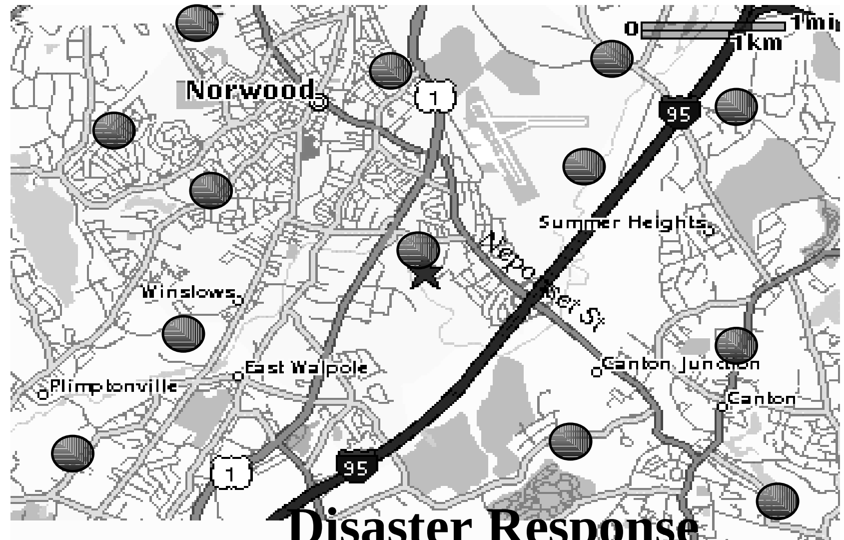
Distributed robotics: *Maja Mataric, Gaurav Sukhatme, Alberto Cerpa*

For more information: estrin@isi.edu

The long term goal

Embed numerous distributed devices to monitor and interact with physical world: in work-spaces, hospitals, homes, vehicles, and “the environment” (water, soil, air…)





Network these devices so that they can coordinate to perform higher-level tasks.

Requires robust distributed systems of tens of thousands of devices.

Overview of research

- Sensor network challenges
- One approach: Directed diffusion
  - Basic algorithm
  - Initial simulation results (Intanagowat)
- Other interesting localized algorithms in progress:
  - Aggregation (Kumar)
  - Adaptive fidelty (Xu)
  - Address free architecture, Time synch (Elson)
  - Localization (Bulusu, Girod)
  - Self-configuration using robotic nodes (Bulusu, Cerpa)
  - Instrumentation and debugging (Jerry Zhao)

The Challenge is Dynamics!

- The physical world is dynamic
  - Dynamic operating conditions
  - Dynamic availability of resources
    - … *particularly energy*!
  - Dynamic tasks
- Devices must adapt automatically to the environment
  - Too many devices for manual configuration
  - Environmental conditions are unpredictable
- Unattended and un-tethered operation is key to many applications

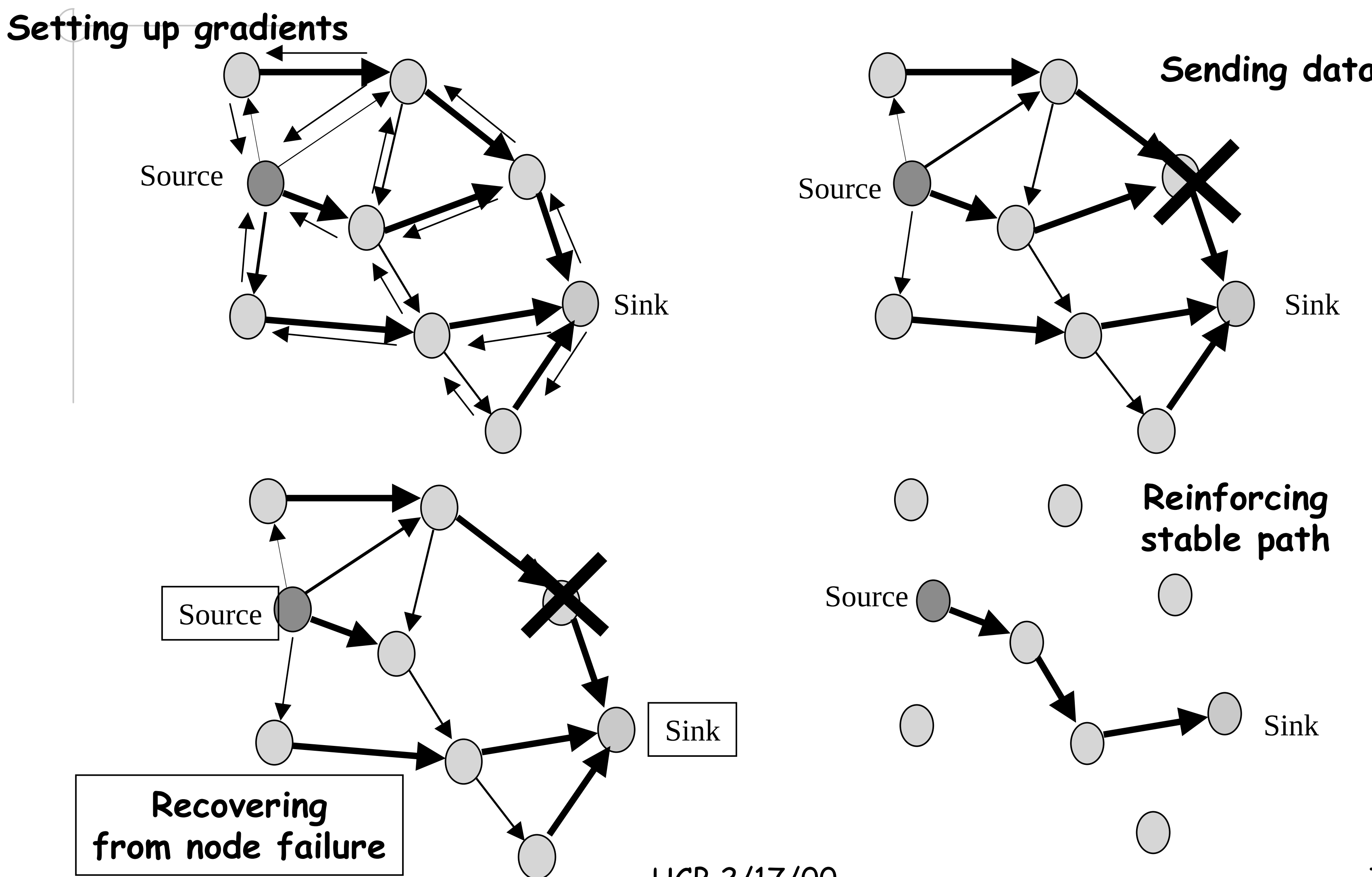
Approach

- Energy is the bottleneck resource
  - And communication is a major consumer--avoid communication over long distances
- Pre-configuration and global knowledge are not applicable
  - Achieve desired global behavior through localized interactions
  - Empirically adapt to observed environment
- Leverage points
  - Small-form-factor nodes, densely distributed to achieve Physical locality to sensed phenomena
  - Application-specific, data-centric networks
  - Data processing/aggregation inside the network

Directed Diffusion Concepts

- Application-aware communication primitives
  - expressed in terms of named data *(not in terms of the nodes generating or requesting data)*
- Consumer of data initiates interest in data with certain attributes
- Nodes diffuse the interest towards producers via a sequence of local interactions
- This process sets up gradients in the network which channel the delivery of data
- Reinforcement and negative reinforcement used to converge to efficient distribution
- Intermediate nodes opportunistically fuse interests, aggregate, correlate or cache data

Illustrating Directed Diffusion



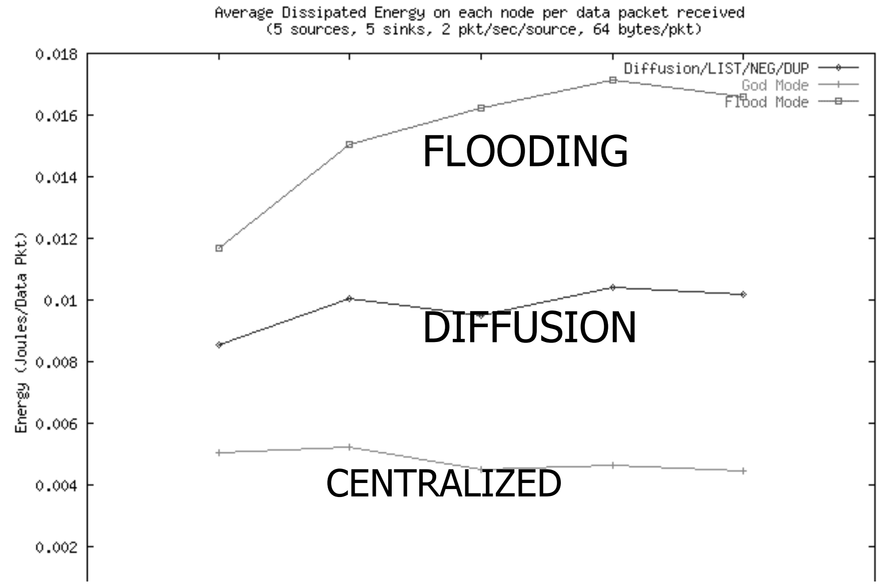
Local Behavior Choices

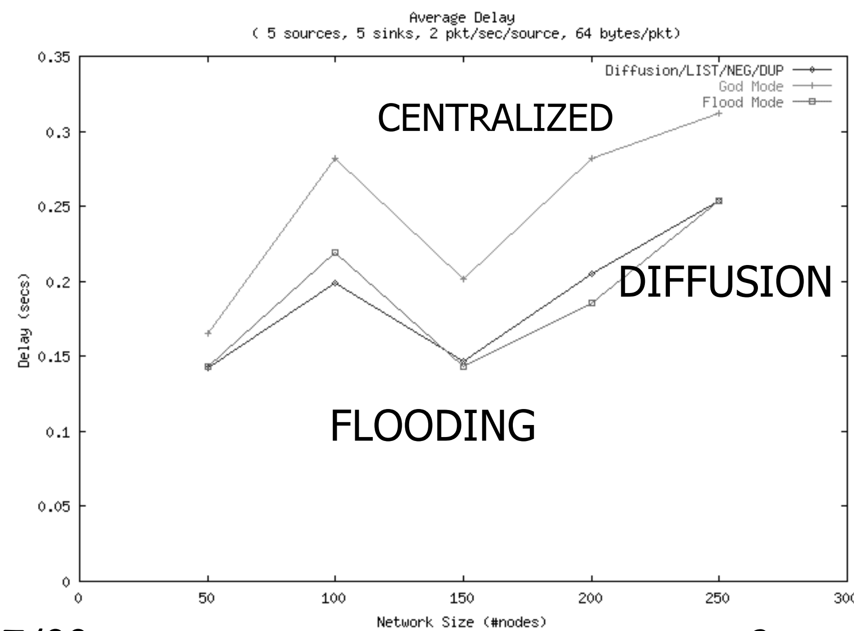
1. For propagating interests
   - In our example, flood
   - More sophisticated behaviors possible: e.g. based on cached information, GPS
2. For setting up gradients
   - Highest gradient towards neighbor from whom we first heard interest
   - Others possible: towards neighbor with highest energy
3. For data transmission
   - Different local rules can result in single path delivery, striped multi-path delivery, single source to multiple sinks and so on.
4. For reinforcement
   - reinforce one path, or part thereof, based on observed losses, delay variances etc.
   - other variants: inhibit certain paths because resource levels are low

Initial simulation studies

(Intanago, Estrin, Govindan)

- Compare diffusion to a)flooding, and b)centrally computed tree ("ideal")
- Key metrics:
  - total energy consumed per packet delivered (indication of network life time)
  - average pkt delay





What we really learnt
(things we don't usually show…because in retrospect they seem so obvious)

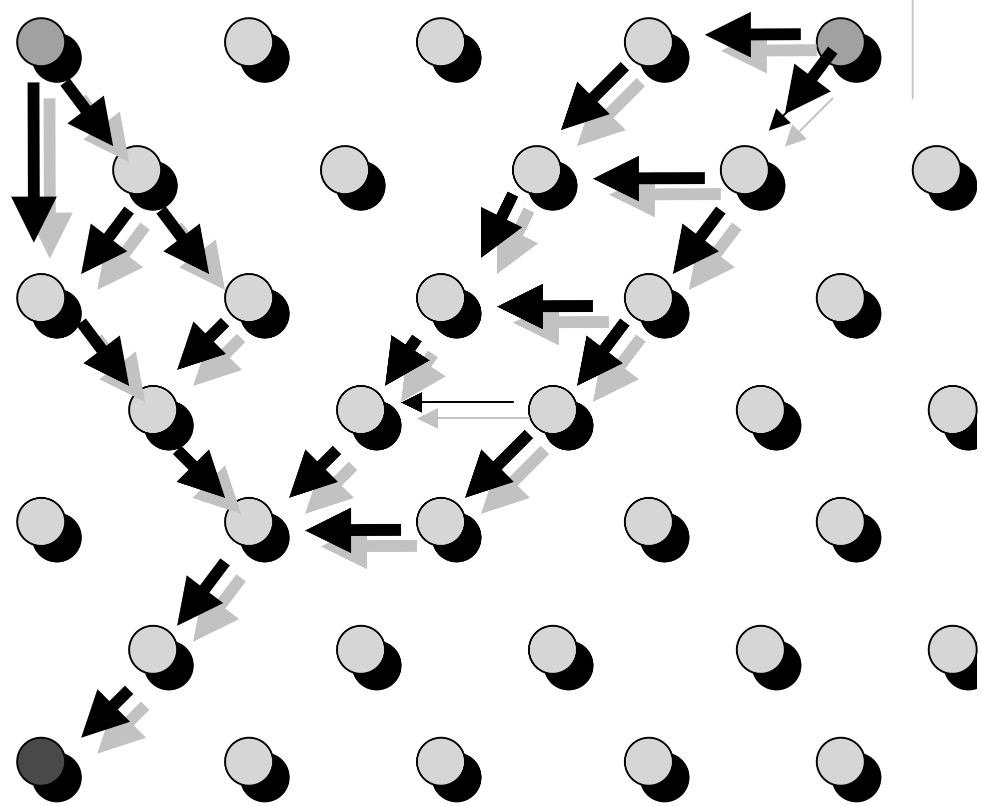
- IDLE time dominates energy consumption…need low duty cycle MAC, driven by application.
  - With 802.11ish contention protocols you might as well just FLOOD
- Easy to get lost in detailed simulations but in the wrong region of operation …
  - Node density, traffic load, stream length, source and sink placement, mobility, etc.

Exploring Diffusion

- Aggregation
- Adaptive Fidelity
- Implications
  - “address free” architecture
  - Need for localization
- Using diffusion
  - System health measurements
  - Robotic nodes

Diffusion based Aggregation
(Kumar, Kumar, Estrin, Heidemann)

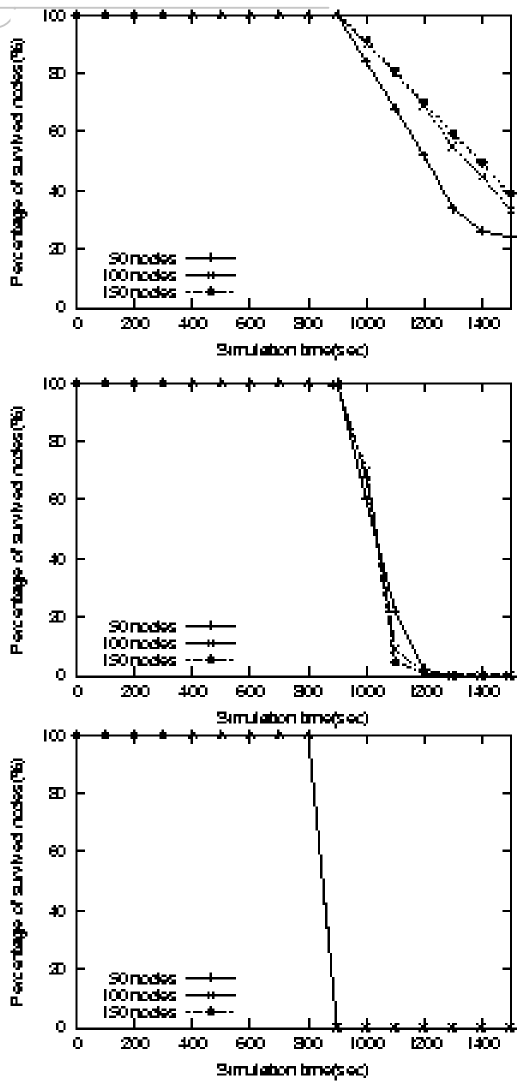
- Scaling requires processing of data INSIDE the net
- Clustering approach:
  - Elect cluster head (various promotion criteria)
  - Aggregation or Hashing (indirection) to map from query to cluster head
- Opportunistic aggregation:
  - Reinforce (request gradient) proportional to aggregatability of incoming data (Amit Kumar)



Adaptive Fidelity
(Xu, Estrin, Heidemann)

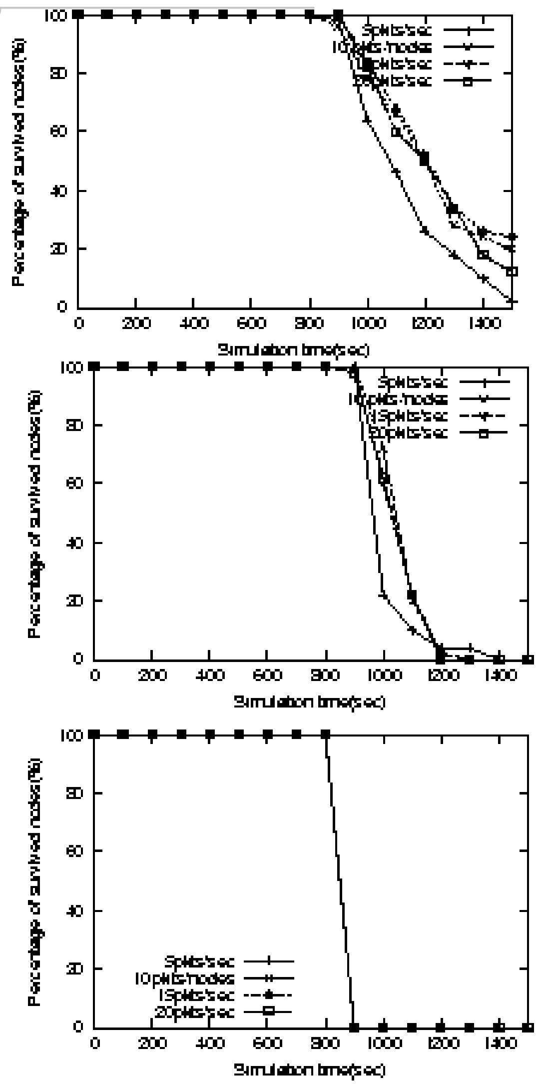
- In densely deployed sensor nets, reduce duty cycle: engage more nodes when there is activity of interest to get higher fidelity
- Adjust node's sleeping time according to the number of its neighbors.
- Initial simulations applied to ad hoc routing
- Performance Metric: Percentage of survived nodes over time.
  - The more nodes survive, the longer network lifetime

Comparison: Density factor



- At the left, from top to the bottom: Adaptive Fidelity, Basic algorithm, regular AODV
- Simulation under 50 nodes, 100 nodes, 150 nodes
- Network lifetime is extended by deploying more nodes only with adaptive fidelity algorithm
- Simulations available (ns-2 based)

Comparison: Traffic Factor



- At the left, from top to the bottom: Adaptive Fidelity, Basic algorithm, regular AODV
- Simulation under different traffic load: 5pkt/s, 10pkt/s, 15pkt/s, 20pkt/s
- Longer network lifetime in adaptive
- The more traffic load, the greater the advantage in terms of network lifetime

Adaptive Fidelity conclusions

- Must be applied at application level (because just listening/having radio on dominates energy dissipation)
  - Unfortunate side effect of resource constraints is the need to give up (some) layering
- Many open questions as to density thresholds and how to design algorithms to exploit it.

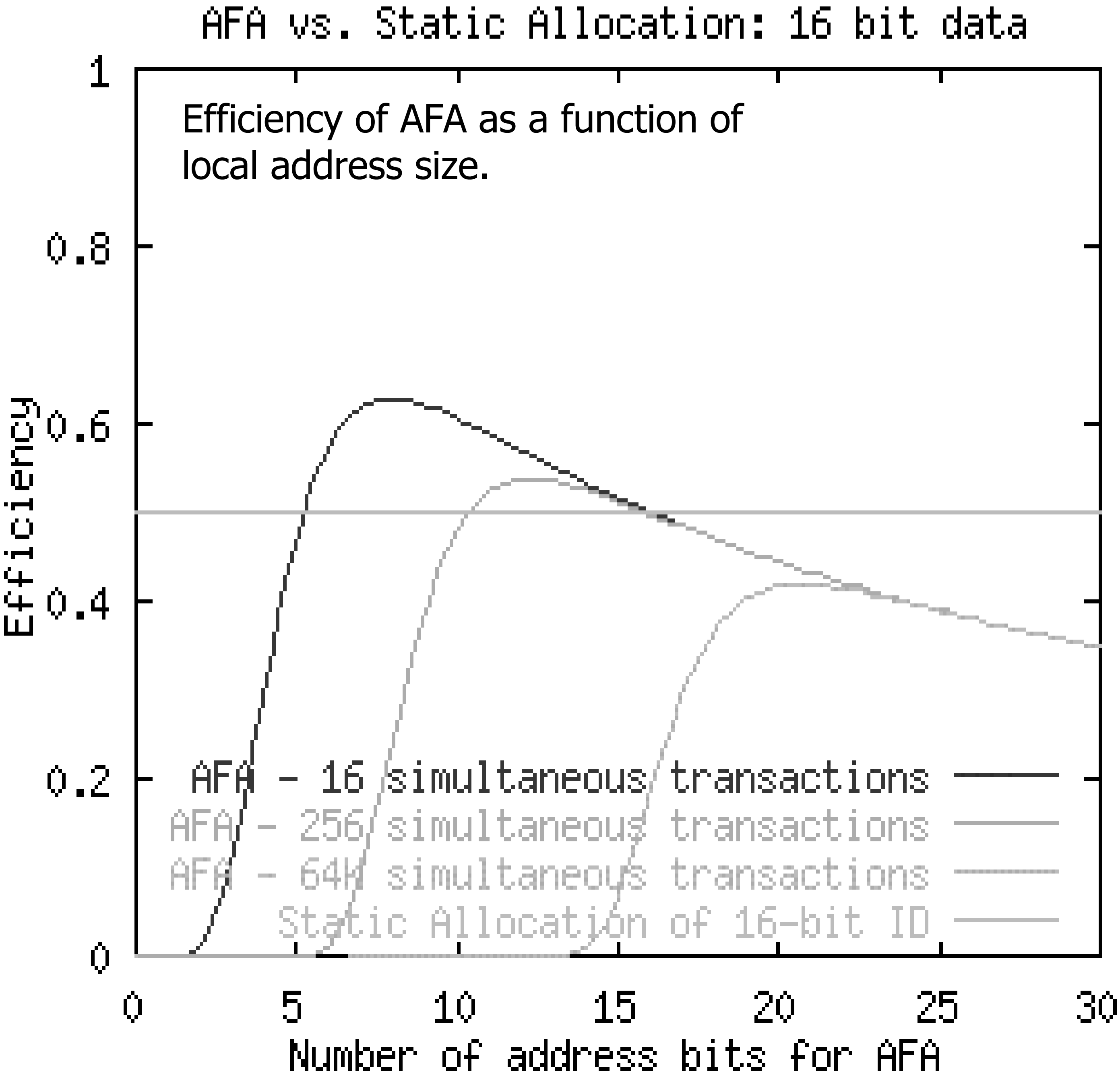
Implications: local addresses?

(Elson, Estrin)

- **Sensor nets**: maximize usefulness of every bit
  - each bit transmitted reduces net lifetime
  - can't amortize large headers for low data rates
  - underutilized address space is bad
- Still need to identify transmitter
  - Reinforcements, Fragmentation
- Use small, random transaction identifiers (locally selected…like multicast addresses)
  - Treat identifier collisions as any other loss
- Address-free method can win in networks with *locality*
  - simultaneous transactions at any one point is much less than in network as a whole

No need for global address assignment…but how inefficient is it?

- AFA optimizes number of bits used per packet
  - o Fewer bits: less overhead per data bit
  - o More bits: less contention loss



Implcations: Need Localization

(Bulusu, Girod)

- Many contexts you cant have GPS on every node
  - form factor
  - energy
  - obstructions
- Beacon architecture
- Signal strength alone problematic/hopeless
- Federated coordinate systems
- Acoustic ranging (client node asks beacons to send chirp and monitors time of flight)
- Self-configuring beacon placement using robotic nodes

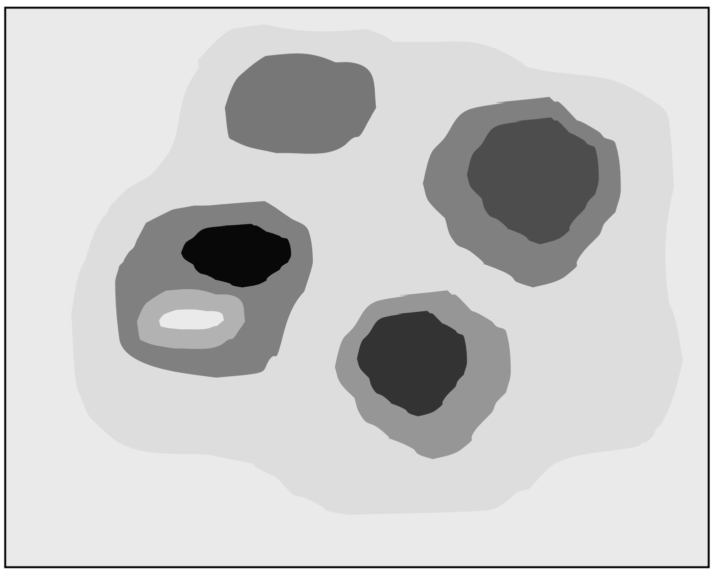
Localization is a critical service (Girod)

- Devices take up physical space
  - Sufficiently fine-grained spatial coordinates provide implicit routing information (e.g. directing interests)
- Location is relevant to many applications
  - Devices are doing things in the world; users need to find them; inputs and outputs to tasks often reference locations
- How can we achieve fine-grained localization?
  - Need sensors to measure distance (ranging)
    - Time arrivals of 3 requested acoustic signals; not signal strength
  - Relative or Global?
    - Relative spatial measurements more accurate because observed phenomena are local, shorter ranges, etc.
    - Global measurements (e.g. GPS) coarser (40m) but provide single coordinate system that can be exported unambiguously
- Combine global scope of GPS with precision of relative sensors: fuse local & global coordinate frames

Localization relies on beacons
(Bulusu, Heidemann, Estrin)

- Precision of localization depends on beacon density/placement
  - Uniform placement not good solution in real environments
  - Obstacles, walls, etc prevent inference based on signal strength/proximity detection
- Self-configuring beacon placement is interesting application for robotic nodes
  - Given obstacles, unpredictable propagation effects, need empirical placement

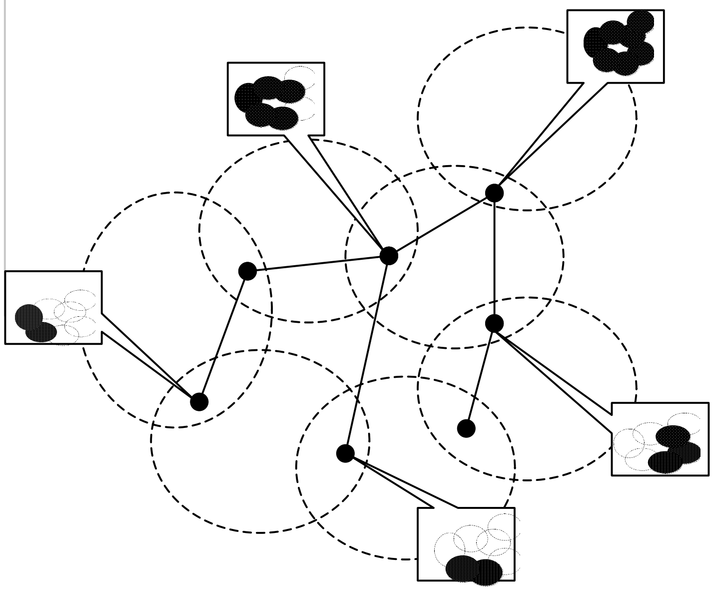
Sensor Network Tomography
(Zhao, Govindan, Estrin)



Tomogram indicating connection quality

- Continuously updated indication of sensor network health
- Useful for
  - performance tuning
    - adjusting sensing thresholds
  - incremental deployment
    - refurbishing sections of sensor field with additional resources
  - self testing
    - validating sensor field response to known input

Sensor Network Tomography: Key Ideas and Challenges



- Kinds of tomograms
  - network health
    - resource-level indicators
  - responses to external stimuli
- Can exchange resource health
  - during low-level housekeeping functions
  - … such as radio synchronization
- Key challenge: energy-efficiency
  - need to aggregate local representations
  - algorithms must **auto-scale**
  - outlier indicators are different

Self configuring networks using and supporting robotic nodes
(Bulusu, Cerpa, Estrin, Heidemann, Mataric, Sukhatme)

- Robotics introduces self-mobile nodes and adaptively placed nodes
- Self configuring ad hoc networks in the context of unpredictable RF environment
- Place nodes for network augmentation or formation
- Place beacons for localization granularity

CONCLUSIONS

- Have just scratched the surface
- We need to put more experimental systems in place and start living in instrumented environments or we risk too many rat-holes and pipe-dreams…
- Long-term and High-impact opportunities:
  - Biological monitoring
  - Environmental sensing
  - Medical applications based on micro and nano scale devices
  - In-situ networks for remote exploration