The Firecracker Protocol

Philip Levis and David Culler
UC Berkeley

Data Dissemination in Sensor Nets

- Sensor net: many low power, wireless “motes”
  - 1-10 KB RAM, 4-8MHz CPU, 10-100Kbs radio
- Dissemination: deliver a data item to every mote in a network
  - Configuration constants
  - Code updates, virtual programs
- Requires a continuous protocol
  - Transient disconnections, network repopulation
- Two metrics: energy efficiency, rate

Broadcast-based Protocols

- Every node forwards
- Energy efficient
  - Can use physical density, opportunistic receptions
- Slow: can't immediately forward
  - Suppression mechanisms, timers
  - CSMA: broadcast storms
  - RTS/CTS: control packet exchange latency

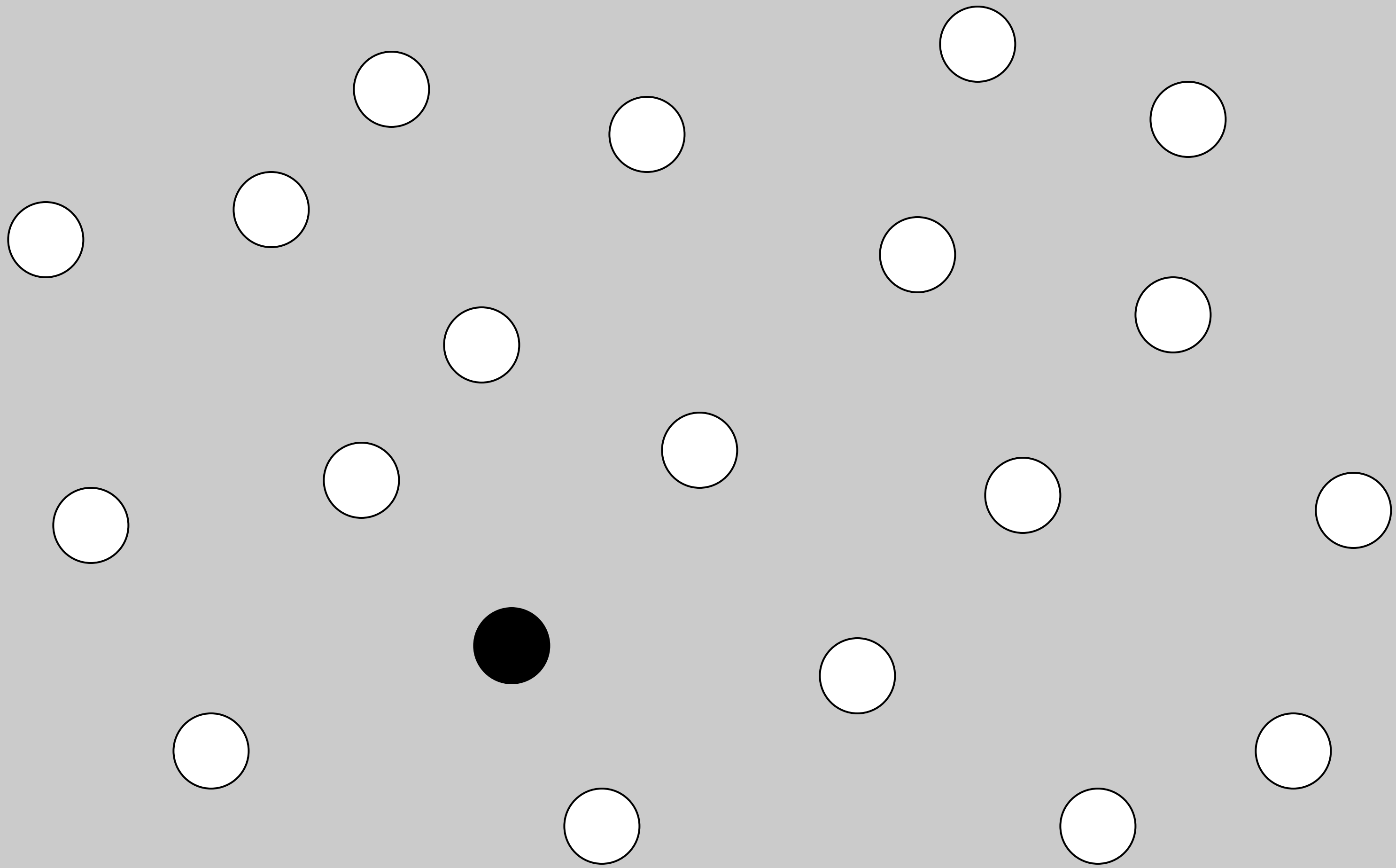
Routing-based Protocols

- One node forwards
- Fast
  – Next hop can immediately retransmit
- Energy inefficient: naming
  – Need many routes to reach entire network
  – Naming every node unfeasible

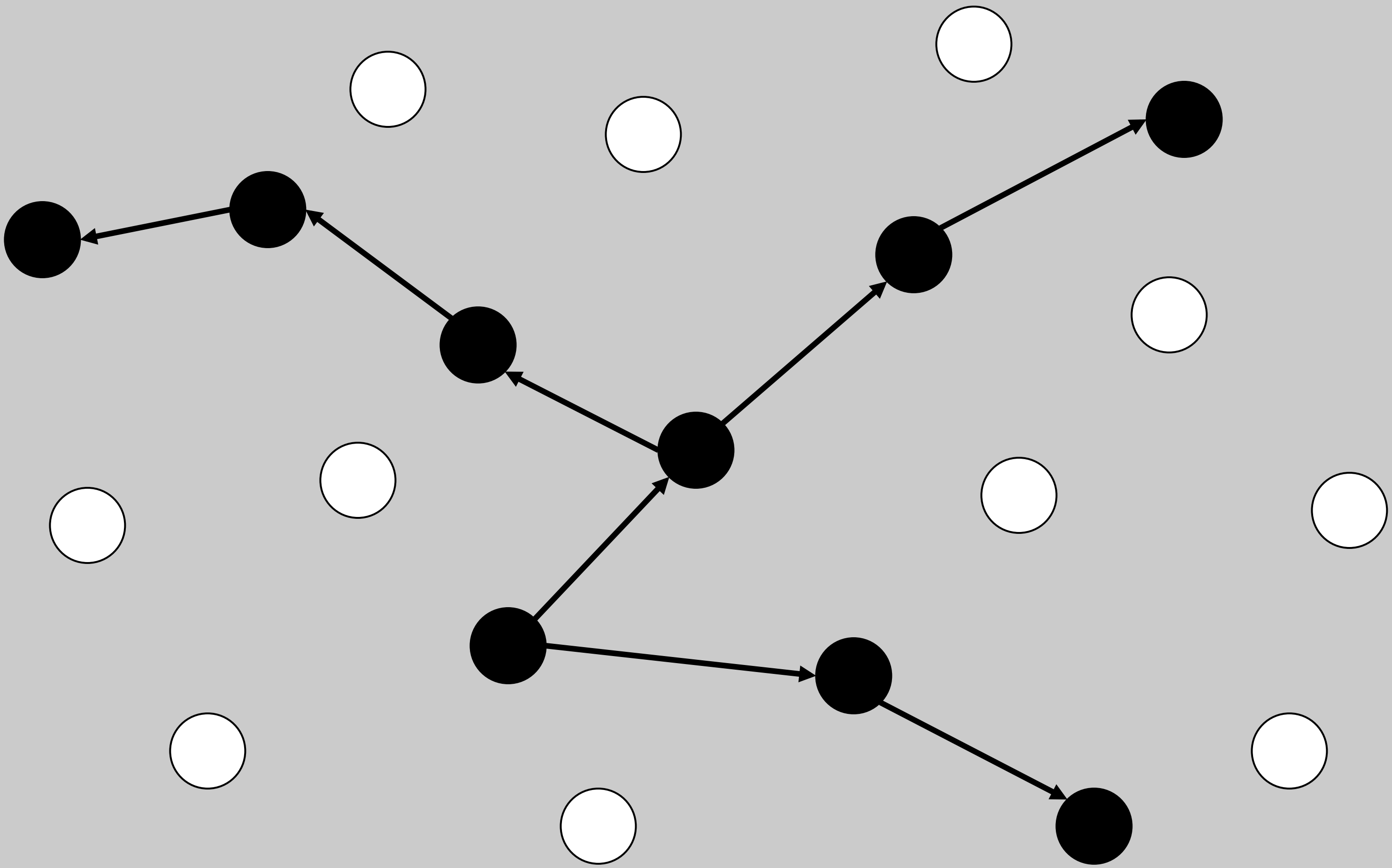
Firecracker Dissemination

- Combine routing and broadcasts
  - Routing’s speed
  - Broadcasting’s efficiency
- Seeding phase
  - Route data to distant points in the network
- Propagation phase
  - Start broadcasting from routes

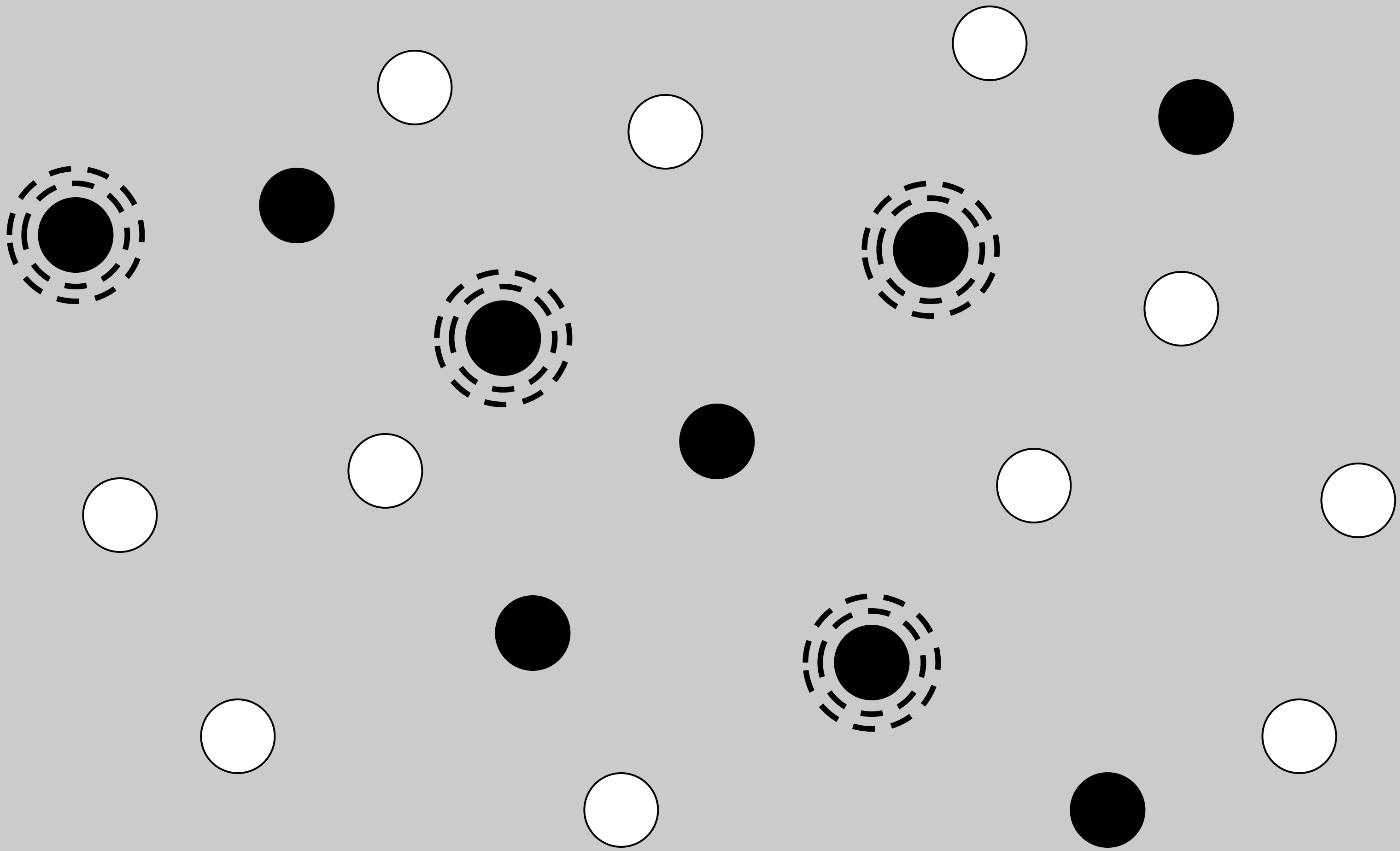
Firecracker Example



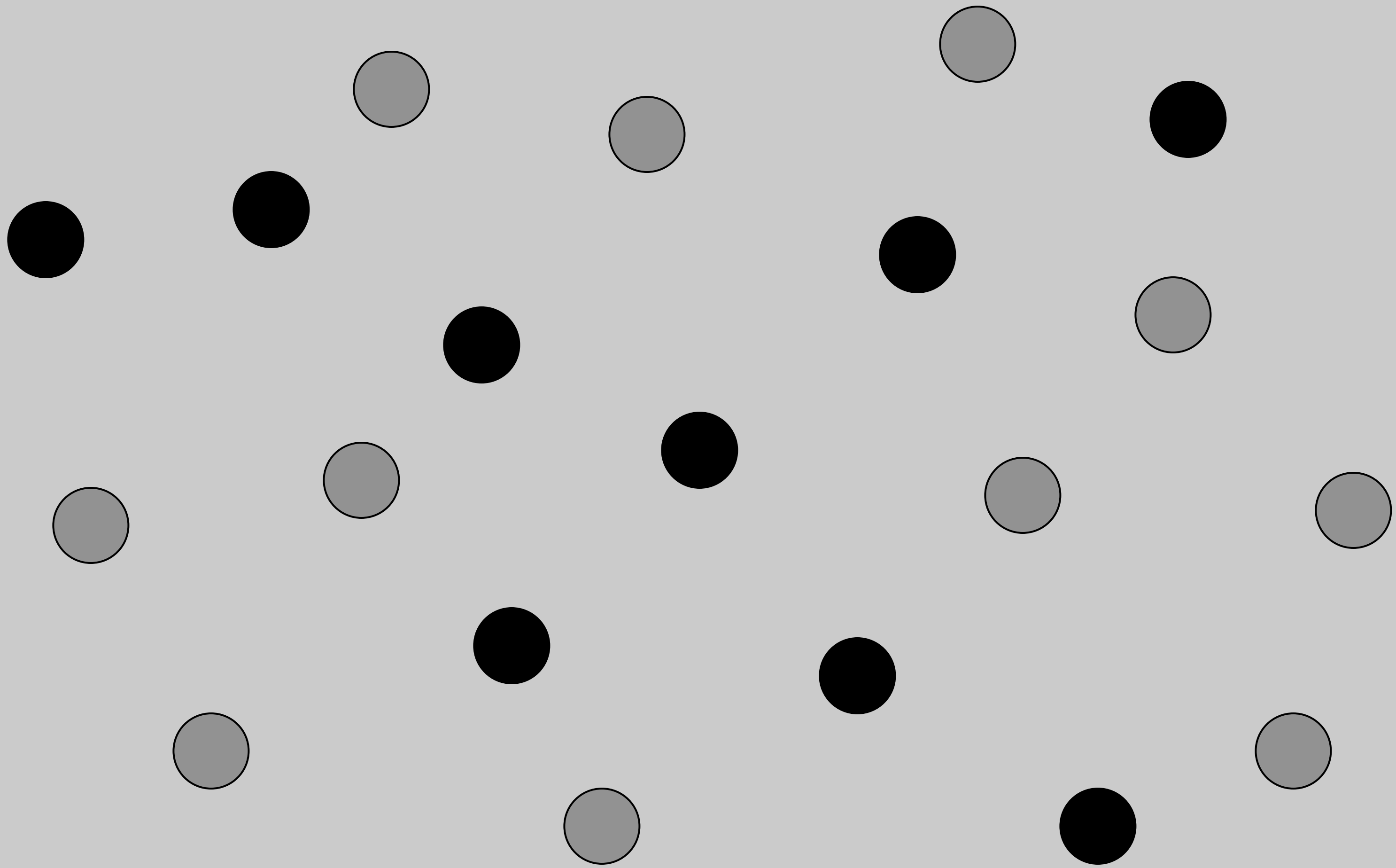
Firecracker Example



Firecracker Example



Firecracker Example



Outline

- Data dissemination
- Sensor networking, Trickle
- Firecracker
- Randomized Seeds
- Conclusion

Outline

- Data dissemination
- Sensor networking, Trickle
- Firecracker
- Randomized Seeds
- Conclusion

Sensor Networking

- Energy is critical, communication is costly
- Local wireless broadcast primitive
  - Unique node identifiers
- Many application requirements, many network protocols
  - Collection
  - Any-to-any (logical coordinates: GEM, BVR, etc.)
  - Local aggregation
  - Dissemination
    - Trickle

Trickle Algorithm

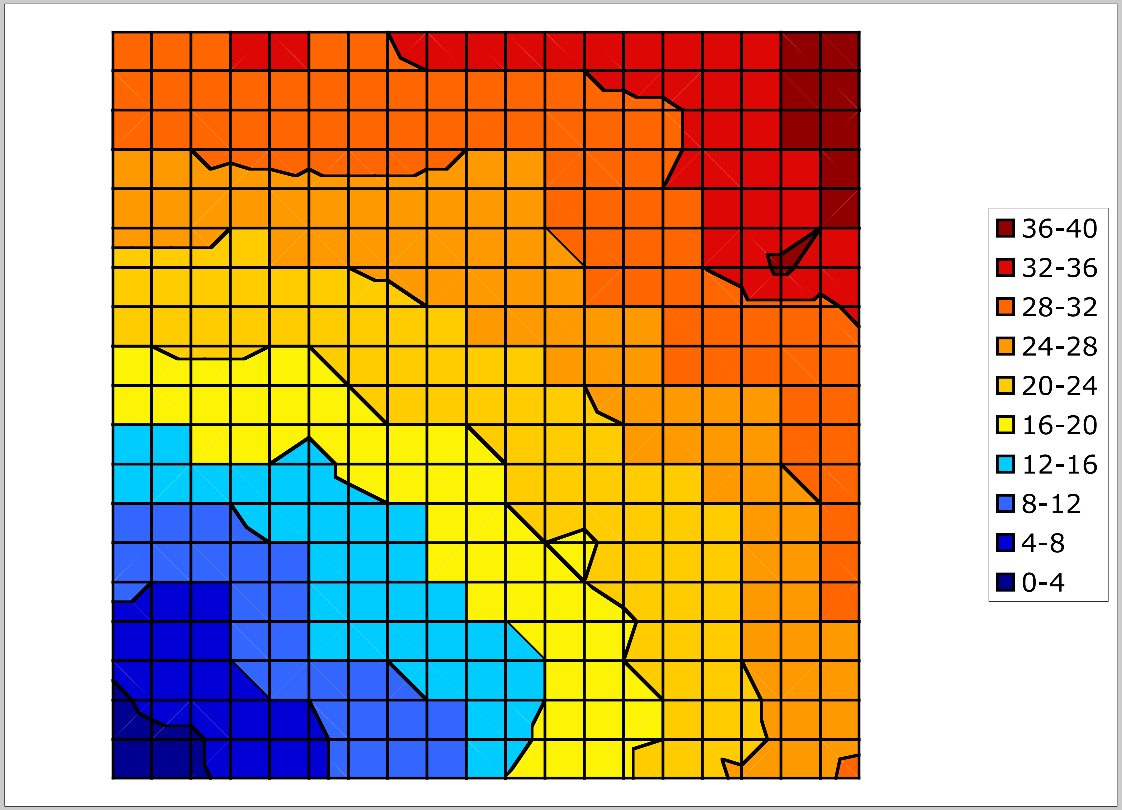
- Periodically broadcast metadata *M*
- Suppression interval of length *T*
- Pick a random point *b* in *T*
  - Broadcast unless you hear *M*
- When *T* expires, double it (up to a max)
- If you hear *M*+, make *T* very small (1 sec)
- If you hear *M*-, send an update
- Trickle plots

Experimental Methodology

- TOSSIM, a TinyOS simulator
- Compiles applications into a simulator engine
- Radio loss model based on empirical distributions
  - Asymmetric
  - Highly variable
- Unit disk interference model
- Bit-level or packet-level simulation
  - We used packet-level

Trickle Plot

- 20x20 grid (400 nodes)
- New datum
- 15 foot spacing
- 32 hop network
- Time to reception in seconds
- Wave of activity



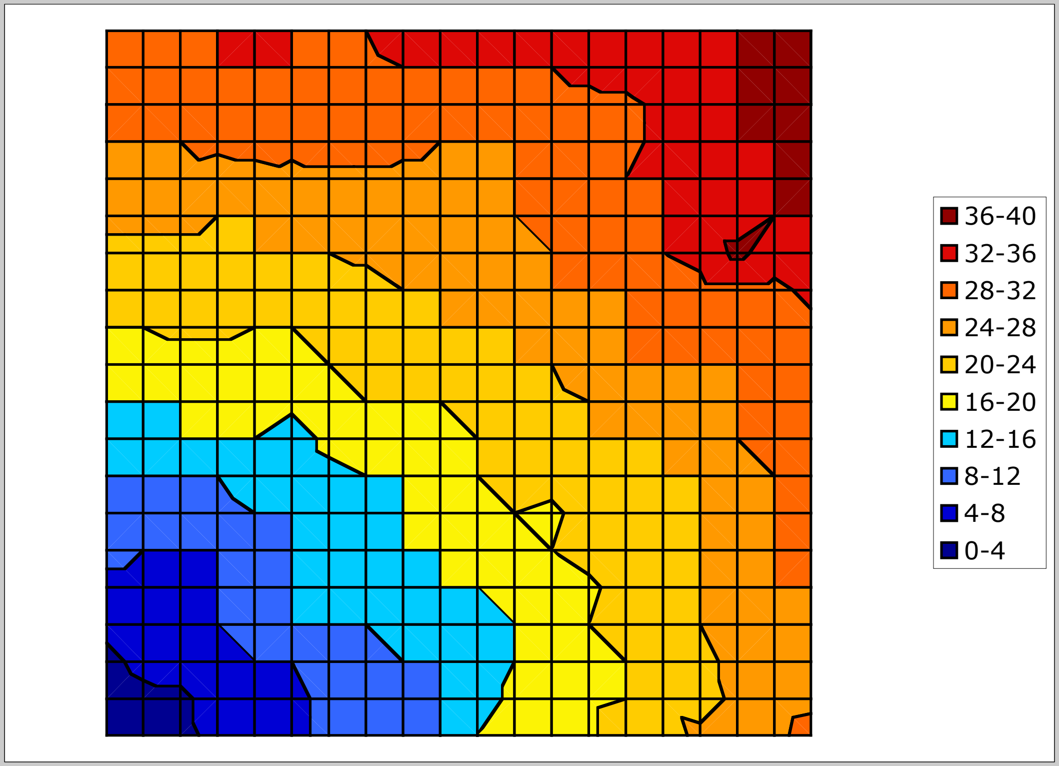
Outline

- Data dissemination
- Sensor networking, Trickle
- Firecracker
- Randomized Seeds
- Conclusion

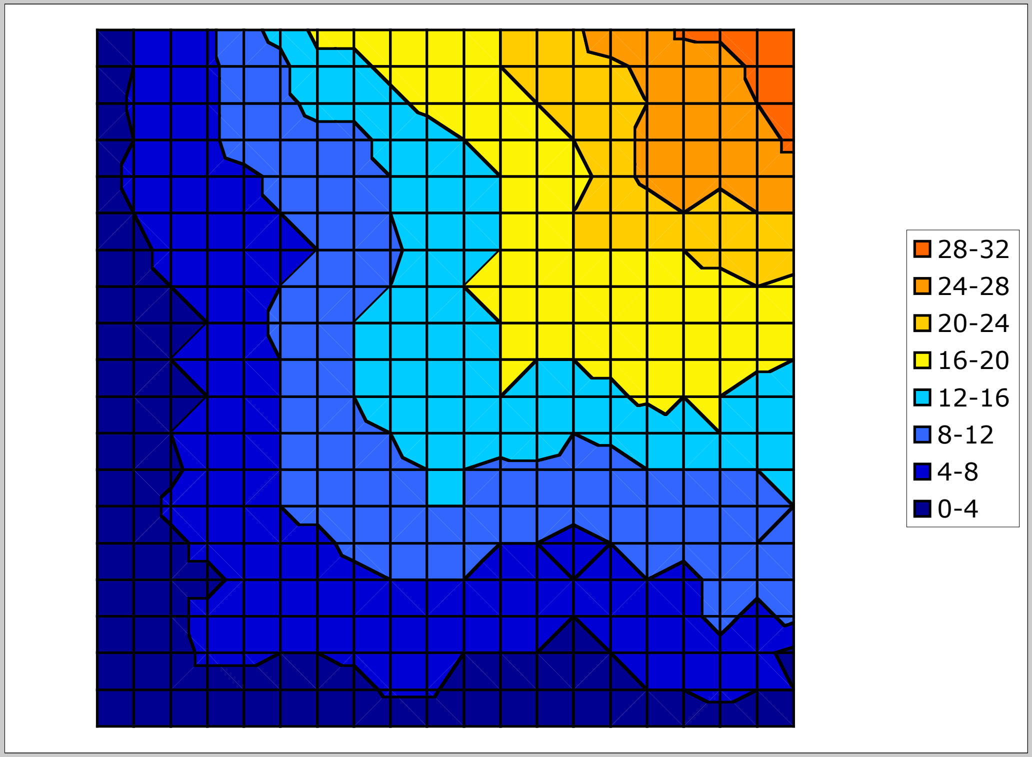
Firecracker

- Start disssemination by seeding network
- Route data to a few distant points
- Start broadcast dissemination along paths
  - Destination, route, snooping
- Example: corners on a grid-based protocol
  - Nodes can forward to manhattan neighbors
  - If two options, select randomly
  - Network density ensures manhattan links exist
- Same methodology as Trickle example

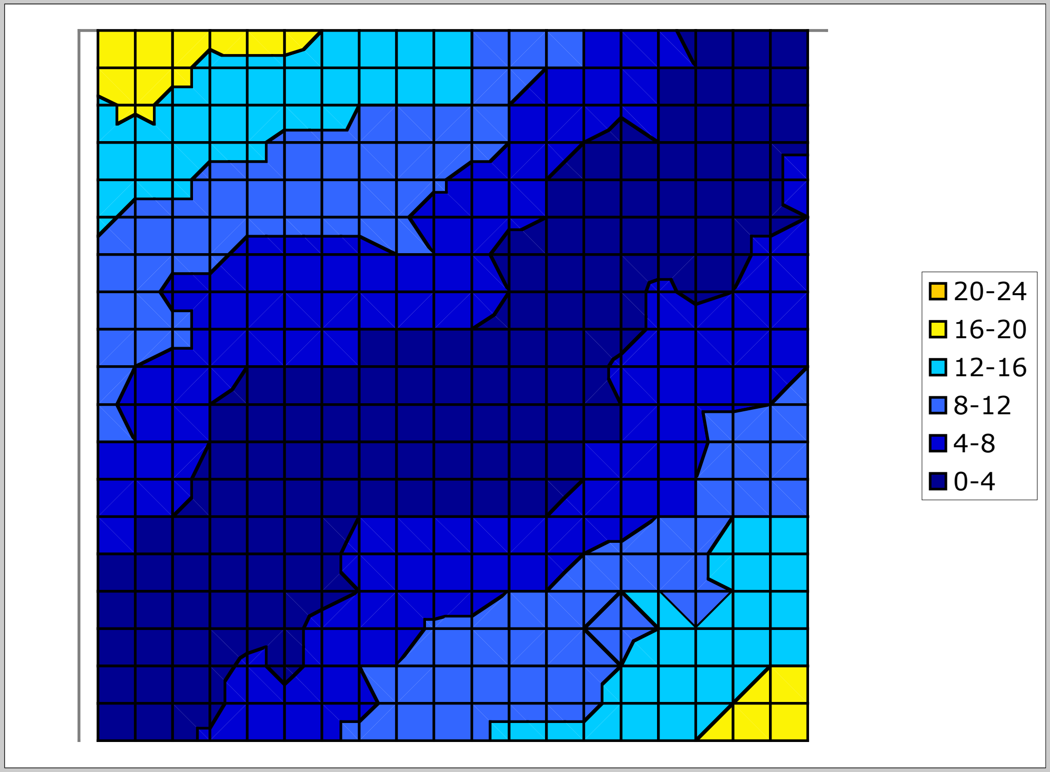
Basic Trickle



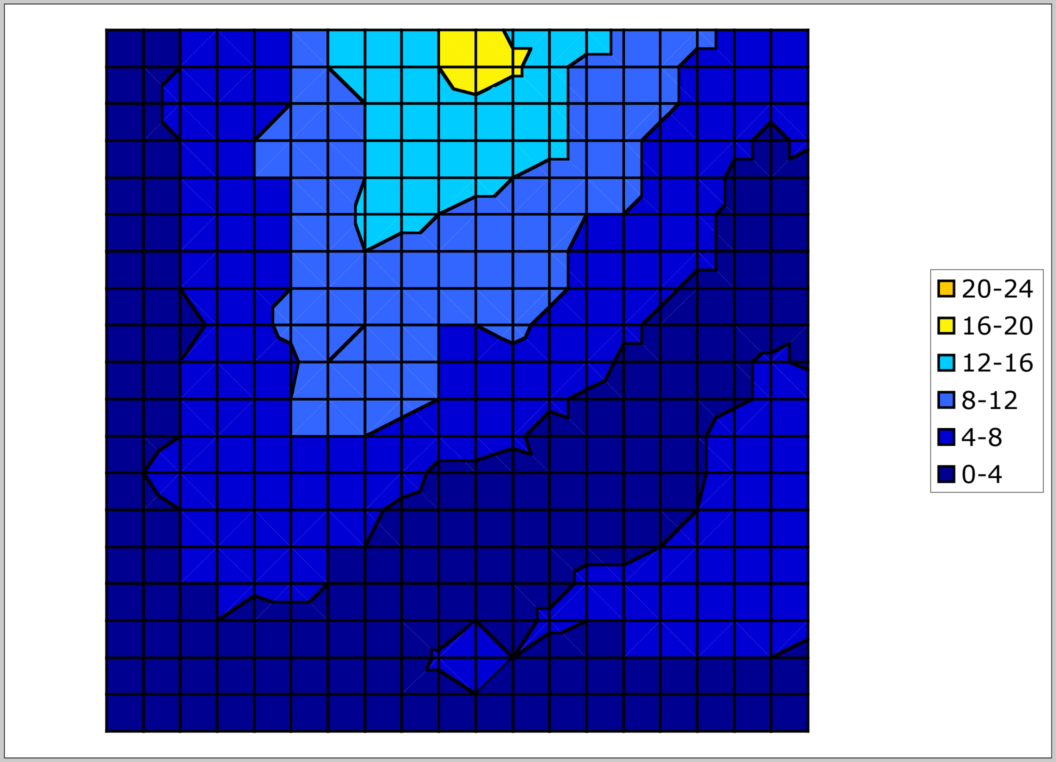
Adjacent Corners



Opposite Corner

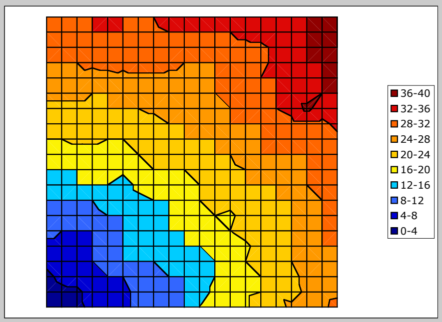


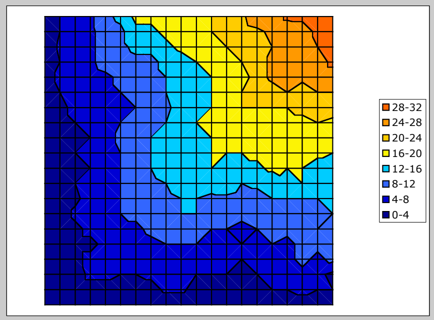
All Corners

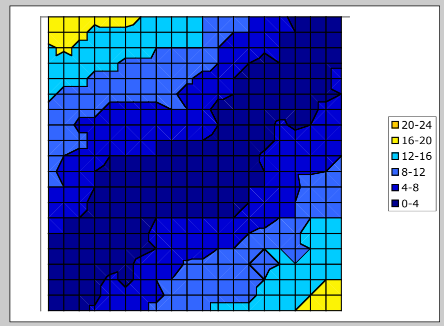


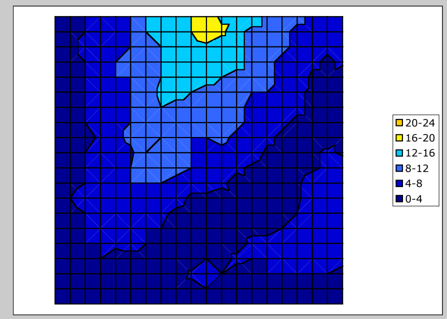
Reception Time Distributions

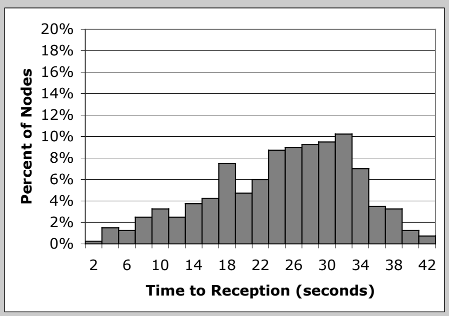
Larger domain (42 s)

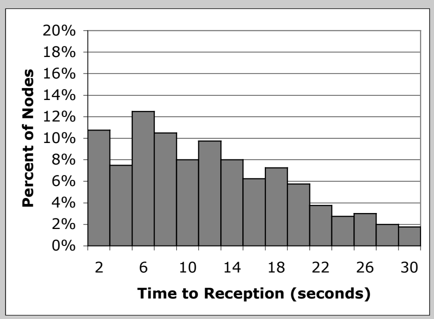


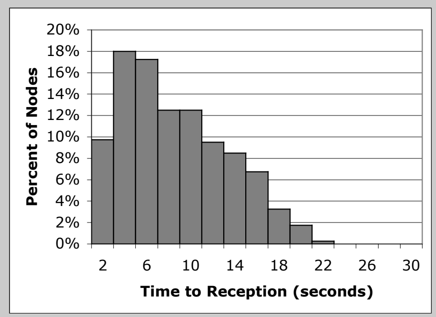


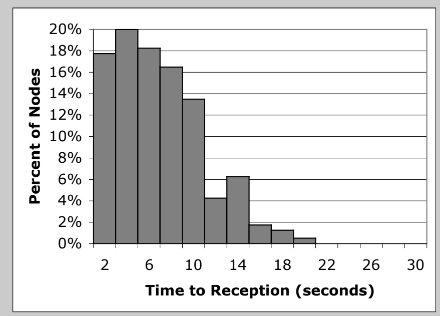






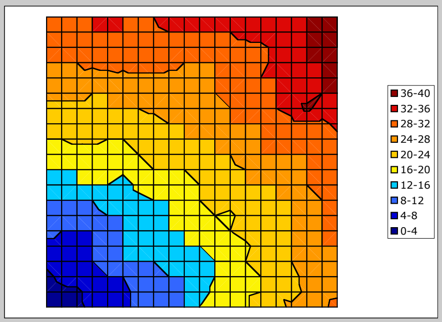


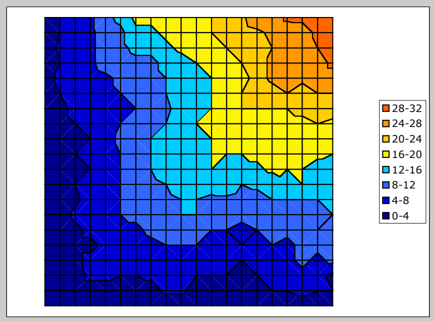


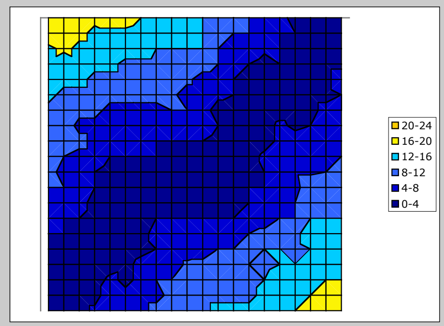


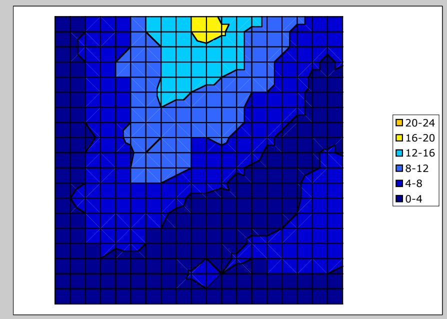
Reception Time Distributions

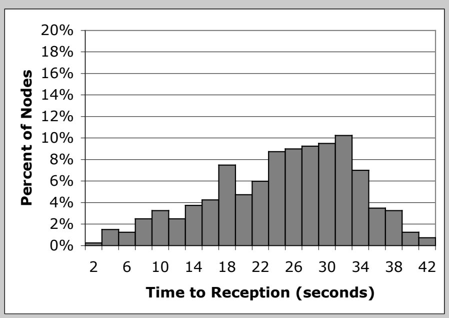
Larger domain (42 s)

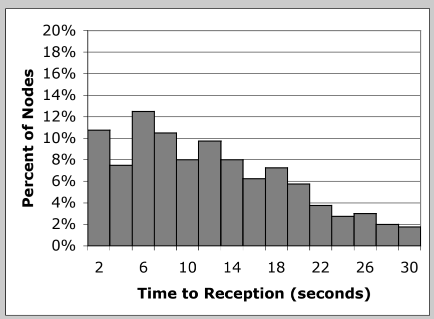


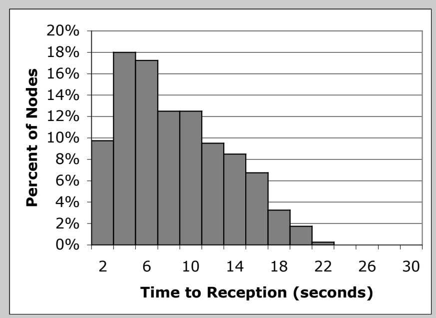


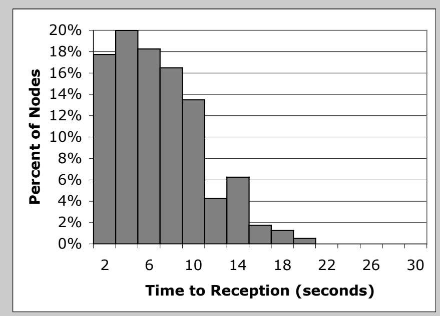












20835 Sends | 19544 Sends | 18275 Sends | 6665 Sends

Routing Reduces Cost

- Routing happens quickly
  - Synchronizes nodes
  - Trickle performance improves
- Fewer nodes need metadata exchanges
  - Metadata is most of the traffic

Hybrid Approach is Beneficial

- Distant seed points
  - Improves rate
  - Reduces cost
- Can’t assume knowing what distant points exist
  - Can’t store all the names
  - Need a way to select seeds
  - Randomization prevents corner cases

Outline

- Data dissemination
- Sensor networking, Trickle
- Firecracker
- Randomized Seeds
- Conclusion

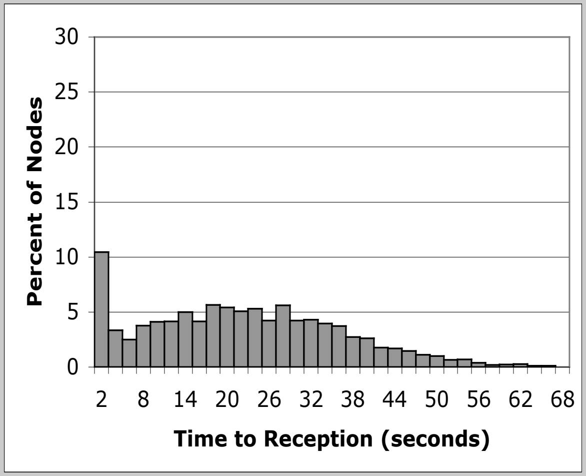
Experimental Methodology

- Use same grid arrangement
- Run twenty experiments, average results

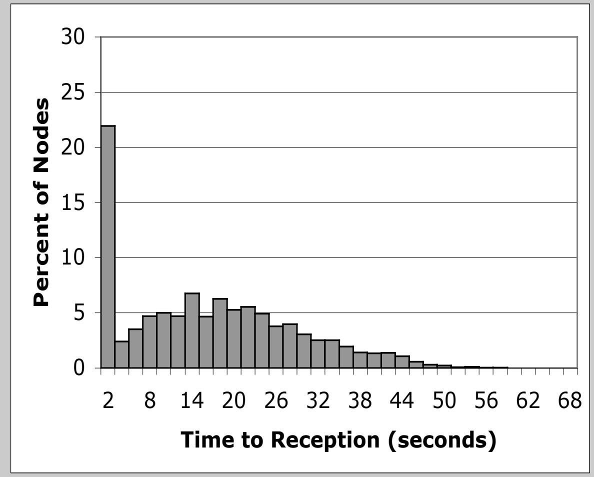
Four Policies

- From corner to one random node in the grid
- From corner to three random nodes
- From center to three random nodes
- From corner to three random distant nodes

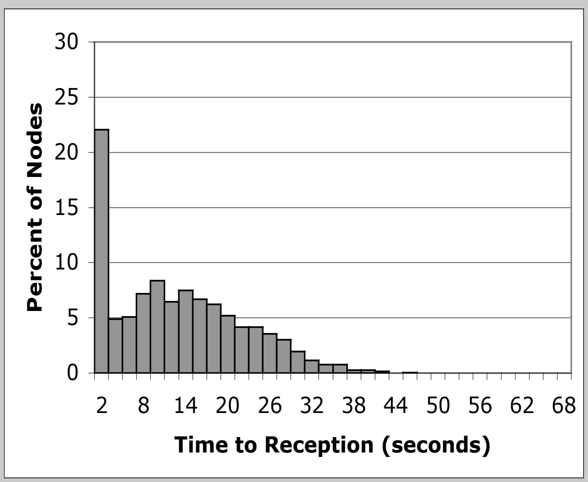
Histograms



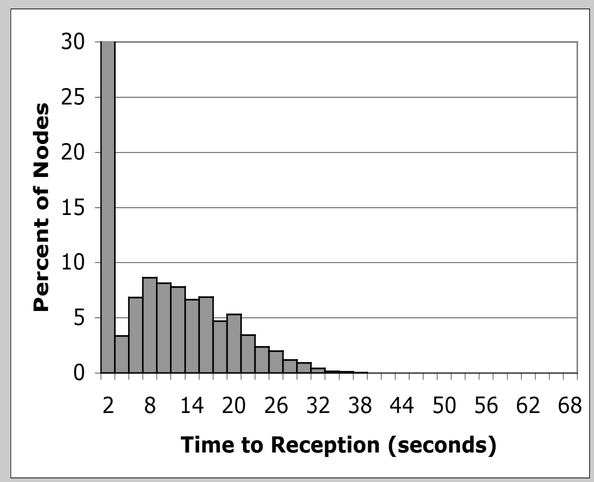
One from corner



Three from corner



Three from center



Three distant from corner

Results

- Picking random nodes works OK
  - Adding more does not improve results a great deal
- Coverage improves from center
- Distant nodes works best
- Need route to *edge* of the network
  - Logical coordinate spaces support this

Outline

- Data dissemination
- Sensor networking, Trickle
- Firecracker
- Randomized Seeds
- Conclusion

Network Protocols

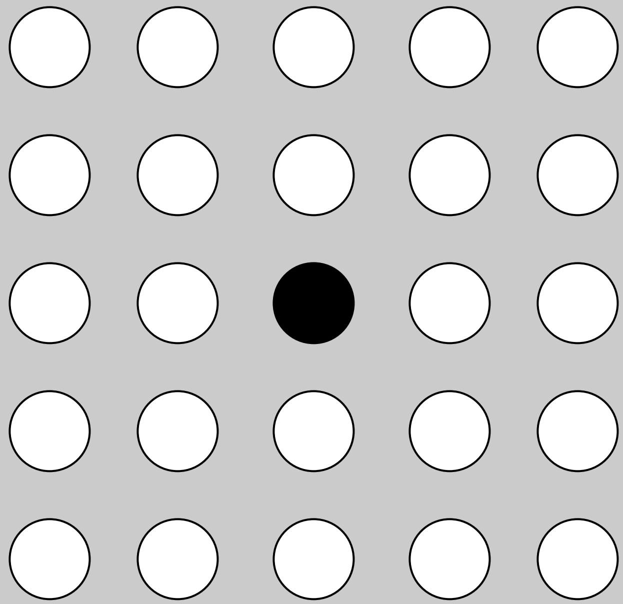
- Varying communication requirements
  - Collection (n to one)
  - Dissemination (one to n)
  - Diffusion (m to n)
  - Local Aggregation
- Forwarding predicates
  - Density estimation
- Predicate and media access interaction
- Routing's scoping enables fast propagation
- Slower broadcasts fill in the holes

Questions and Discussion

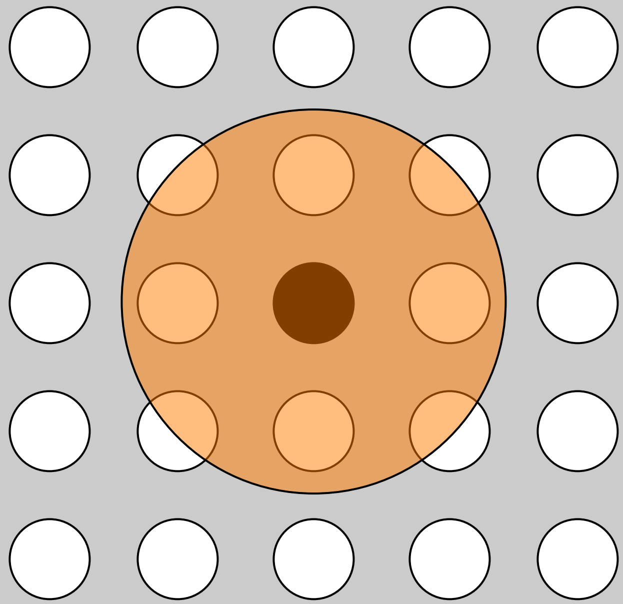
Any-to-Any Routing

- Current protocols use logical coordinates
  - GEM (Graph Embedding, polar coordinates)
  - BVR (Beacon Vector Routing, n-dimensional)
- GPSR uses geographic coordinates
  - Requires localization
  - Virtual coordinates may be possible
- Data dissemination benefits from being able to name a distant node (*network* distance)

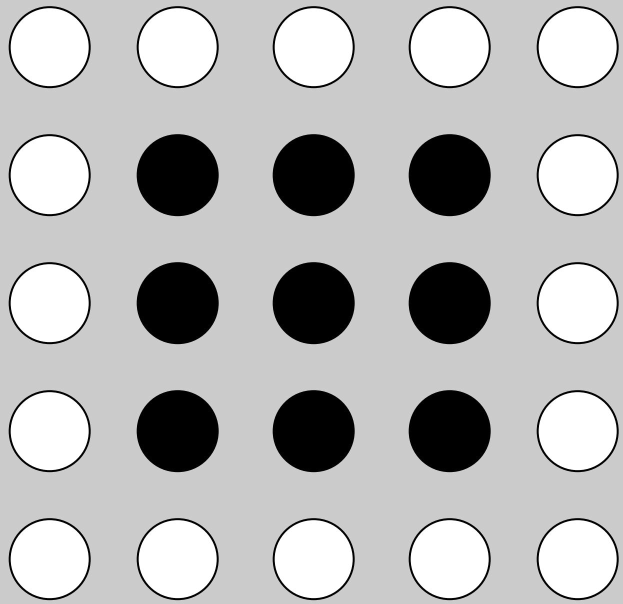
Broadcasting



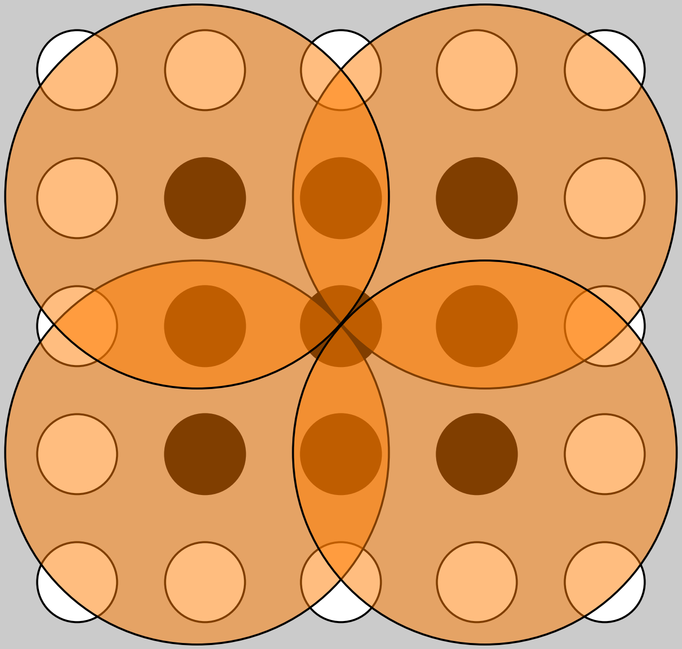
Broadcasting



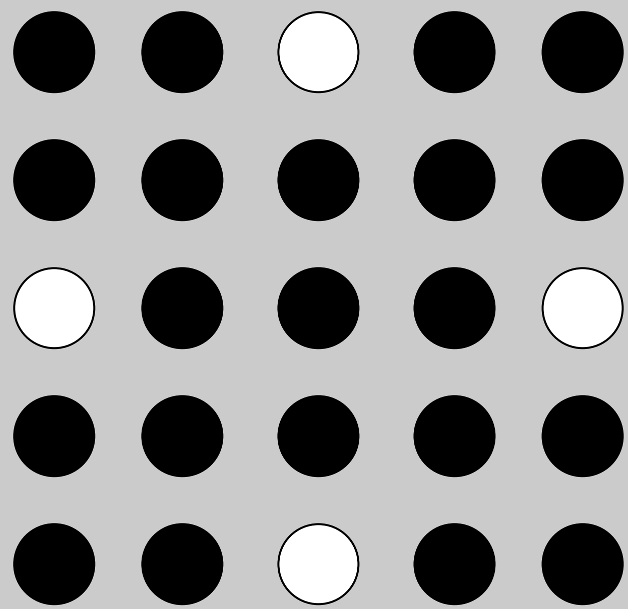
Broadcasting



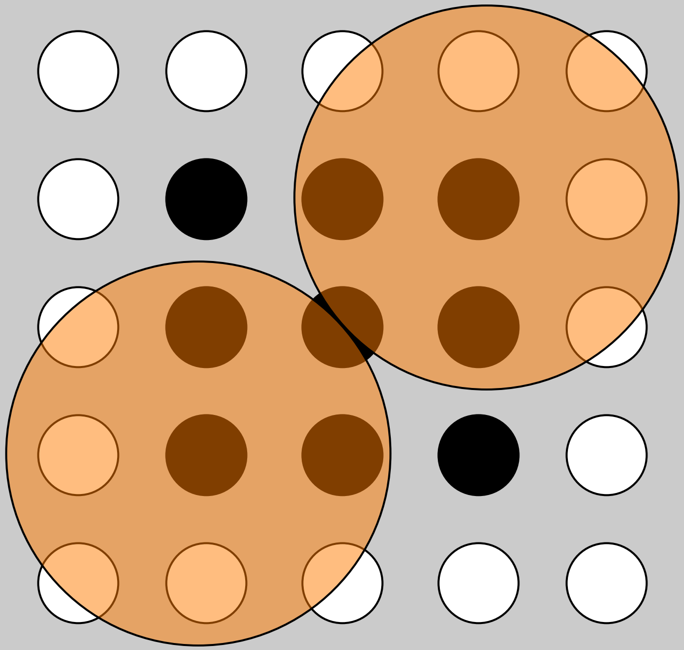
Broadcasting



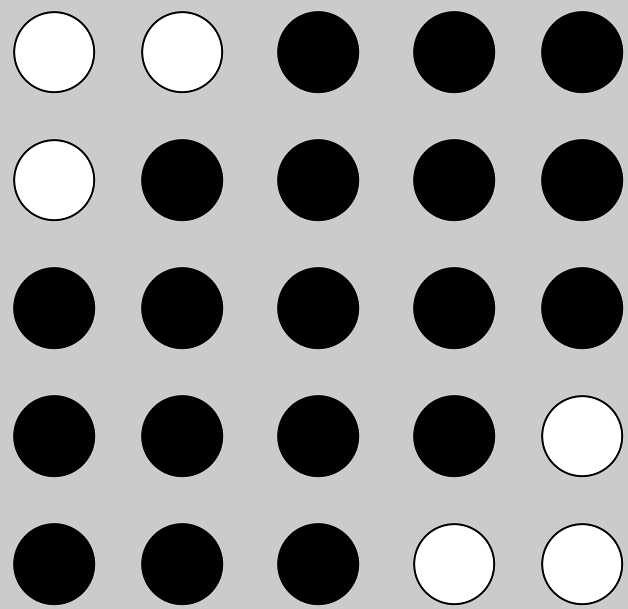
Broadcasting



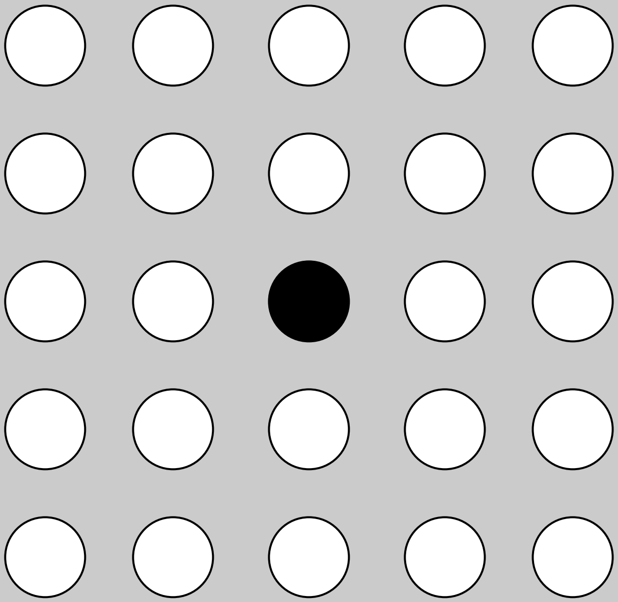
Broadcasting



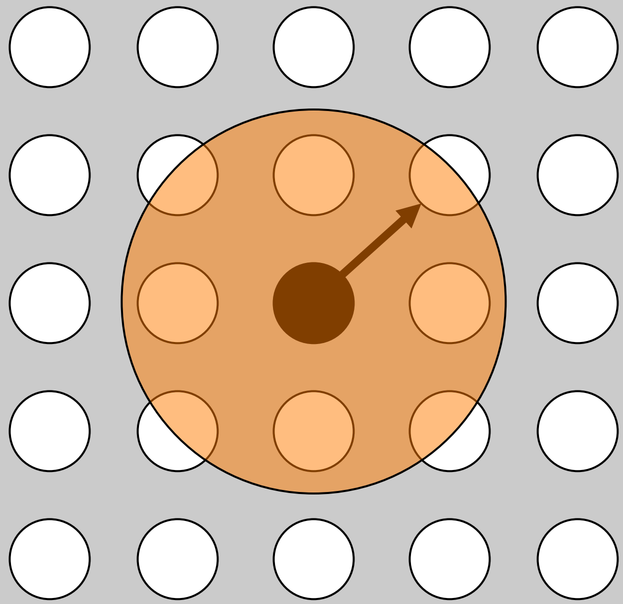
Broadcasting



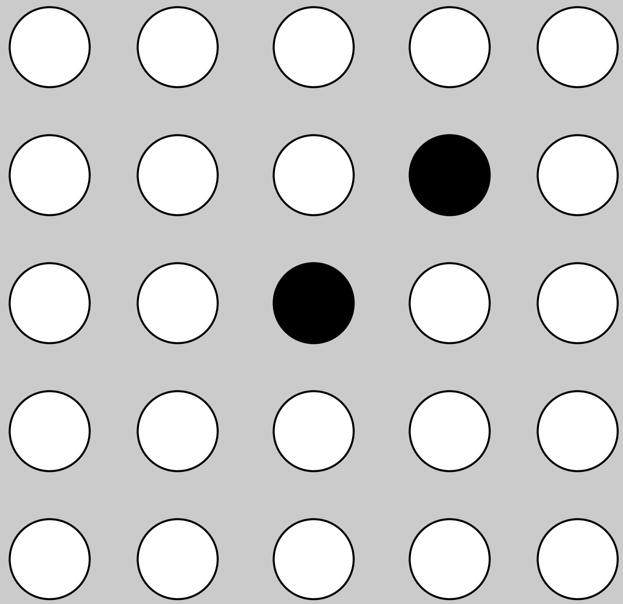
Routing



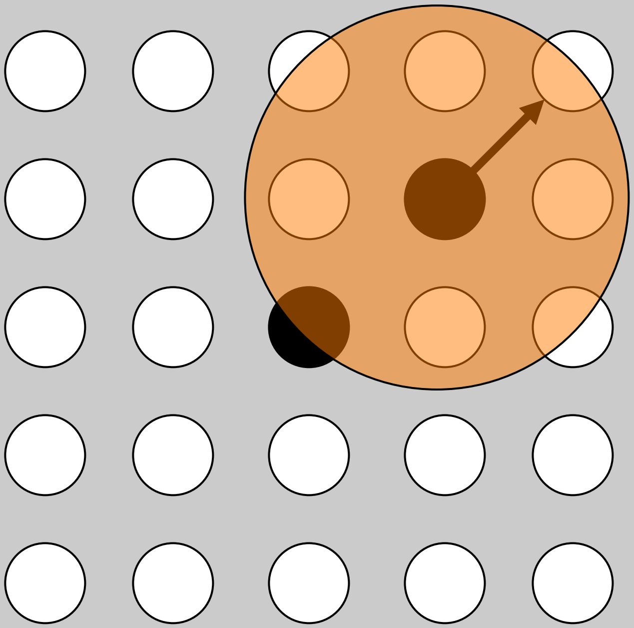
Routing



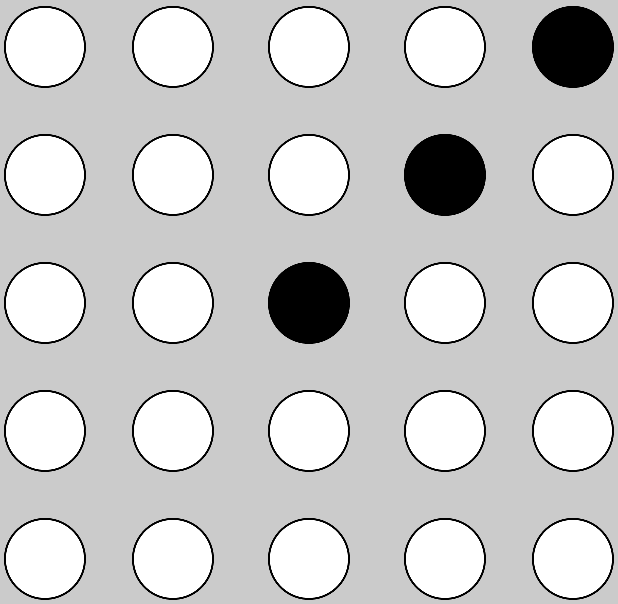
Routing



Routing



Routing



Routing, With Snooping