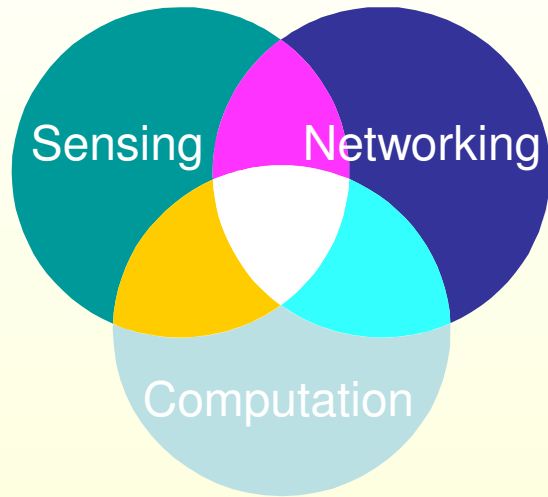
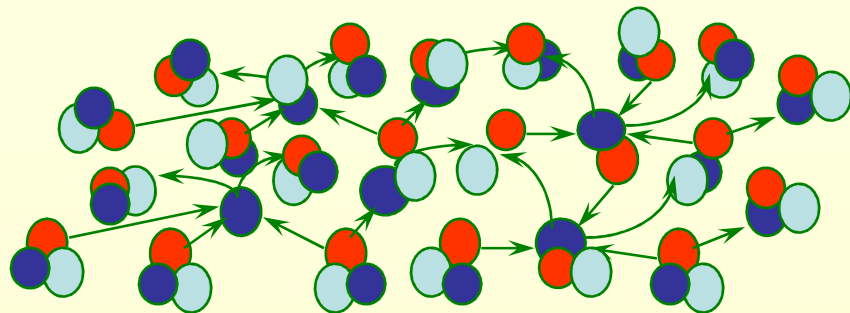


CS321: Networking Sensors II: More on Geographic Routing



Leonidas Guibas
Computer Science Dept.
Stanford University



Geographic Routing in Practice

[Y. -J. Kim, R. Govindan, B. Karp, S. Shenker.
On the Pitfalls of Geographic Routing,
Proc. of the 3rd International Workshop on DIALM,
Principles of Mobile Computing, September 2005]

Certain Questionable Assumptions

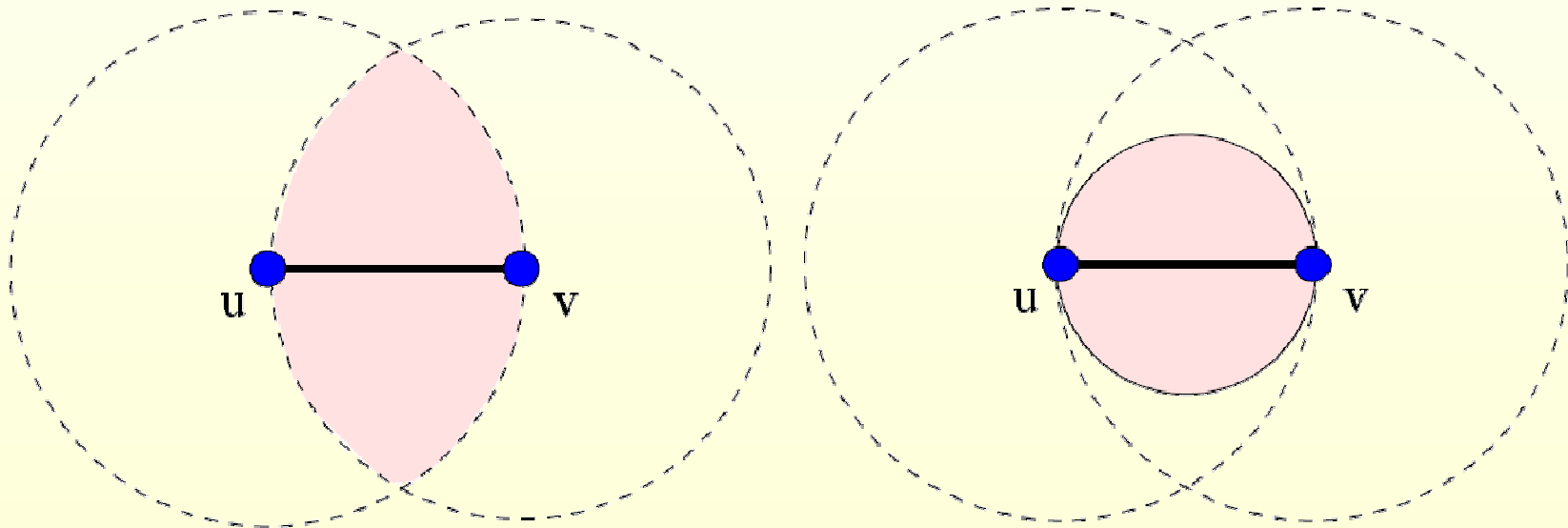
- Nodes know their locations **accurately**.
- The network topology follows the **unit disk graph** (UDG) model.
- These are two **BIG** assumptions.
- Localization is hard, both in theory and in practice.
- Unit disk graph model is simply not valid in practice.

Three Pathologies in the Planarization Process

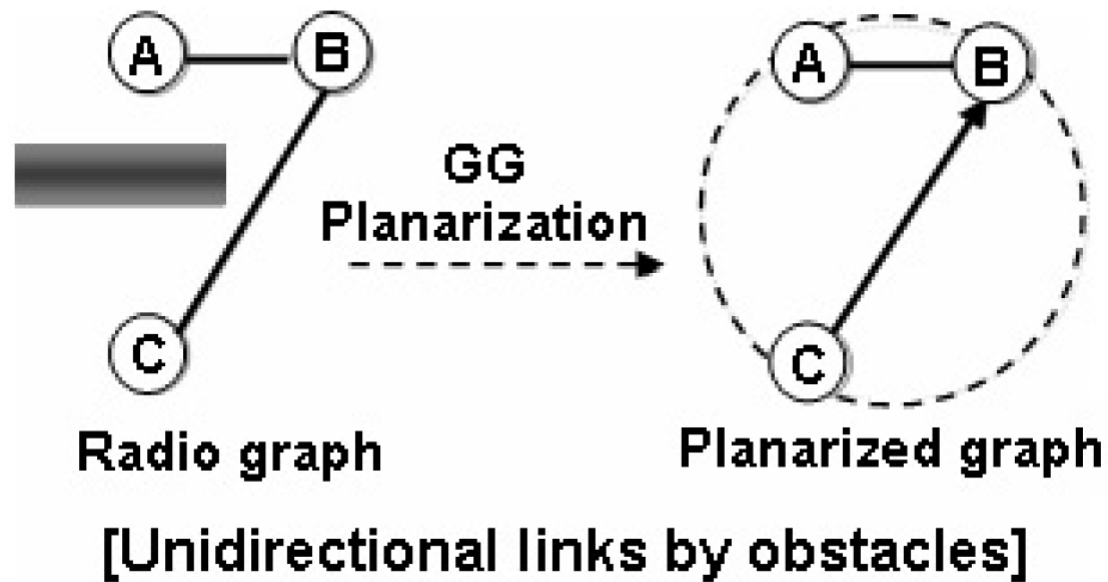
- Too many links may be removed, disconnecting the graph
 - Unidirectional links may be present
 - Crossing links may be present
- These pathologies can result in significant routing failures

Relative Neighborhood Graph and Gabriel Graph

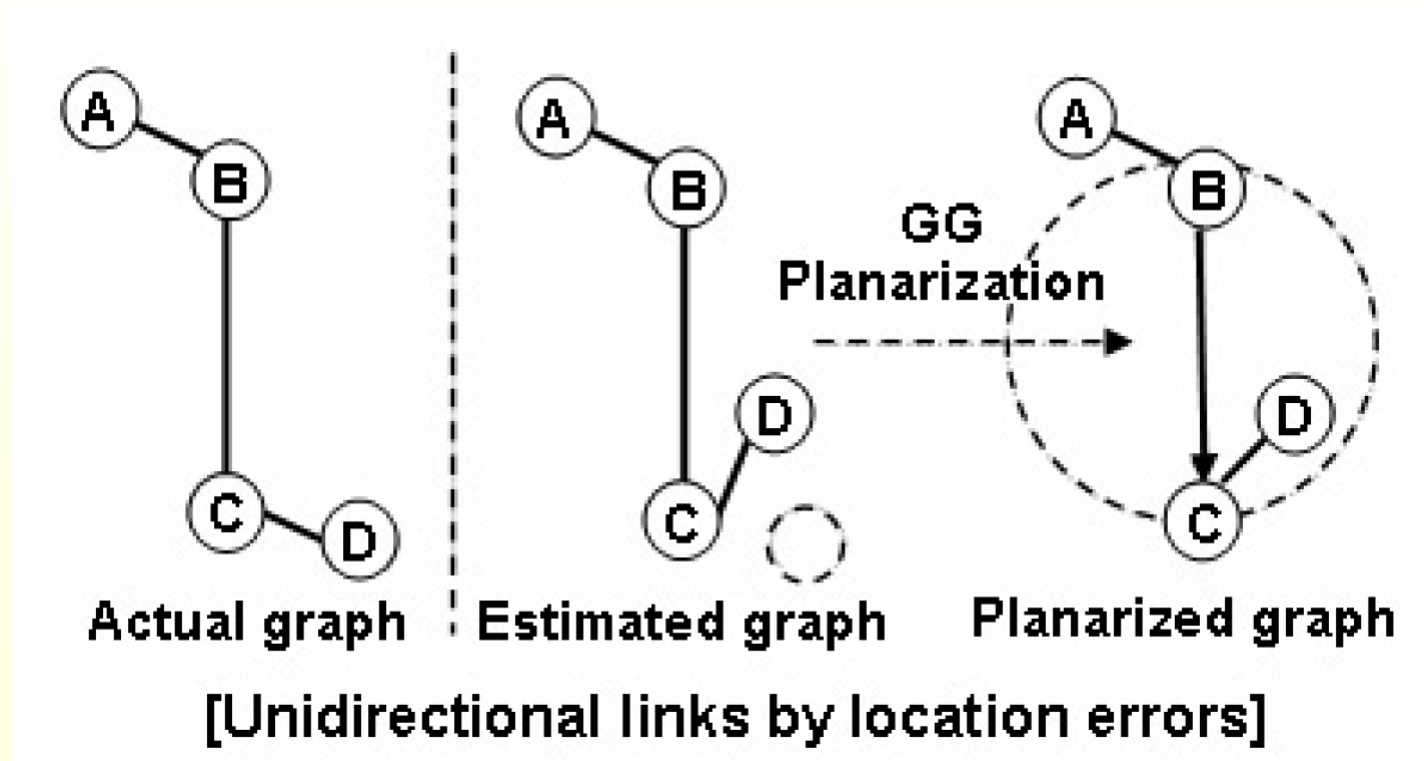
- Relative Neighborhood Graph (RNG) contains an edge uv if the lune is empty of other points.
- Gabriel Graph (GG) contains an edge uv if the disk with uv as diameter is empty of other points.
- Both can be constructed in a distributed way.



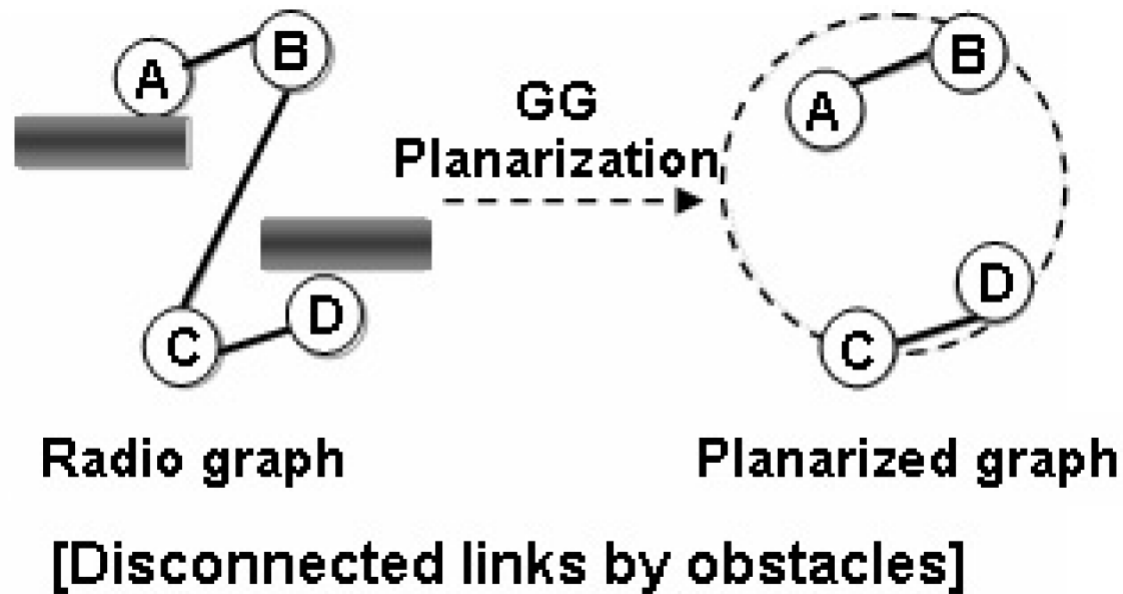
Planarization Problems: GG



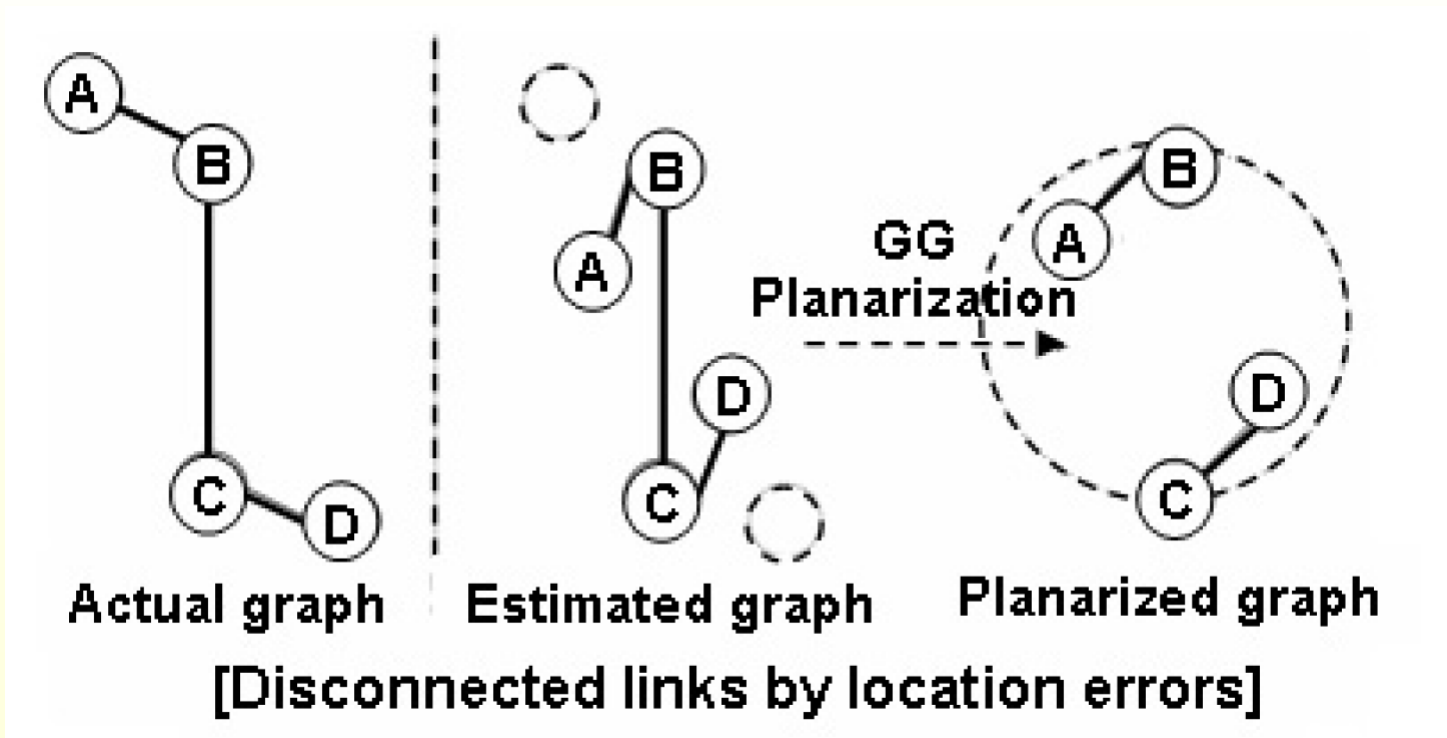
Planarization Problems: GG



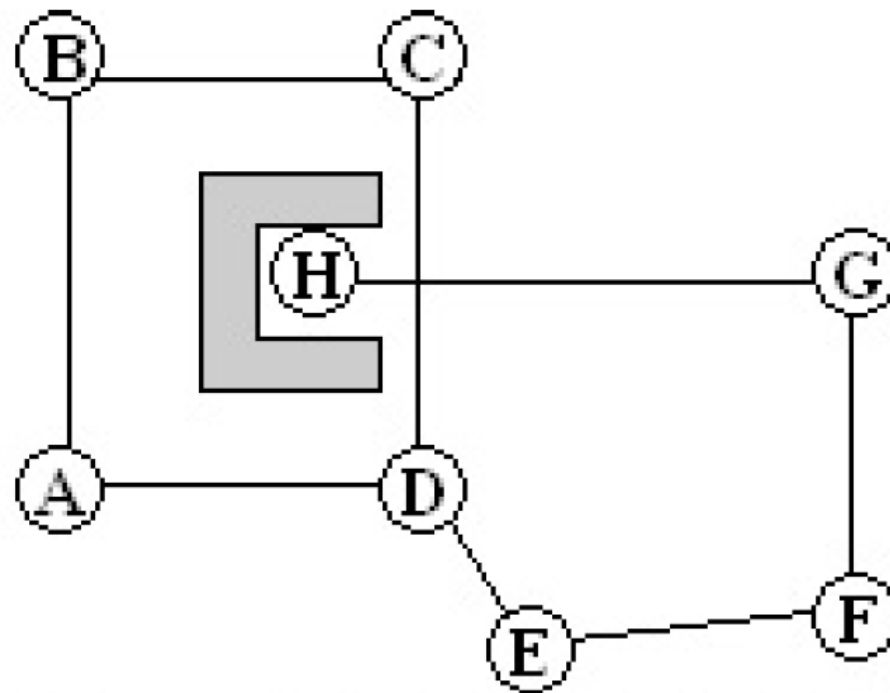
Planarization Problems: GG



Planarization Problems: GG

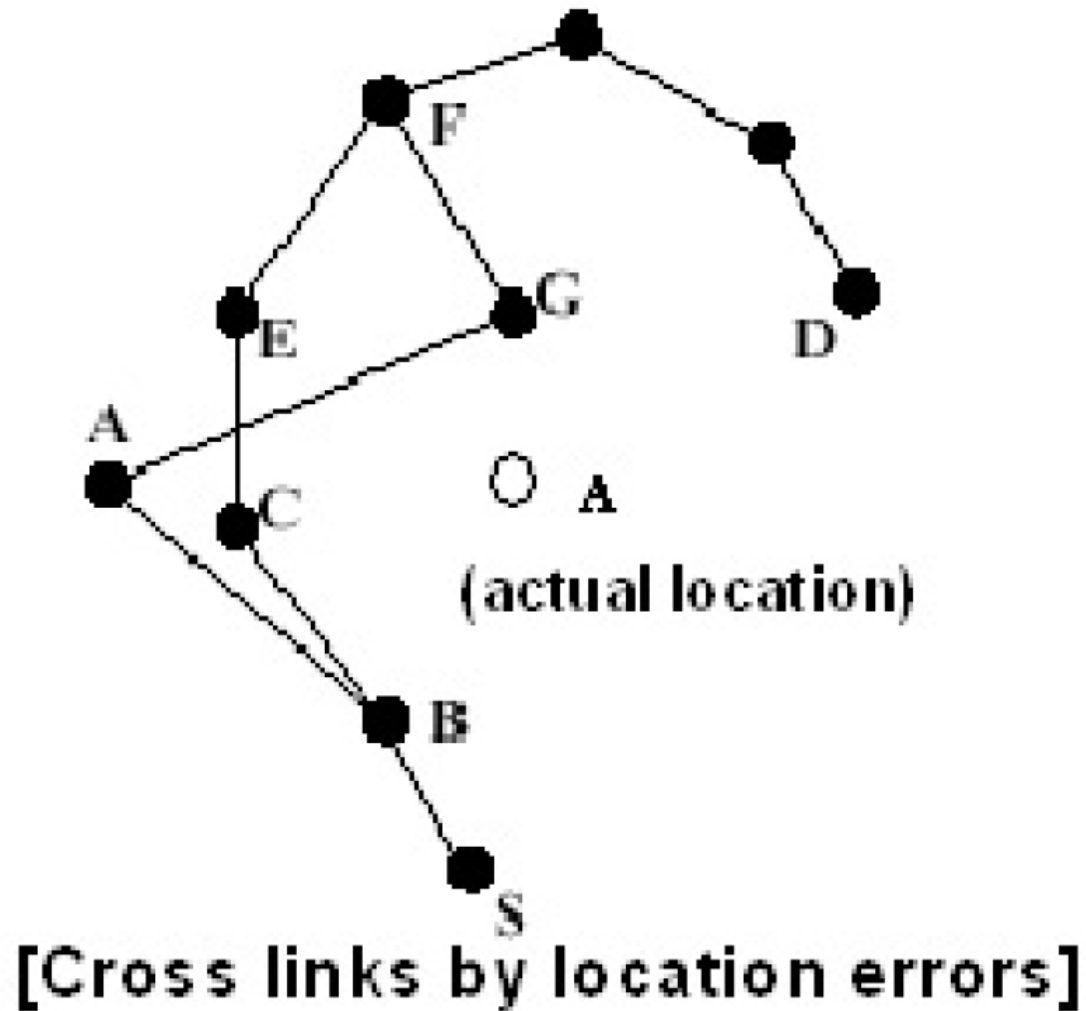


Planarization Problems: GG



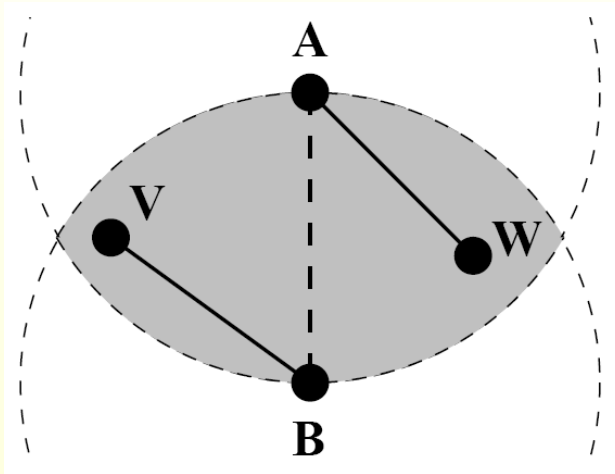
[Cross links by obstacles]

Planarization Problems: GG

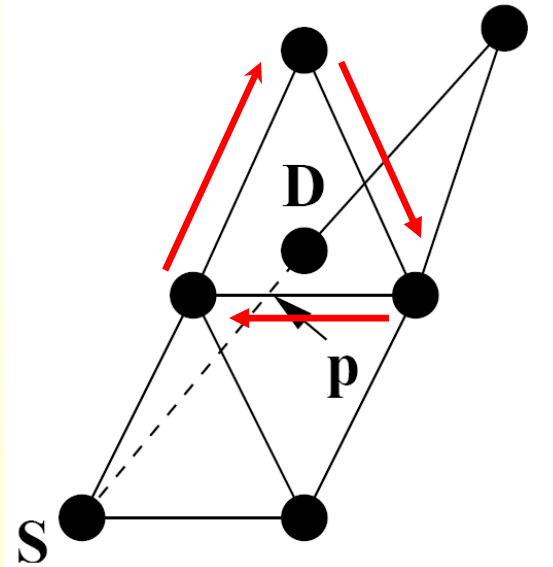


Routing Difficulties w. Irregular Radio Ranges

- Network is partitioned.
- Crossing links.



Edge AB is removed.



No crossing of line SD closer than point p.

Resulting Routing Failures

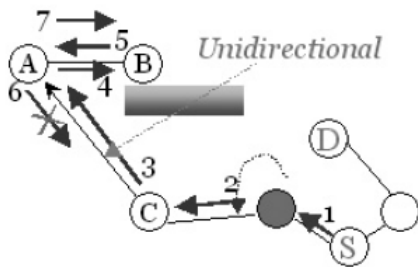


Figure 7: Routing failure caused by unidirectional links.

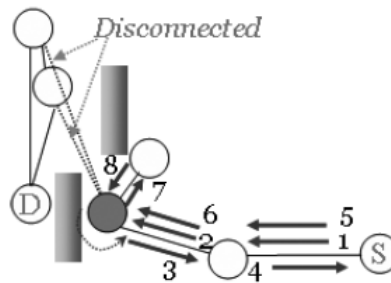


Figure 8: Routing failure caused by disconnected links.

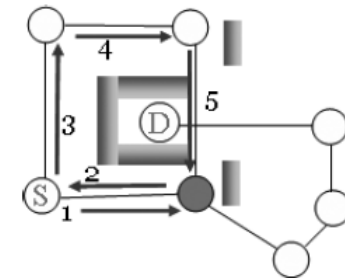
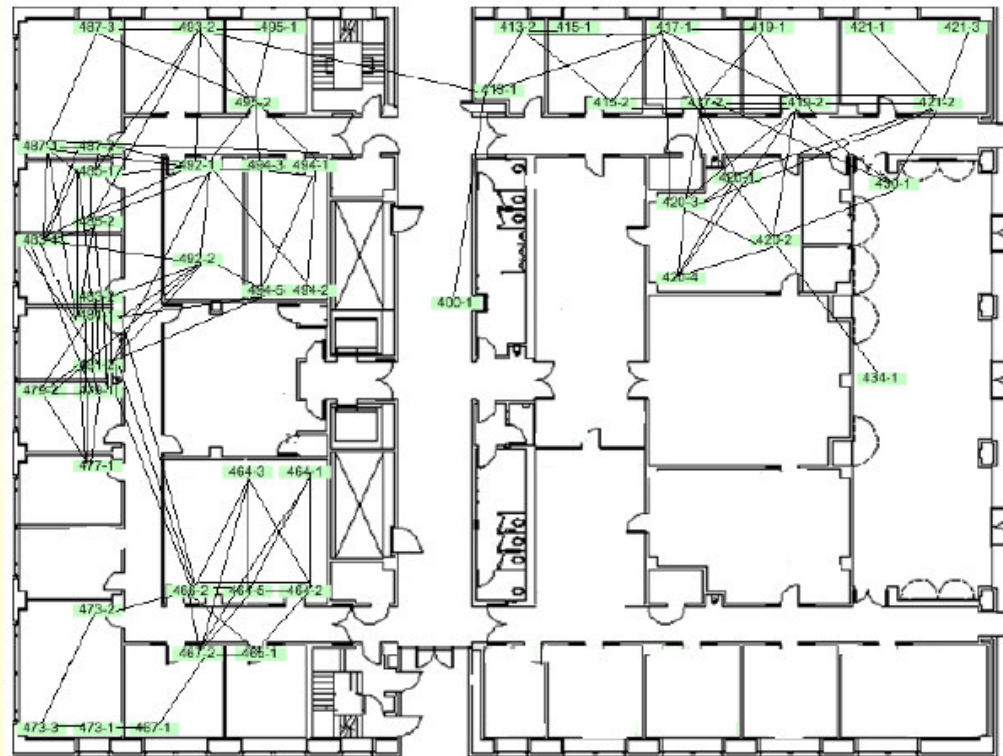


Figure 9: Routing failure caused by cross links.

Testing GPSR on a Real Testbed

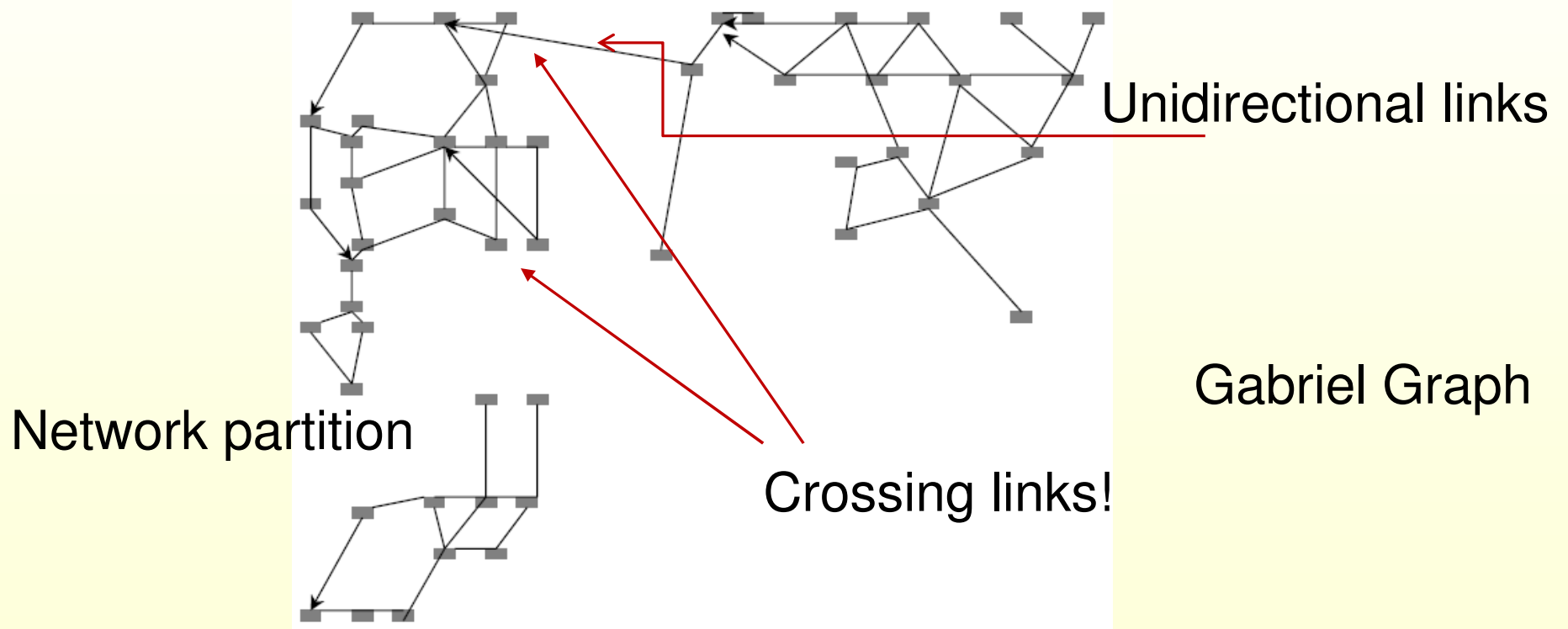
- GPSR only succeeds on only 68.2% directed node pairs.



A 50-node testbed at Soda Hall (UC Berkeley)

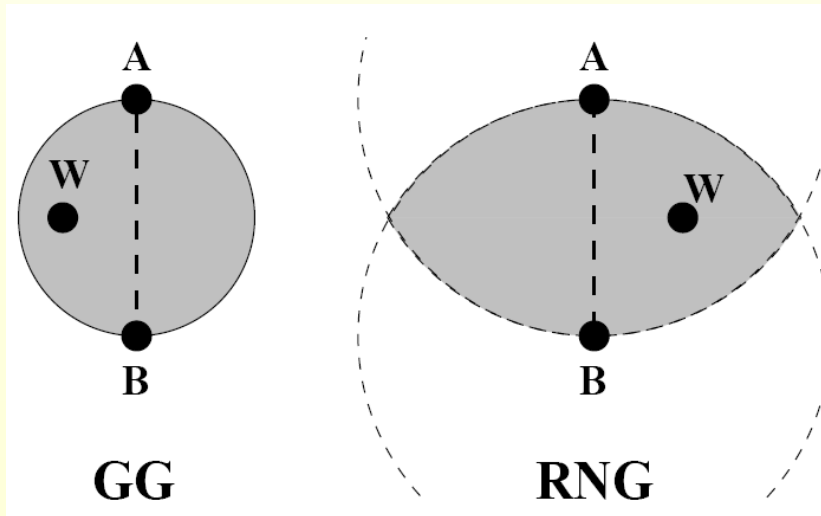
Planarization Pathologies

- All pathologies happen!



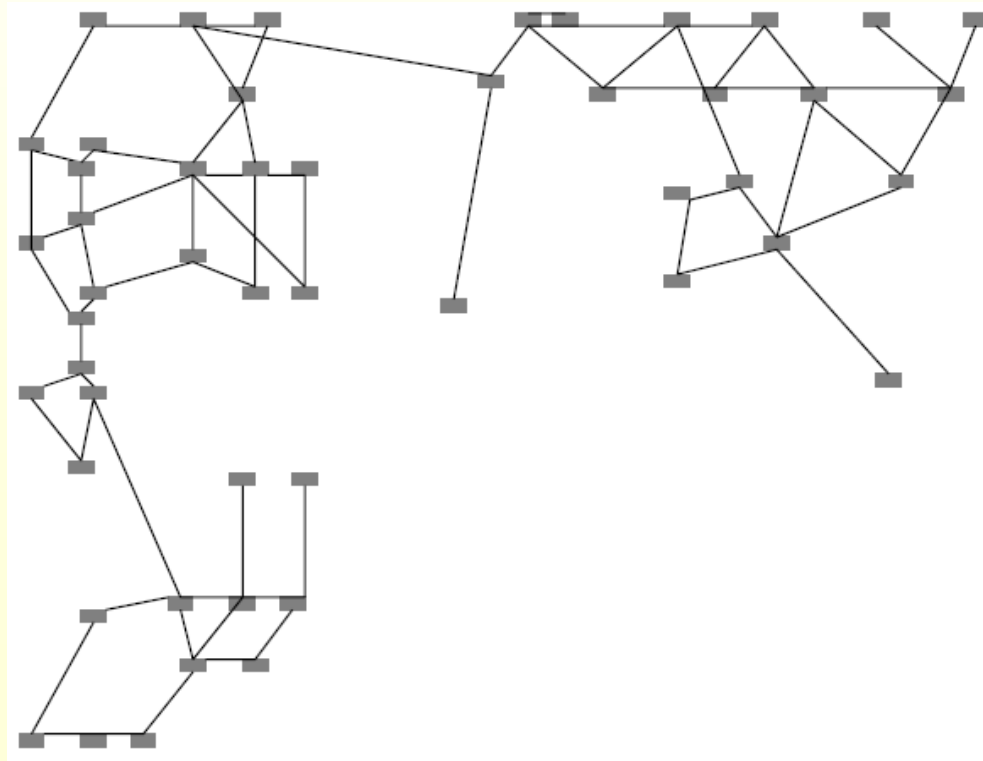
A Small Fix for the Asymmetric Links

- The irregular radio range fails the planar graph construction.
- A small fix by using the **mutual witness test**:
- The link AB is removed only if there is witness that is seen by **both** A and B.



A Small Fix for the Asymmetric Links

- Removed disconnection, but
- Still leaves crossing links.
- Only improves the success rate of GPSR to 87.8%.



A Fix of a Different Nature ...

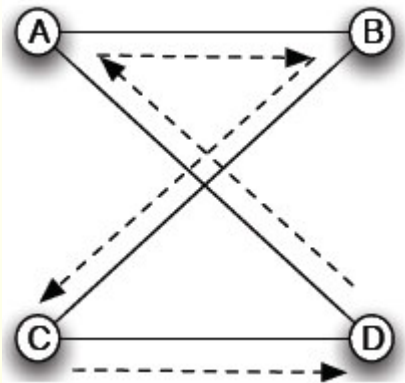
[Y.-J. Kim, R. Govindan, B. Karp, S. Shenker.
Geographic Routing Made Practical,
Proc. USENIX Symposium on
Networked Systems Design and Implementation
(NSDI) , pp. 217-230, May 2005]

Cross Link Detection Protocol (CLDP)

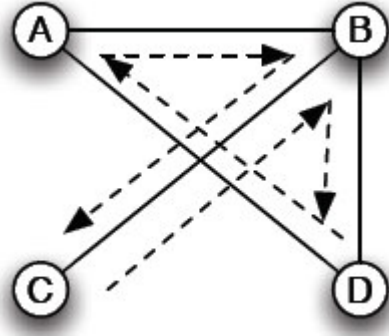
- Try to do face routing on a non-planar network.
- Eliminate non-OK crossings and keep the graph connected.
- Each node **probes** each of its links to see if it's crossed by other links. Links classified as **routable**, or **non-routable**.
- How to probe? Record the link to be probed in packet, do face routing (RH rule) and mark all crossings.

Cross Link Detection Protocol

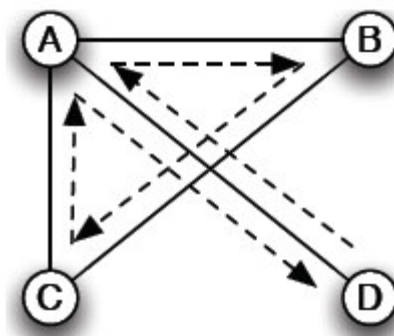
- Start from D and do face routing.



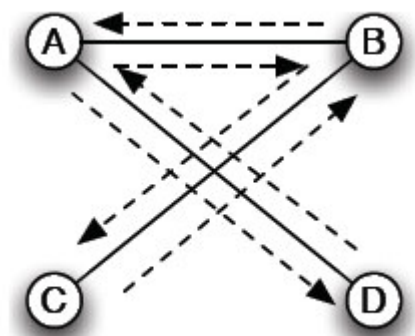
Remove either
AD or BC



Can't
remove BC



Can't
remove AD

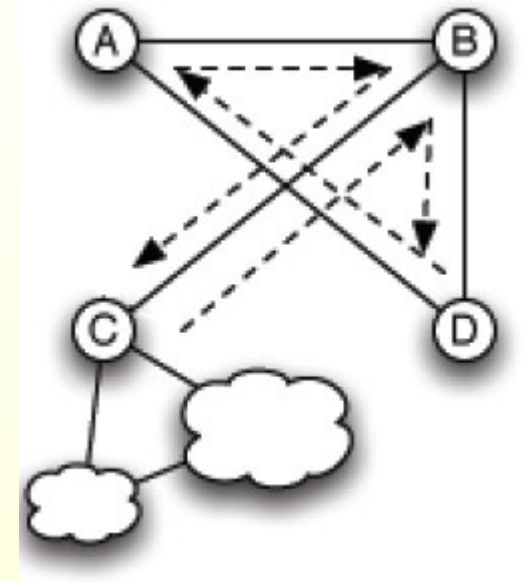


Can't remove
either

Observation: a not-OK link is traversed twice, once in each direction.

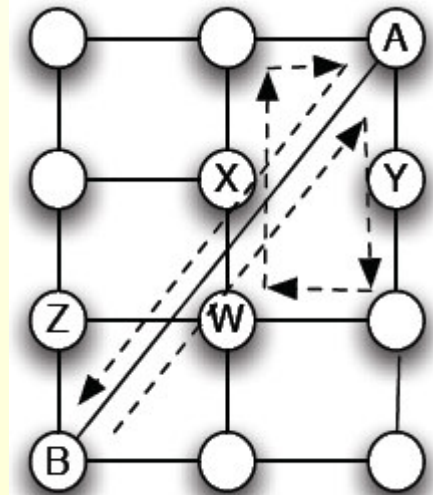
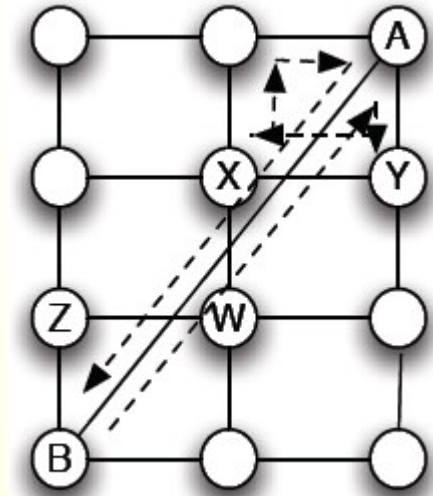
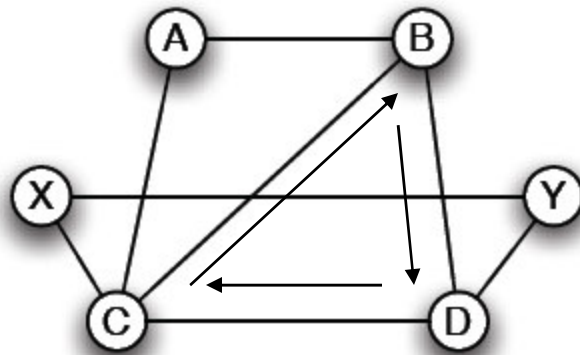
Cross Link Detection Protocol

- A link is not removable, if it is traversed twice.
- At a crossing of L with L' remove the removable link. If neither is removable, do nothing.
- First cut at a protocol: do the probing sequentially.
- For different probing sequences, one can get different graphs.
- Or, probe in a lazy fashion.



Multiple Crossing Links

- If a link is crossed by multiple other links, we probe it multiple times.
- Probing a pair of cross links may not find all the crossings, if they are obscured by other links.



Issues with CLDP

- How many probes? In what order?
- Can we probe the links concurrently?
 - Lock a link when it is probed
- Say we finish all the probes, and do face routing on the graph. Can we guarantee that the face routing always succeeds?

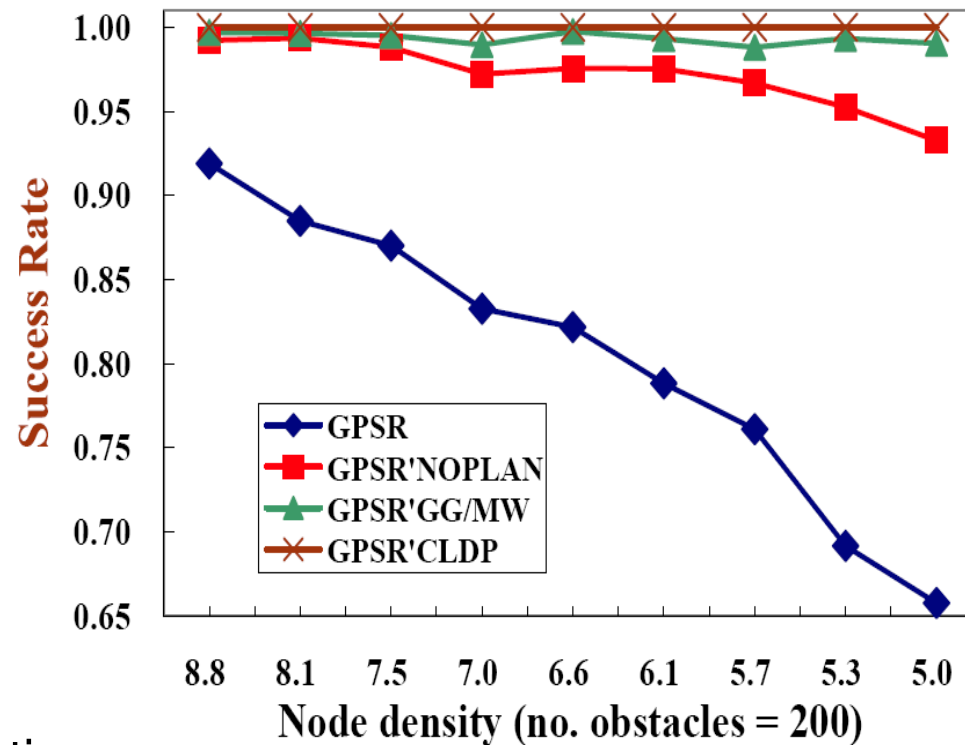
Concurrent CLDP Protocol

- CLDP can be made to work in a distributed manner
- If a connected graph G has at least one crossing, then there is at least one face w. a crossing
- Geographic routing never fails on a connected graph G after CLDP removes links

Success Rate

- TOSSIM
- 200 nodes

Radio graphs with 200 obstacles

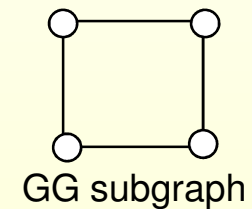
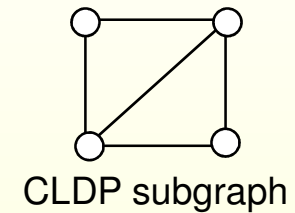
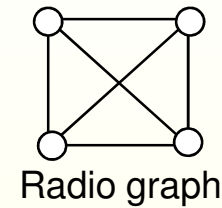
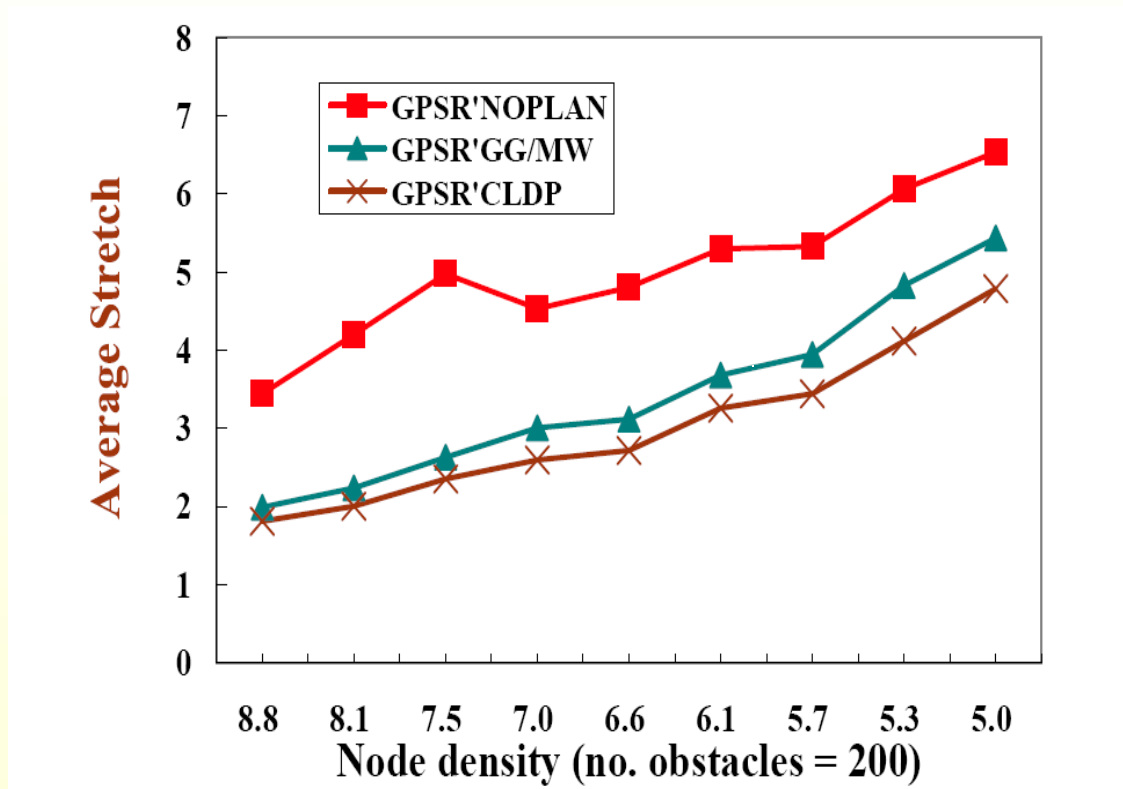


Observations

- CLDP is perfect on radio graphs with obstacles.
- GPSR performs poorly due to partitions and unidirectional links

Stretch

- *measured path length / shortest path length*

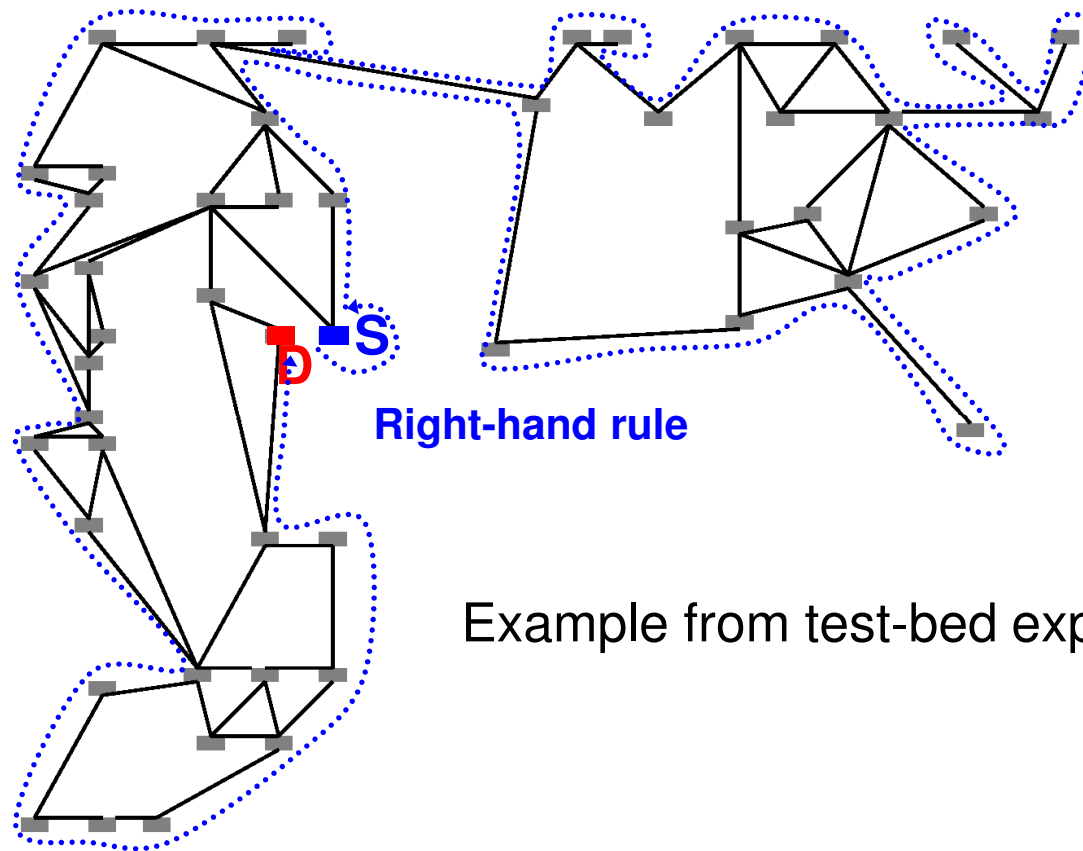


Observations

- CLDP (2~4) outperforms GPSR'GG/MW
 - GG planarization removes more links than CLDP

Worst Case Stretch

- Worst case stretch – $O(n)$, n : number of nodes
- Worst case path length – $O(n/l)$, l : optimal path length



Summary on Geographic Routing

- Geographical routing is nice in that it requires
 - No flooding
 - No routing table maintenance
 - Is highly scalable
- But planarization is tricky, and
- Face routing is nice in theory, but challenging in practice.

Additional Considerations

- We noted that the trouble is due to face routing.
- Is greedy routing robust to localization noise?
- Can we ignore the real coordinates and use instead **virtual coordinates** for routing? – These may be easier to obtain ...

Virtual Coordinates

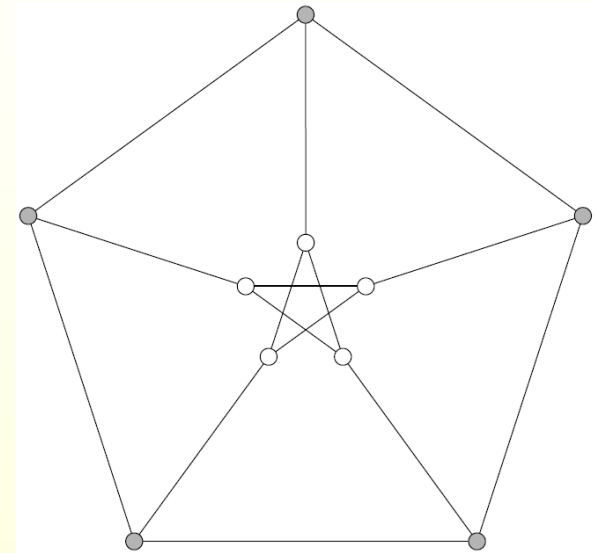
[Rao, A., Ratnasamy, S., Papadimitriou, C.,
Shenker, S., and Stoica, I.

Geographic Routing Without Location Information.

Proc. 9th Annual international Conference on
Mobile Computing and Networking
(MobiCom, 2003, 96-108]

Rubber Band Drawing of a Graph

- All edges are rubber bands.
- Nail down some nodes S in the plane, let the graph go.
- Theorem: the algorithm converges to a unique state – rubber band representation extending S .



Peterson graph with one pentagon nailed down.

Rubber Band Drawing of a Graph

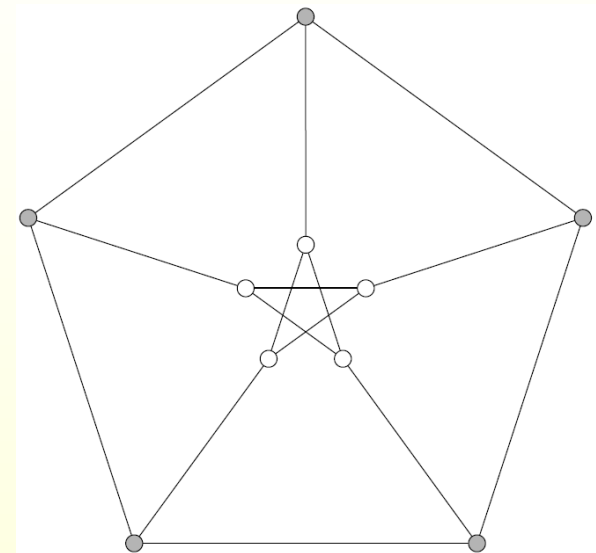
- The rubber band algorithm minimizes the total energy:

$$\mathcal{E}(x) = \sum_{ij \in E} |x_i - x_j|^2$$

- Claim: $\mathcal{E}(x)$ is convex.

$$\mathcal{E}(x) = \sum_{ij \in E} \sum_{k=1}^d (x_{ik} - x_{jk})^2.$$

- When any x_i goes to infinity, $\mathcal{E}(x)$ goes to infinity. So we have a **unique** global minimum.



Peterson graph with one pentagon nailed down.

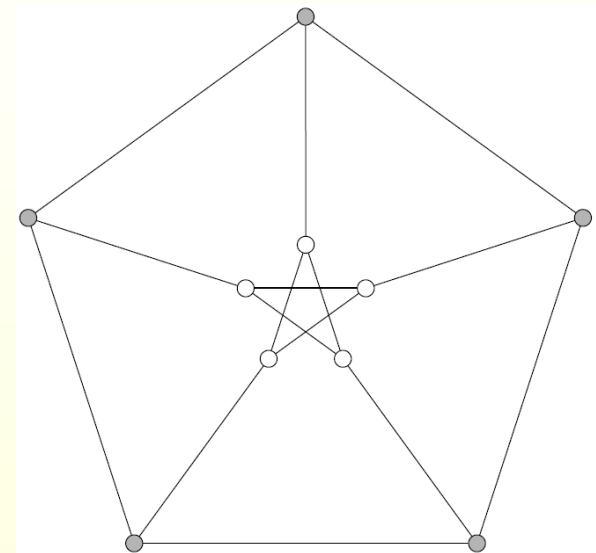
Rubber Band Drawing of a Graph

- How does the rubber band representation look like?

- $\partial E(x)/\partial x_i = 0$.

neighbors $\rightarrow \sum_{j \in N(i)} (x_j - x_i) = 0$.

- The rubber band connecting i and j pulls i with force $x_j - x_i$. The total force acting on x_i is 0.
- The graph is at equilibrium.



Peterson graph with one pentagon nailed down.

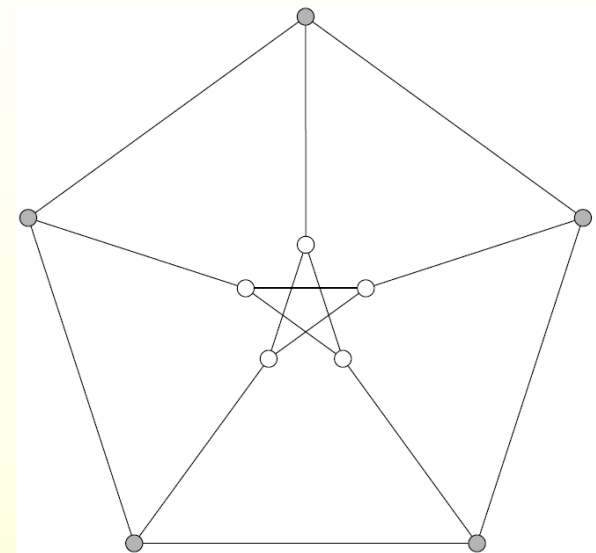
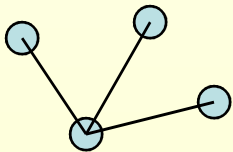
Rubber Band Drawing of a Graph

$$\sum_{j \in N(i)} (x_i - x_j) = 0.$$

1. Every free node is at the center of gravity of its neighbors.

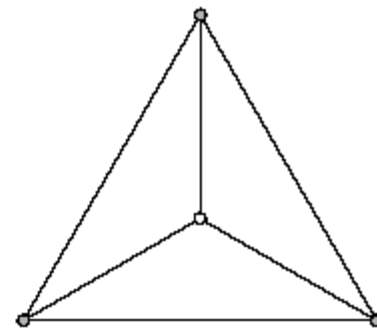
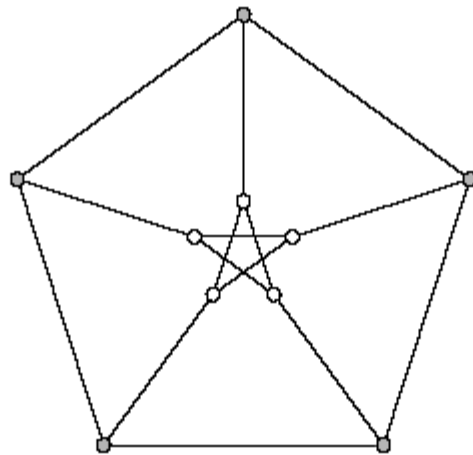
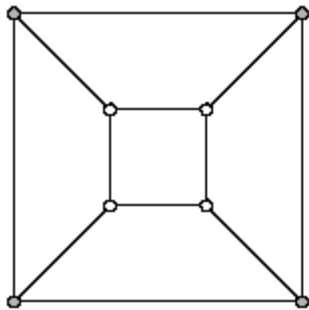
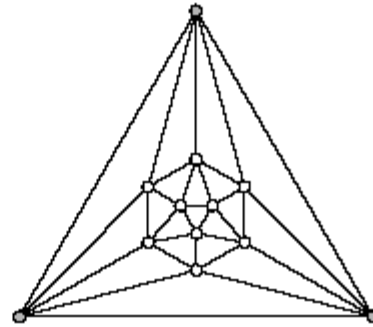
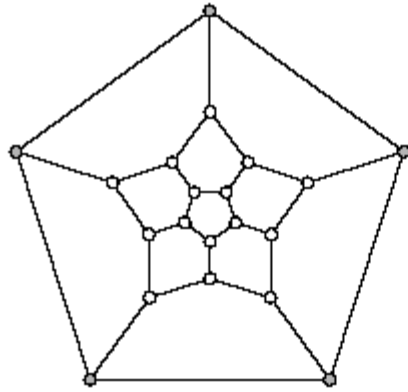
$$x_i = \frac{1}{d_i} \sum_{j \in N(i)} x_j.$$

2. no reflex vertices – free nodes go inside the CH of the pinned nodes



Peterson graph with one pentagon nailed down.

More Examples



Rubber Band Algorithm

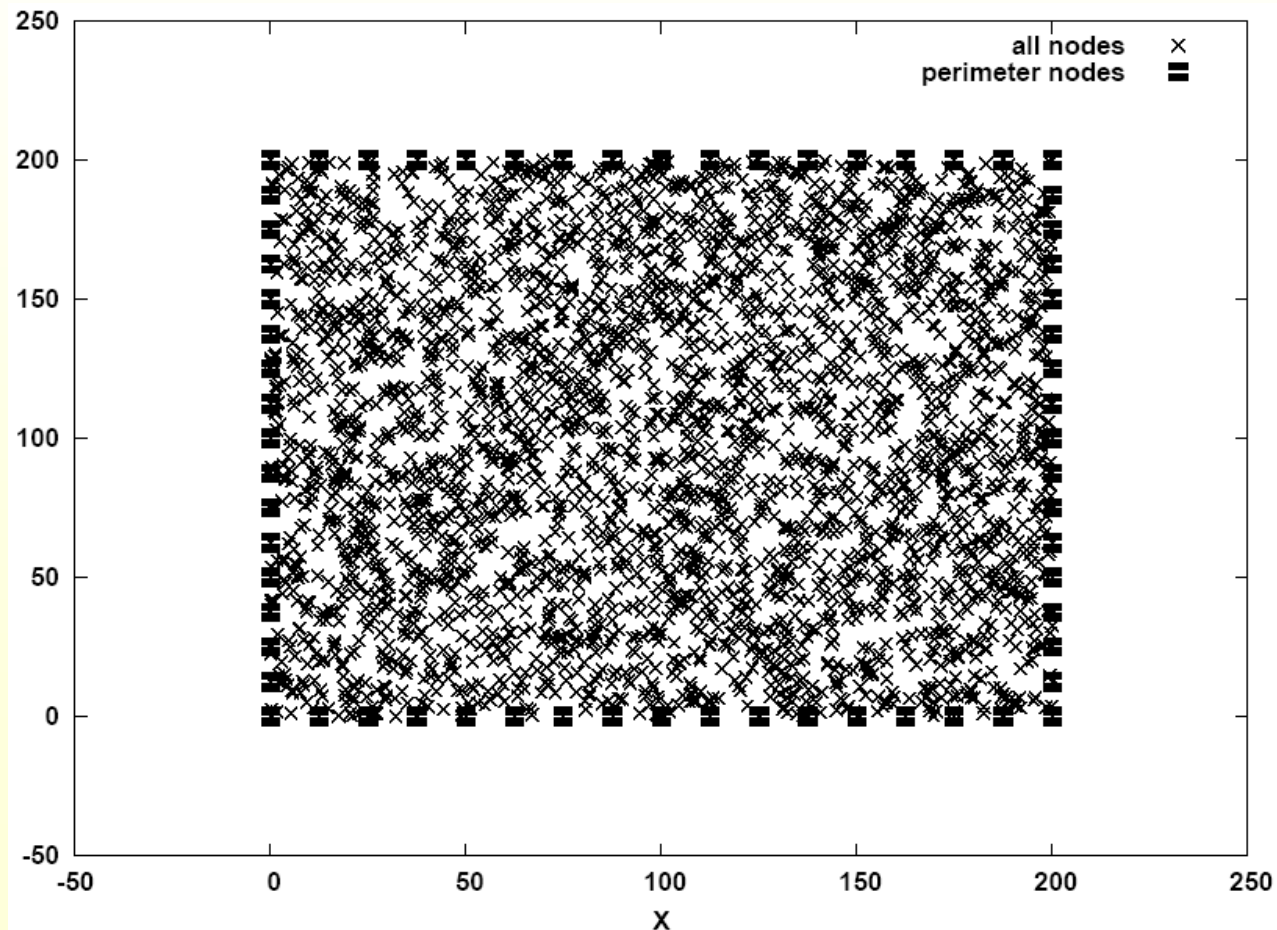
- Recall the mass-spring model
- First we assume nodes on the boundary know their location
- Fix the nodes on the outer boundary
- Iterative algorithm:
 - Every node moves to the center of gravity of its neighbors

$$x_i \leftarrow \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} x_j$$

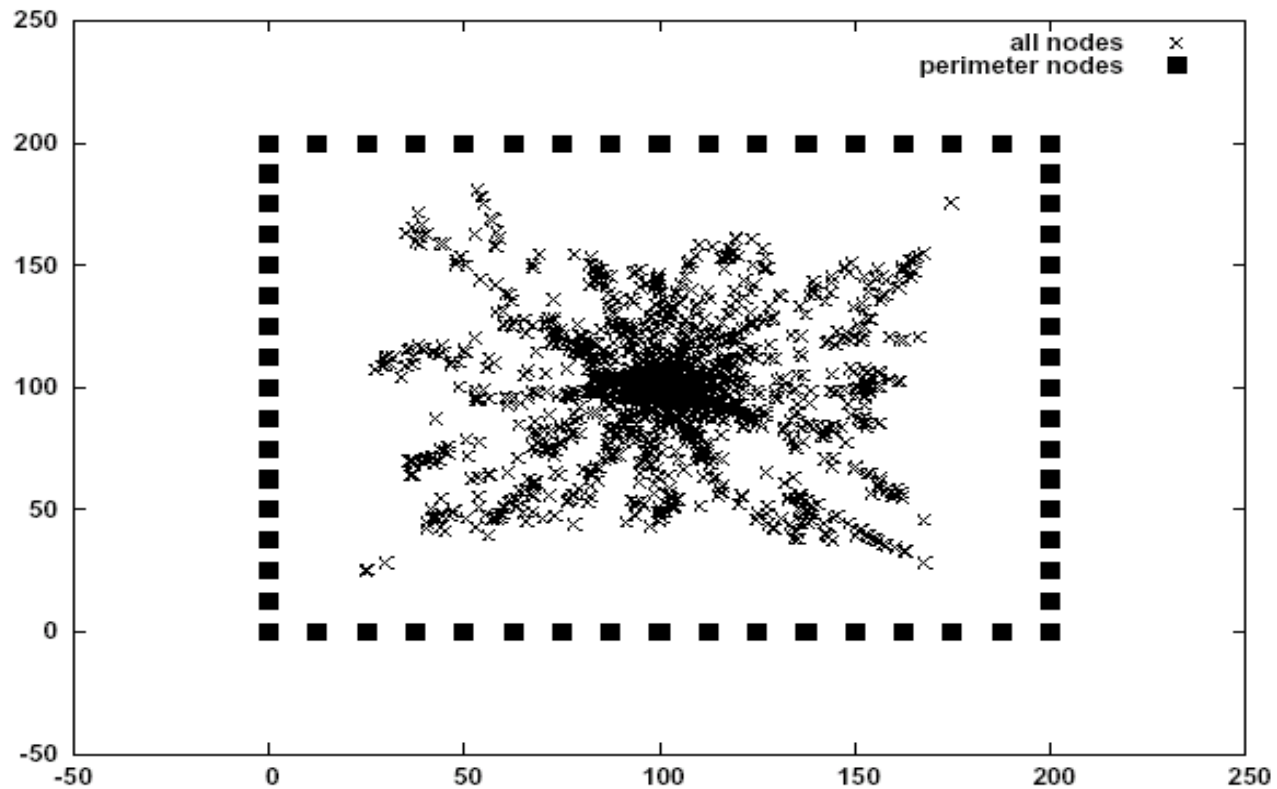
- Until no node moves more than distance δ

A Network with 3200 Nodes

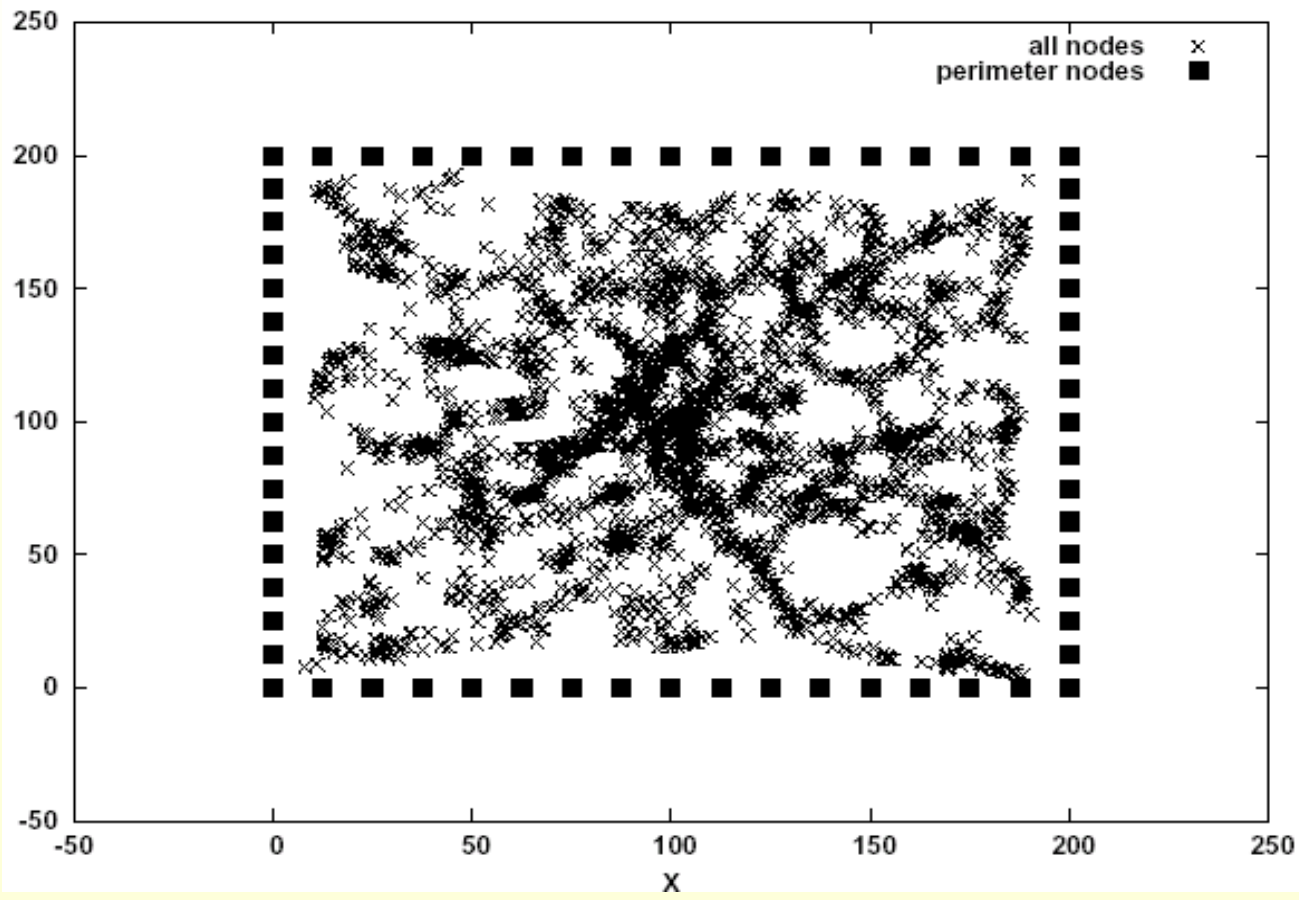
- Greedy routing success rate: 0.989, avg path length 16.8



Perimeter Nodes are Known (10 iterations)

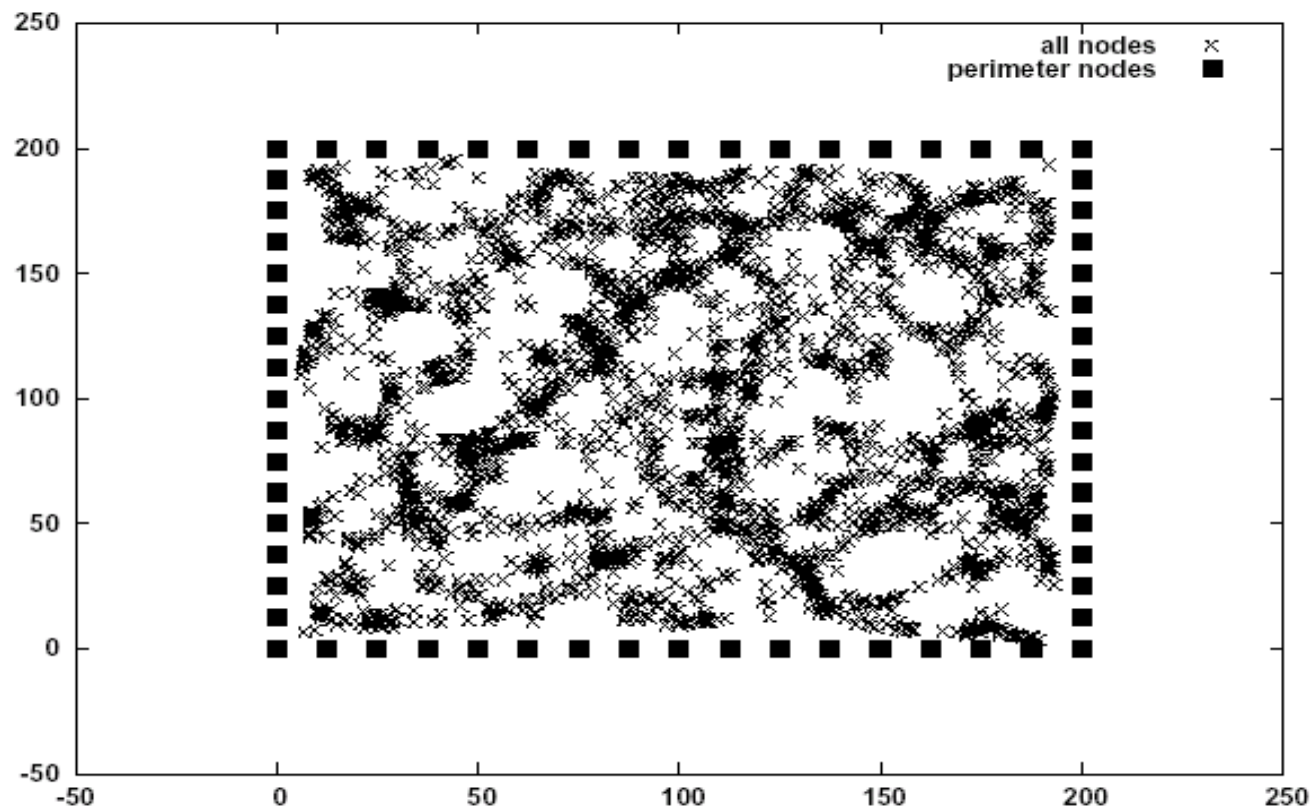


Perimeter Nodes are Known (100 iterations)



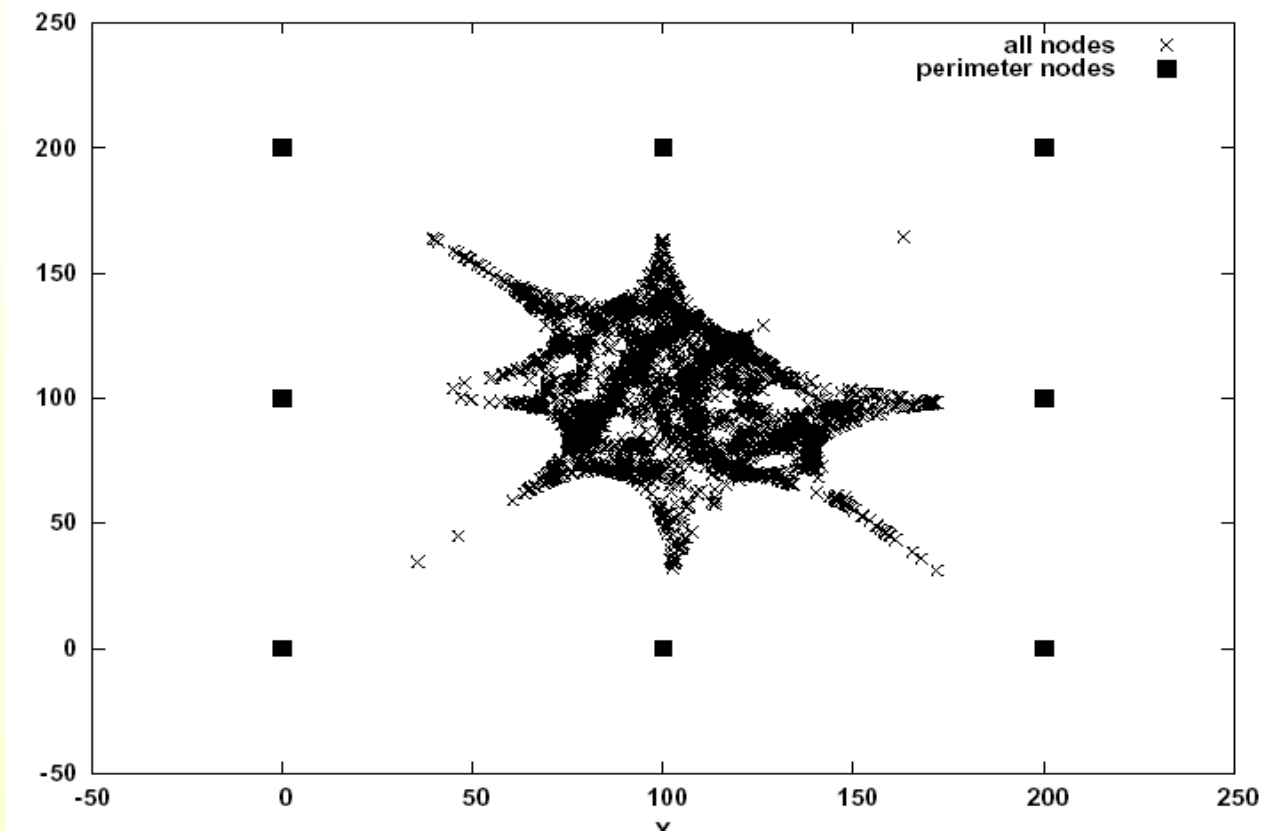
Perimeter Nodes are Known (1000 iterations)

- Greedy routing success rate: **0.993**, avg path length 17.1



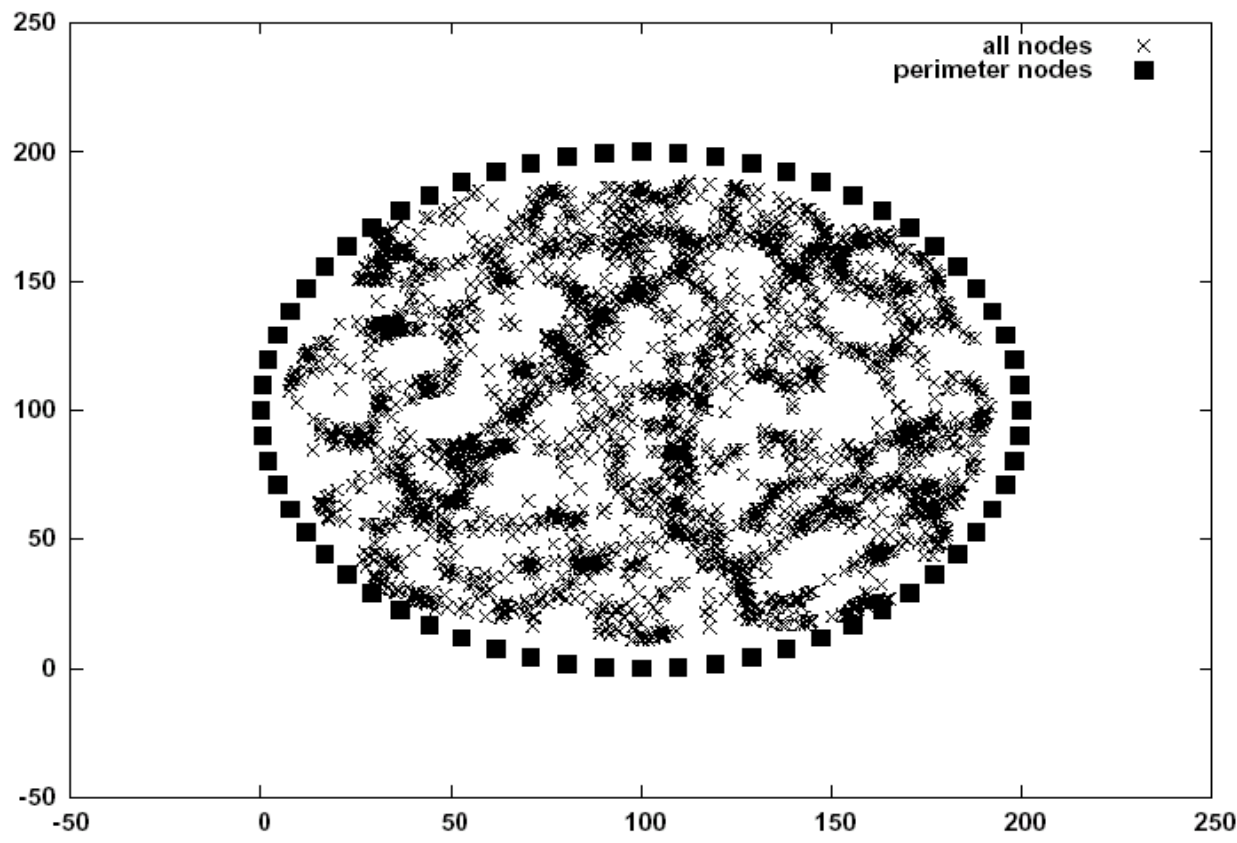
Resiliency of the Rubber Band Approach

- Greedy routing success rate: **0.981**, avg path length 17.3



Resiliency of the Rubber Band Approach

- Greedy routing success rate: **0.99**, avg path length 17.1



How to Fix Perimeter Nodes?

- Need nodes on the perimeter to “stretch” out the net
- First assume we know nodes on the perimeter, but not the locations
 1. Each perimeter sends hello messages
 2. All the nodes record hop counts to each perimeter node
 3. The hop count between every pair of perimeter node is broadcast to all perimeter nodes (quite expensive)
 4. Embed perimeter nodes in the plane, say by any localization algorithm

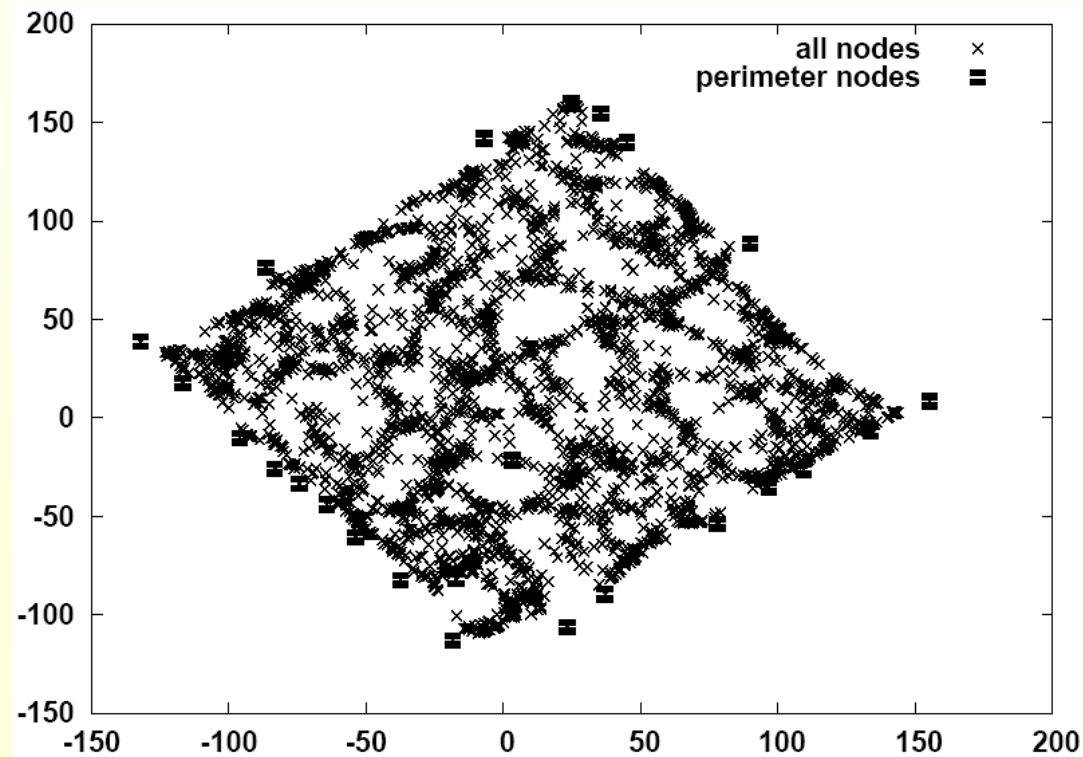
$$\sum_{i,j \in \text{perimeter_set}} (\text{measured_dist}(i, j) - \text{dist}(i, j))^2$$

Perimeter Nodes

1. The embedding only gives **relative positions**:
include 2 bootstrapping beacons in the embedding of perimeters.
 - Use the center of gravity as origin
 - 1st bootstrap node defines the positive x-axis
 - 2nd bootstrap node defines the positive y-axis
2. Non-perimeter nodes actually have the distances to all perimeter nodes, and embed themselves.
 - Gives good initial positions for the rubber band algorithm.

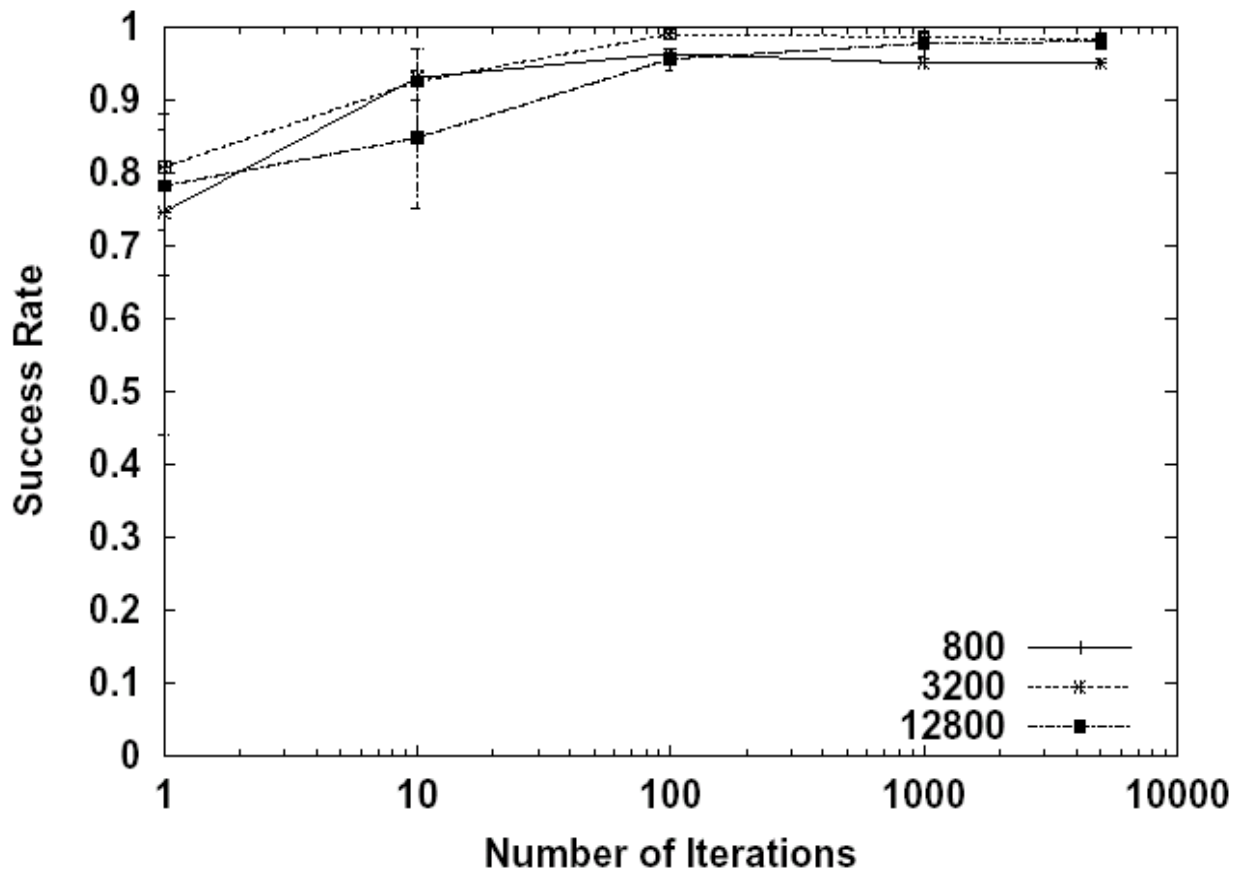
How to Find Perimeter Nodes?

- The bootstrapping nodes send hello messages to everyone.
- The node which is the farthest among all its 2-hop neighbors will identify itself as a perimeter node.

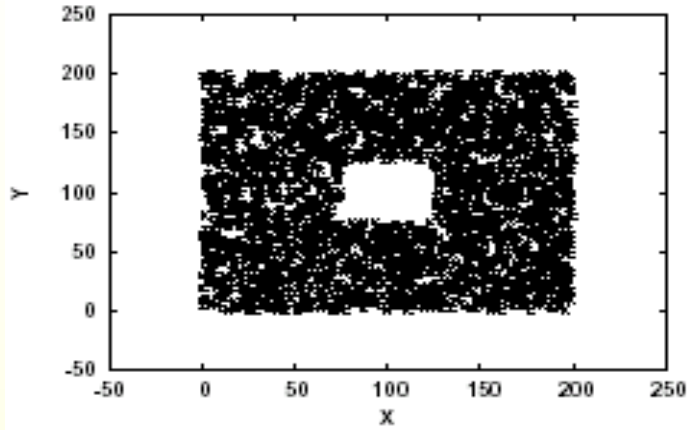


Success Rate of Greedy Routing

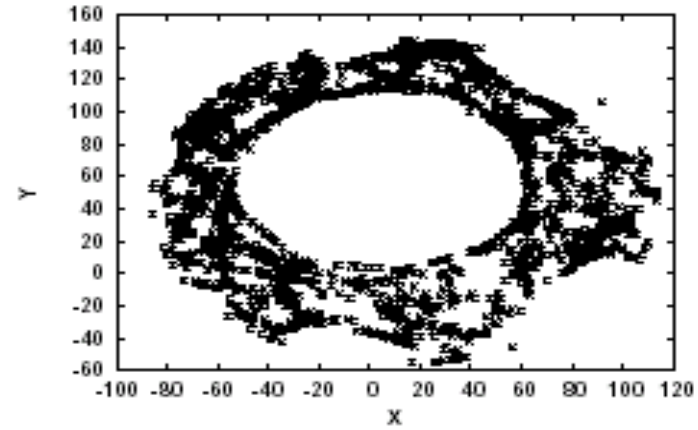
- Success rate on virtual coordinates is comparable with true coordinates, when the sensors are dense and uniform.



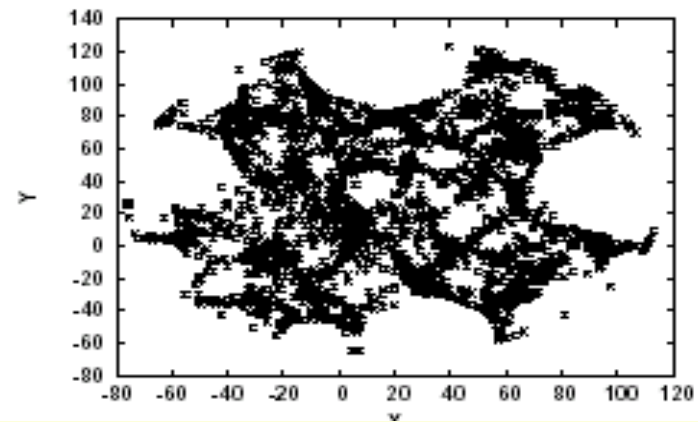
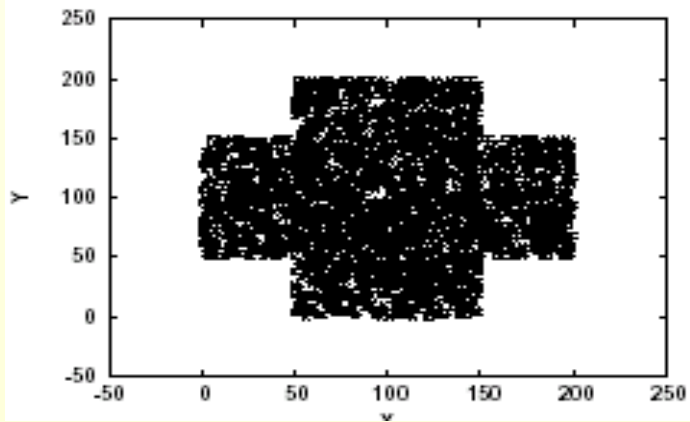
Weird Shapes



(a)

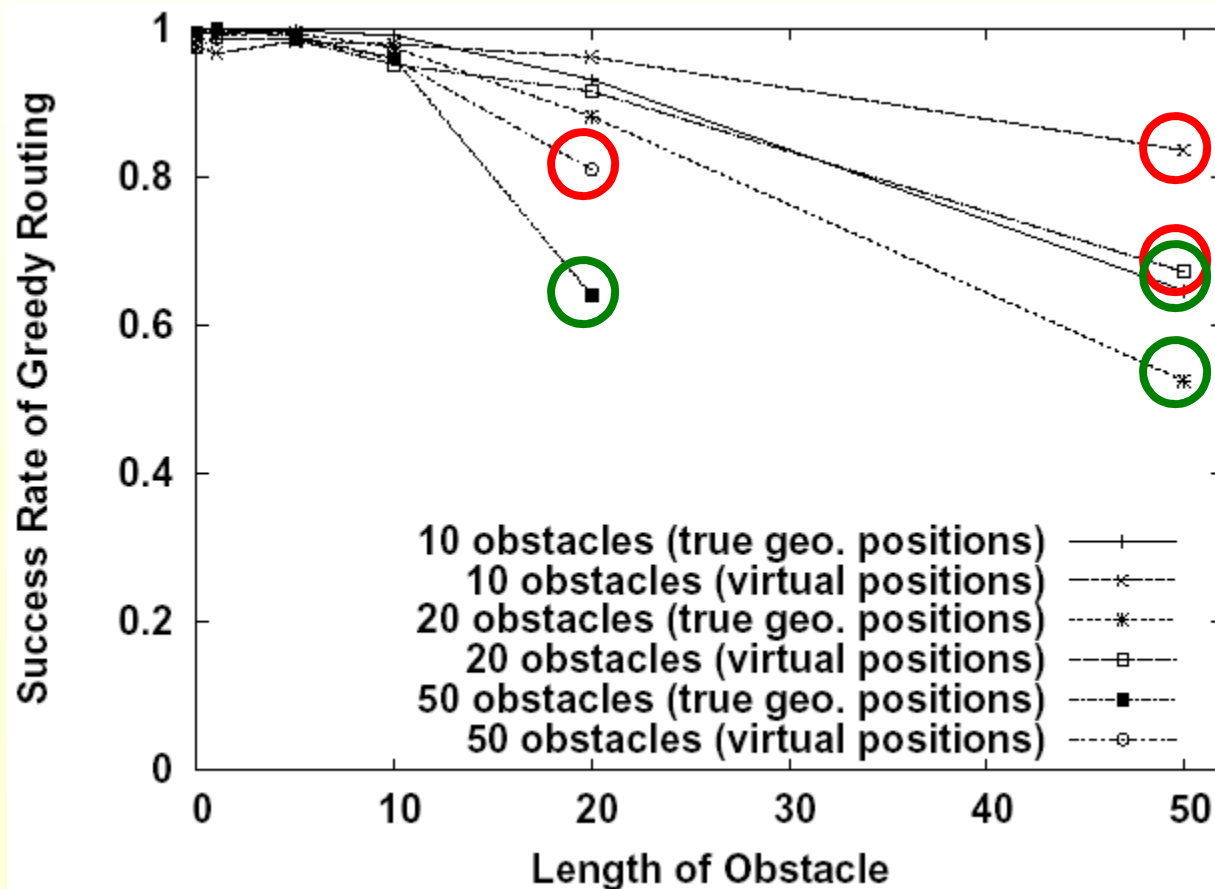


(b)



Obstacles

- Success rate on virtual coordinates degrades when there are a lot of obstacles, but better than w. true coordinates.



Planarization is Costly!

- Planarization is hard for real networks
 - GG and RNG don't work perfectly
- Planarization is complicated & costly!
 - CLDP (Kim et al., 2005)

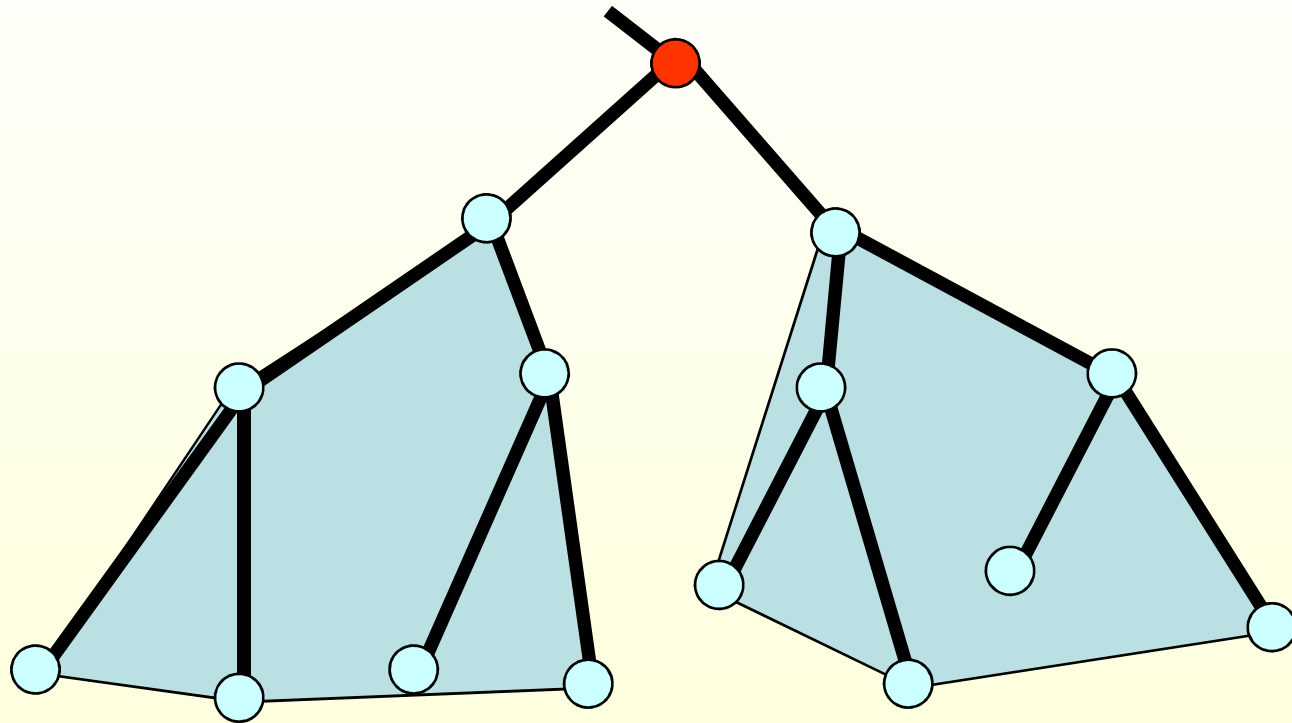
Doing without Face Routing

[Ben Leong, Barbara Liskov, and Robert Morris.
Geographic Routing without Planarization.
Proceedings of the 3rd Symposium on Network
Systems Design and Implementation (NSDI 2006).
San Jose, CA, May 2006]

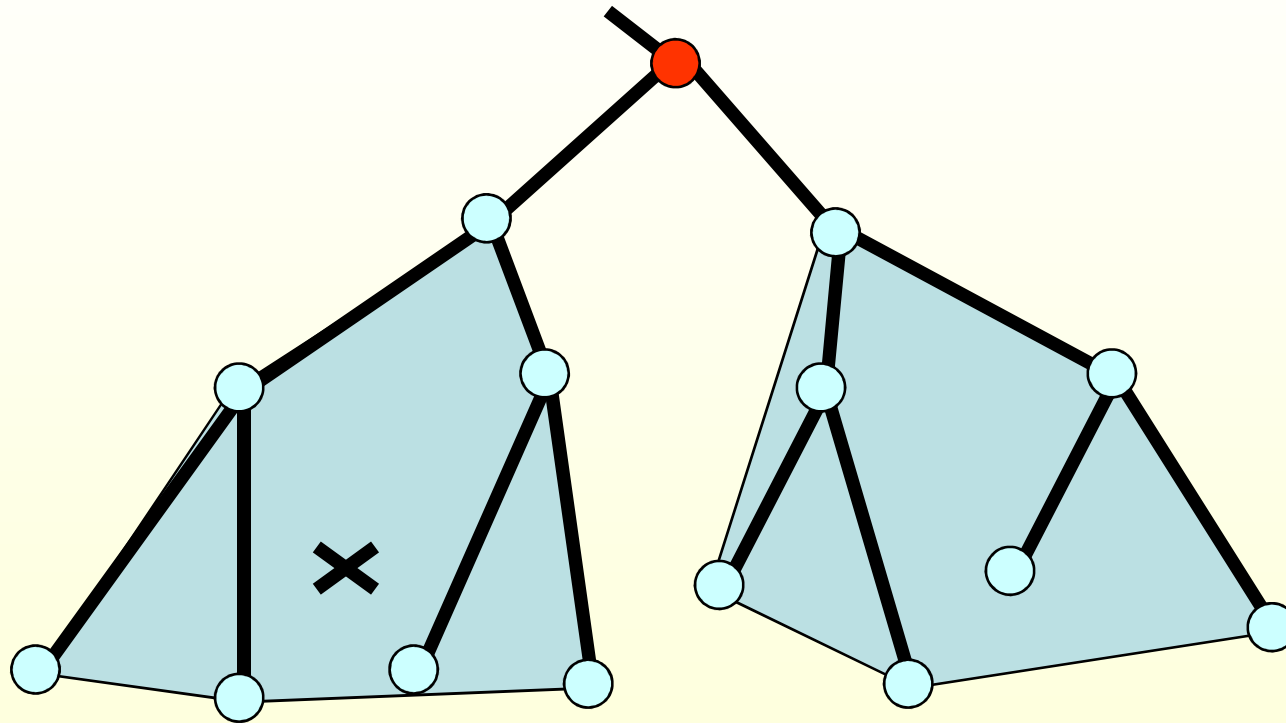
Greedy Distributed Spanning Tree Routing (GDSTR)

- Route on a spanning tree
- Use convex hulls to “summarize” the area covered by a subtree
 - convex hulls tells us what points are possibly reachable
 - reduces the subtree that must be traversed (smaller search problem)

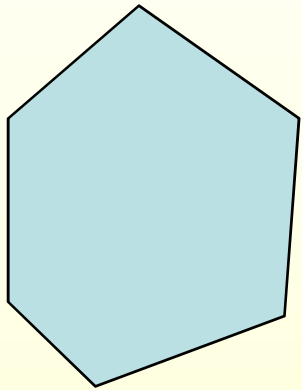
Hull Tree



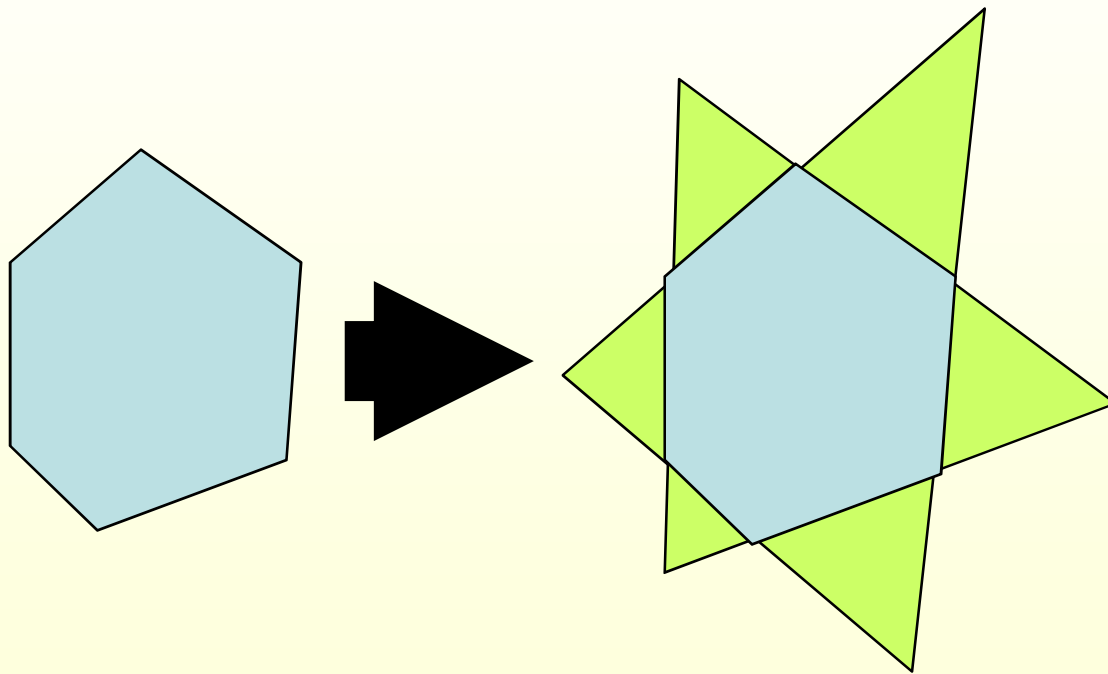
Hull Tree



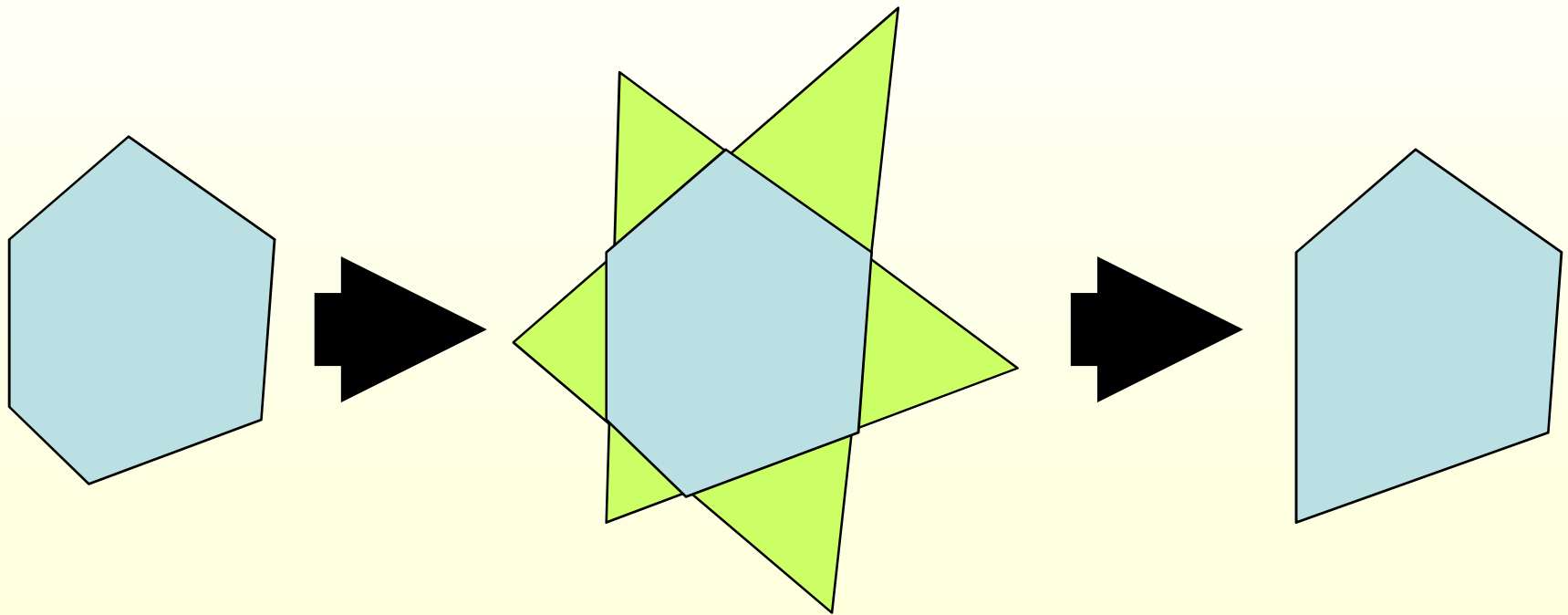
Reducing Convex Hulls



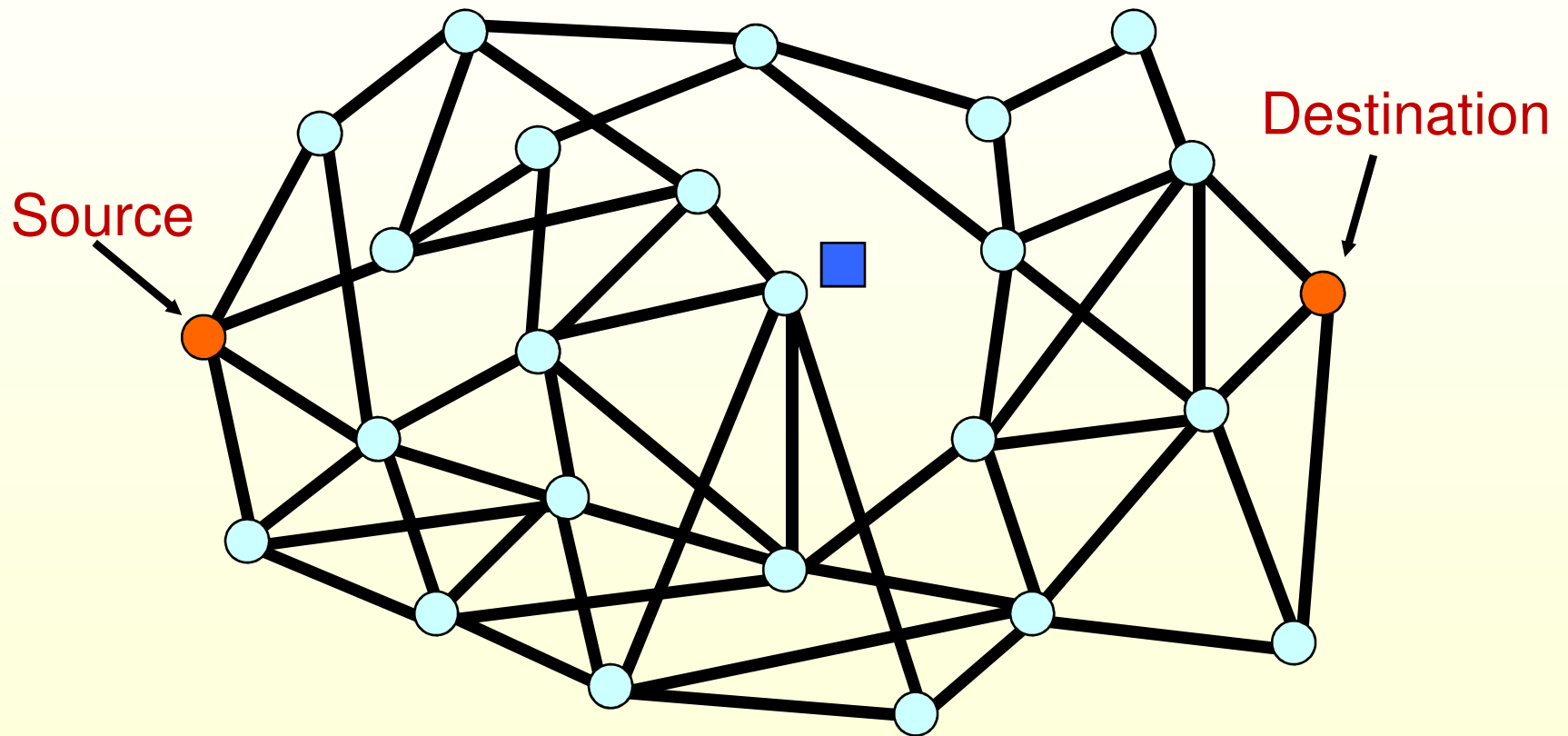
Reducing Convex Hulls



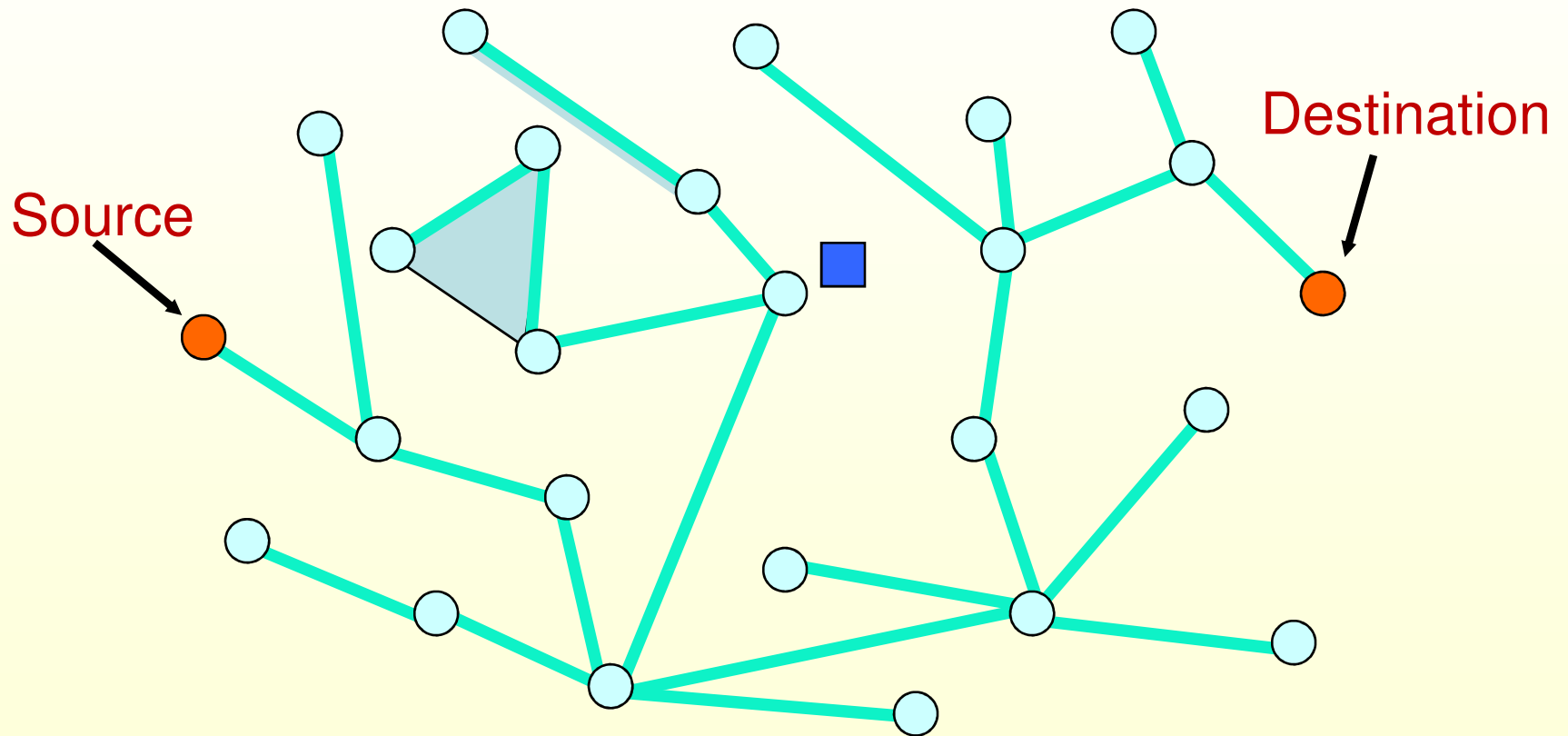
Reducing Convex Hulls



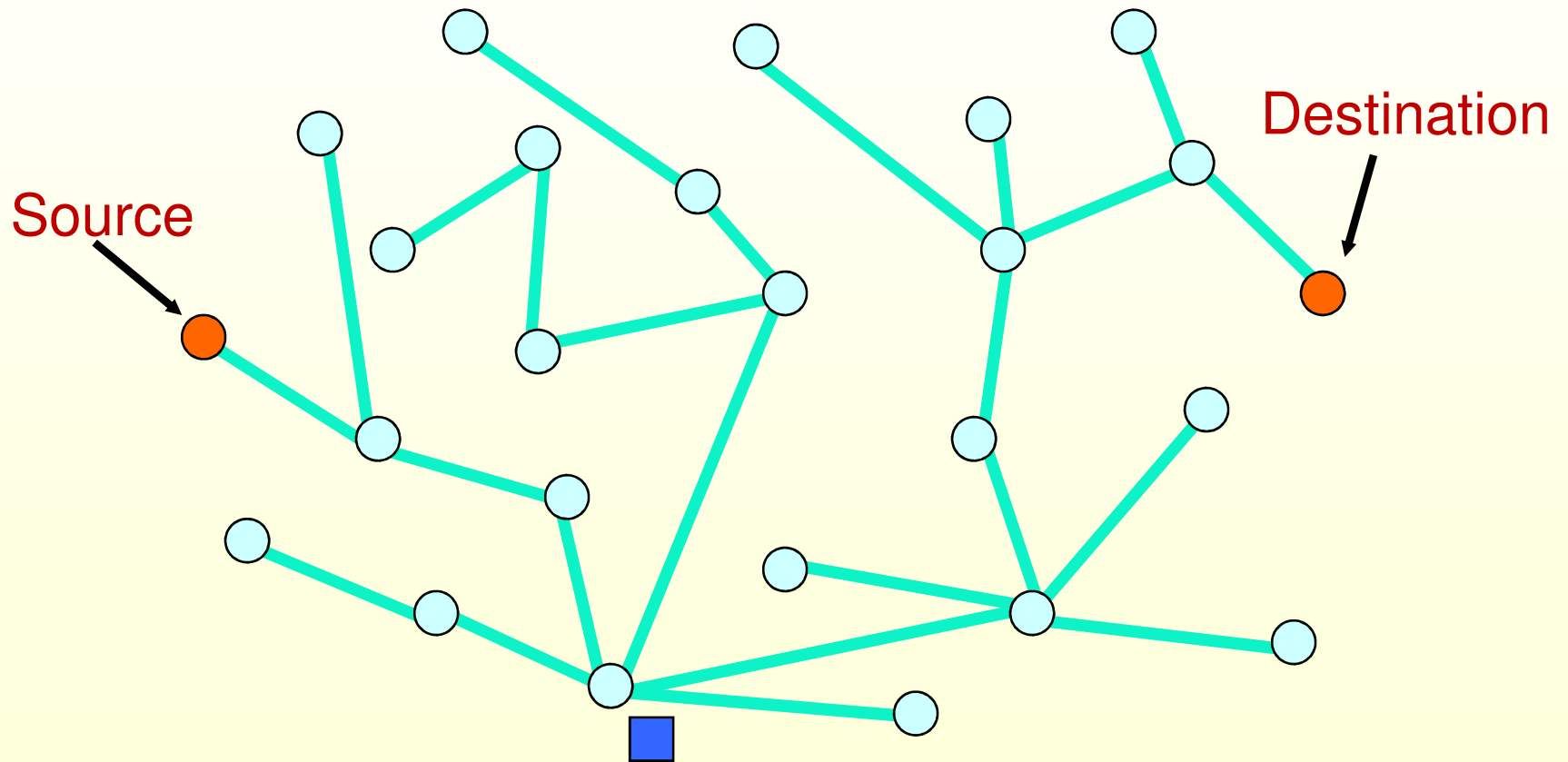
GDSTR Example



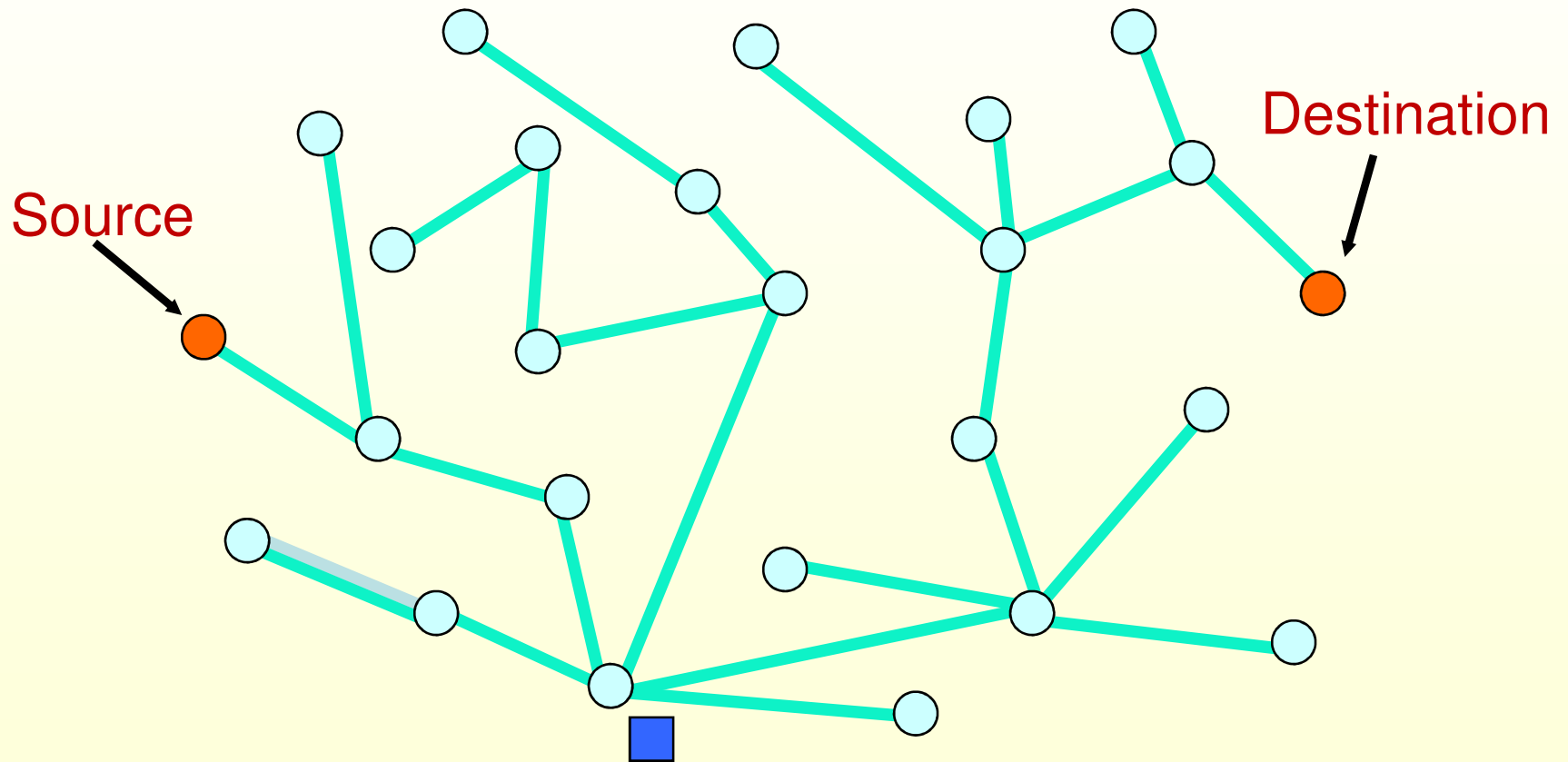
GDSTR Example



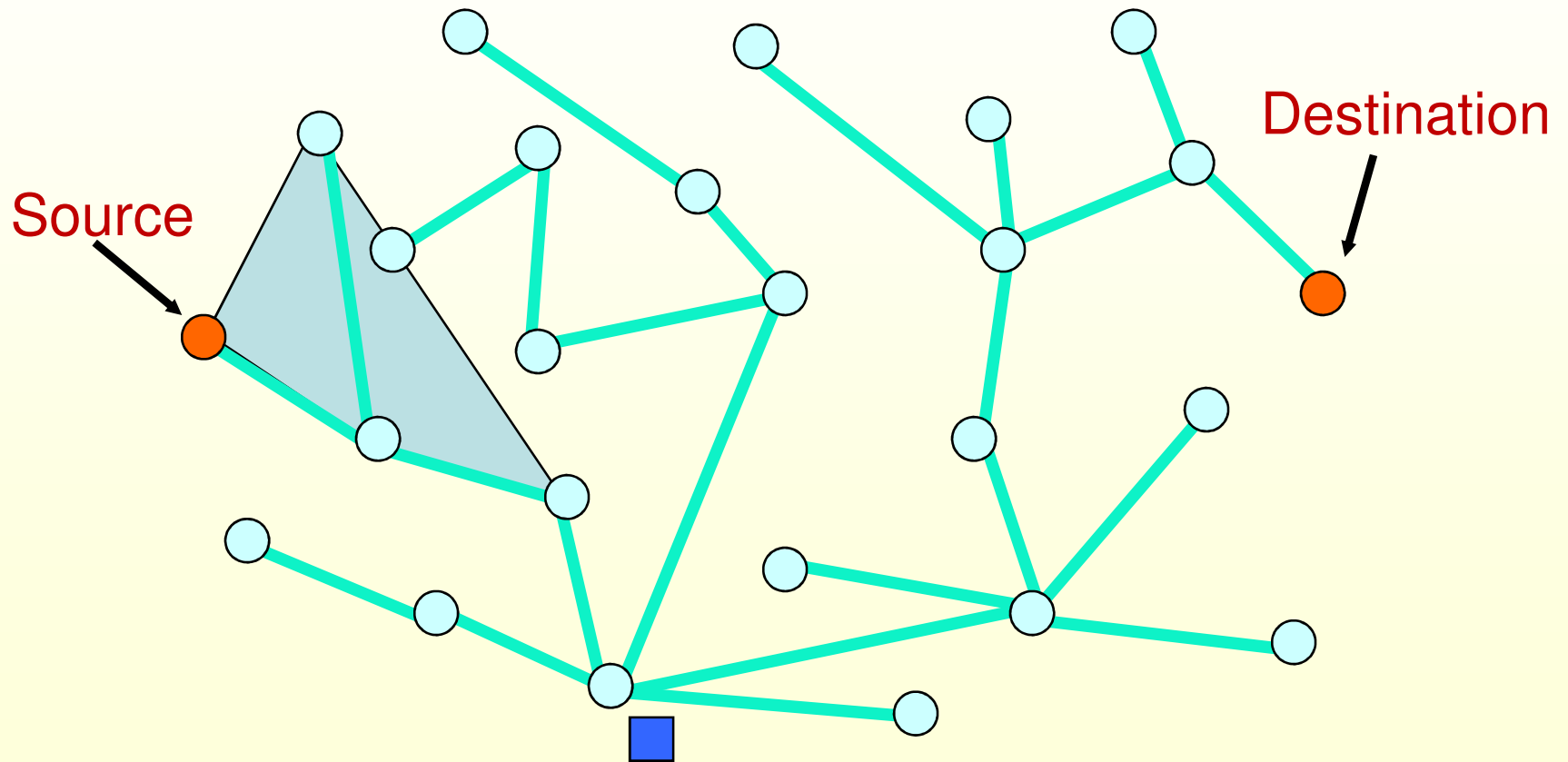
GDSTR Example



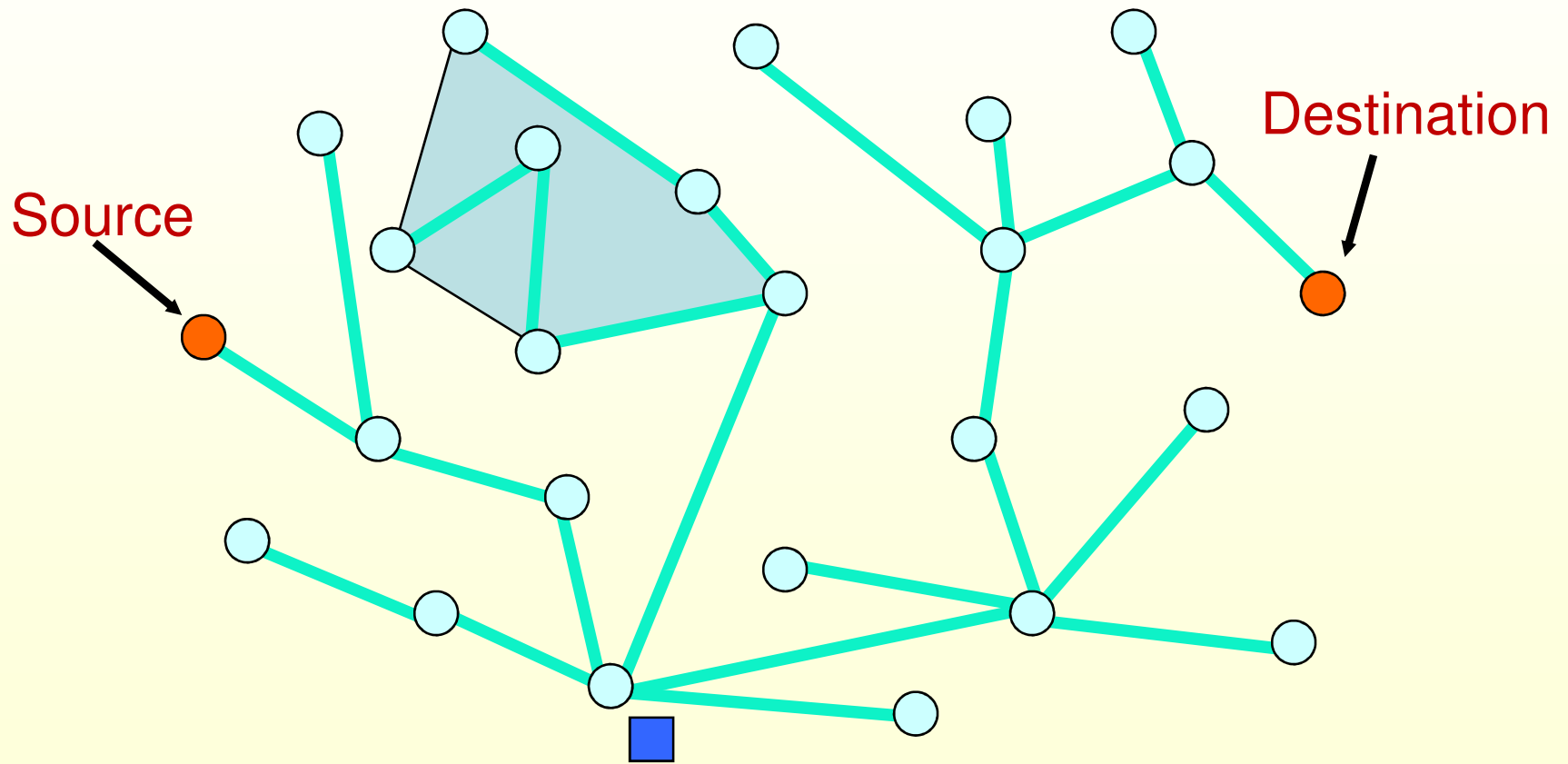
GDSTR Example



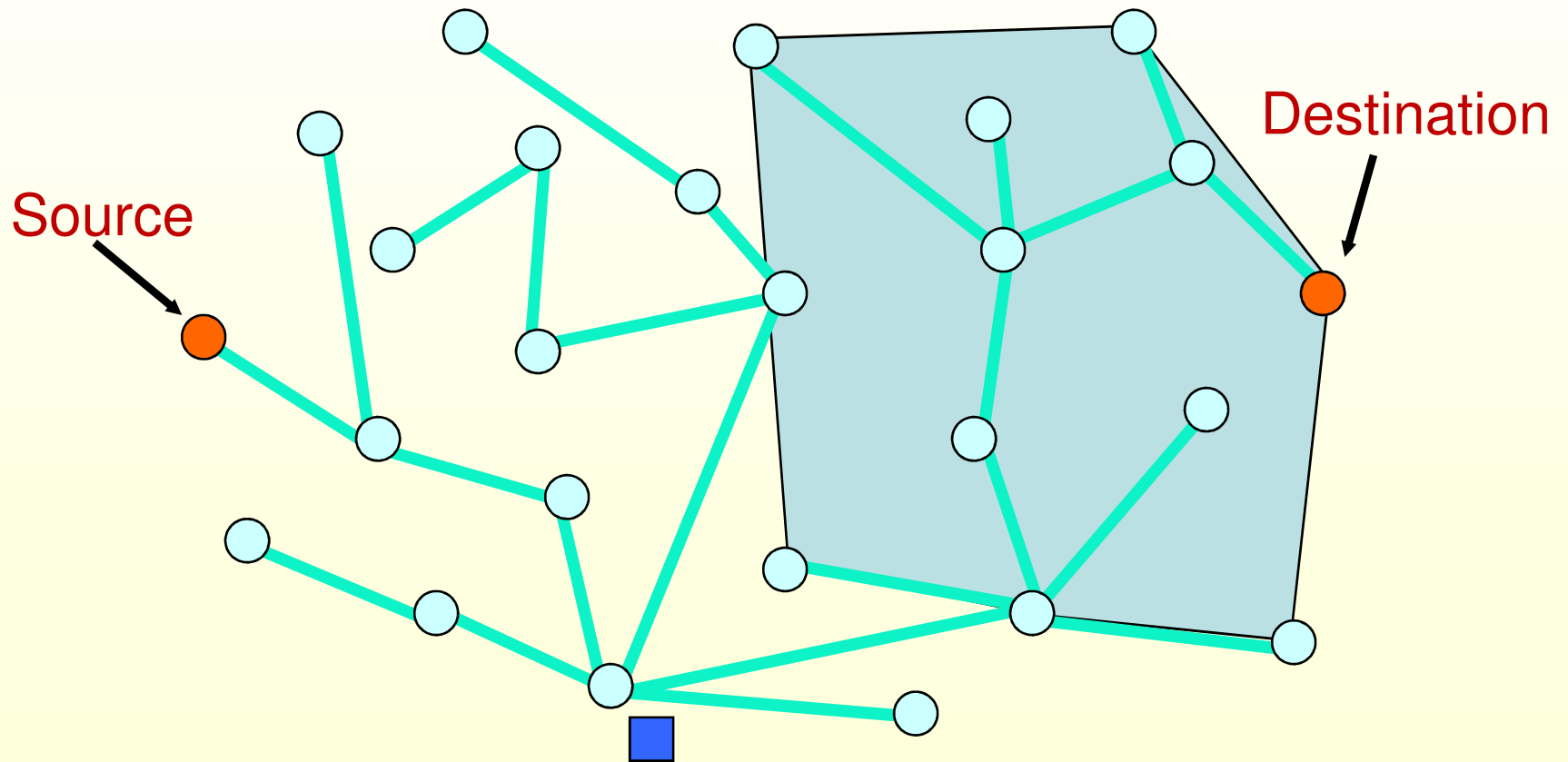
GDSTR Example



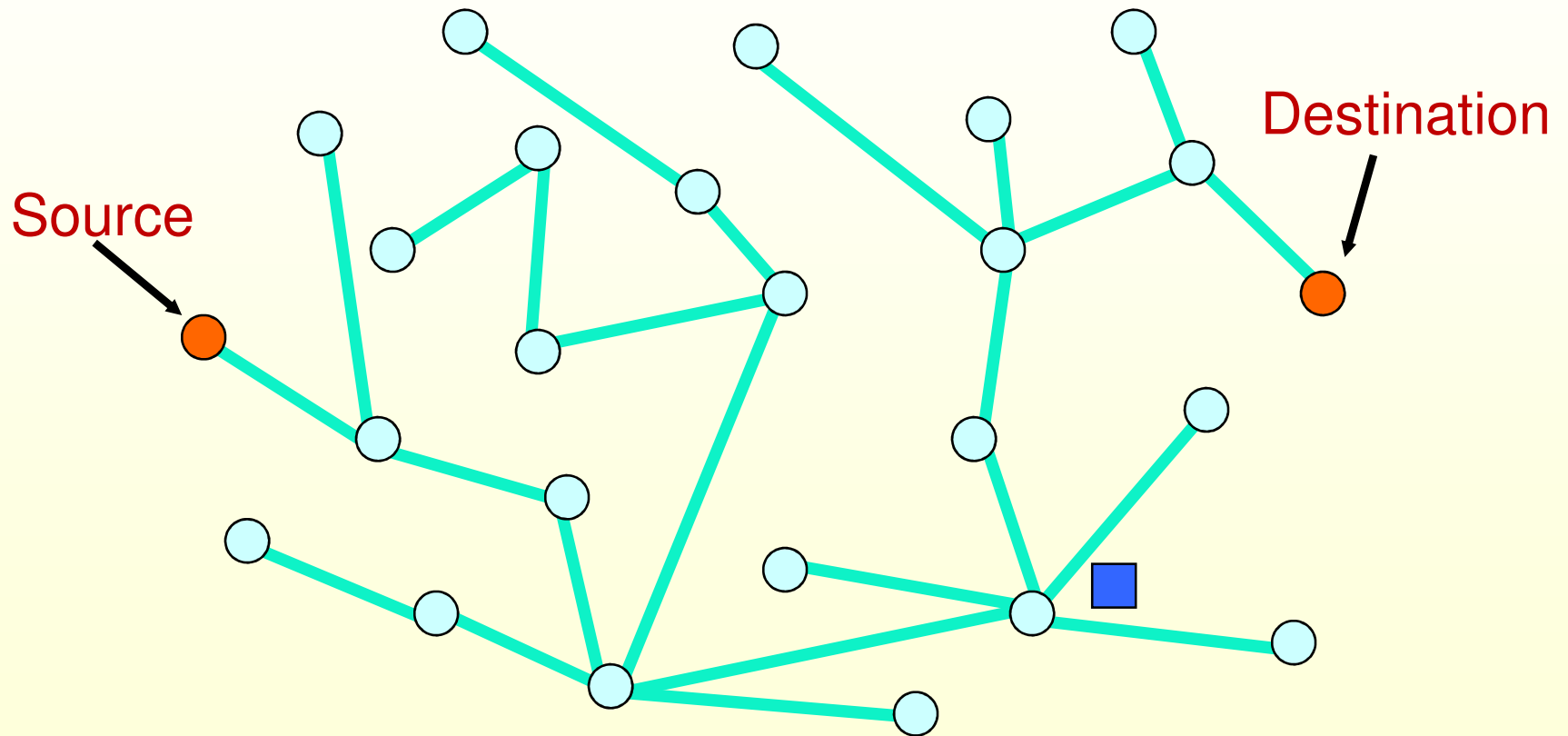
GDSTR Example



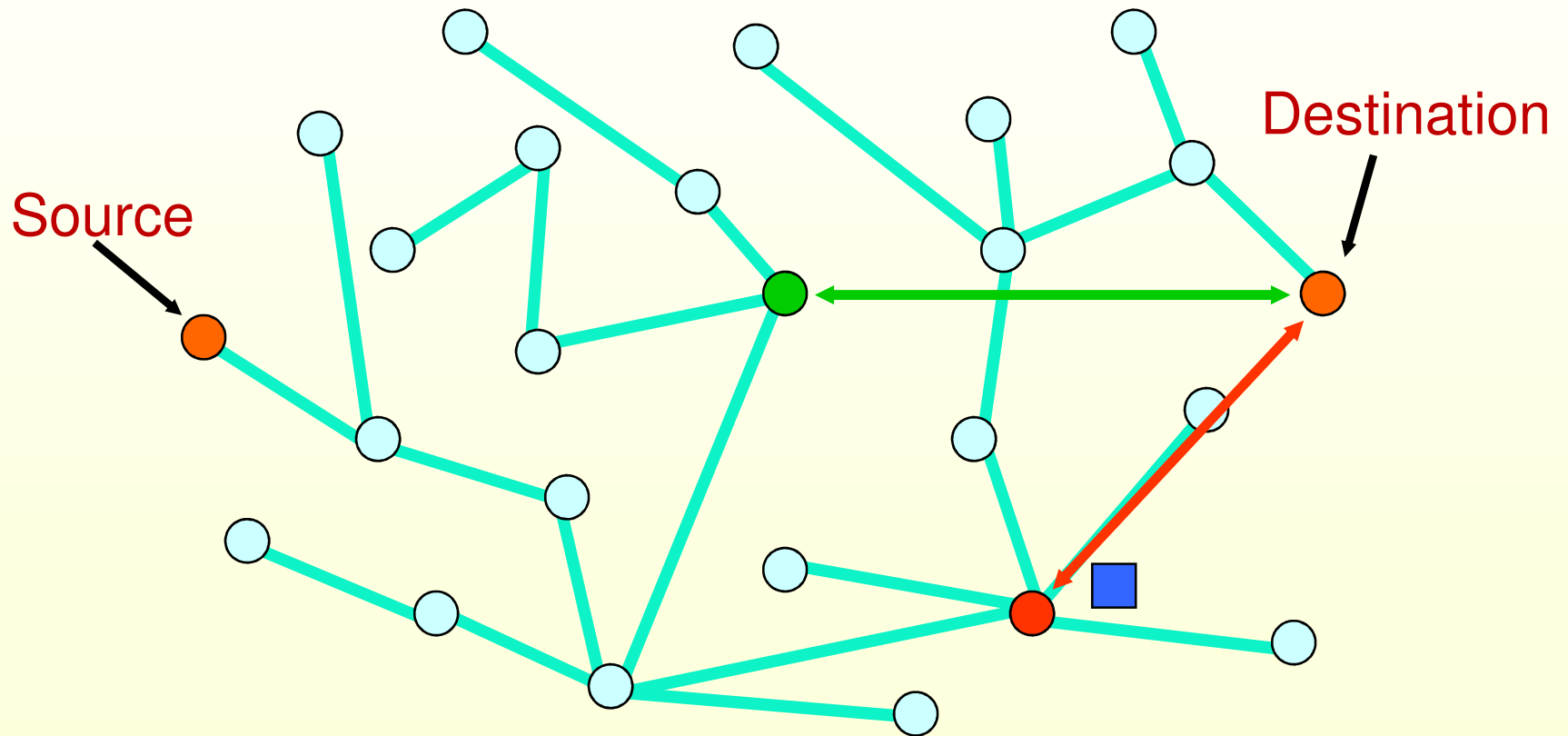
GDSTR Example



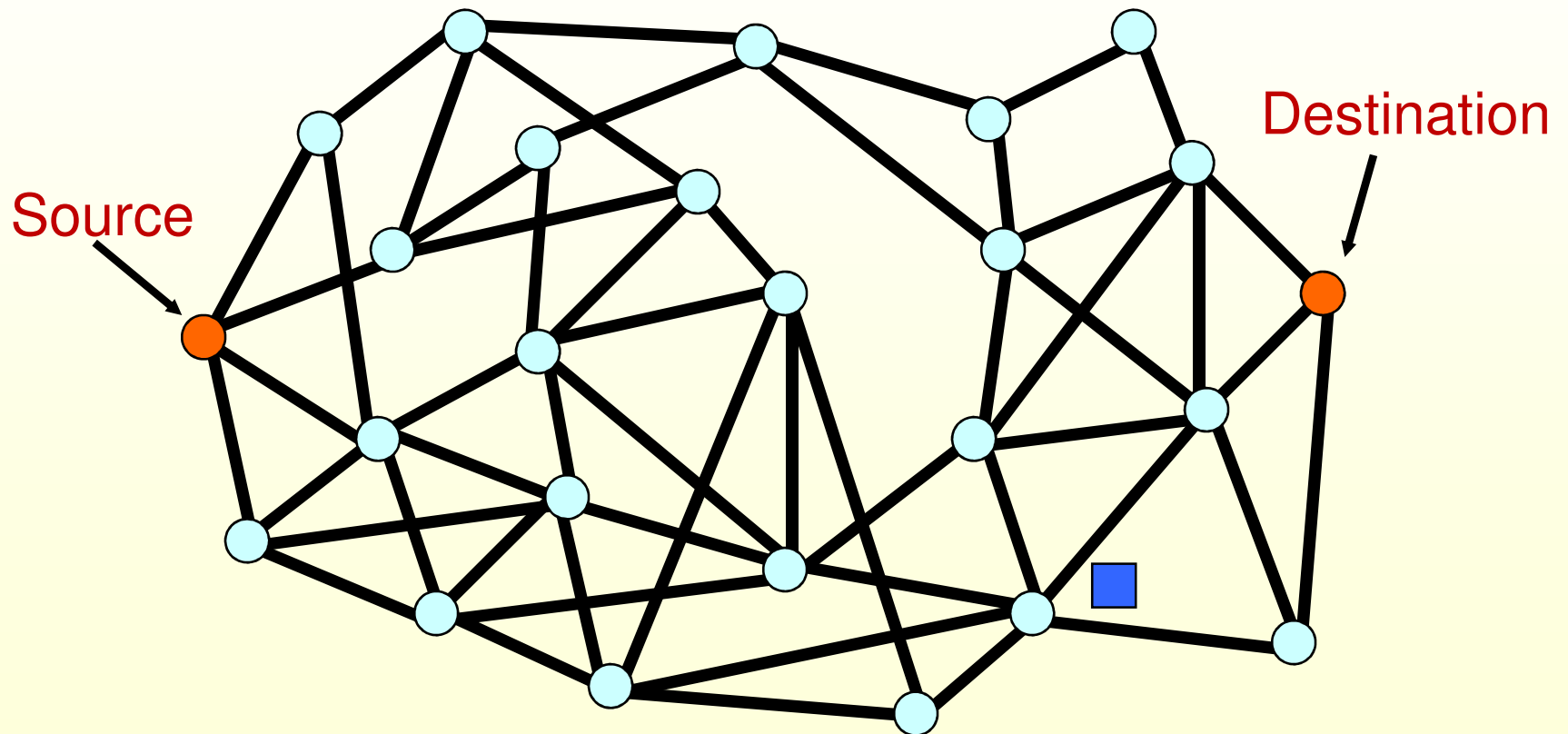
GDSTR Example



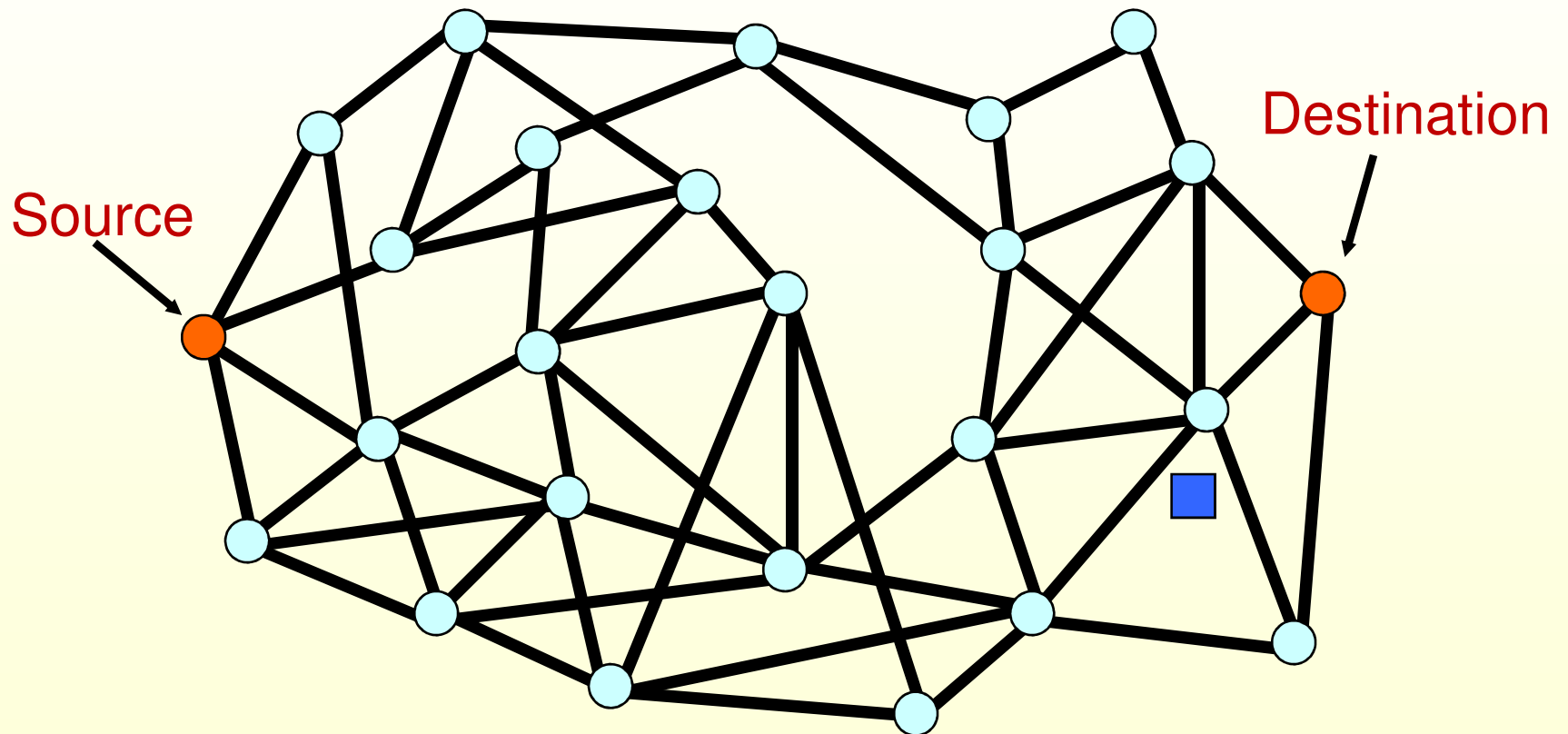
GDSTR Example



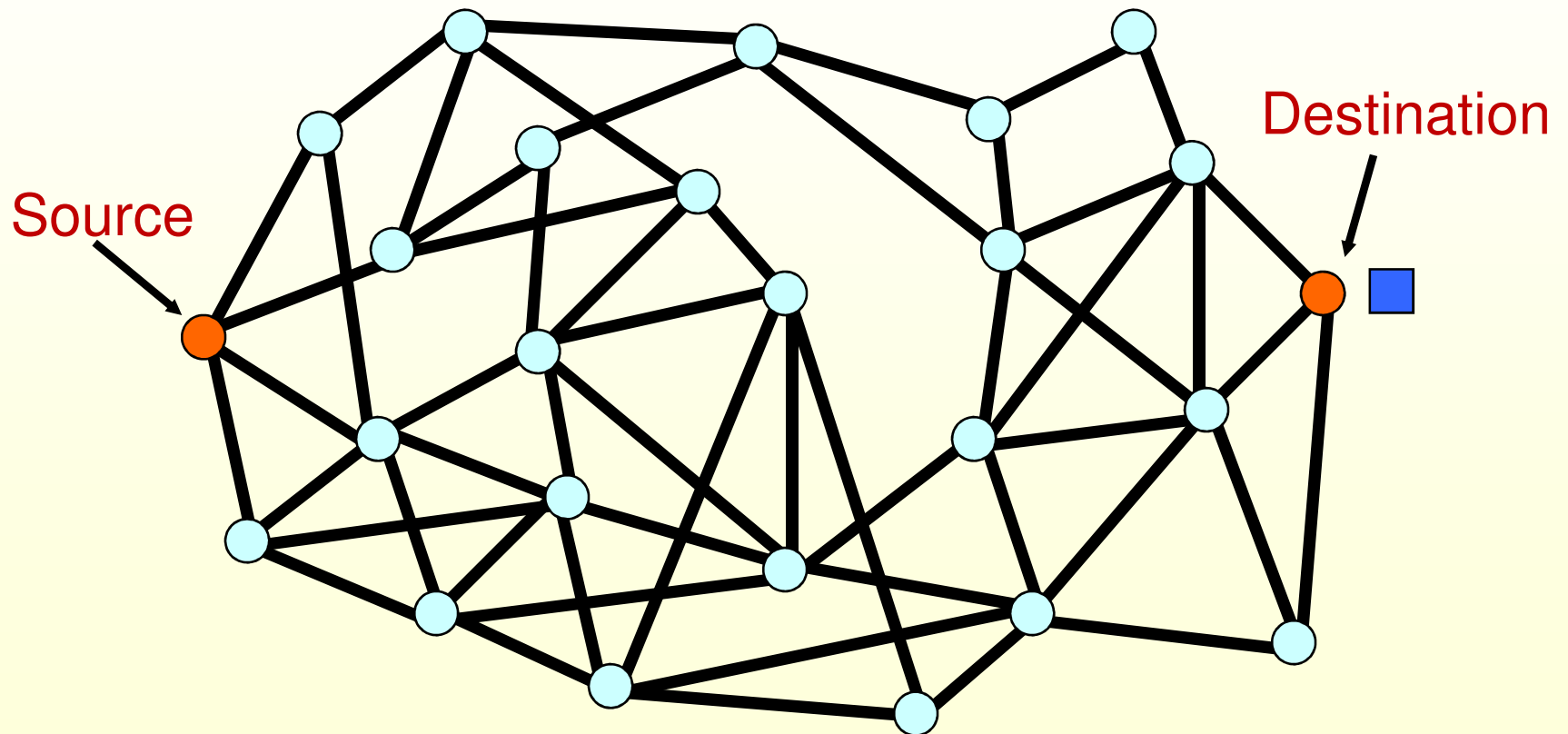
Revert to Greedy Forwarding



Revert to Greedy Forwarding



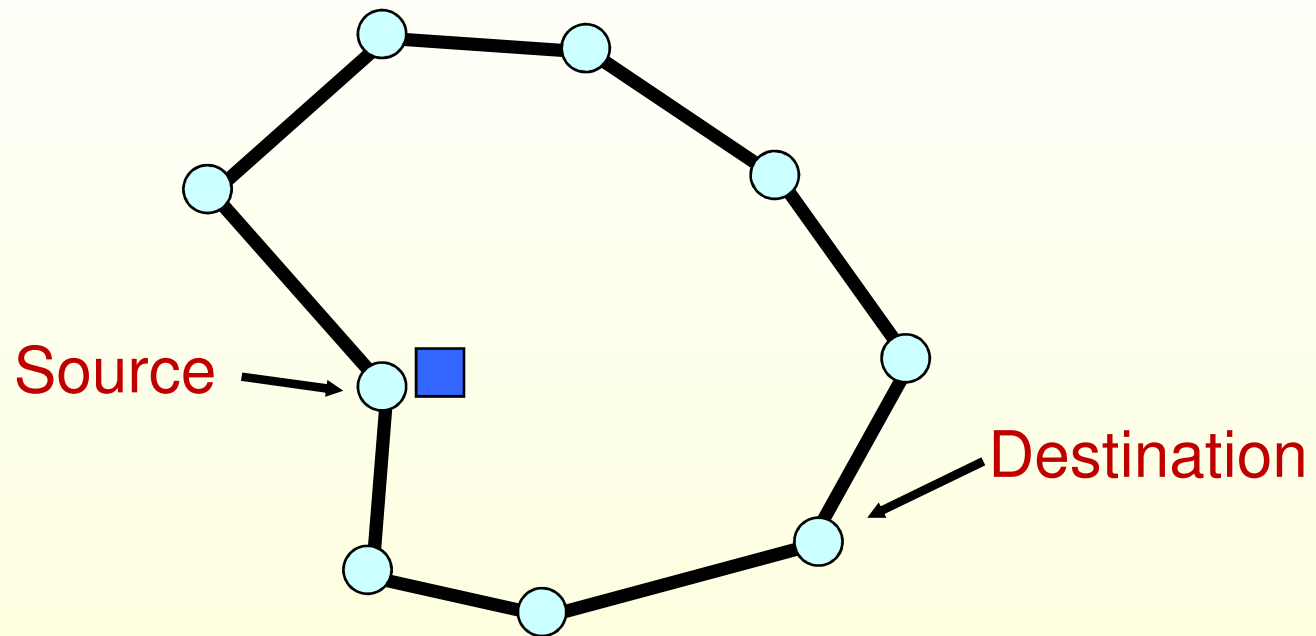
Revert to Greedy Forwarding



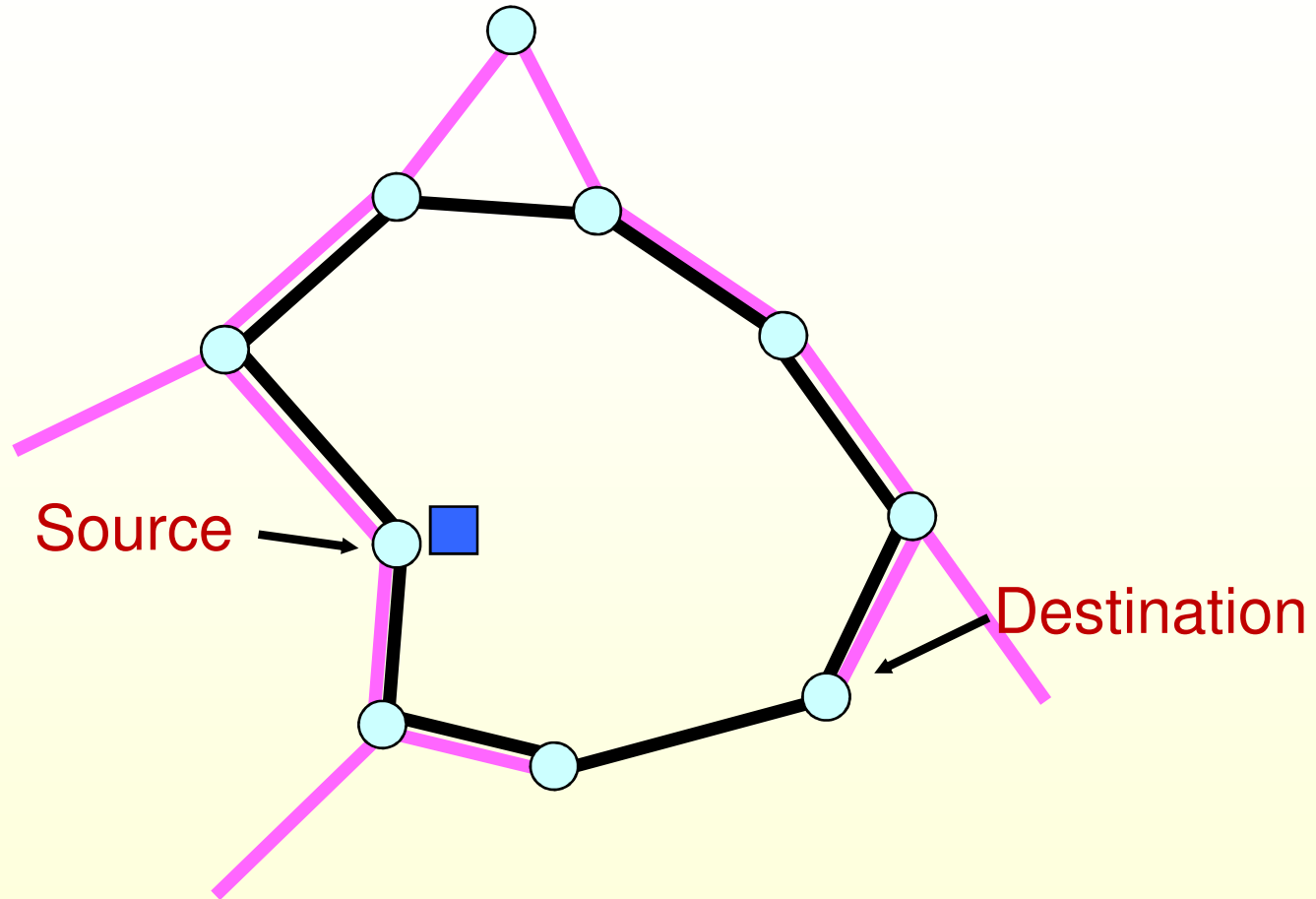
Issues

- Choosing forwarding direction
 - multiple hull trees
- Undeliverable packets
 - conflict hulls

Using Multiple Trees

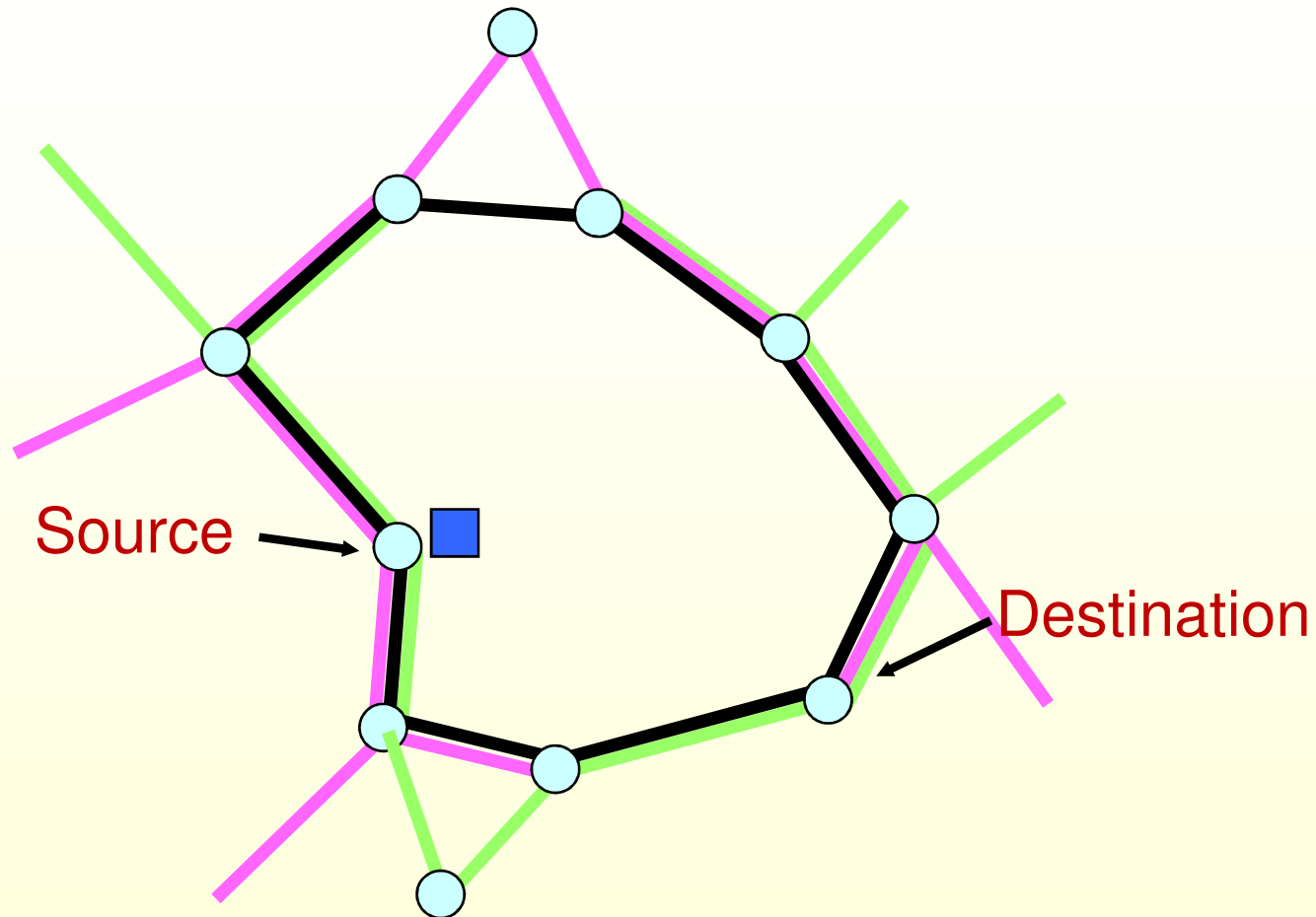


Using Multiple Trees



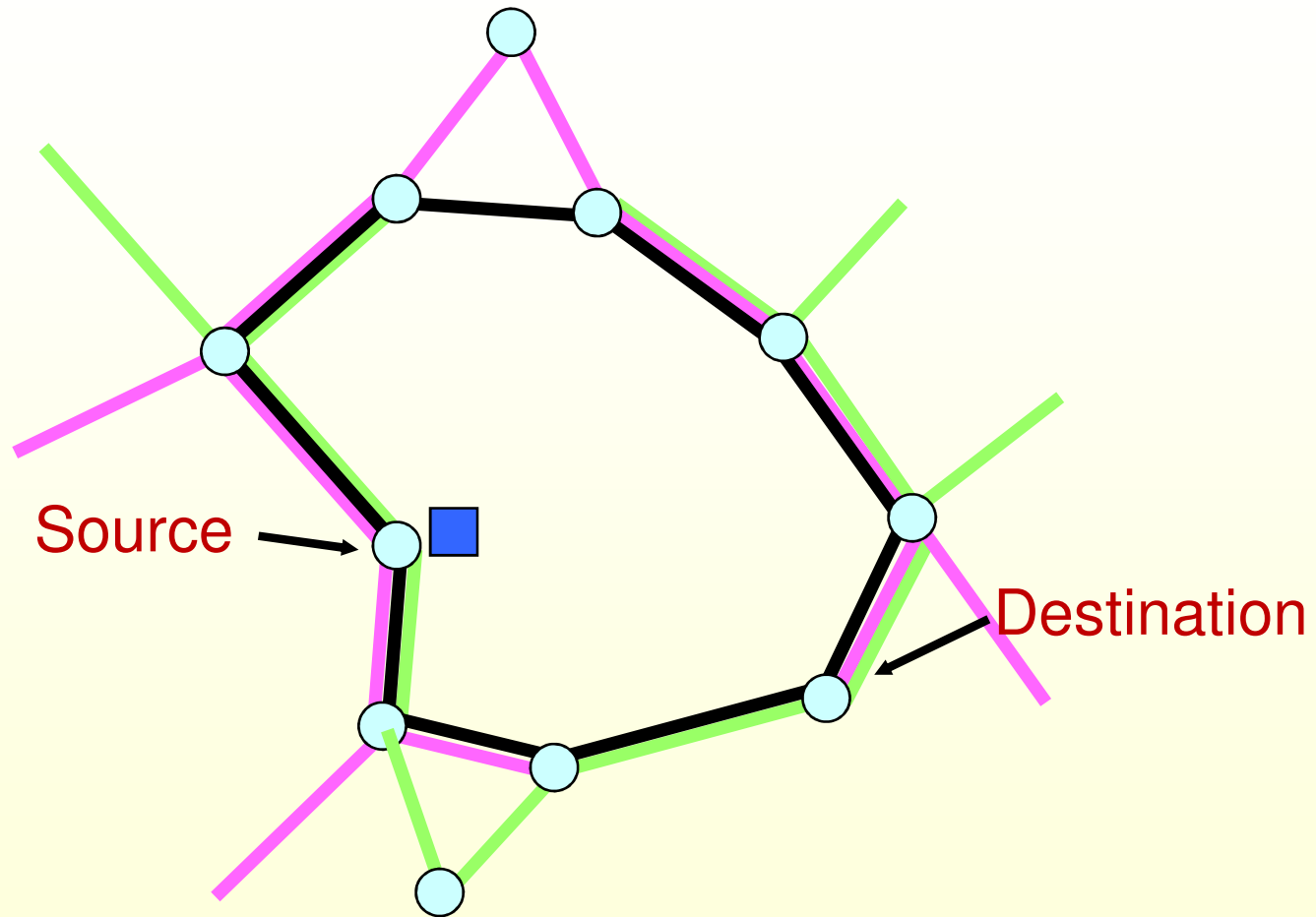
With one tree, may be forced to route in “bad” direction.

Using Multiple Trees



Two extremal-rooted trees are usually sufficient to “approximate” a void

Using Multiple Trees



Pick tree with root closest to the destination

Summary: Routing

- Try greedy forwarding
- Dead end:
 - choose tree
 - record start node
 - traverse subtree
- If possible, revert to greedy forwarding
- Back to start node: packet undeliverable

Theorem

Given a pair of nodes s and t in connected graph G , GDSTR guarantees packet delivery from s to t

Building Hull Trees

- Convex hull info in *keepalive* messages
- Choose roots:
 - minimal and maximal x-coordinates
- Want compact trees
 - minimal hop count from root
- Aggregate convex hulls from leaves to root
- Conflict hull info percolates from root to leaves

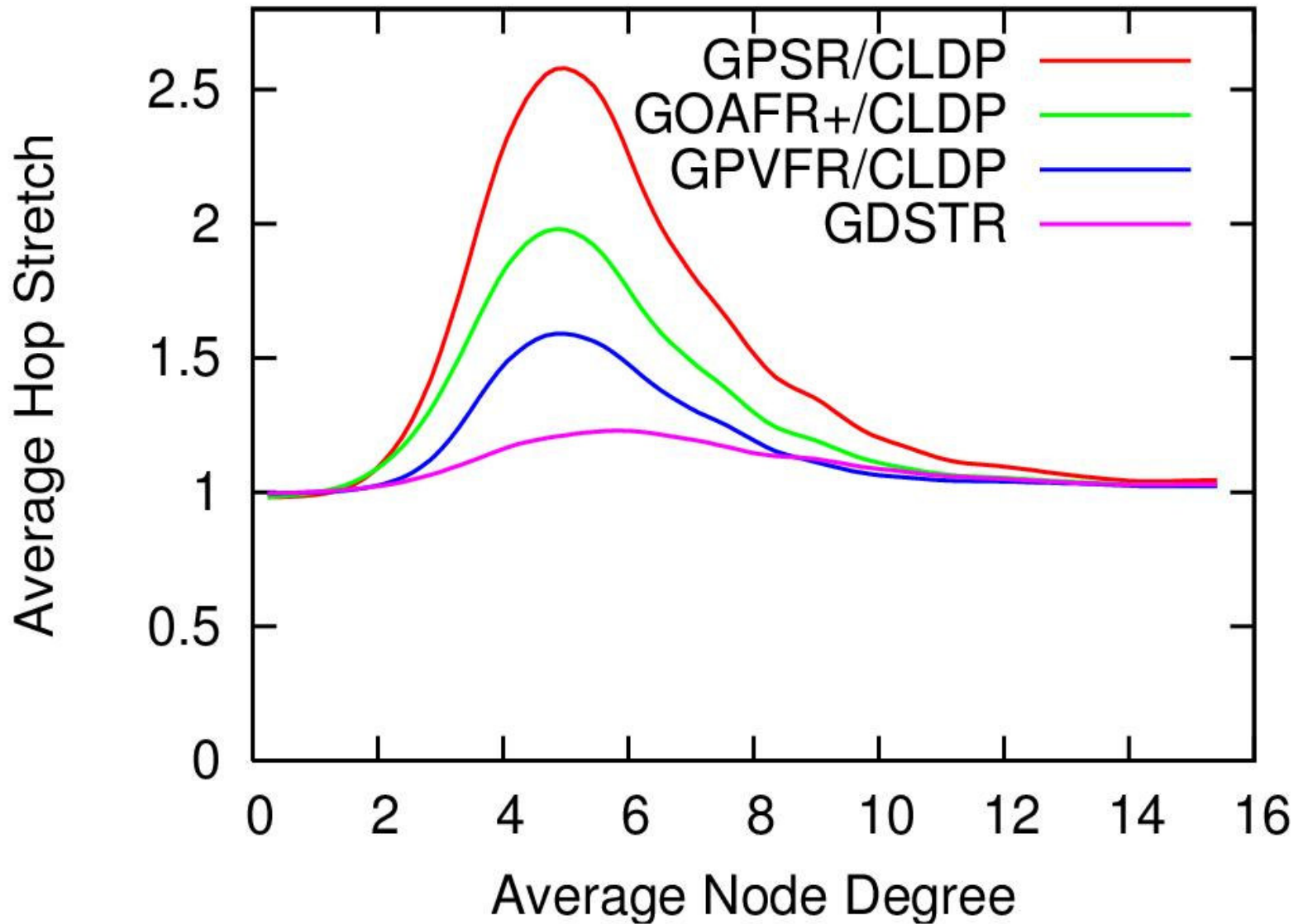
Simulation Results

- Measured 2 routing metrics:
 - Path Stretch
 - Hop Stretch
- Topologies
 - range of network densities
(average node degree)
 - larger networks up to 5,000 nodes
 - low/high density
 - low/high obstacle density

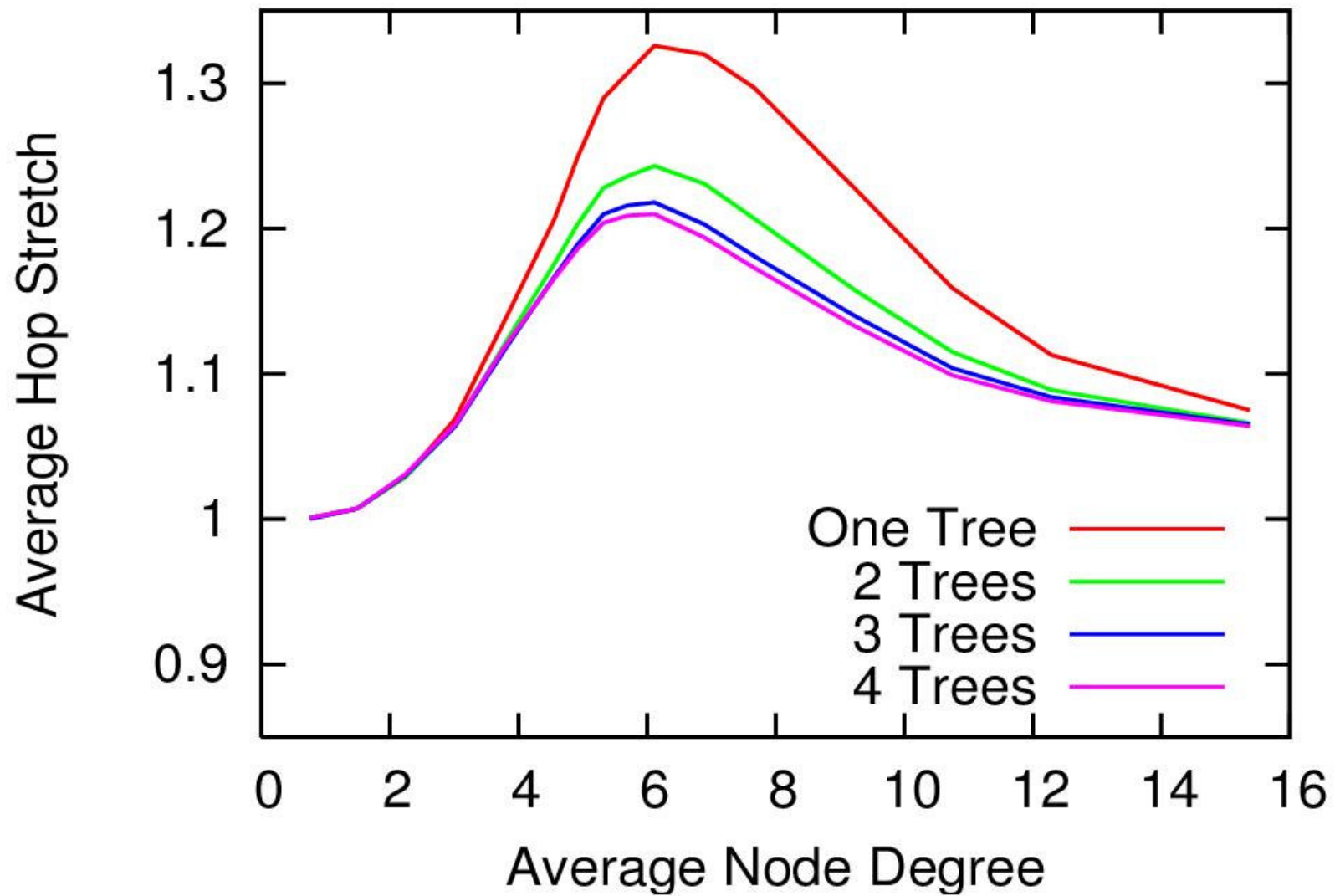
Simulation Results

- Compare with
 - GPSR (Karp, 2001),
 - GOAFR+ (Kuhn, 2003) and
 - GPVFR (Leong et al., 2005)
- under CLDP planarization (Kim et al., 2005)
- Measured costs and compared with CLDP:
 - storage
 - bandwidth

Hop Stretch



Hop Stretch



Costs

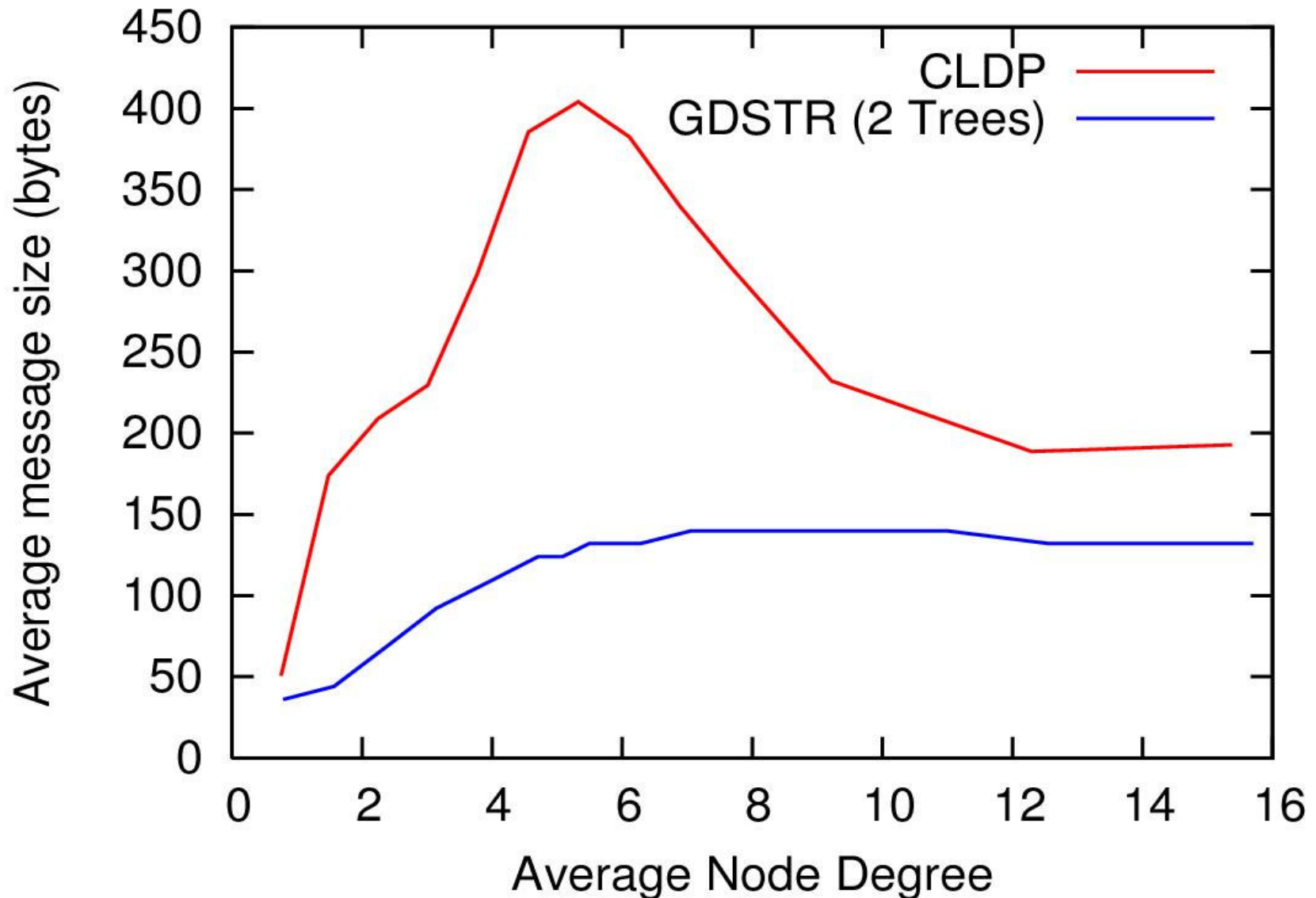
- Computation:

- convex hull computation: $O(\log n)$ operations [Graham's scan]

- Storage: < 1 kb

- Bandwidth

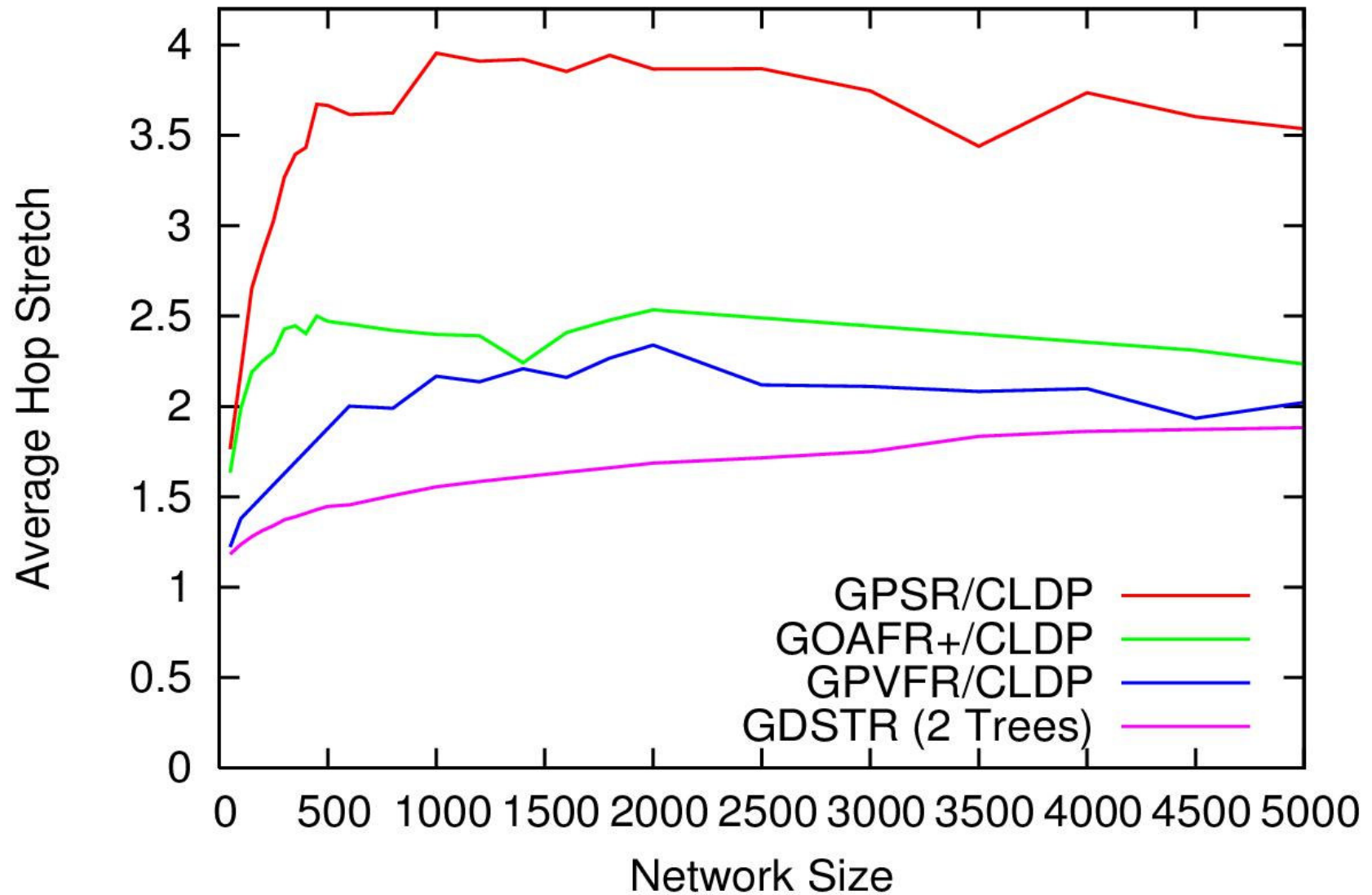
Message Sizes



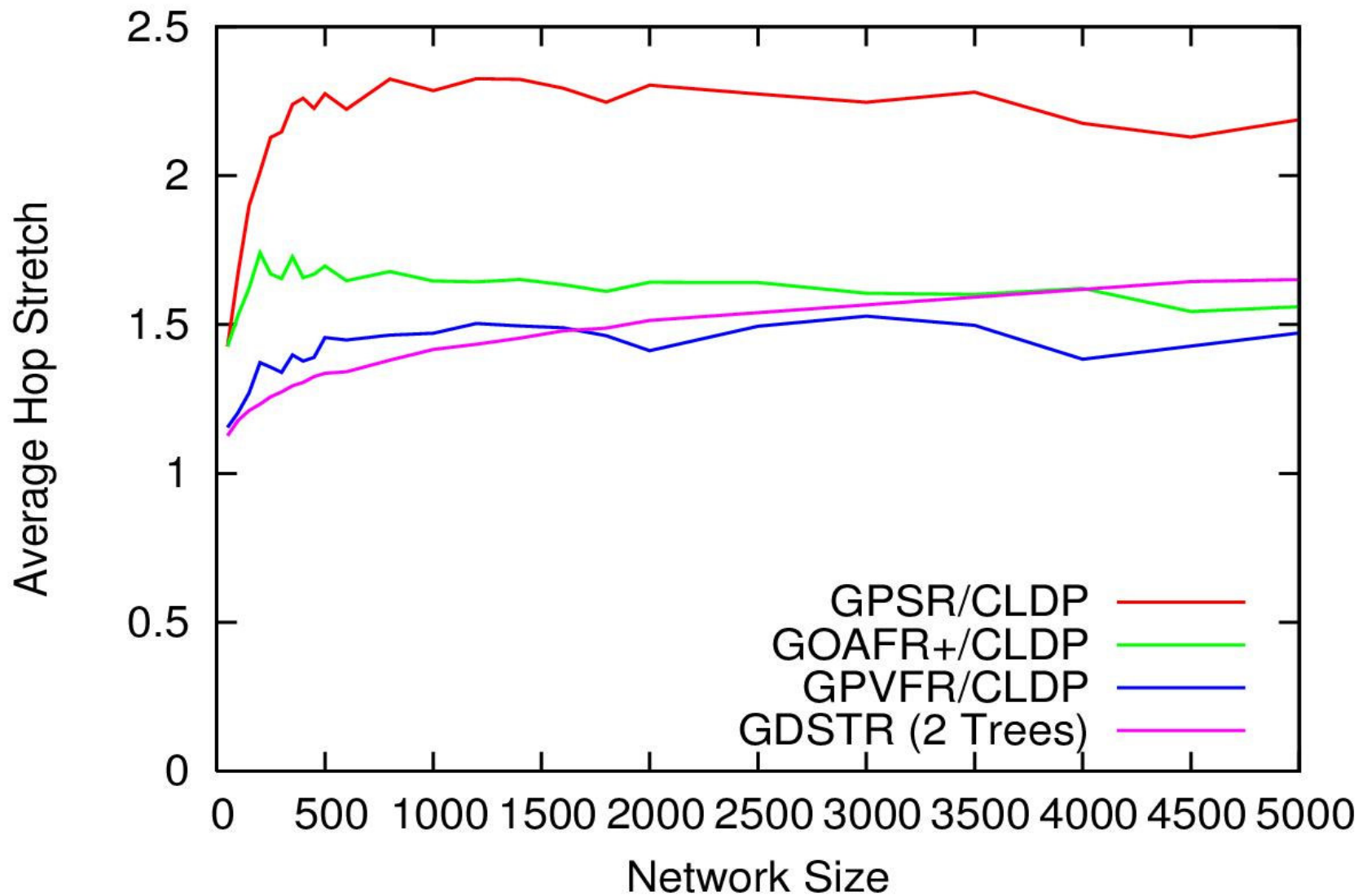
Summary

- Maintenance cost one order of magnitude less than CLDP (face routing)
- Better routing performance (stretch) – up to 20% better

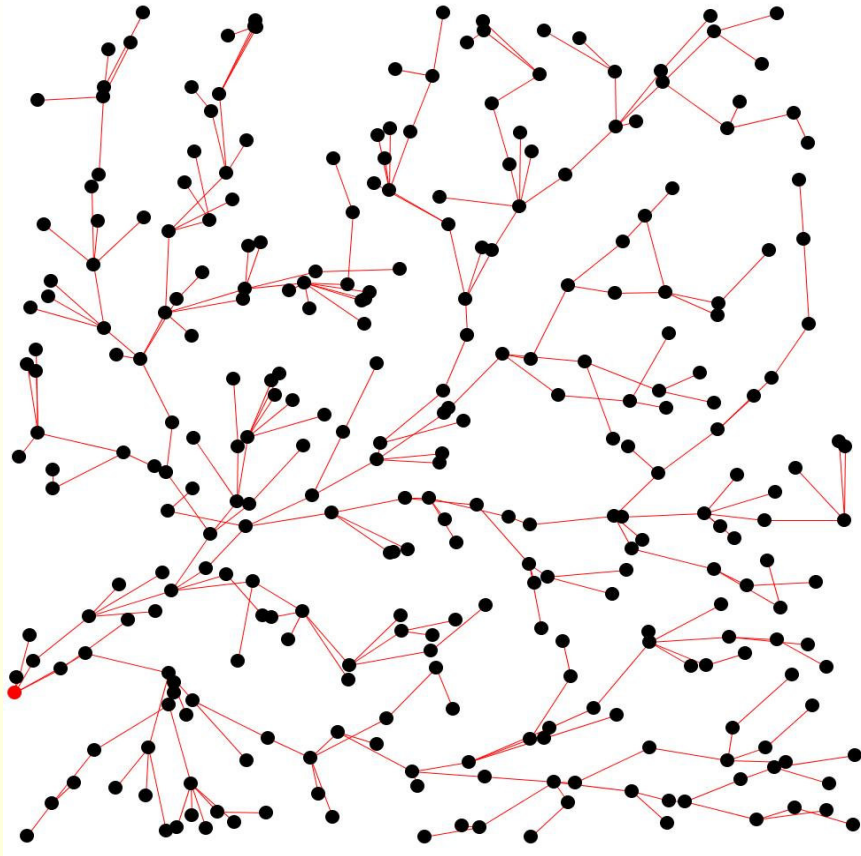
Large Voids



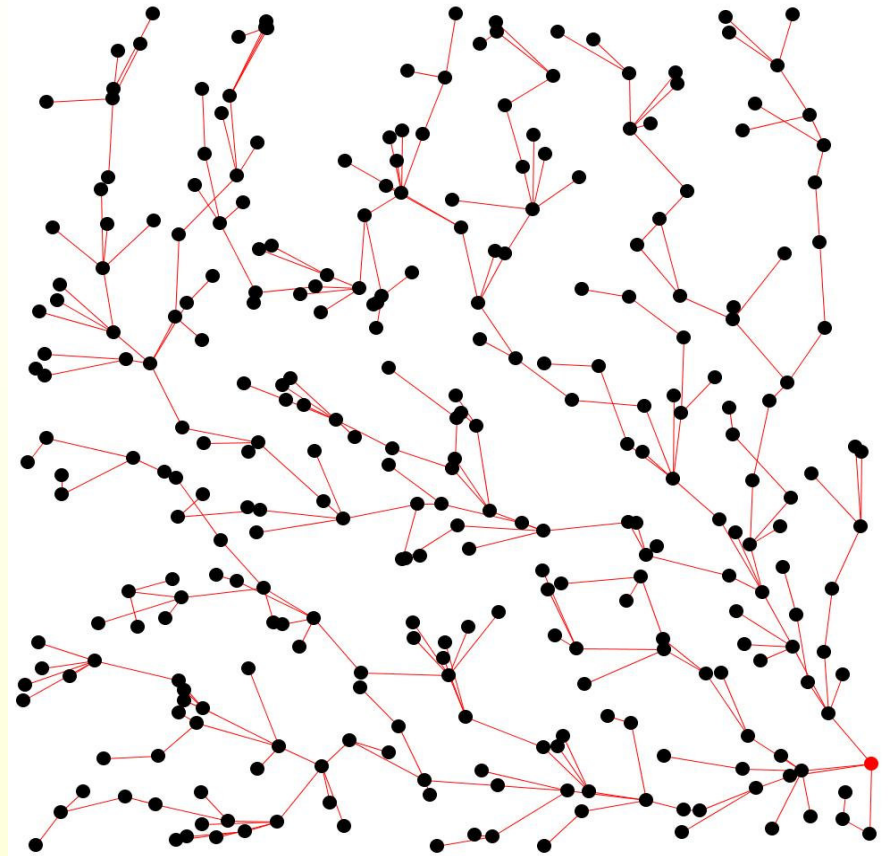
Small Voids



Example GDSTR Hull Trees

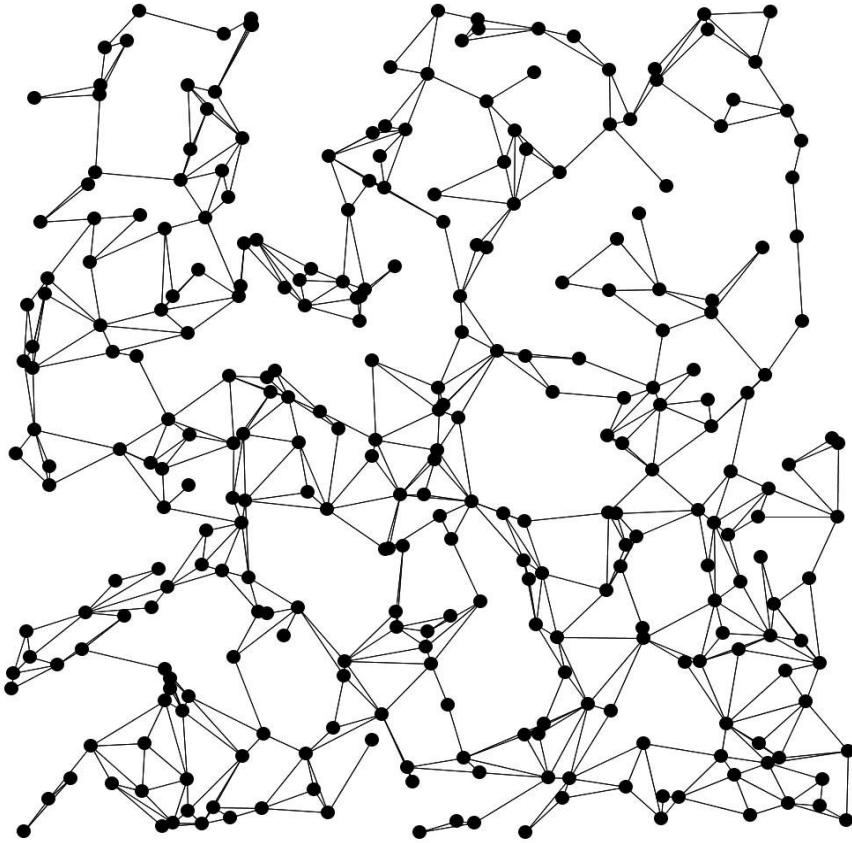


Minimal-x Tree

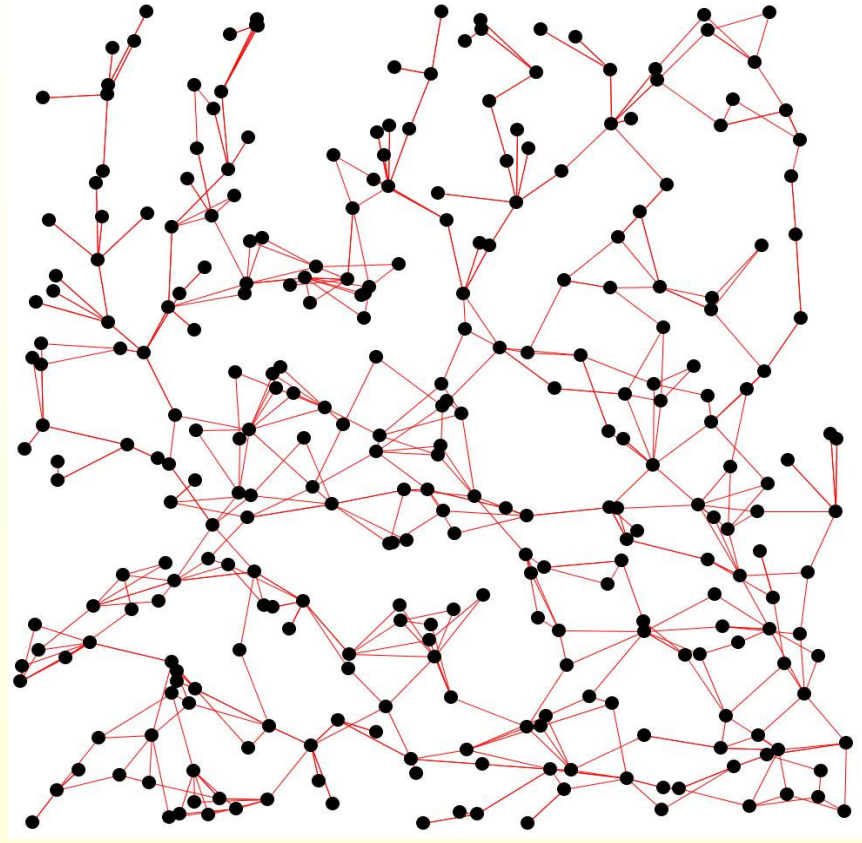


Maximal-x Tree

Comparing Routing Topologies

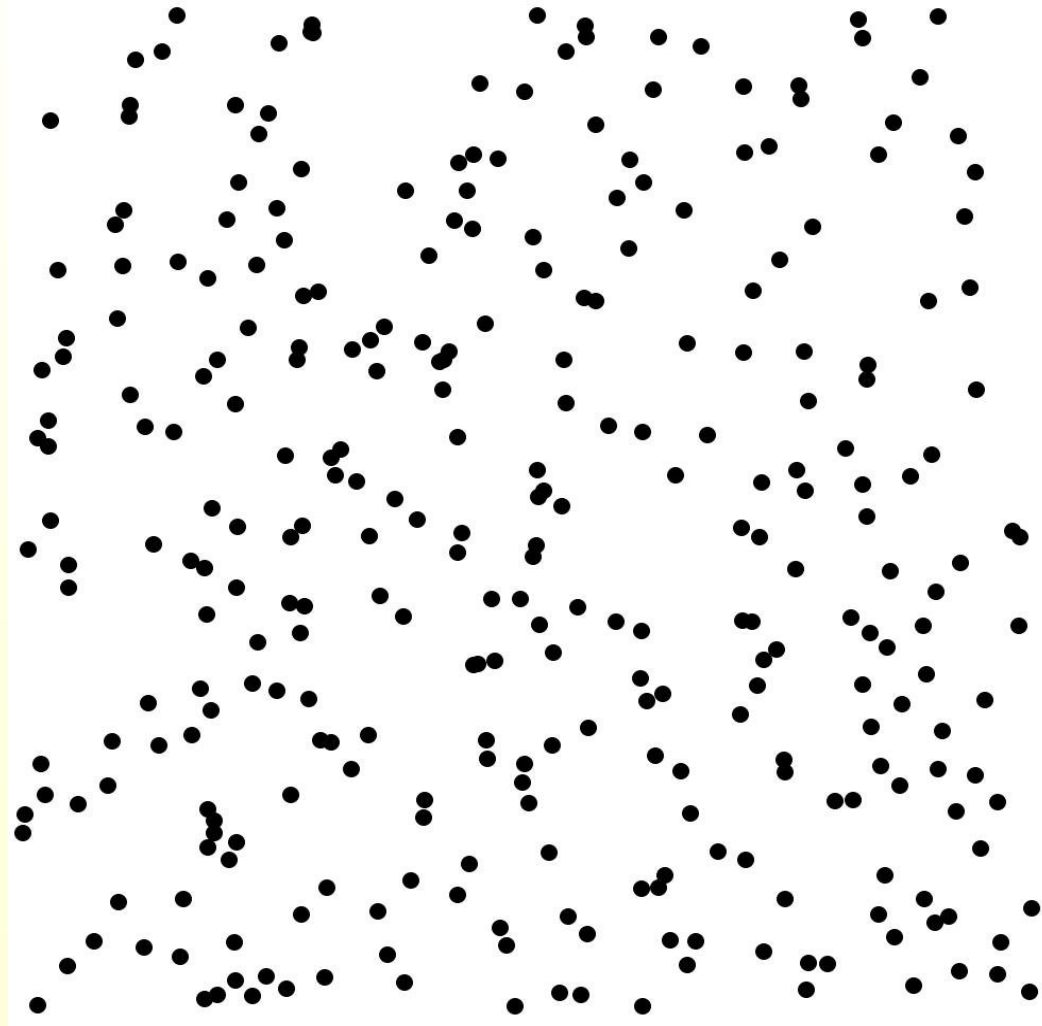


Planar Graph
(CLDP)

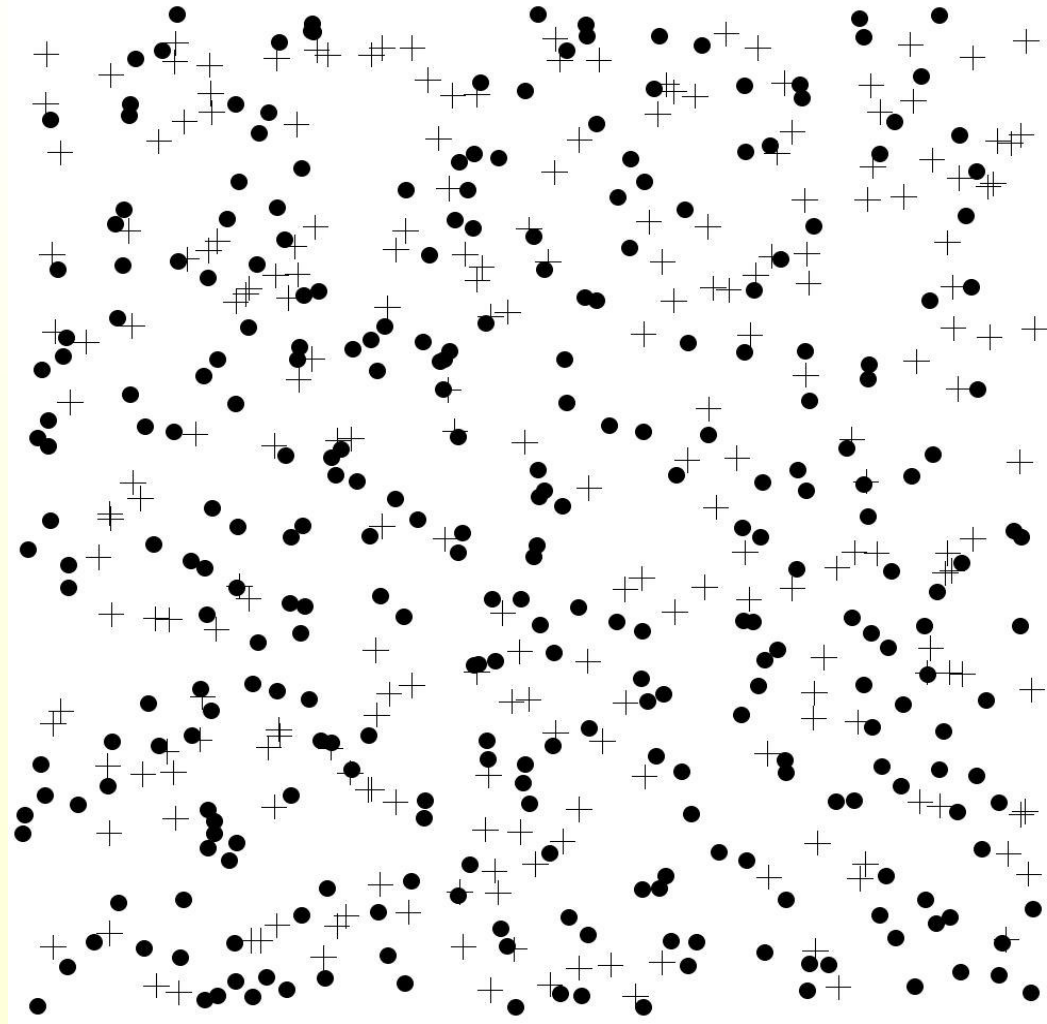


Two Trees

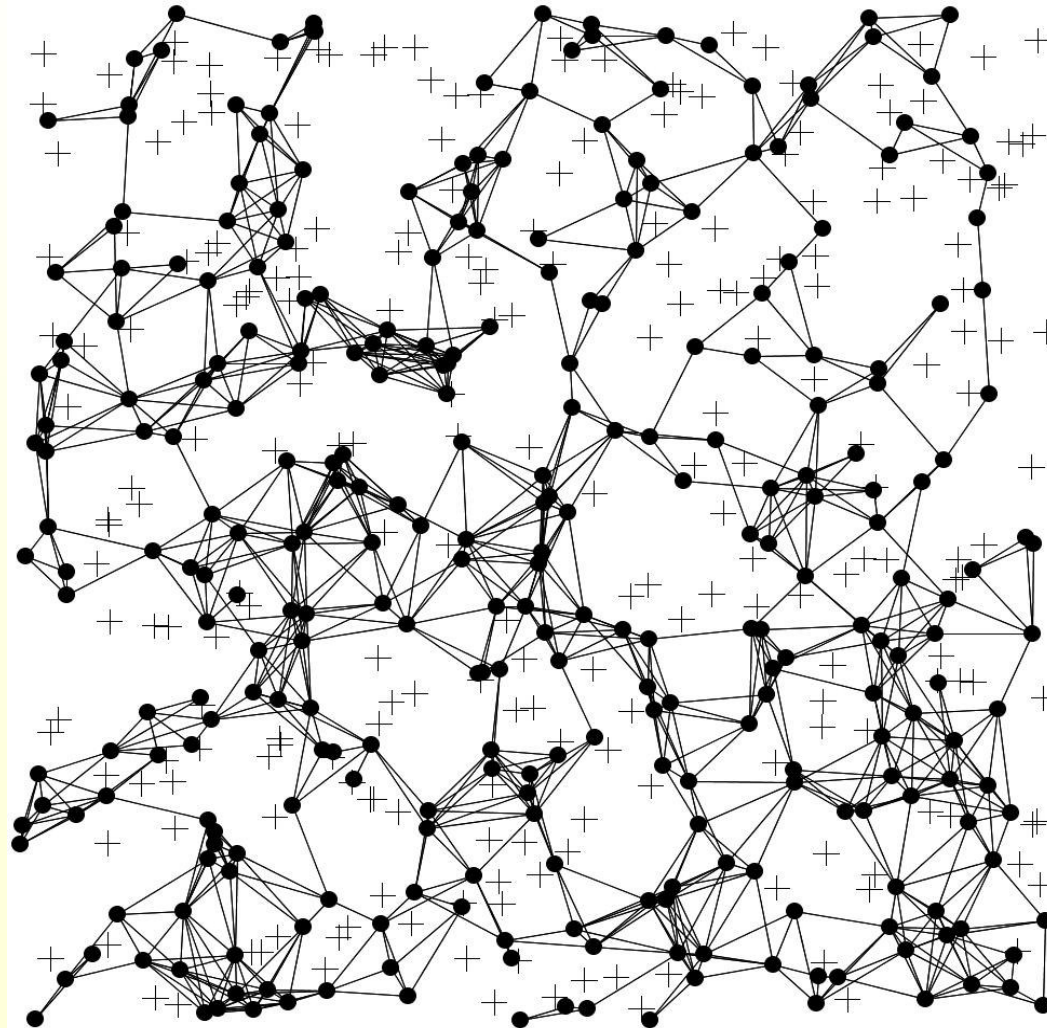
Example- Nodes



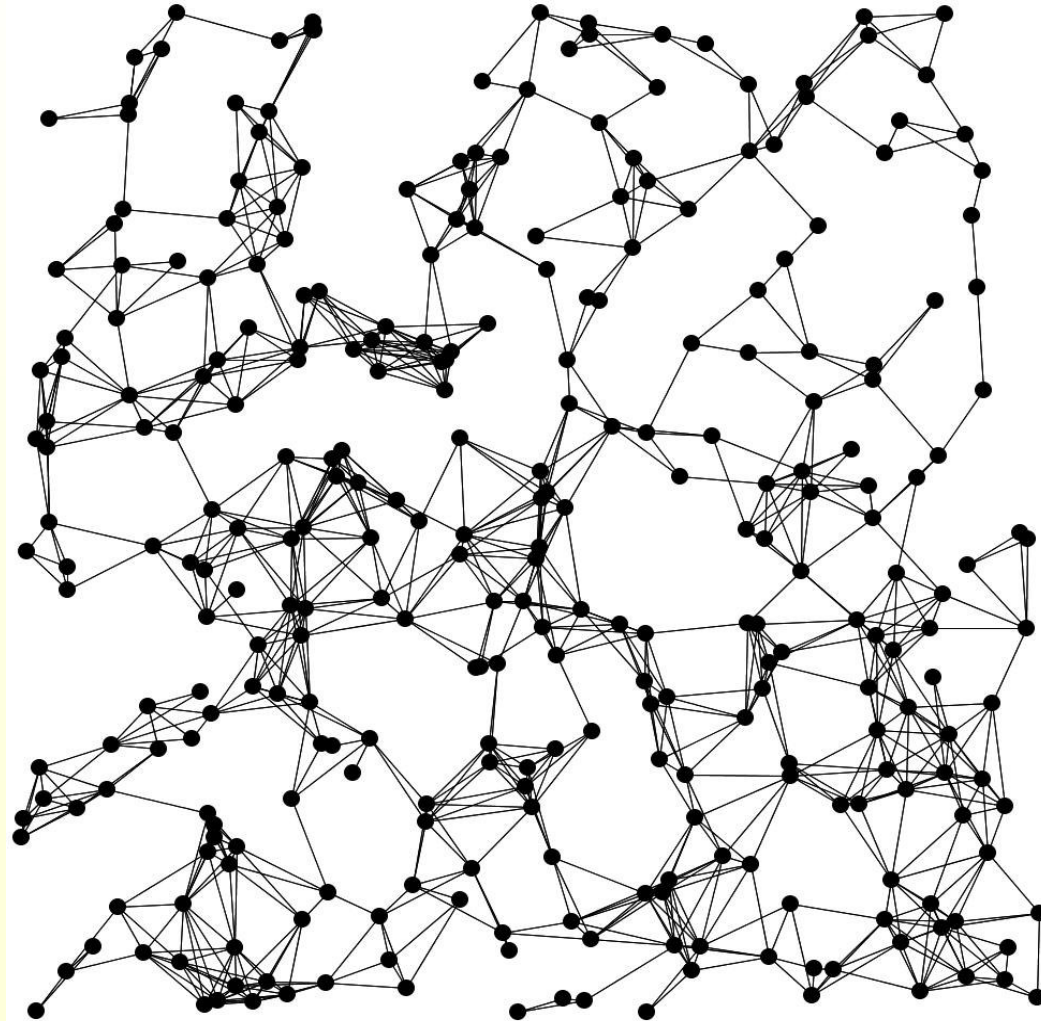
Add Obstacles



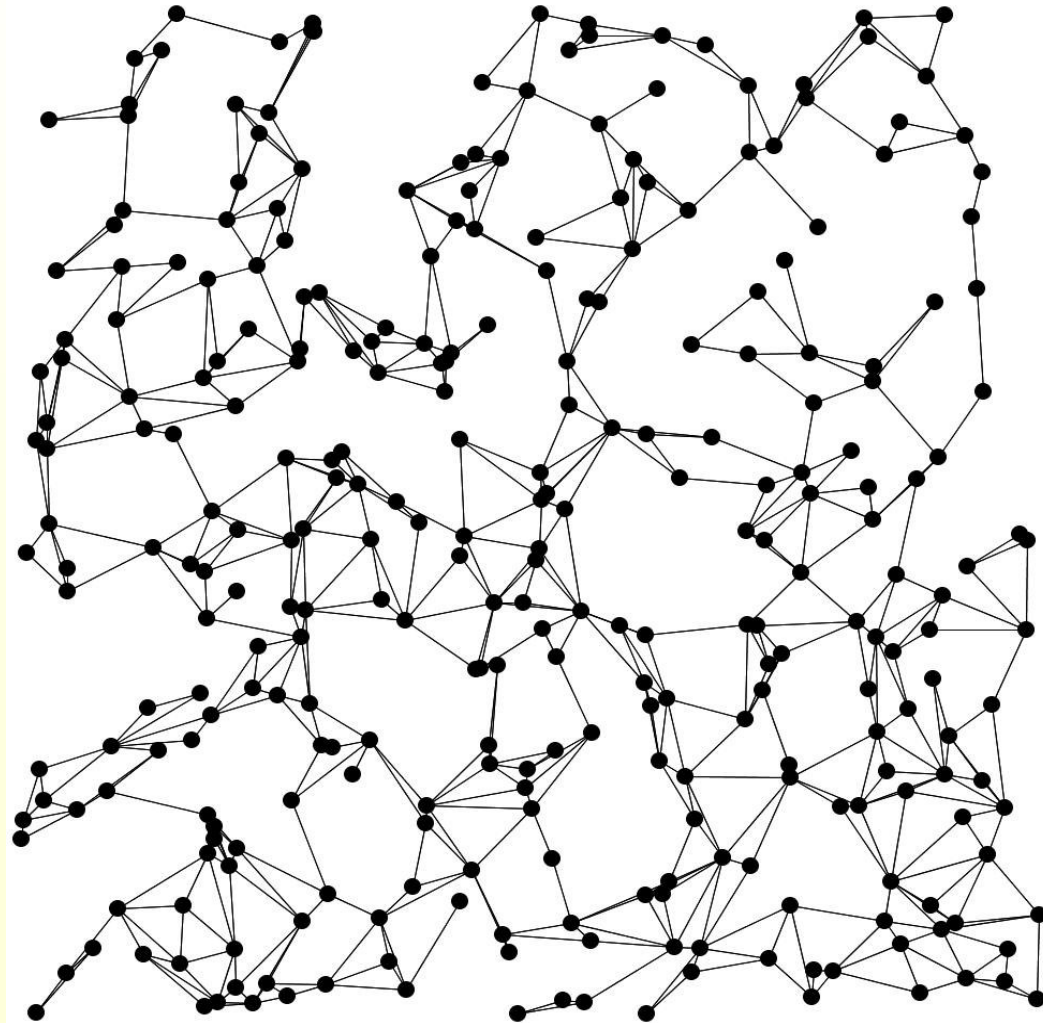
Connectivity



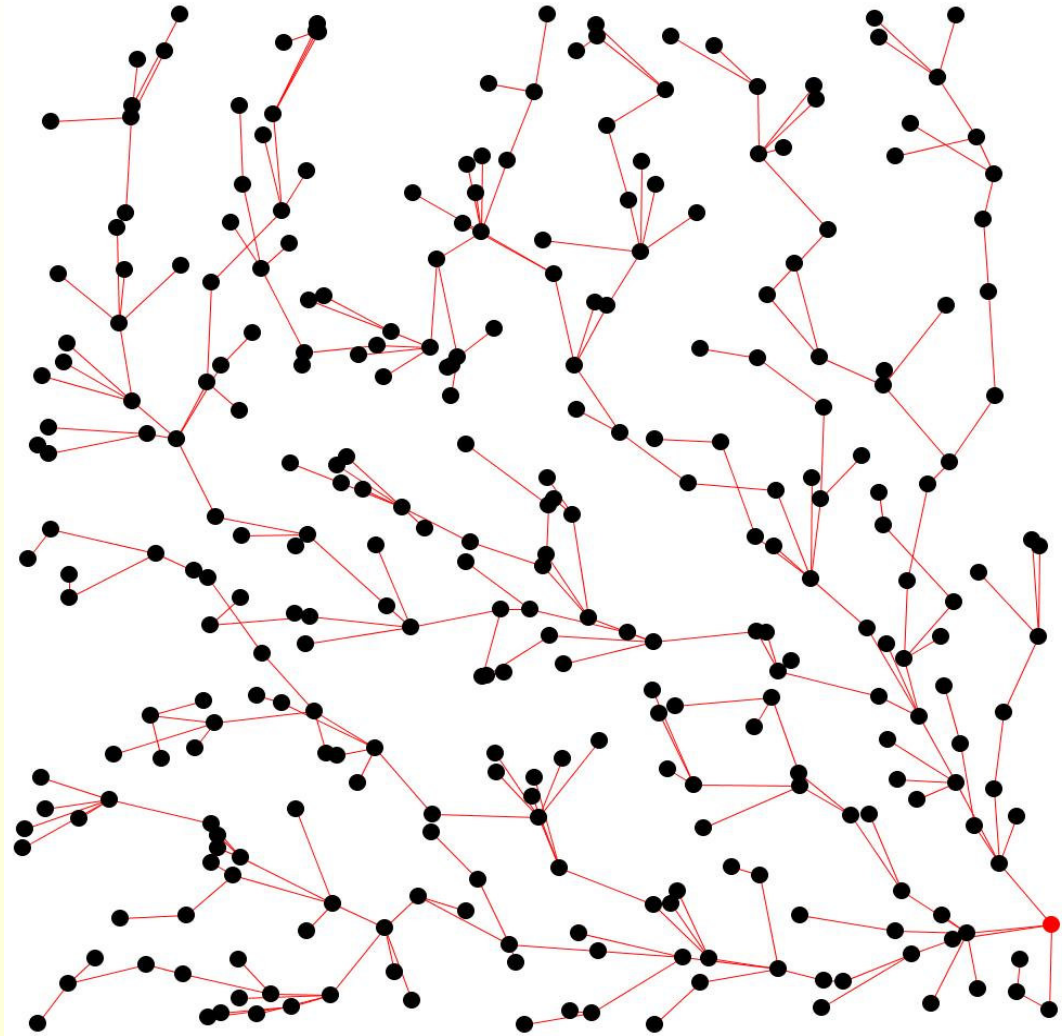
Connectivity



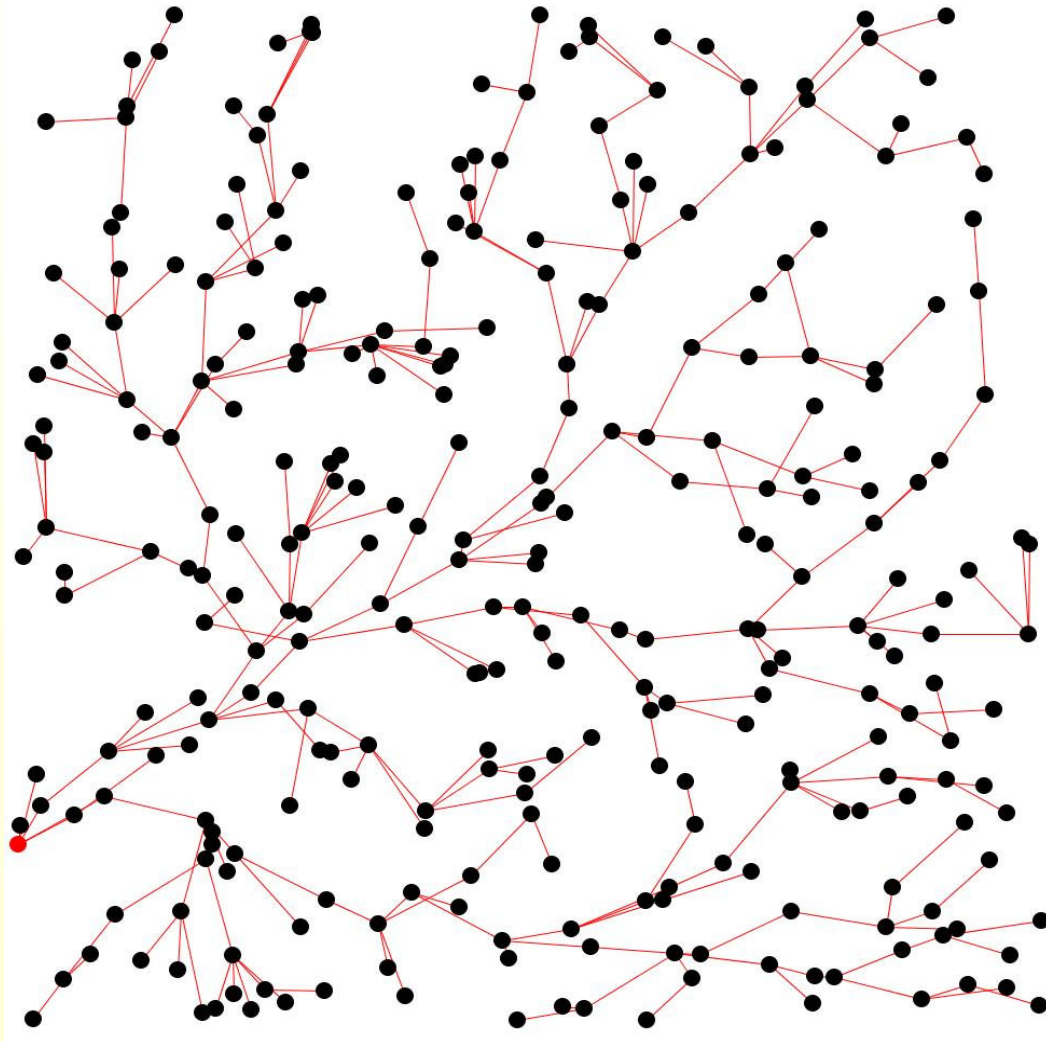
Planarized Graph



One Tree



The Other Tree



GDSTR Summary

- Sparse networks
 - GDSTR chooses correct forwarding direction more often than face routing
- Moderately dense networks
 - Faces are small, forwarding direction is inconsequential
 - Trees do not “approximate” small voids well
- Ultra-dense networks
 - Greedy forwarding works all the time!

Geographic Routing

Conclusions

- Geographical forwarding is quite **robust** to localization errors, or reasonable virtual coordinates
- Geographical forwarding can easily **scale** to tens of thousands of nodes with acceptable overhead
- For dense uniform sensor layout, we can eliminate the need for face routing altogether
- Rubber band virtual coordinates respect the connectivity better than the true coordinates
- CLDP, GDSTR are viable practical algorithms

The End