

Location in SpeckNets

Ryan McNally
Edinburgh University
rmcnally@staff.ed.ac.uk

Outline

- SpeckNet
 - Unfriendly environment
- Algorithm constraints
 - Existing approaches not quite right
- Two location algorithms
 - For bare-bones networks
 - For nodes with additional sensors

Specks

- Equipped with computation, storage, communications, sensors, power source
- Physically small
 - Limited capabilities – especially power
- Mass produced
 - Limited testing

SpeckNet

- Large numbers of specks
 - Redundancy, sensory fidelity
- Uncontrolled deployment
 - Reduces deployment difficulty/costs
- No special infrastructure
 - Compute servers, GPS beacons, etc
- Possibly mobile

Algorithm requirements

- Failure tolerant
 - Specks run out of power, eaten by wildlife, destroyed by meteorites, etc
- Resource efficient
 - Computation, storage, communication, power
- Scalable
 - Possibly large networks
- Adaptive
 - Mobile networks change topography

Location

- Extremely useful for specks to know their location
- Routing
 - No need for routing tables
- Data context
 - “Node 4278 has detected a fire!”
 - “Great! Where?”
 - “ ... ”

Existing approaches

- Compute server
 - SPoF, data density, dynamic network
- Super-nodes
 - Deployment, failure consequence
- Range Sensing
 - Difficult to get right
 - Radios are *terrible* at it
 - Ultrasonics are bulky/noisy/power-hungry

Algorithm approach

- Neighbours imply constraints on location
 - Specks gather information about their neighbourhood, compute a location estimate, and share it with their neighbours
- Self-reliant
 - Specks only directly concerned with their own location, helping others is incidental
- Run continuously, converge on ideal solution
 - Easy accuracy/consumption trade-off

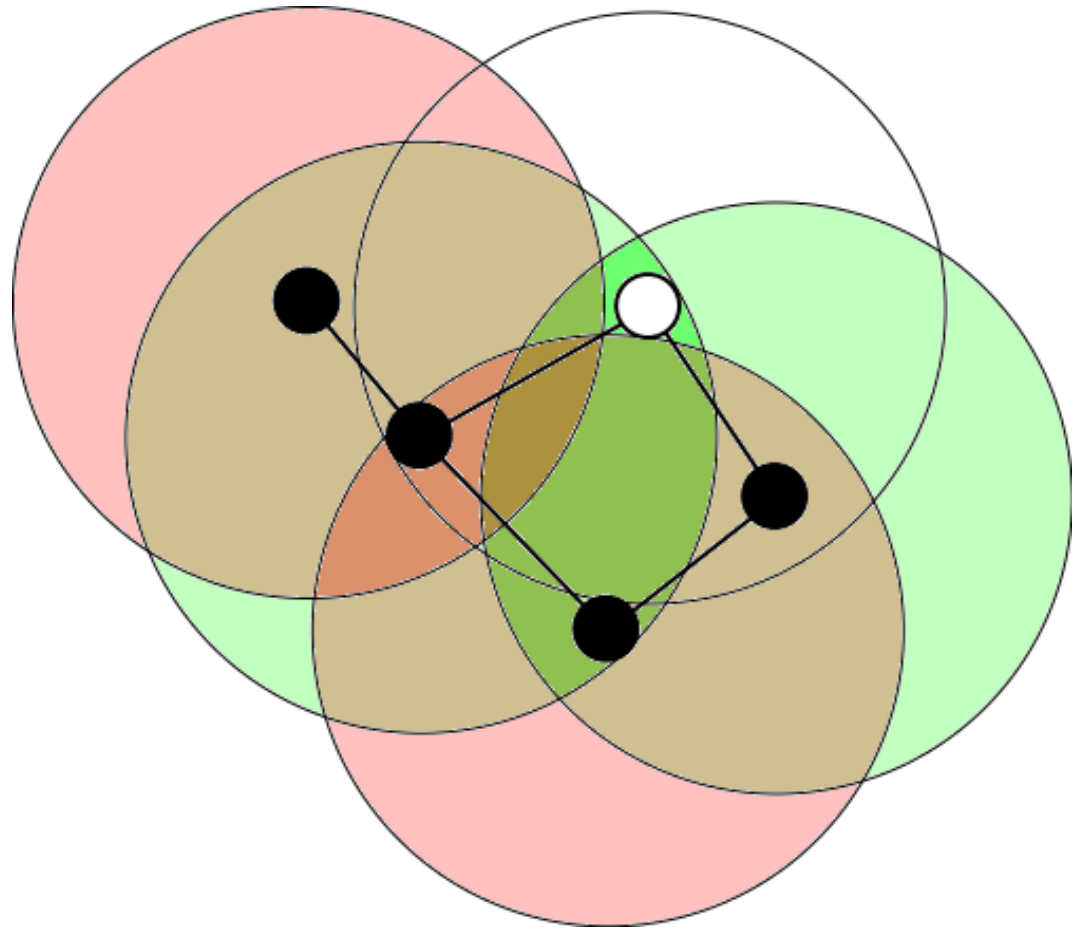
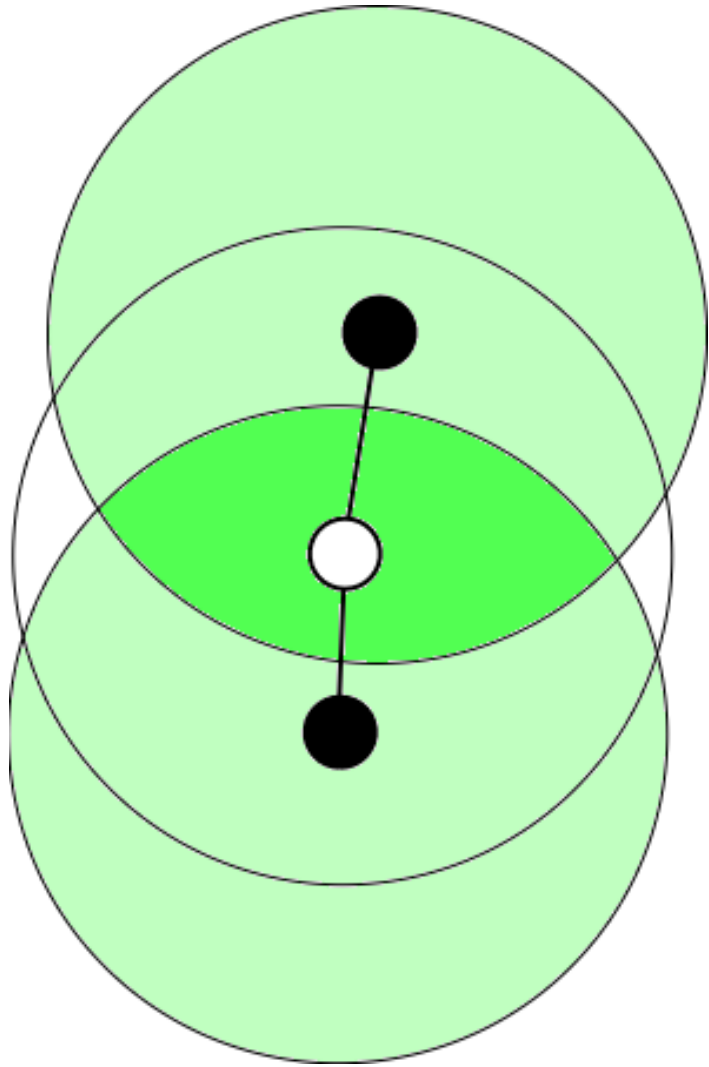
Agreeing on an origin

- Specks must agree on a common reference point for their coordinate system
- Each node starts life thinking that it lies at the origin
- Defers to specks with lower ID
- Smells like a single point of failure, but isn't
 - The agreement on a reference point is the important bit, it doesn't matter if the lowest ID speck fails

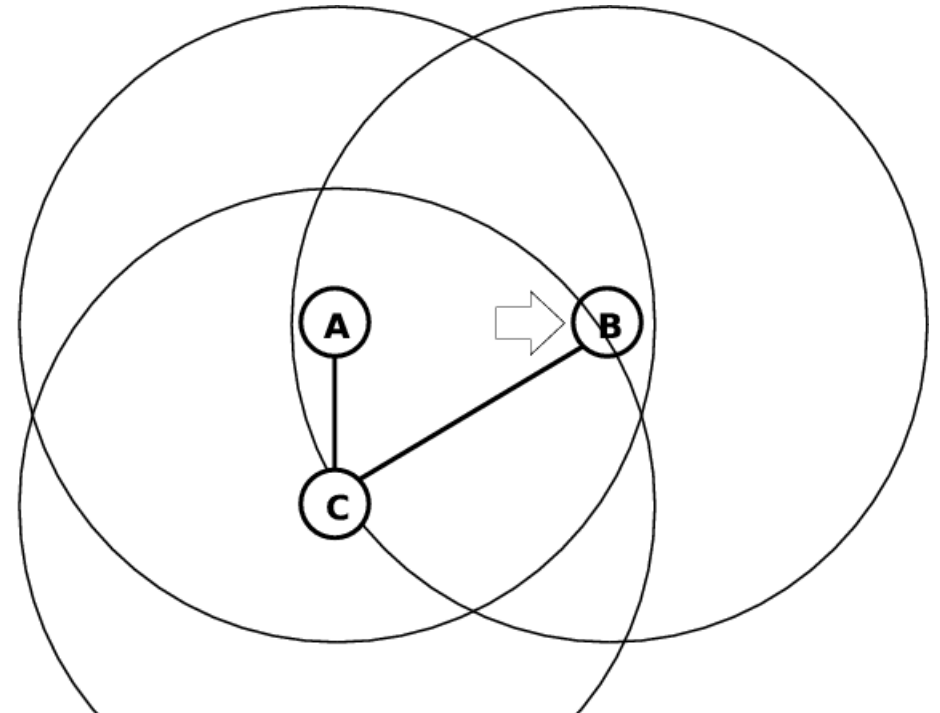
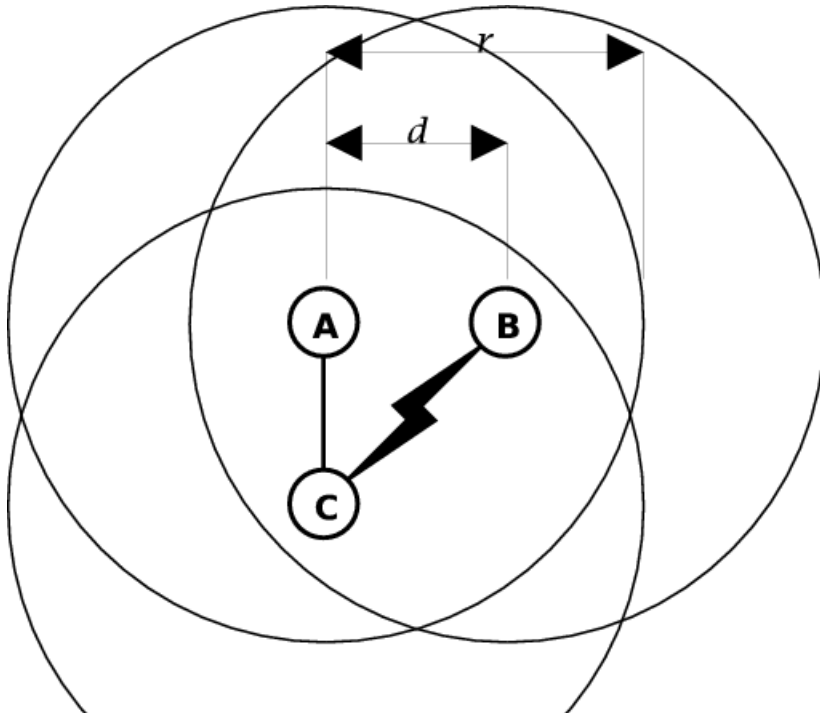
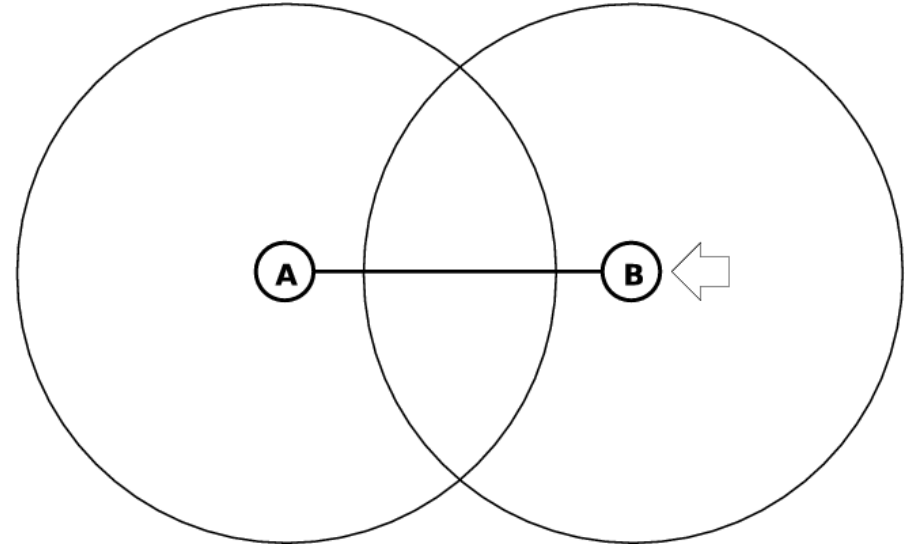
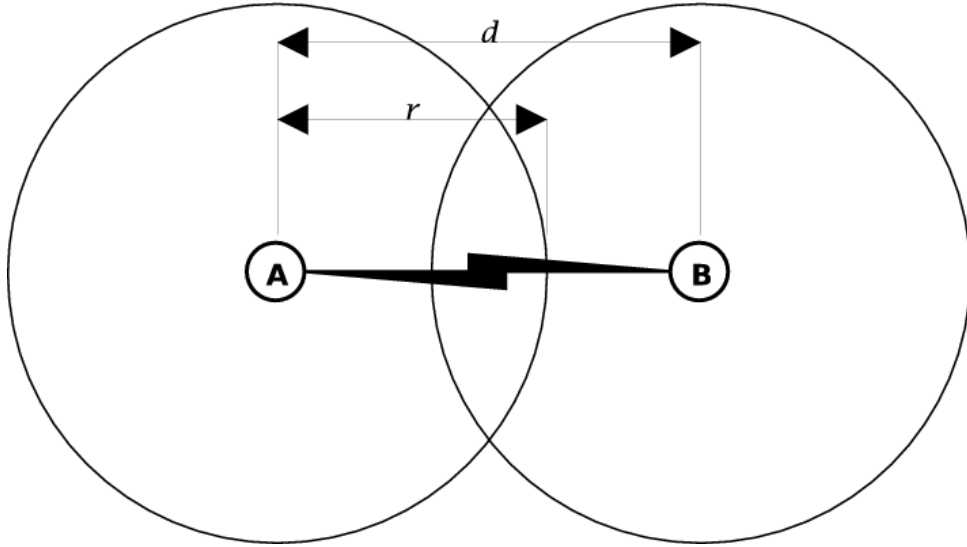
Distributed Relaxation

- Nodes have only basic capabilities
 - Radio communications
- Radio links imply distance constraints
 - 1-hop neighbours must be within radio range
 - 2-hop neighbours may be outside of radio range
- Simple numerical technique to solve constraints
 - Repeatedly solve each constraint in isolation to converge on a solution to all constraints

Constraints



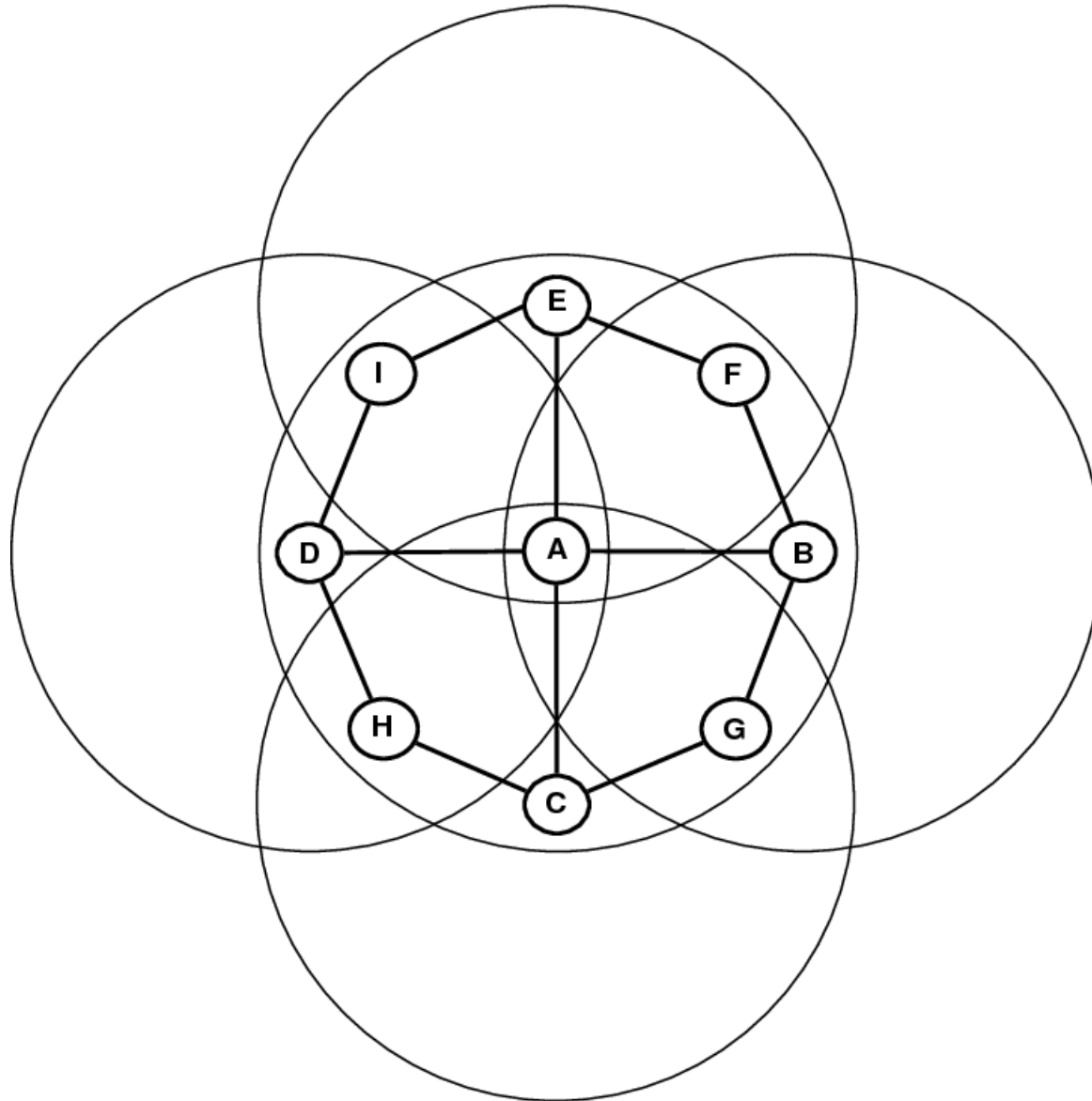
Solving constraints



Seeding the system

- In the beginning, no node knows where it is
 - Constraints are meaningless
- Require a small cluster of nodes to have a reasonable location estimate
 - System can grow from there
- Manual seeding...
 - Choose some specks, tell them where they are
- Exploit mutual ignorance

Seed formation



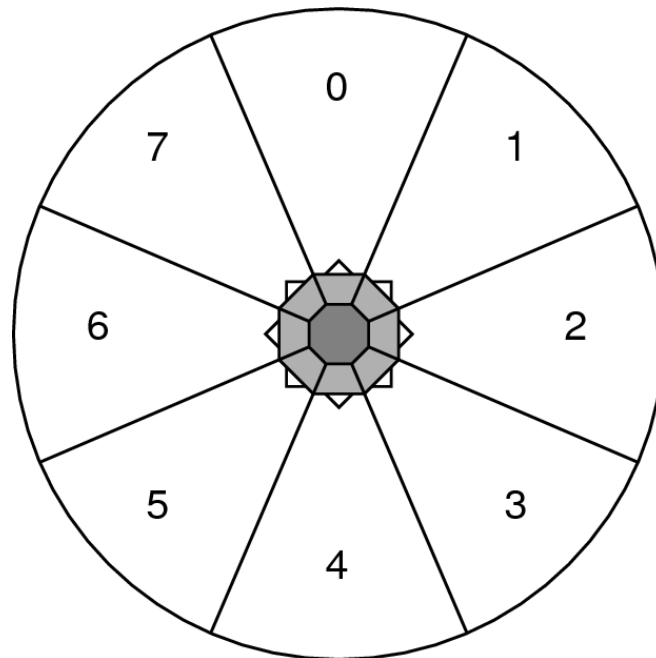
Message contents

- Own
 - ID
 - Origin ID
 - Location estimate
- List of neighbours'
 - IDs
 - Location estimates

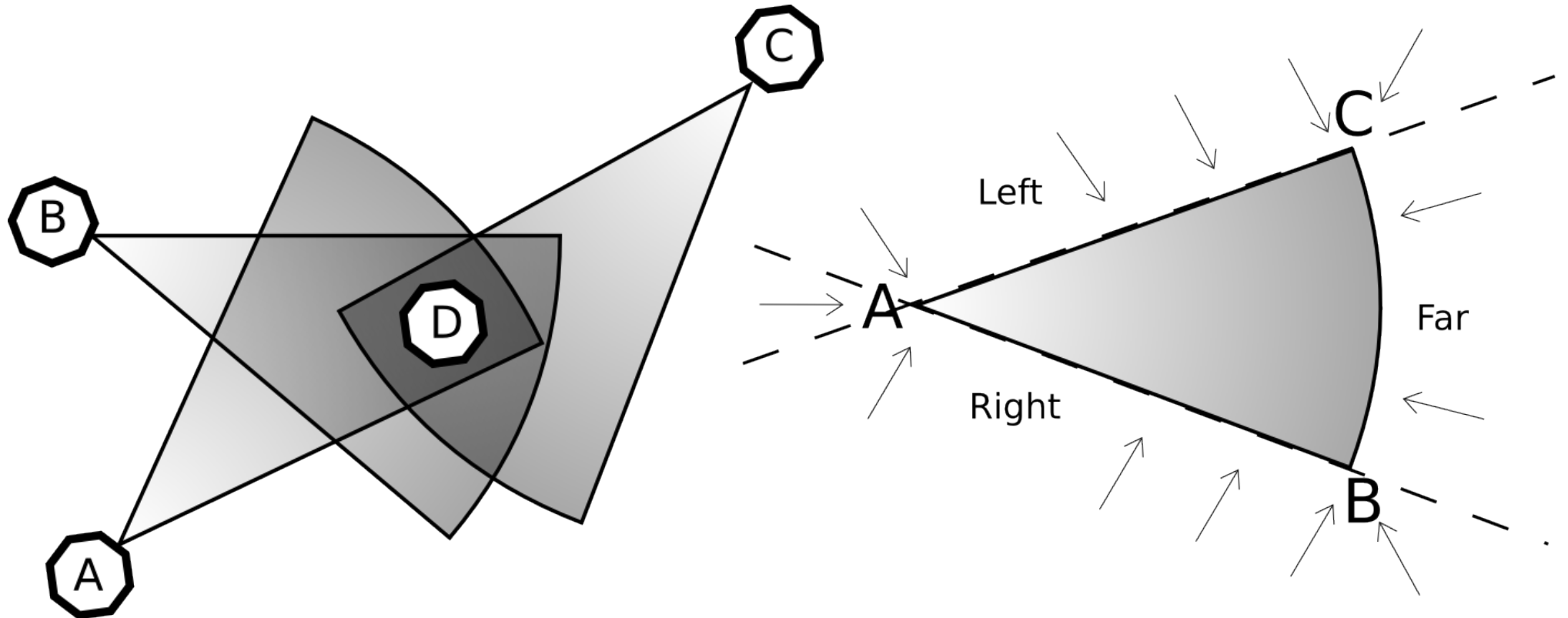
Demo

Sectoring

- Nodes have arrays of optical transceivers
 - Can make directional transmissions and receptions
 - Useful for more than location – reduces channel contention, unwanted reception

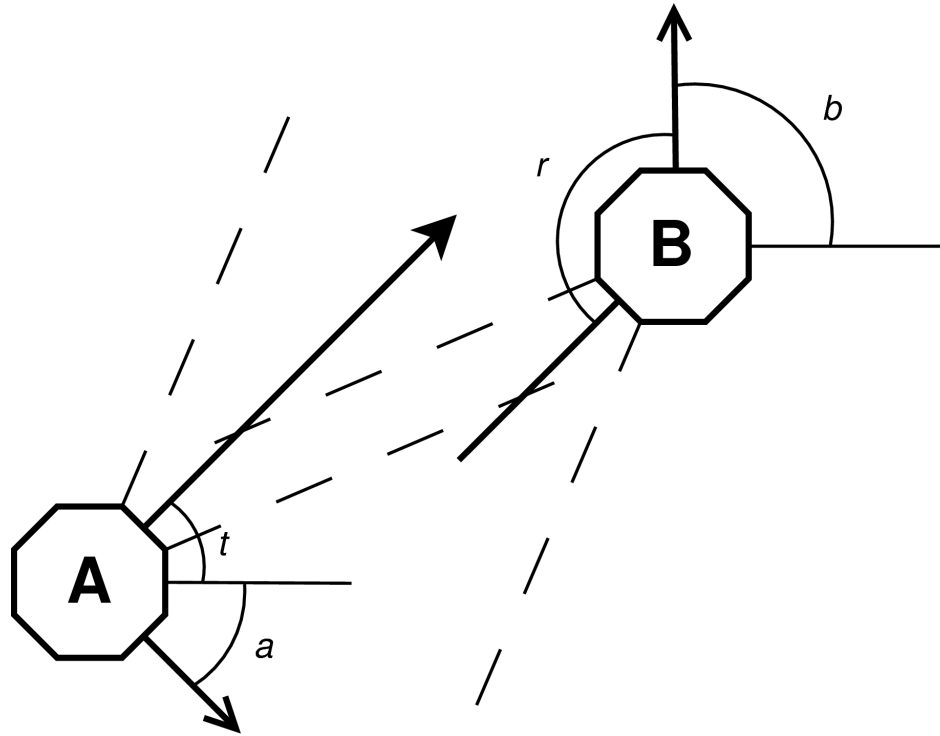


Constraints



Angle estimation

- Nodes need to compute their orientation in order to predict their transmission direction



Message contents

- Own
 - ID
 - Origin ID
 - Location estimate
 - Transmission angle estimate

Demo