Greedy Virtual Coordinates for Geographic Routing

Ben Leong
National University of Singapore

Barbara Liskov, Robert Morris
Massachusetts Institute of Technology

Background

- Geographic routing is a promising approach for wireless networks
  - Each node has an x-y coordinate
  - Stores little (constant) state per node
  - Easy to repair

Geographic Routing

- Try greedy forwarding
- Dead end
  - switch to guaranteed routing mode
  - either face or hull tree routing
- Whenever possible, switch back to greedy forwarding
  - because greedy forwarding gives good performance [Xing et al., 2004]

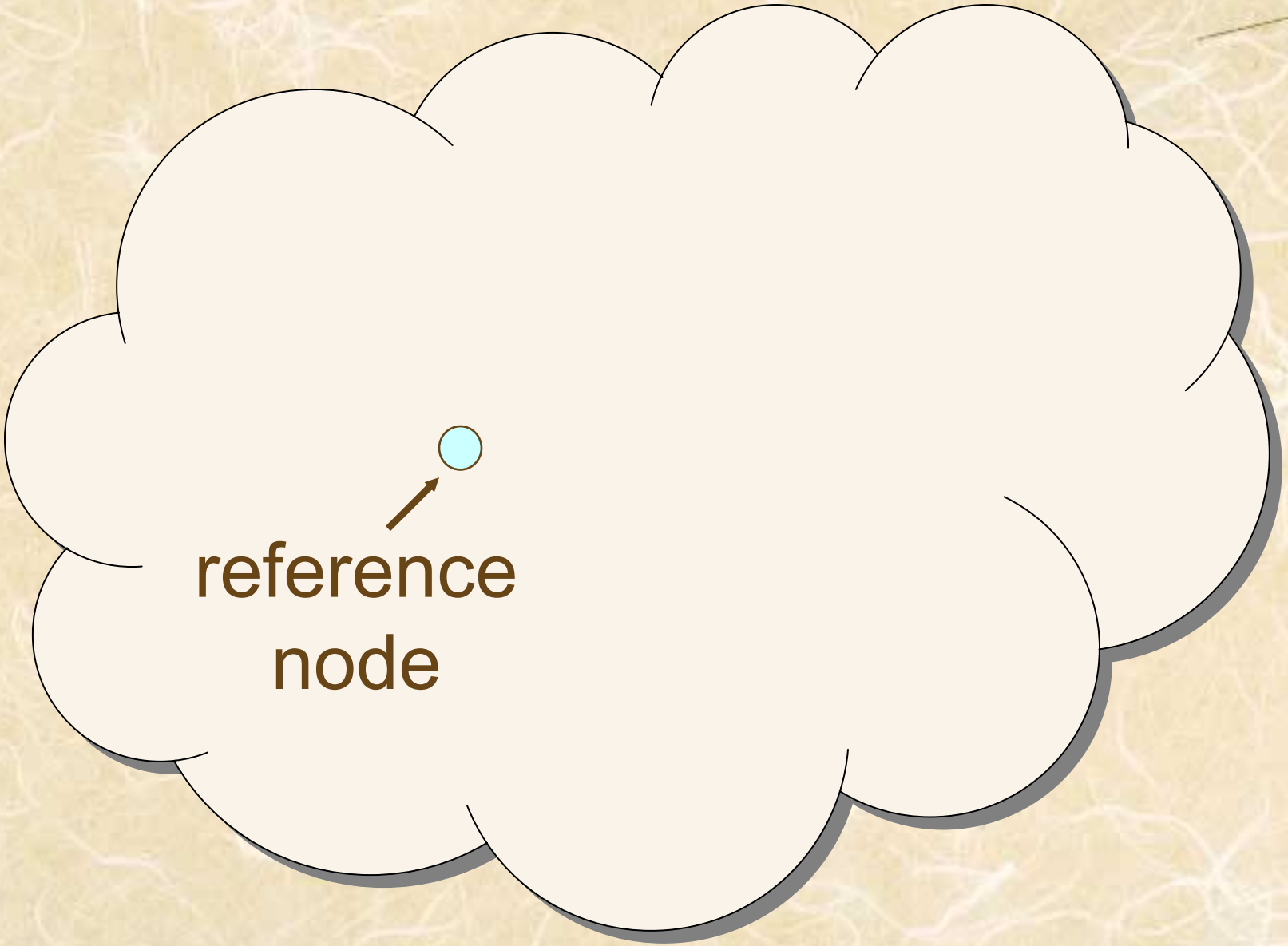
Case for Virtual Coordinates

- Not always feasible to have GPS for each node
- Virtual coordinates are sometimes better, e.g. sensornet on ship
- Physical locations are not required (Rao et al., 2003)
- Previous work: good for dense networks
- Know: greedy forwarding is efficient
- Challenge: *can we assign coordinates so that greedy forwarding always works?*

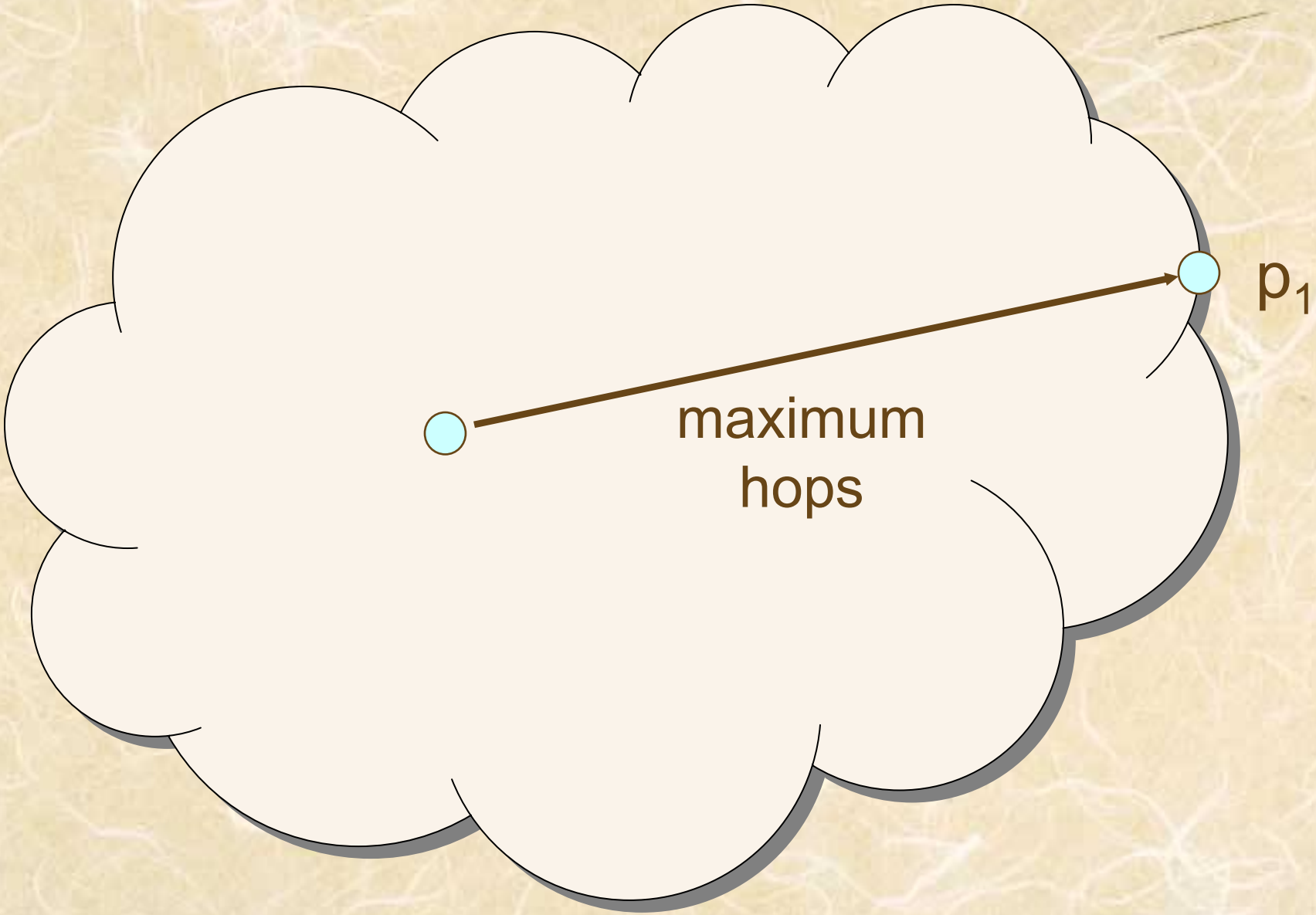
Greedy Embedding Spring Coordinates (GSpring)

- Start from initial coordinates
- Simulate physical spring system with repulsion forces
- Incrementally adjust nodes to make topology more convex
- Introduce damping and hysteresis to ensure system converges

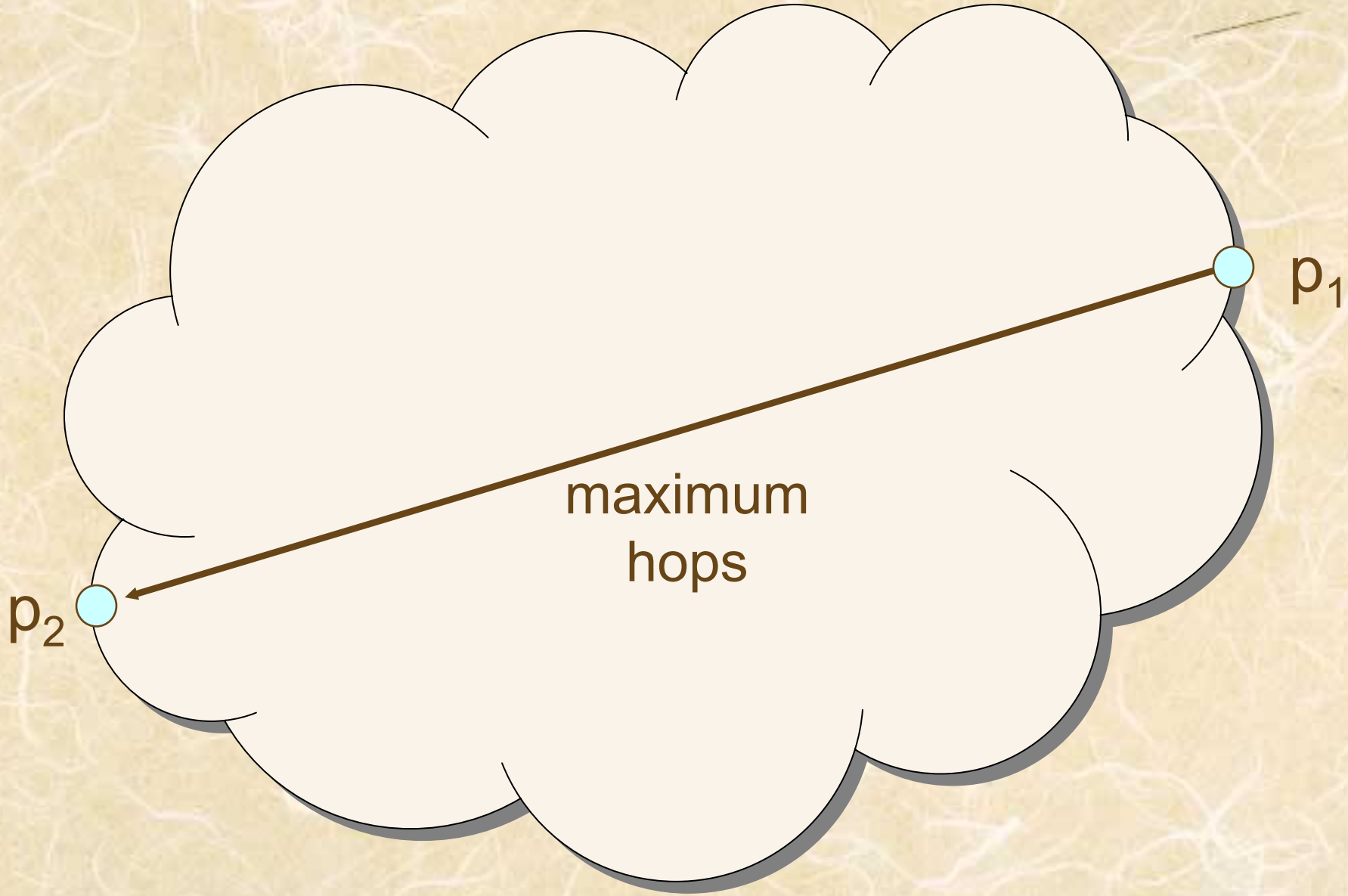
Determining Initial Coordinates



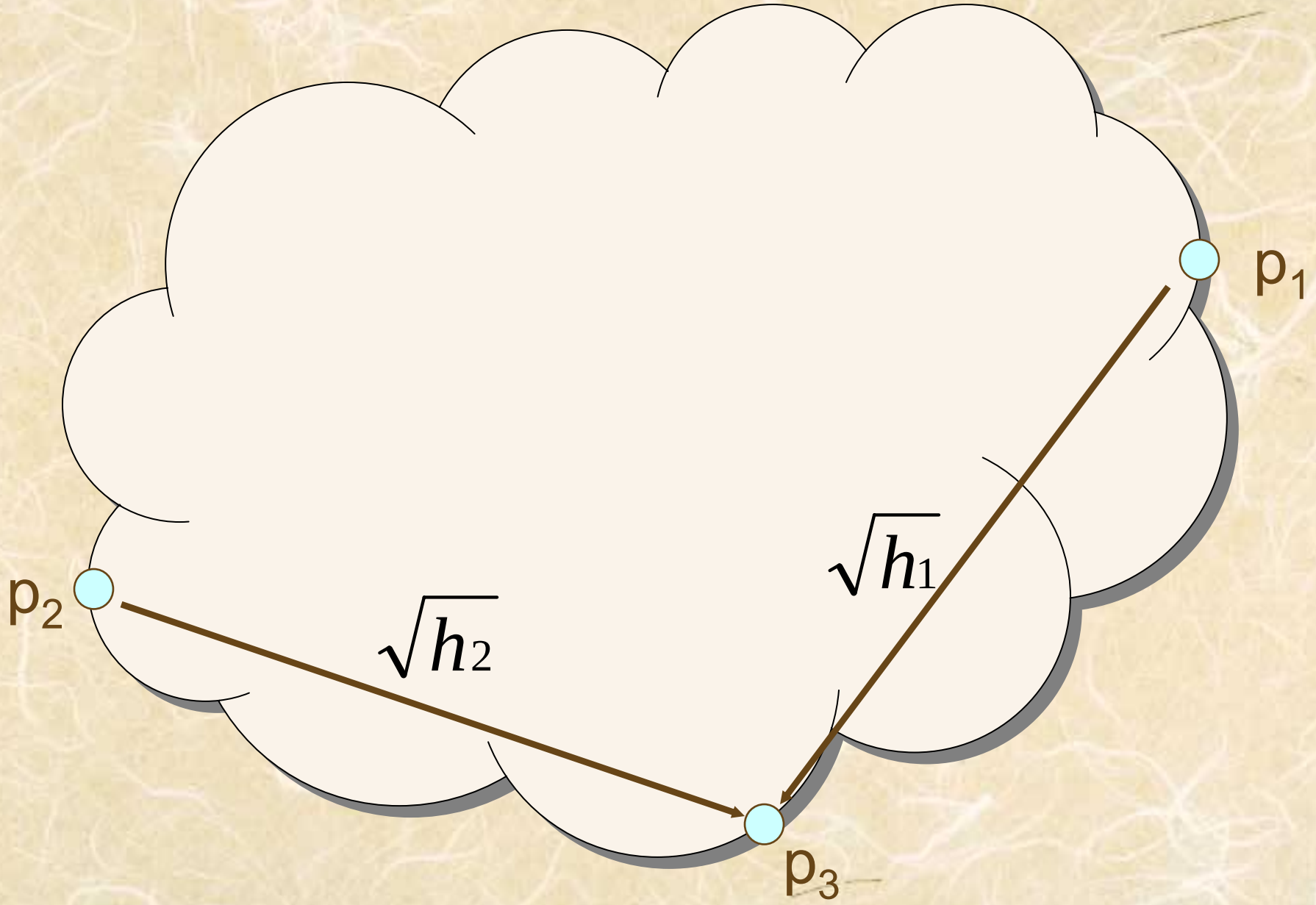
Determining Initial Coordinates



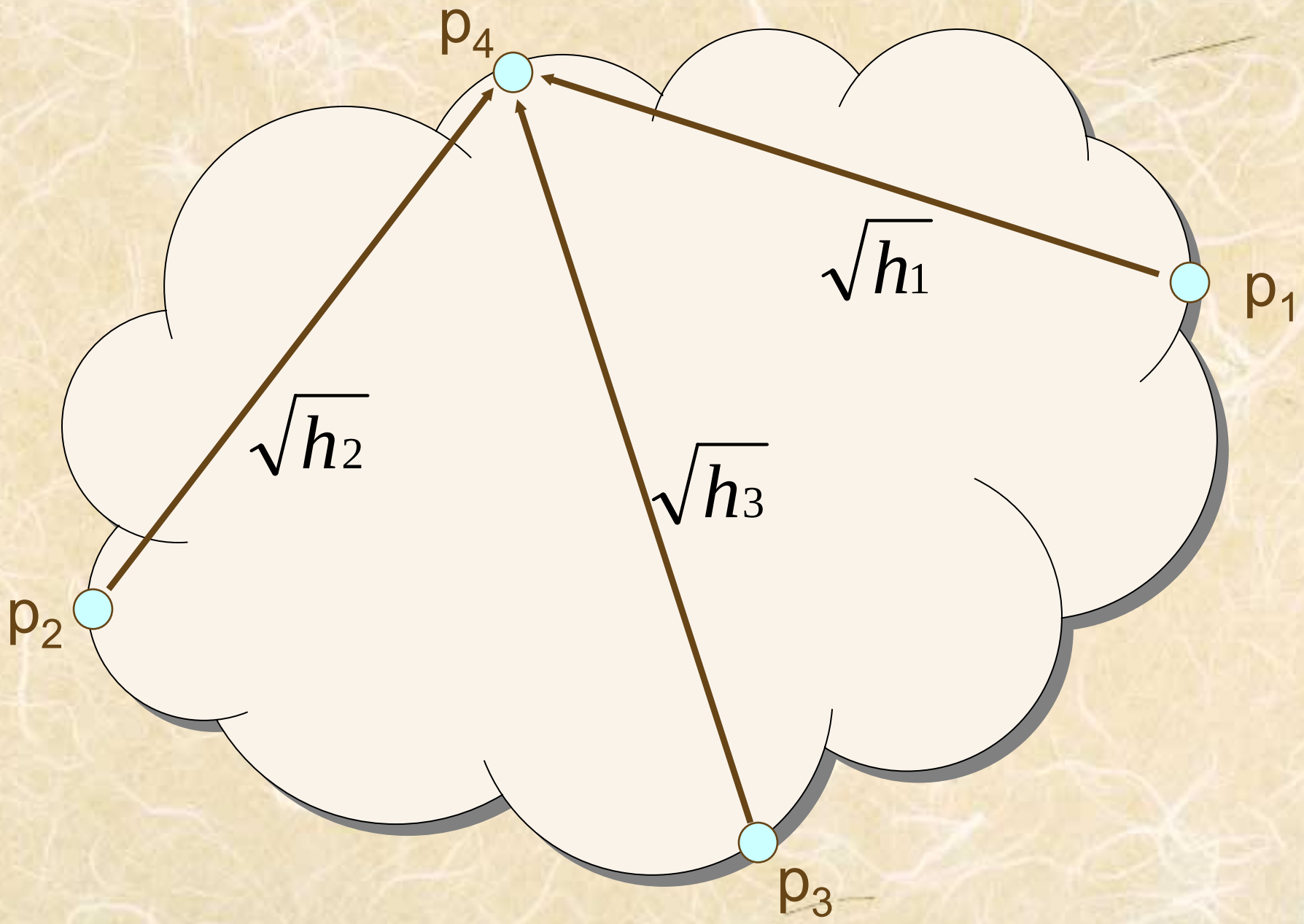
Determining Initial Coordinates



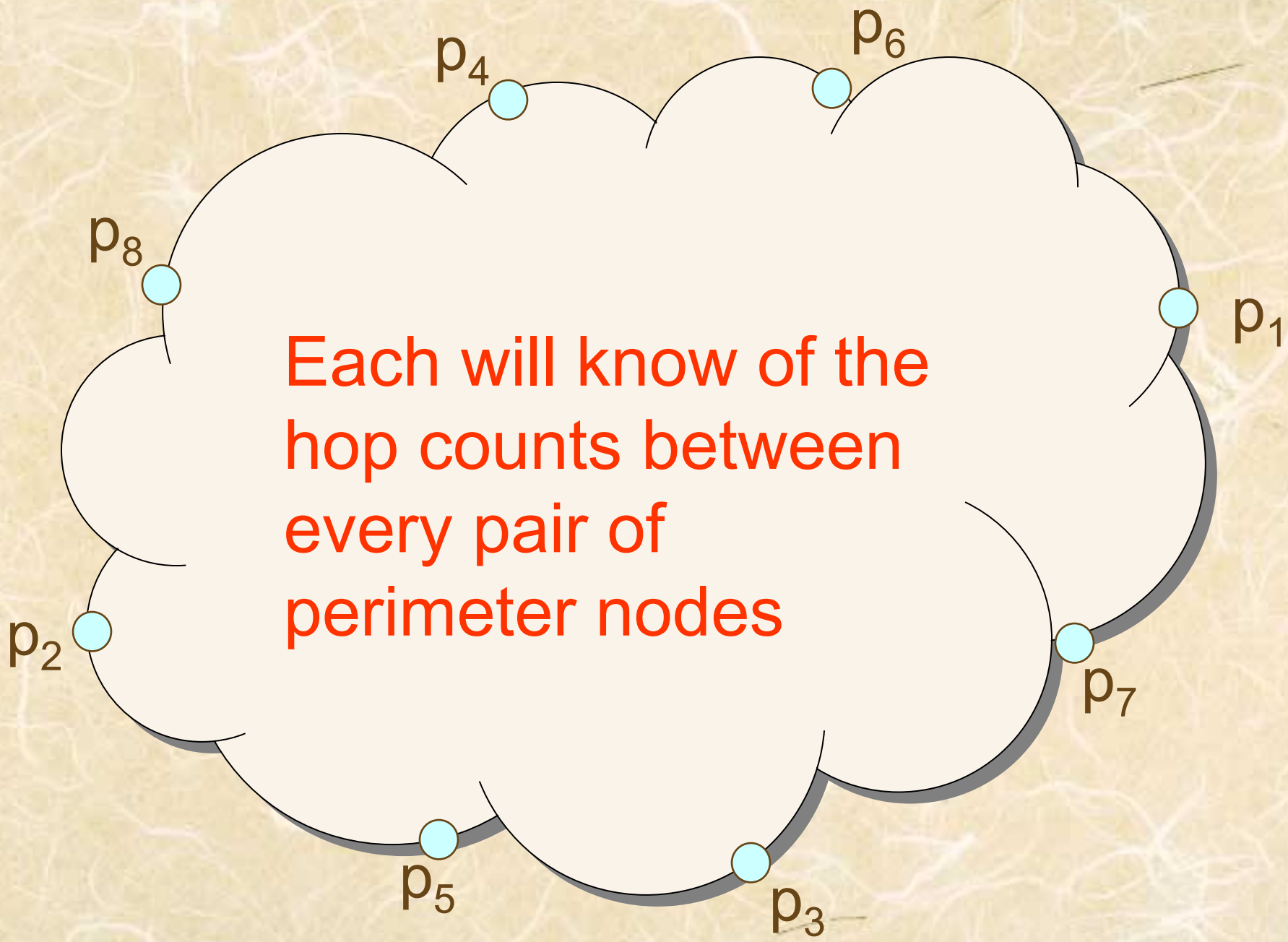
Determining Initial Coordinates



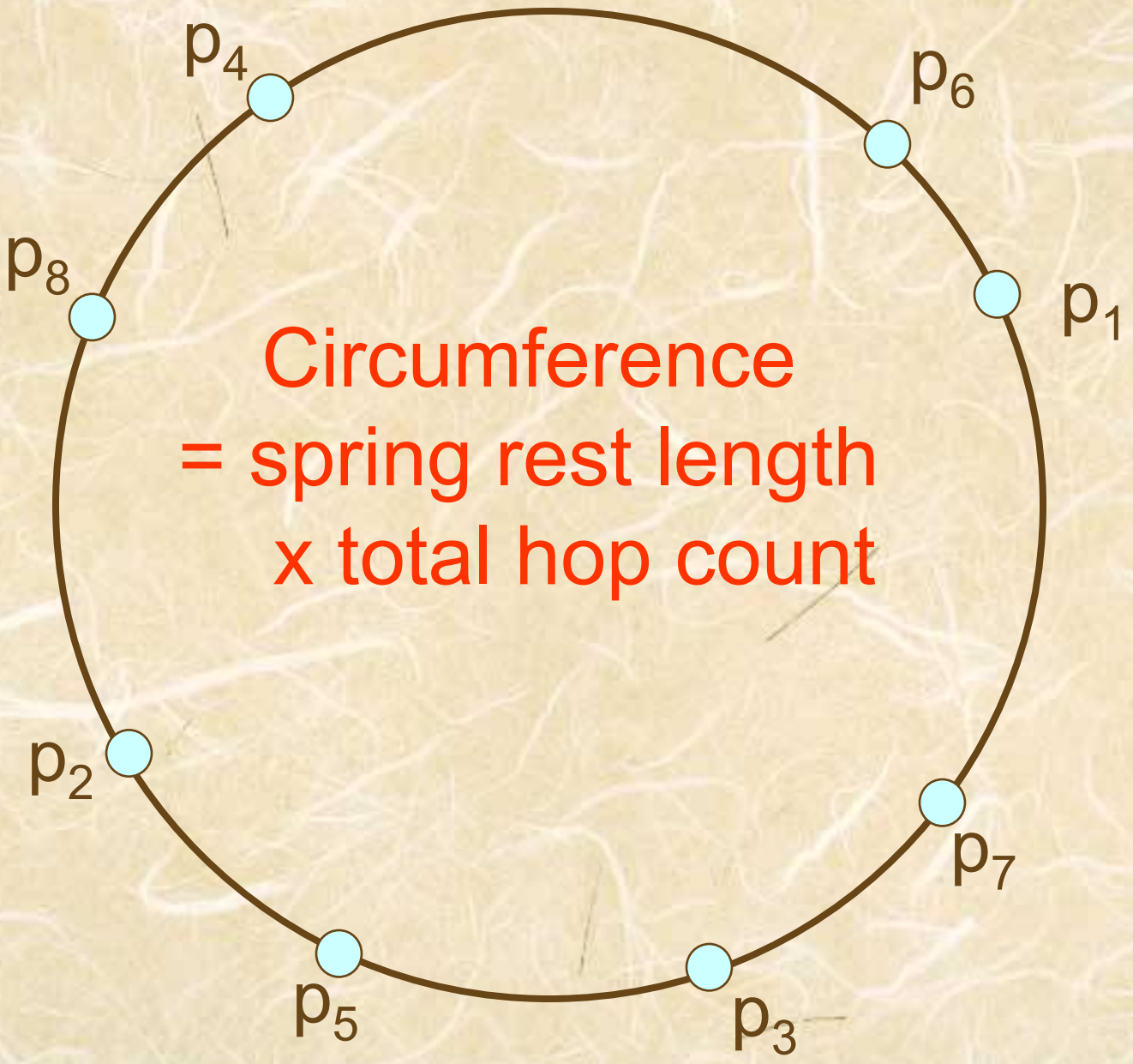
Determining Initial Coordinates



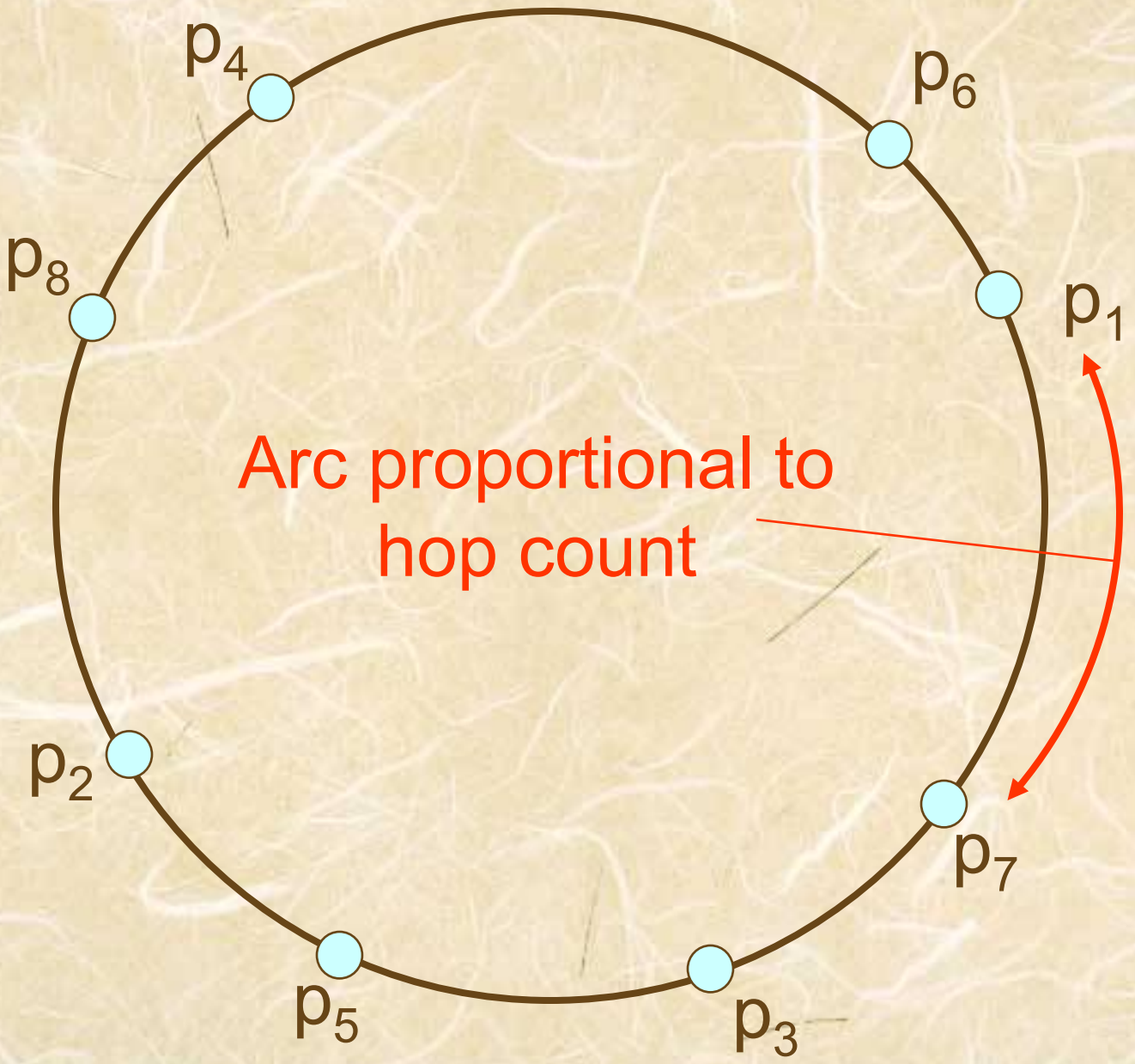
Determining Initial Coordinates



Projection onto Circle



Projection onto Circle



Determining Initial Coordinates

- After perimeter nodes determined
  - matrix of hop counts between them
- Determine cyclical ordering of nodes
- Project nodes onto a circle
- Interpolate for the nodes in between
- Some nodes can wait

Key idea: stretch network toplogy out in the virtual space like a trampoline!

Spring Relaxation Update Rule

- Spring force:

$$F_{ij} = \kappa \times (l_{ij} - |\, x_i - x_j \,|) \times u(x_i - x_j)$$

(Hooke's Law)

- Net force:

$$F_i = \sum_{j \neq i} F_{ij}$$

- Update rule:

$$x_i = x_i + \frac{\min(|\, F_i \,|, \alpha_t)}{|\, F_i \,|} F_i$$

Greedy Embedding

Graph where given any two distinct nodes *s* and *t*, there is a neighbor of *s* that is closer to *t* than *s*.

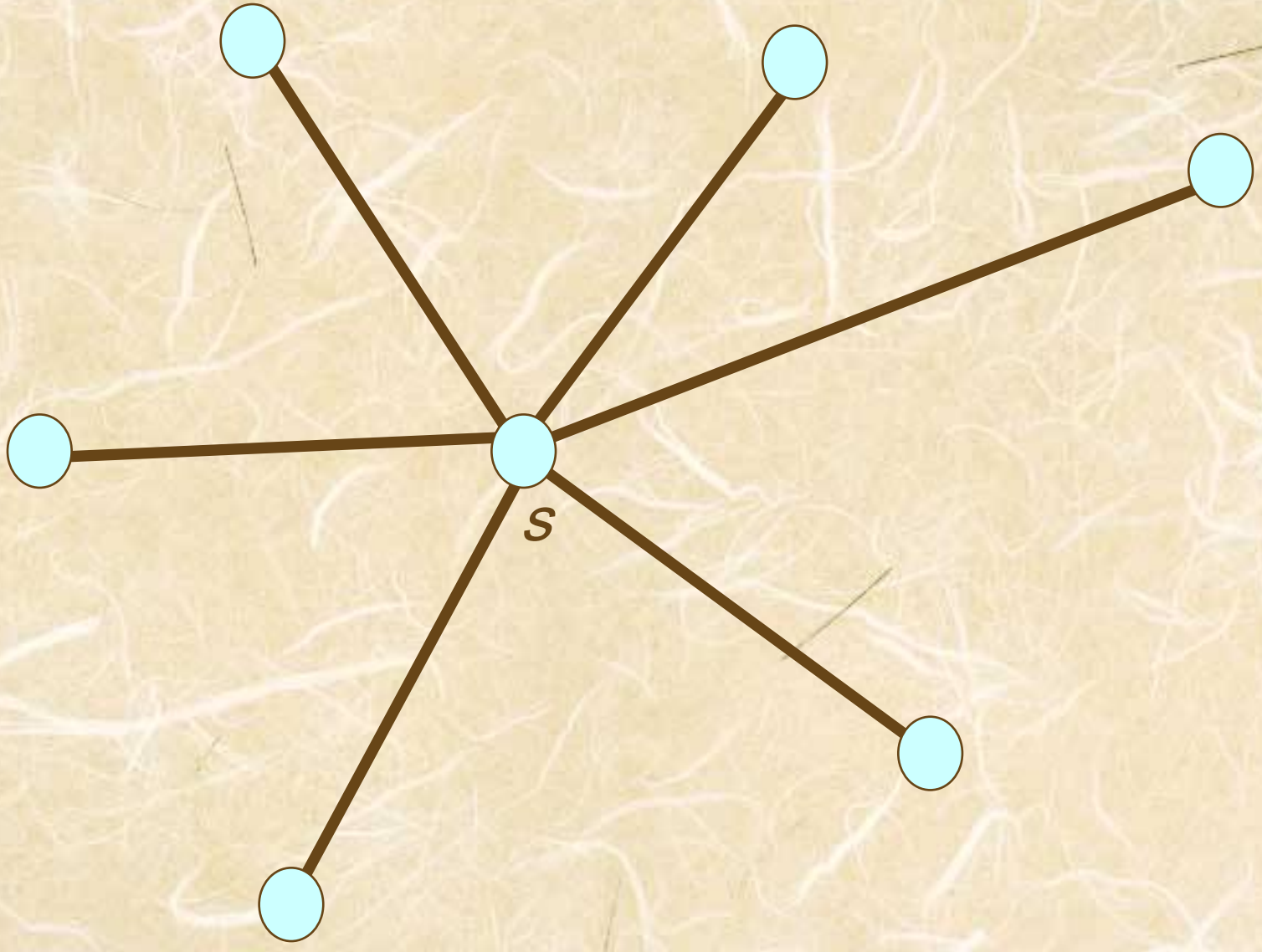
- Greedy forwarding works between any pair of nodes

- Here's a thought:

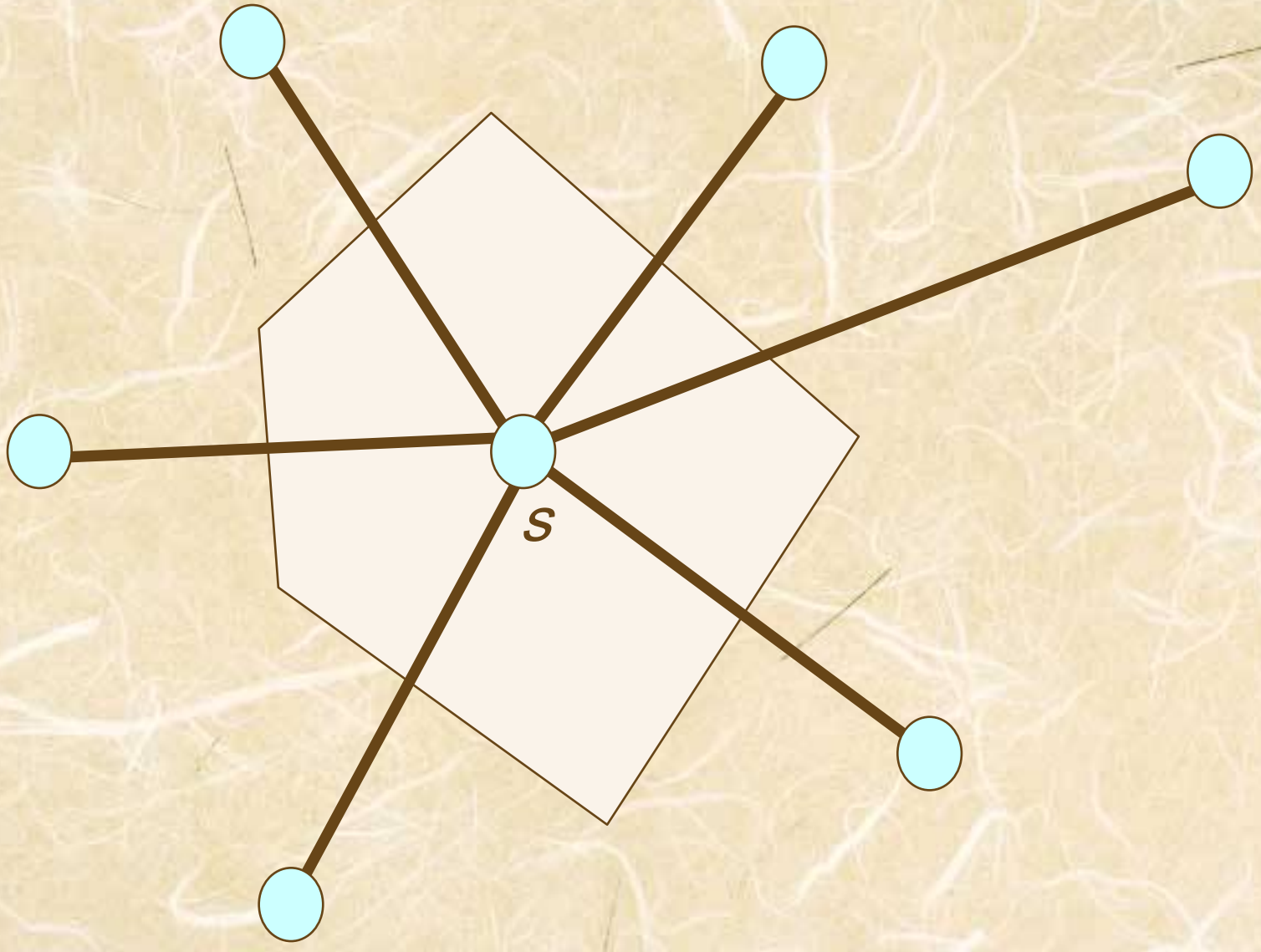
If we pick virtual coordinates such that resulting graph is a greedy embedding, we can achieve good routing performance

HOW? ☺

Region of Ownership



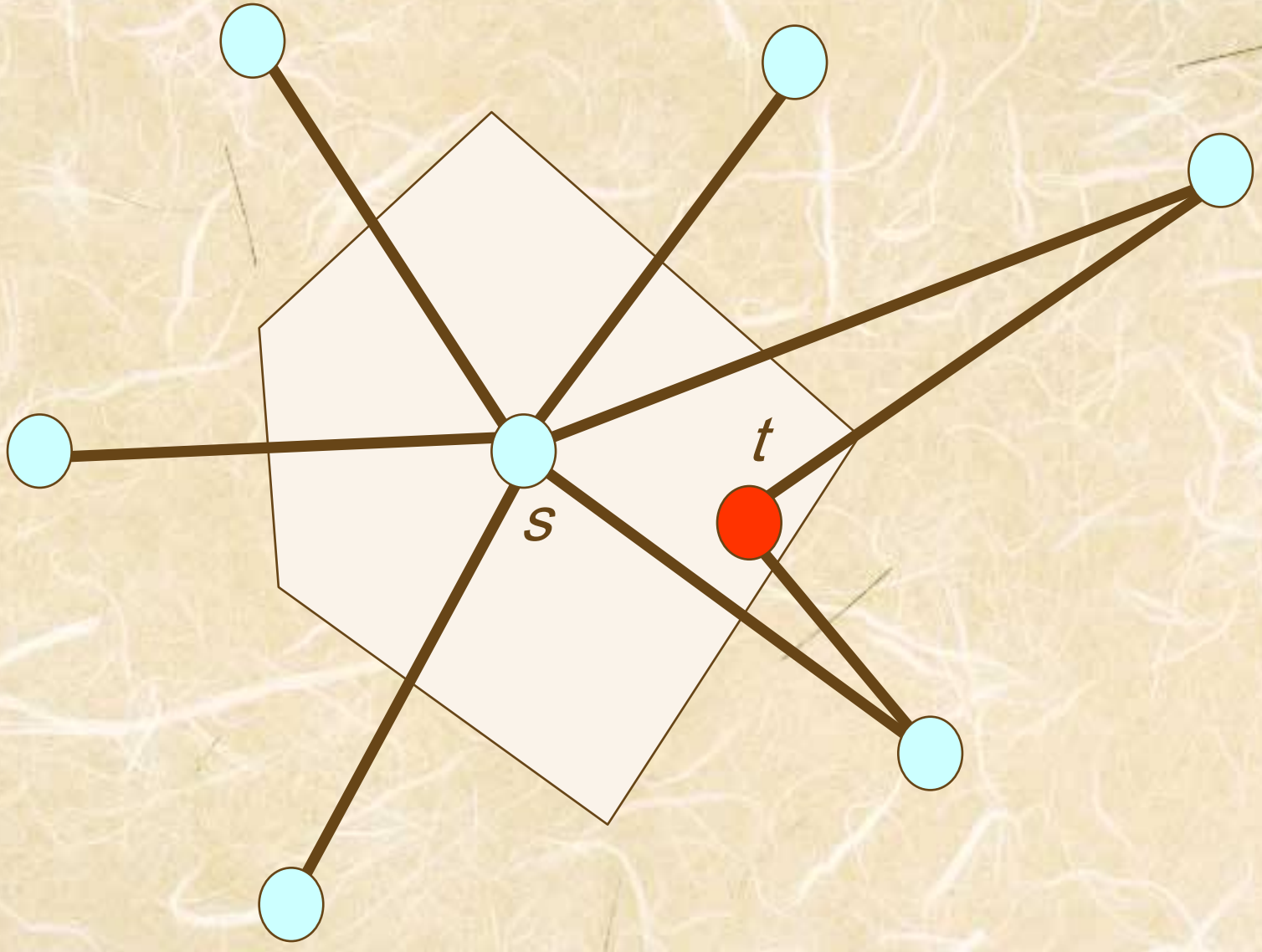
Region of Ownership



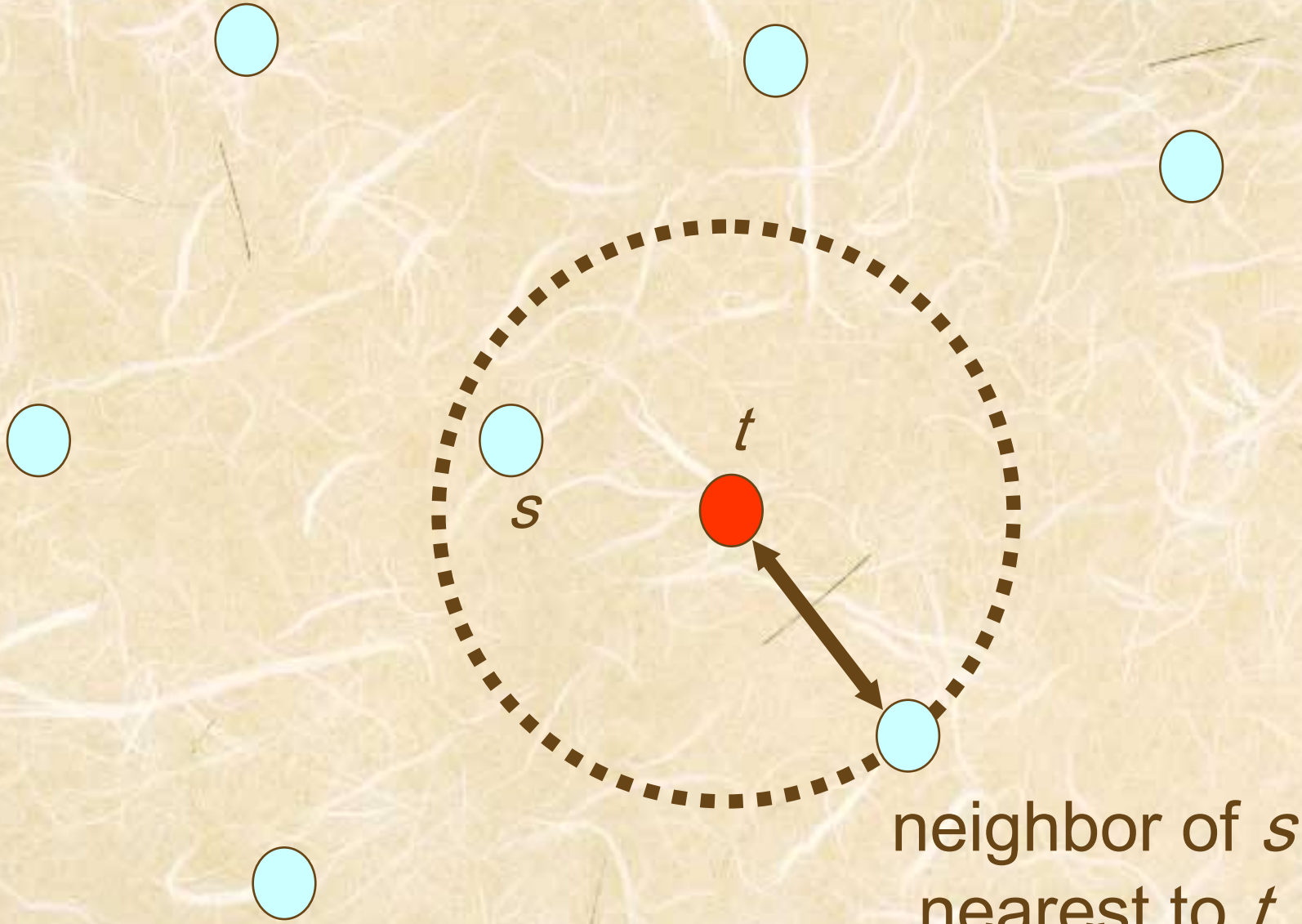
Theorem

An embedding of a Euclidean graph is greedy if and only if the region of ownership of every vertex does not contain any other vertices of the graph.

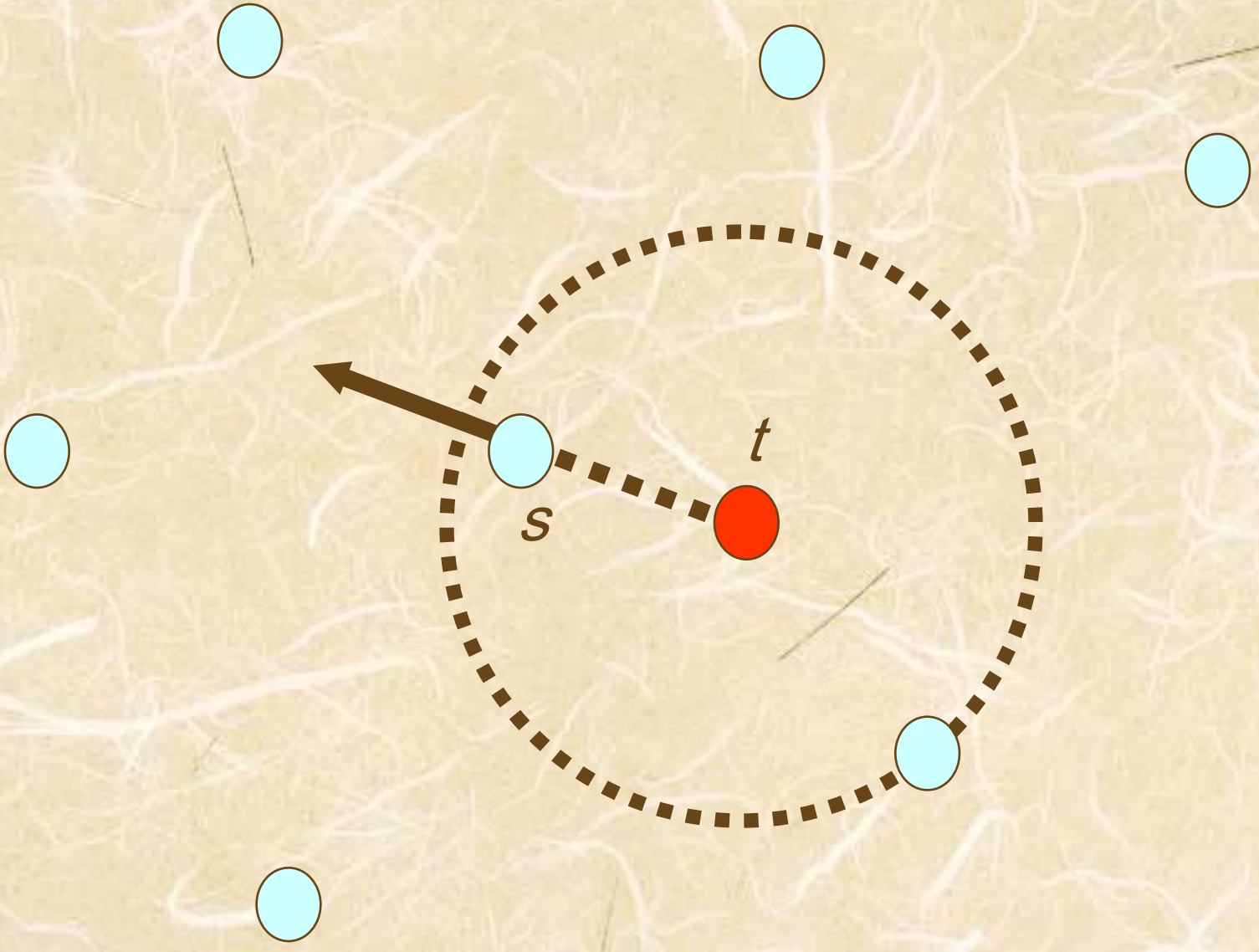
Greedy Embedding Adjustment



Greedy Embedding Adjustment



Greedy Embedding Adjustment



Greedy Embedding Update Rule

- Repulsion force:

$$R_{ik} = \delta \times u(x_i - x_k)$$

- Net force:

Spring forces

Repulsion forces

$$F_i = \sum_{j \neq i} F_{ij} + \frac{\min(|\sum_{k \neq i} R_{ik}|, R_{max})}{|\sum_{k \neq i} R_{ik}|} \sum_{k \neq i} R_{ik}$$

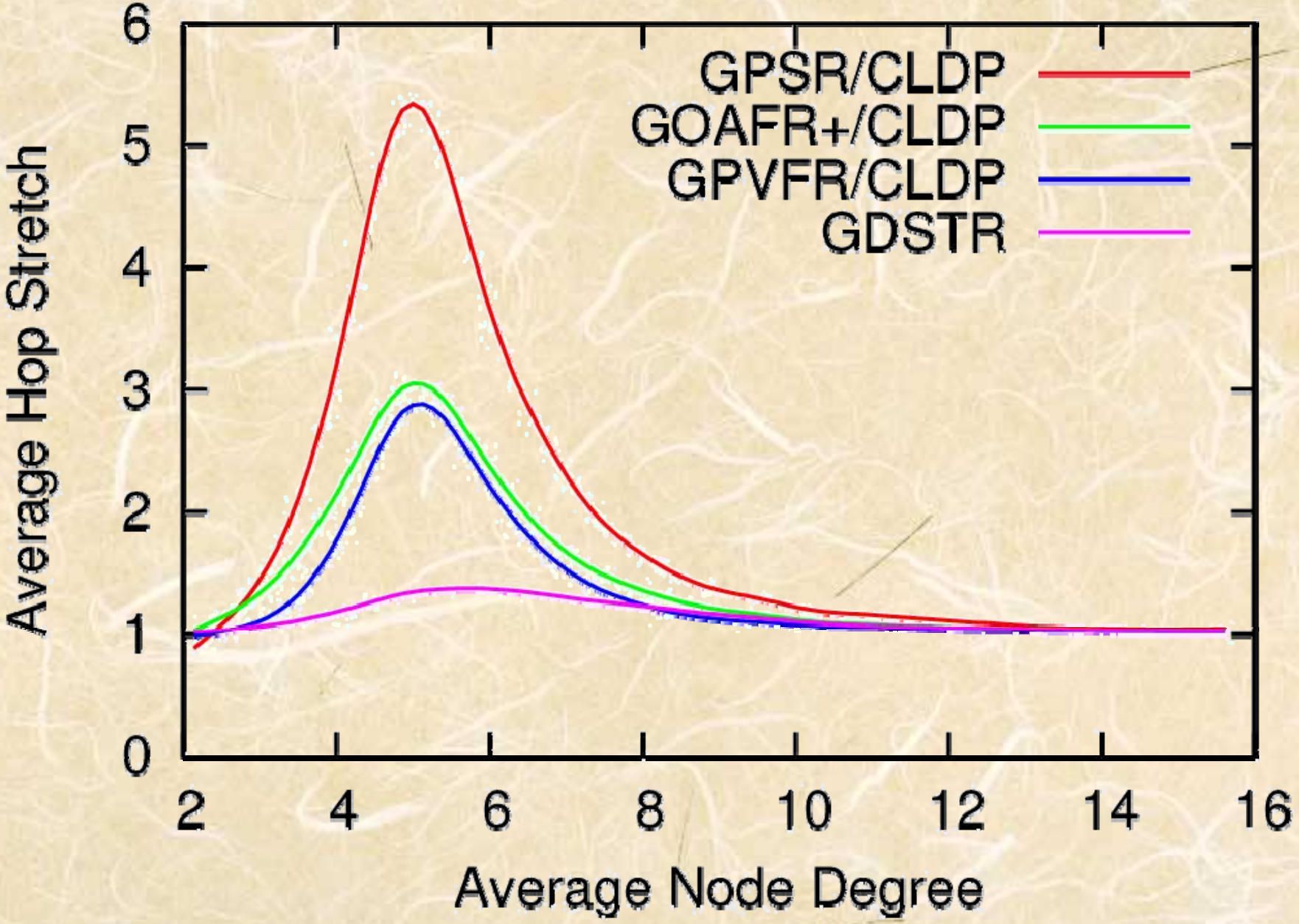
Greedy Embedding Spring Coordinates (GSpring)

- Once a node has stabilized, use geocast to determine nodes in region of ownership
- Use damping and hysteresis to ensure system converges

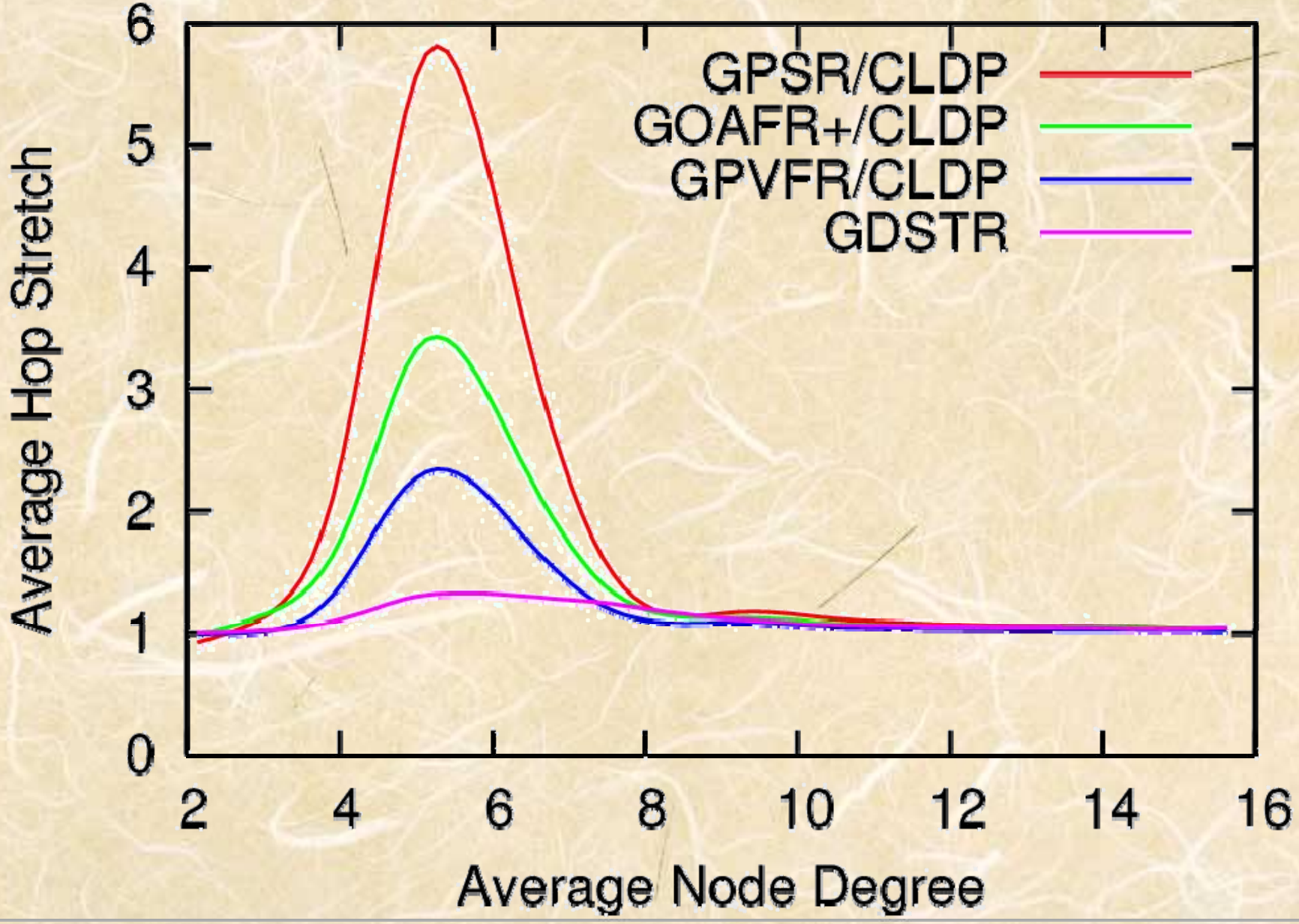
Performance

- Measured Hop Stretch
- Topologies
  - range of network densities (average node degree)
  - larger networks up to 2,000 nodes
    - low/high density
    - obstacles

Actual Physical Coordinates



GSpring Coordinates

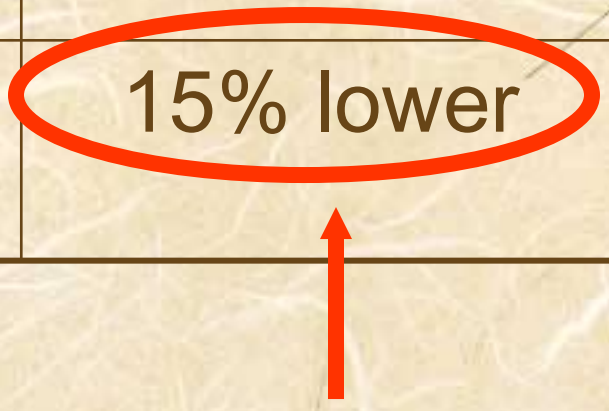


Performance

- Routing algorithm: GDSTR
- Compare with
  - actual coordinates
  - NoGeo (Rao et al., 2003)
- Measured costs:
  - iterations required
  - geocast messages

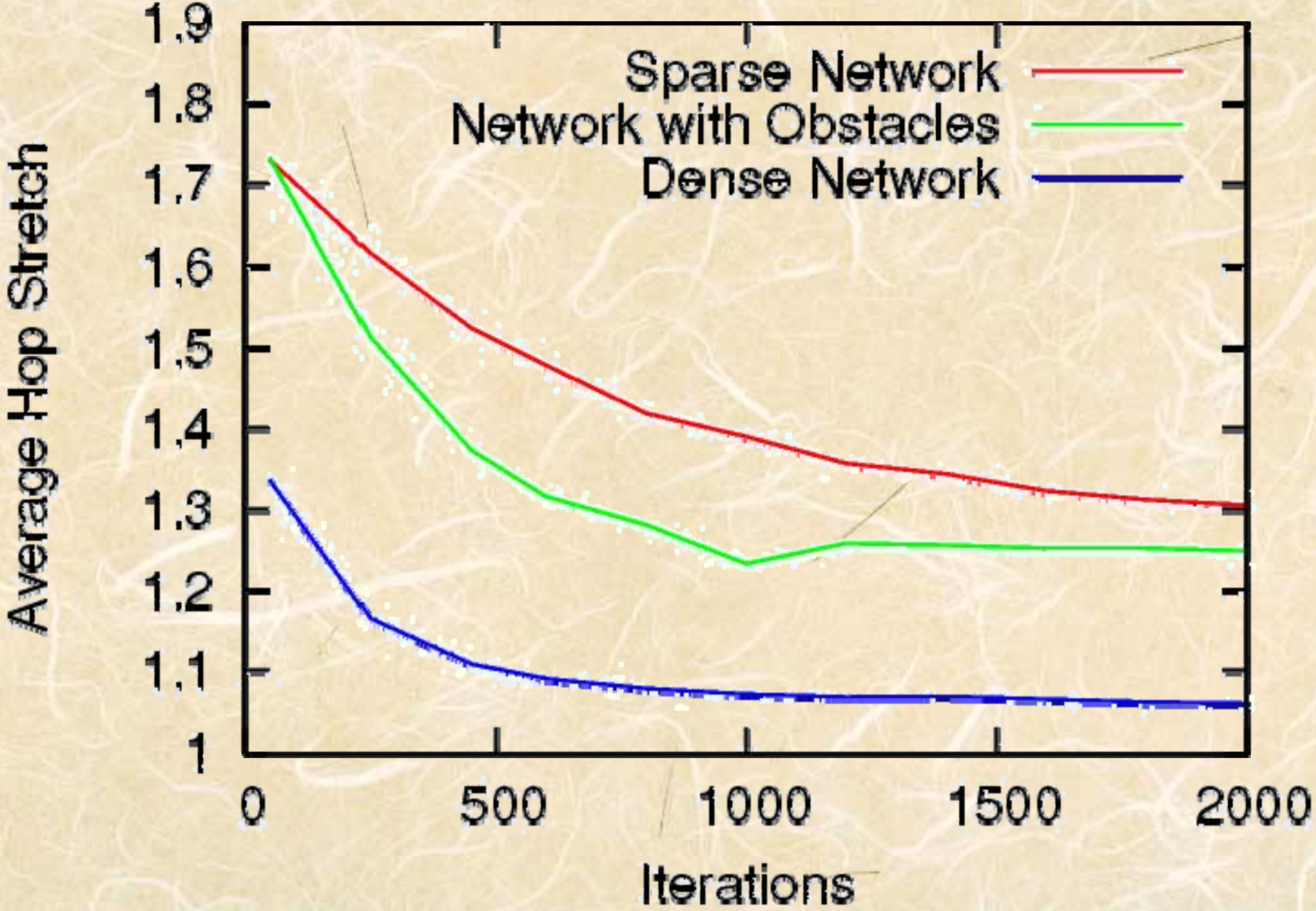
Performance: Hop Stretch

Network Type	Physical coordinates	NoGeo
Sparse UDG	Same	30% lower
Dense UDG	Same	50% lower
Obstacle	15% lower	50% lower

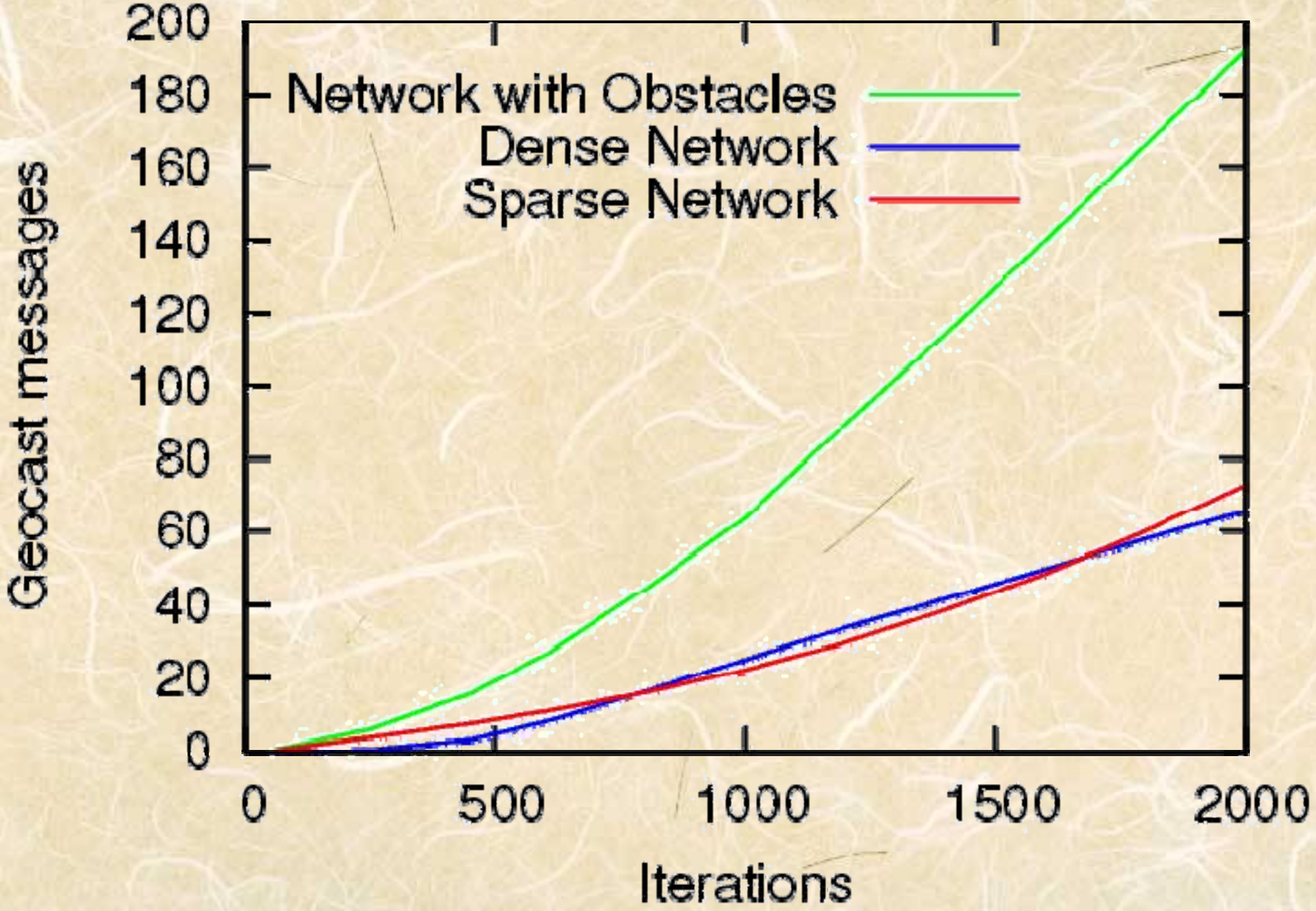


Can do better than actual coordinates!

Performance over Time



Messaging Costs



Summary

- Two key ideas:
  - Initial coordinates: stretch network out like a trampoline
  - To make topology more complex: need to move nodes out of each others “regions of ownership”

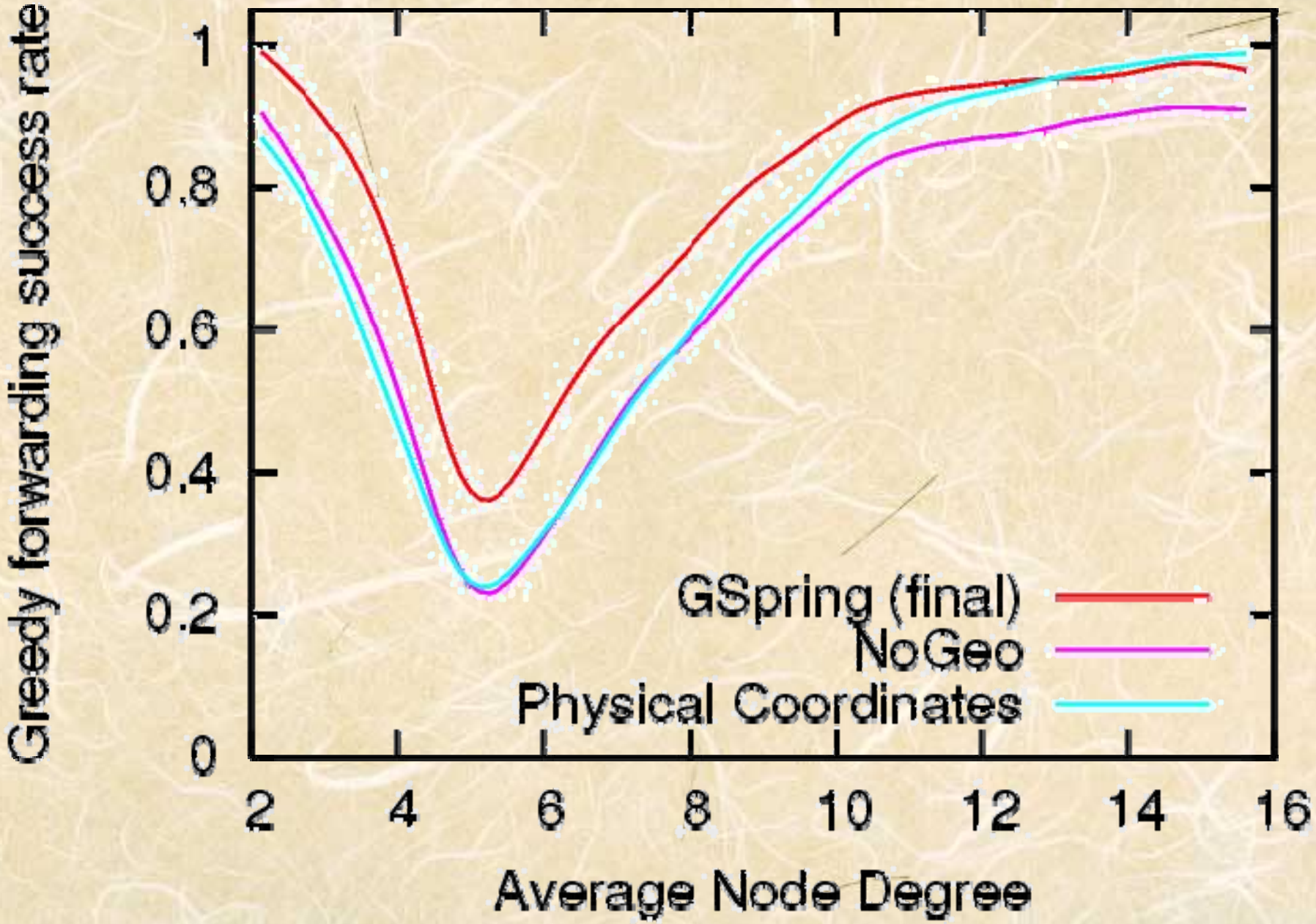
Future Work

- Evaluate GSpring in a real wireless deployment
- Study theoretical properties
- Region of Ownership generalizable to higher dimensions
  - Can we do achieve greedy embeddings more easily?

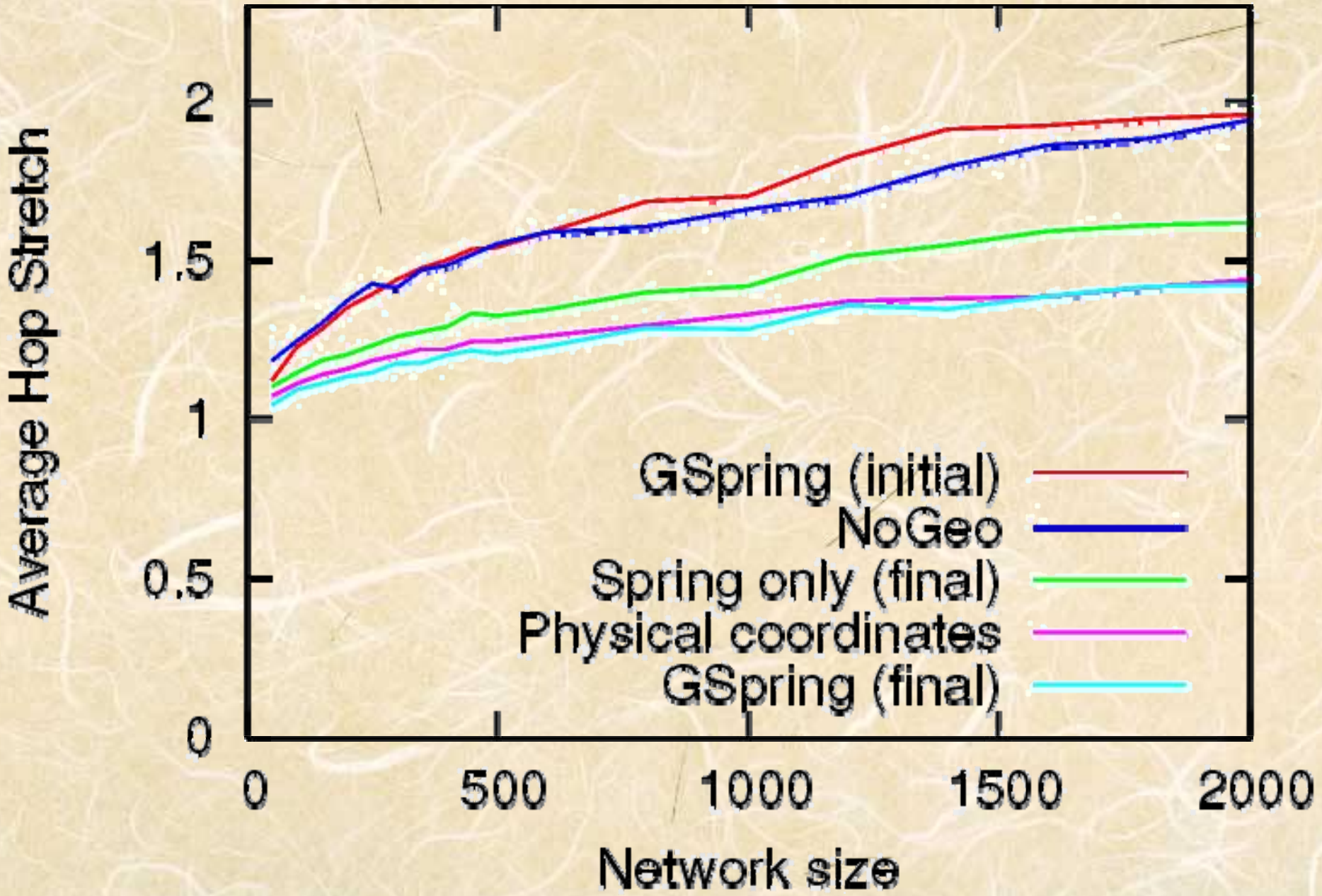
Conclusion

- Hard to find greedy embedding with local, distributed algorithm
  - more greedy improves routing performance
- Good for networks with obstacles
  - converts concave voids into convex ones
- “Embedding routing table into coordinate system”

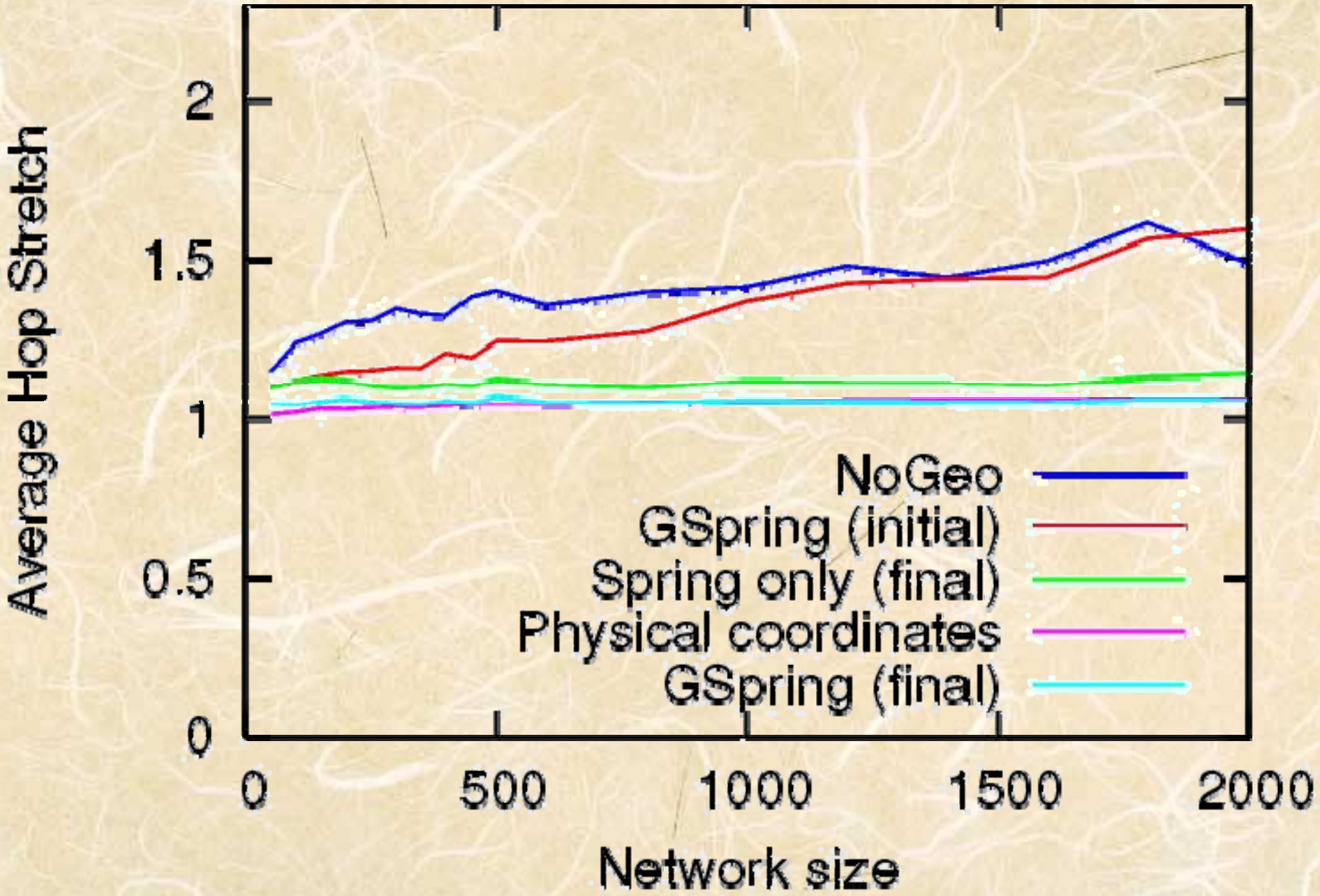
Greedy Forwarding Success



Sparse UDG Networks



Dense UDG Networks



Networks with Obstacles