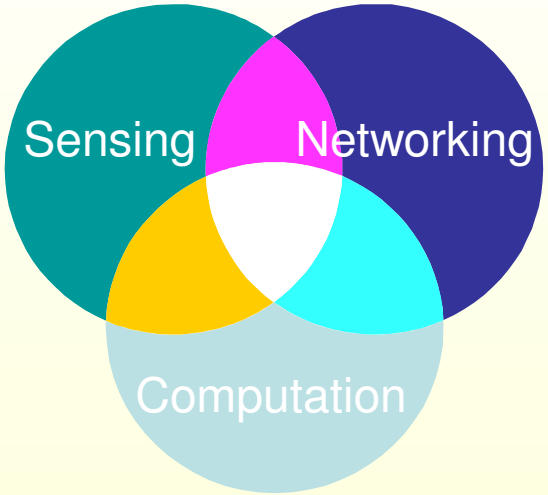
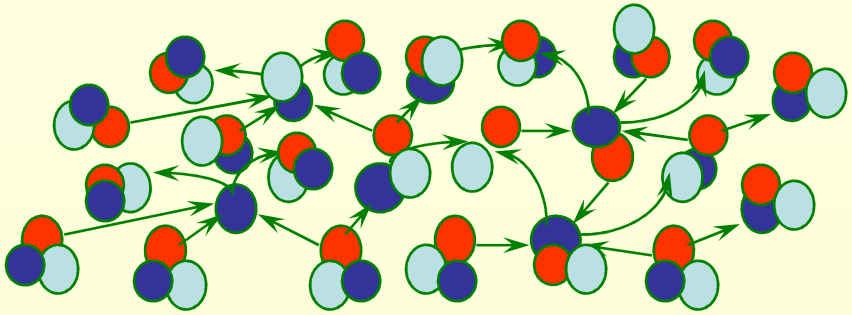
CS321: Networking Sensors I: Geographic Routing



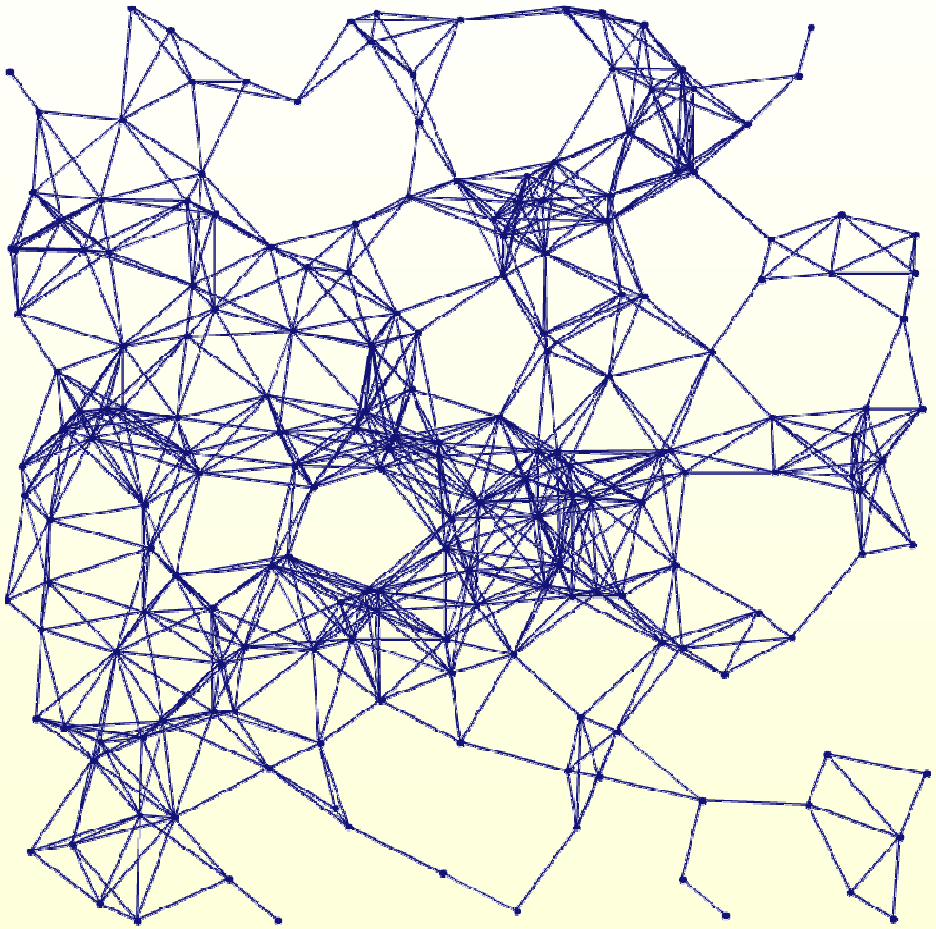
Leonidas Guibas
Computer Science Dept.
Stanford University



Networking Sensor Nodes

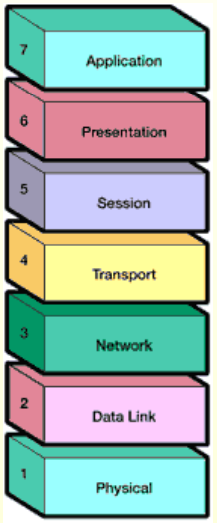
Networking Sensors

- Networking is a crucial capability for sensor networks -- networking allows:
  - Synchronization and localization
  - Placement of sensors close to signal sources
  - Collaborative information processing
- But it is also one of the most demanding:
  - radio communication consumes the most energy
  - nodes, and especially links, can be unreliable and unstable; this must be mitigated by the network protocol stack



A Hierarchy of Issues

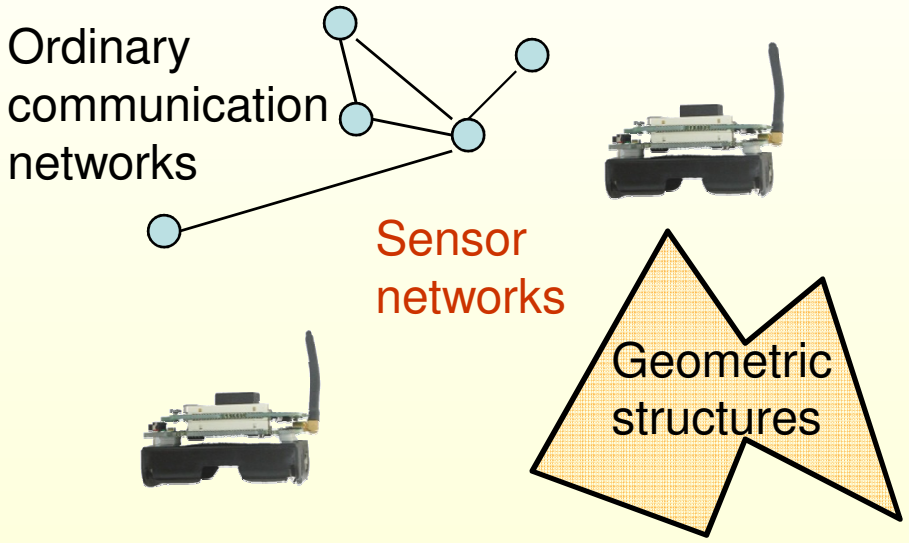
- Low-level issues regarding the network protocol stack (physical, MAC layers).
  - Medium access
  - Collision avoidance
- A lot of work especially on the MAC (medium access control) layer.
- High-level routing and transport layer issues.
  - Route discovery
  - End-to-end reliable transport
- Large active research area.



A Dilemma: Living Between Two Worlds

- Nodes are embedded in a physical space. Should we adopt the naming and routing structures already available in the host space?
- Or should we invent a space that better reflects the true network topology, and use that instead?

- What if our sampling is bad?

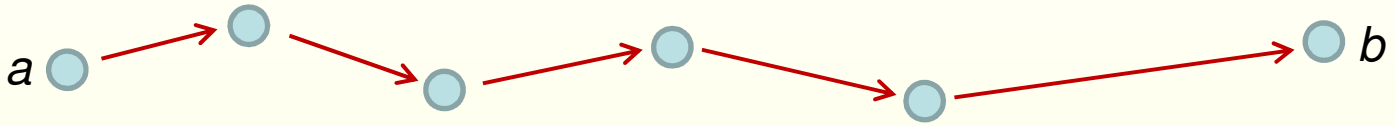


- What if the network is volatile?

Point-to-Point Routing

Routing in Ad Hoc Networks

- Obtain route information about pairs of nodes wishing to communicate.



- Proactive protocols: maintain routing tables at each node that are updated as changes in the network topology are detected.
  - Heavy overhead with high network dynamics (caused by link/node failures or node movement).
  - Mostly not practical for ad hoc networks, at least at the level of maintaining per node information.

Routing in Ad Hoc Networks

- Reactive protocols: routes are constructed on demand. No global routing tables are maintained.
- Due to the high rate of topology changes, reactive protocols are more appropriate for ad hoc networks.
  - Ad hoc on demand distance vector routing (AODV, Perkins and Royer ‘99)
  - Dynamic source routing (DSR, Johnson ‘94)
- However, both depend on expensive flooding for route discovery.

Routing Desiderata

- Guaranteed delivery
- Good path quality
- Energy awareness
- Robustness to low-level link volatility

Measures of Path Quality

- First and foremost, a protocol should guarantee packed delivery, whenever such delivery is possible
- Second, the quality of the path produced should be good when compared to the optimal path available. Different path costs can be used:

$$c(\pi) = \sum_{e \in \pi} l^d(e),$$

$d = 0,1,2,3,4,...$

d = 0, hop length
d = 1, normal path length
d = 2, 3, 4 ..., energy costs

- These can be made roughly equivalent by assuming a constant node density or a minimum node spacing
  - This can be attained by a node clustering process that guarantees some minimum spacing between cluster heads

Geographic Routing and Variations

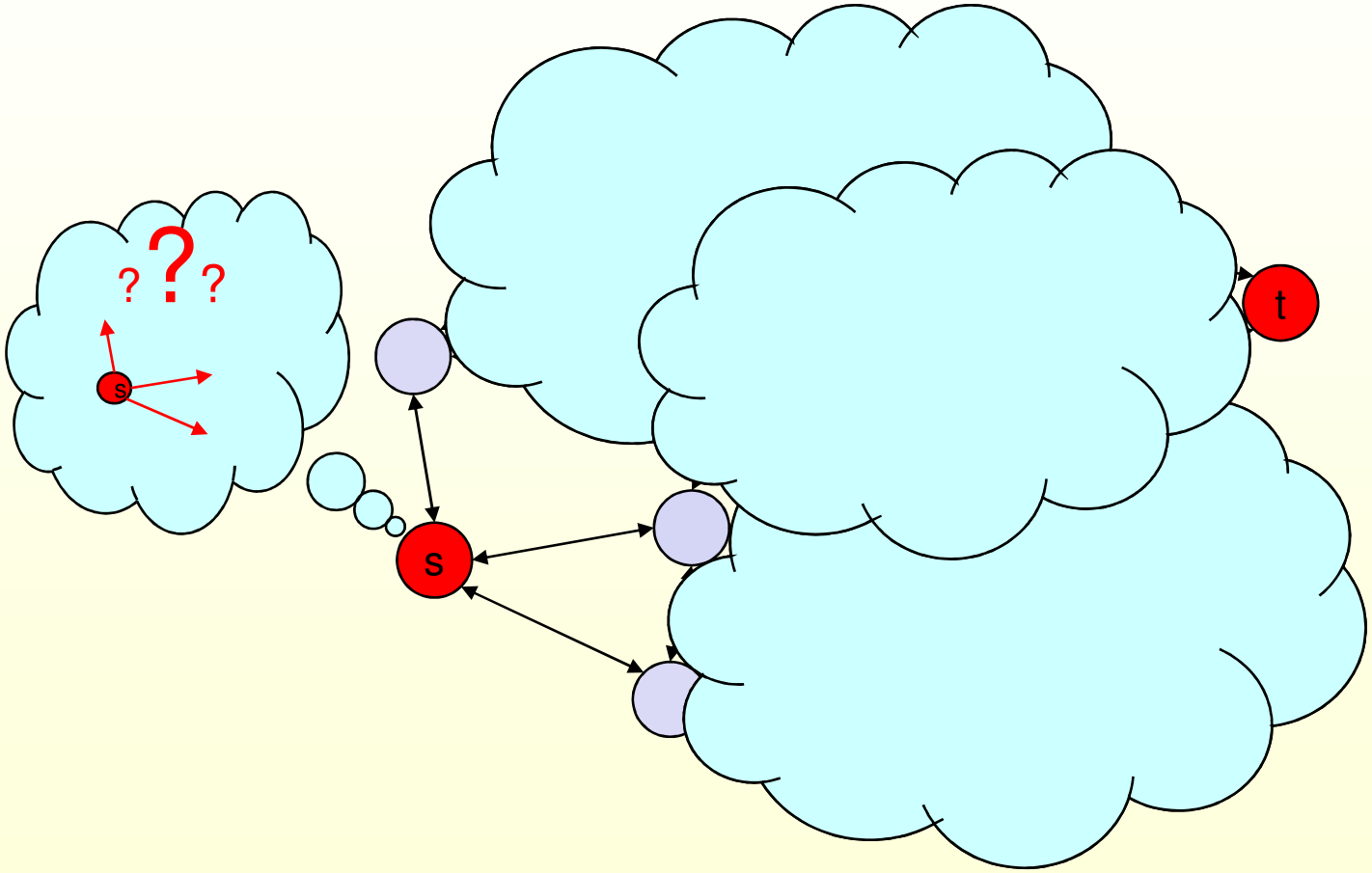
Geographic Routing

- "Data-centric" routing: routing is frequently based on a nodes' attributes and sensed data, rather than on pre-assigned network addresses.
- Geographic routing uses a node's location to discover a path to that node.

Geographic Routing

- Assumptions:
  - Nodes know their geographical location
  - Nodes know their 1-hop neighbors (but have no global routing tables)
  - Routing destinations are specified geographically (a location, or a geographical region)
  - Each packet can hold a small amount ($O(1)$) of routing information.
  - The connectivity graph is modeled as a unit disk graph (UDG, or qUDG).

Geographic Routing

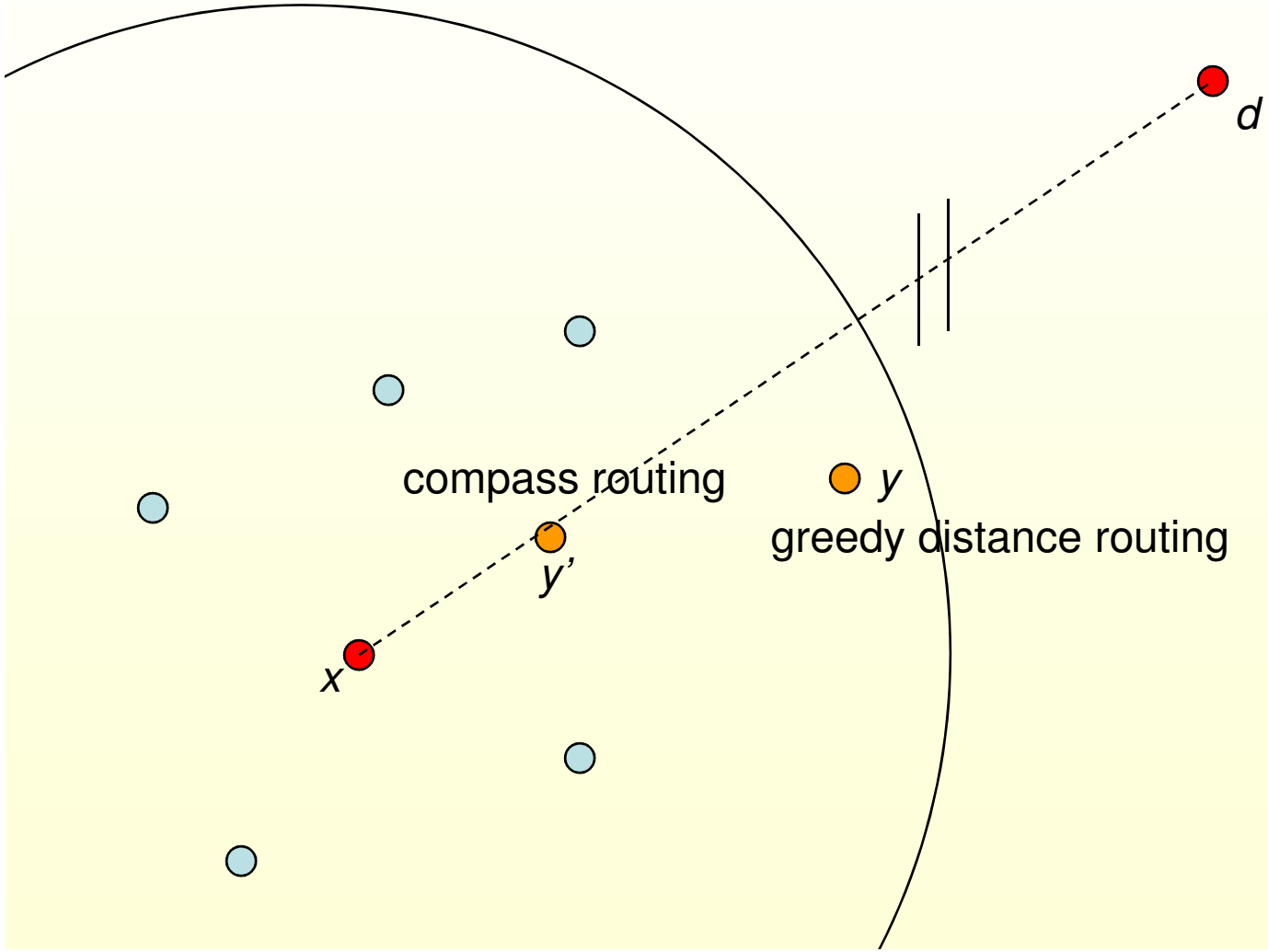


[From Kuhn et. al. ’03]

Geographic Routing

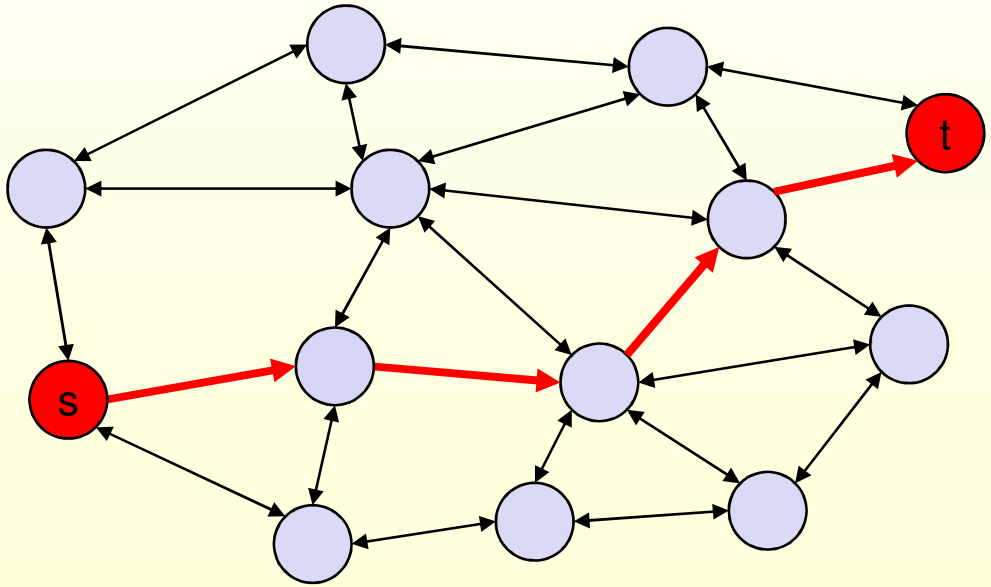
- The source node knows
  - The location of the destination node;
  - The location of itself and its 1-hop neighbors.
- Geographic forwarding, or greedy geographic routing: send the packet to the 1-hop neighbor that makes most progress towards the destination.
  - No flooding is involved.
- Many ways to measure “progress”.
  - The one closest to the destination in Euclidean distance: “distance routing”.
  - The one with smallest angle towards the destination: “compass routing”.

Neighbor Choice



Greedy Geographic Routing

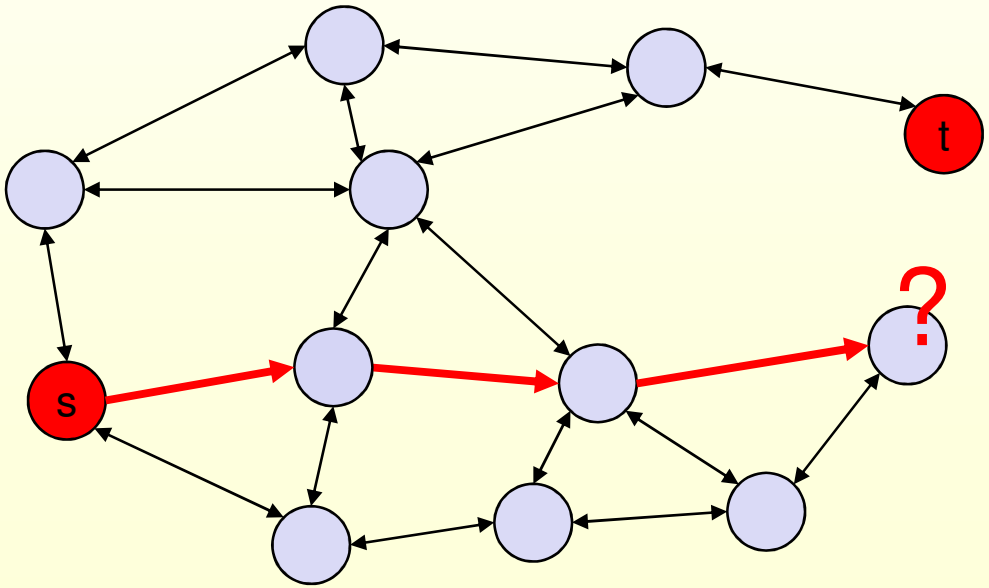
- Each node forwards message to its “best” neighbor



[From Kuhn et. al. ’03]

Greedy Geographic Routing

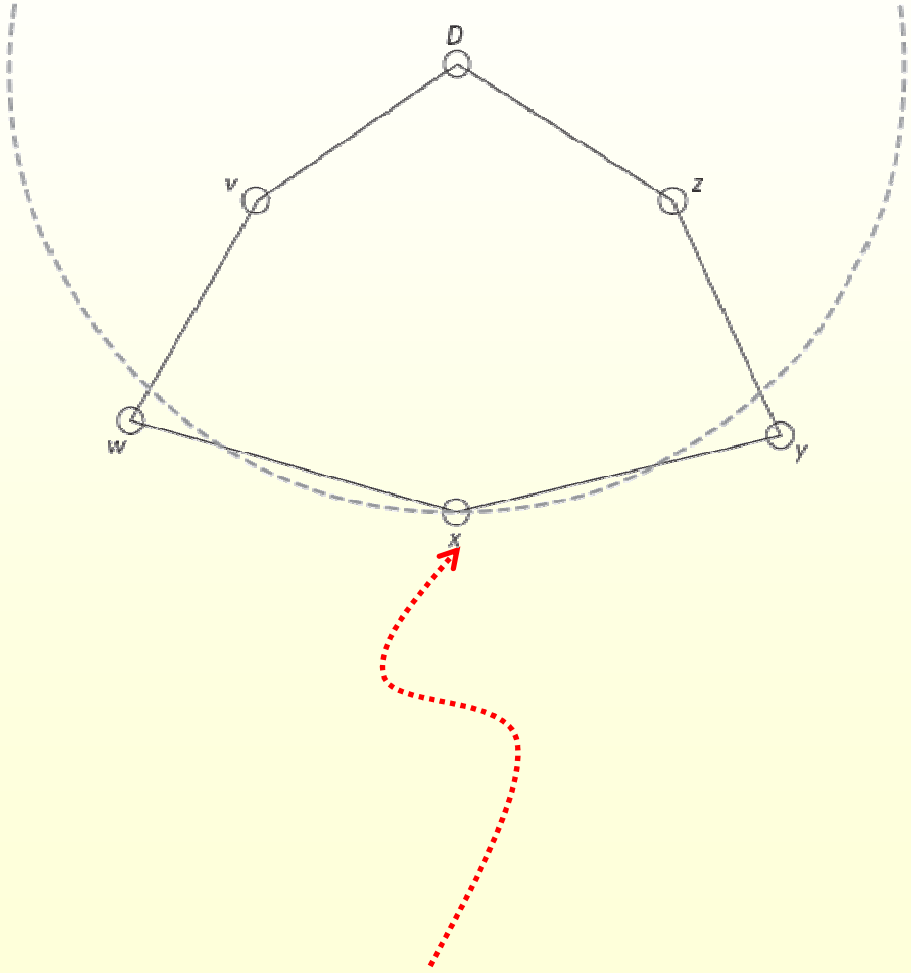
- Each node forwards message to its “best” neighbor



[From Kuhn et. al. ’03]

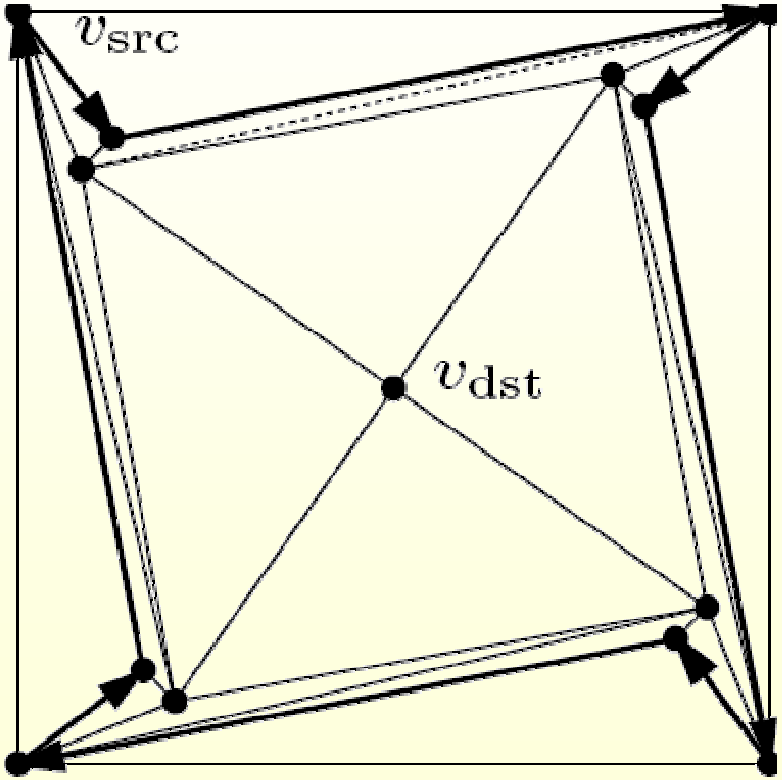
Greedy Protocols Can Get Stuck

- The intermediate node *x* can be a local optimum towards the destination
- In general, local optima will arise if the node graph contains "holes" – areas with no sensor nodes
- To prove that such situations cannot happen we need to assume special properties about the connectivity graph *G*



Compass Routing May Get Stuck in Loops

- Compass routing may get in a loop.



Send packets to the neighbor with smallest angle towards the destination

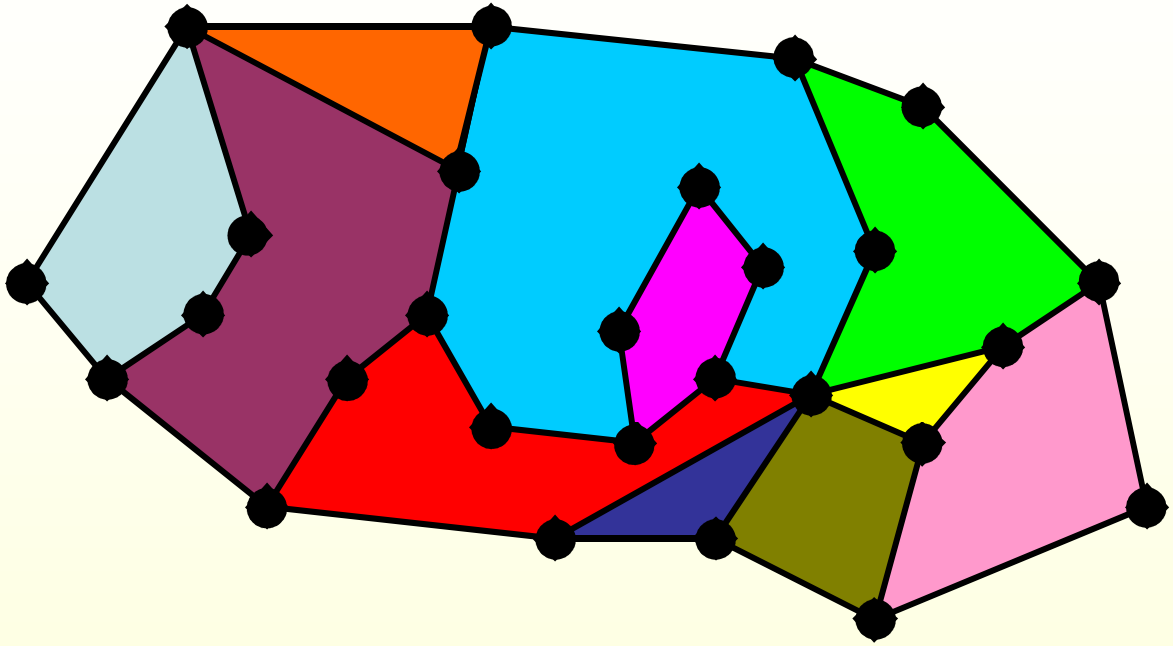
Still ...

- Geographic routing is attractive:
  - very simple algorithm
  - very small state per node and packet
  - local, greedy method
  - still works if some links or nodes fail
- How can we fix the local minima problem?

  Planarize the communication graph!

Graph Planarization

Planarizations



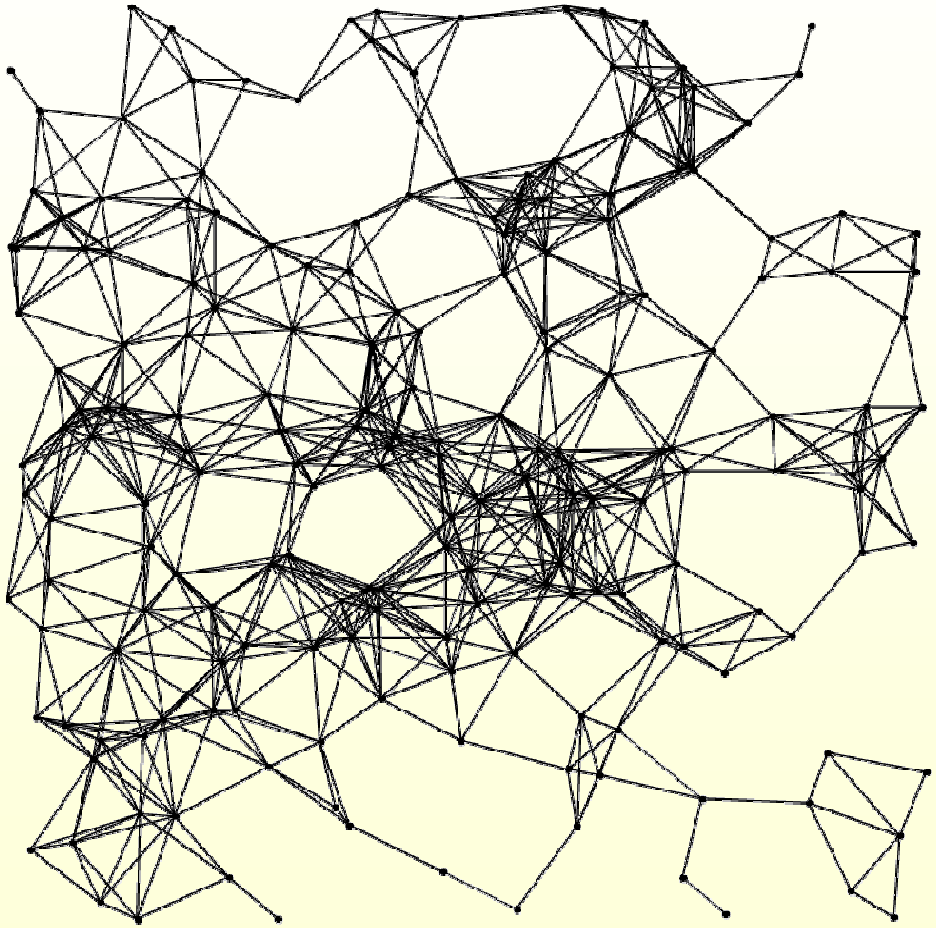
A planar straight-line graph has no crossing edges. It subdivides the plane into regions called faces.

As we will see, walking around its faces provides another way to navigate the graph

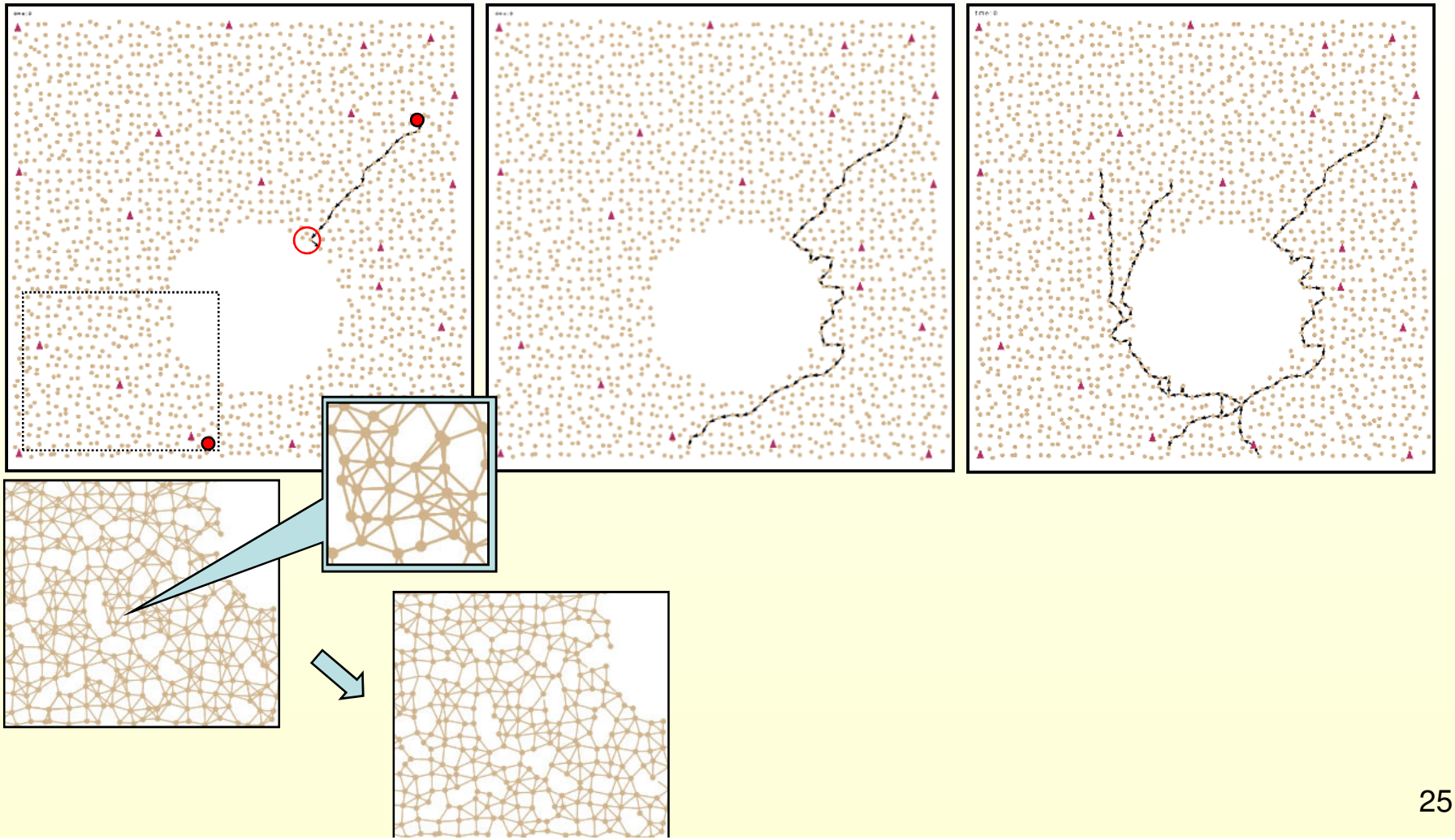
Could we improve routing performance by disabling certain 1-hop communications?

An Example of Self-Crossings in a UDG

200 nodes randomly deployed in a 2000×2000 meters region. Radio range =250 meters



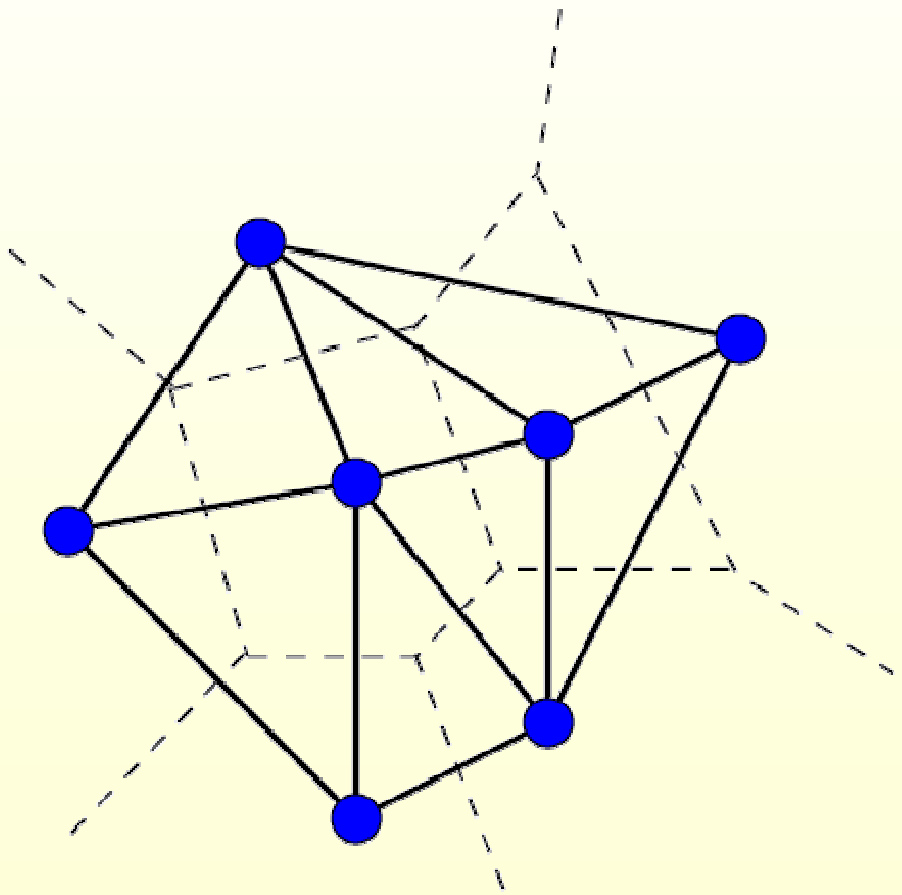
Planarization: Edge Removal



Some Geometric Graphs, Known to be Planar

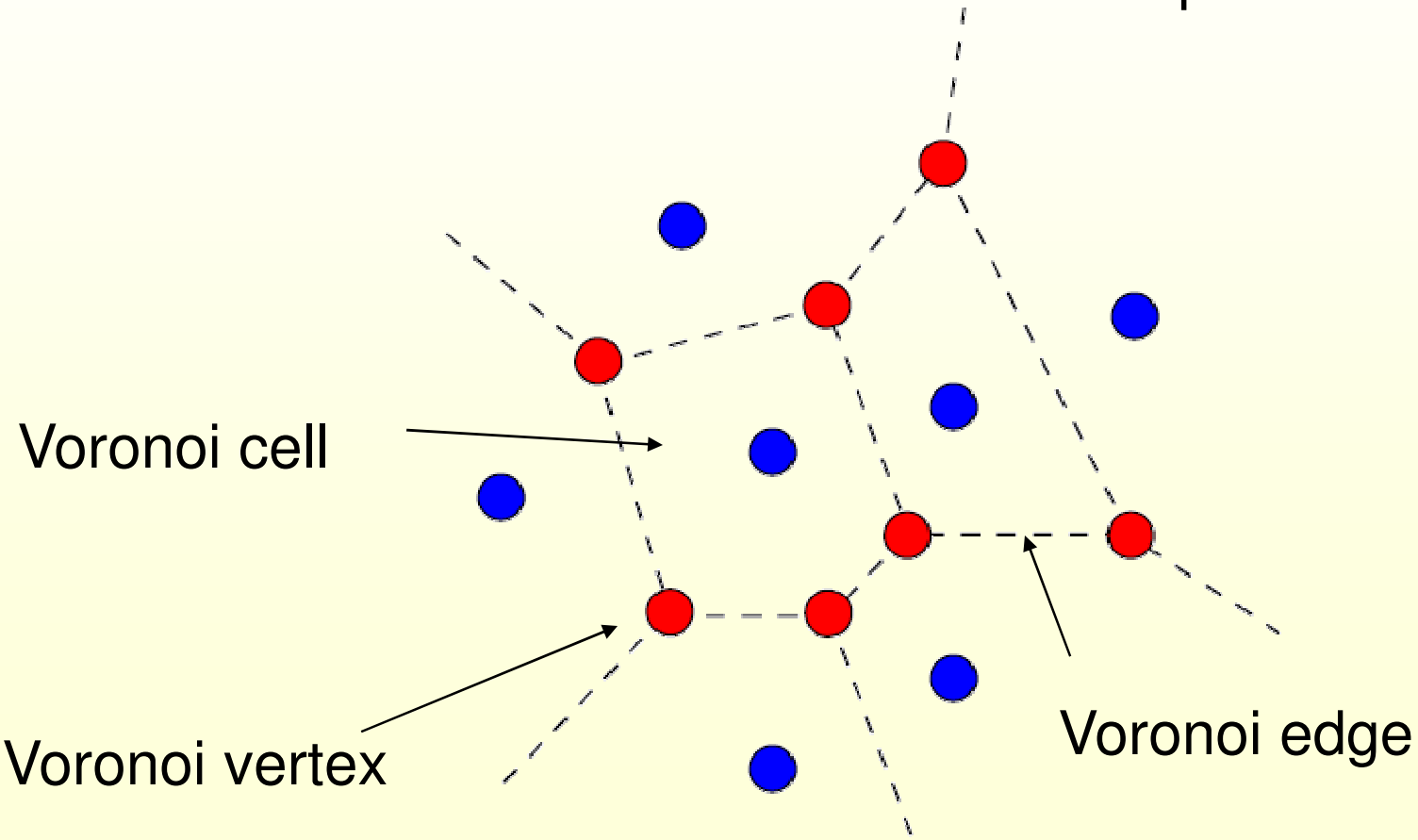
Delaunay Triangulation

- First proposed by B. Delaunay in 1934.
- Numerous applications since then.



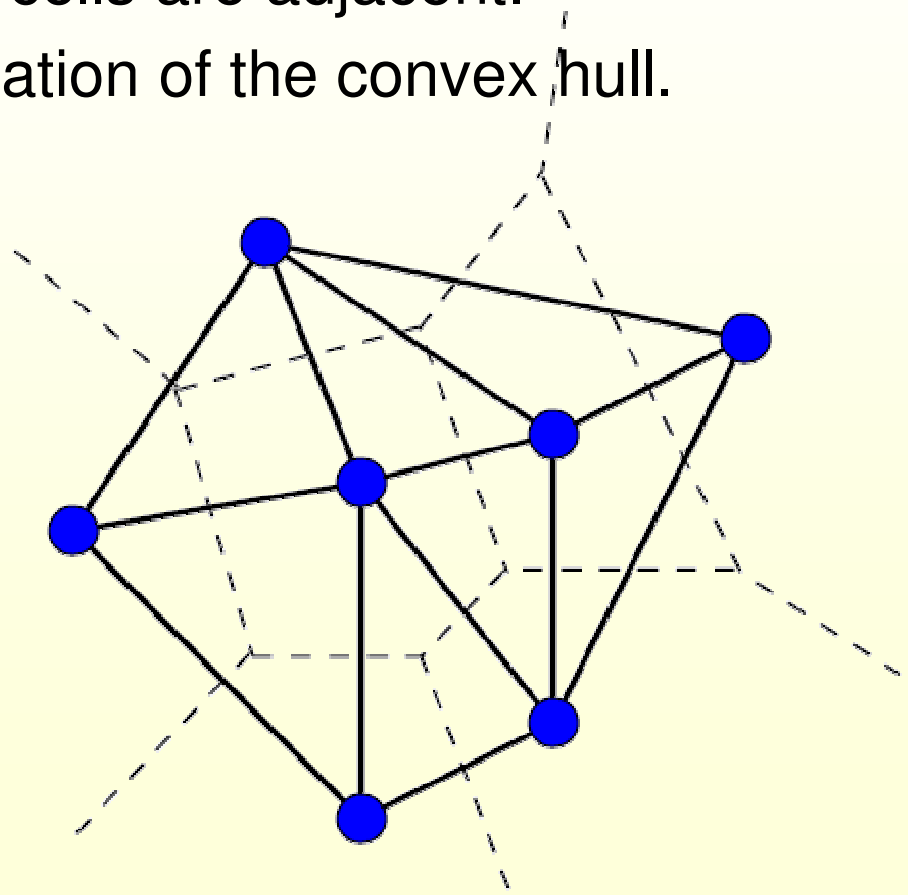
Voronoi Diagram

- Partition the plane into cells such that all the points inside a cell have the same closest point.



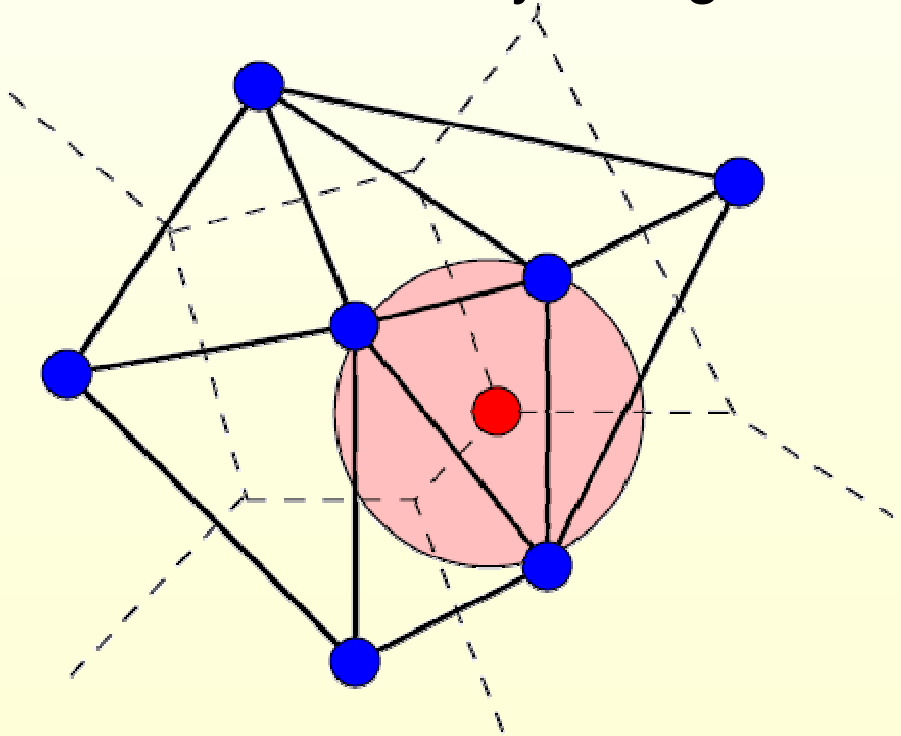
Delaunay Triangulation

- Dual of Voronoi diagram: Connect an edge if their Voronoi cells are adjacent.
- Triangulation of the convex hull.



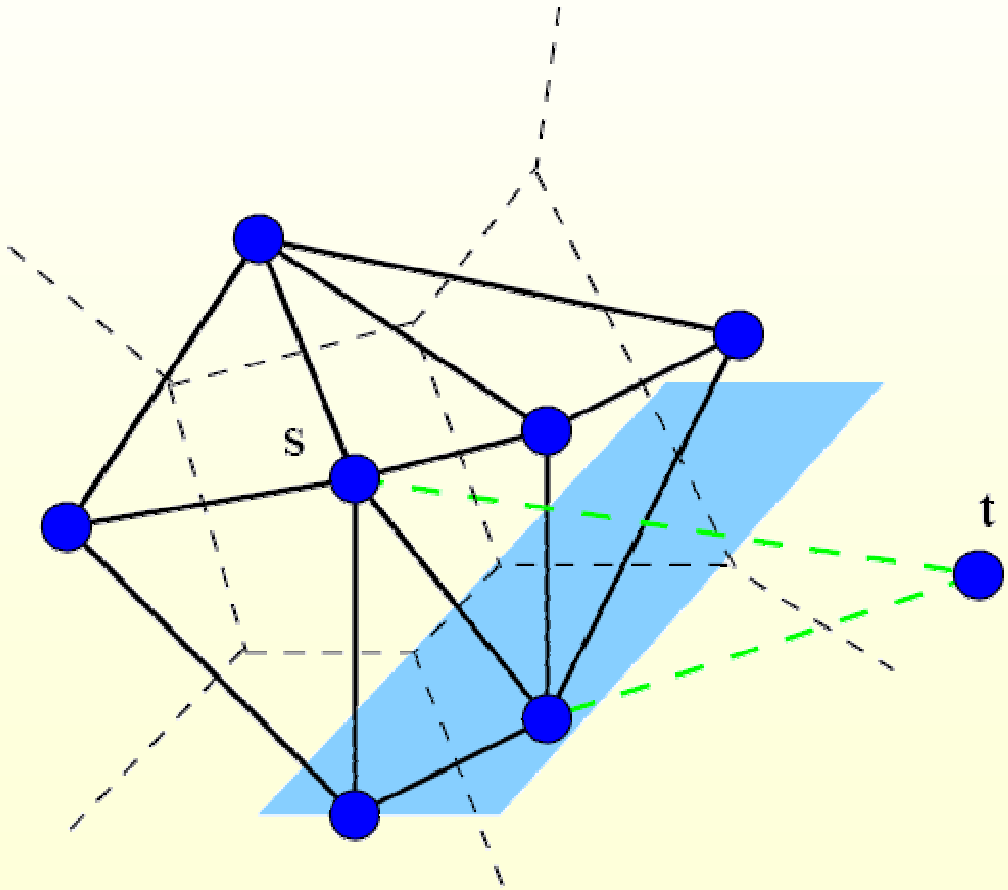
Delaunay Triangulation

- "Empty-circle property": the circumcircle of a Delaunay triangle is empty of other points.
- The converse is also true: if all the triangles in a triangulation are locally Delaunay, then the triangulation is a Delaunay triangulation.



Greedy Routing on Delaunay

- Claim: Greedy routing on DT never gets stuck.



Delaunay Triangulation

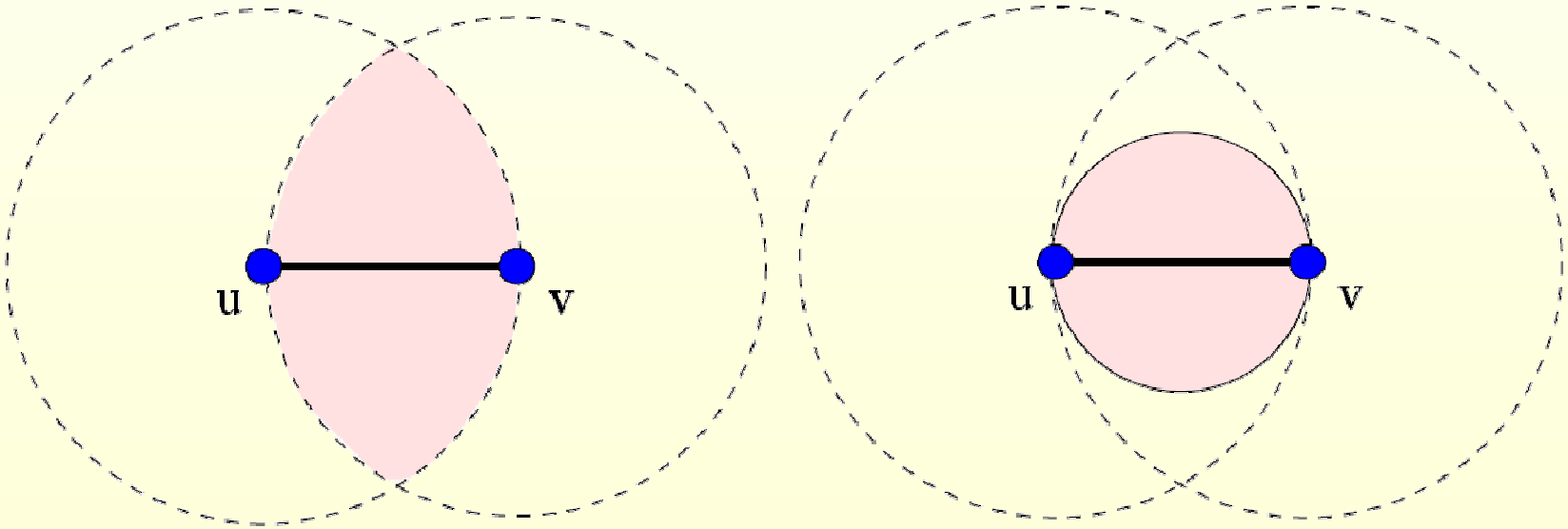
- For an arbitrary point set, the Delaunay triangulation may contain long edges.
- Centralized construction.

- If the nodes are uniformly placed inside a unit disk, the longest Delaunay edge is $O((\log n/n)^{1/3})$. [Kozma et.al. PODC'04]

- Next: 2 planar subgraphs that can be constructed in a distributed way: relative neighborhood graph (RNG) and the Gabriel graph.

Relative Neighborhood Graph and Gabriel Graph

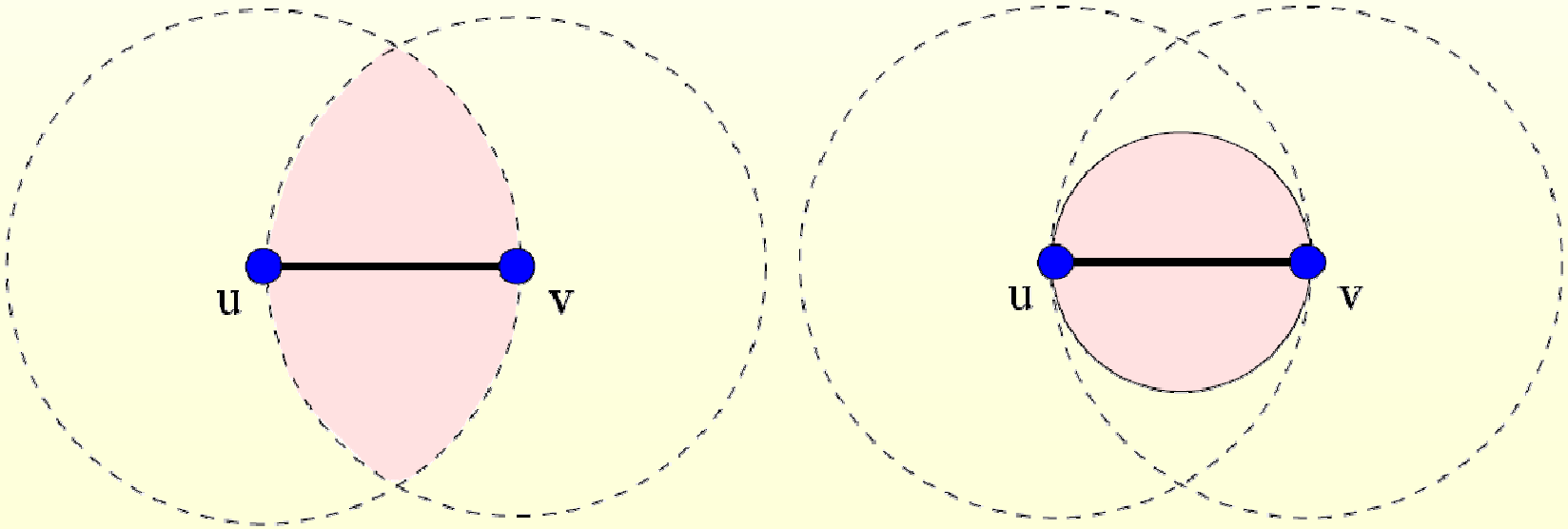
- Relative Neighborhood Graph (RNG) contains an edge uv if the lune is empty of other points.
- Gabriel Graph (GG) contains an edge uv if the disk with uv as diameter is empty of other points.
- Both can be constructed in a distributed way.



Relative Neighborhood Graph and Gabriel Graph

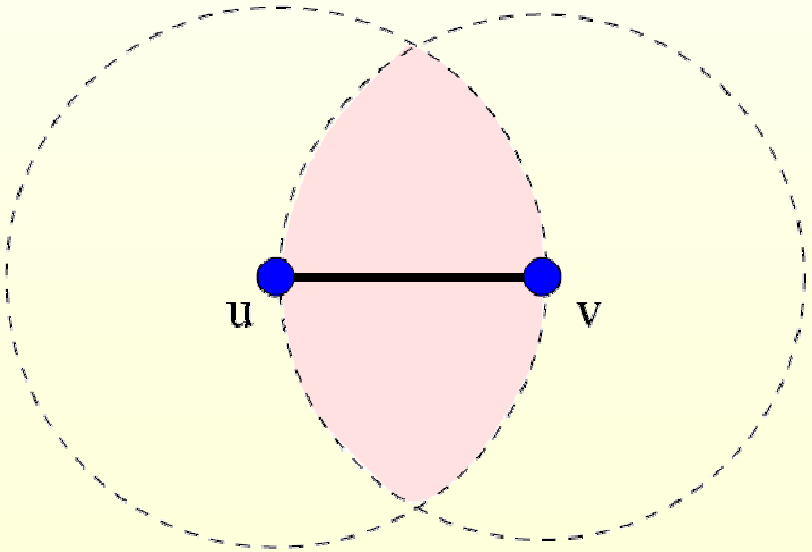
- Claim: MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ Delaunay

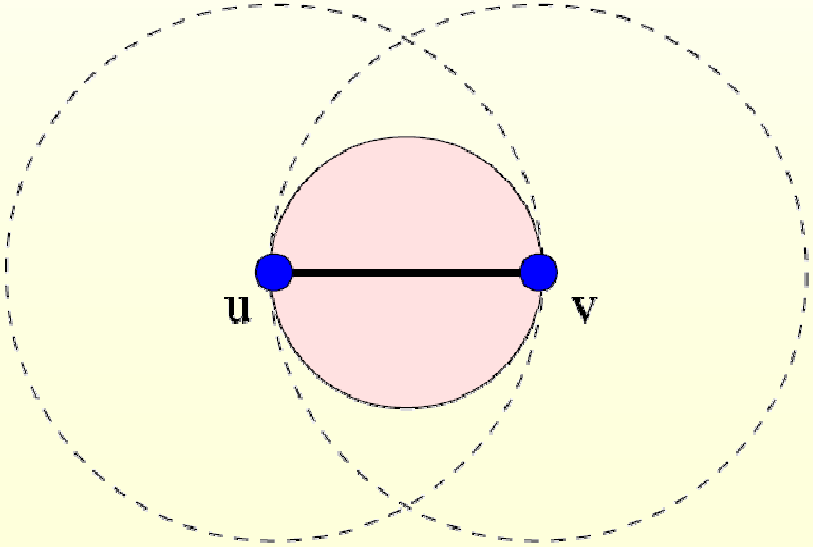
- Thus, RNG and GG are planar (Delaunay is also planar) and keep the connectivity (MST has the same connectivity of UDG).



MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ Delaunay

1. RNG $\subseteq$ GG: if the lune is empty, then the disk with uv as diameter is also empty.
2. GG $\subseteq$ Delaunay: the disk with uv as diameter is empty, then uv is a Delaunay edge.

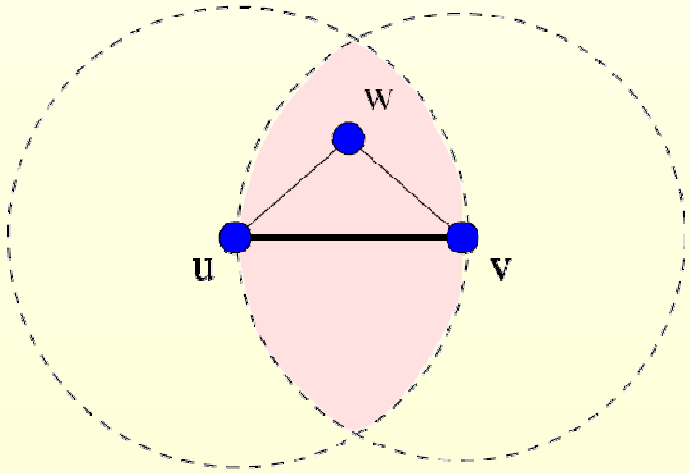




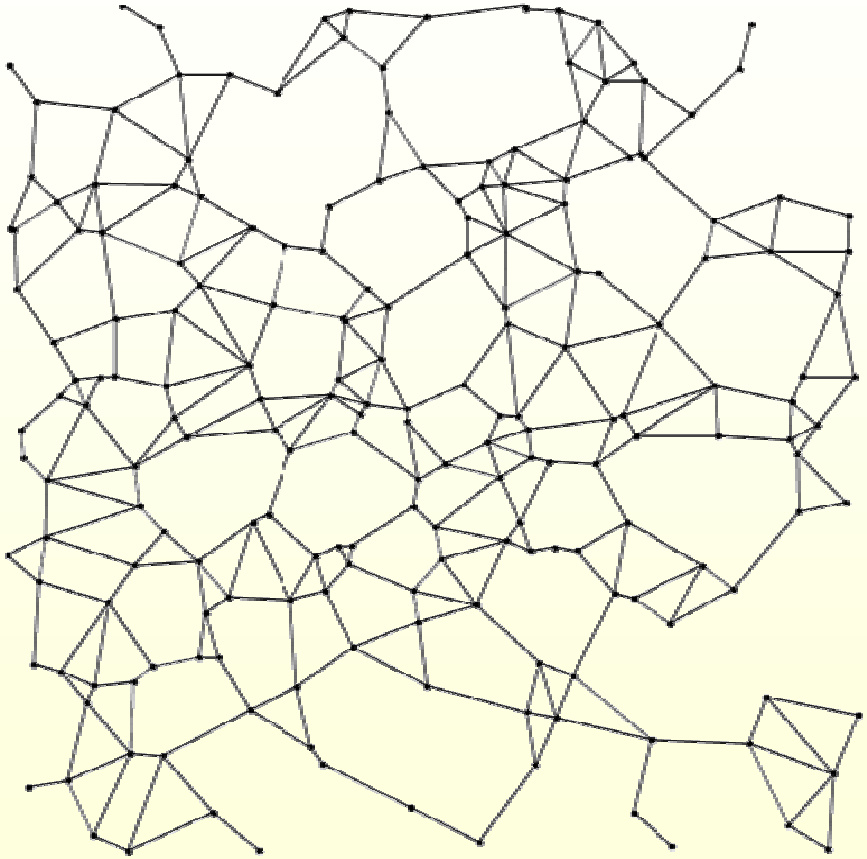
MST ⊆ RNG ⊆ GG ⊆ Delaunay

3. MST ⊆ RNG:
   - Assume uv in MST is not in RNG, there is a point w inside the lune. |uv|>|uw|, |uv|>|vw|.
   - Now we delete uv and partition the MST into two subtrees.
   - Say w is in the same component with u, then we can replace uv by wv and get a lighter tree. ➔ contradiction.

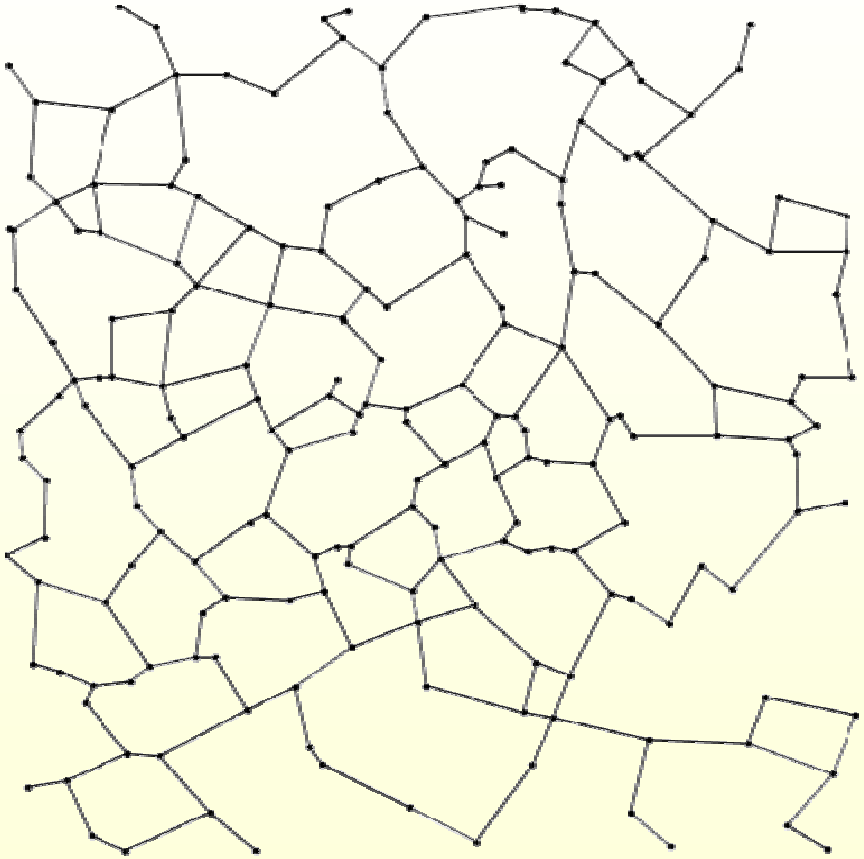
RNG and GG are planar (Delaunay is planar) and keep the connectivity (MST has the same connectivity of UDG).



An example of GG and RNG



GG



RNG

Some Problems Remain

- Both RNG and GG remove some edges ➔ a short path may not exist!
- The shortest path on RNG or GG might be much longer than the shortest path on the original network.
- Even if the planar subgraph contains a short path, can greedy routing and face routing find a short one?

Face Routing on Planar Graphs

Face Routing

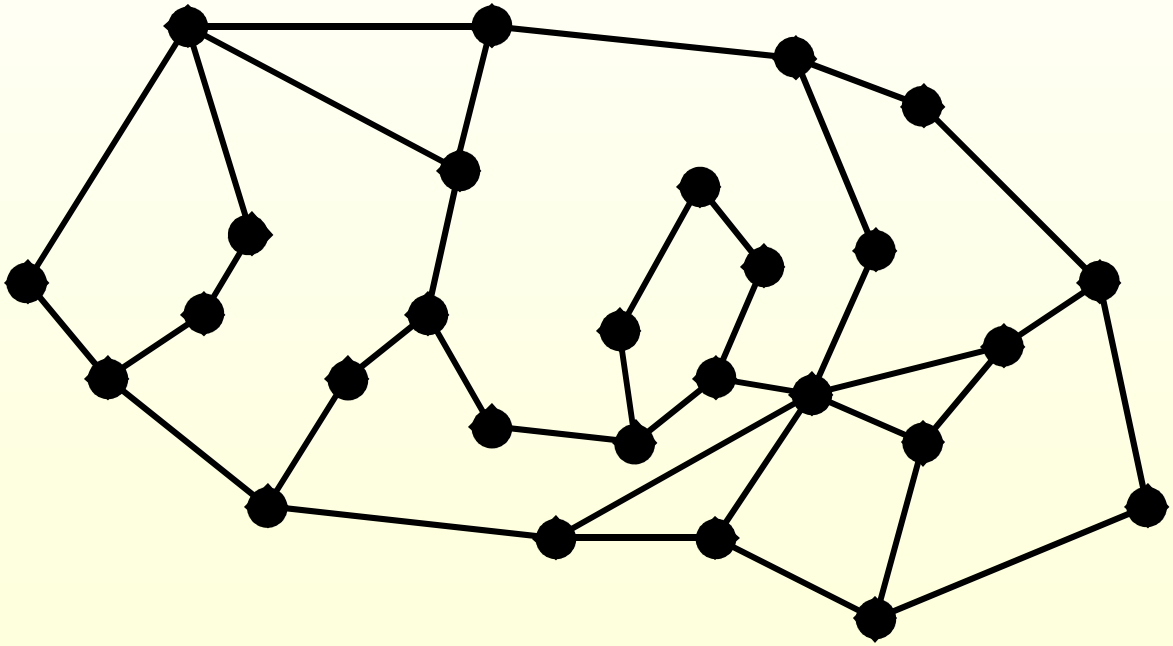
Geometric routing algorithm for planar graphs

Distance between any two nodes is lowerbounded by a constant d_0

$\Rightarrow \Omega(1)$-model

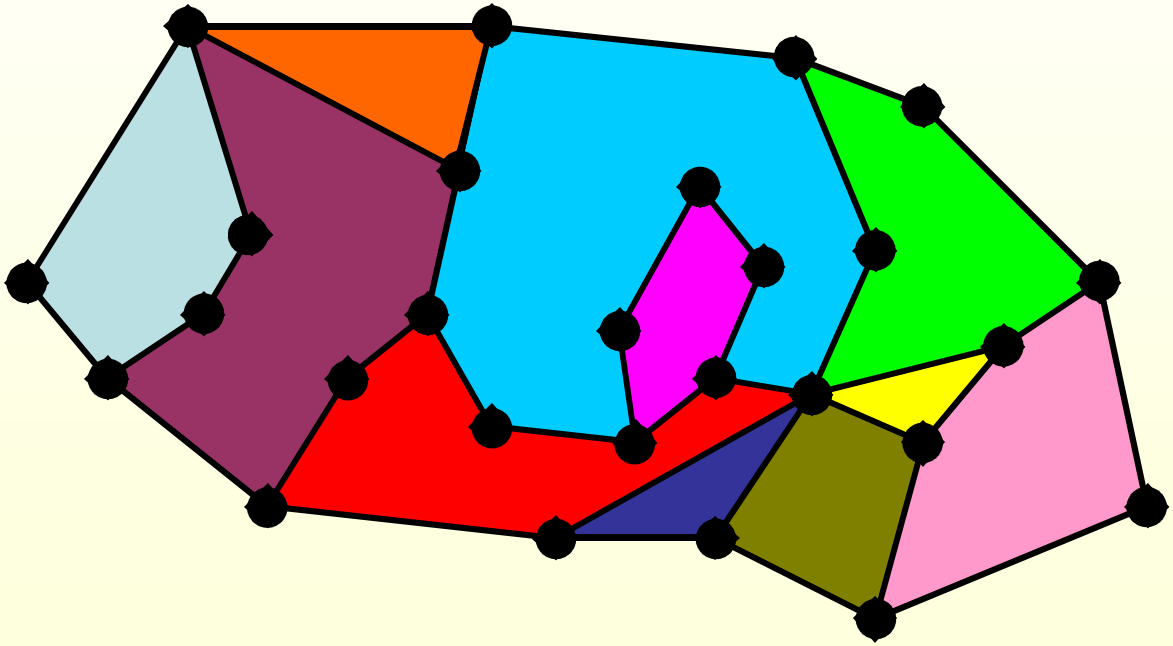
Compass Routing on Geometric Networks
[Kranakis, Singh, Urrutia, '99]
Asymptotically Optimal Geometric Mobile AdHoc Routing
[Kuhn, Wattenfofer, Zollinger, '02]

Face Routing (FR)



[From Kuhn et. al. ’02]

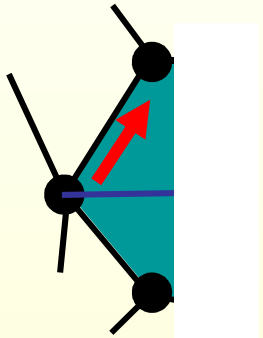
Face Routing (Faces)

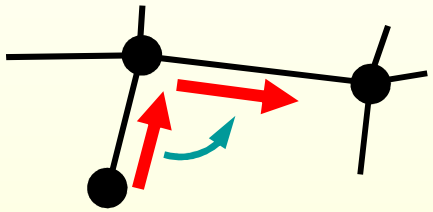


[From Kuhn et. al. ’02]

Perimeter/Face Routing Properties

- All necessary information is stored in the message
  - Source and destination positions are given
  - Point of transition to next face needs to be chosen

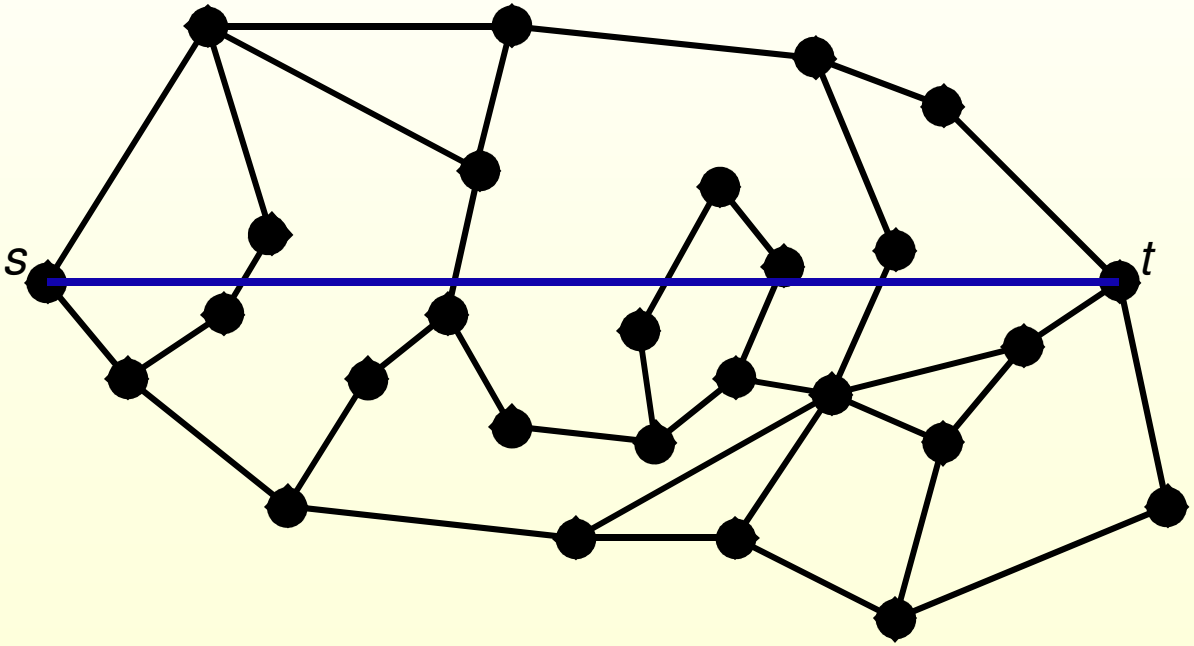




"Right Hand Rule"

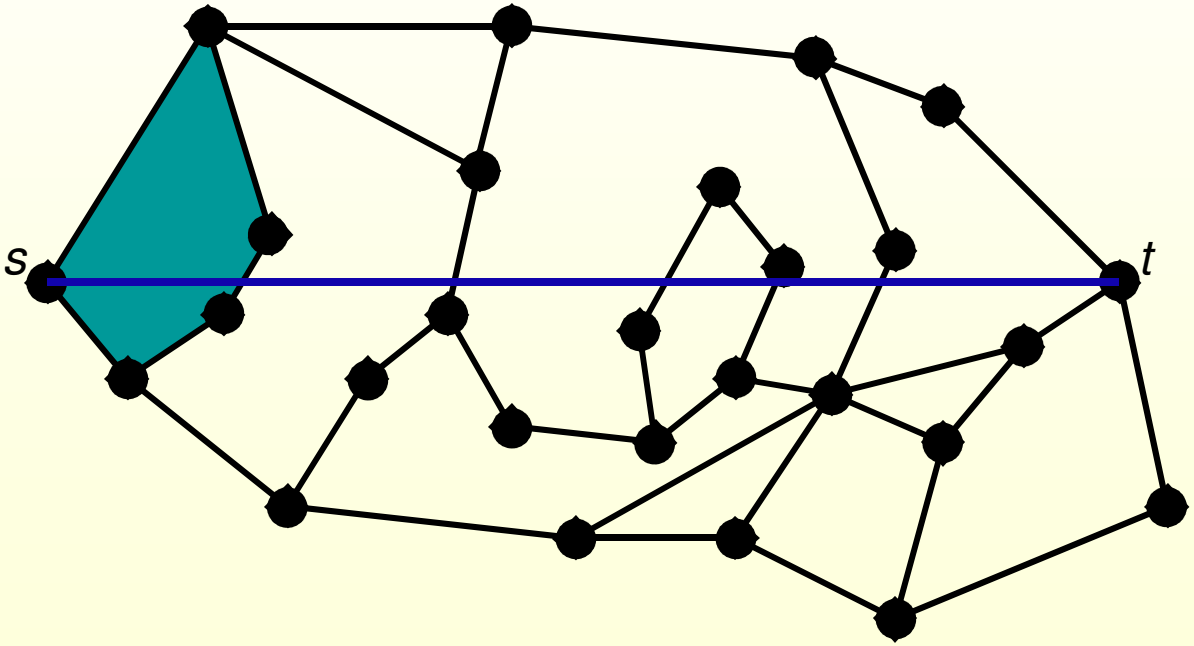
- Completely local:
  - Knowledge about direct neighbors' positions sufficient
  - Faces are implicit, only local neighbor ordering around each node is needed

Face Routing



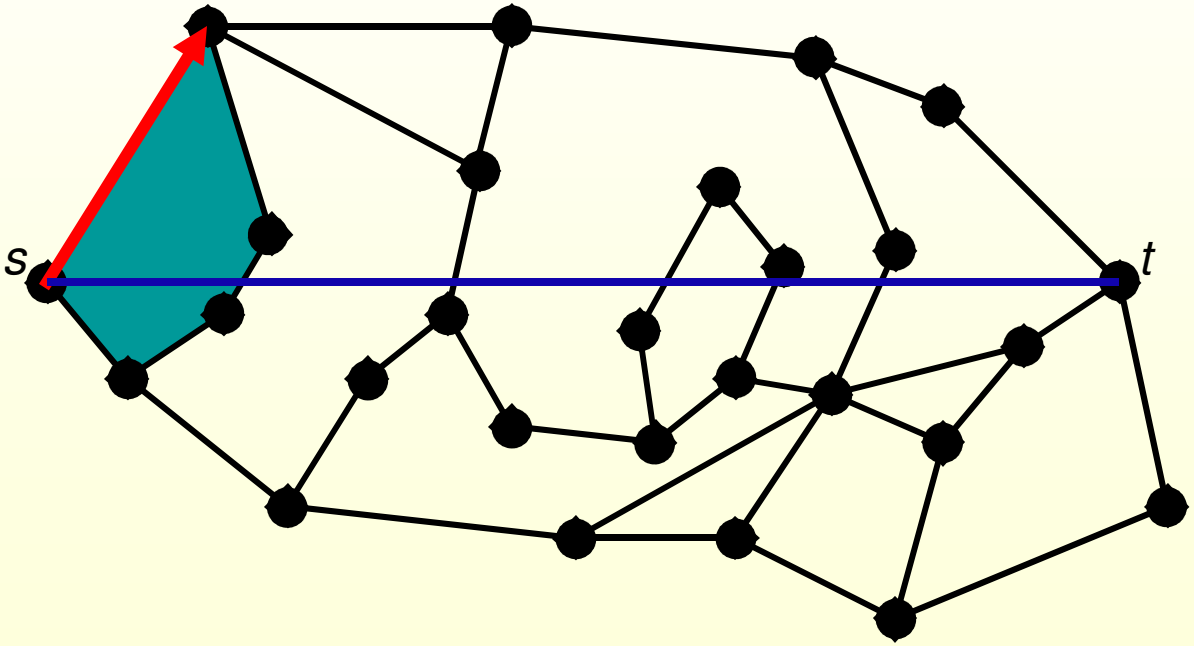
[From Kuhn et. al. ’02]

Face Routing



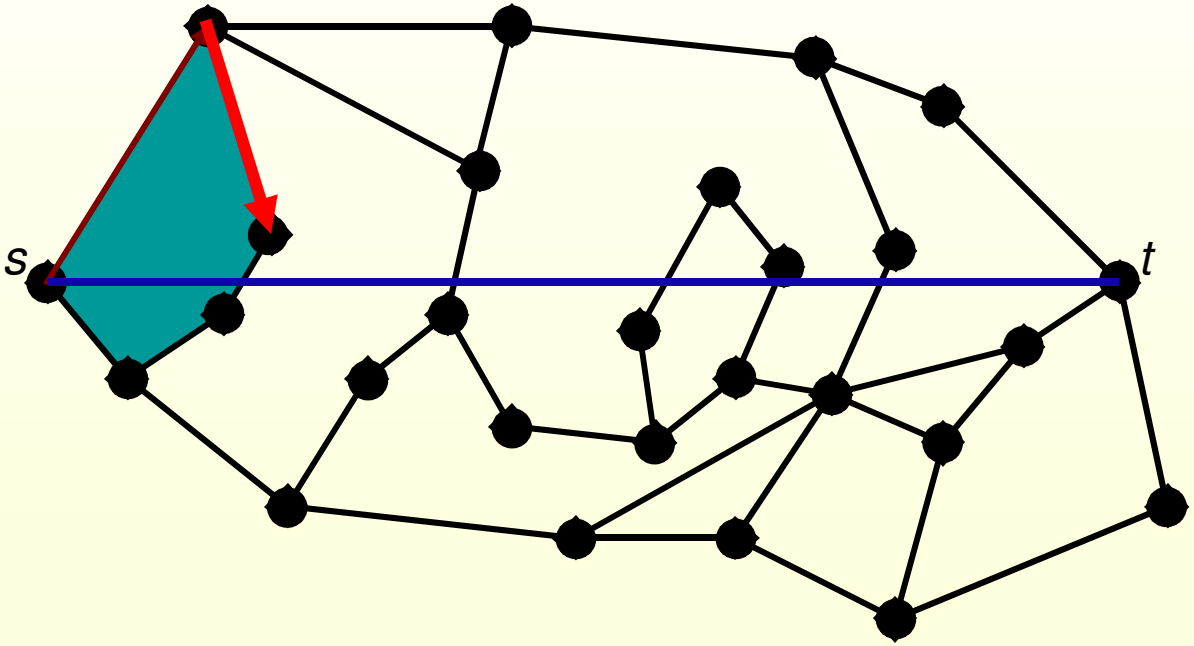
[From Kuhn et. al. ’02]

Face Routing



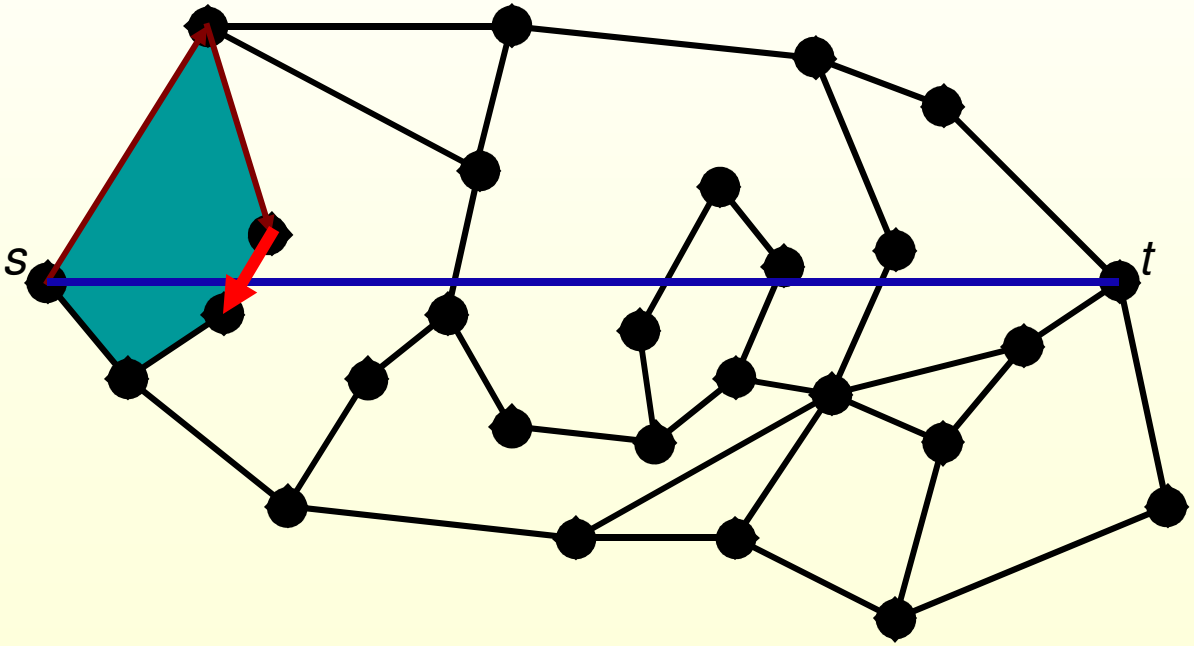
[From Kuhn et. al. ’02]

Face Routing



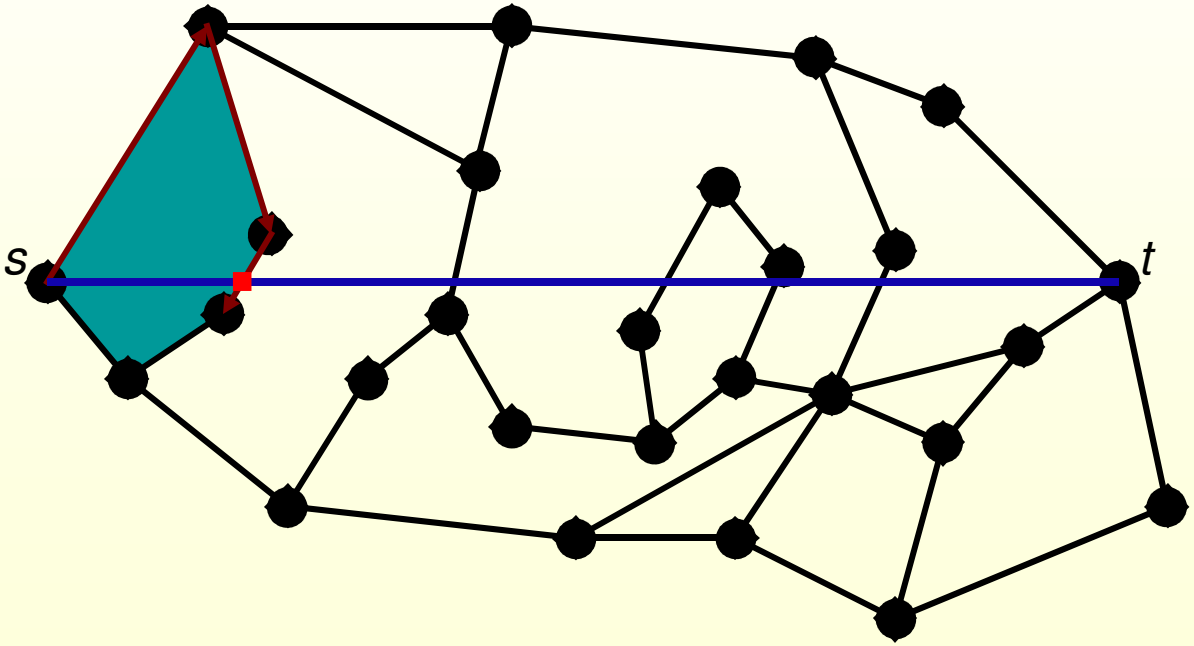
[From Kuhn et. al. ’02]

Face Routing



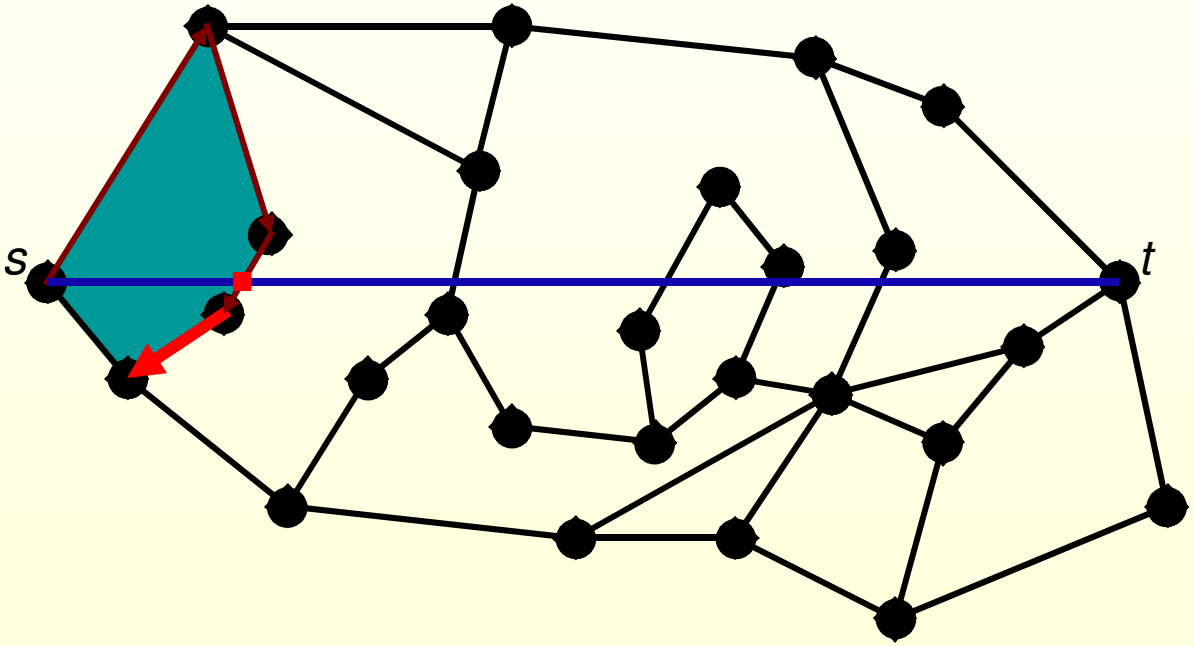
[From Kuhn et. al. ’02]

Face Routing



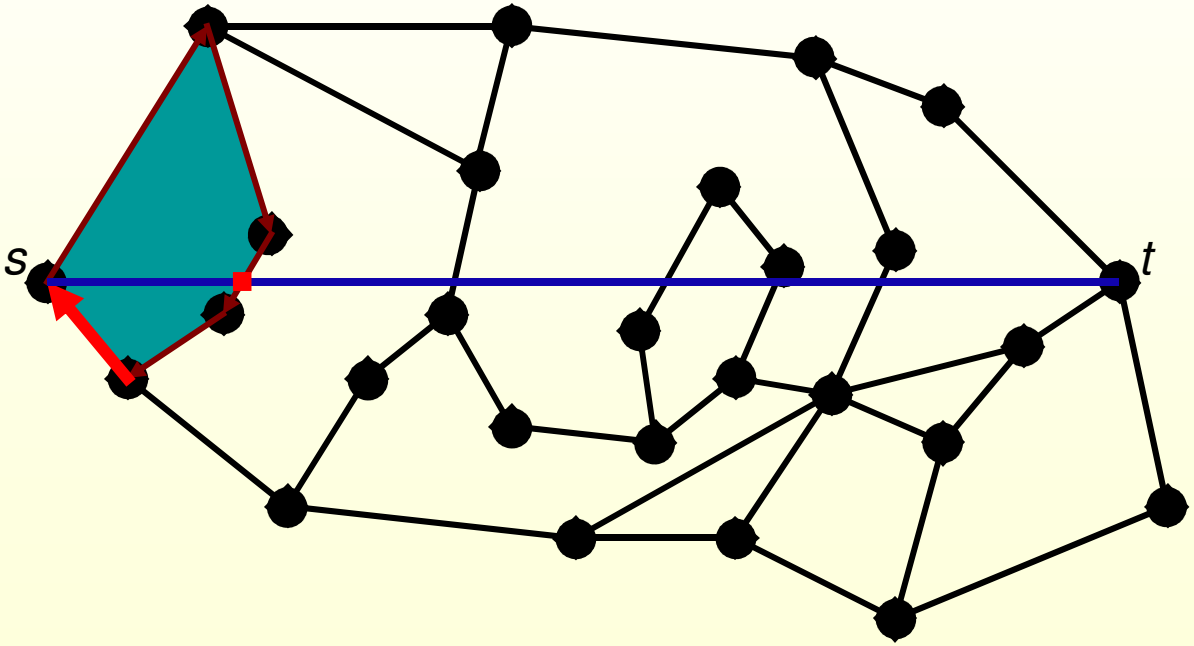
[From Kuhn et. al. ’02]

Face Routing



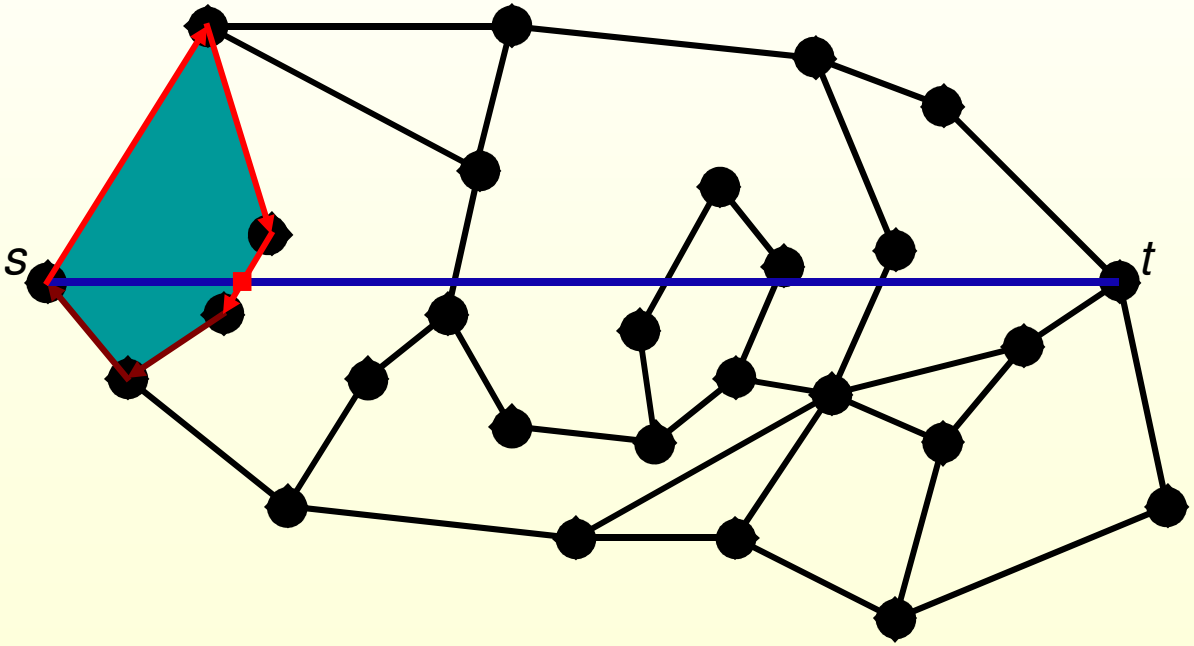
[From Kuhn et. al. ’02]

Face Routing



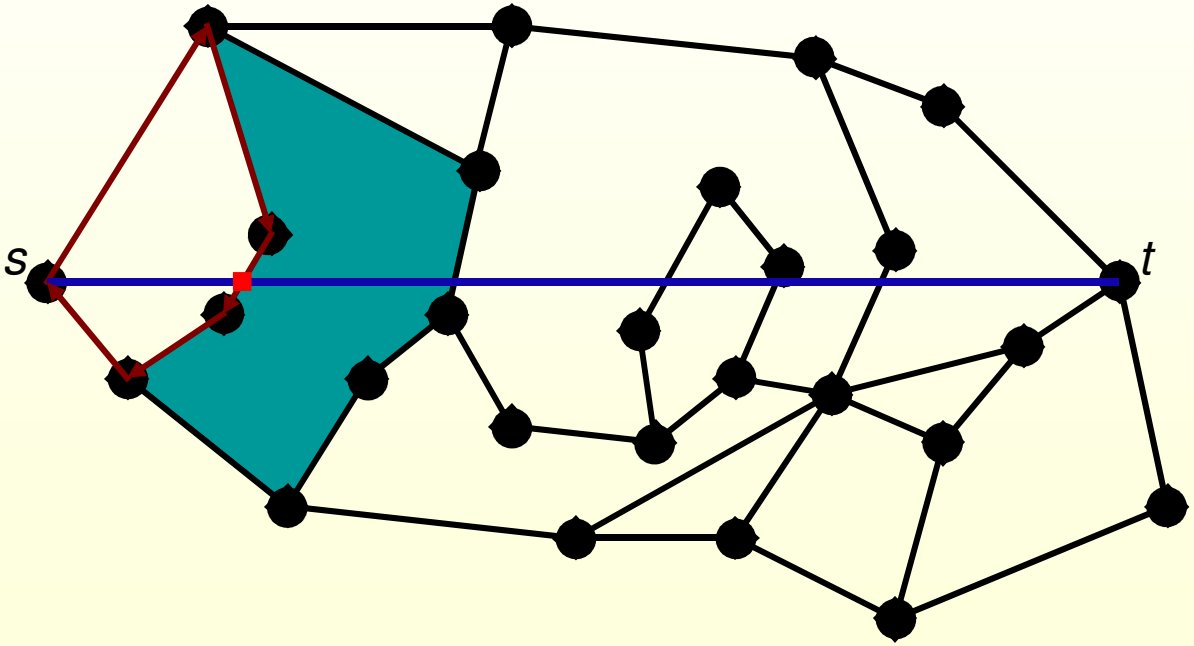
[From Kuhn et. al. ’02]

Face Routing



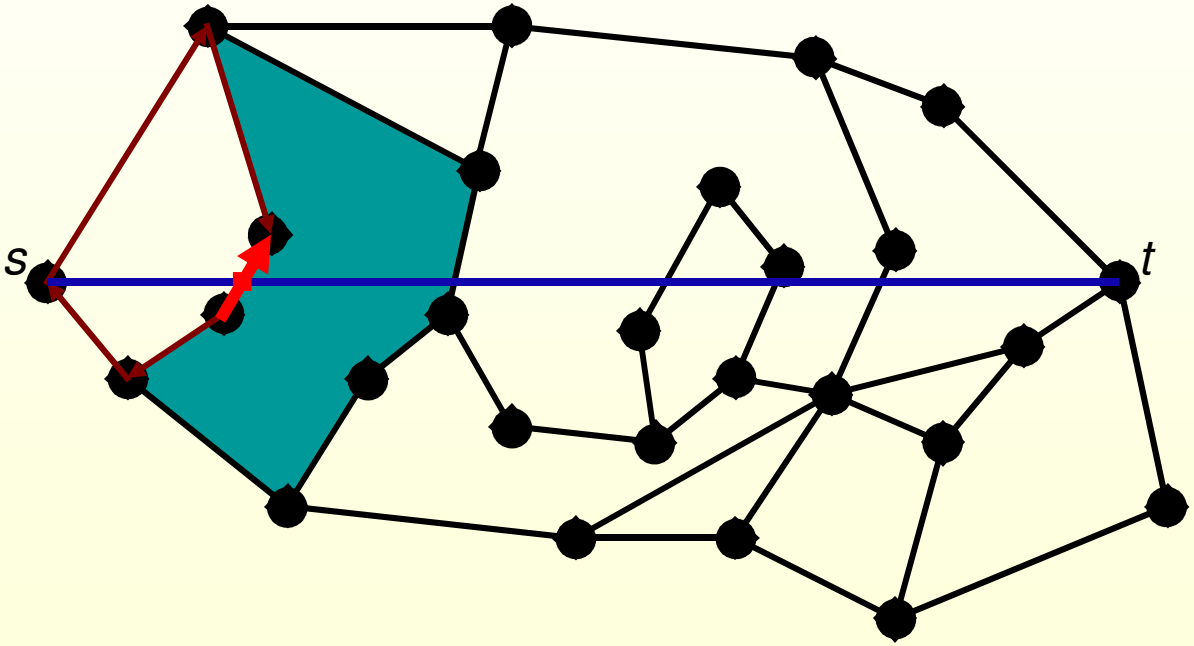
[From Kuhn et. al. ’02]

Face Routing



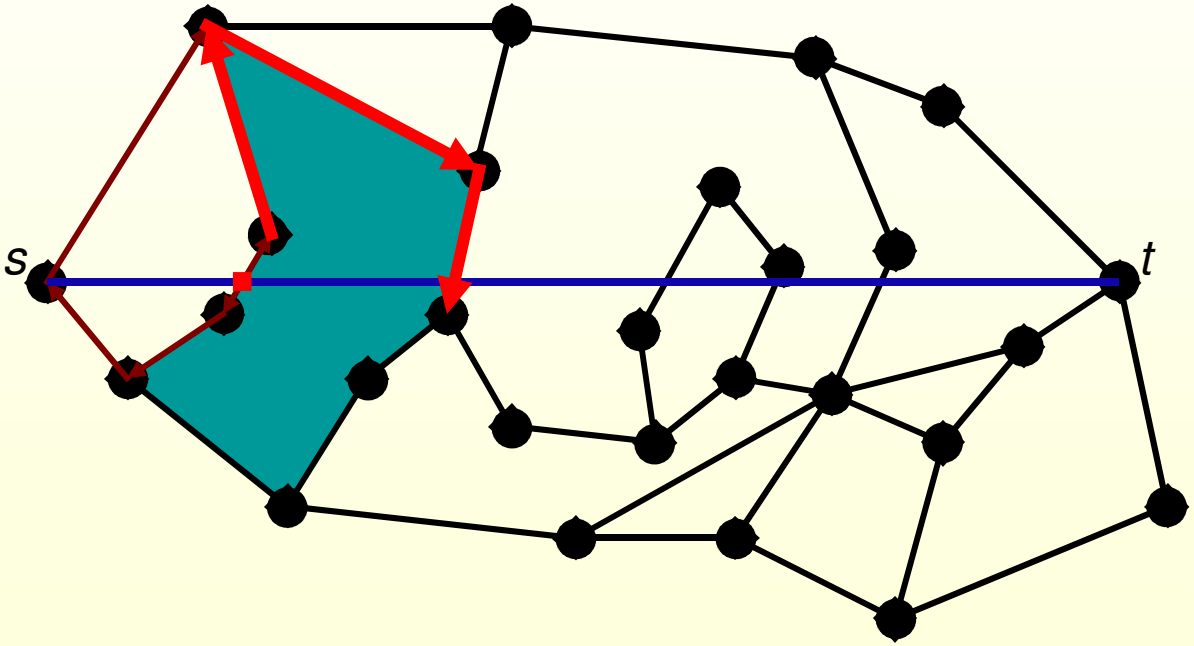
[From Kuhn et. al. ’02]

Face Routing



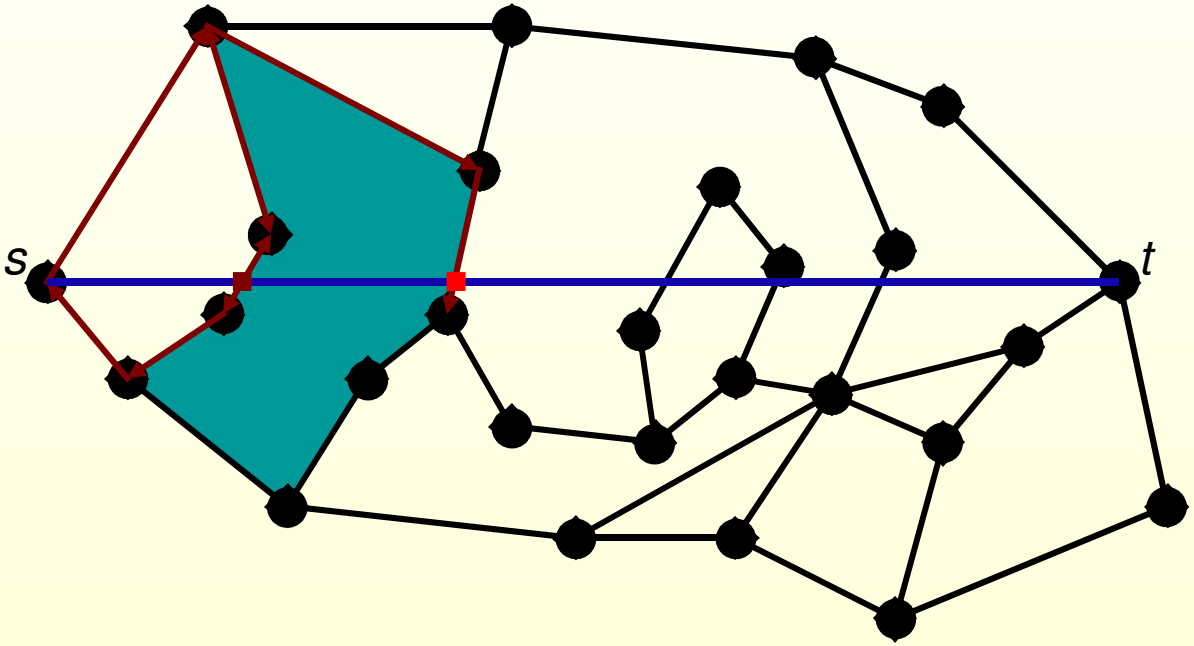
[From Kuhn et. al. ’02]

Face Routing



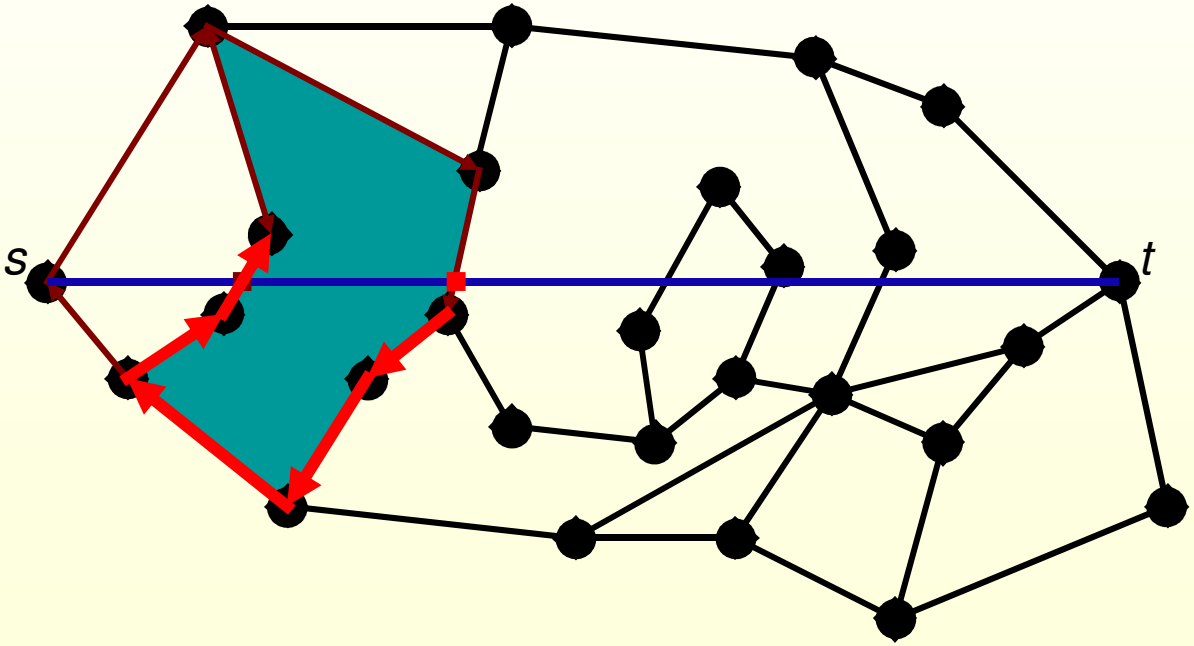
[From Kuhn et. al. ’02]

Face Routing



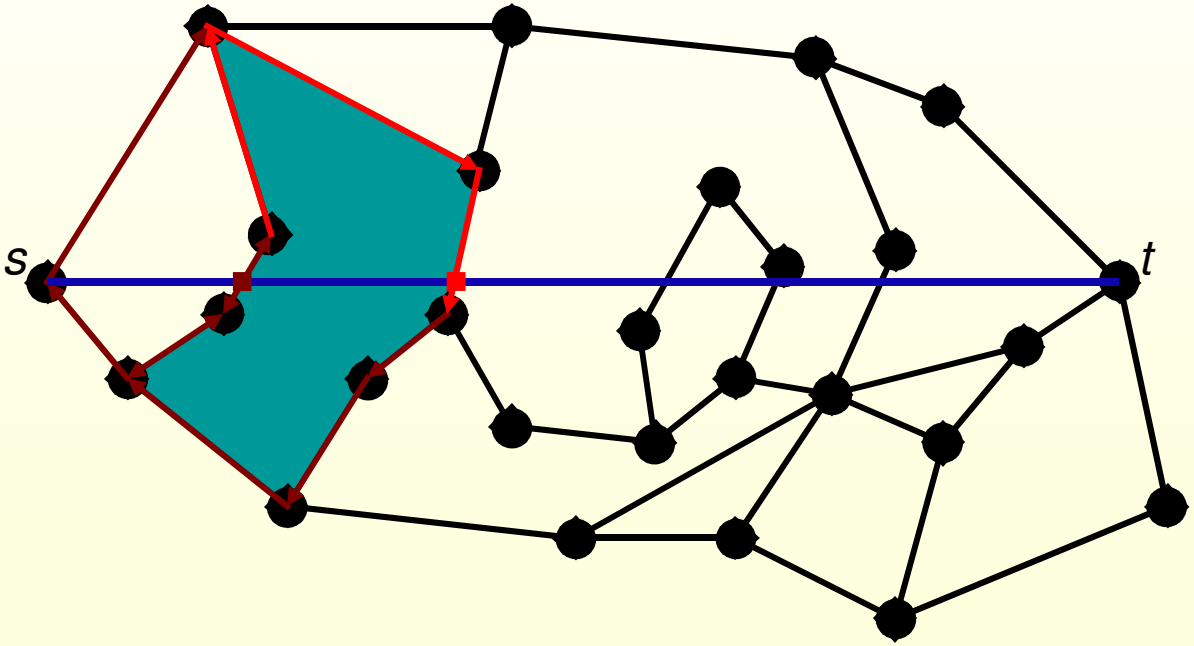
[From Kuhn et. al. ’02]

Face Routing



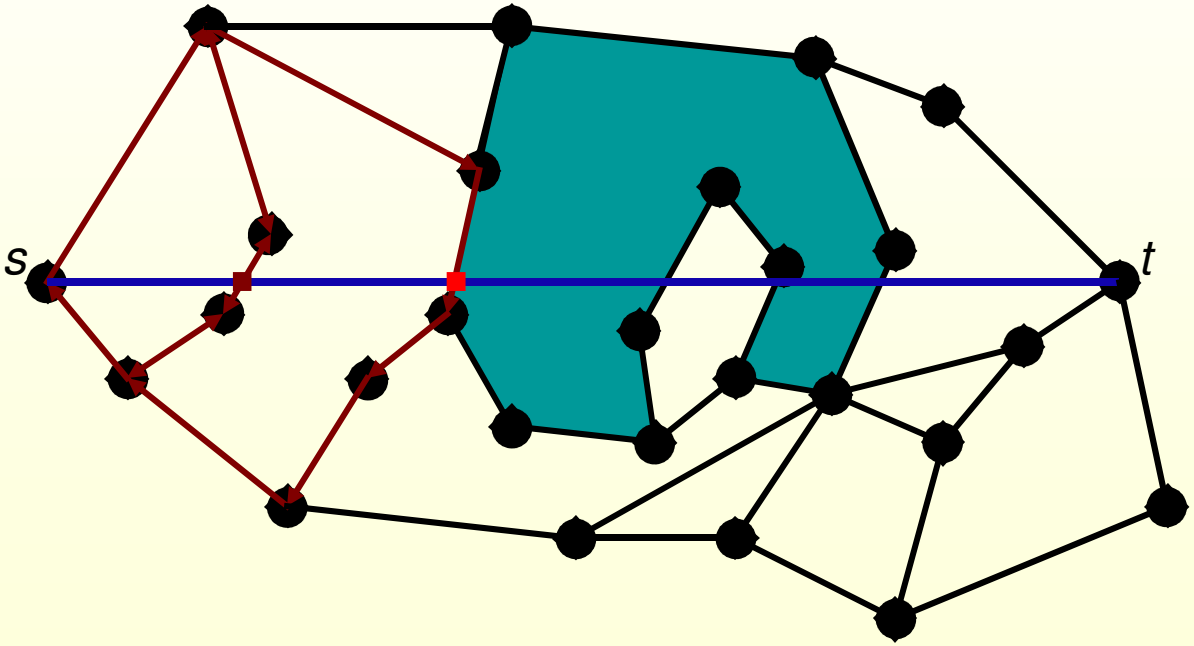
[From Kuhn et. al. ’02]

Face Routing



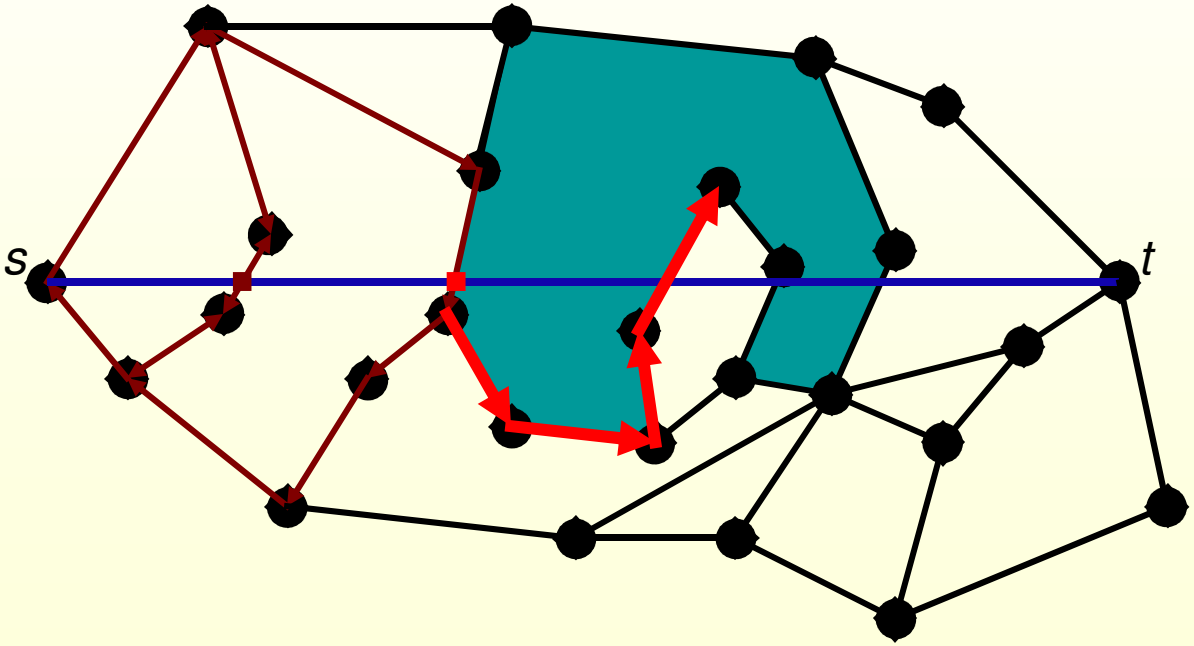
[From Kuhn et. al. ’02]

Face Routing



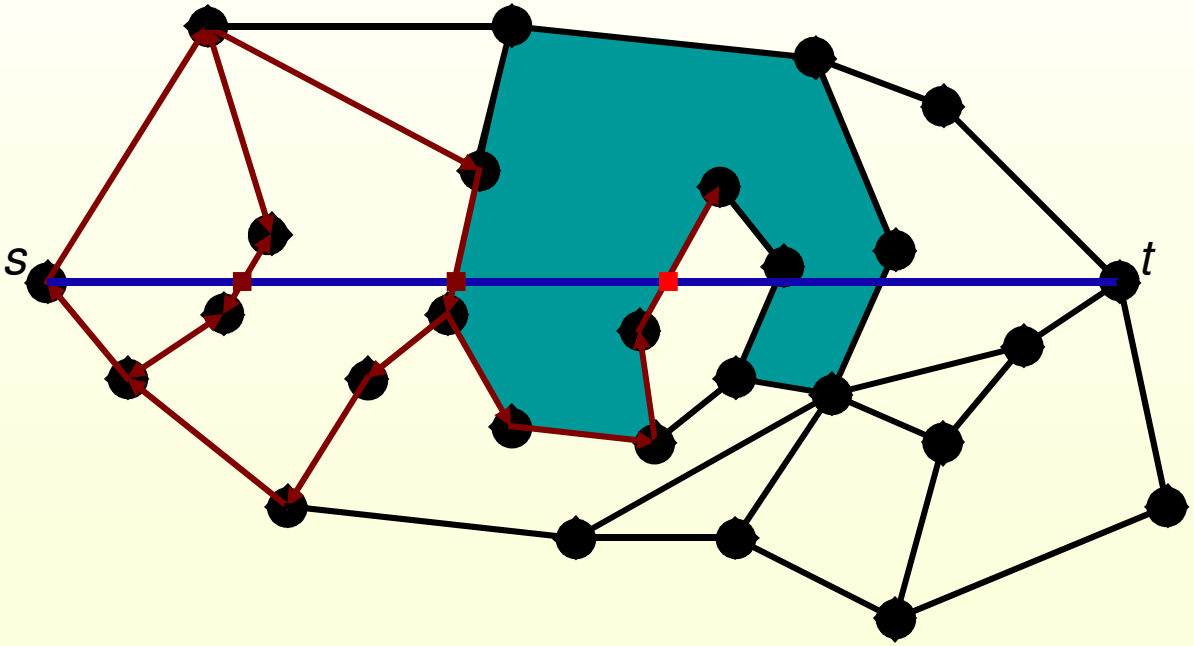
[From Kuhn et. al. ’02]

Face Routing



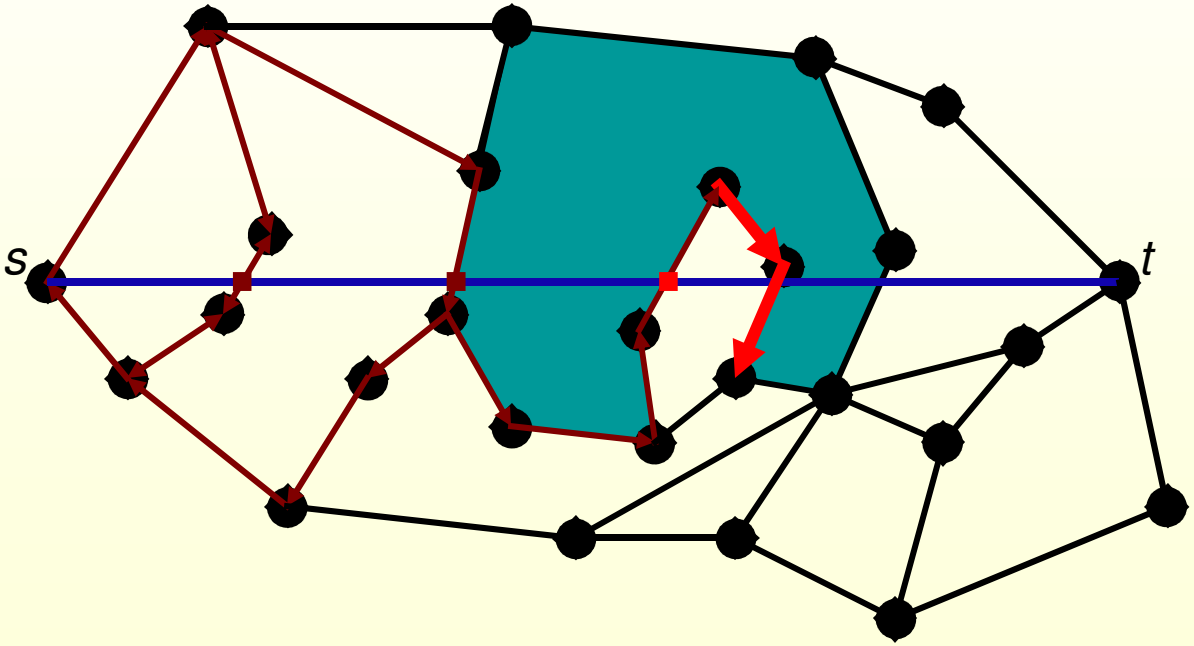
[From Kuhn et. al. ’02]

Face Routing



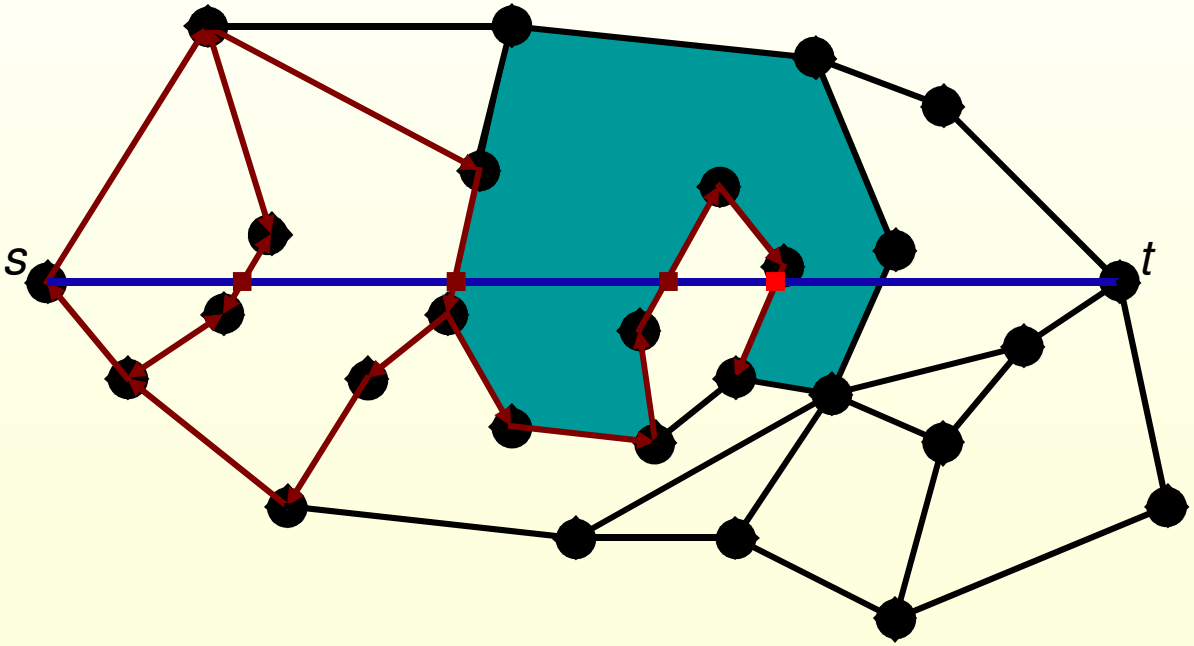
[From Kuhn et. al. ’02]

Face Routing



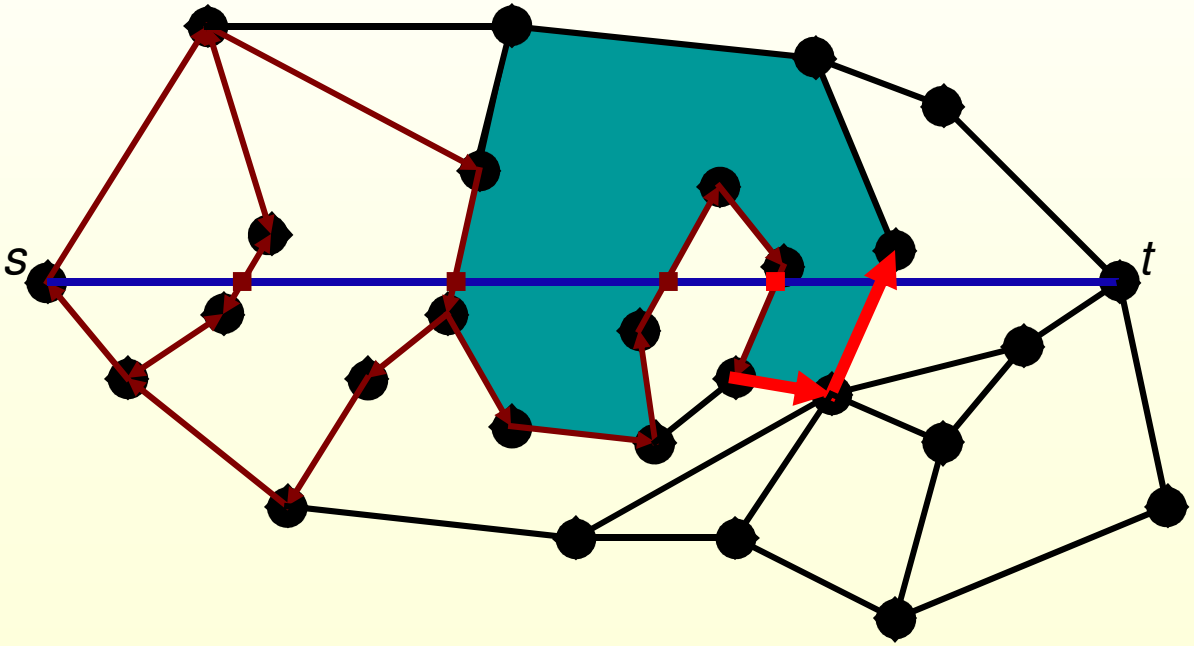
[From Kuhn et. al. ’02]

Face Routing



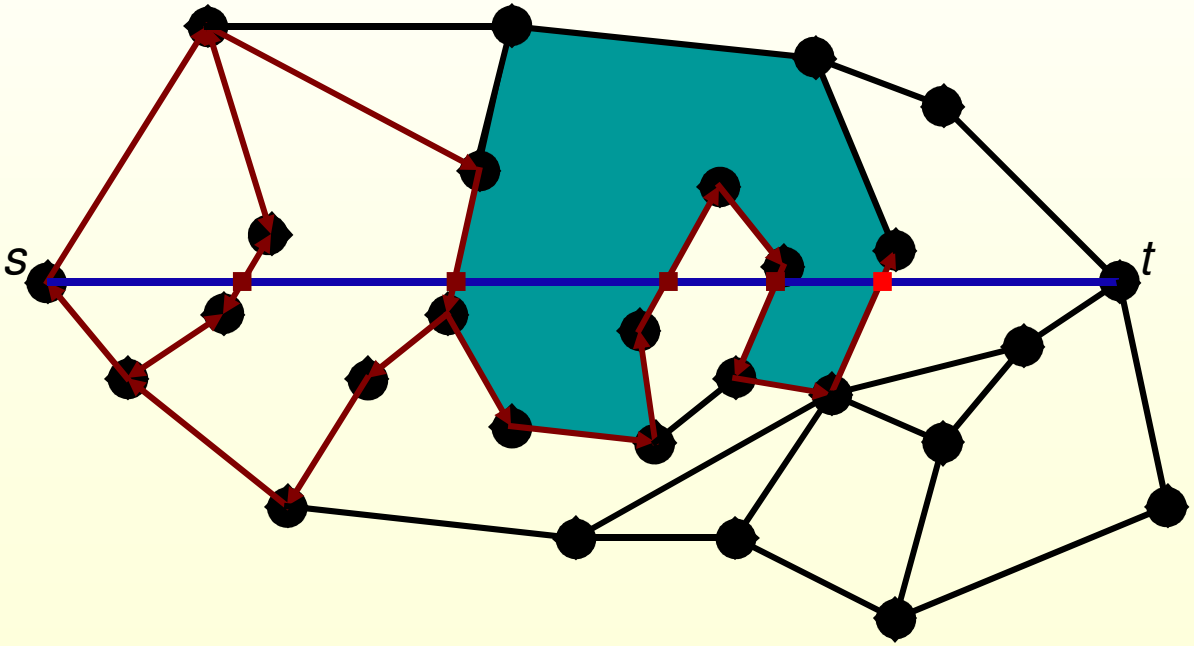
[From Kuhn et. al. ’02]

Face Routing



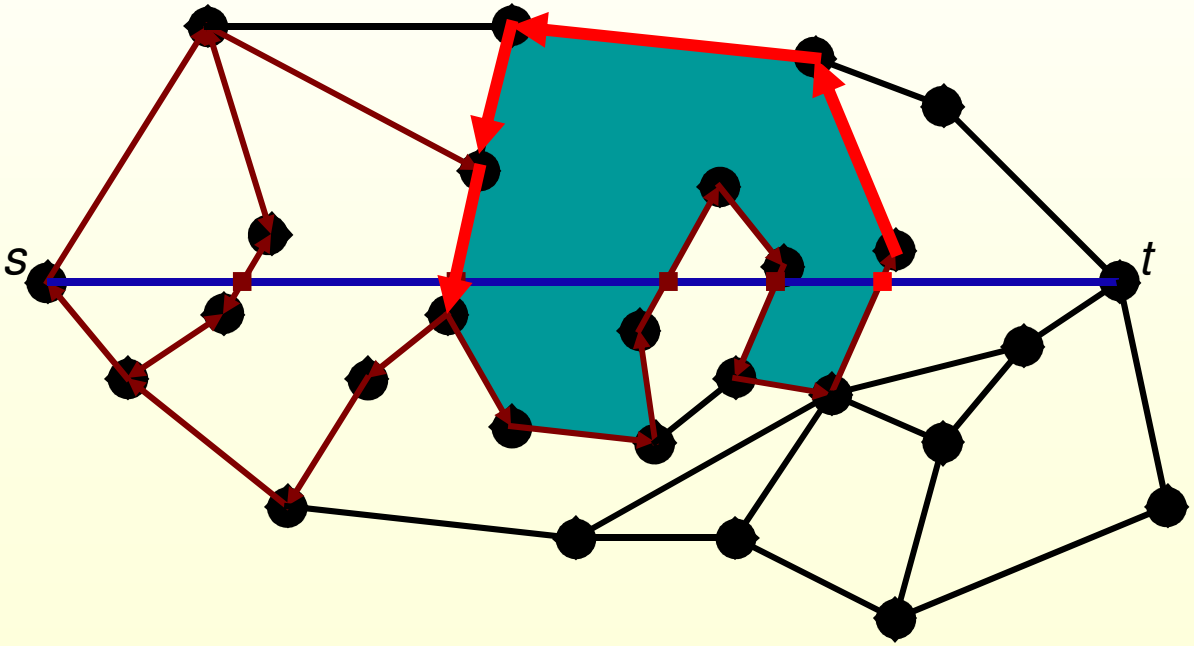
[From Kuhn et. al. ’02]

Face Routing



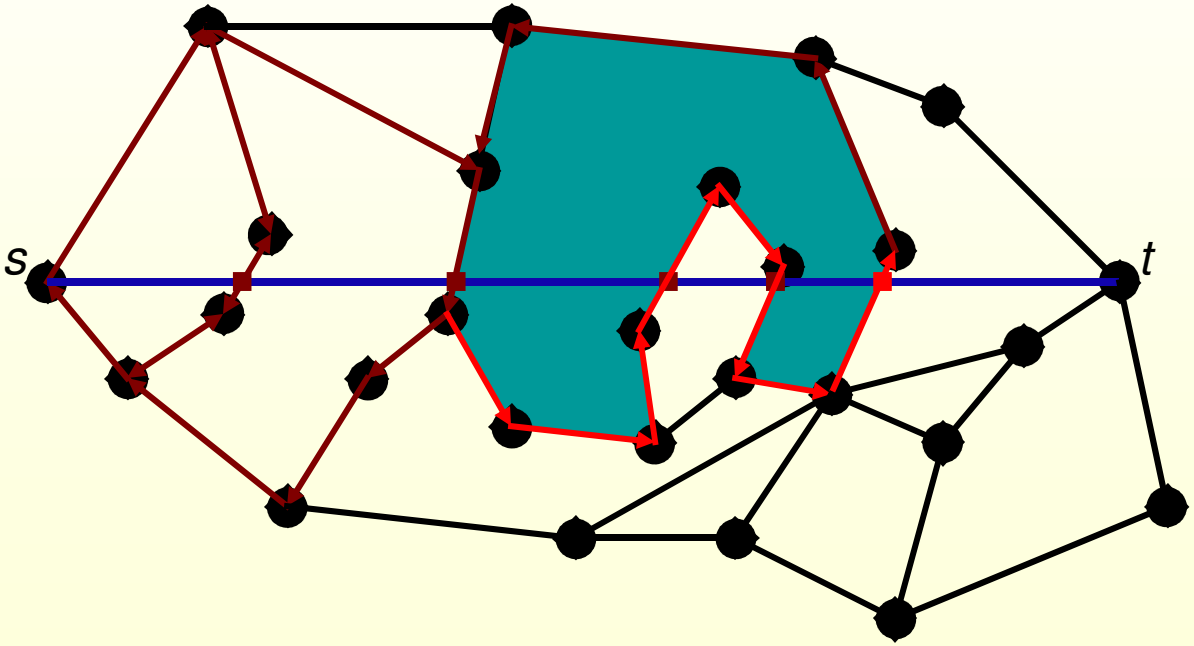
[From Kuhn et. al. ’02]

Face Routing



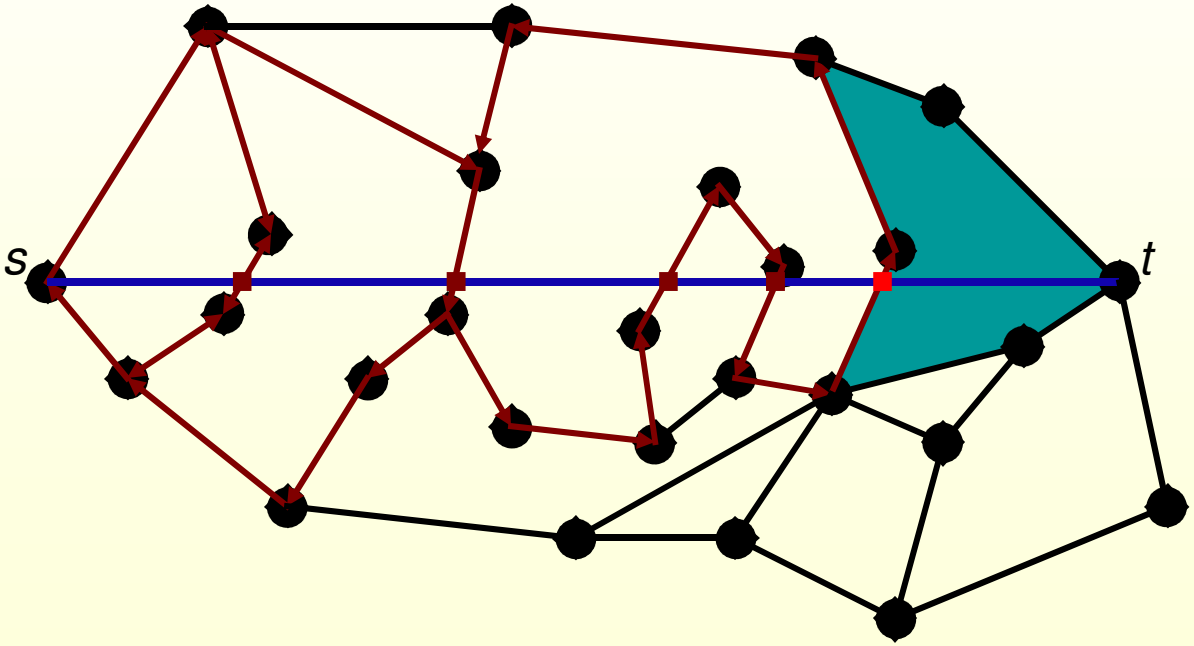
[From Kuhn et. al. ’02]

Face Routing



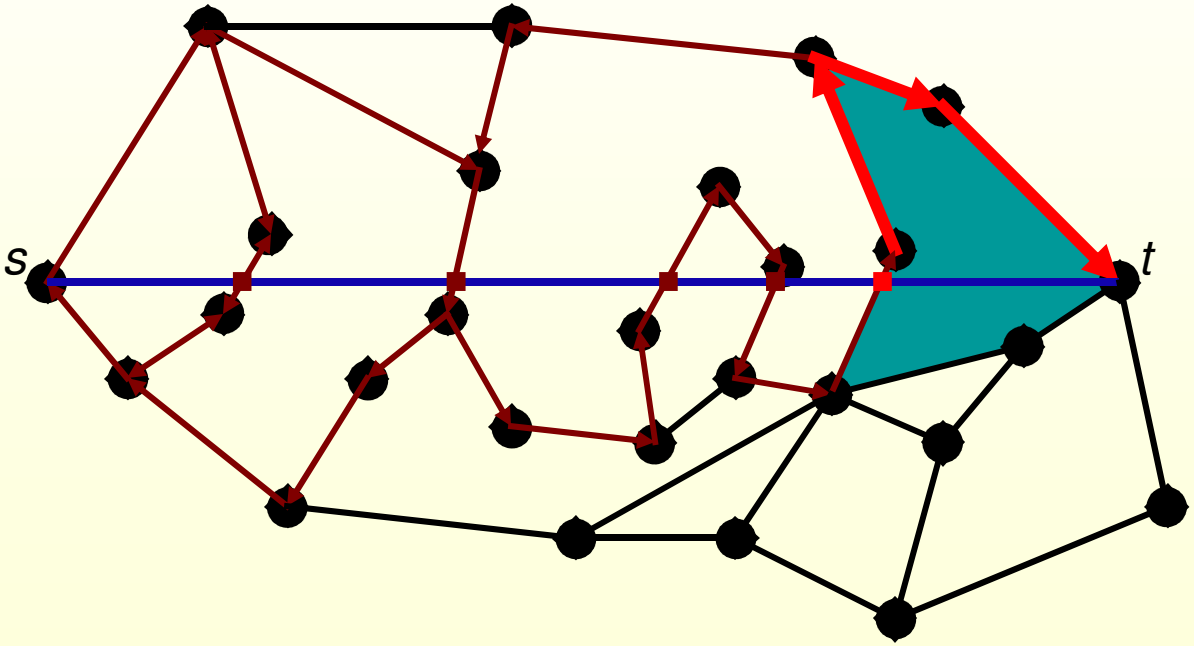
[From Kuhn et. al. ’02]

Face Routing



[From Kuhn et. al. ’02]

Face Routing



[From Kuhn et. al. ’02]

Face Routing (Analysis)

Lemma:

Face Routing always finds a path to the destination. The total cost of Face Routing is $O(n)$.

[From Kuhn et. al. '02]

Face Routing (Analysis)

Proof Sketch:

- each face is explored at most once

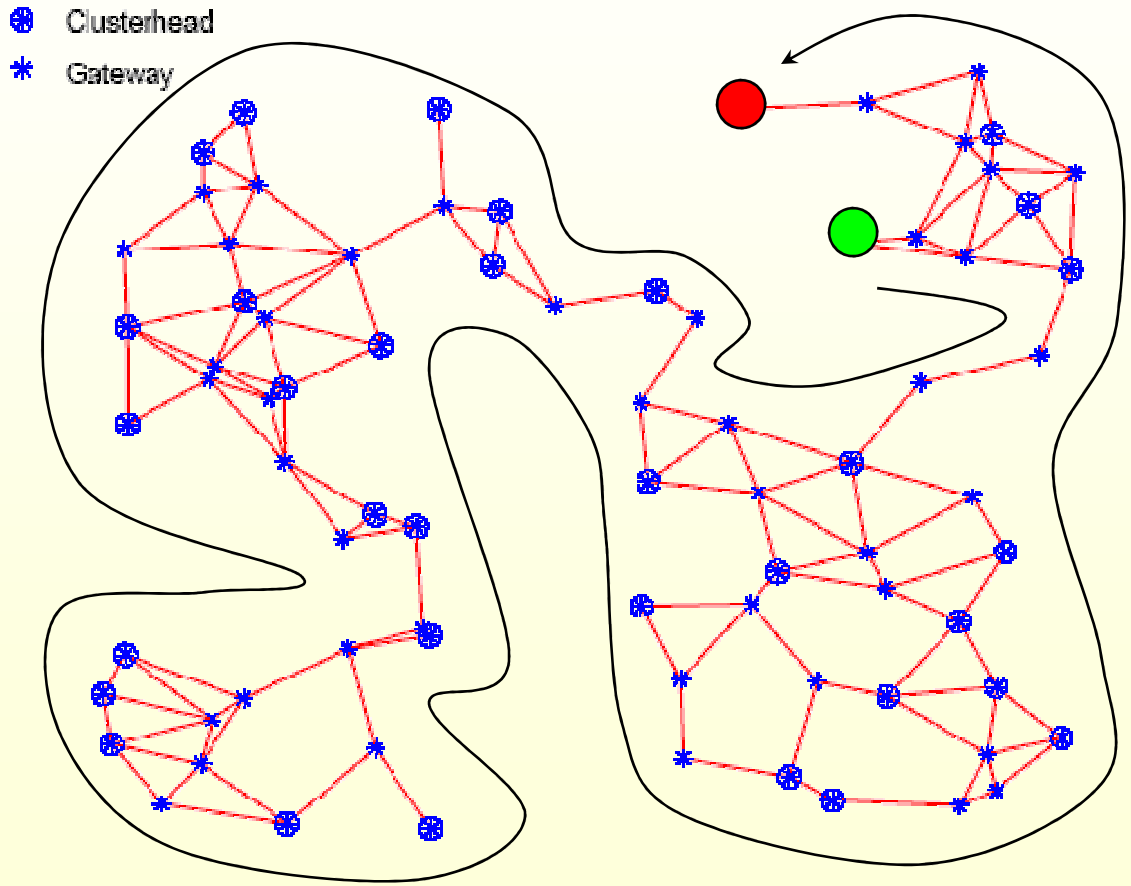
  $\Rightarrow$ each edge is traversed at most four times

- there are at most $3n$-6 edges (Euler's formula)

[From Kuhn et. al. '02]

Non-Optimality of Face Routing

- What if we choose the wrong side?

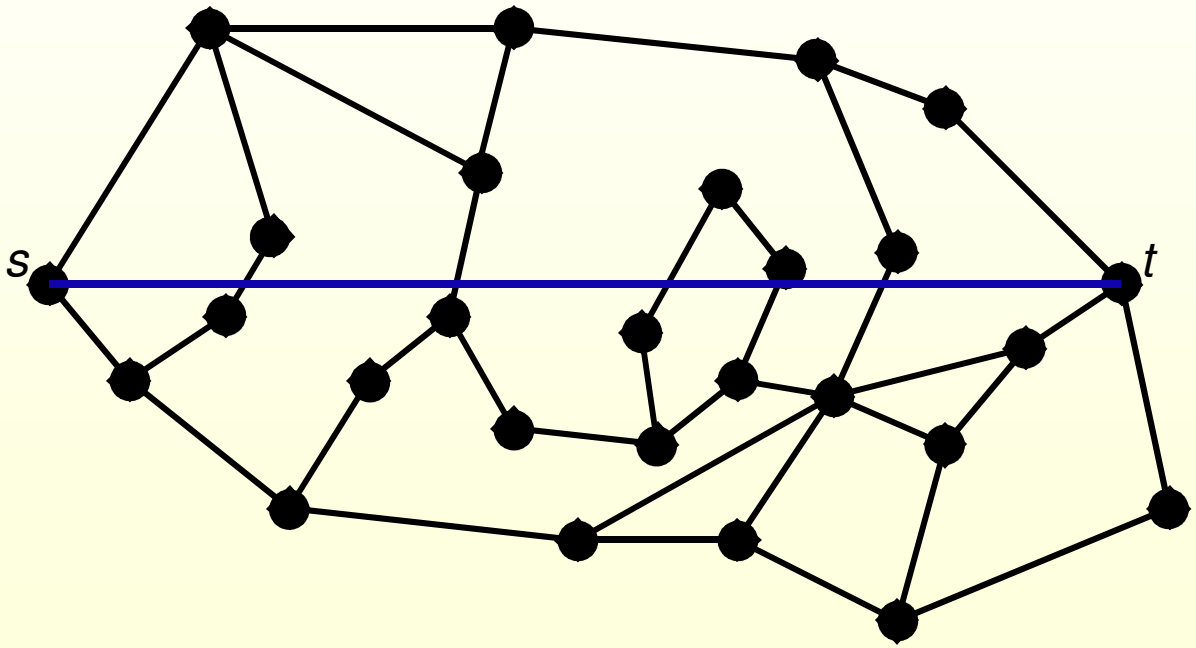


Problem in Face Routing

- Face Routing always reaches destination
- However, even if source and destination are close to each other, Face Routing can take O(n) steps.
- We would like to have an algorithm, whose cost is a function of the cost of an optimal path.

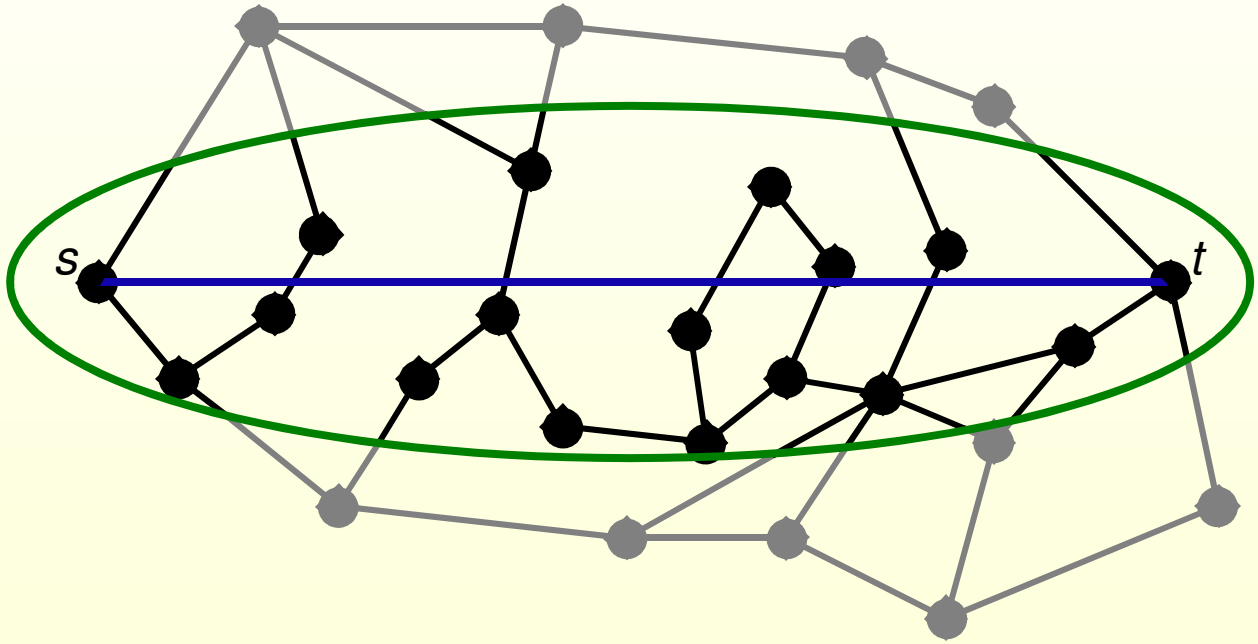
[From Kuhn et. al. '02]

Adaptive Face Routing (AFR)



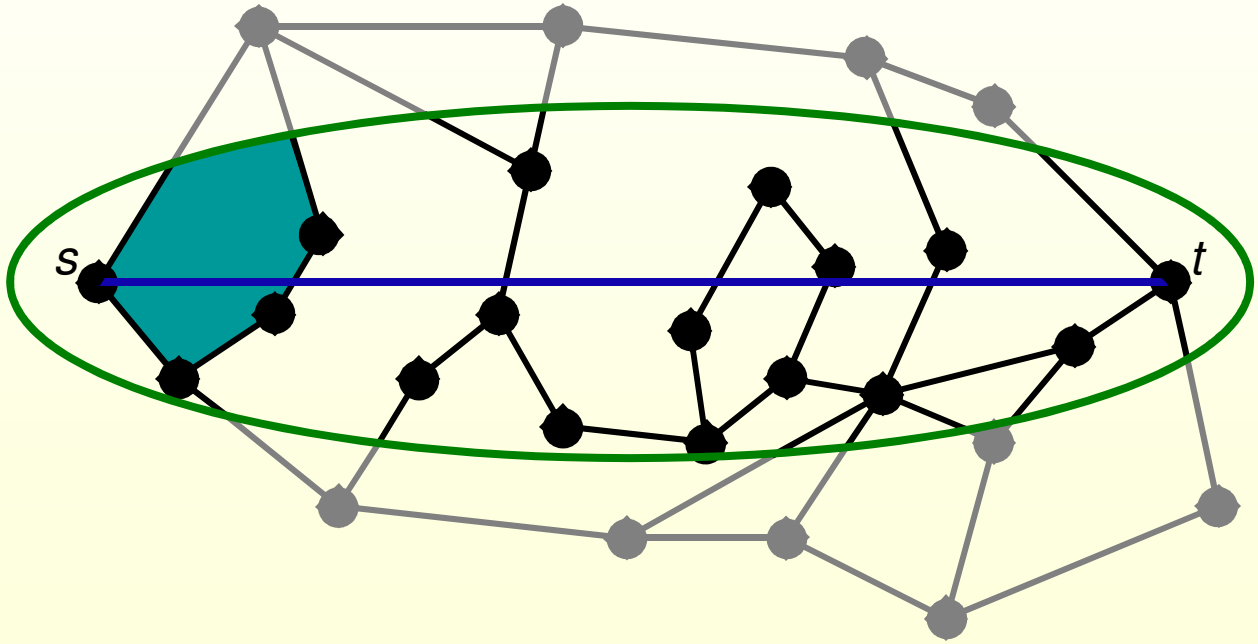
[From Kuhn et. al. ’02]

Adaptive Face Routing



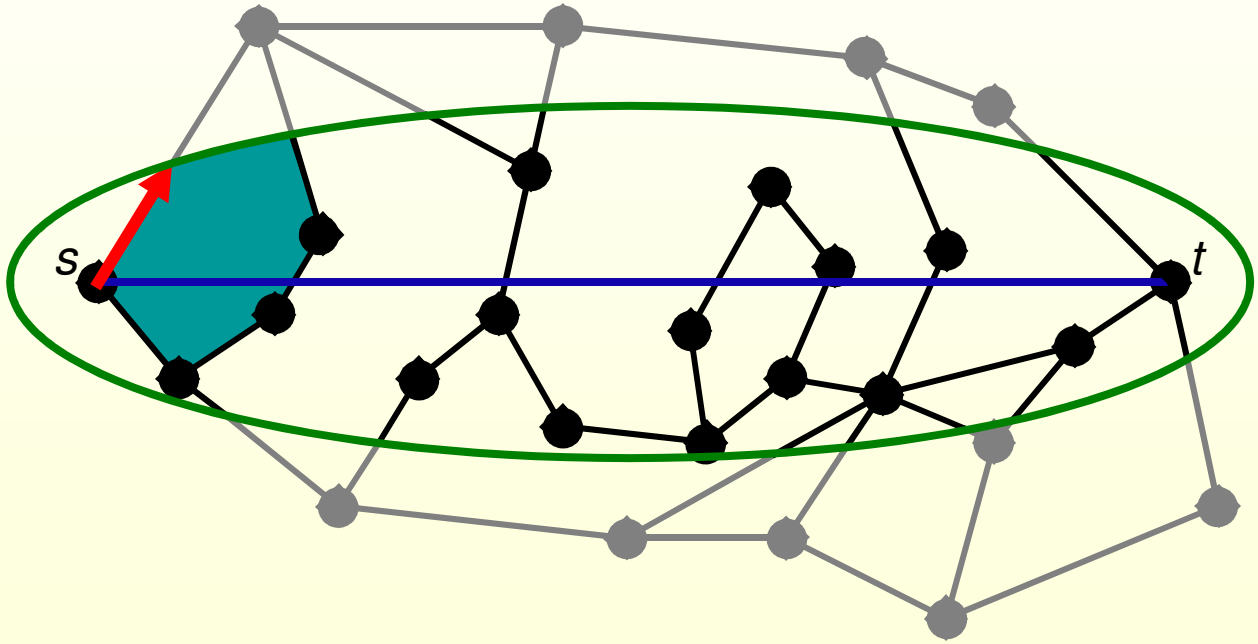
[From Kuhn et. al. ’02]

Adaptive Face Routing



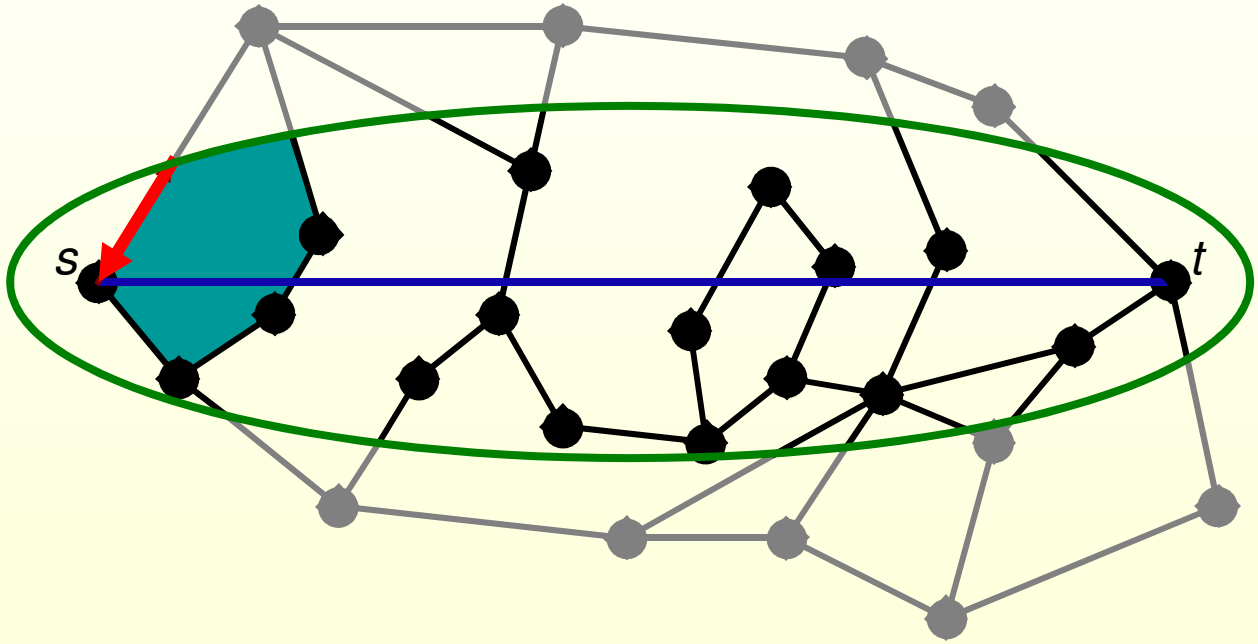
[From Kuhn et. al. ’02]

Adaptive Face Routing



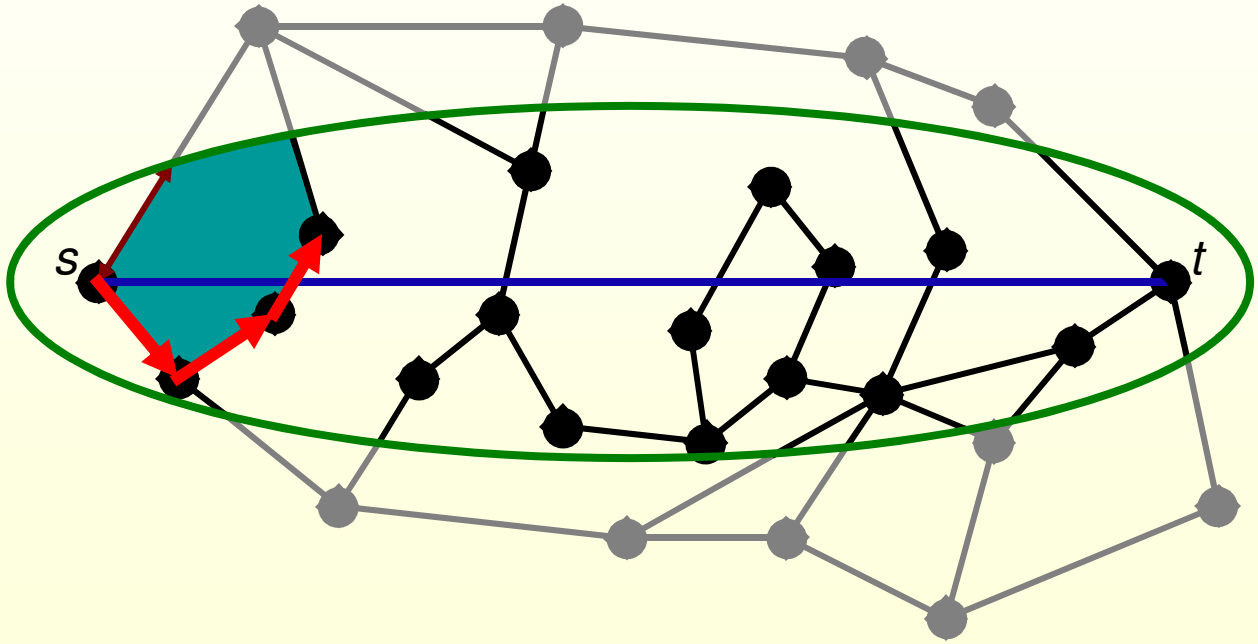
[From Kuhn et. al. ’02]

Adaptive Face Routing



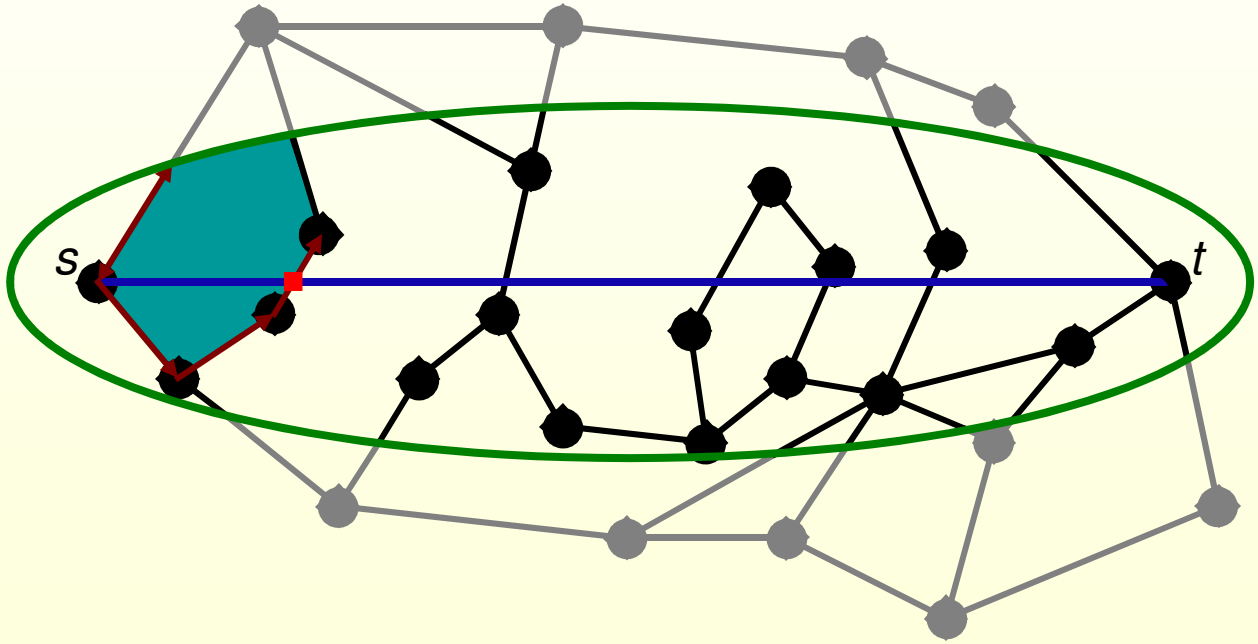
[From Kuhn et. al. ’02]

Adaptive Face Routing



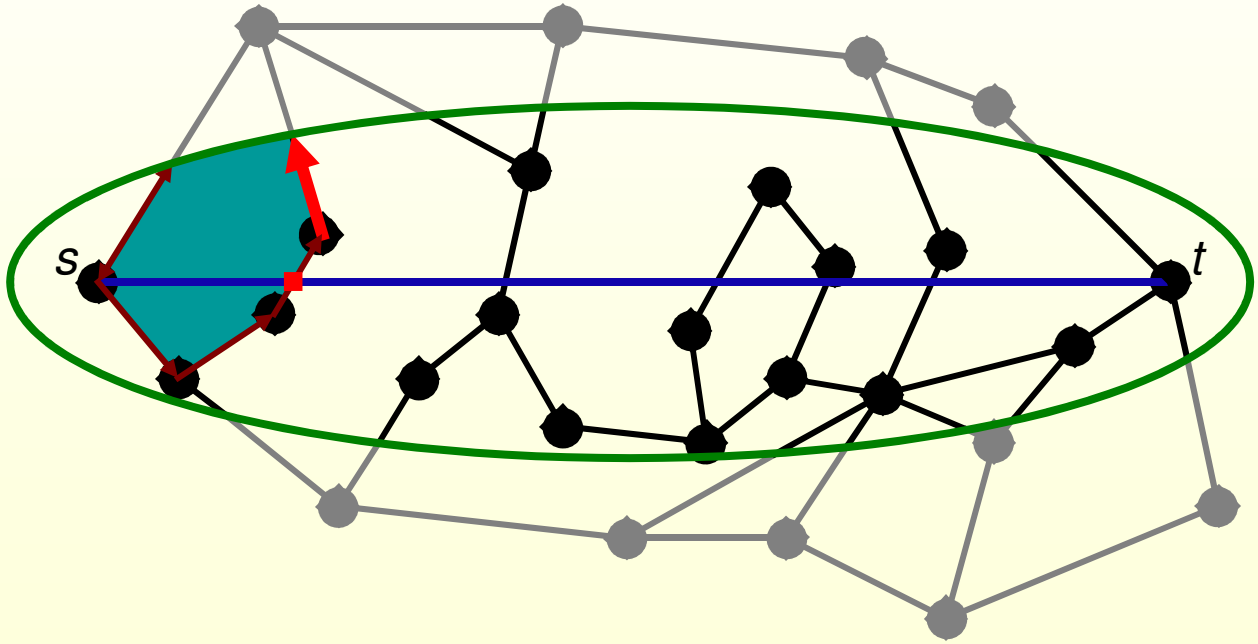
[From Kuhn et. al. ’02]

Adaptive Face Routing



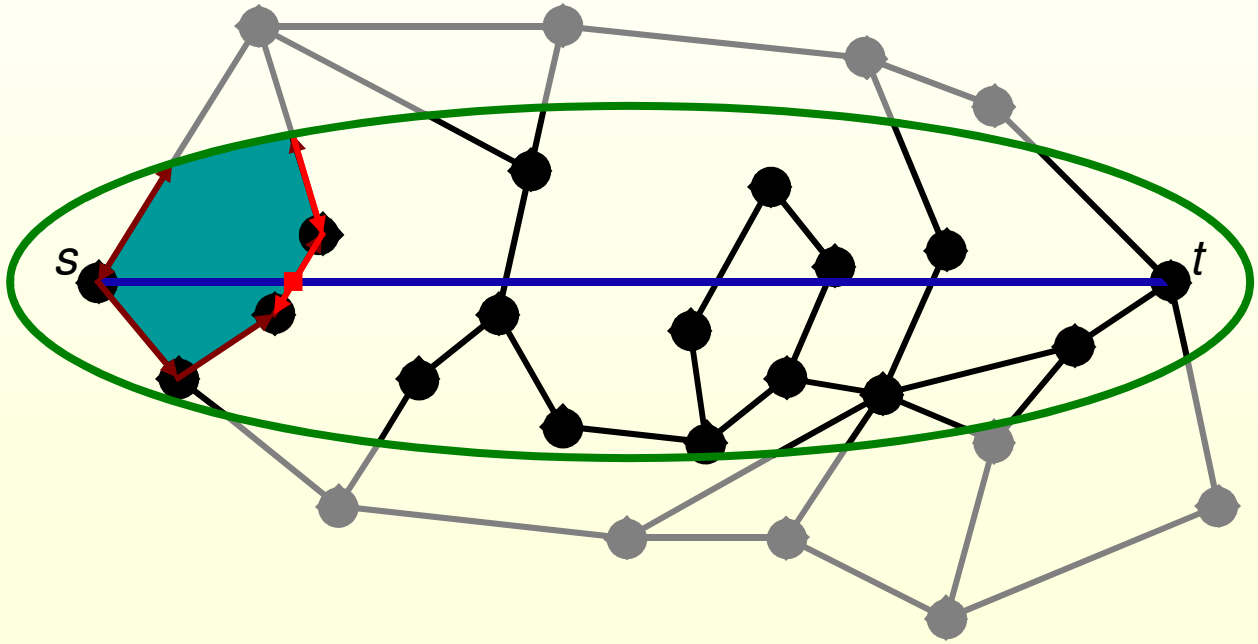
[From Kuhn et. al. ’02]

Adaptive Face Routing



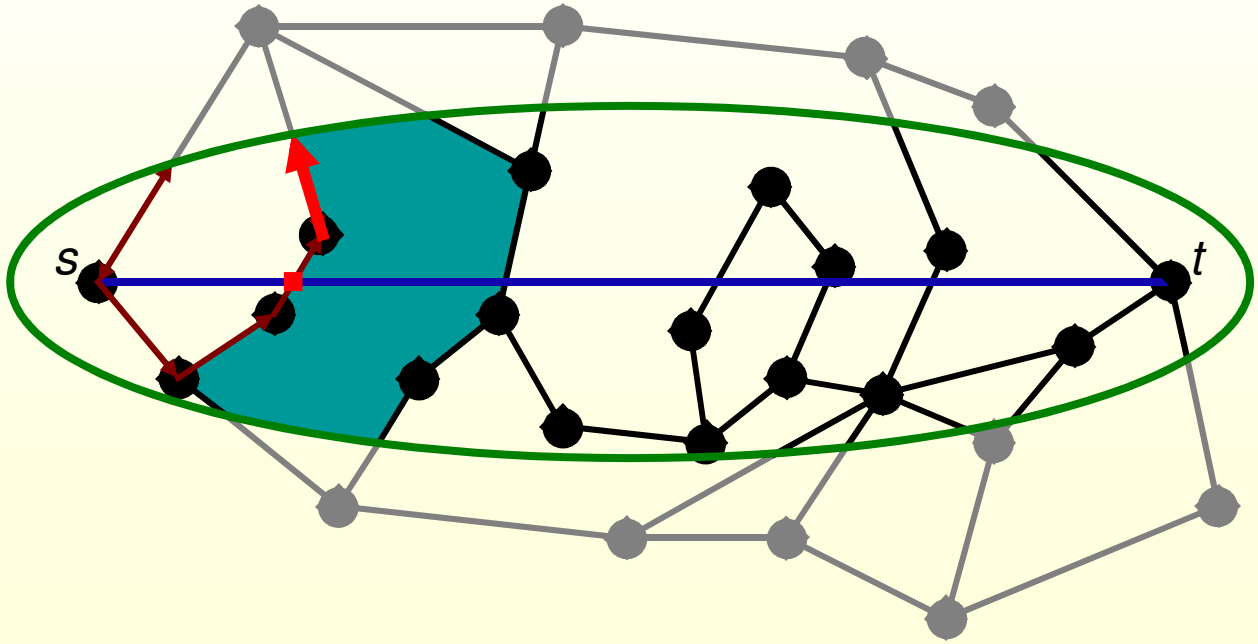
[From Kuhn et. al. ’02]

Adaptive Face Routing



[From Kuhn et. al. ’02]

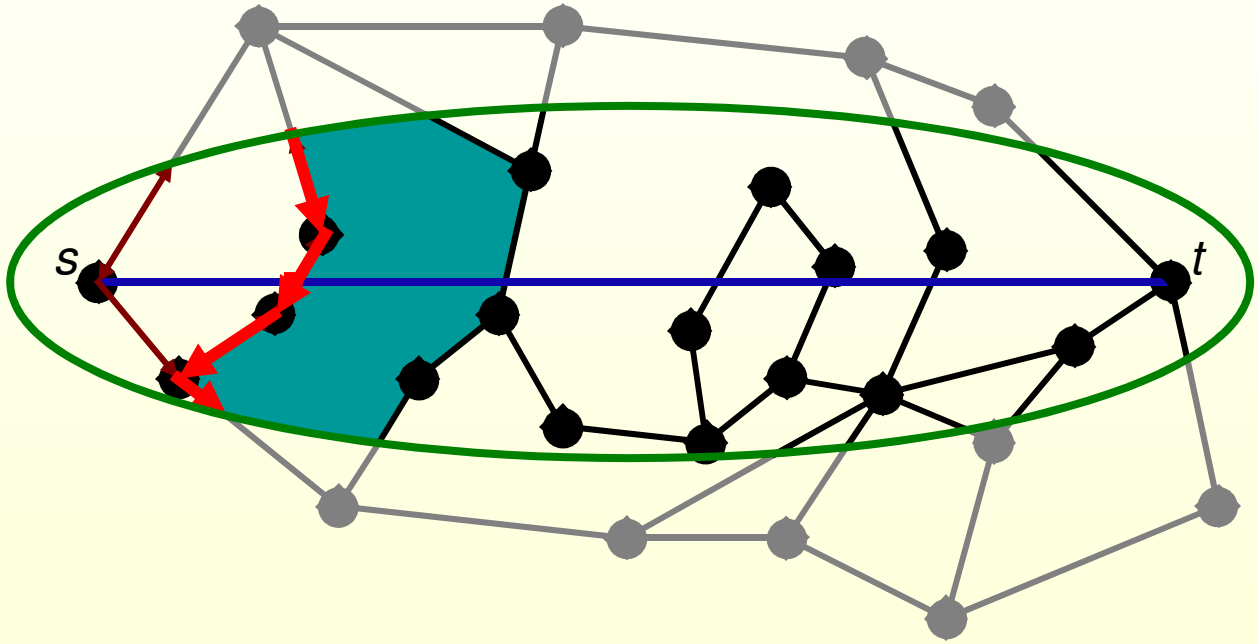
Adaptive Face Routing



[From Kuhn et. al. ’02]

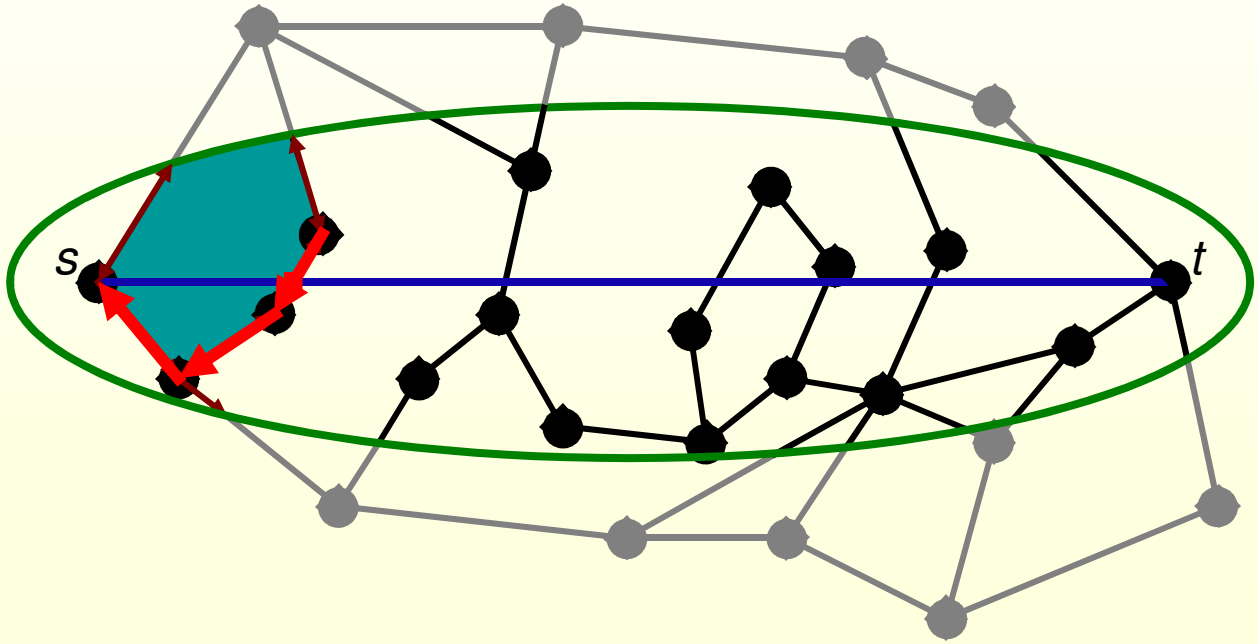
Adaptive Face Routing

Ellipse is too small, go back!



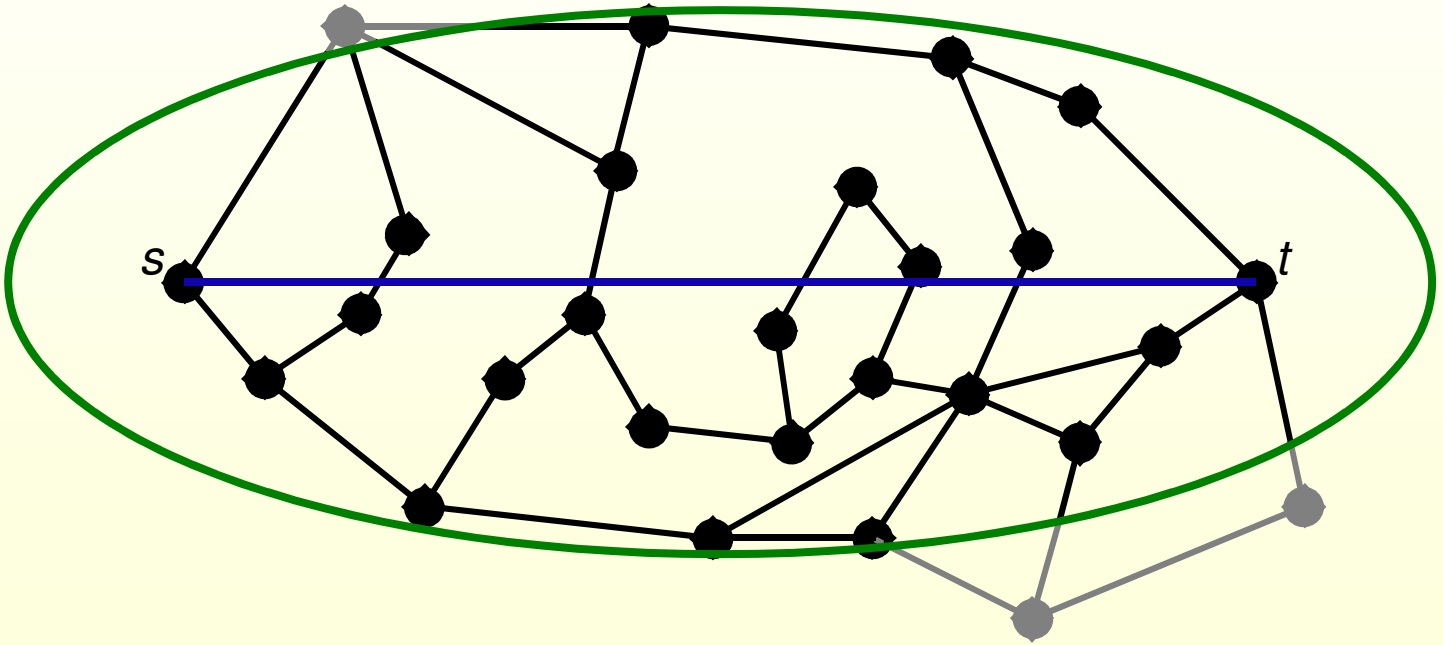
[From Kuhn et. al. ’02]

Adaptive Face Routing



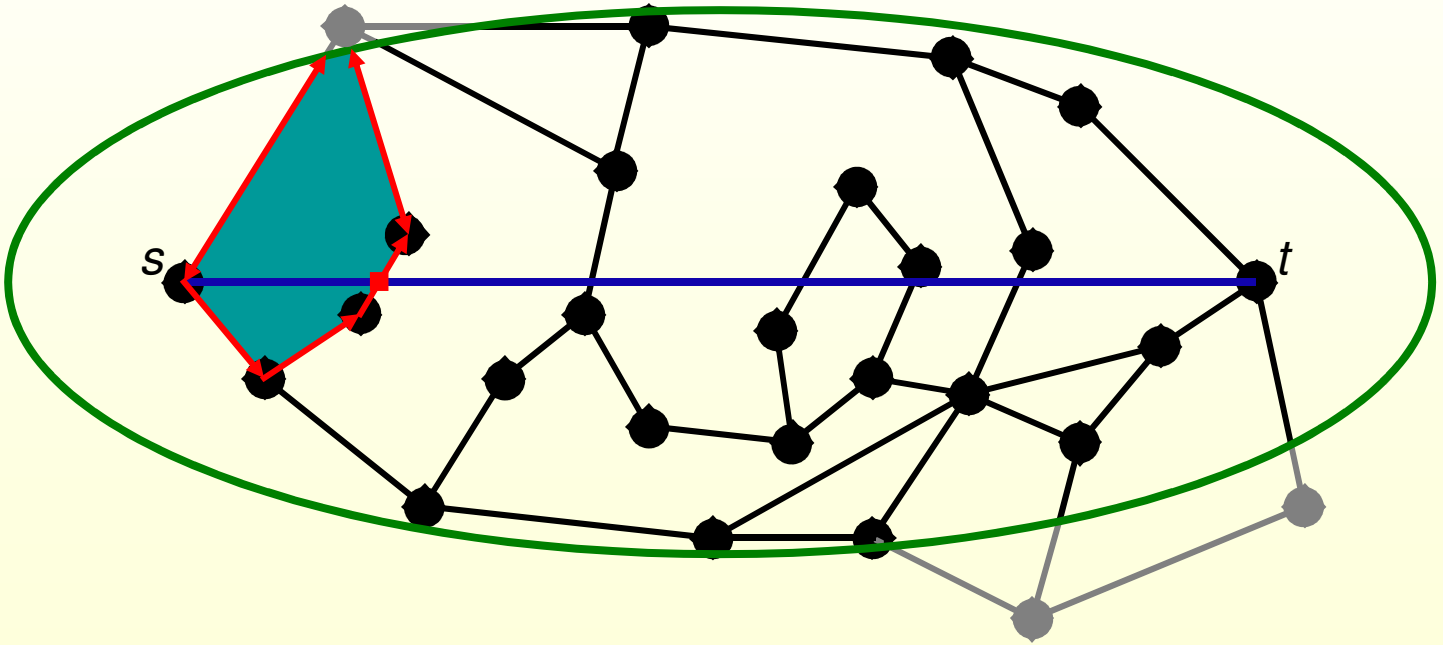
[From Kuhn et. al. ’02]

Adaptive Face Routing



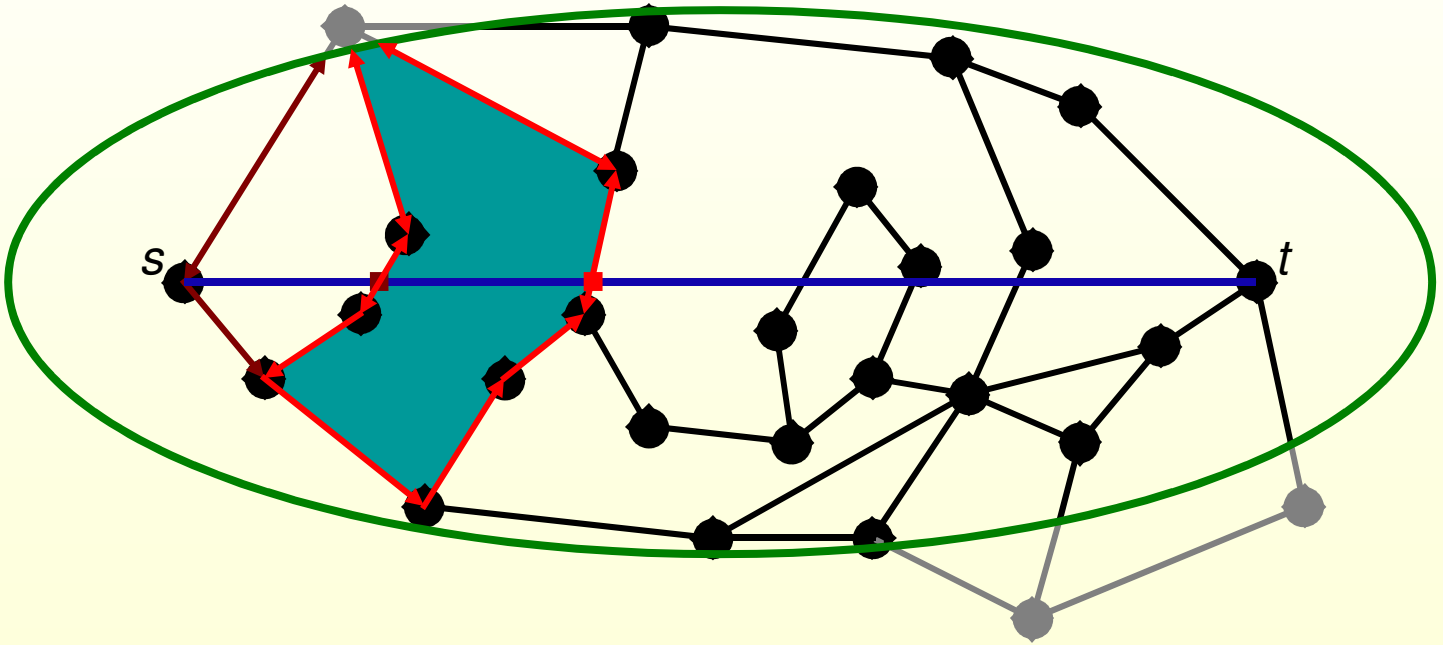
[From Kuhn et. al. ’02]

Adaptive Face Routing



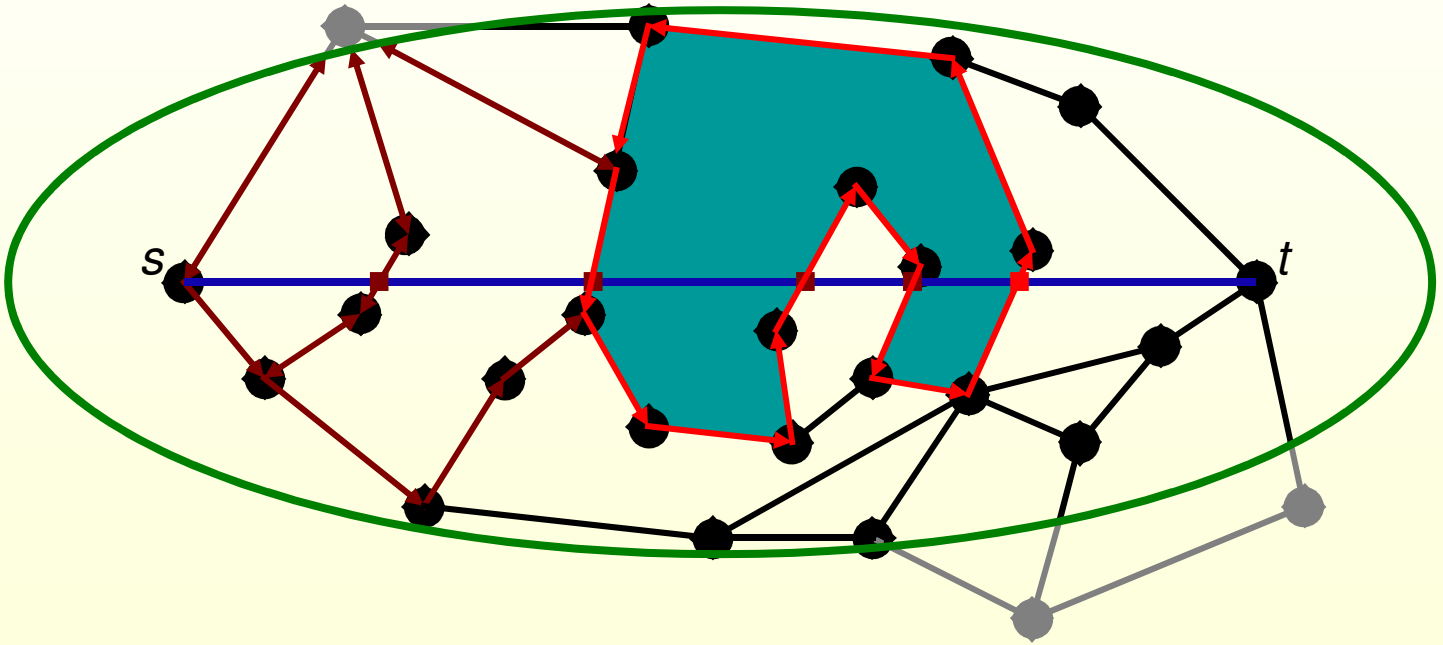
[From Kuhn et. al. ’02]

Adaptive Face Routing



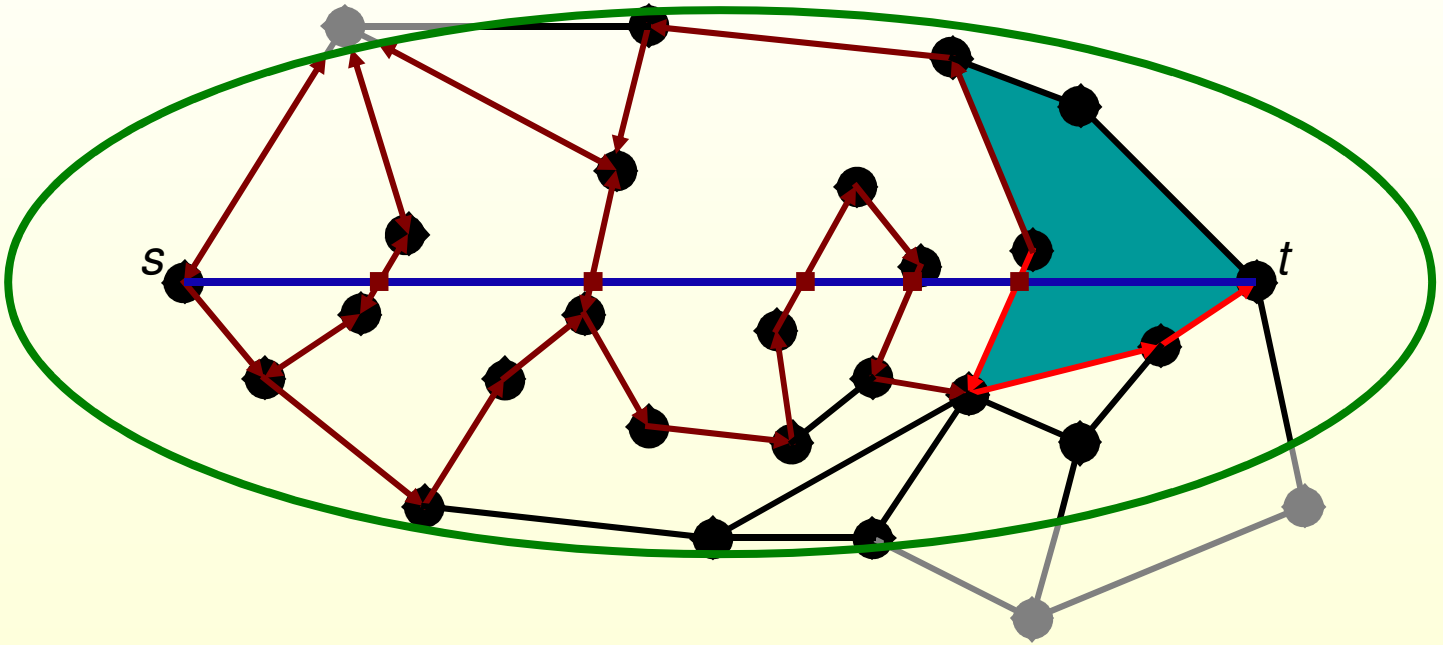
[From Kuhn et. al. ’02]

Adaptive Face Routing



[From Kuhn et. al. ’02]

Adaptive Face Routing



[From Kuhn et. al. ’02]

AFR Complexity

Theorem 1:

Let c_* be the cost (link, Euclidean, or energy) of an optimal path between two nodes on the UDG. Applying AFR on GG∩UDG then terminates with cost $O(c_*^2)$.

[From Kuhn et. al. '02]

AFR Complexity, Proof I

Lemma:

For each used ellipse $\mathcal{E}$, the cost is linear in the number of nodes in $\mathcal{E}$.

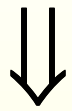
Collorary:

In the $\Omega(1)$-model, for each used ellipse $\mathcal{E}$, the cost is linear in area covered by $\mathcal{E}$.

[From Kuhn et. al. '02]

AFR Complexity, Proof II

Ellipses grow exponentially



Lemma:

The cost, AFR needs to route a packet, is linear in the area covered by the last used ellipse.

[From Kuhn et. al. '02]

AFR Complexity, Proof III

Lemma:

Using an ellipse $\mathcal{E}$, AFR finds a path from s to t iff there is such a path inside $\mathcal{E}$.

Lemma:

All paths of (Euclidean) length smaller or equal to c are inside an ellipse whose area is in $O(c^2)$.

[From Kuhn et. al. '02]

AFR Complexity, Proof IV

All the lemmas together now prove Theorem 1.

[From Kuhn et. al. ’02]

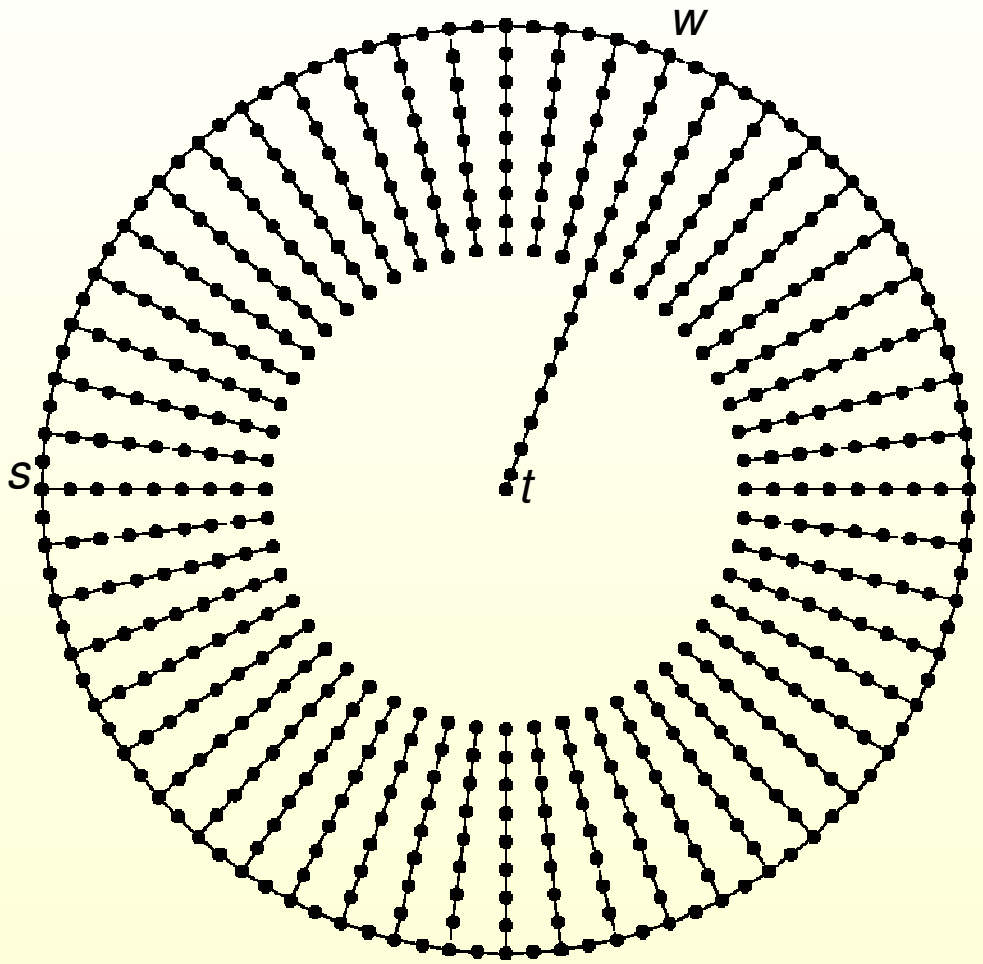
Lower Bound

Theorem 2:

Let c_* be the cost (link, Euclidean, or energy) of an optimal path between two nodes on a UDG $\mathcal{G}$. For each geographic routing algorithm, there is a graph for which the cost is $\Omega(c_*^2)$.

[From Kuhn et. al. '02]

Lower Bound, Proof



Cost of optimal path:

$$c_* \leq (\pi + 1)R$$

Exp. cost for an algorithm $\mathcal{A}$:

$$\mathsf{E}[c(\mathcal{A})] \geq \frac{\pi R}{2} \cdot \Theta(R)$$
$$= \Theta(R^2)$$

[From Kuhn et. al. '02]

Main Theorem

Theorem 3:

On the Unit Disk Graph in the $\Omega(1)$-model, AFR is asymptotically optimal.

(follows directly from Theorems 1 and 2)

[From Kuhn et. al. '02]

Remarks

$\Omega(1)$ restriction can be dropped by clustering

- works fine for link and Euclidean distance (still $\Theta(c_*{}^2)$)
- For energy, it can be shown that the cost of a geometric routing alg. cannot be bounded by a function of c_* alone.

[From Kuhn et. al. '02]

But in Practice

- Many more variations on face routing are possible
- But, greedy geographic forwarding works much better in practice than all facing routing variants
- So let's combine the two:
  - Use greedy geographic forwarding as long as we make progress
  - Switch to a face routing protocol only when we are stuck
  - Go back to greedy geographic forwarding as soon as possible
- The mix, however, can destroy the optimality of face routing
- This can be fixed, but it takes some care ...

Variations in Geographic Routing

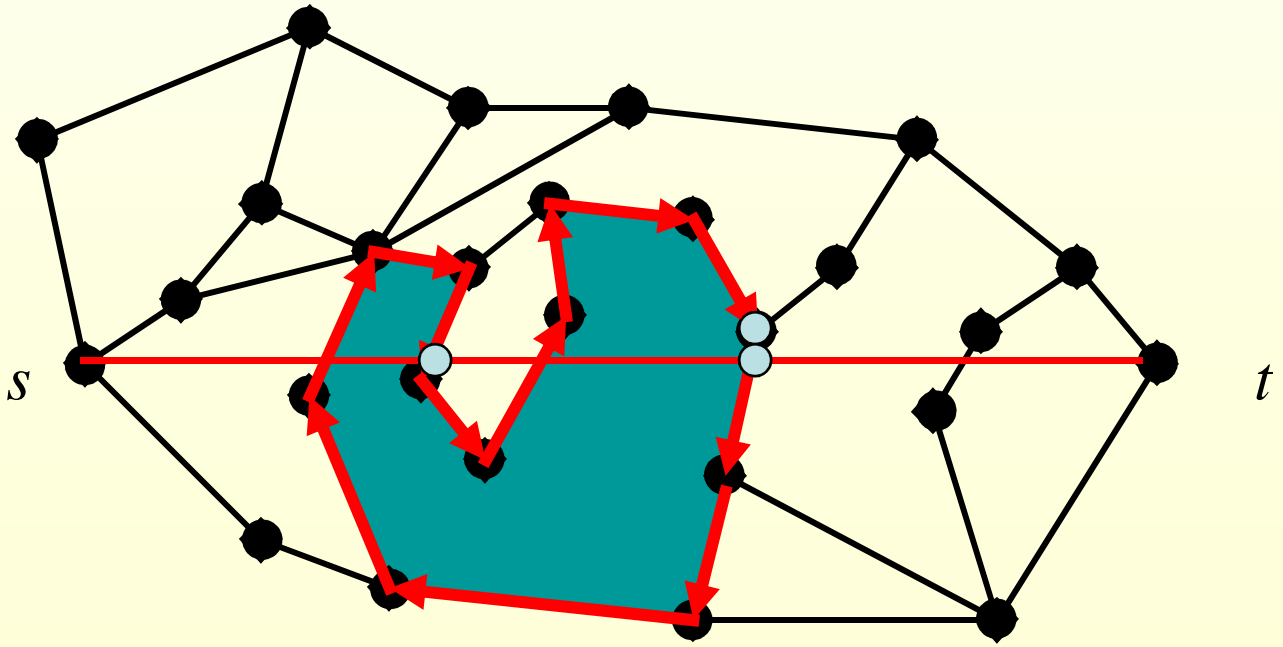
Kleinrock et al.	Various 1975ff	MFR et al.	Geometric Routing proposed
Kranakis, Singh, Urrutia	CCCG 1999	Face Routing	First correct algorithm
Bose, Morin, Stojmenovic, Urrutia	DialM 1999	GFG	First average-case efficient algorithm (simulation but no proof)
Karp, Kung	MobiCom 2000	GPSR	A new name for GFG
Kuhn, Wattenhofer, Zollinger	DialM 2002	AFR	First worst-case analysis. Tight $\Omega(c^2)$ bound.
Kuhn, Wattenhofer, Zollinger	MobiHoc 2003	GOAFR	Worst-case optimal and average-case efficient, percolation theory
Kuhn, Wattenhofer, Zhang, Zollinger	PODC 2003	GOAFR+	Improved GOAFR for average case, analysis of cost metrics

GPSR = Greedy Perimeter Stateless Routing

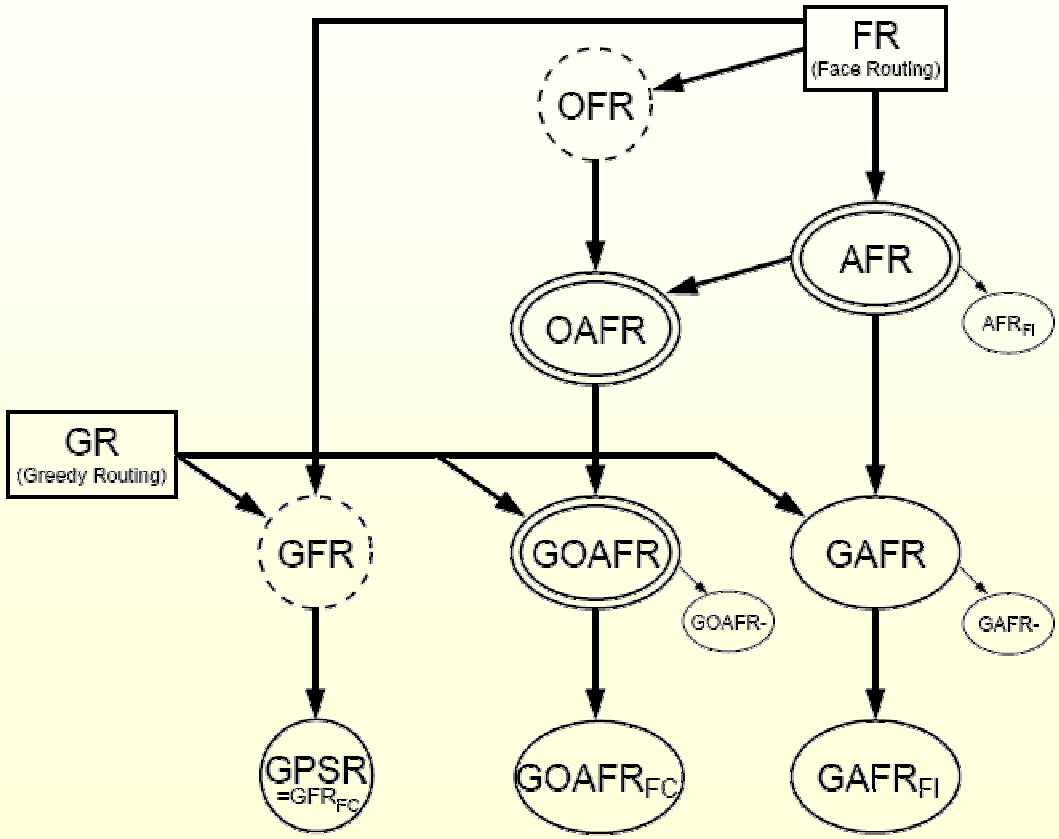
[From Kuhn et. al. '03]

Face Transition Possibilities

- In literature there are four ways of switching faces:
  1. Best intersection (AFR)
  2. First intersection (GPSR, GFG)
  3. Closest node other face routing (GOAFR+)
  4. Closest point other face routing



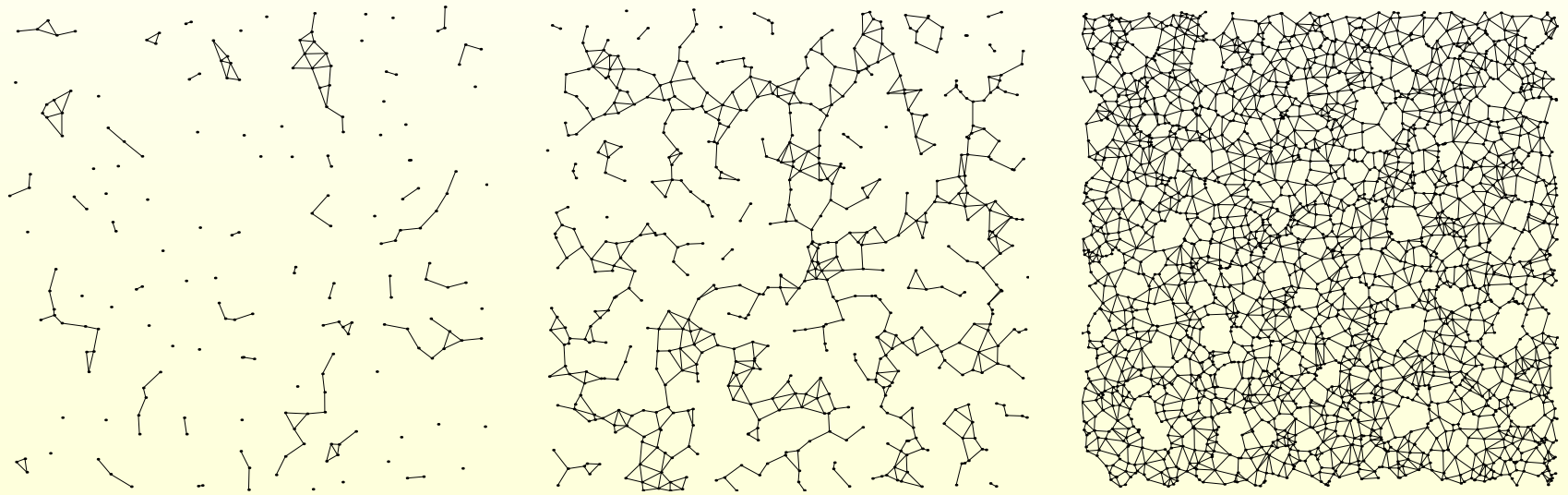
Family Tree of Geographic Routing Algorithms



[From Kuhn et. al. '03]

Average Case Performance of Routing Algorithms

Not interesting when graph is too sparse
Not interesting when graph is too dense
Critical density range ("percolation")
- Shortest path is significantly longer than Euclidean distance

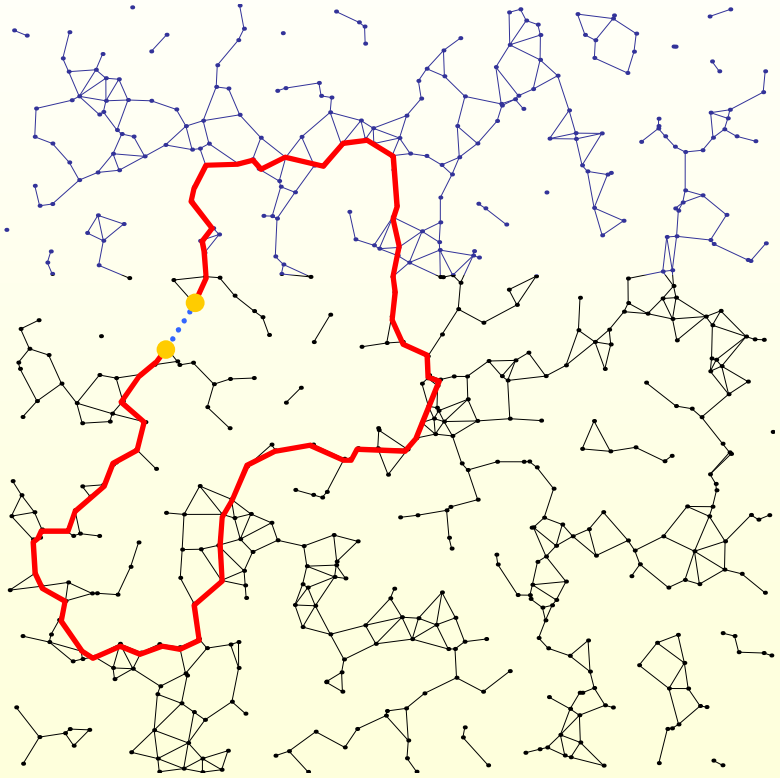


[From Kuhn et. al. '03]

Critical Density: Shortest Path vs. Euclidean Distance

Shortest path is significantly longer than Euclidean distance

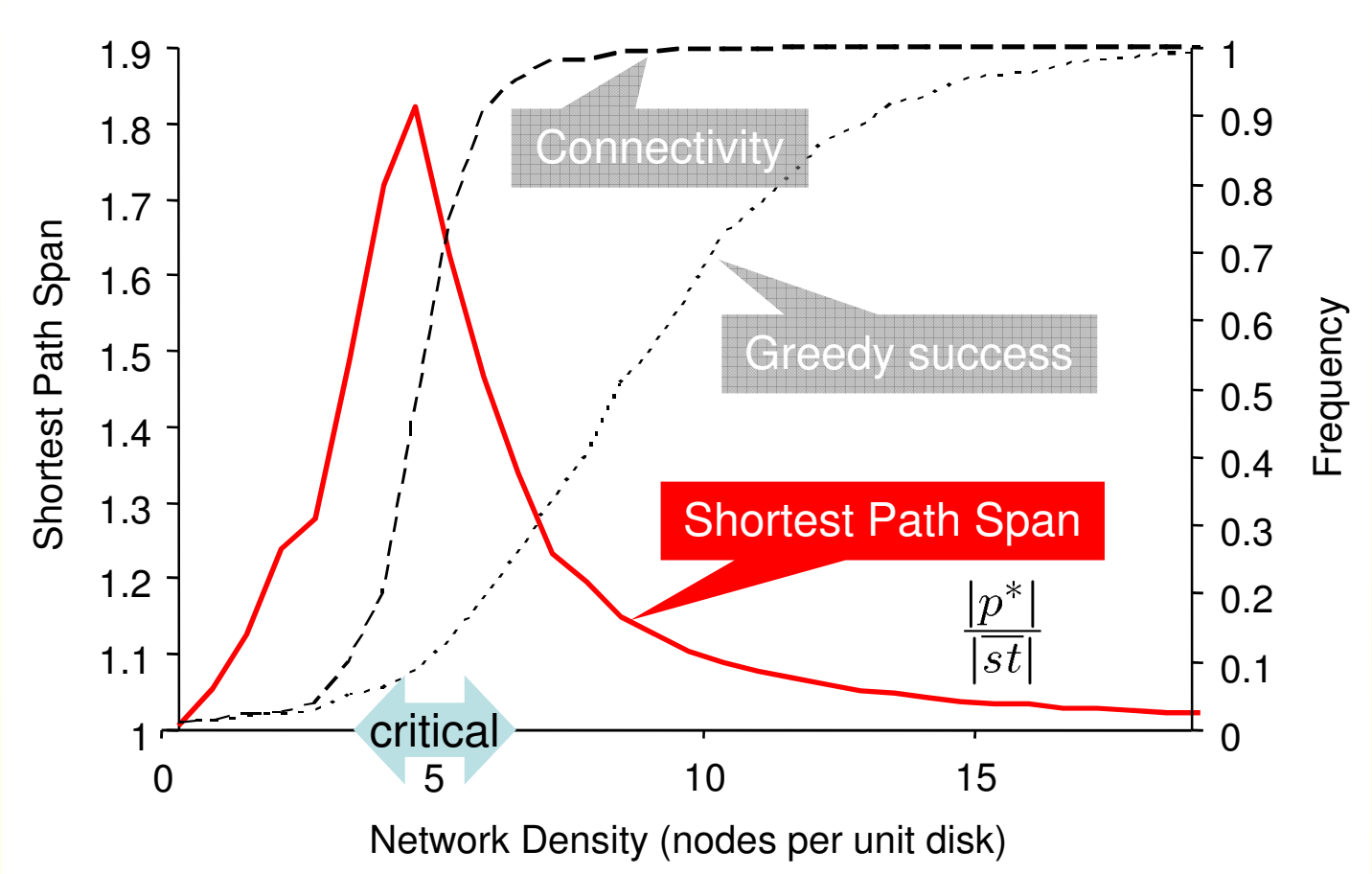
$$\frac{|p^*|}{|\overline{st}|}$$



Critical density range mandatory for the simulation of any routing algorithm (not only geometric)

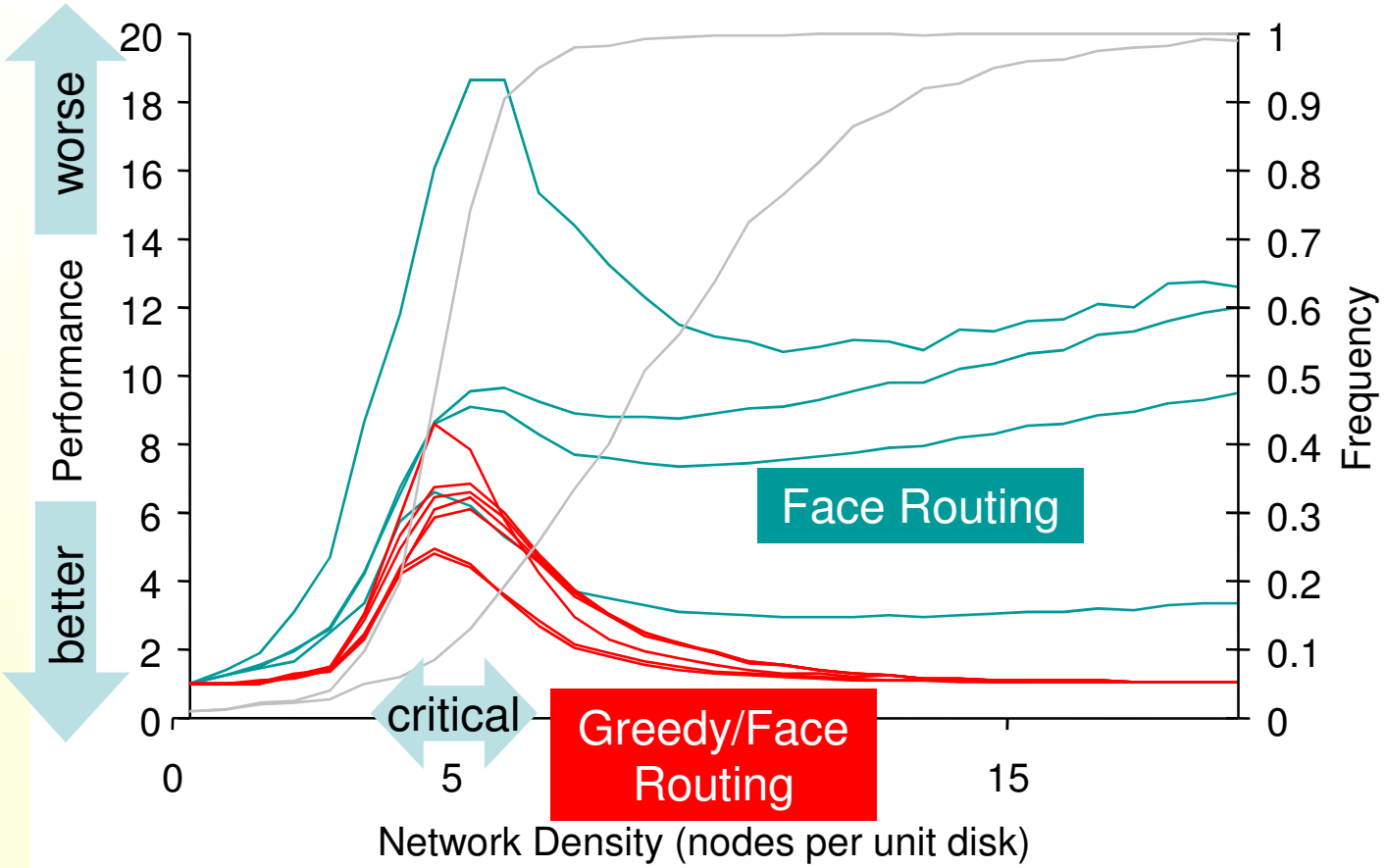
[From Kuhn et. al. '03]

Randomly Graphs: CTR



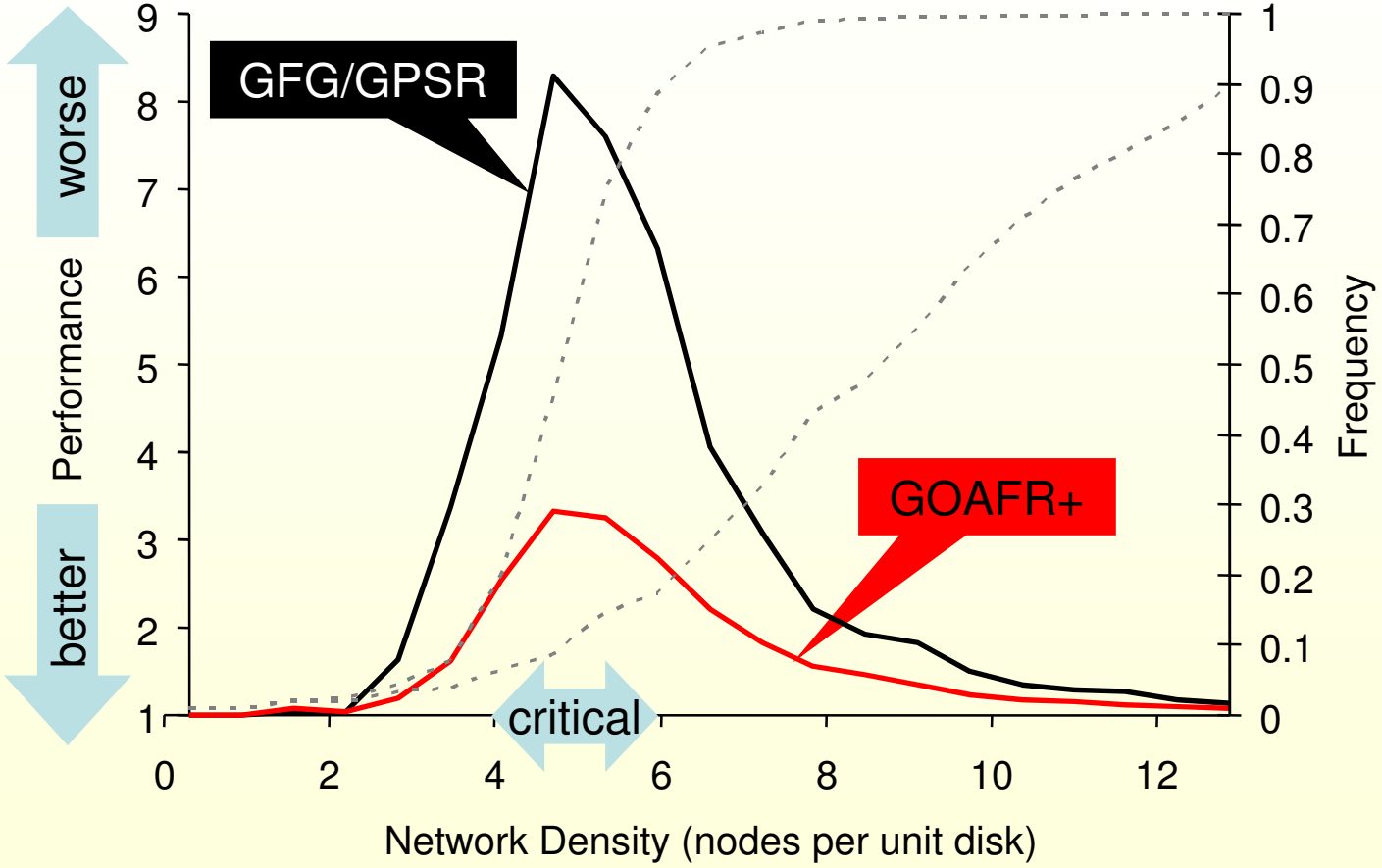
[From Kuhn et. al. '03]

Face vs. Greedy/Face



[From Kuhn et. al. '03]

Simulation on Randomly Generated Graphs



[From Kuhn et. al. '03]

Geographic Routing Summary

	Correct Routing	Worst-Case Optimal	Avg-Case Efficient	Comprehensive Simulation
Greedy Routing (MFR)			(✓)	
Face Routing	✓			
GFG/GPSR	✓		✓	
AFR	✓	✓		
GOAFR	✓	✓	✓	✓

[From Kuhn et. al. '03]

Class Issues

- Please sign up for the class on Axess
- If you are thinking of doing your own project, look in the proceedings of
  - IPSN, Int. Conference on Information Processing for Sensor Networks
  - SenSys, Conference on Embedded Networked Sensor Systems
  - DCOSS, Conference on Distributed Computing of Sensor Systems
- A project proposal (one or two pages) is due on Wednesday, October 10.

Class Web Site

- http://graphics.stanford.edu/courses/cs321-07-fall
- For protected areas, log in as
  - user = CS321
  - password = WSN

The End