Universidad Politécnica
de Madrid

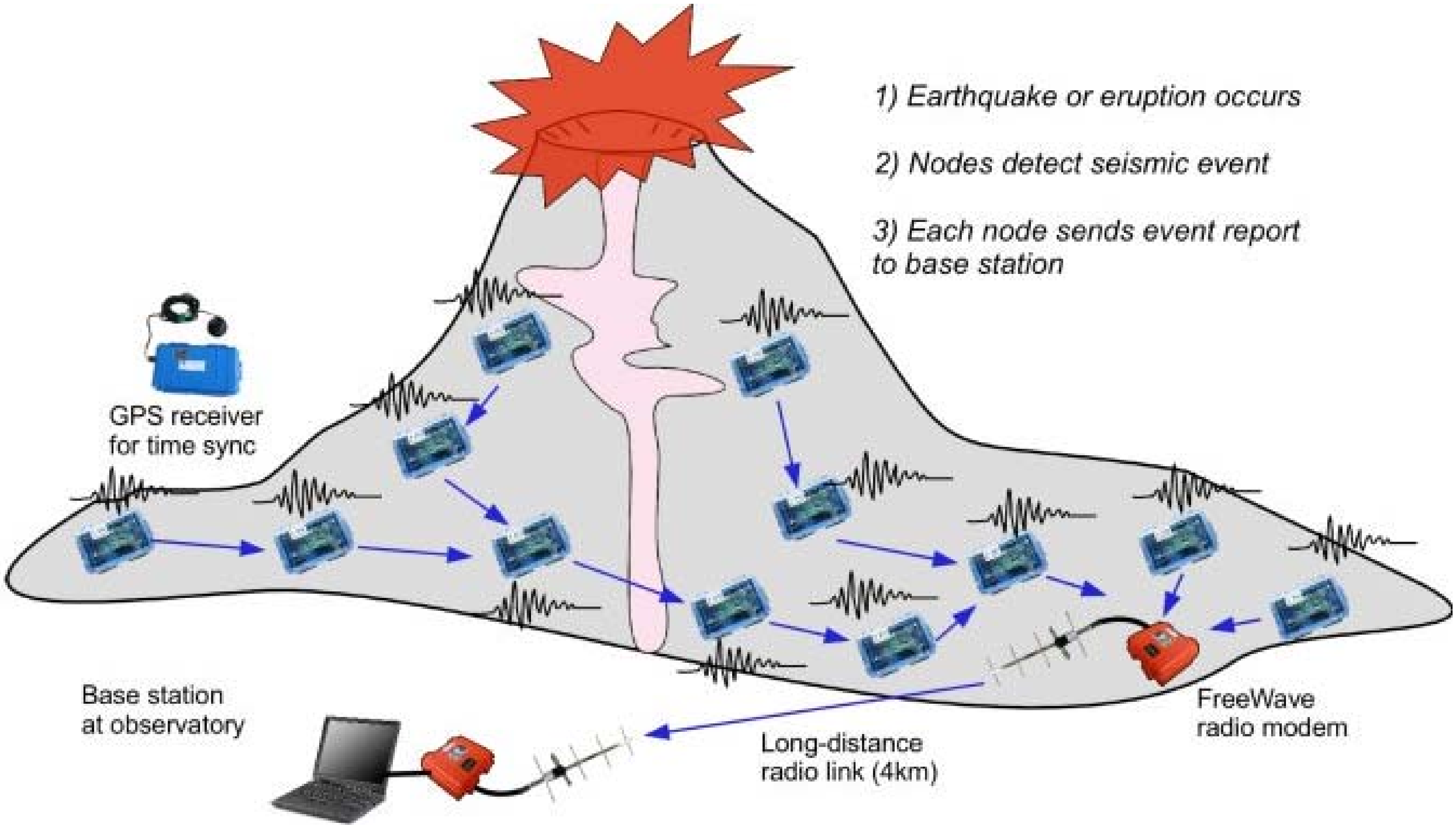
Geometric networks

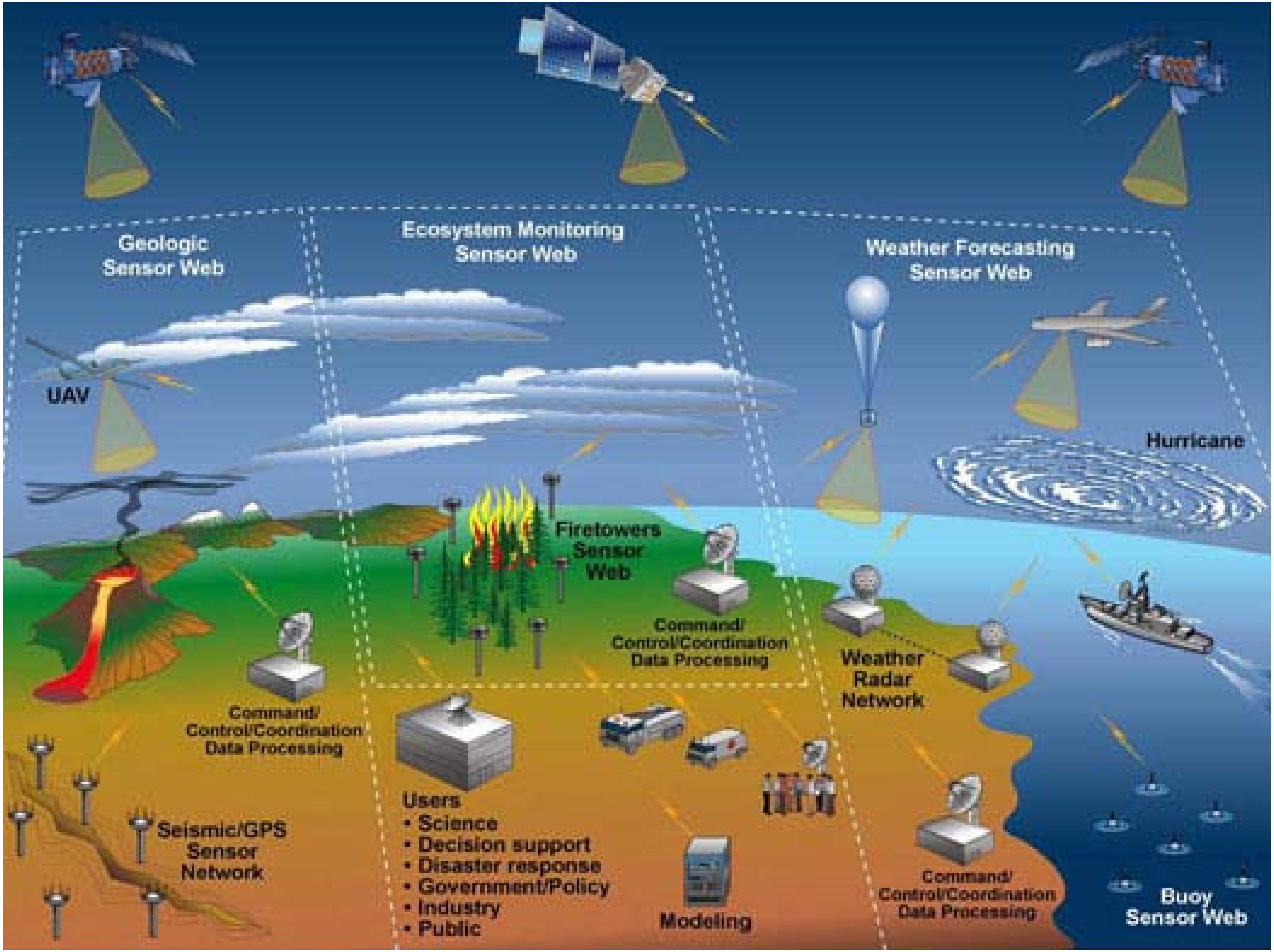
Global problems, local solutions

Gregorio Hernández
UPM

UPC Computational Geometry Seminar, November 17, 2010

Wireless networks

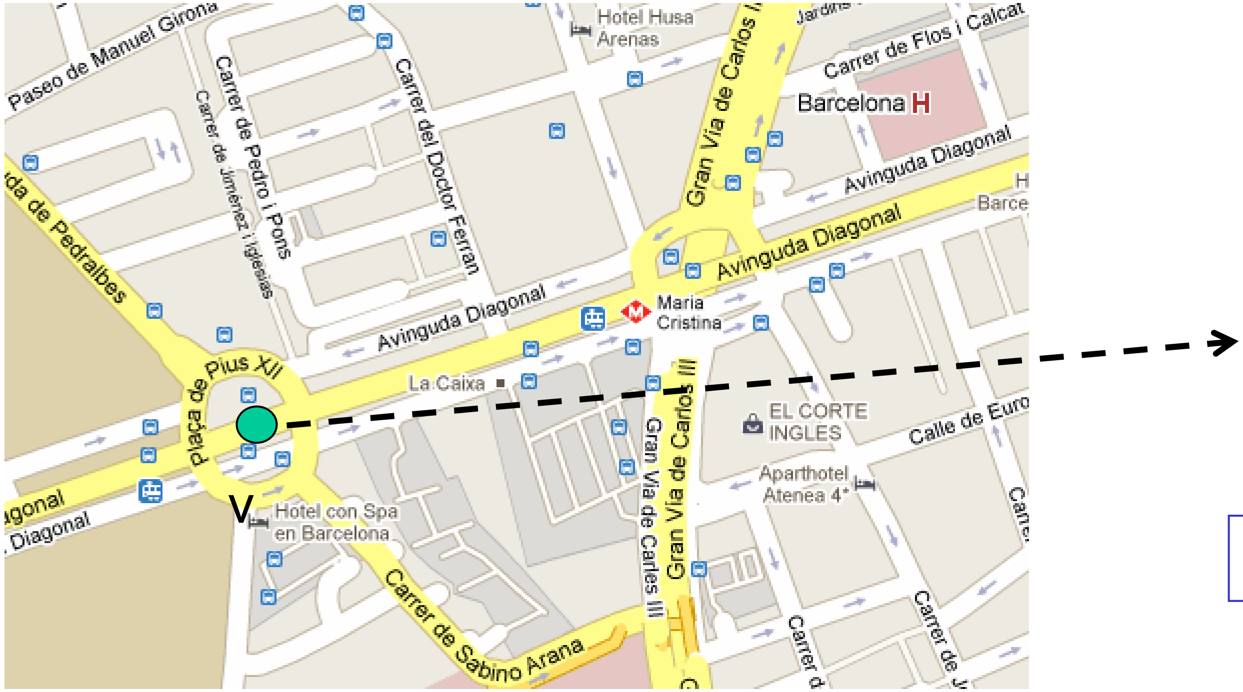



Geologic Sensor Web
Ecosystem Monitoring Sensor Web
Weather Forecasting Sensor Web
UAV
Hurricane
Firetowers Sensor Web
Command/ Control/Coordination Data Processing
Command/ Control/Coordination Data Processing
Weather Radar Network
Seismic/GPS Sensor Network
Users
• Science
• Decision support
• Disaster response
• Government/Policy
• Industry
• Public
Modeling
Command/ Control/Coordination Data Processing
Buoy Sensor Web

…other networks

…other networks

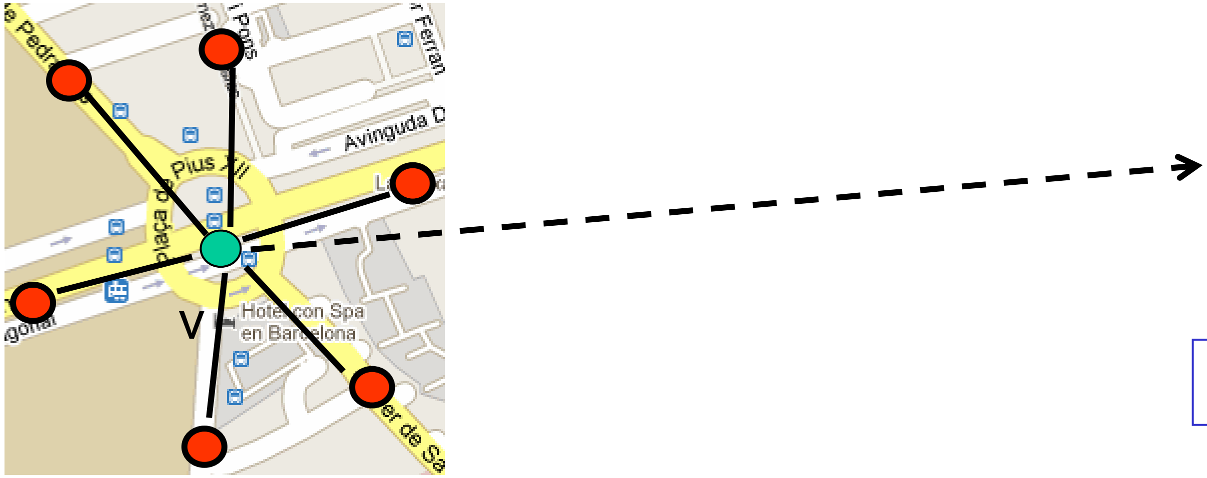
A tourist in Barcelona



Sagrada Familia

…other networks

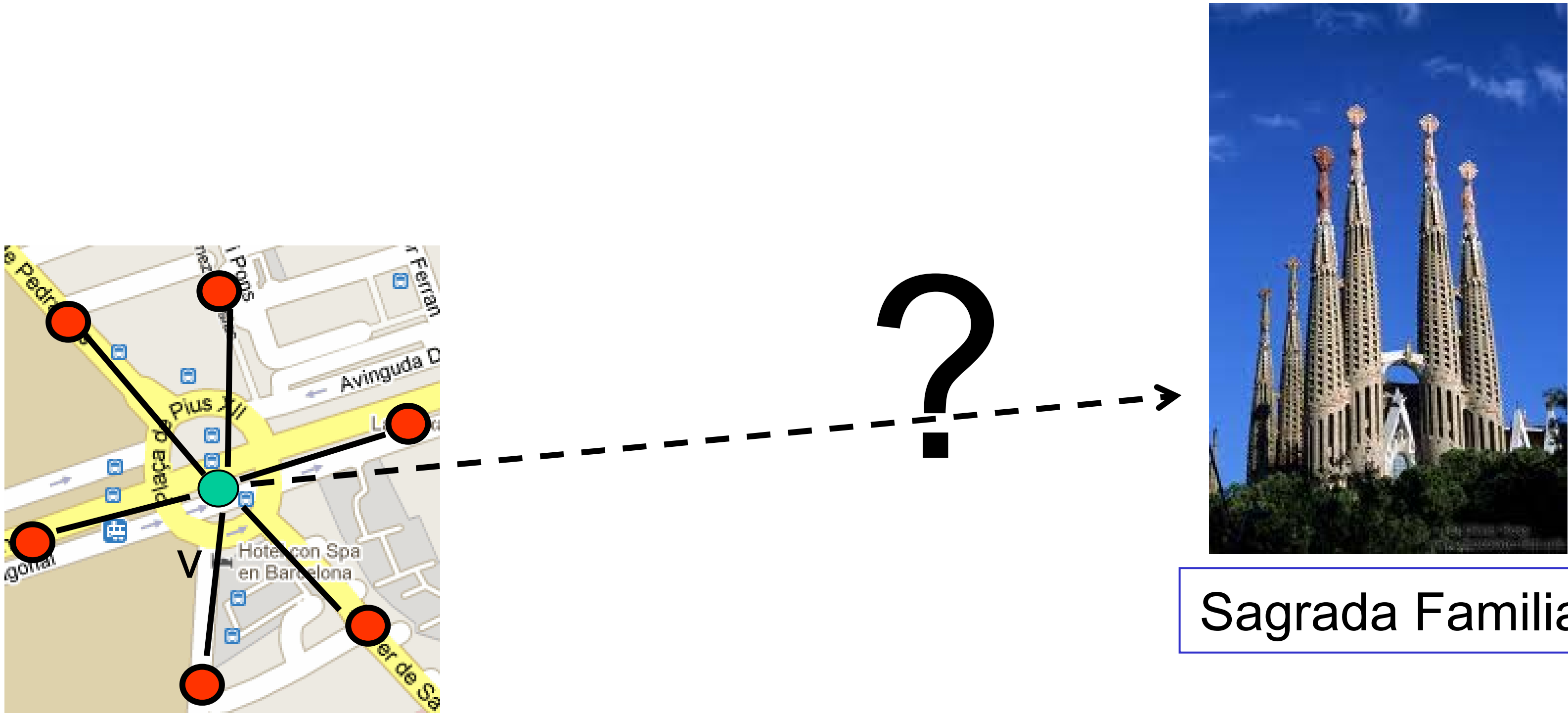
A tourist in Barcelona



Sagrada Familia

NO MAP
Local information (coordinates of v, target, and neighbors N(v))
Limited memory allocation
Ecologically sound algorithms

…other networks



Sagrada Familia

How can we move in
an unknown network?

Wireless networks

With the sensors ...

No map!

No paths!

Wireless networks

(1) How to organize the network?

(2) How to send messages?

(3) How to recover, store and index the data of the network?

Wireless networks

(1) How to organize the network? **DESIGN**

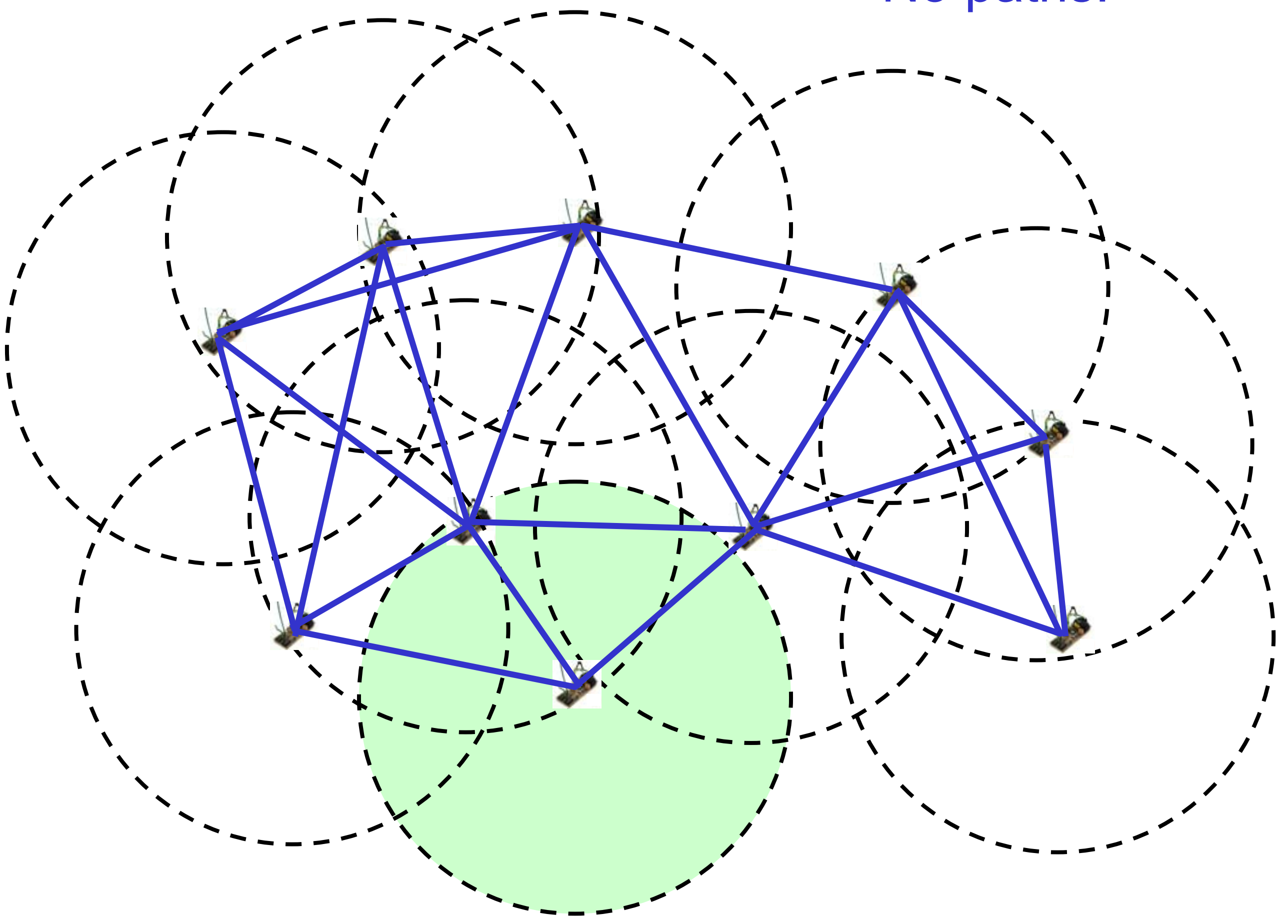
(2) How to send messages? **ROUTING**

Wireless networks

With the sensors ...

No map!

No paths!

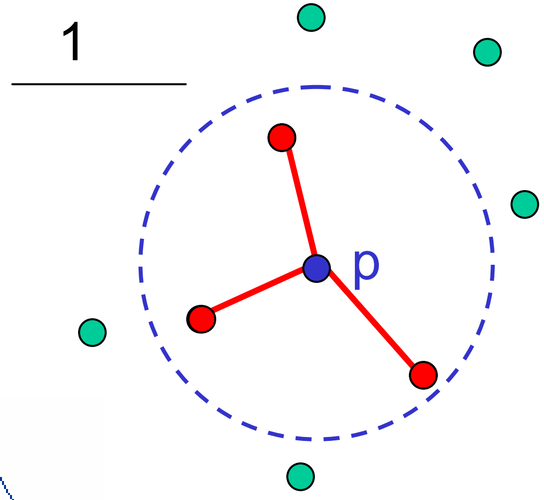


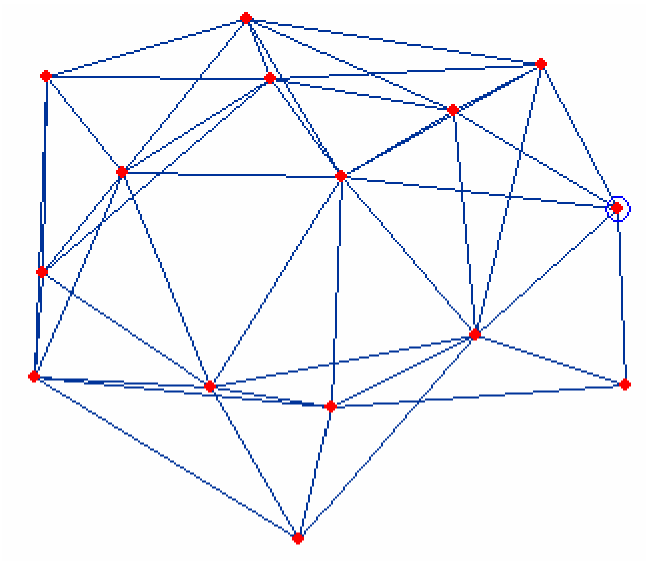
Wireless networks

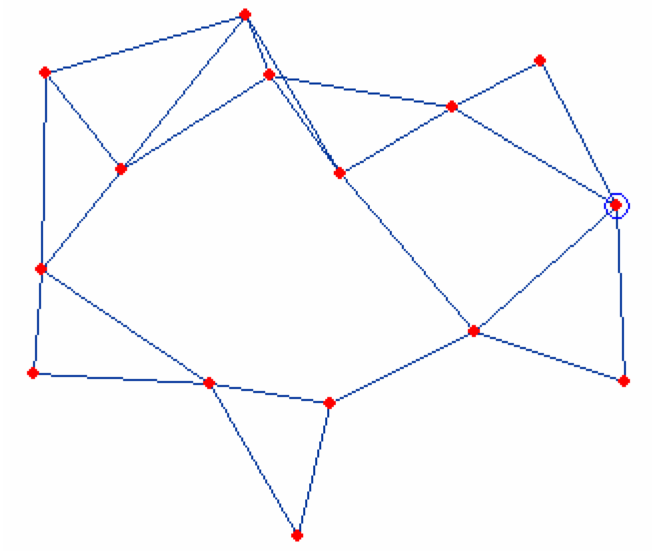
NETWORK MODEL

Unit Disk Graph UDG

Neighbors of p are the points of S contained in the circle of center p and radius 1







Wireless networks

LOCAL ALGORITHM

For each node we know:

(1) Position of u
(2) Neighbors of u (until distance k)
(3) The subyacent graph is UDG(S)

Wireless networks

ROUTING PROBLEM

PROBLEM 1

Let G be a geometric planar network. Is there a deterministic algorithm that allows an agent *A* standing at a vertex *s* to travel to a vertex *t* of *G* under the following conditions:

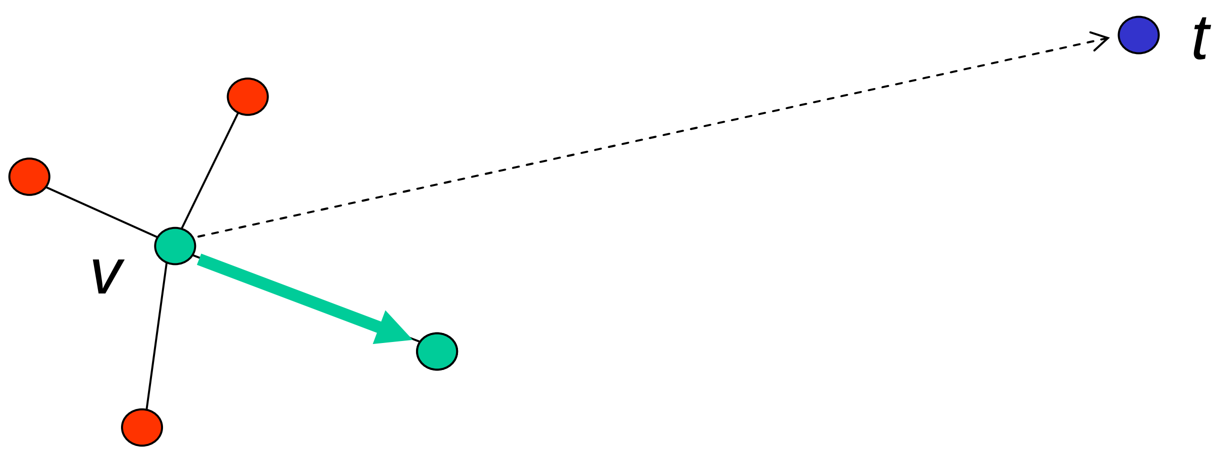
(1) *A* has a constant amount of memory; that is, at any point in time *A* knows the position of *s* and *t*, and the positions of a constant number of nodes in *G*.

(2) When the agent visits a vertex *x* of *G*, it can use the list of vertices (and their positions) adjacent to x

(3) *A* is not allowed to leave any marks along its way

Wireless networks **ROUTING**

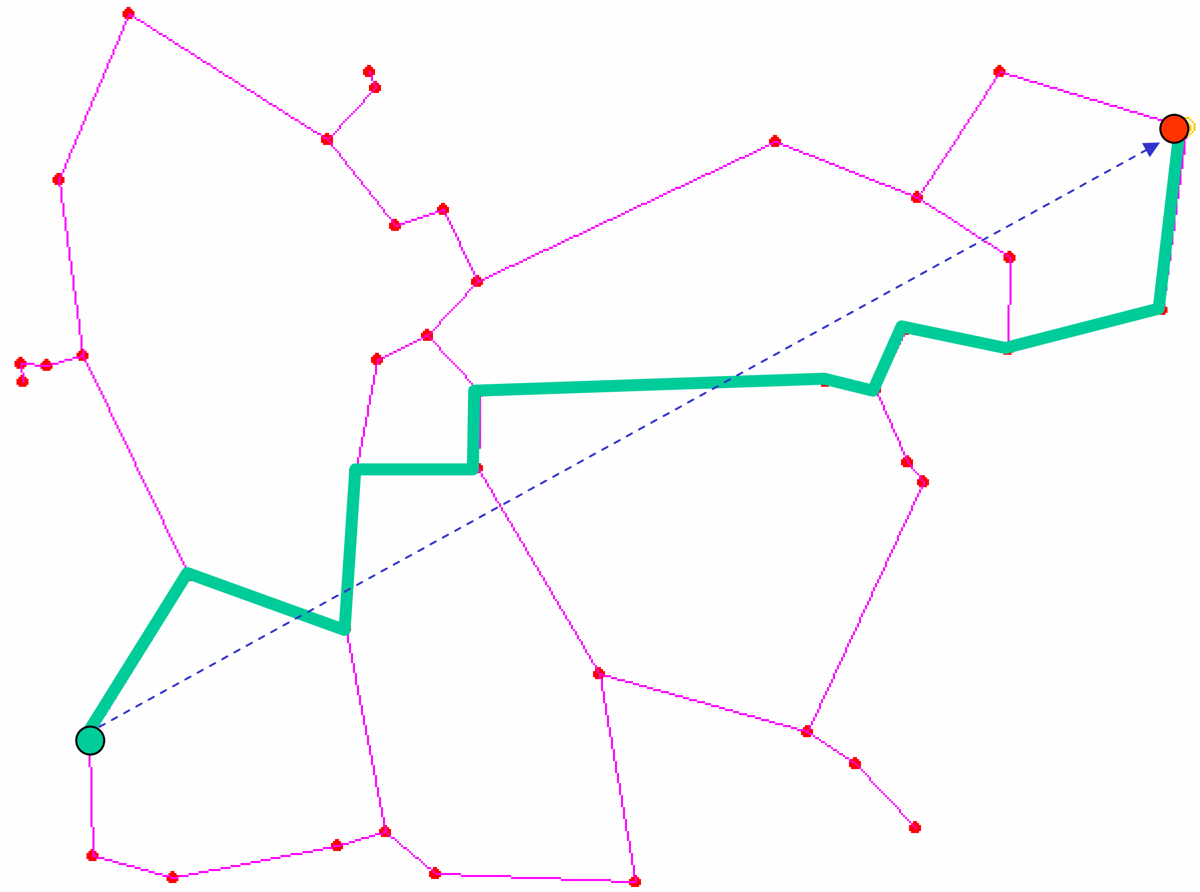
Greedy Routing



We greedily route to the neighbor which is closest to the target

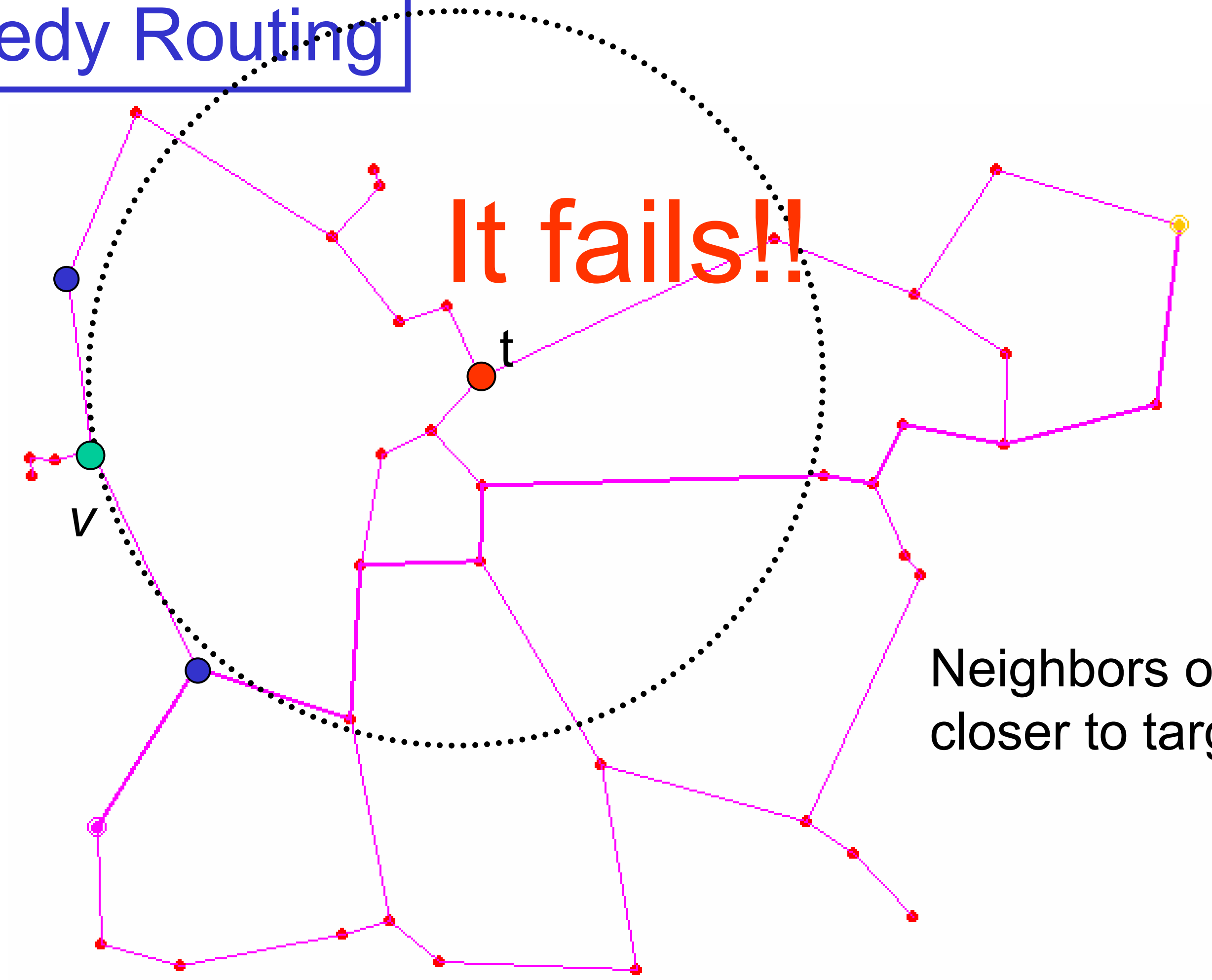
Wireless networks **ROUTING**

Greedy Routing



Wireless networks **ROUTING**

Greedy Routing



Neighbors of *v* are not closer to target

Wireless networks **ROUTING**

Greedy Routing

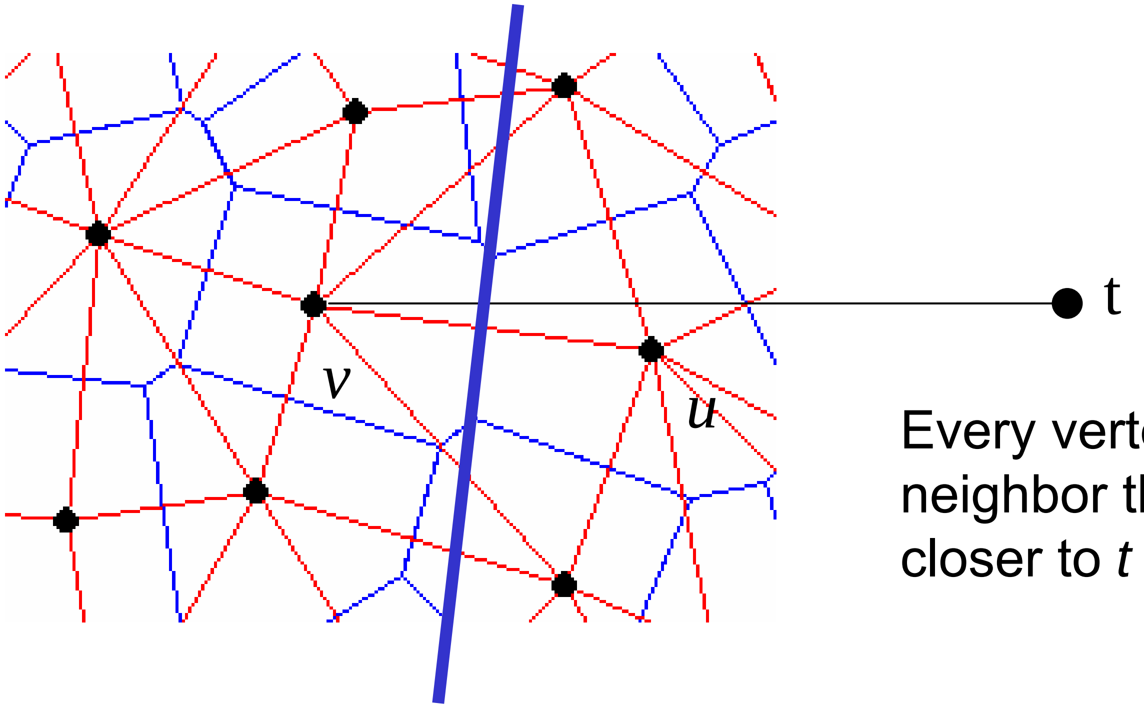
Memoryless algorithm

- Fails on some graphs
- Fails on some triangulations
- Always works for Delaunay Triangulations

Wireless networks **ROUTING**

Greedy Routing

Always works for Delaunay Triangulations

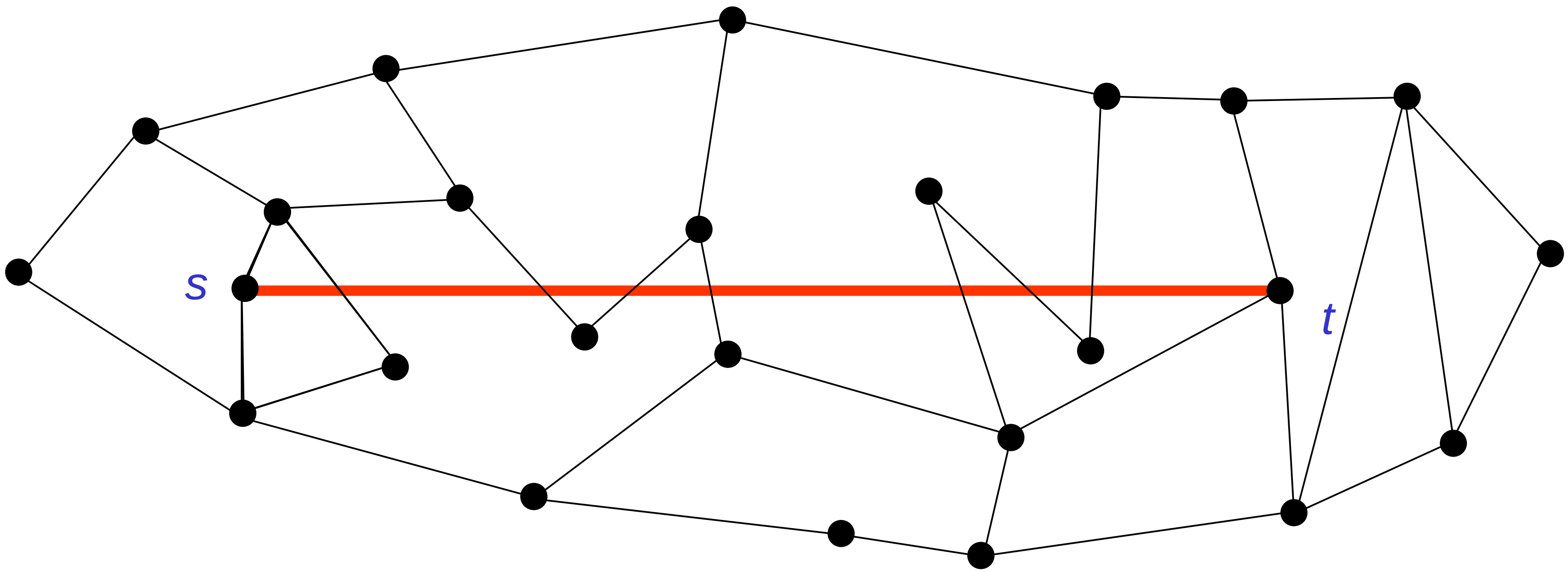


Every vertex *v* of **DT** has a neighbor that is strictly closer to *t* than *v*

Wireless networks **ROUTING**

Face Routing

Kranakis, Singh, Urrutia, '99

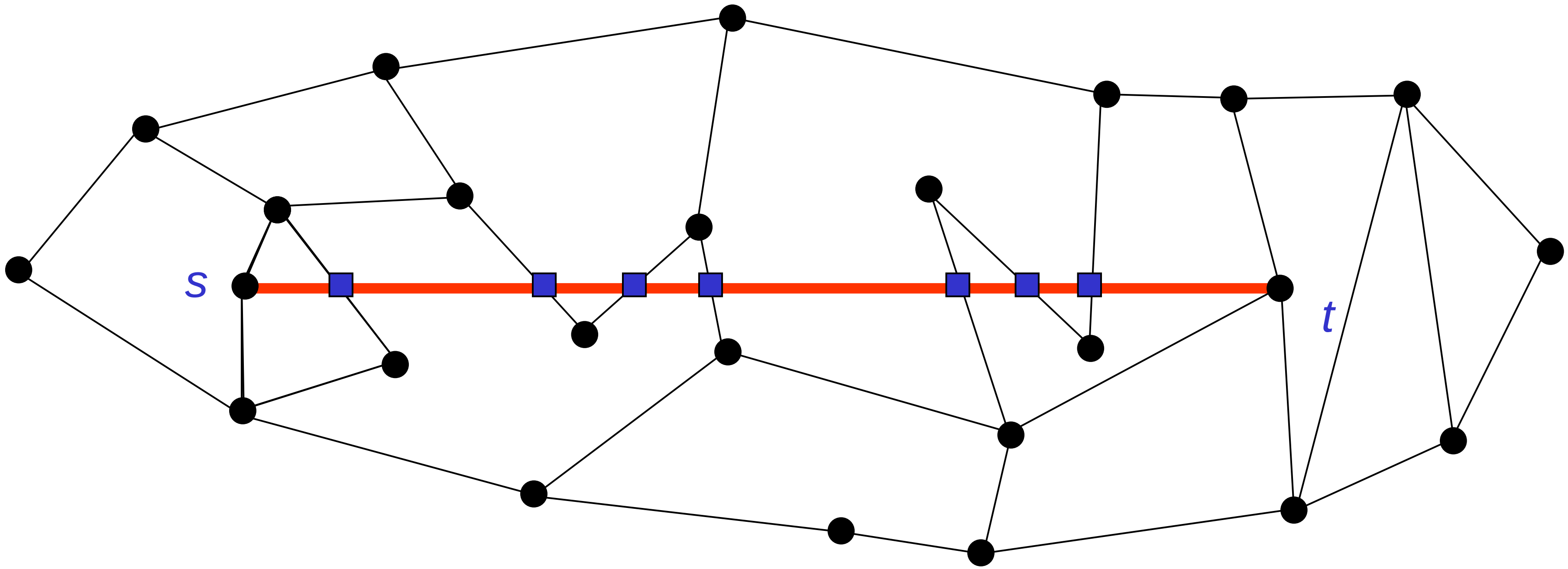


Route along the boundaries of the faces that lie on the source-target line st

Wireless networks **ROUTING**

FACE ROUTING

1. Let F be the face incident to the source *s*, intersected by line *s*,*t*



2. Explore the boundary of F; remember the point *p* where the boundary intersects with (*s*,*t*) which is nearest to *t*.
   Go back to *p*, switch the face and repeat step 2 until you hit the target *t*

Wireless networks **ROUTING**

FACE ROUTING

Theorem
Face routing terminates on any simple planar graph in O(n) steps, where n is the number of the nodes in the network

Proof:
It is straightforward to deduce that we reach the the destination *t*
We can order the faces that intersect the (*s,t*) line, therefore we never visit a face twice.
Each edge is in at most two faces, therefore each edge is visited at most 4 times.
Since a simple planar graph has at most 3n–6 edges, the algorithm terminates in O(n) steps.

FACE ROUTING

Theorem
Face routing terminates on any simple planar graph in O(n) steps, where n is the number of the nodes in the network

Constant memory algorithm

Vertex of the face closest to target
Last vertex reached

ROUTING PROBLEM

FACE ROUTING

PROBLEM 1

Let G be a geometric planar network. Is there a deterministic algorithm that allows an agent *A* standing at a vertex *s* to travel to a vertex *t* of *G* under the following conditions:

(1) *A* has a constant amount of memory; that is, at any point in time *A* knows the position of *s* and *t*, and the positions of a constant number of nodes in *G*.

(2) When the agent visits a vertex *x* of *G*, it can use the list of vertices (and their positions) adjacent to x

(3) *A* is not allowed to leave any marks along its way

Wireless networks **ROUTING**

ROUTING PROBLEM

FACE ROUTING

PROBLEM 1

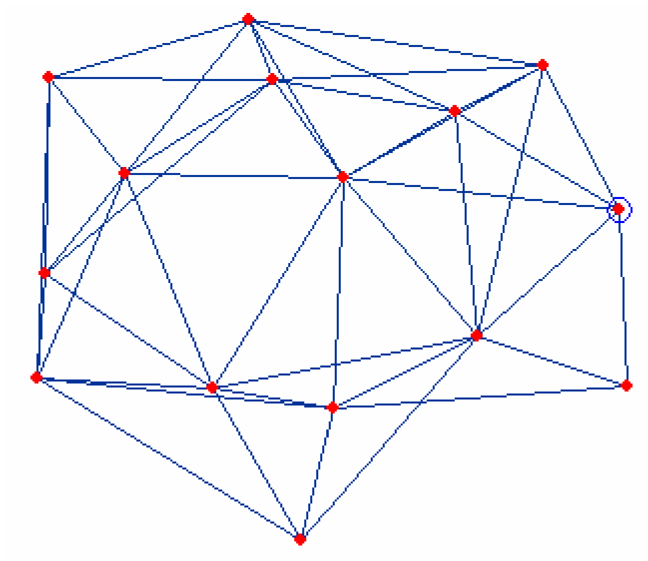
Let G be a geometric **planar** network. Is there a deterministic algorithm that allows an agent *A* standing at a vertex *s* to travel to a vertex *t* of *G* under the following conditions:

(1) A has a constant amount of memory; that is, at any point in time A knows the position of s and t, and the positions of a constant number of nodes in G.

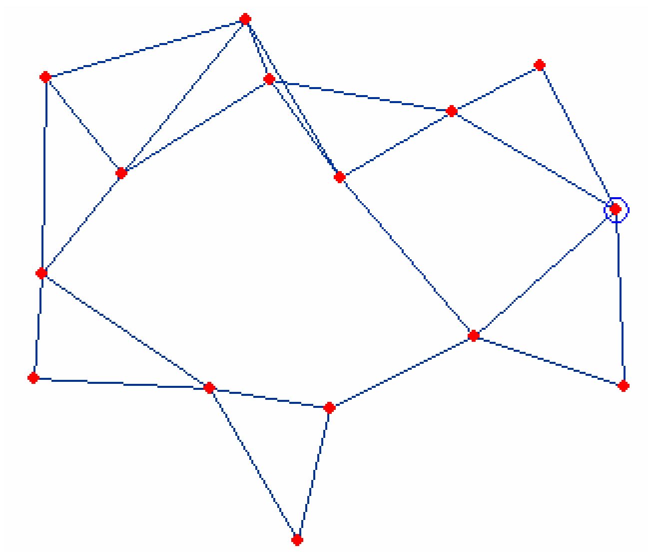
(2) When the agent visits a vertex *x* of *G*, it can use the list of vertices (and their positions) adjacent to x

(3) *A* is not allowed to leave any marks along its way

Wireless networks



Unit Disk Graph UDG



Unit Disk Graph, UDG, canNOT be a planar graph

Wireless networks

LOCAL ALGORITHM FOR ROUTING

PROBLEM 2

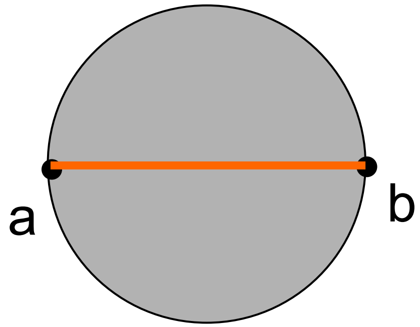
Given a UDG network N, find a local algorithm to extract a planar subgraph H such that if N is connected, then the subgraph H is also connected.

Solving problem 2 and using face routing in the resulting planar subgraph would inmediately give an on-line local algorithm for routing in UDG networks

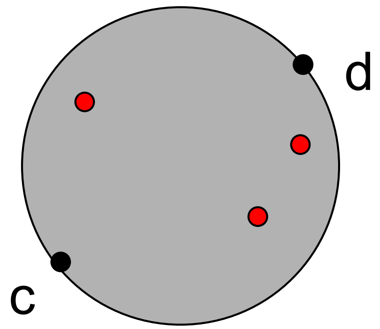
Gabriel Graph

Proximity graphs GG

p,q are adjacents if the circle of diameter pq does not contain in its interior other points of S



a, b are Gabriel neighbors

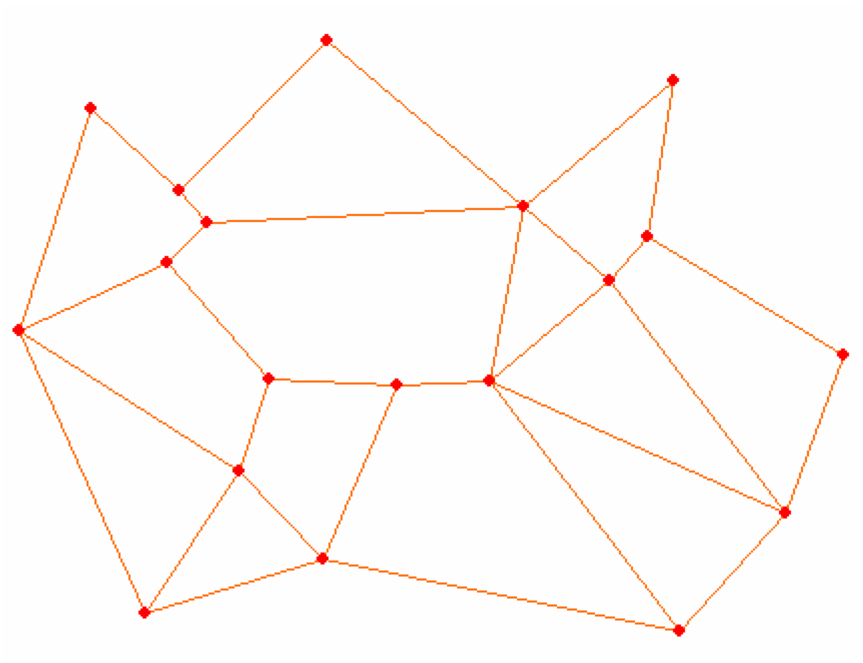


c, d are not Gabriel neighbors

Gabriel Graph

Gabriel Graph GG

p,q are adjacents if the disk of diameter pq does not contain in its interior other points of S

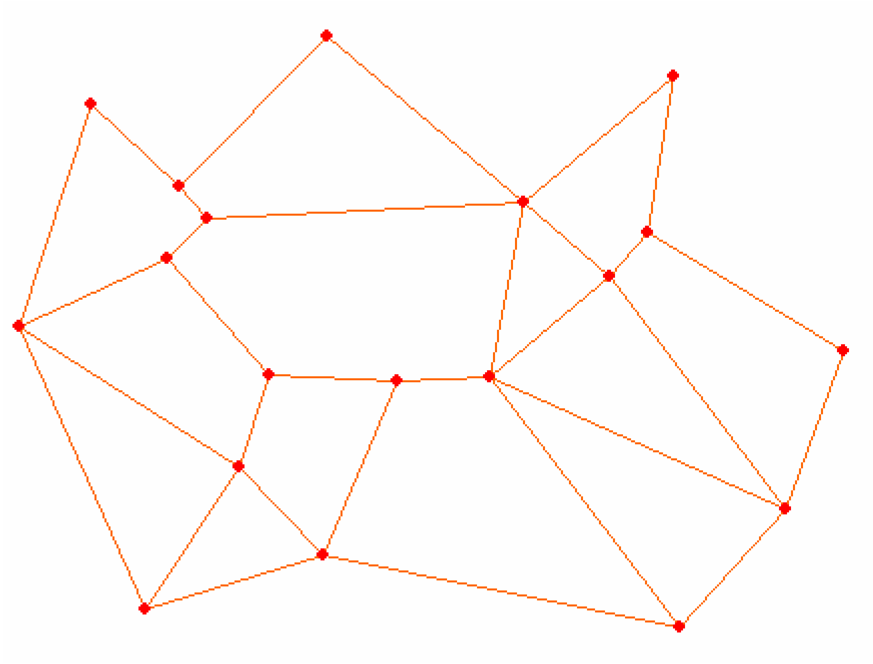


Gabriel Graph

Gabriel Graph GG

p,q are adjacents if the disk of diameter pq does not contain in its interior other points of S

$DT \supset GG \supset EMST$



Unit Disk Graph UDG

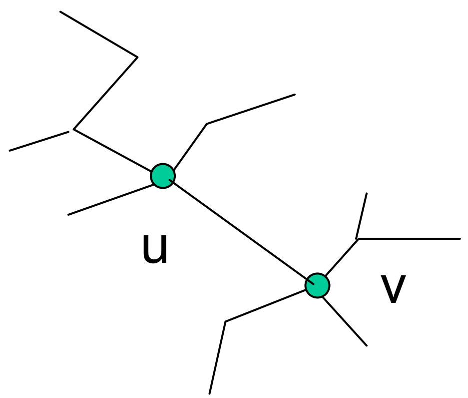
Problem 2
Given UDG(S), construct a planar subgraph locally

Lemma 1

Given S, if UDG(S) is connected, then UDG(S)$\cap$GG is connected

Proof: Let T=MST(S), we know that GG contains T. Therefore it is sufficient to prove that if UDG is connected then it contains T=MST(S)

We suppose there exists an edge uv in T with lenght >1



T-(uv) has two components , T_u, T_v

Unit Disk Graph UDG

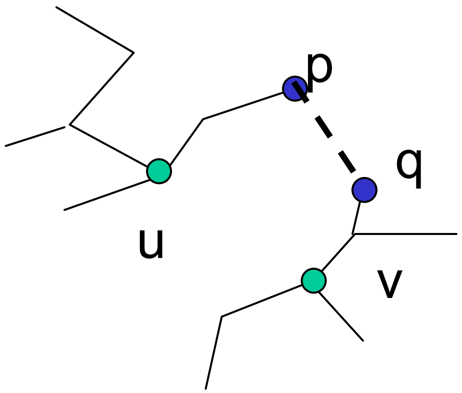
Problem 2
Given UDG(S), construct a planar subgraph locally

Lemma 1

Given S, if UDG(S) is connected, then UDG(S)$\cap$GG is connected

Proof: Let T=MST(S), we know that GG contains T. Therefore it is sufficient to prove that if UDG is connected then it contains T=MST(S)

We suppose there exists an edge uv in T with lenght >1



T-(uv) has two components , T_u, T_v

UDG connected, then exists pq, |pq|<1

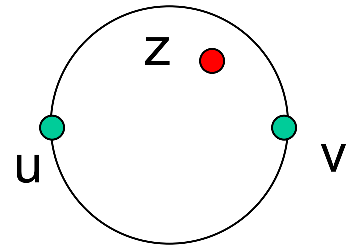
The tree T-(uv)+(pq) weighs less than T!!

Unit Disk Graph UDG

Problem 2
Given UDG(S), construct a planar subgraph locally

If (uv) is an edge of UDG(S) and (uv)$\notin$GG(S)
then exists a point z inside the disk of diameter uv

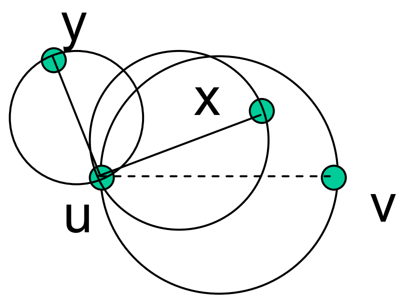
witness



Lemma 2

If (uv) $\notin$GG(S) and z is a witness, then the edges uz and vz belong to UDG(S)

Each vertex u deletes in UDG(S) the edges which do not belong to GG(N(u)$\cup${u})



Unit Disk Graph UDG

Problem 2
Given UDG(S), construct a planar subgraph locally

Algorithm GABRIEL-UDG

For each neighbor $x \in N(u)$
if $disc(u,x) \cap (N(u) - \{u,v\}) \neq \emptyset$ then delete the edge ux

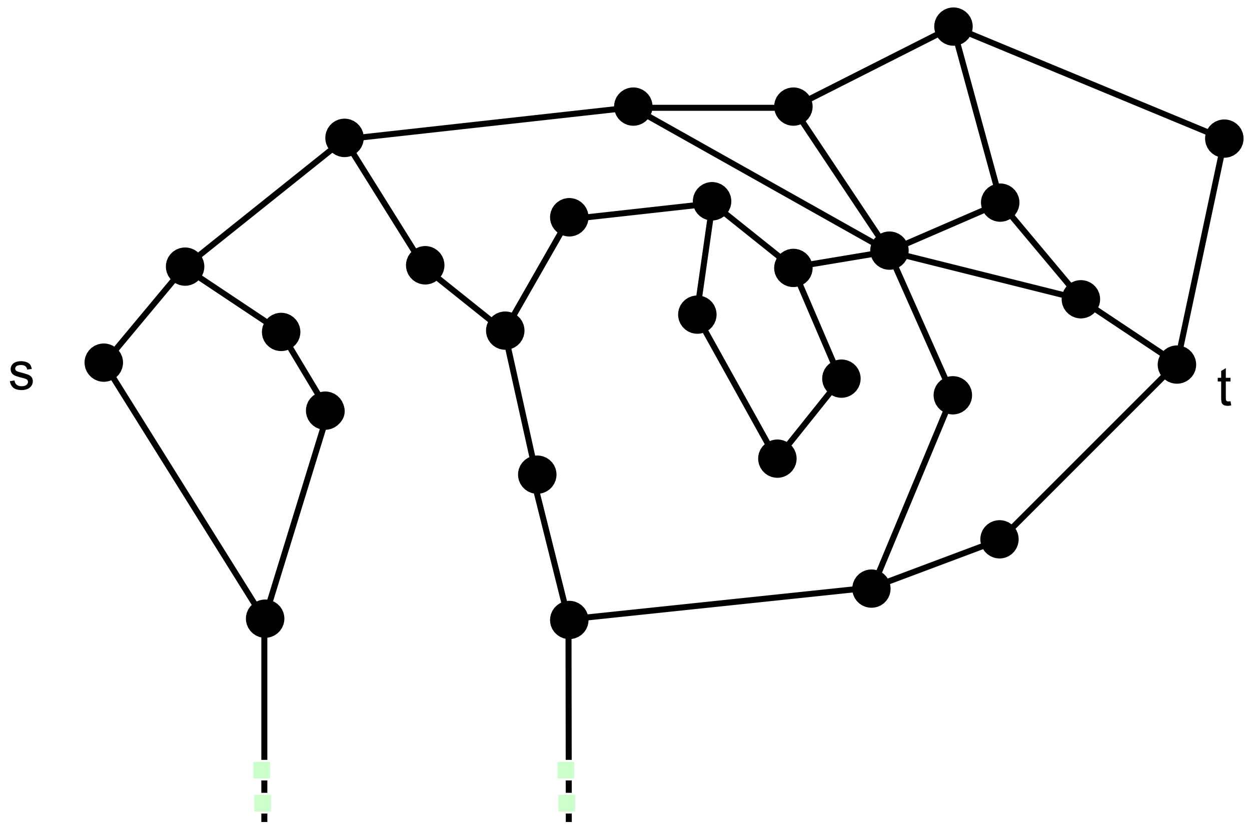
The cost in each vertex u is $O(d^2)$ where d=degree(u)

Wireless networks **ROUTING**

FACE ROUTING and GABRIEL GRAPH solve the problem, but ….

… the path, is a good path?

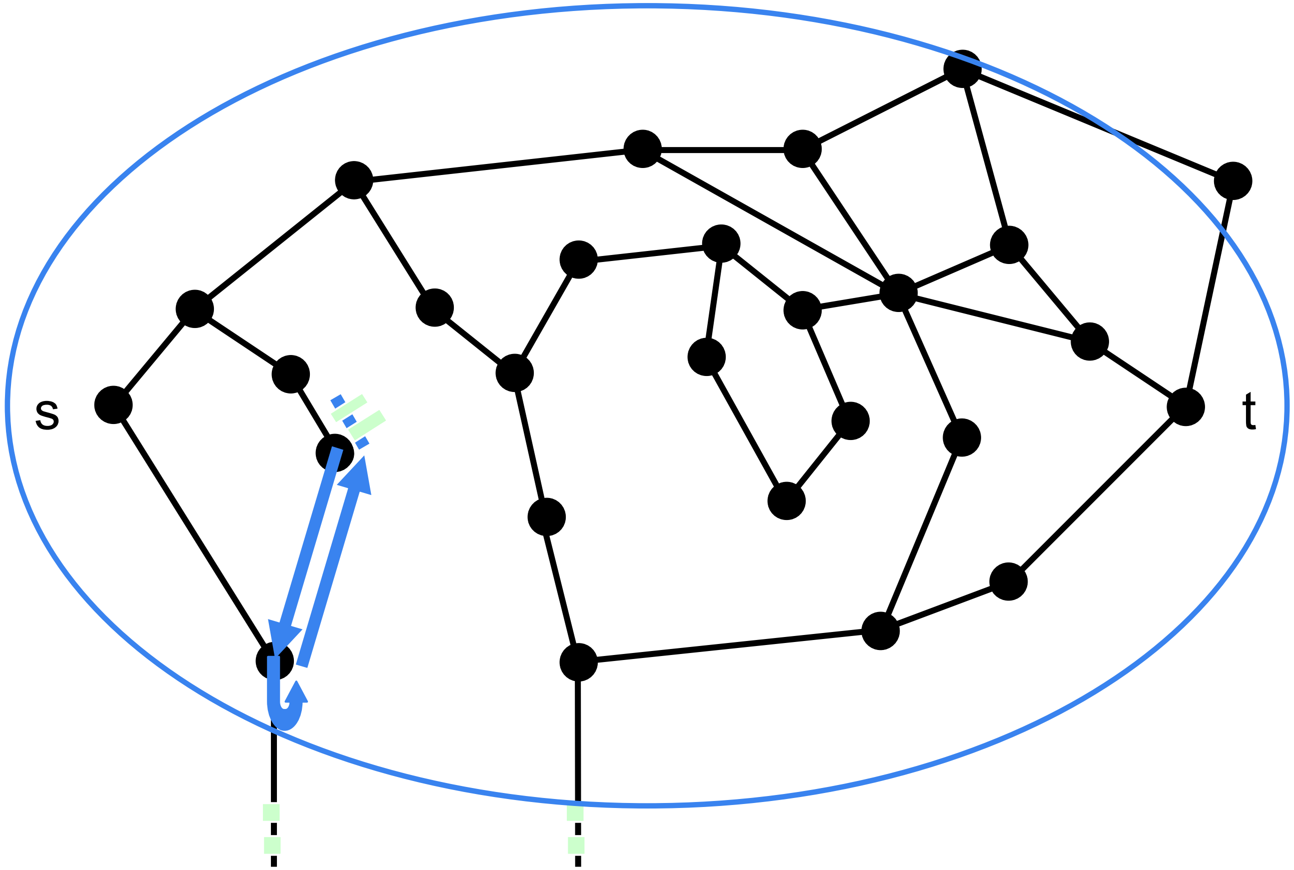
Can be very bad compared to the optimal route



How to improve face routing?

Wireless networks **ROUTING**

ADAPTIVE FACE ROUTING **AFR**



Wireless networks **ROUTING**

ADAPTIVE FACE ROUTING **AFR** **Analysis**

Theorem

If the optimal s-t route in the UDG has cost k, then AFR terminates with a route whose cost is $O(k^2)$.

“cost” c = c hops

LOWER BOUND

Let k be the length of the optimal path between s and t. There are networks for a which the route found by any local algorithm has cost $\Omega(k^2)$

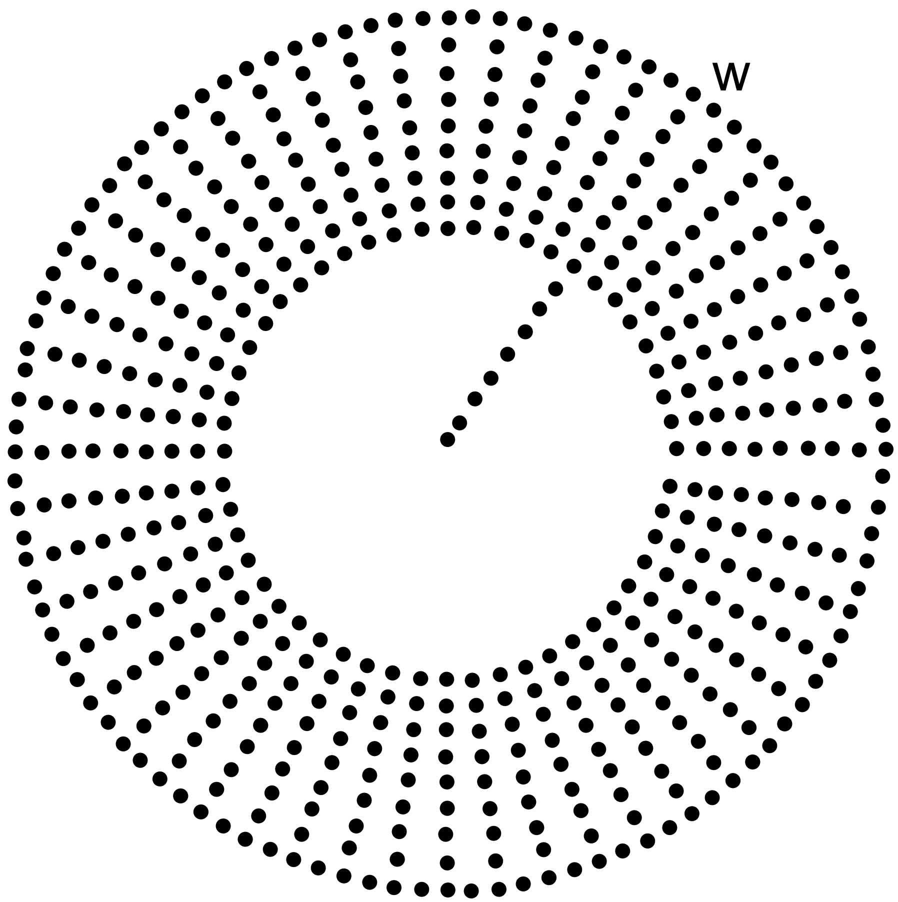
Kuhn, Wattenhofer, ‘02

Wireless networks **ROUTING**

LOWER BOUND

On a circle we evenly distribute 2k nodes such that the distance between two neighboring points is exactly 1.

For every second node of the circle we construct a chain of $k/2\pi$ - 1 nodes arranged on a line pointing towards the center.
The distance between two neighbors of a chain is exactly 1
Node w is one taken arbitrarily on the circle. The chain of w consists of k/π nodes with distance 1



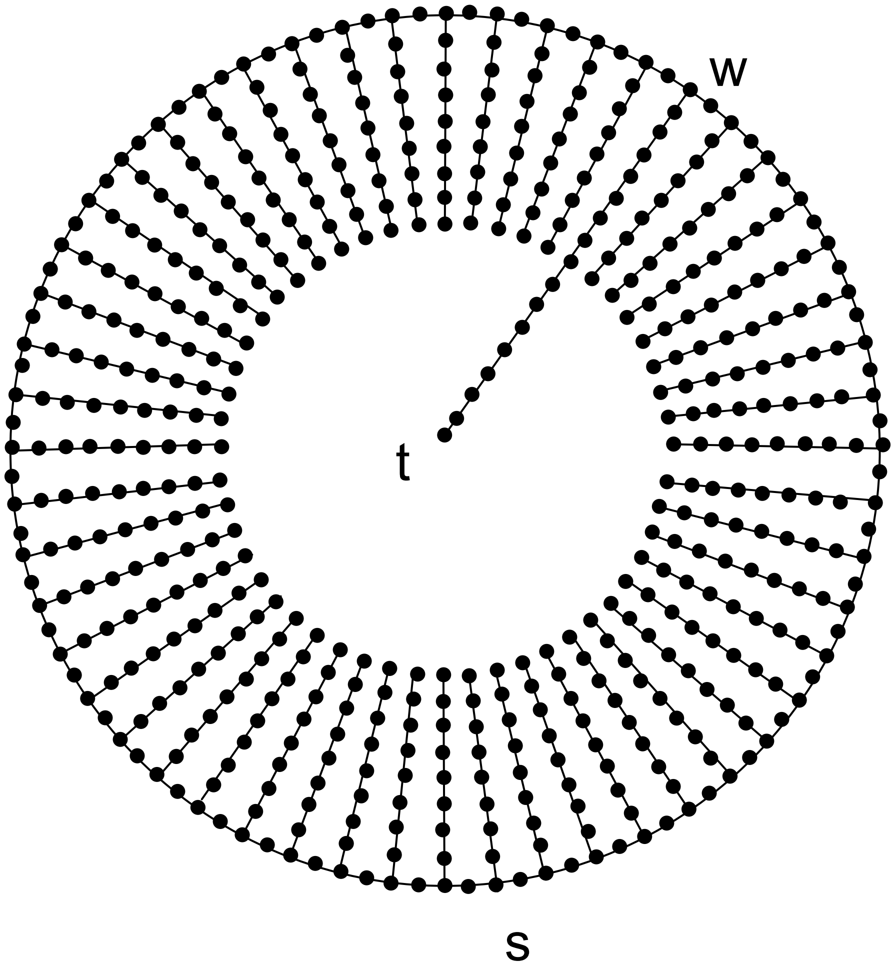
Wireless networks **ROUTING**

LOWER BOUND

The unit disk graph has k chains with Θ(k) nodes each.

We route from an arbitrary node on the circle (the source s) to the center of the circle (the destination t).

An optimal route between s and t follows the shortest path on the circle to w, and then directly follows w's chain to t with total length $c \in O(k)$



Wireless networks ROUTING

LOWER BOUND

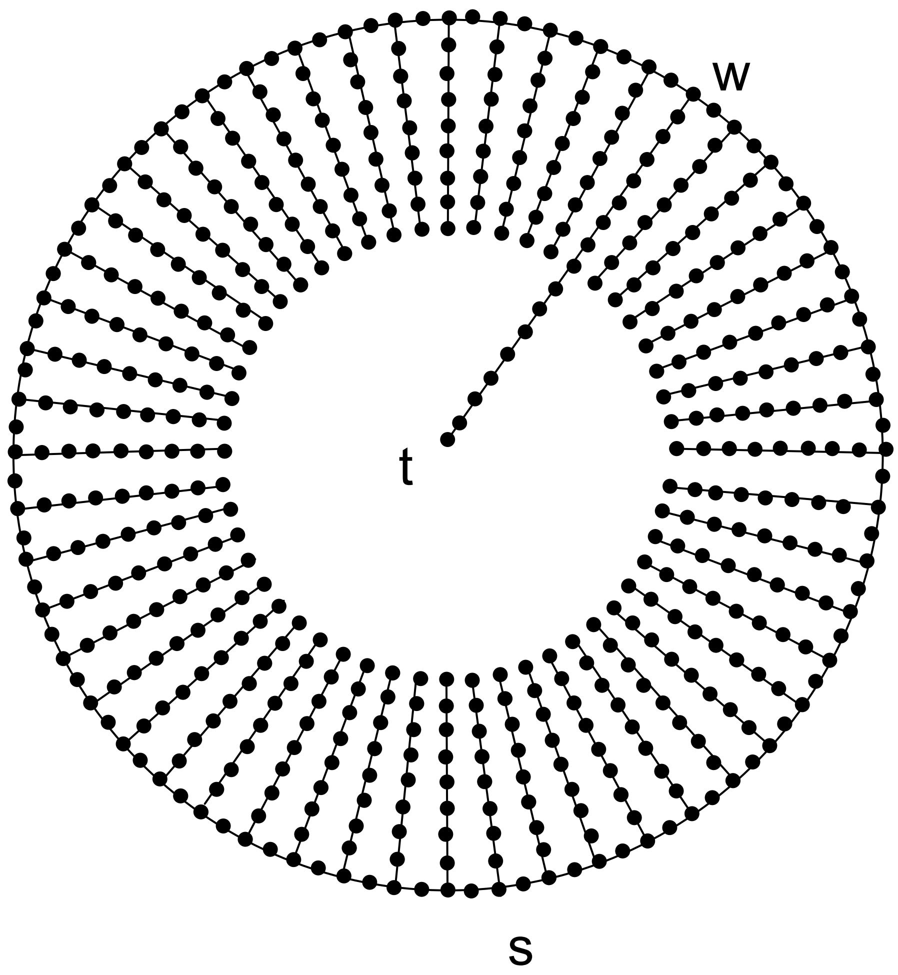
The unit disk graph has k chains with Θ(k) nodes each.

A geometric routing algorithm has to find the “correct” chain w.

Since there is no routing information stored at the nodes, this can only be achieved by exploring all the chains.

Any deterministic algorithm needs to explore all the chains until it finds the chain w.

The algorithm will therefore explore Θ(k^2) (instead of only O(k)) nodes



Theorem

Adaptive Face Routing is asymptotically optimal

Wireless networks ROUTING

ROUTING PROBLEM

Adaptive FACE ROUTING

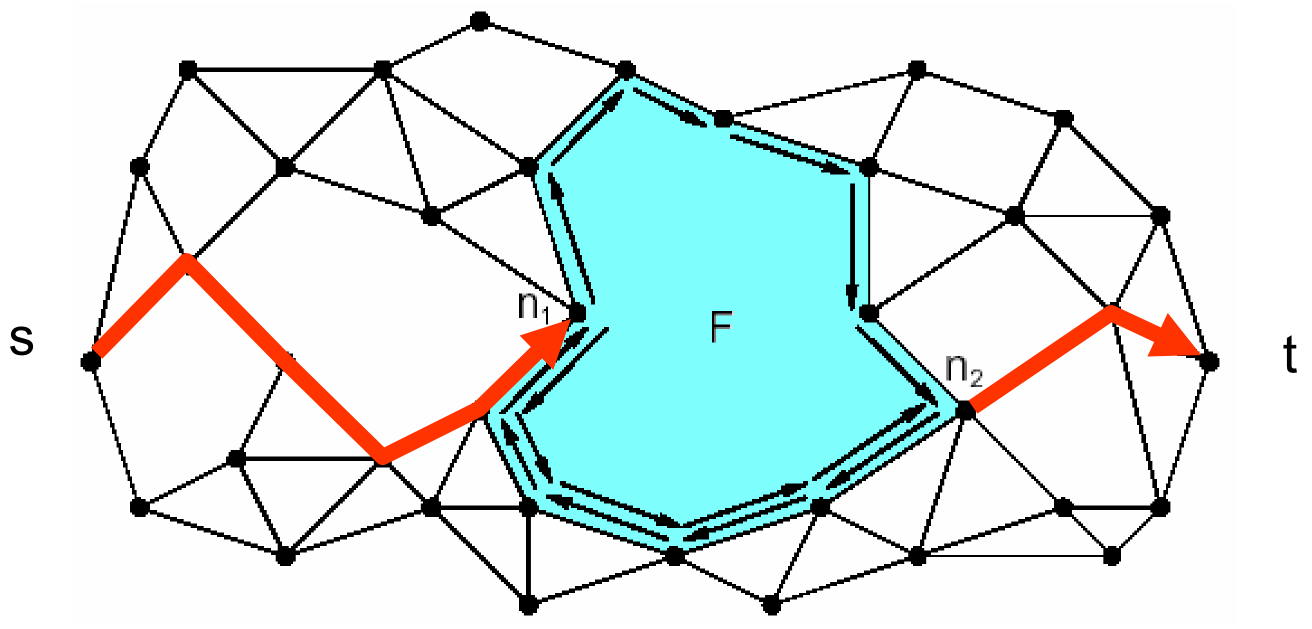
GREEDY ROUTING

GREEDY – Other Adaptive FACE ROUTING
GOAFR

Wireless networks **ROUTING**

ROUTING PROBLEM

GREEDY – Other Adaptive FACE ROUTING
GOAFR



1. Route greedily as long as possible
2. Circumvent "dead ends" by use of face routing
3. Then route greedily again

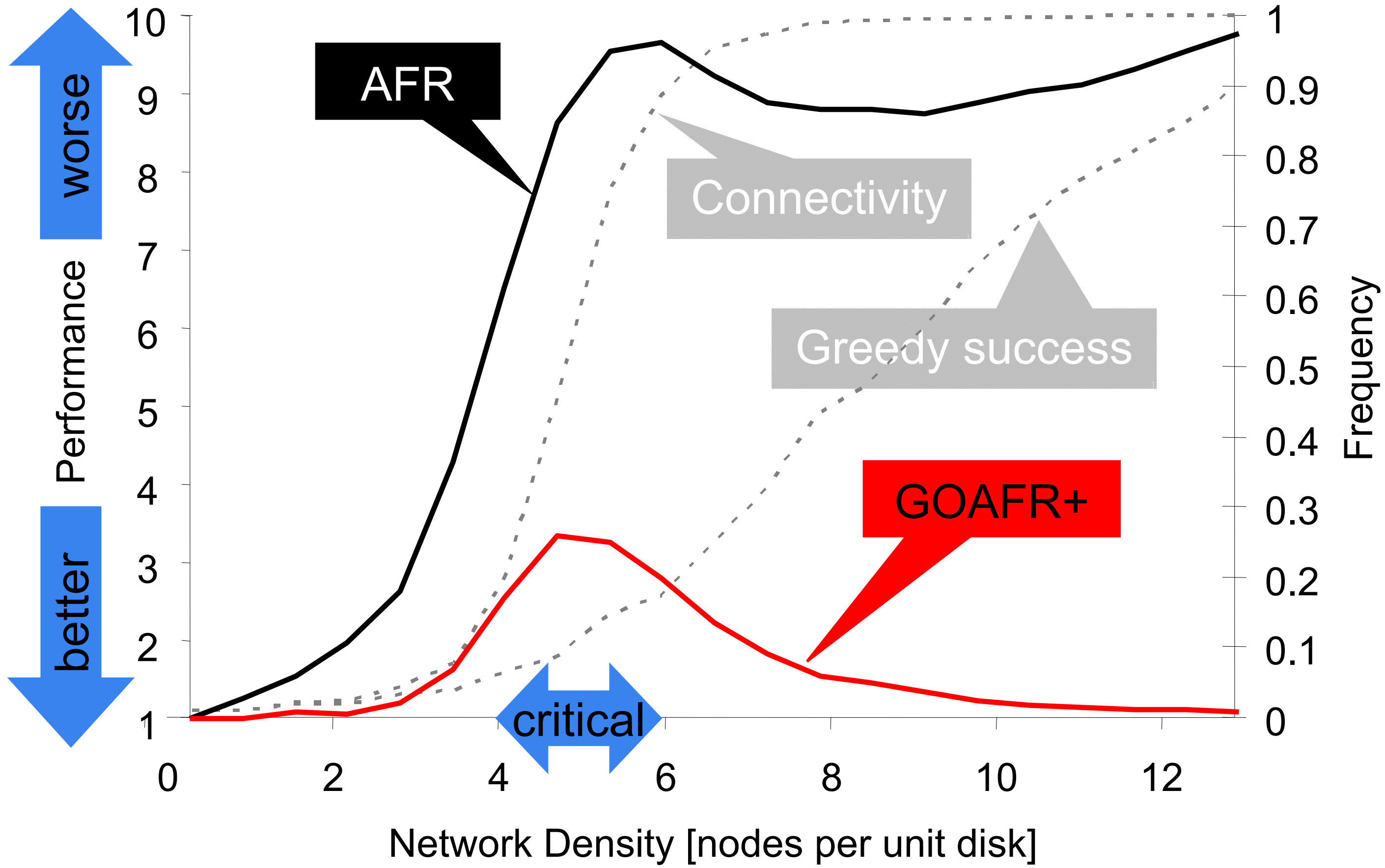
ROUTING PROBLEM

GREEDY – Other Adaptive FACE ROUTING GOAFR

Theorem

GOAFR is still asymptotically worst-case optimal…
…*and* it is efficient in practice, in the average-case

Wireless networks **ROUTING**



Kuhn, Wattenhofer

Routing 3D

Wireless networks are in three-dimensional space

Can we translate 2D solutions to the 3D space?

- Greedy Routing?
- Face Routing?

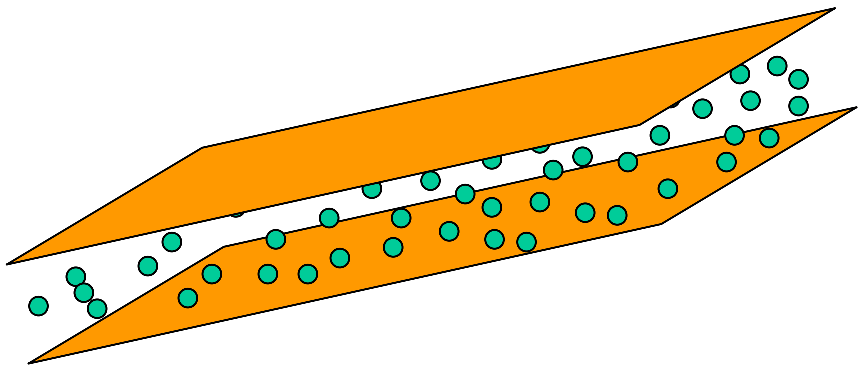
There is not local memoryless routing algorithm, that deliver messages deterministically in 3D

Durocher et al. 2008

Routing 3D

There is not local memoryless routing algorithm, that deliver messages deterministically in 3D

If the nodes are contained within a slab of thickness $1/\sqrt{2}$, there exists a 2-local routing algorithm that succeeds for UnitBallGraph(S).

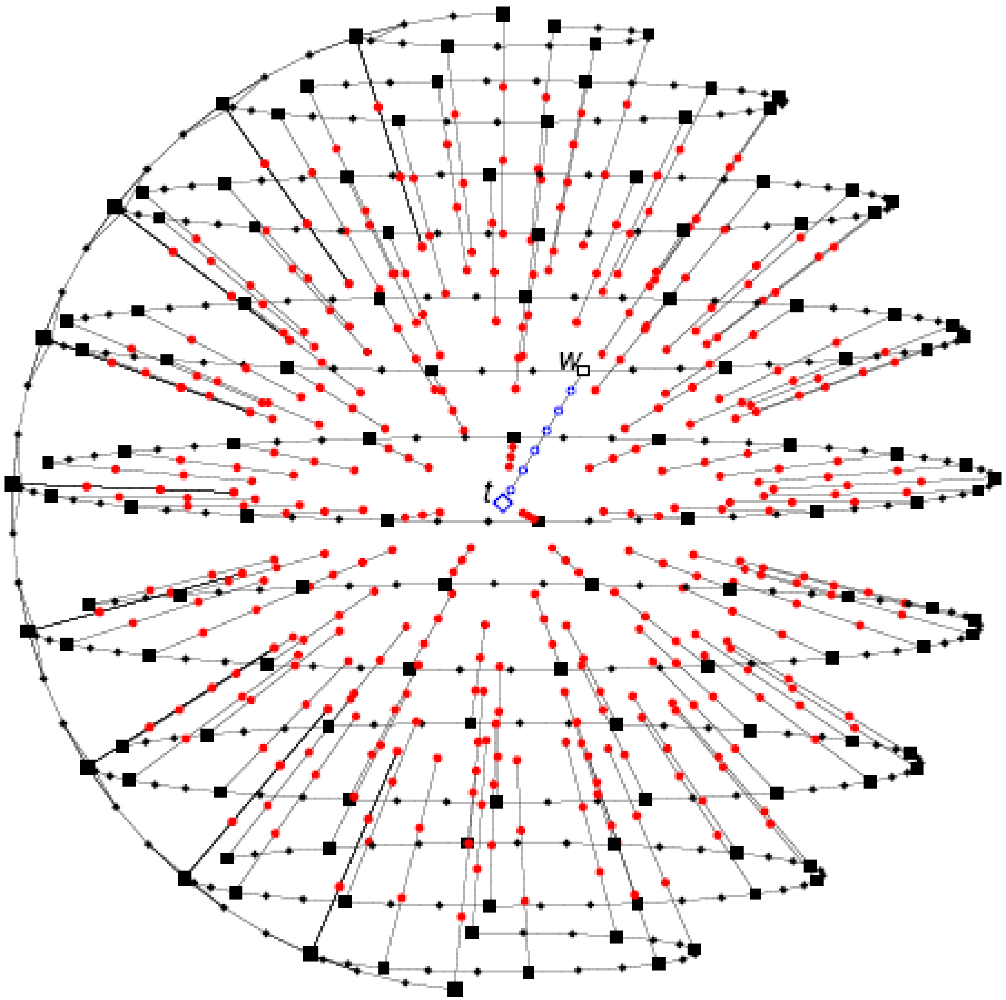


Routing 3D

and randomized?

There are networks for which the route found by any randomized geometric routing algorithm has expected length $\Omega(d^3)$

Flury, Wattenhofer, '08



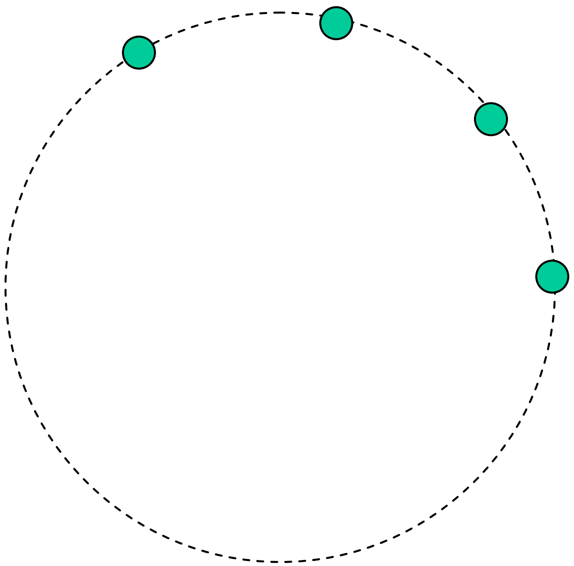
Local Algorithms

Local Algorithms MST

MST(UDG) Minimum Spanning Tree

It is NOT possible to calculate MST(S) with a local algorithm

S



n points on a circle C such that the distances between two consecutive nodes on C are $1-\varepsilon_i$ $i=1, .. , n$

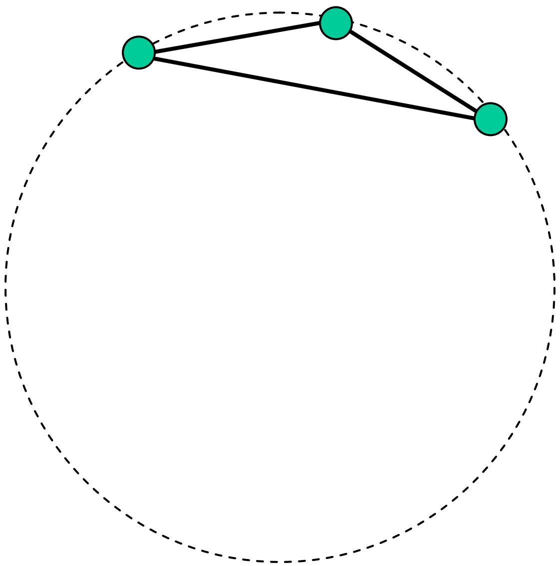
Finding MST(S) is equivalent to identifying the smallest ε_i

Therefore is not possible to obtain MST(S) locally

Local Algorithms DT

It is NOT possible to calculate DT(S) with a local algorithm

S



The circle may be arbitrarily large

Local Algorithms

UnitDiskGraph

Czyzowicz et al. '07, '08

• Dominating set	5-approx
• Connected Dominating Set	7.4-approx
• Vertex Color	7-approx

Wiese, Kranakis, '08

PTAS for Independent Set, Vertex Cover, Dominating Set

Local Algorithms

Graphs

Kuhn, Moscibroda, Wattenhofer, ‘04

Many graph problems cannot be solved locally on general graphs.
Min Vertex Cover, Min Dominating Set, Max Independent Set, …

Lenzen, Wattenhofer

’08 Dominating Set (planar graph), 74-aprox.
‘10 (bounded arboricity a) $O(a^2)$-aprox.

Schneider, Wattenhofer
’10 MaxIndependentSet (bounded independence)

Polishchuk, Suomela, ‘09
Vertex Cover (bounded degree), 3-aprox.

REFERENCES

- E. Kranakis, H. Singh, J. Urrutia, *Compass routing on geometric networks*, CCCG'99
- P. Morin, *Online routing in geometric graphs*, 2002
- F. Kuhn, R. Wattenhofer, A. Zollinger, DialM 2002, MobiHoc 2003
- N. Linial, *Locality in distributed graph algorithms*. Siam J. Comp. 1992
- J. Urrutia, *Local solutions for global problems in wireless networks,* J. Discrete Algorithms, 2007
- "Distributed Computing Group" ETH Zürich, http://dcg.ethz.ch/index.html