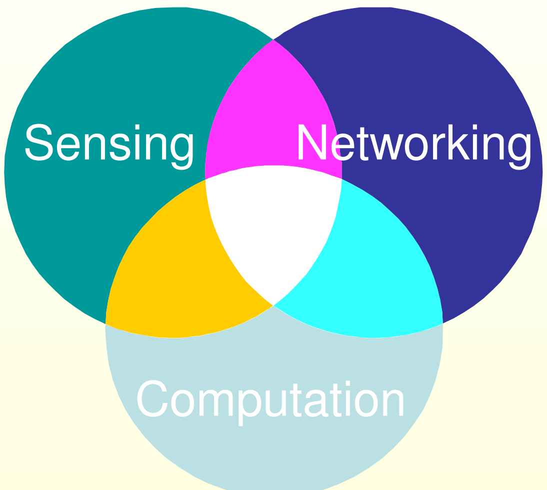
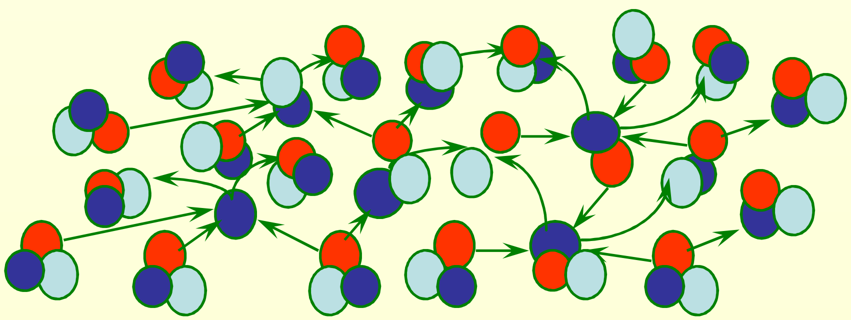
CS321: Networking Sensors III: Landmark Methods



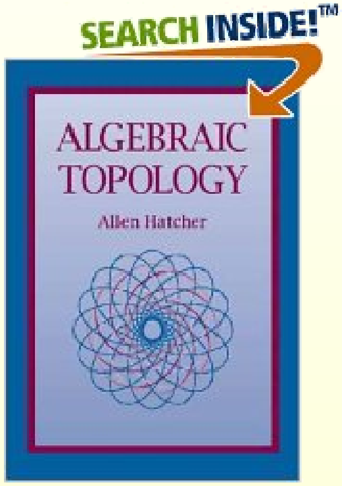
Leonidas Guibas
Computer Science Dept.
Stanford University



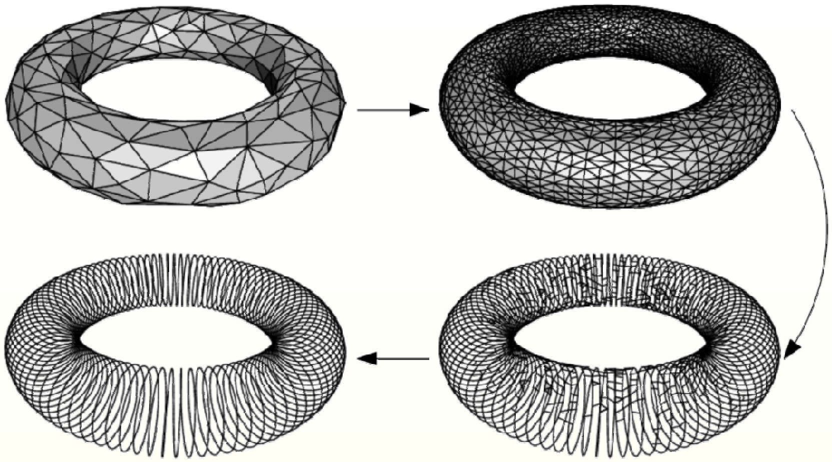
Landmark methods for

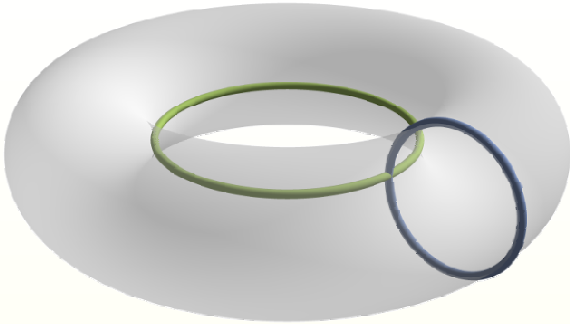
- network structure discovery
- navigation

Another Little Excursion, Into Algebraic Topology This Time



Algebraic Topology, Allen Hatcher, Cambridge University Press, 2002

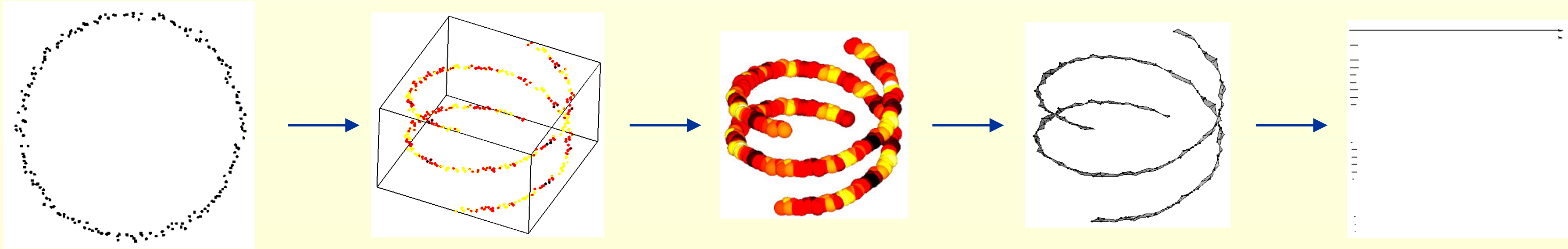




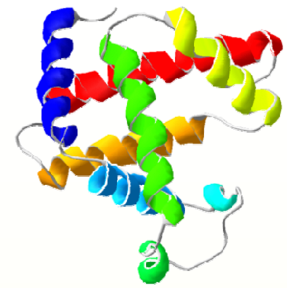
Network Structure Discovery, Naming and Routing Structures

Algebraic topology, surface reconstruction, machine learning

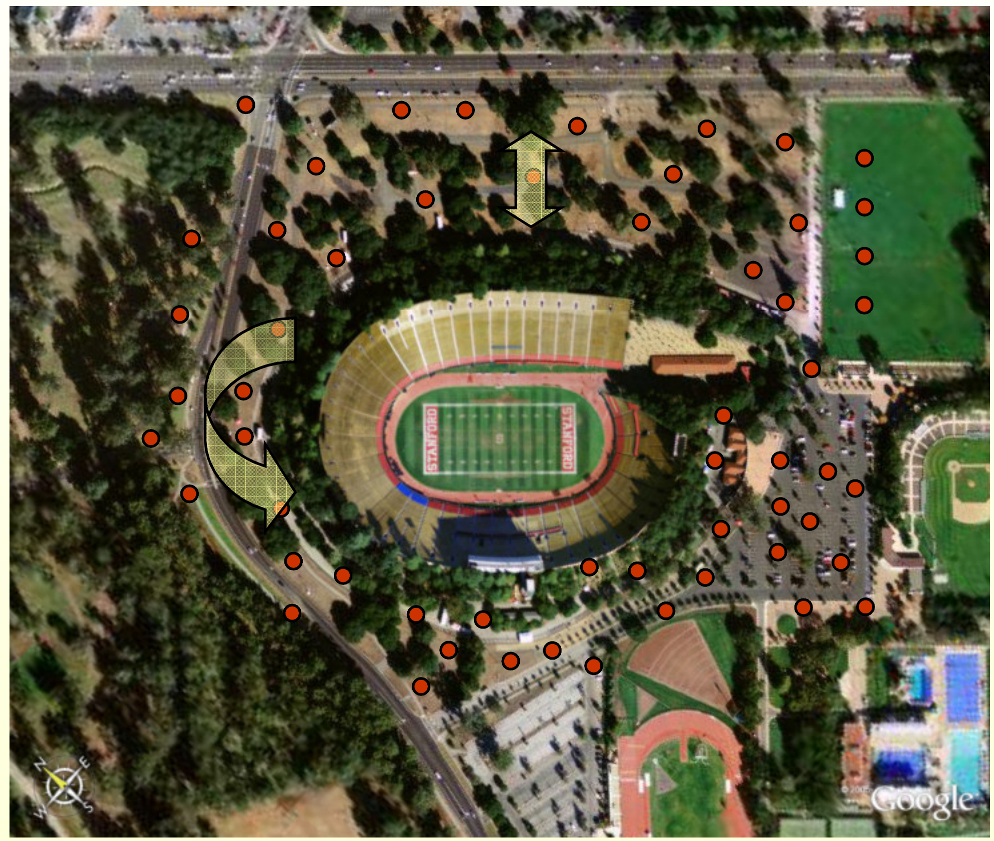
Stability under communication link variability



Structure Discovery

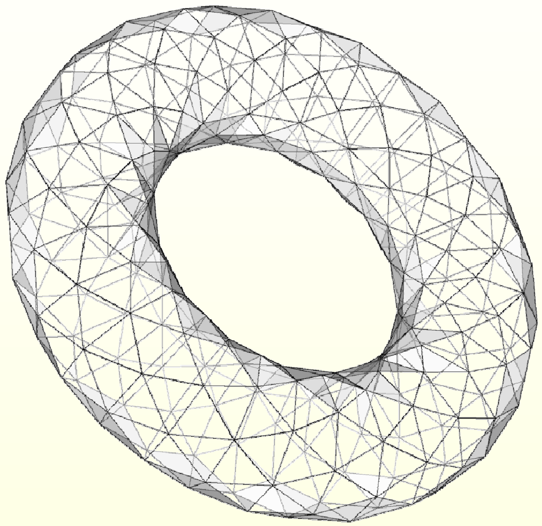


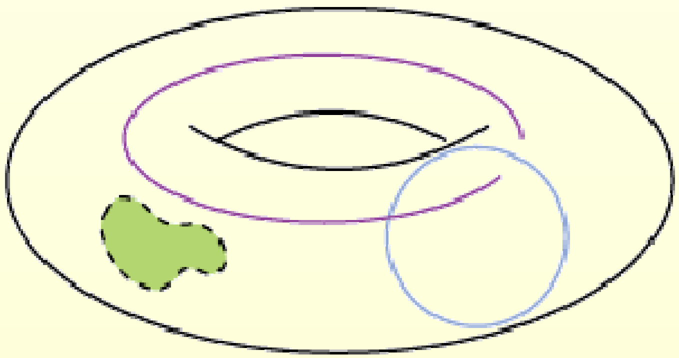
- Consider a static *ad hoc* network – e.g., a sensor network
- Although there is no mobility of nodes, there can be volatility in both links and nodes
- Are there useful global structures preserved under this volatility?
- Can these be discovered by lightweight methods?
  - layout morphology
  - signal landscape morphology
- And be used to establish
  - information highways
  - sensor node collaboration groups



Topological Descriptors for Morphological Analysis

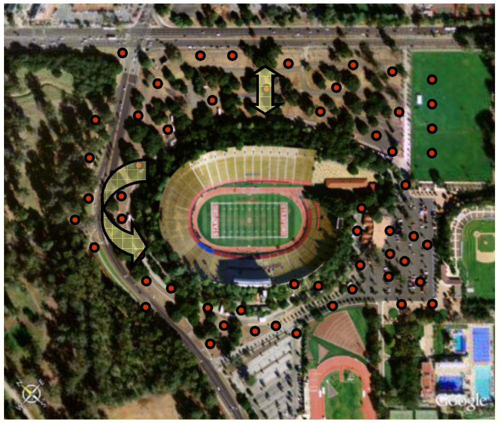
- Geometry
  - What does the shape look like?
  - Intuitive & familiar
  - Quantitative
  - Local
  - Low-level
- Topology
  - How is a space connected?
  - Qualitative
  - Abstract & alien
  - Global & coarse
  - High-level





Understand obstructions to connectivity

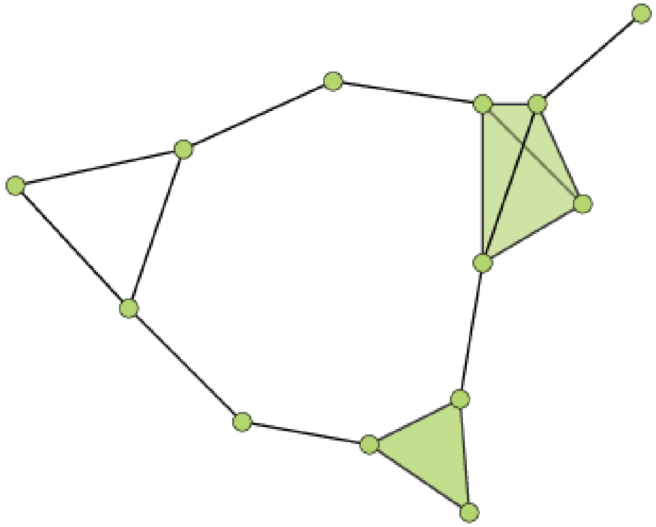
Simplicial Complexes



- Isolated points or samples have no useful topology
  1. Need to approximate the hidden space the points have been sampled from
  2. Compute a discrete representation from which topology can be extracted

- General Technique (Čech complex)
  1. Put ε-balls (shape functions) around points
  2. Capture intersection patterns of these balls

- *k*-simplex: vertex, edge, triangle, tetrahedron, …



Topology via Homology

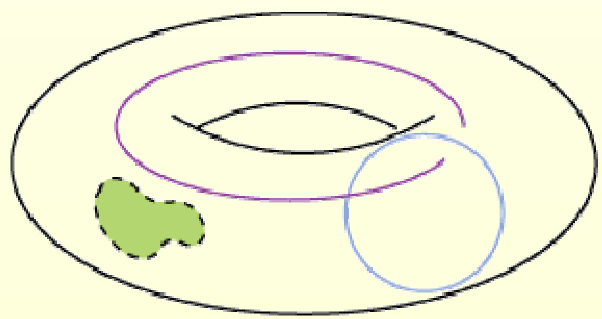
- Homology Groups – capture connectivity of space (really of the discrete complex)

- Extend notion of (non-trivial) cycle to all dimensions

- Vector space
- kth Betti number β_k = rank H_k

$$H_k = Z_k / B_k$$

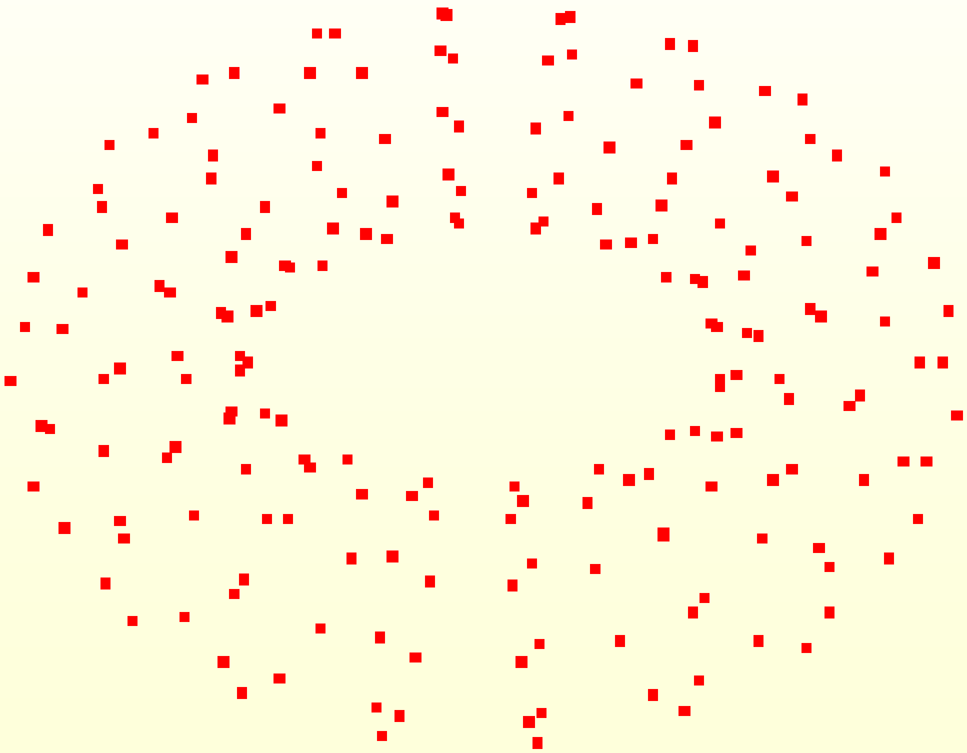
- Geometric Interpretation
  - β_0: # components
  - β_1: # tunnels or loops
  - β_2: # voids



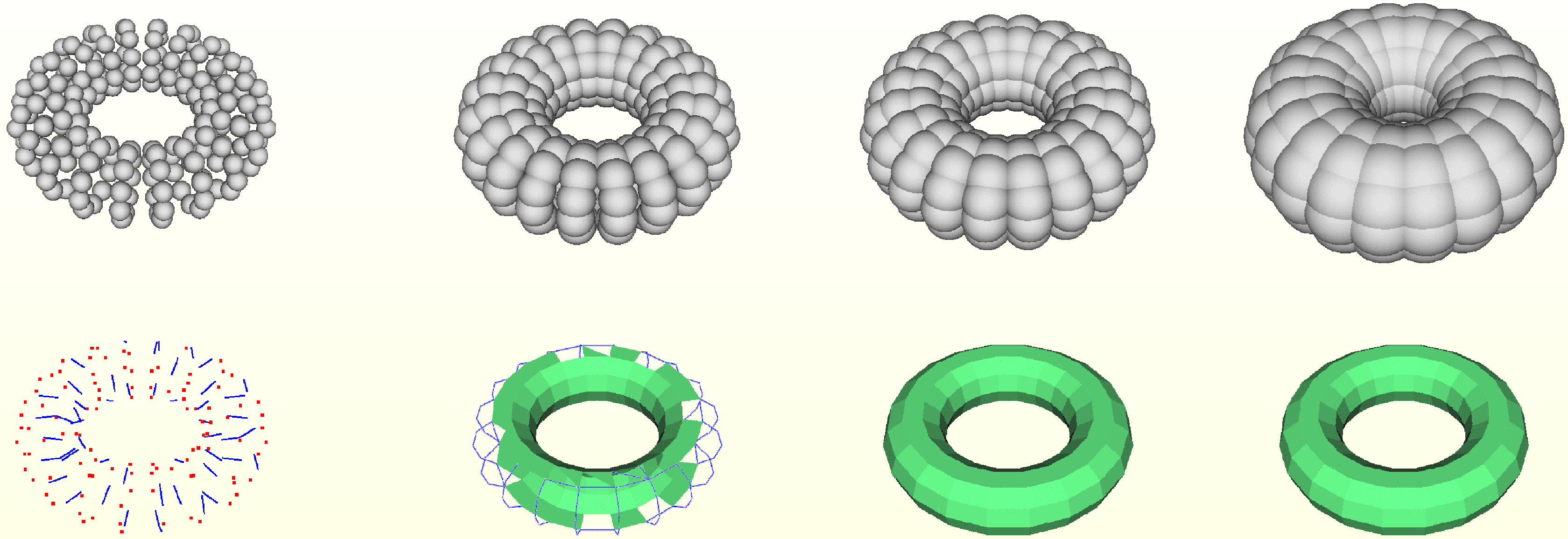
1, 2, 1

higher-order connectivity

Detecting a Torus from Samples



Question of Scale

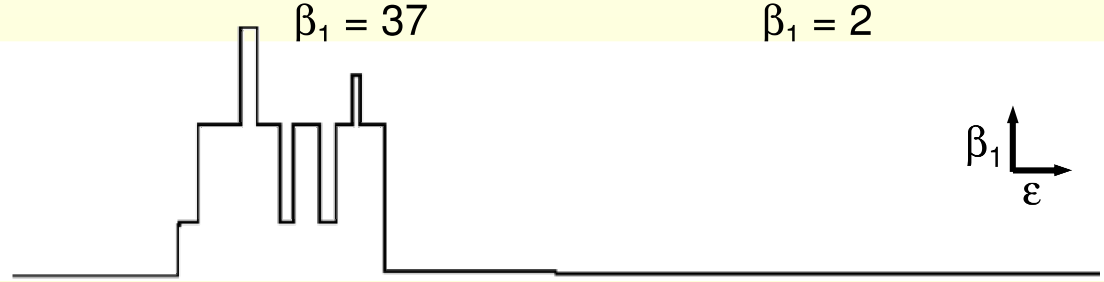


$\beta_0 = 150$, $\beta_1 = 0$

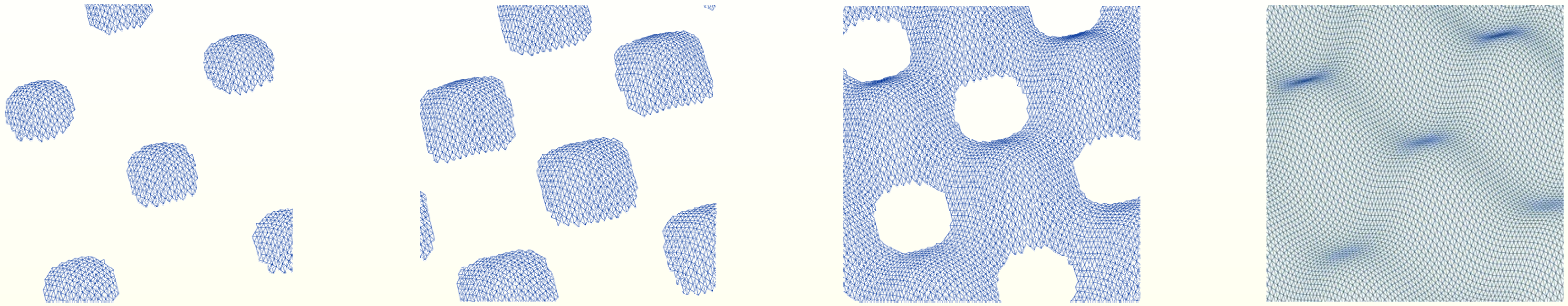
$\beta_0 = 1$, $\beta_1 = 37$

$\beta_0 = 1$, $\beta_1 = 2$

$\beta_0 = 1$, $\beta_1 = 1$



Filtrations Through Scale Space

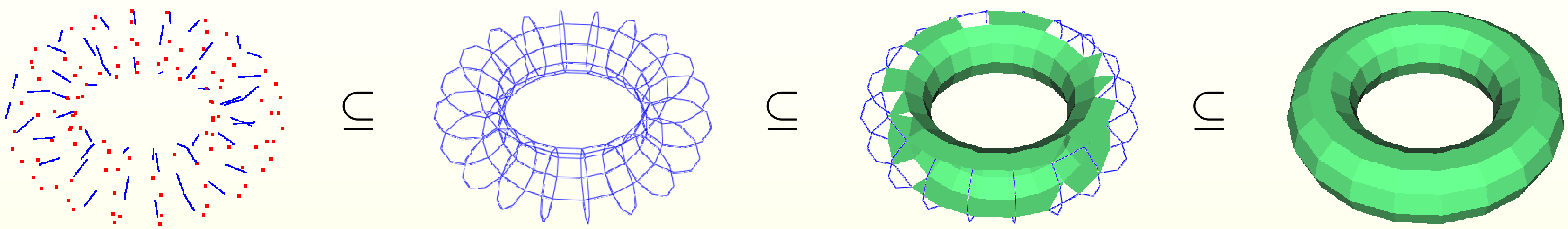


- A filtration of a complex *K* is a nested sequence of increasing complexes:

$$\emptyset = K^0 \subseteq K^1 \subseteq K^l \subseteq \cdots \subseteq K^m = K$$

- Geometric measures dictate ordering
  - Distance (Čech-like complexes, e.g. α-, Rips, Witness, etc.)
  - Local Density
  - Manifolds equipped with Morse functions

Inductive Systems



K^{250} $\subseteq$ K^{500} $\subseteq$ K^{994} $\subseteq$ K^{1452}

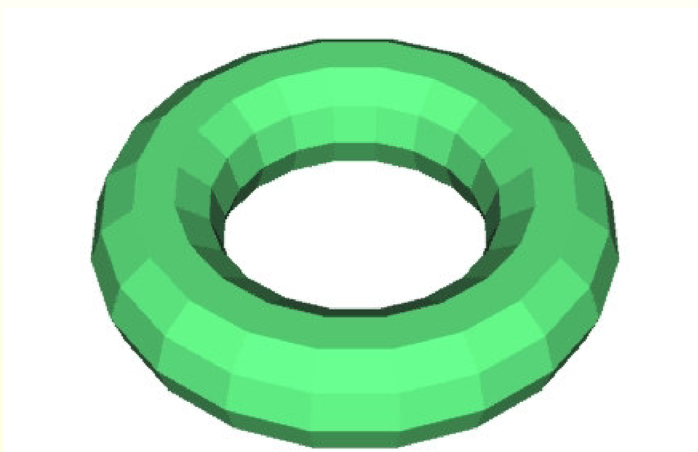
$$K^{250} \xrightarrow[i]{} K^{500} \longrightarrow K^{994} \longrightarrow K^{1452}$$

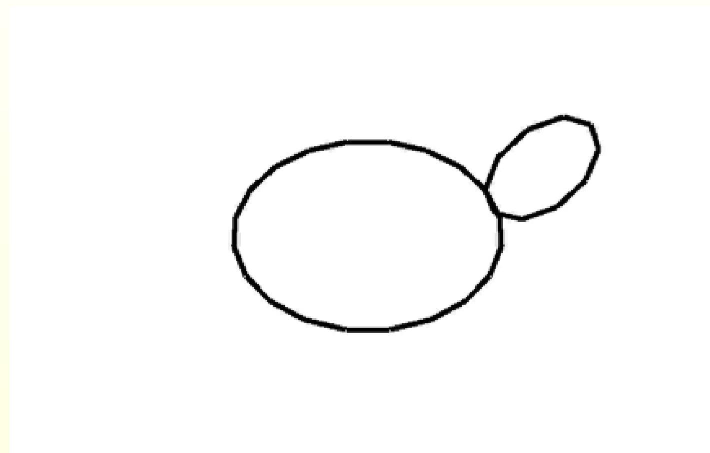
Functoriality ↓

$$H_k(K^{25}) \longrightarrow H_k(K^{50}) \longrightarrow H_k(K^{994}) \longrightarrow H_k(K^{1452})$$

Idea: Follow basis elements from birth to death
while maintaining compatible bases

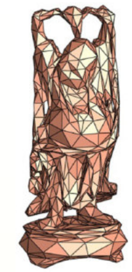
Consistent Bases Exist





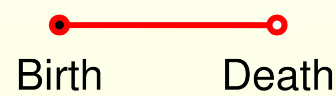
Persistent Homology

[Zomorodian. Edelsbrunner, Letcher '02]

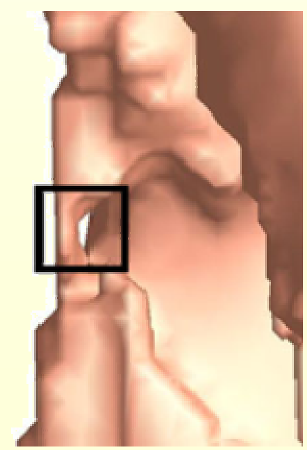


- Homology: $H_k(K^l) = Z_k(K^l) / B_k(K^l)$
- The p-persistent k-th Homology group

$$H_k^{l,\,p} = Z_k^{\,l} / (B_k^{\,l+p} \cap Z_k^{\,l})$$

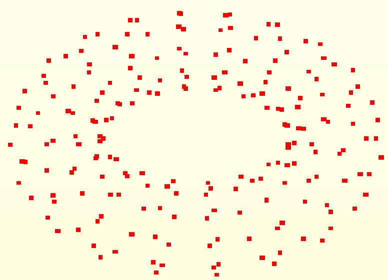


Persistent topological features are part of the shape; transient ones may be noise.

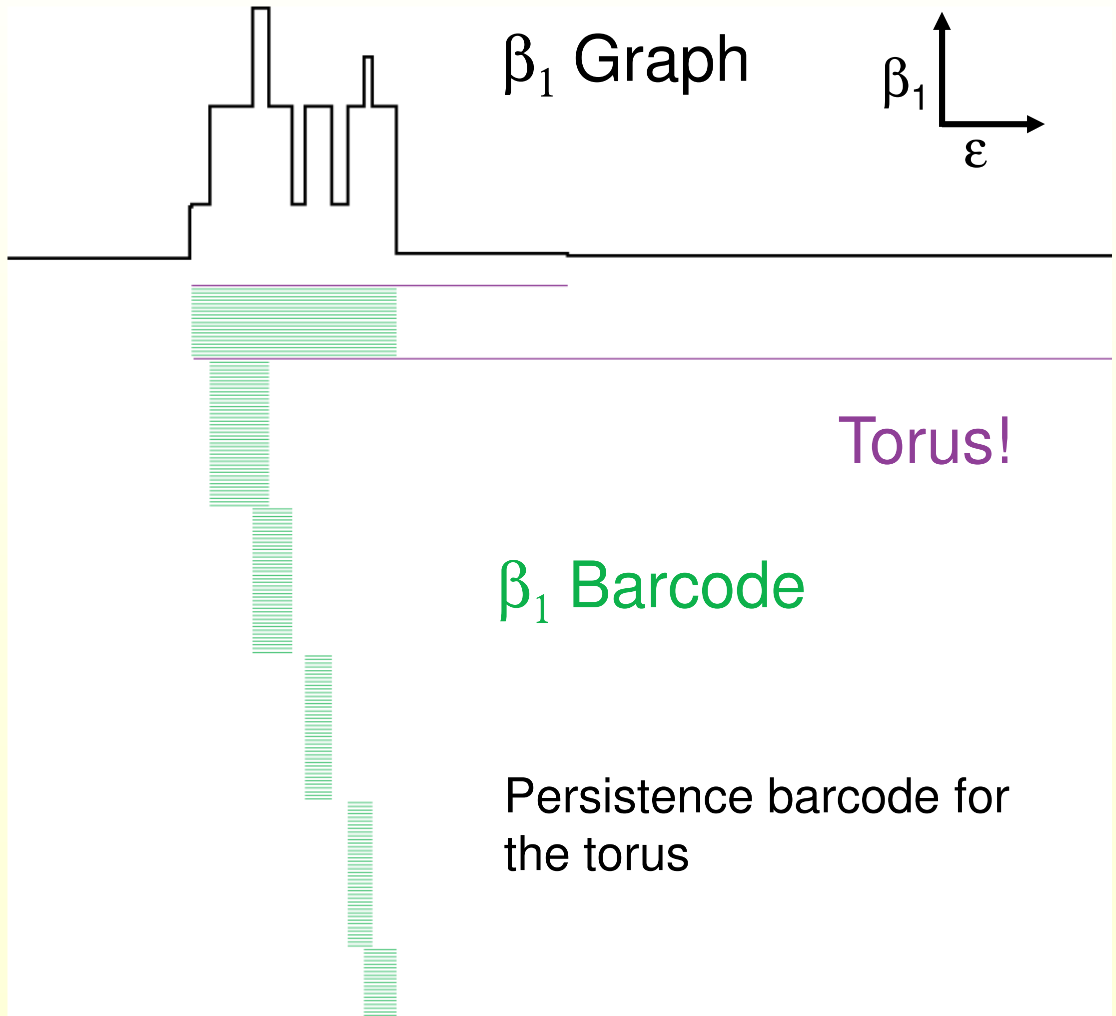


- Persistence Barcode: multiset of intervals

Deconstructing the Graph

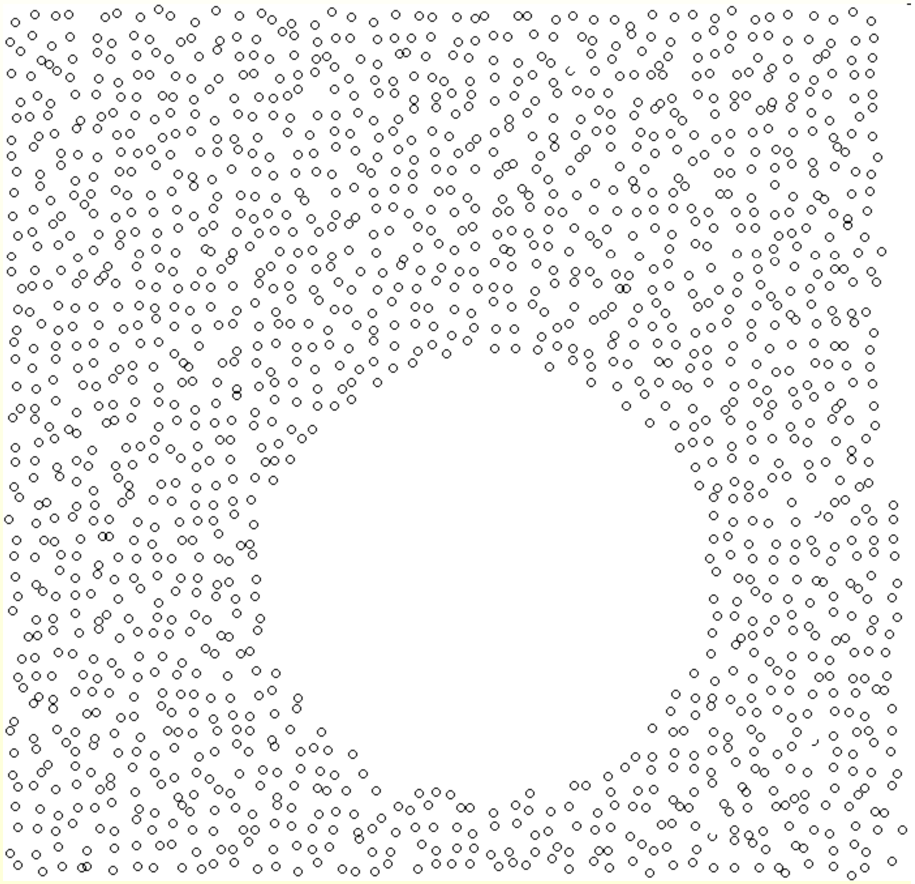


PCD



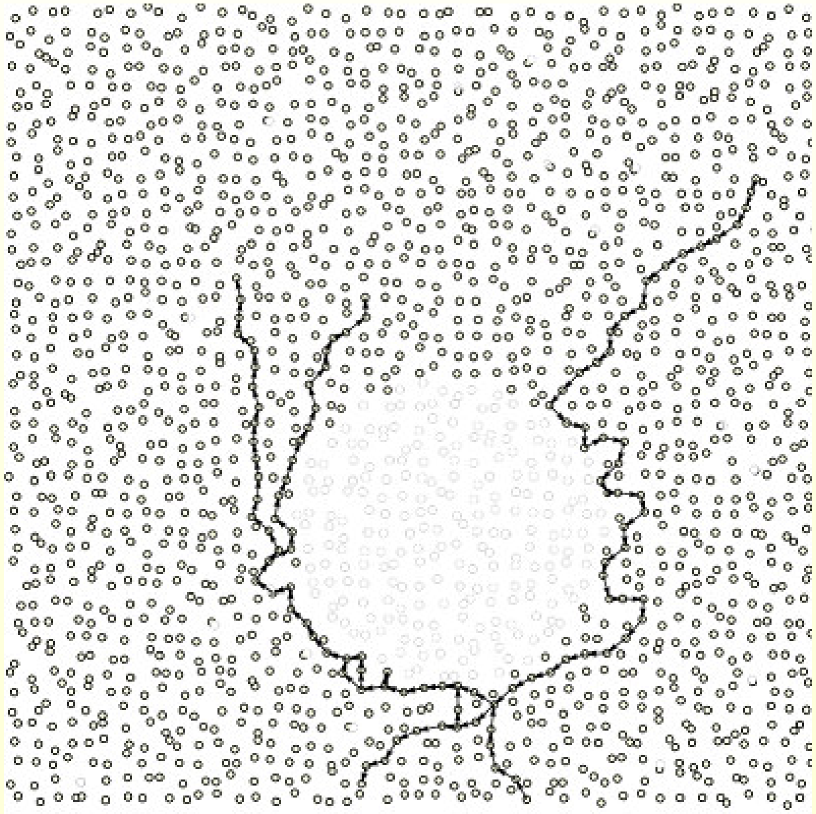
Persistence barcode for the torus

Understanding Node Layouts via Simplicial Complexes



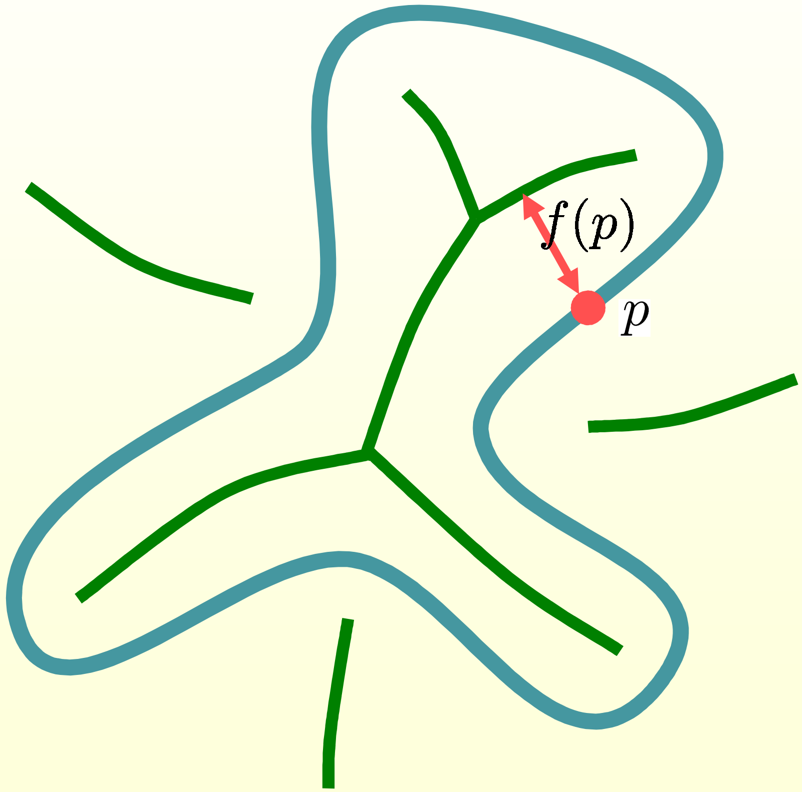
Impact of Higher-Order Topological Features

- Lakefront driving unevenly depletes nodes
- Causes holes to enlarge and then merge
- Boundary and hole detection has been extensively investigated [Fekete et. al ‘04, Fang et. al. ’05 ‘07, Funke et. al. ’05 ‘06, Ghrist et. al. ‘05. Kroller et. al. ‘06, Wang et. al. ‘06]



GPSR [Karp and Kung, Mobicom ‘00] routes

Local Feature Size: a Sampling Condition



LFS function *f* = distance to medial axis

For each point *p* on the shape, there needs to be a sample within α *f*(*p*) of *p*

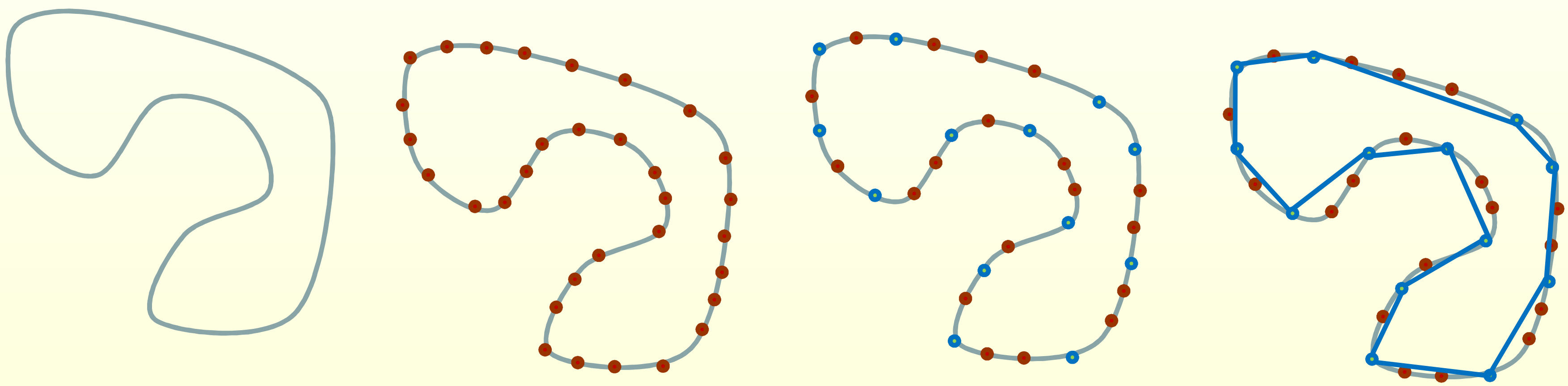
Then strong (topological, geometric) reconstruction guarantees can be given for Delaunay-based methods

[Amenta et. al, '98, …]

Reconstructed complex is a subcomplex of the Delaunay complex *D* of the samples

Sampling the Samples: Using Landmarks

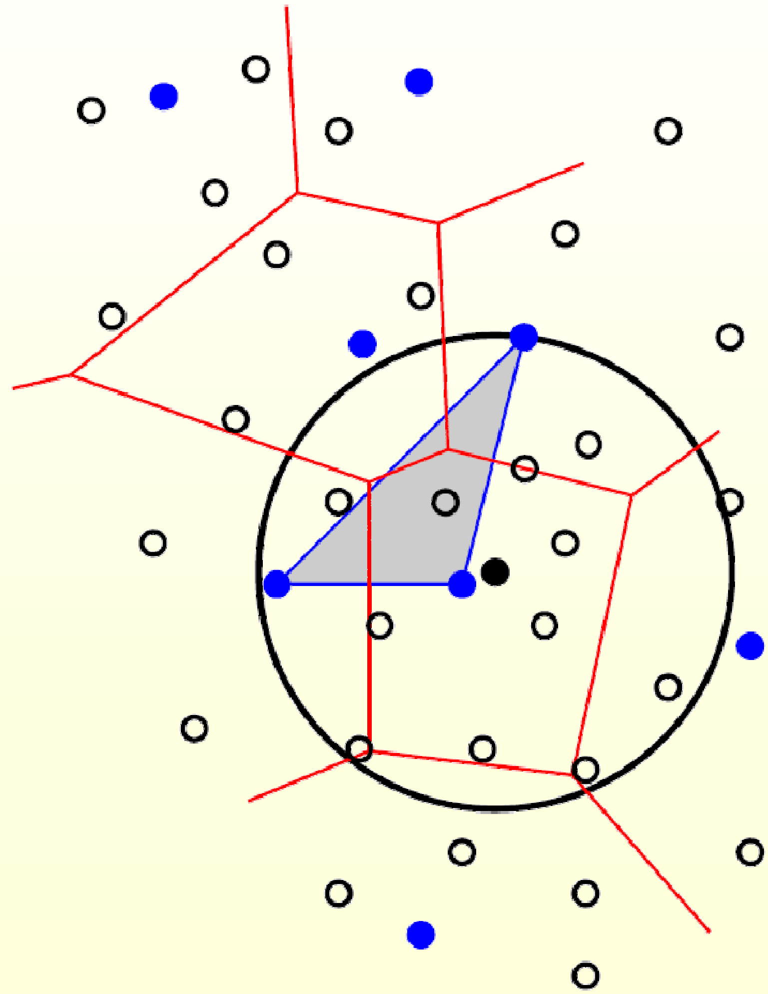
- How can we check a sampling condition if we don't know the underlying shape/space?
- Idea: select a set of landmark samples; let the other samples act as proxies for the missing shape/space.



- given samples (hopefully satisfying certain density conditions)
- choose landmarks
- form complex on landmarks, w. advice from other samples
- reconstruct w. guarantees

Witness Complex *W*

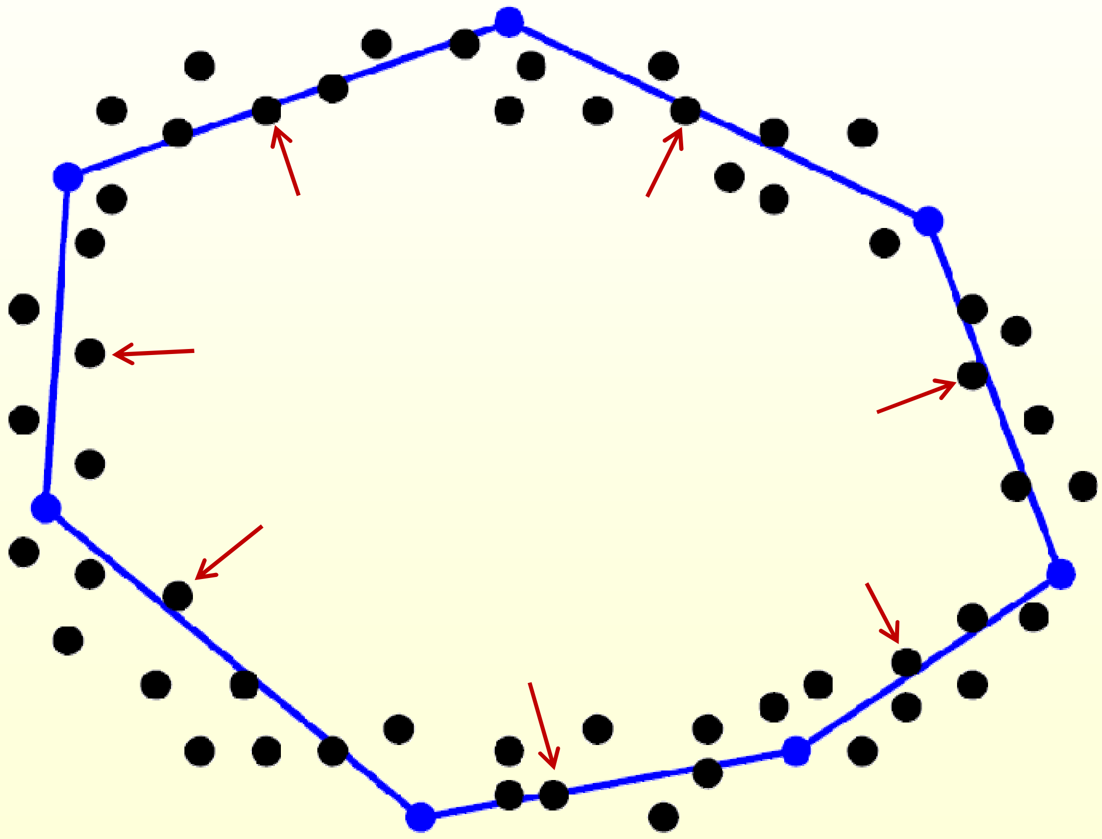
- Computing the Delaunay complex *D* is a challenge when no coordinates are available
- *W* is based only on distance comparisons
- Under some conditions, $W = D$
- More often than not, $W \subset D \subset W_{\text{relaxed}}$
- Ideal for geodesic or intrinsic metric settings



Witness complexes are a poor man's substitute for Delaunay complexes

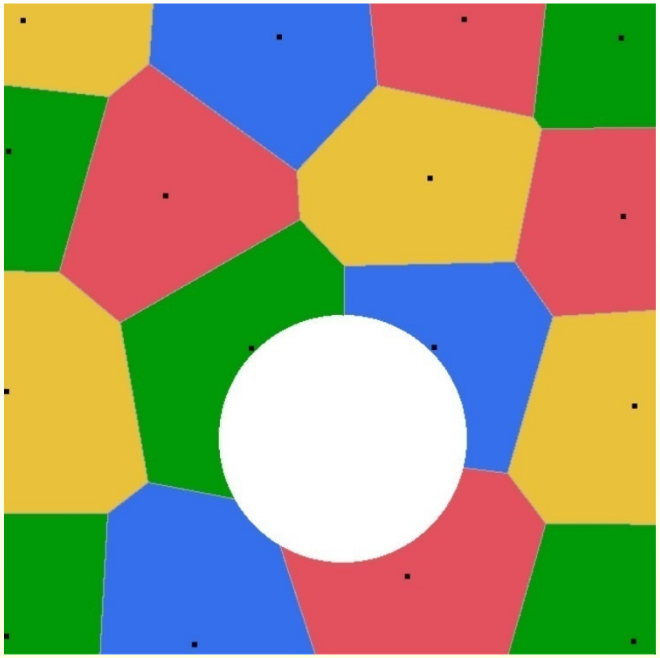
[de Silva, Carlsson '04]

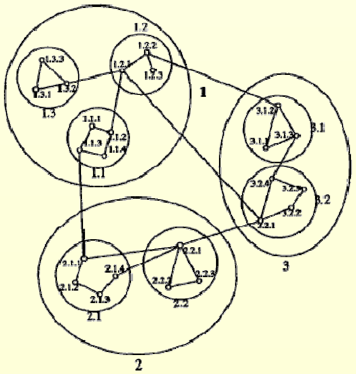
Reconstruction, Using Witness Complexes



Back to Communication Networks

- `Shape from proximity', using landmark network distances (hopcounts) as rough approximations to geodesic distances
- No geographic coordinates are necessary
- Variations on the distance-based witness complex is a perfect vehicle for trying to capture the morphology of a domain
- Must choose landmarks so that the Voronoi tiles are `nice' – simply connected and more





Landmark literature
[Tsuchiya '88, …]

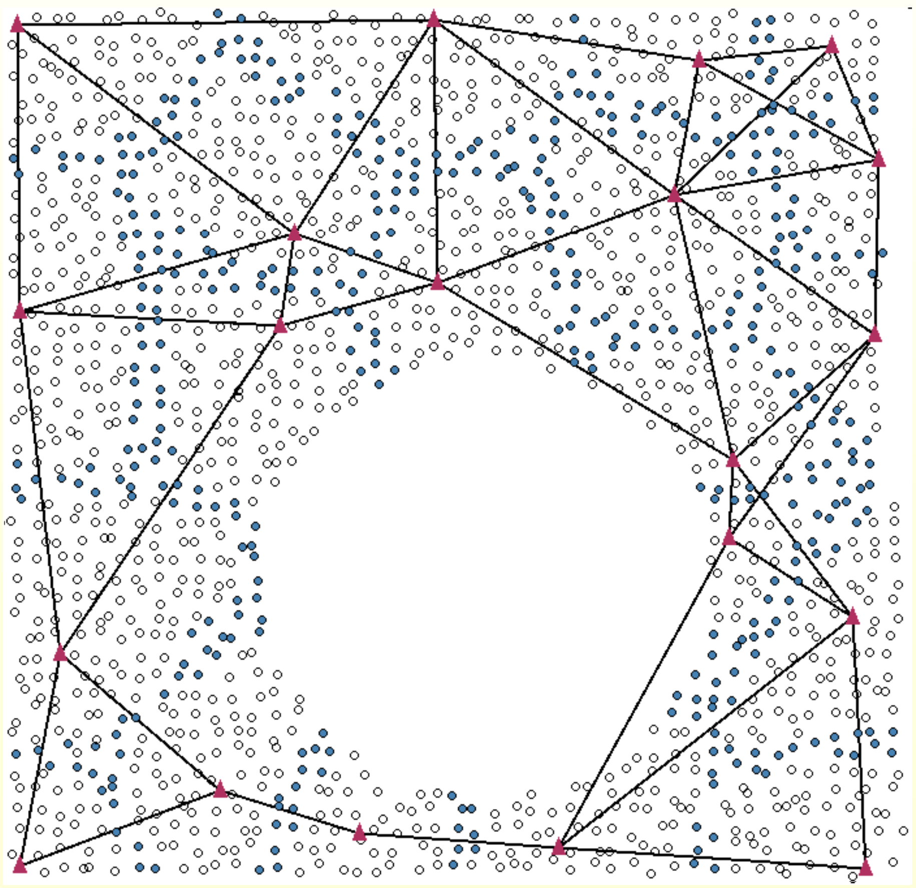
The Nerve Theorem

If a domain D is covered by a family of open sets U_α, $\alpha \in I$, and if for every subset S, $S \subseteq I$, the intersection $\cap_{\alpha \in S} U_\alpha$ is either empty or contractible, then a geometric realization of the nerve complex of the covering is homotopic to D.

[J. LeRay, ‘45]

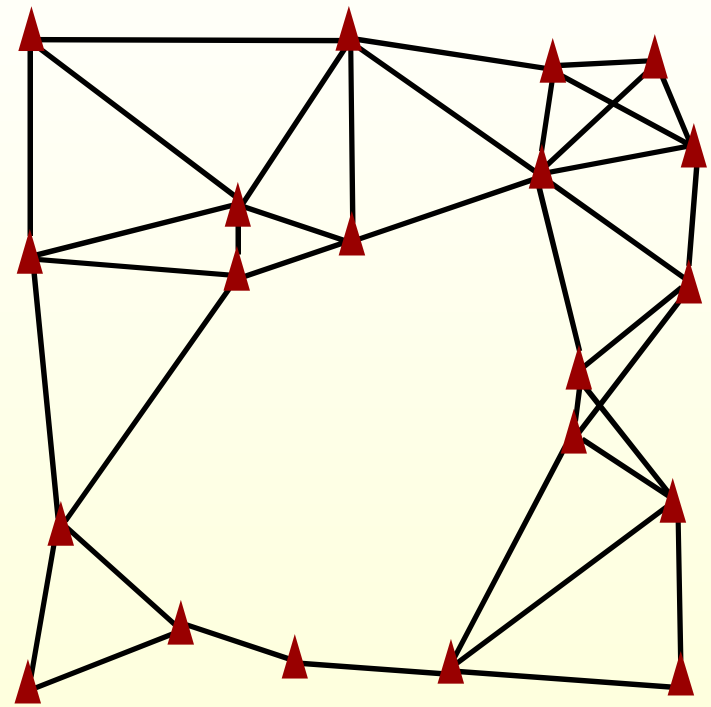
Geodesic Voronoi and Delaunay Complexes

- Given a communication graph G on nodes with distances defined by hop counts
- Perform structure discovery:
  - Select a set of landmarks
  - Construct the Landmark Voronoi Complex (LVC)
  - Extract the Combinatorial Delaunay Complex (CDC) on the landmarks



G and CDC D(L) are Homotopic

- D(L) is compact – topology capture has complexity dependent on the number of large-scale features in the environment
- D(L) is stable – low- level link volatility unlikely to affect the combinatorial complex



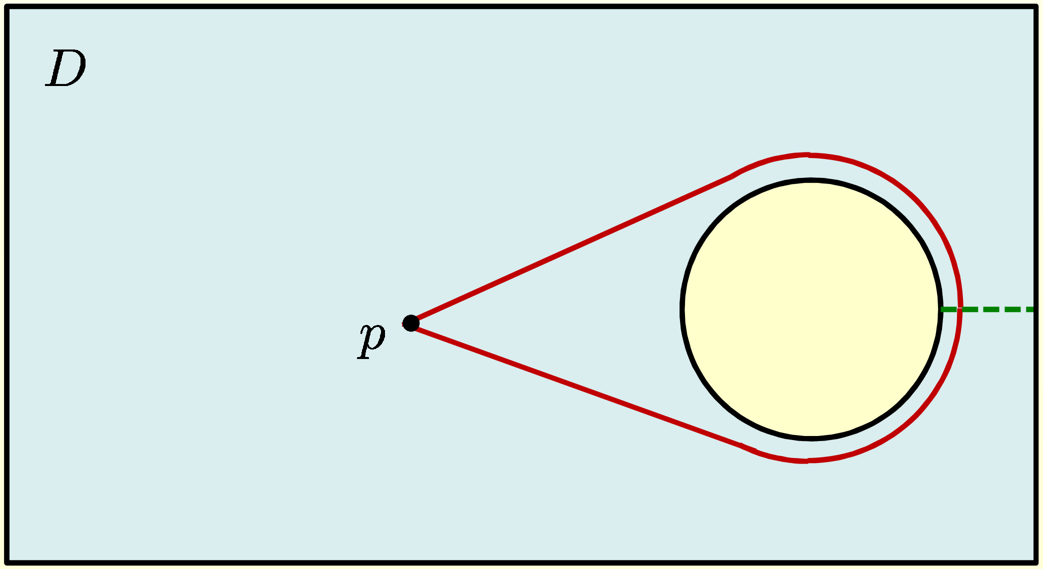
D(L) is a global network atlas that can be known to all landmarks, or even all nodes

Homotopy Feature Size

The homotopy feature size at *p* in a domain *D* is ½ of the length of the shortest non-trial loop through p.
It is also the distance from p to its cut locus.

This defines a sufficient sampling density in a geodesic Voronoi/Delaunay

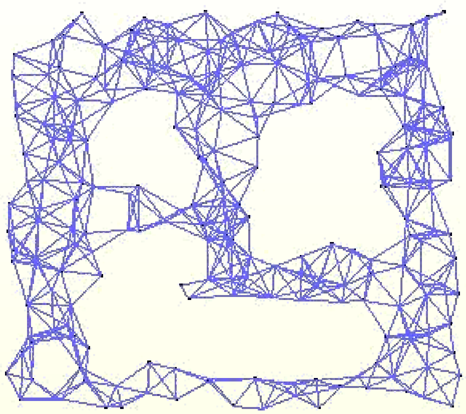
It is a very coarse sampling condition



Landmark Selection to Meet HFS Criterion

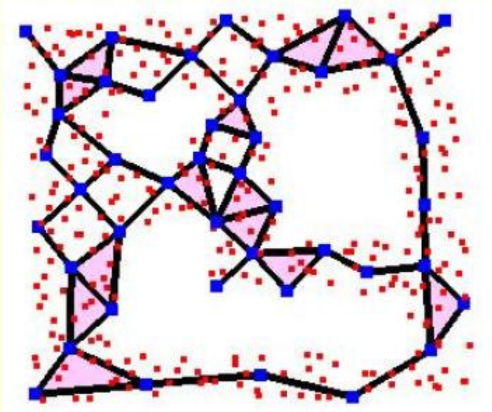
- Sequential:
  - The *i*-th landmark is chosen arbitrarily among any unclaimed nodes
  - It performs a local flood to learn its hfs (distance to cut locus), but restricted by the previously chosen landmarks
  - The landmark claims a portion of its geodesic hfs ball as its Voronoi cell
- Distributed
  - Nodes wake up at random times
  - If unclaimed, they elevated themselves to landmark status and perform the hfs flood described above

Persistent Homology Across Relaxed Witness Complexes

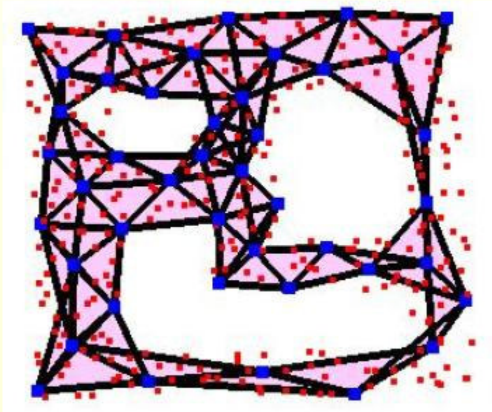


<u>Theorem.</u> Under hfs sampling conditions, the persistent homology from the witness to the relaxed witness complex is true homology of the underlying domain

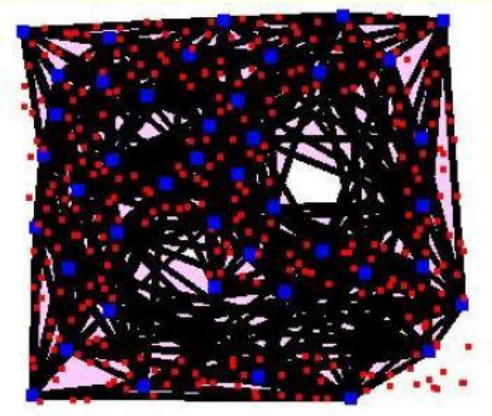
[G., Oudot, Gao, Wang '08]



$\subseteq$

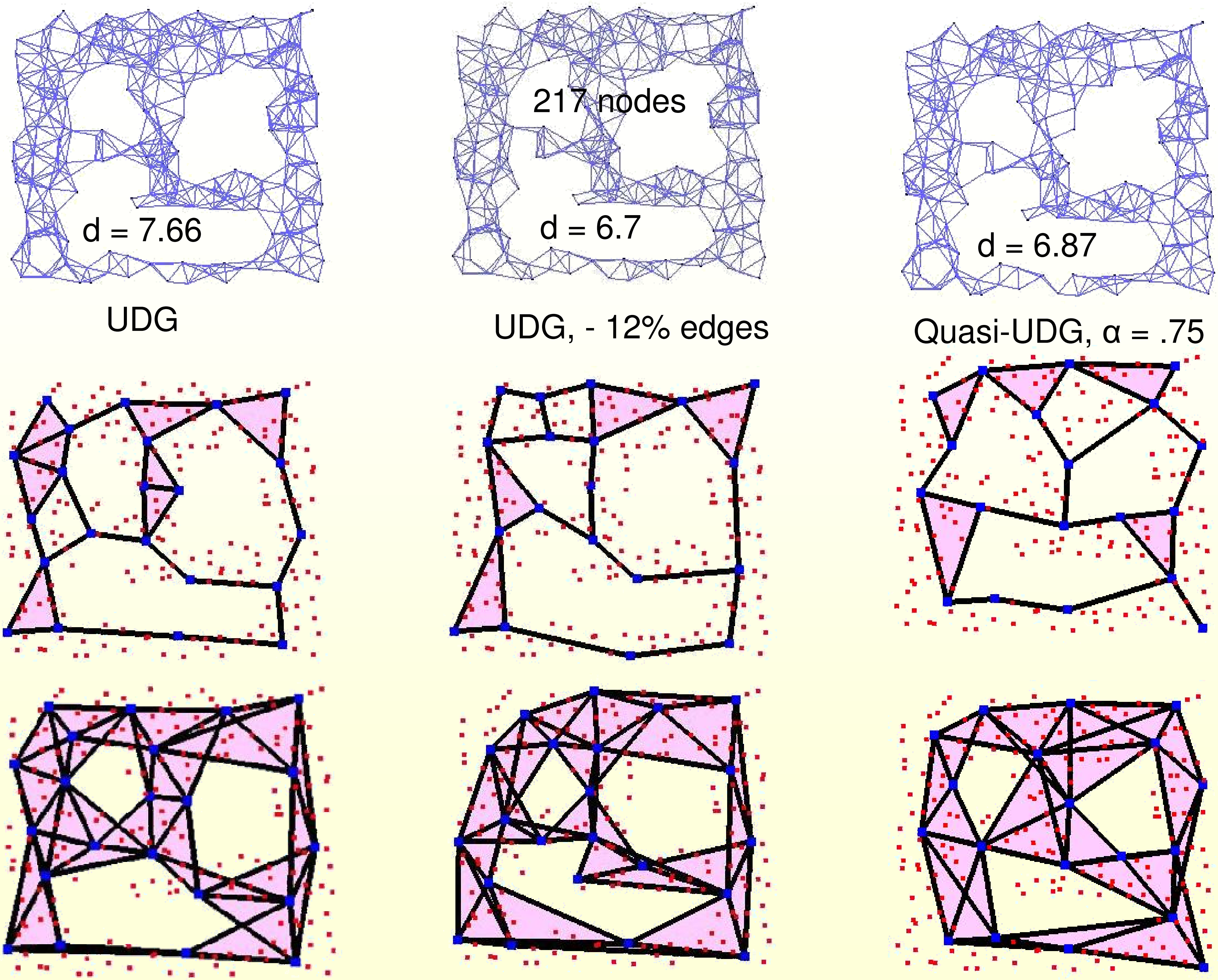


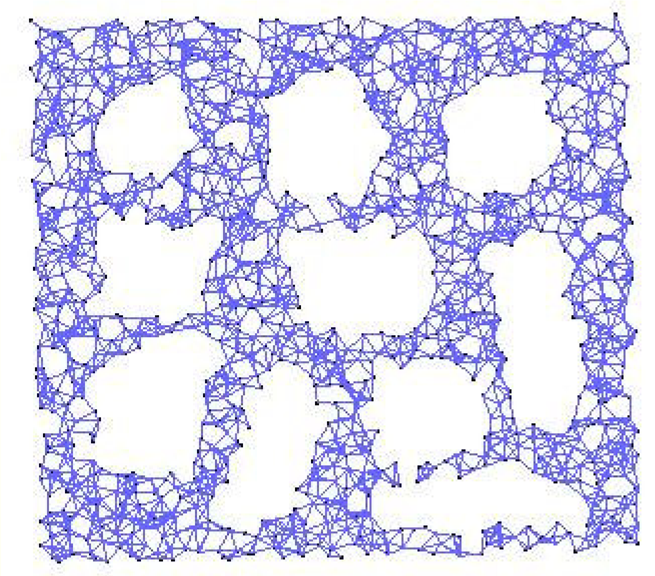
$\subseteq$



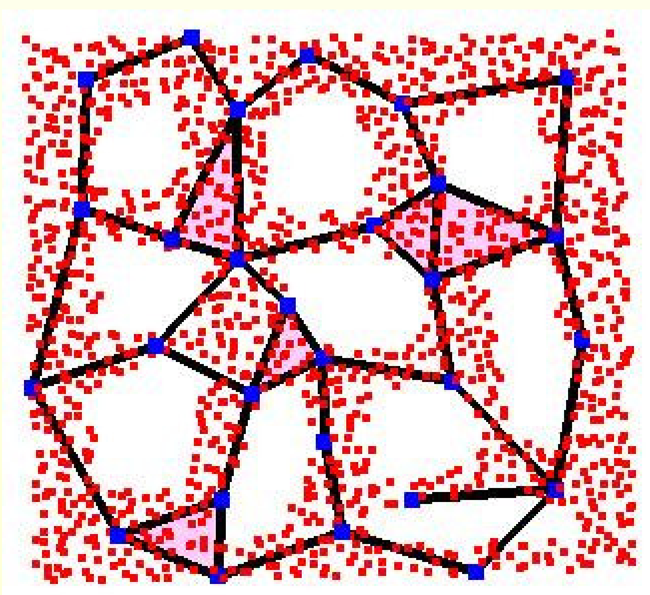
$\nu = 0$ $\nu = 2$ $\nu = 11$

v is the number of \`extra neighbors' allowed

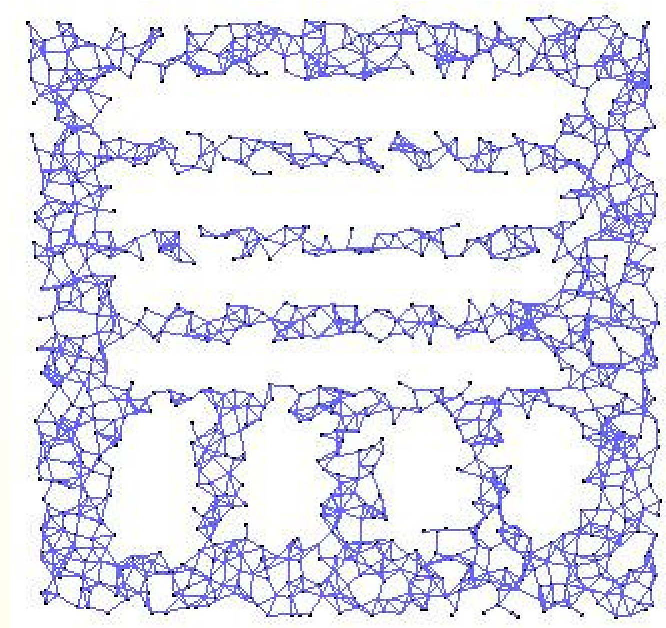

217 nodes
d = 7.66
d = 6.7
d = 6.87
UDG
UDG, - 12% edges
Quasi-UDG, α = .75



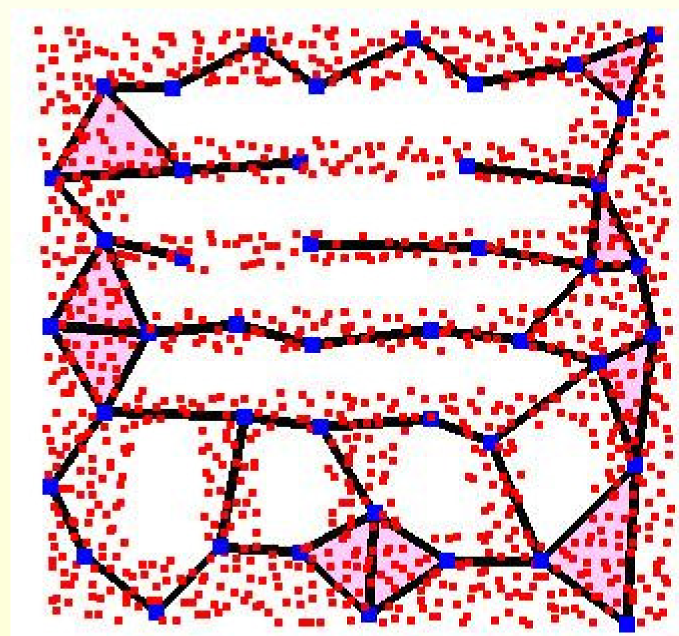
avg. degree = 5.0



Quasi-UDG,
α = .8

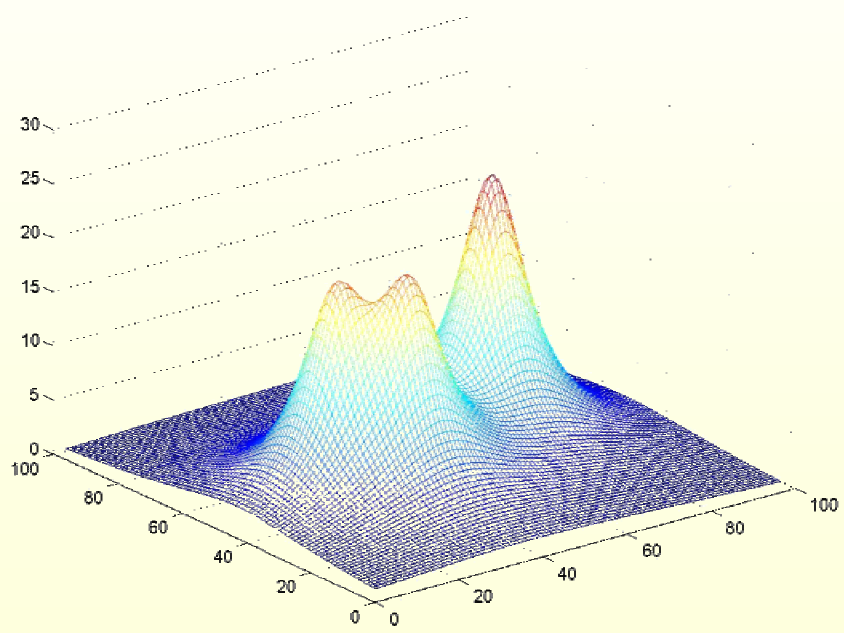


avg. degree = 6.4

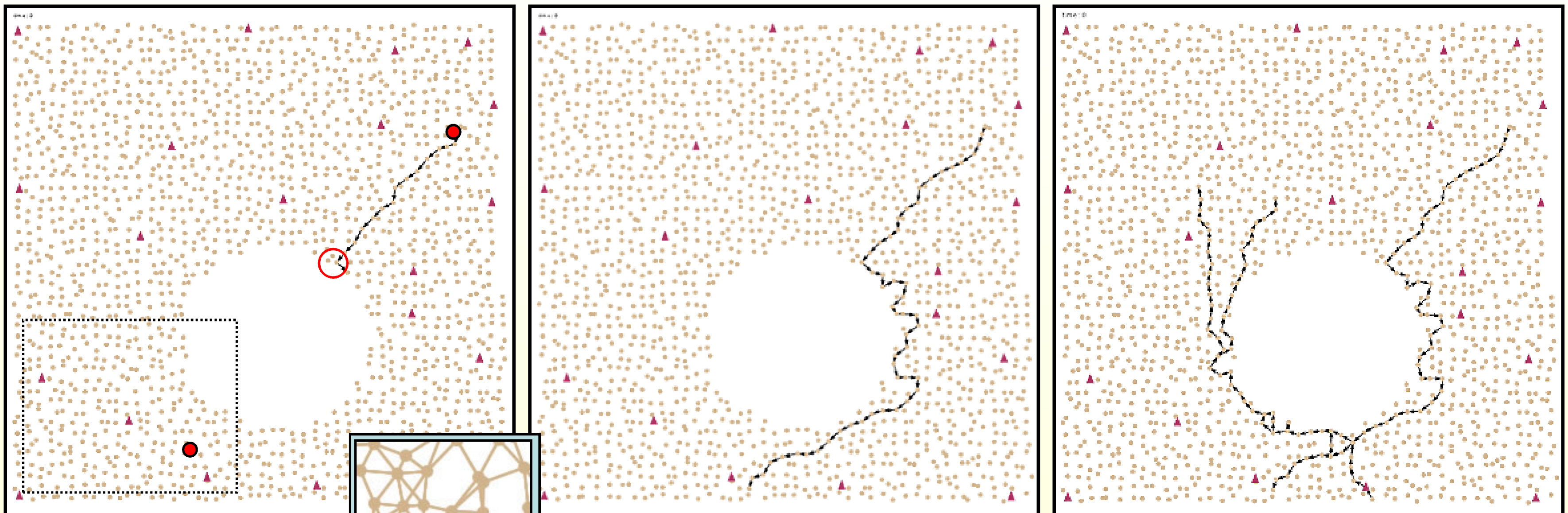


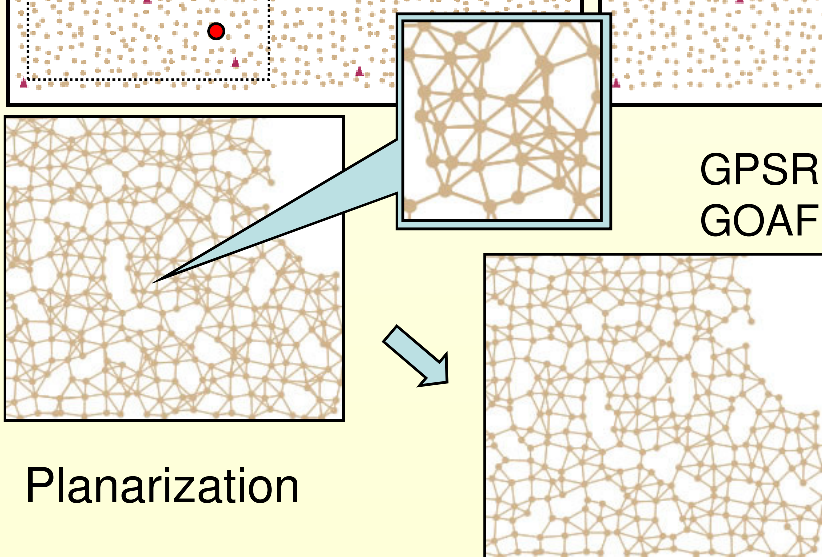
Exploiting Global Structure

- Protocols based on local information only are attractive for mobile and ad hoc networks
- They give rise to greedy distributed algorithms
- Which, however, may get stuck in local minima or maxima
- But perhaps these can be avoided by exploiting the availability of some global structure information



Geographic Routing: Location Provides Global Structure





GPSR [Karp, Kung '00]
GOAFR+ [Kuhn, Wattenhofer, Zhang, Zollinger '03]

These require building a planar subgraph of the connectivity graph – not a robust process

CLDP, LCR [Kim, Govindan, Karp, Shenker '06]
GDSTR [Leong, Liskov, Morris, '06]

Planarization

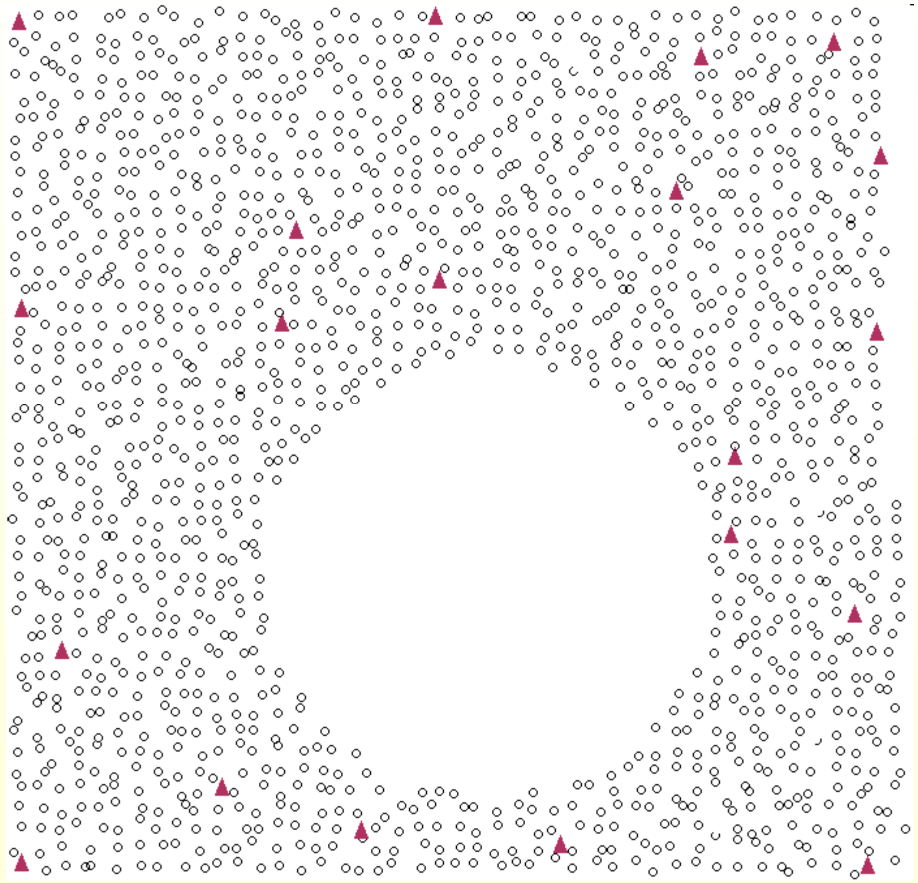
Structure Discovery, Naming and Routing Based on Connectivity Information Only, No Global Embedding

- A two-tier approach, utilizing combinatorial Delaunay complexes and local coordinates (GLIDER)

[Fang, Gao, G., Silva, Zhang, Infocom ’05]

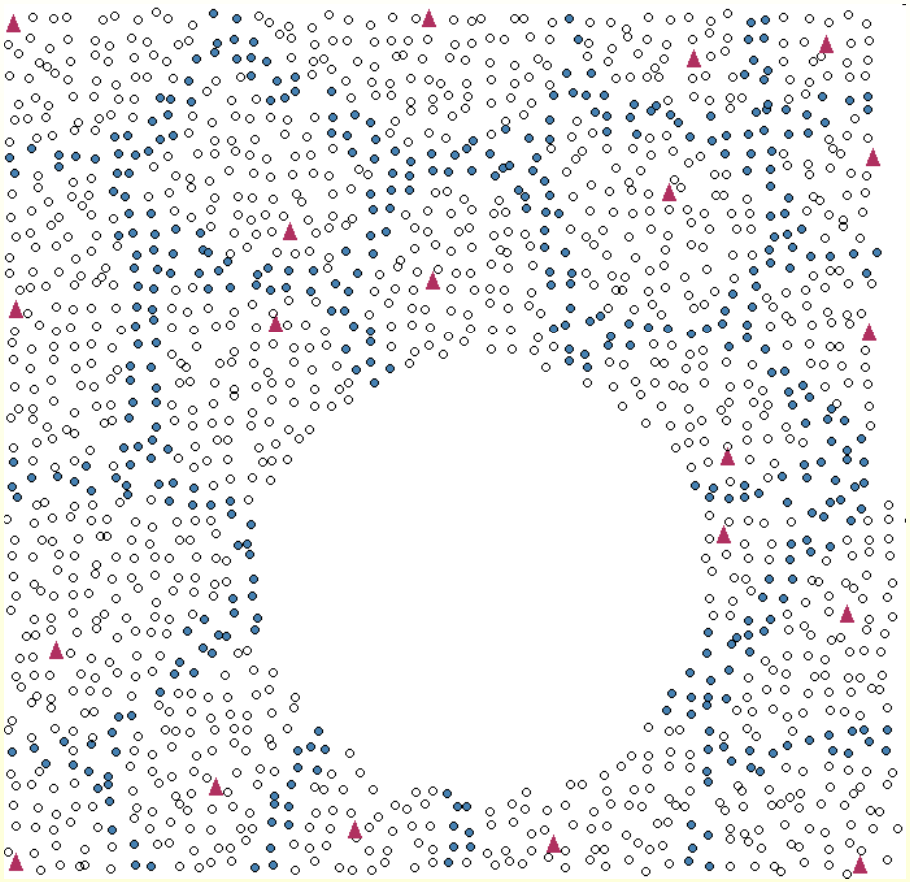
Combinatorial Delaunay Graph

- Given a communication graph on sensor nodes, with path length expressed in shortest path hop counts
- Select a set of landmarks
- Landmarks flood the network. Each node learns its hop count to each landmark
- Construct the Landmark Voronoi Complex (LVC)



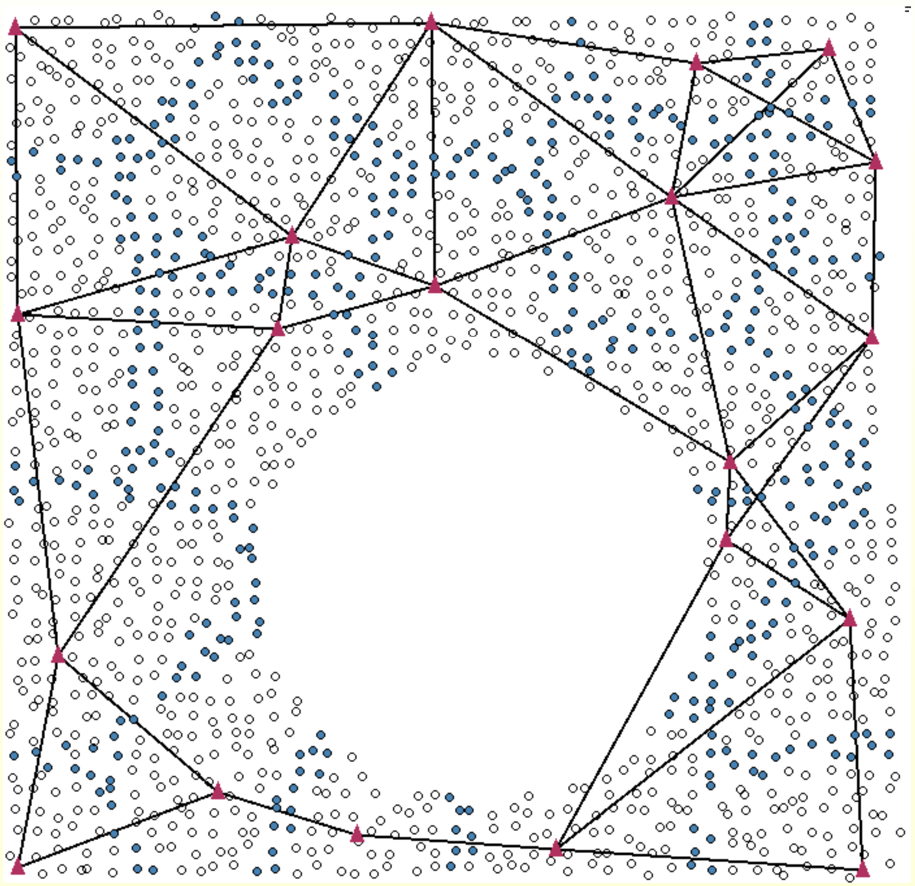
Combinatorial Delaunay Graph

- Construct the Landmark Voronoi Complex (LVC)
- Each sensor identifies its closest landmark
- A sensor is on a tile boundary if it has two closest landmarks
- If the floodings are reasonably synchronized, then restricted flooding up to the boundary nodes is enough



Combinatorial Delaunay Graph

- Construct Combinatorial Delaunay Triangulation (CDT) on landmarks
- If there is at least one boundary node between landmarks *i* and *j*, then there is an edge *ij* in CDT
- Holes in the sensor field map to holes in CDT
- The CDT is broadcast to the whole network



Virtual Coordinates

Each node stores virtual coordinates (d_1, d_2, d_3, … d_k),

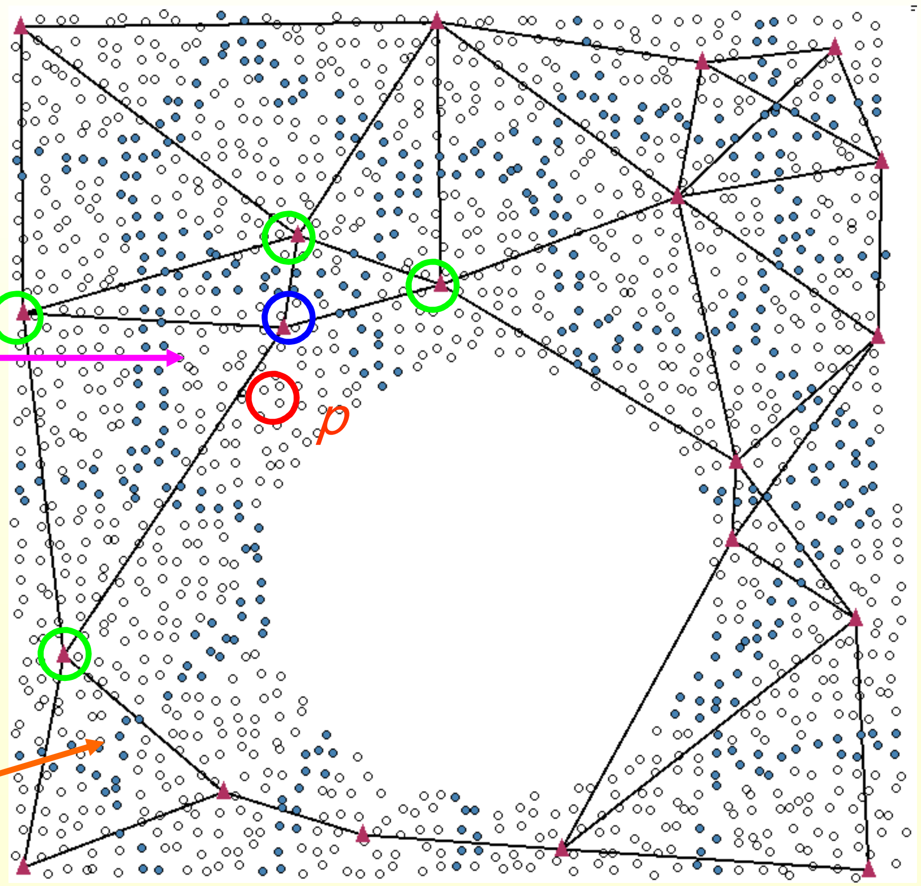
d_k= hop count to the kth reference landmark (home+neighboring landmarks)

home landmark
(think about post-office)

resident tile

reference
landmarks

Boundary
nodes



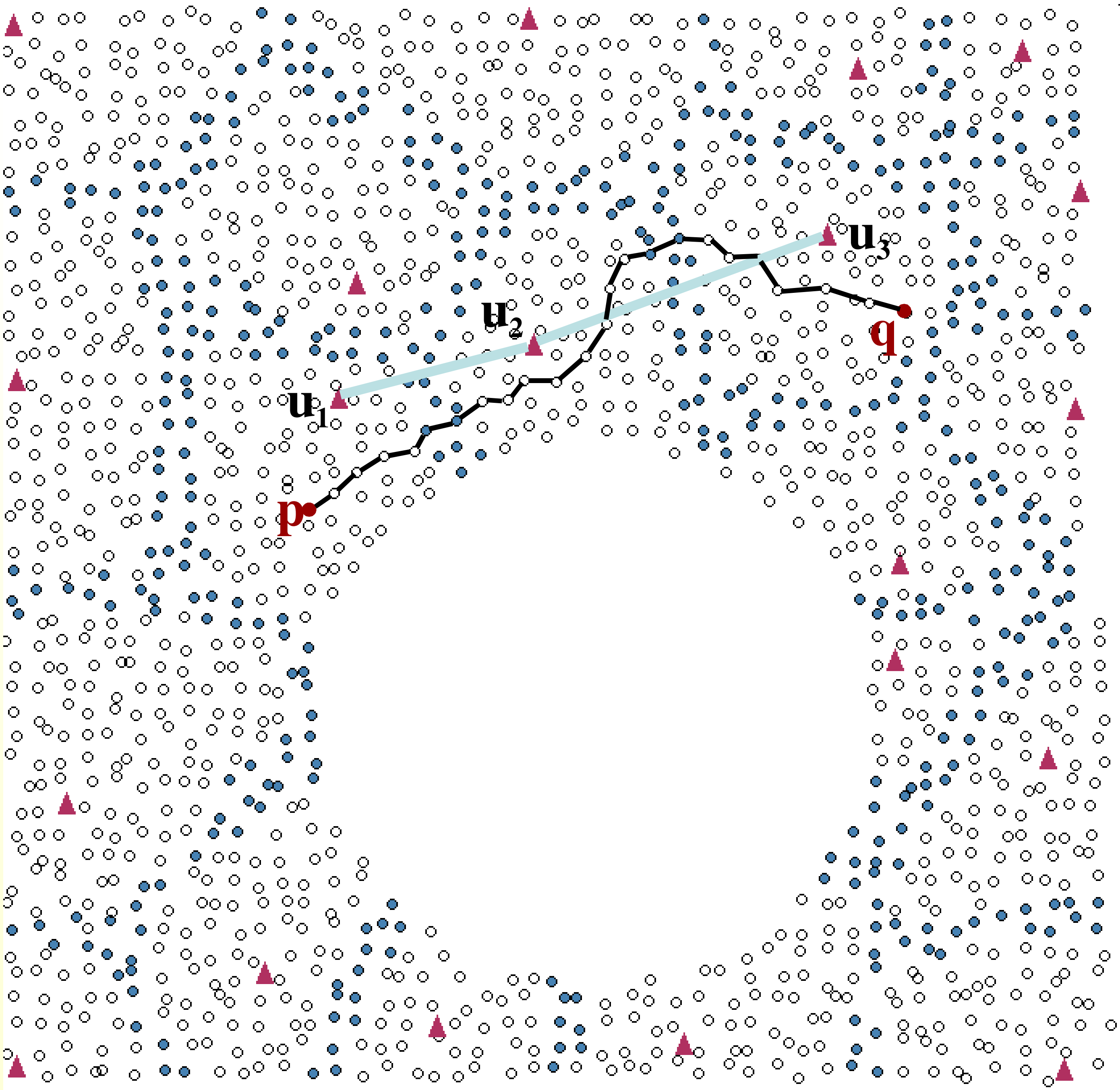
Local Routing, but with Global Guidance: GLIDER

- Global Guidance

  the CDC D(L) encodes global connectivity information that is accessible to every node for proactive route planning on tiles.

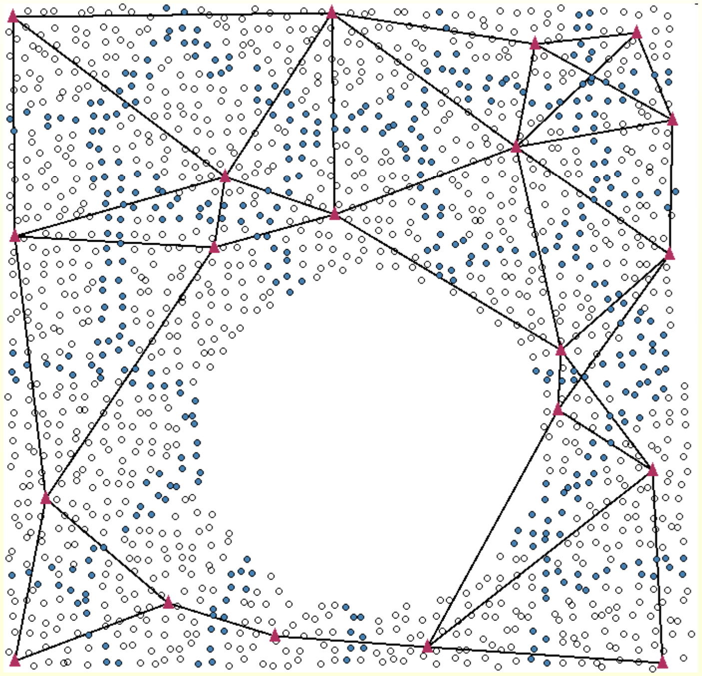
- Local Routing

  high-level routes on tiles are realized as actual paths in the network by using local reactive protocols.


u3
u2
u1
p
q

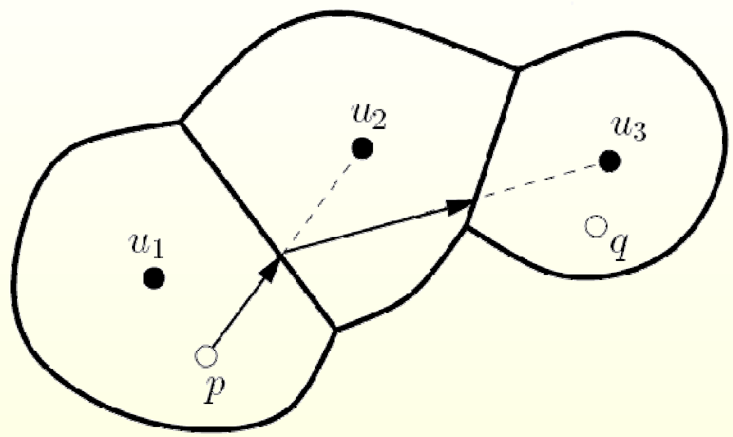
Information Stored at Each Node

- The parents on the shortest paths to its home landmark, and its neighbor landmarks
- A bit to record if the node is on the boundary of a tile
- Its **local coordinates** for intra-tile greedy routing
- Landmark nodes store the atlas D(L)



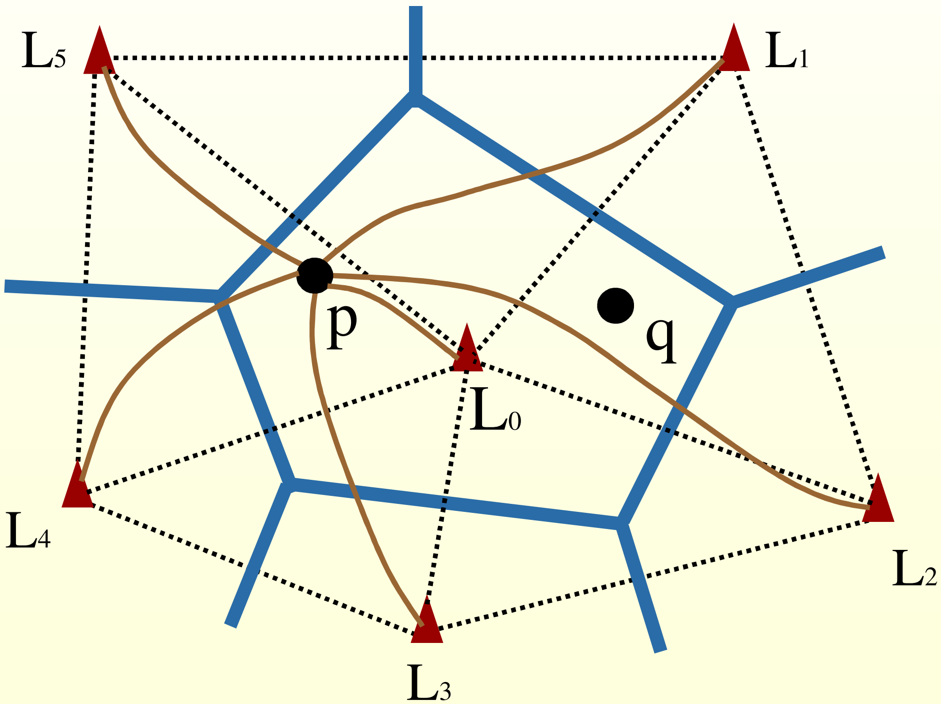
GLIDER -- Routing

- Routing
  - Global route plan
  - Local route
    - inter-tile
    - intra-tile



Both follow certain gradients ...

Going the Last Mile: Local Coordinates and Greedy Routing



Reference landmarks for p:
$L_0, \ldots L_k$ — $T(p) = L_0$

Let $s = \text{mean}(pL_0^2, \ldots, pL_k^2)$

Local virtual coordinates:

$c(p) = (pL_0^2 - s, \ldots, pL_k^2 - s)$

(centered metric)

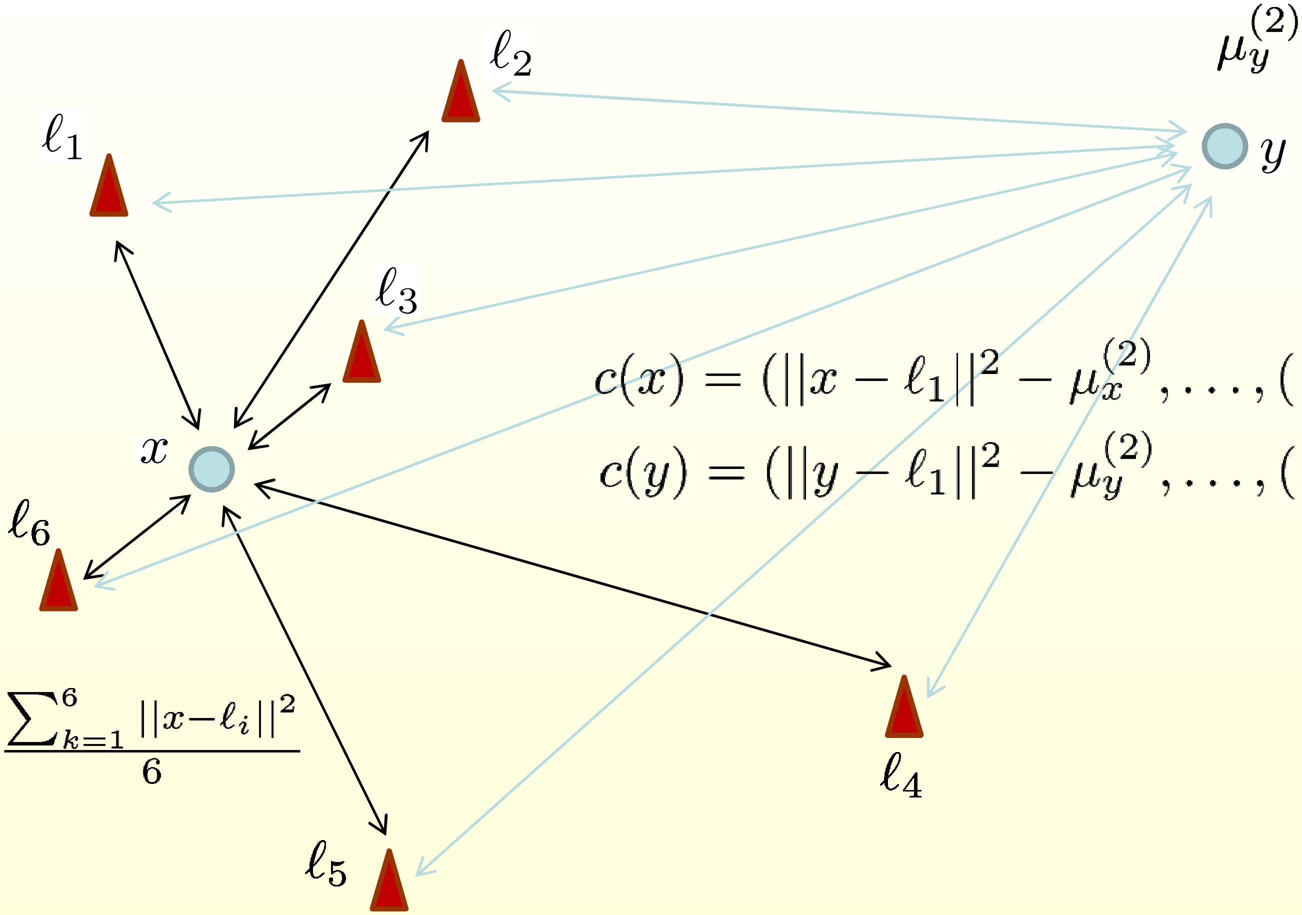
Distance function:

$d(p, q) = |c(p) - c(q)|^2$

Greedy strategy: to reach q, do gradient descent on the function d(p, q)

Landmark-Based Navigation

$$\mu_y^{(2)} = \frac{\sum_{k=1}^{6} ||y - \ell_i||^2}{6}$$



$$c(x) = (||x - \ell_1||^2 - \mu_x^{(2)}, \ldots, (||x - \ell_6||^2 - \mu_x^{(2)})$$

$$c(y) = (||y - \ell_1||^2 - \mu_y^{(2)}, \ldots, (||y - \ell_6||^2 - \mu_y^{(2)})$$

$$\mu_x^{(2)} = \frac{\sum_{k=1}^{6} ||x - \ell_i||^2}{6}$$

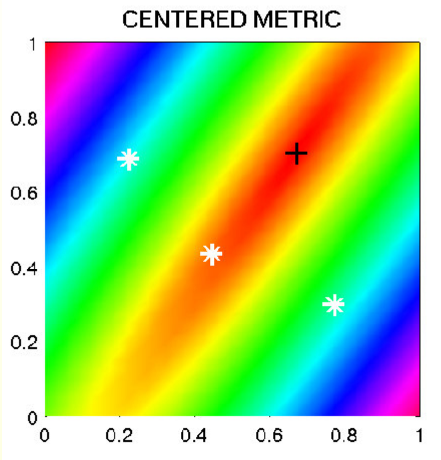
$$d(x, y) = ||c(x) - c(y)||^2$$

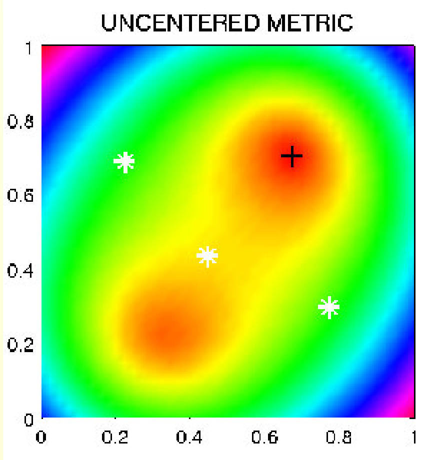
Local Landmark Coordinates – No Local Minima

Navigation, without localization ...

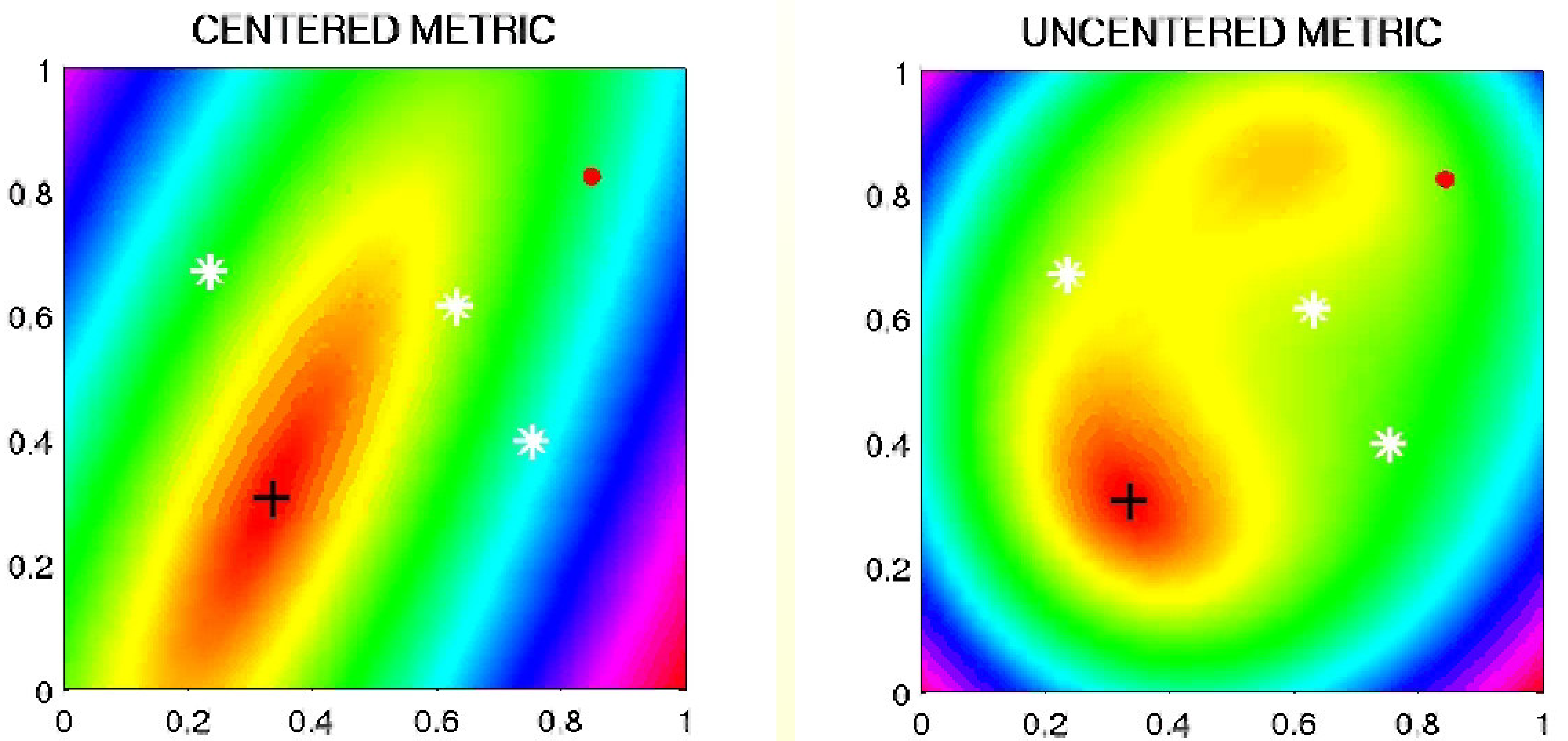
- Theorem: In the continuous Euclidean plane, gradient descent on the function d(p, q) always converges to the destination q, provided that there are at least three non-collinear landmarks.

- In the discrete case, we empirically observe that landmark gradient descending does not get stuck on networks with reasonable density (each node has on average six neighbors or more).





Centered vs. Uncentered Metrics



Node Density vs. Success Rate of Greedy Routing

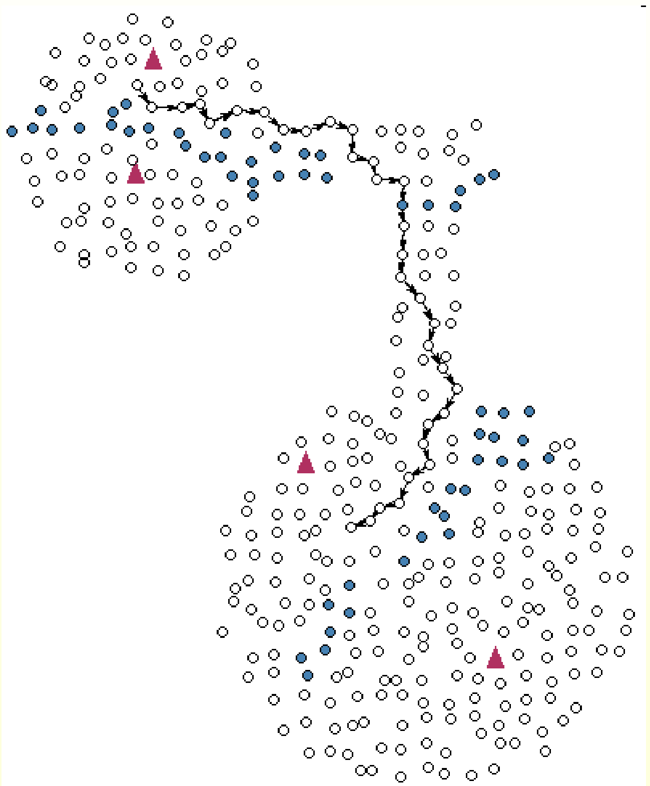
In the discrete case, we empirically observe that following the landmark gradient descending does not get stuck on networks with reasonable density (each node has on average six neighbors or more).

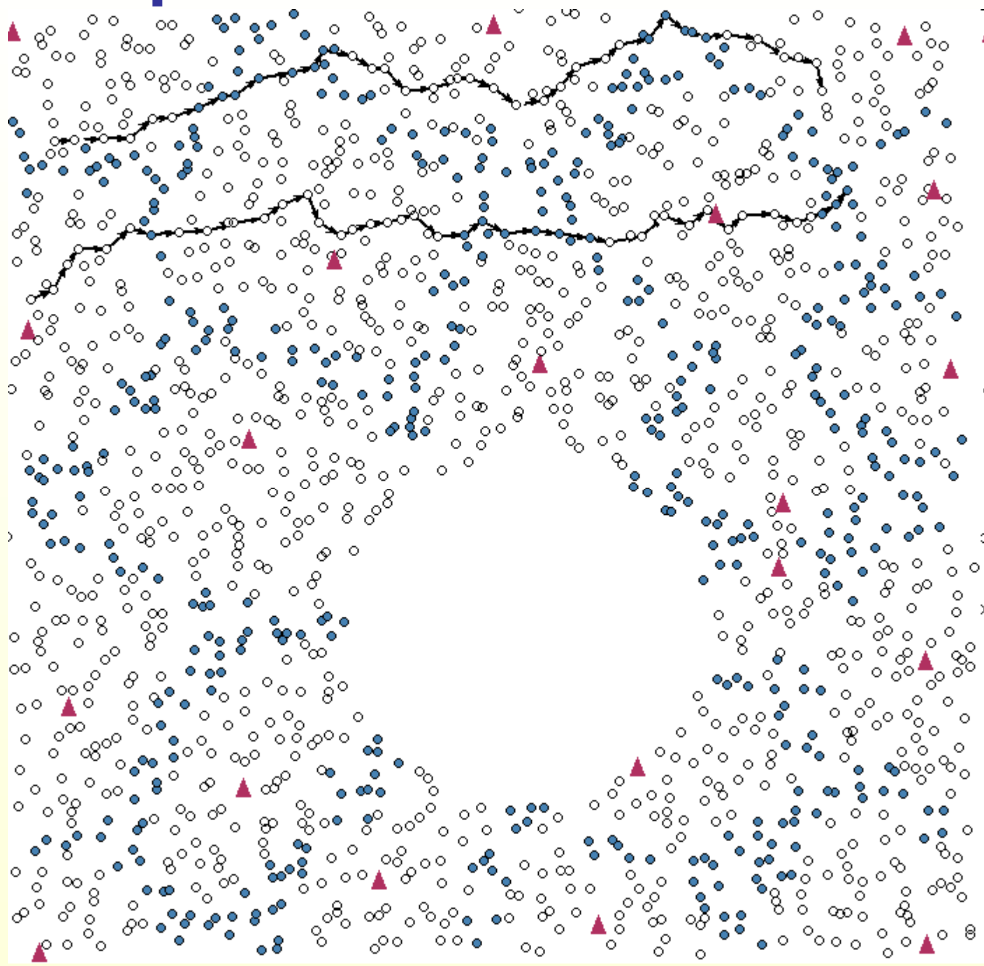
2000 nodes distributed on a perturbed grid.

Perturbation ~ Gaussian(0, 0.5r), where r is the radio range

average number of neighbors	2.9	3.2	4.1	≥ 5.3
percentage of success	20	70	95	100

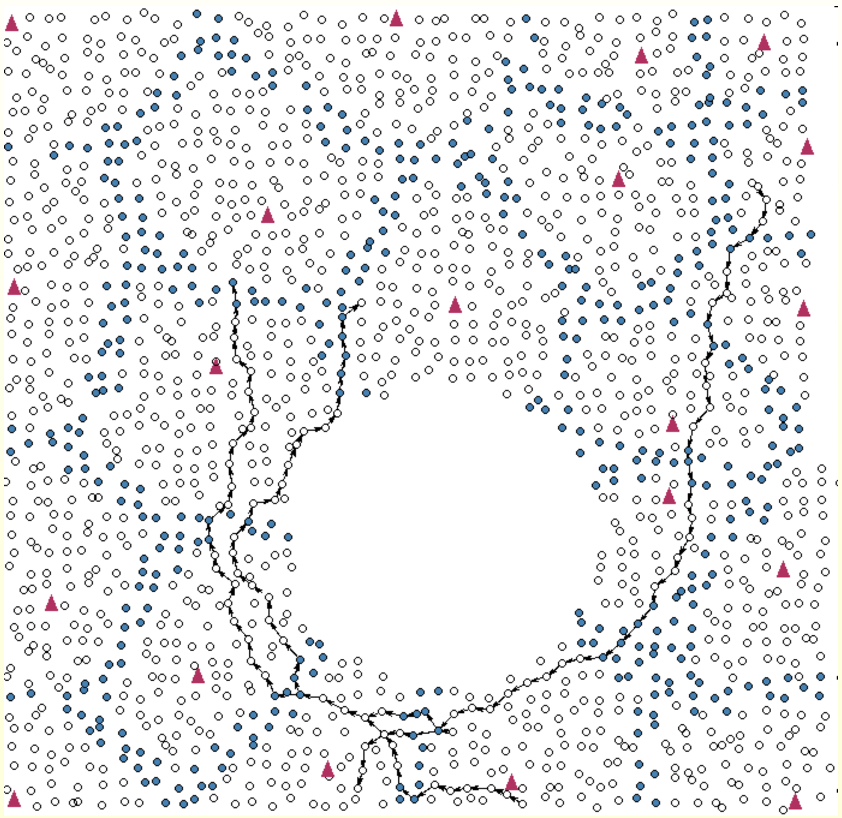
Examples



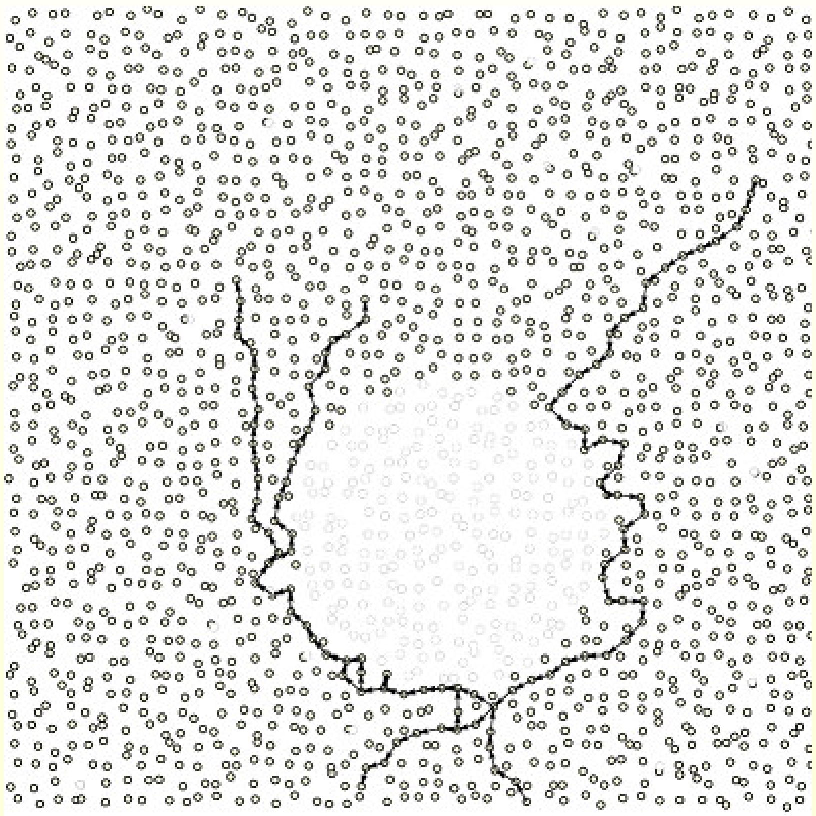


Each node on average has 6 one-hop neighbors.

Simulations – Path Length and Load Balancing



GLIDER

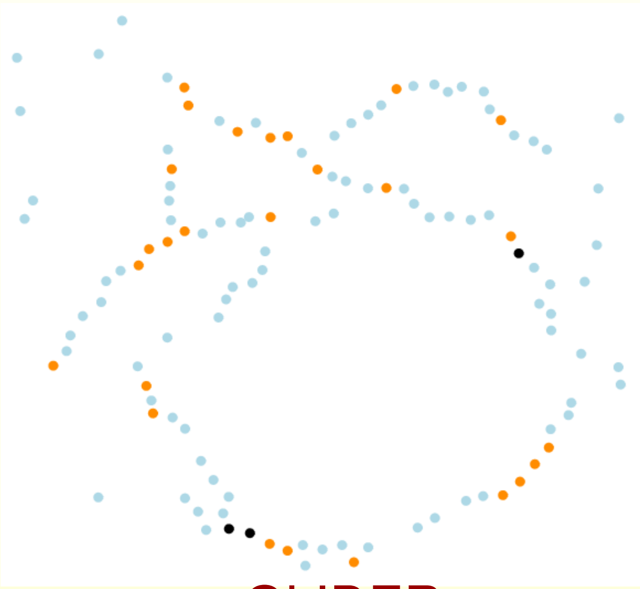


GPSR

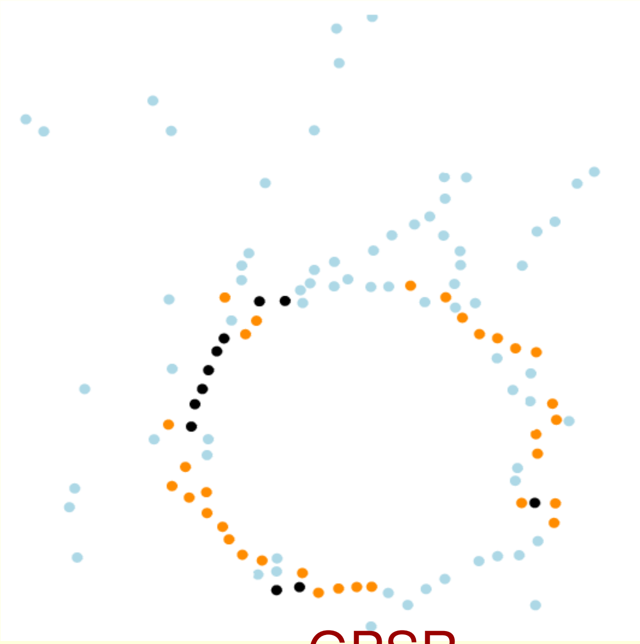
Each node on average has six one-hop neighbors

Simulations – Hot Spot Comparison

Randomly picked 45 source and destination pairs, each separated by more than 30 hops.



GLIDER



GPSR

Blue (6-8 transit paths), orange (9-11 transit paths), black (>11 transit paths)

Stability

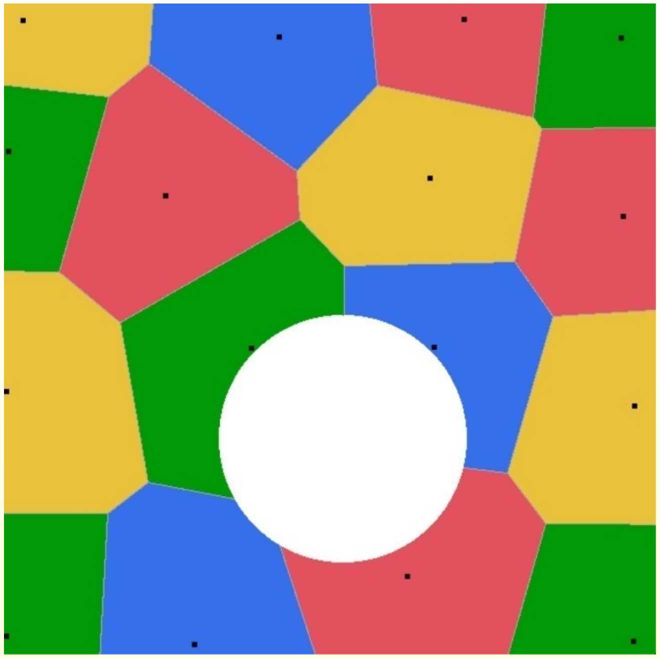
- Landmark failure.
  - Not a big problem as the landmark is simply a point of reference.
  - Not like gateway.
- Combinatorial Delaunay edge change?
  - Requires big change in the network topology (merge of holes, disconnect a band of nodes, etc).

Open issues in GLIDER

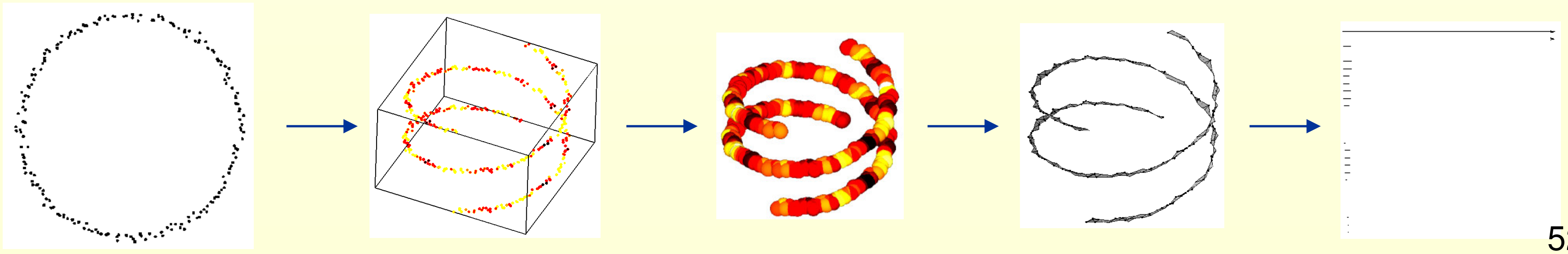
- Further study of landmark selection protocols
- What can we say about the intra-tile greedy routing on a discrete network?
- -- One of the challenges is that the hop count is a rough estimation of the Euclidean distance.

GLIDER Summary

- Naming and routing based connectivity information alone
- Local node names
- Lightweight, compact, robust global guidance for local greedy methods
- CVD and CDC provide tools for detecting the field morphology, including boundaries, holes, narrow passages, etc.

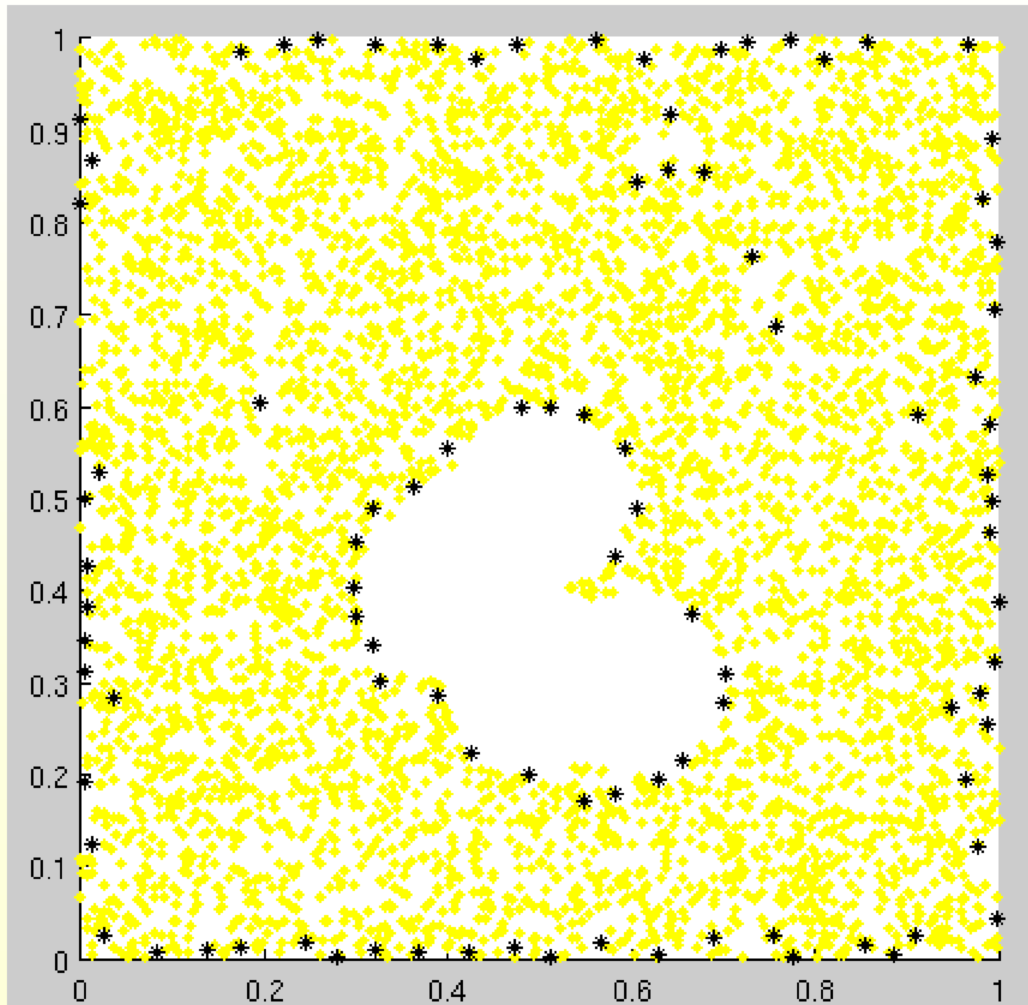


"Shape from Proximity"

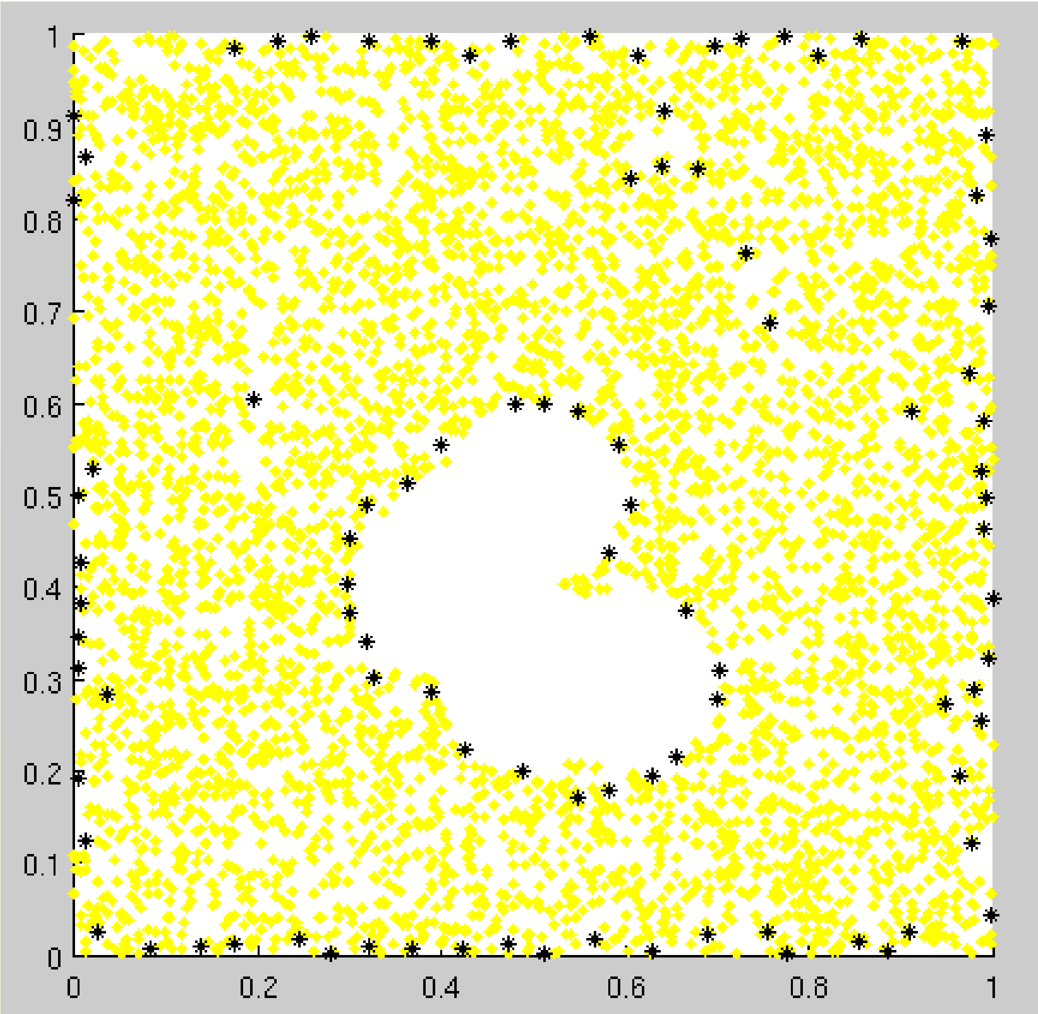


Sensor Layout Morphology via CVC+CDC

Sensor field boundary detection



Boundary Detection



Structure Discovery, Naming and Routing Based on Connectivity Information Only, No Global Embedding

- II. A global approach, using all landmarks (beacon vector routing, or BVR)

[Fonseca, Ratnasamy, Zhao, Tien Ee, Culler, Shenker and Stoica ’05]

Beacon-Vector: Algorithm

- Two pieces
  - Deriving node network coordinates
  - Forwarding rules

Beacon-Vector: Deriving Corrdinates

1. We have *r* beacon nodes ($B_0,B_1,\ldots,B_r$) flood the network; a node q's position, P(q), is its distance in hops to each beacon

   $P(q) = \langle B_1(q), B_2(q),\ldots,B_r(q) \rangle$

2. Node *p* advertises its coordinates using the *k* closest beacons (we call this set of beacons $C(k,p)$)
3. Nodes know their own and neighbors' positions
4. Nodes also know how to get to each beacon

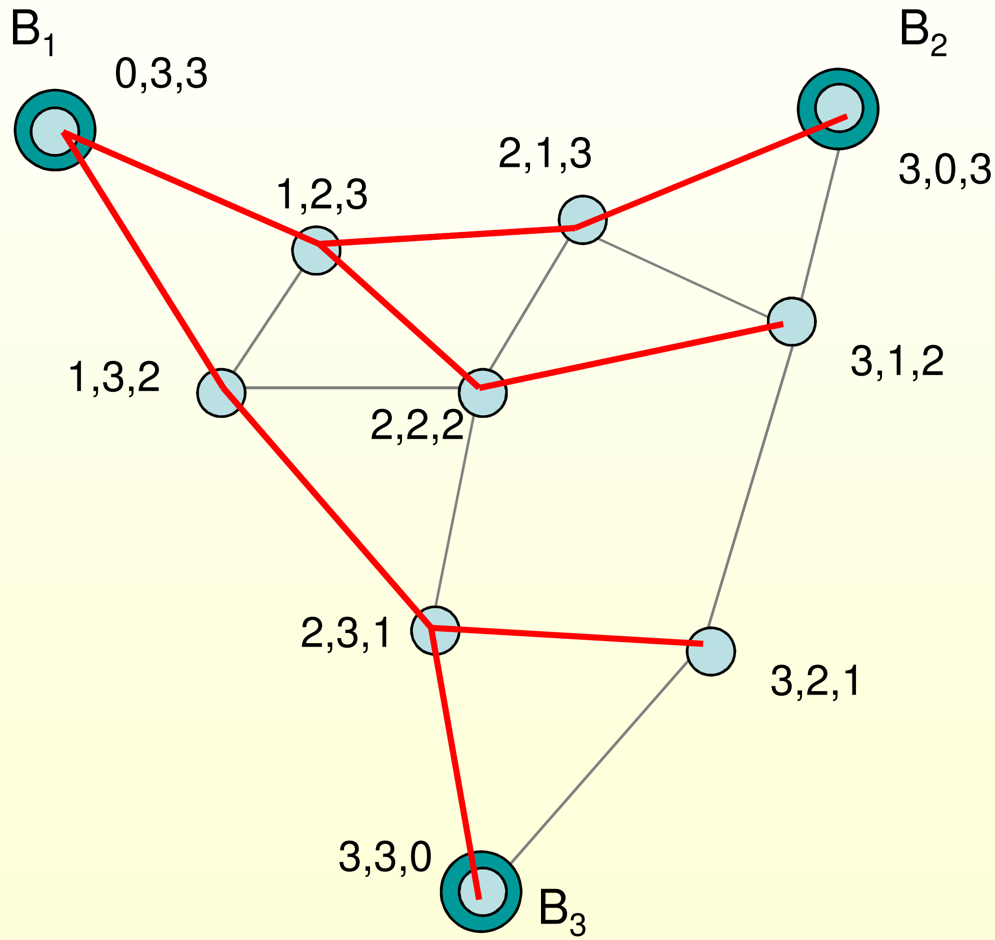
Beacon-Vector: Forwarding

1. Define the distance between two nodes *p* and *q* as

$$\text{dist}_k(p,q) = \sum_{i \in C(k,q)} \omega_i \left| B_i(p) - B_i(q) \right|$$

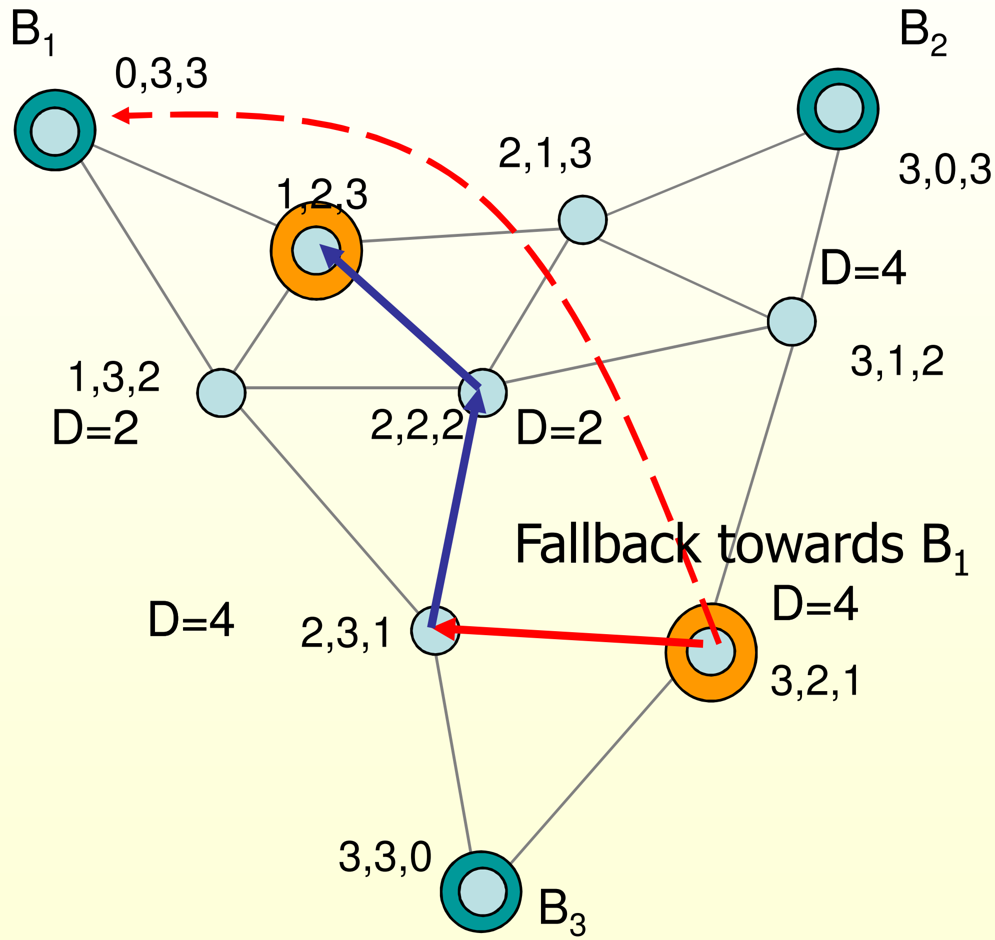
2. To reach destination q, choose neighbor to reduce $\text{dist}_k(*,q)$
3. If no neighbor improves, enter Fallback mode: route towards the beacon which is closer to the destination
4. If Fallback fails, and you reach the beacon, do a scoped flood

Simple example



Simple example

Route from 3,2,1 to 1,2,3



Routing Metric (BVR)

- Routing metric:
  - Pulling landmarks for *p* (those closer to destination *q*).

$$\delta_k^+(p,q) = \sum_{i \in C(k,p)} \max(p_i - q_i, 0)$$

  - Pushing landmarks (further to destination)

$$\delta_k^-(p,q) = \sum_{i \in C(k,p)} \max(q_i - p_i, 0)$$

p_i = hop distance of *p* to beacon *i*
q_i = hop distance of *q* to beacon *i*

Routing Metric (BVR)

- Routing metric: Choose a neighbor that minimizes

$$\delta_k = A\delta_k^+ + \delta_k^-$$

$A \approx 10$

- No theoretical understanding of the performance.

Evaluation - Simulation

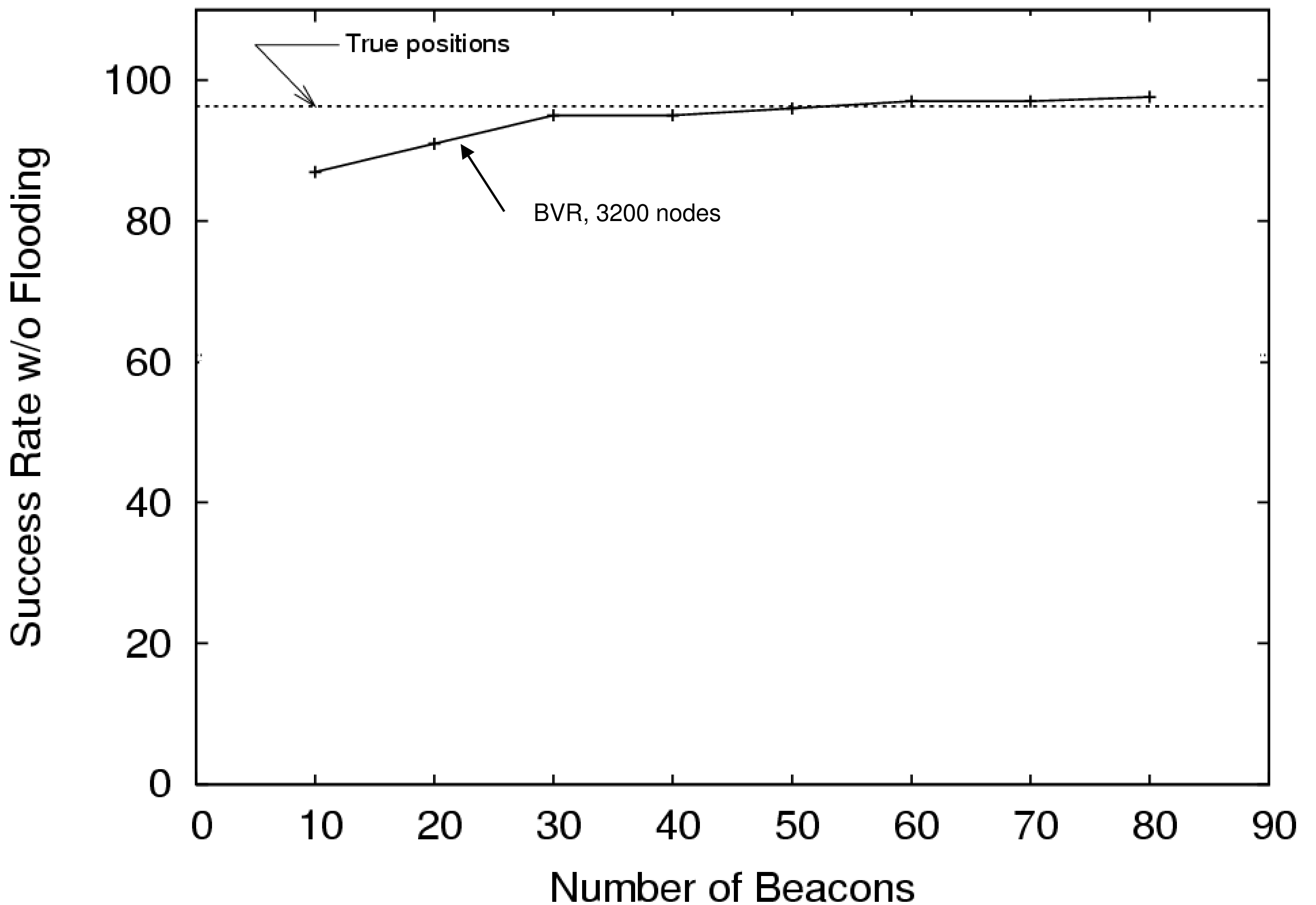
- Packet level simulator in C++
- Simple radio model
  - Circular radius, “boolean connectivity”
  - No loss, no contention
- Larger scale, isolate algorithmic issues

Evaluation - Implementation

- Real implementation and testing in TinyOS on mica2dot Berkeley motes
- 4KB of RAM!
  - Judicious use of memory for neighbor tables, network buffers, etc
- Low power radios
  - Changing and imperfect connectivity
  - Asymmetric links
  - Low correlation with distance
- Two testbeds
  - Intel Research Berkeley, 23 motes
  - Soda Hall, UCB, 42 motes

BVR Simulation Results

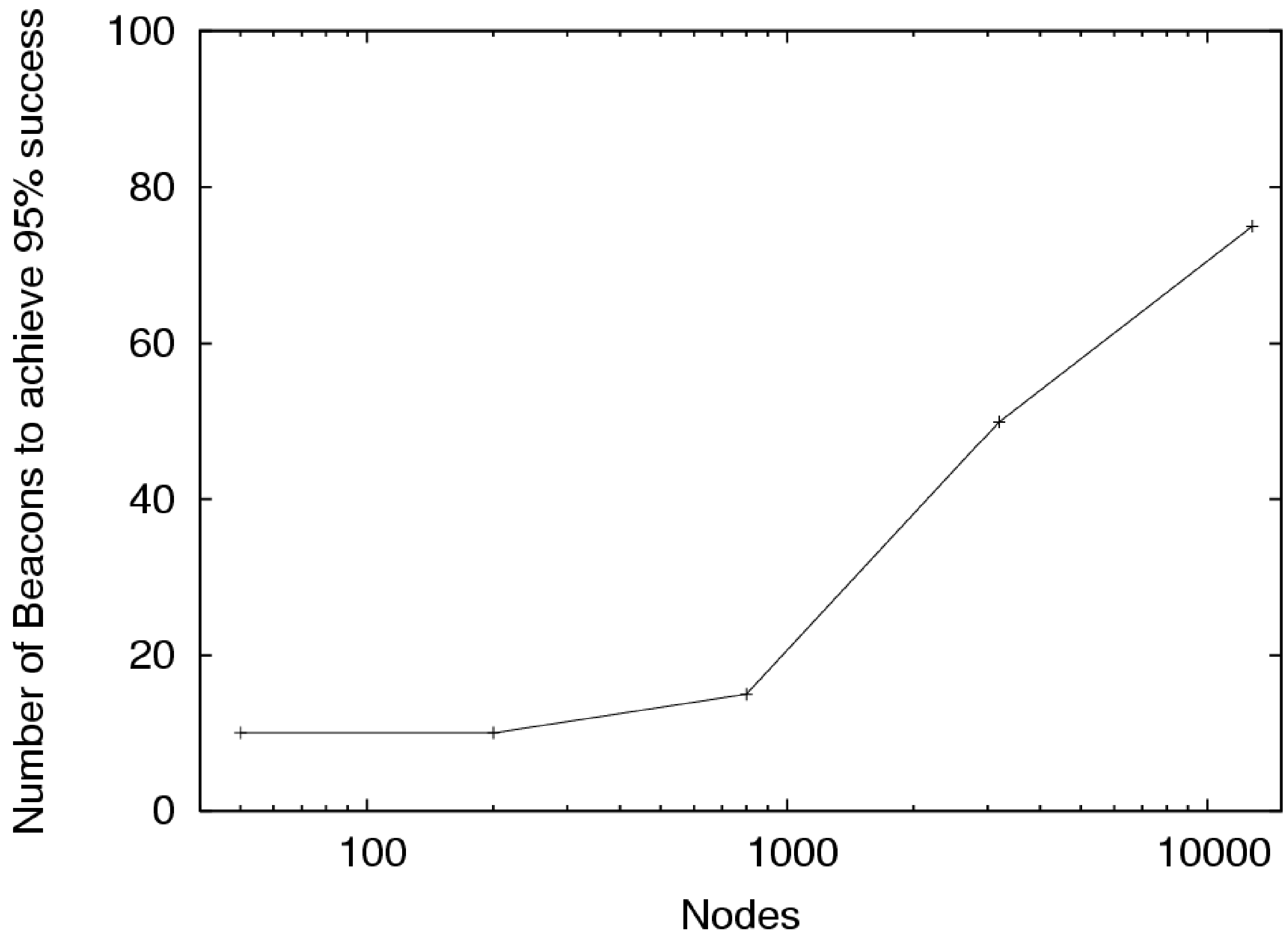
Effect of the number of beacons



Can achieve performance comparable to using true (geographic) positions

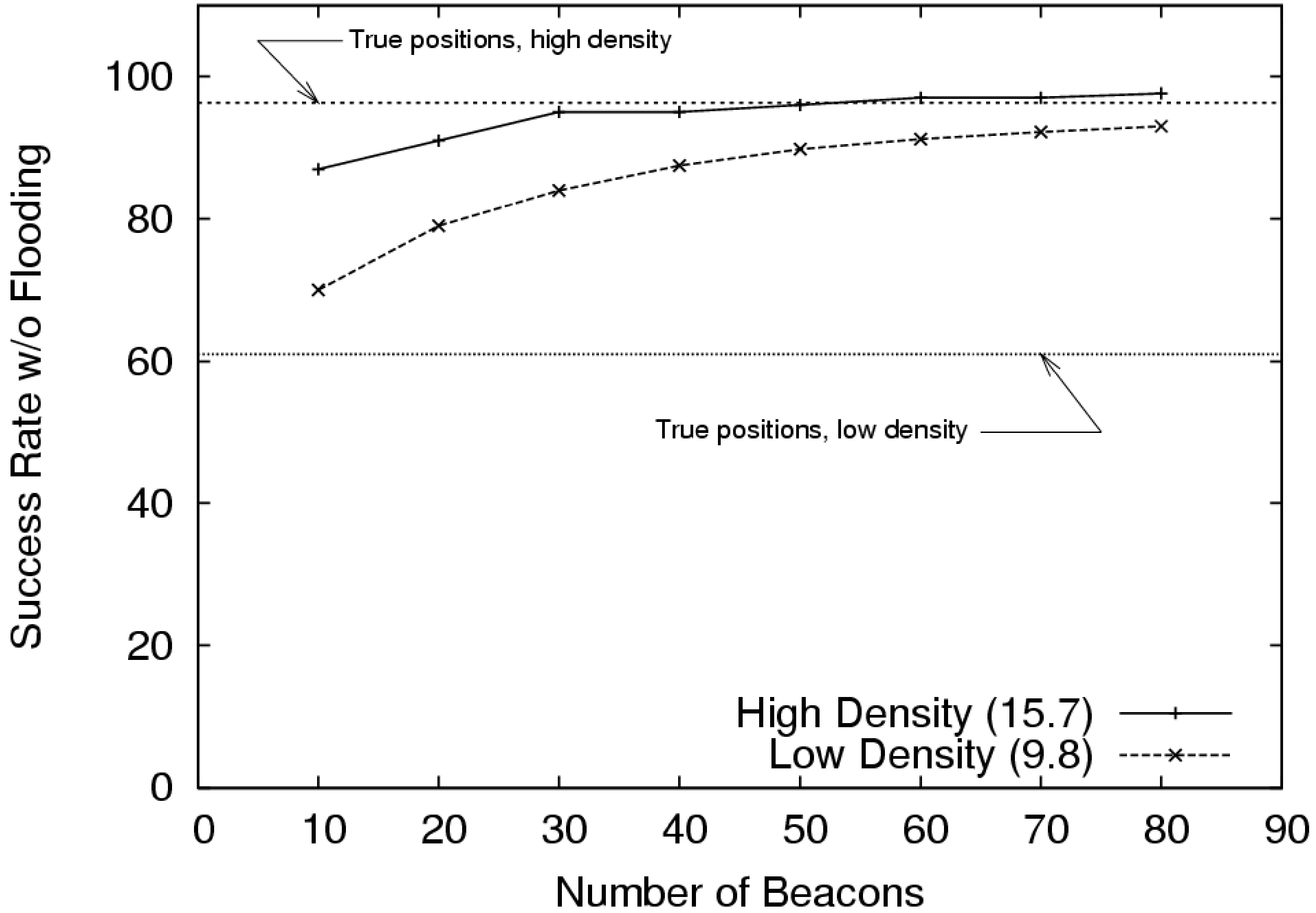
Scaling the Number of Nodes

Number of beacons needed to sustain 95% performance



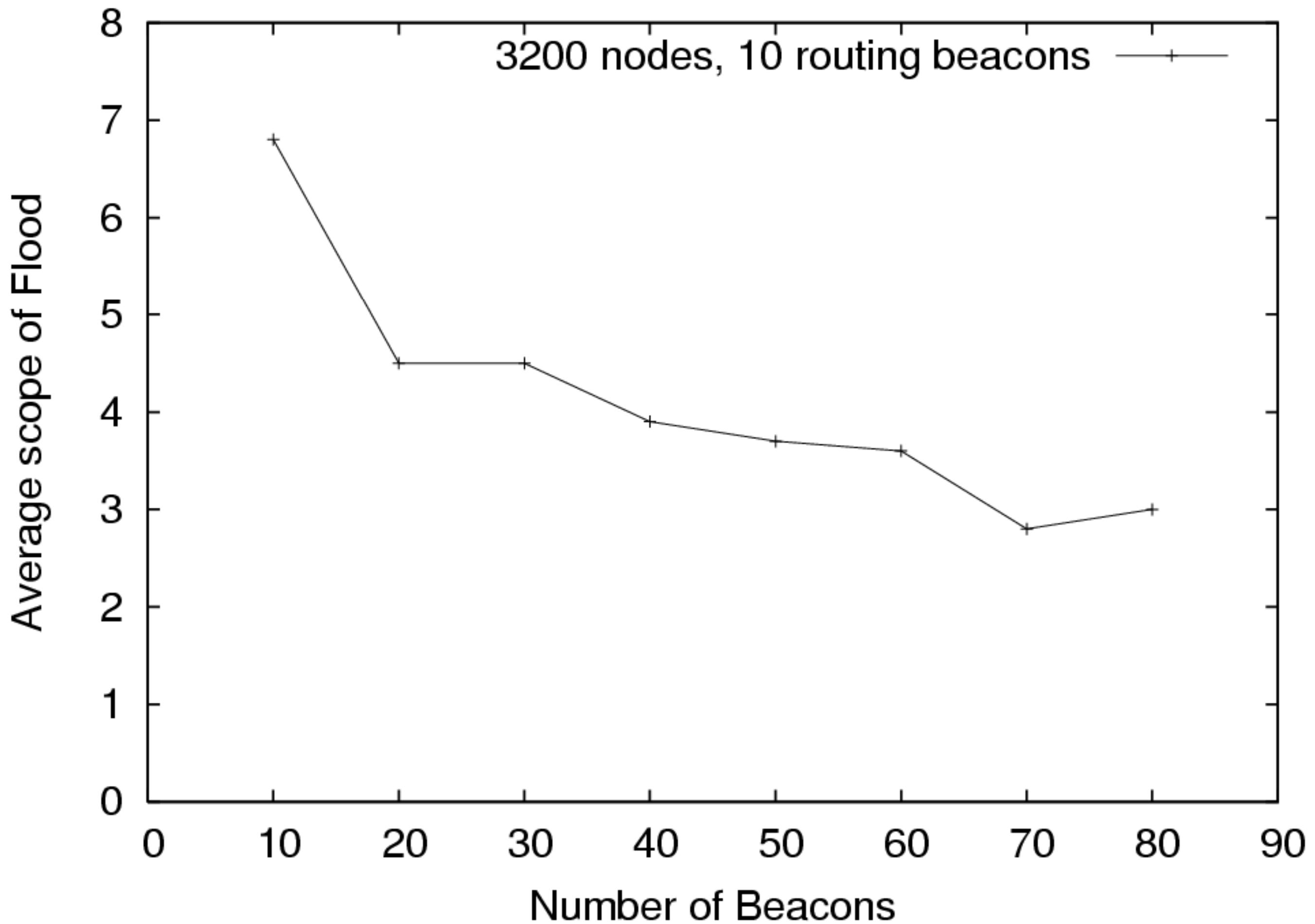
Beaconing overhead grows slowly with network size (less than 2% of nodes for larger networks)

Effect of Density



Great benefit for deriving coordinates from connectivity, rather than positions

Scope of Floods



Other results from simulation

- Stretch was consistently low
  - Less than 1.1 in all tests
- Performance with obstacles
  - Modeled as walls in the network ‘arena’
  - Very low sensitivity for obstacles, differently from geographic forwarding

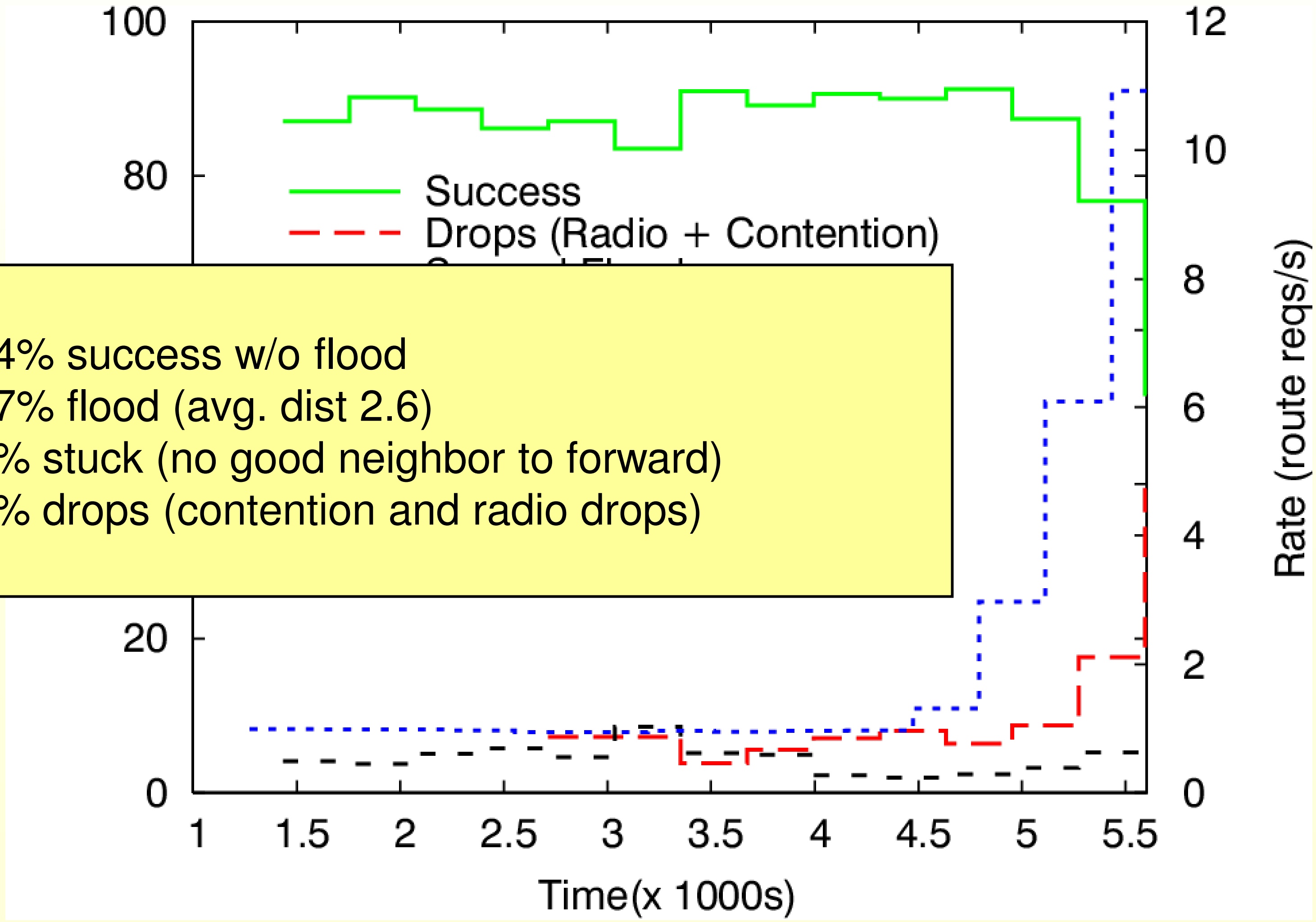
Simulation Results

- Performance similar to that of Geographic Routing (small fraction of floods)
- Small number of beacons needed (<2% of nodes for over 95% of success rate w/o flooding)
- Scope of floods is small
- Resilient to low density and obstacles
- Low stretch

BVR
Implementation Results

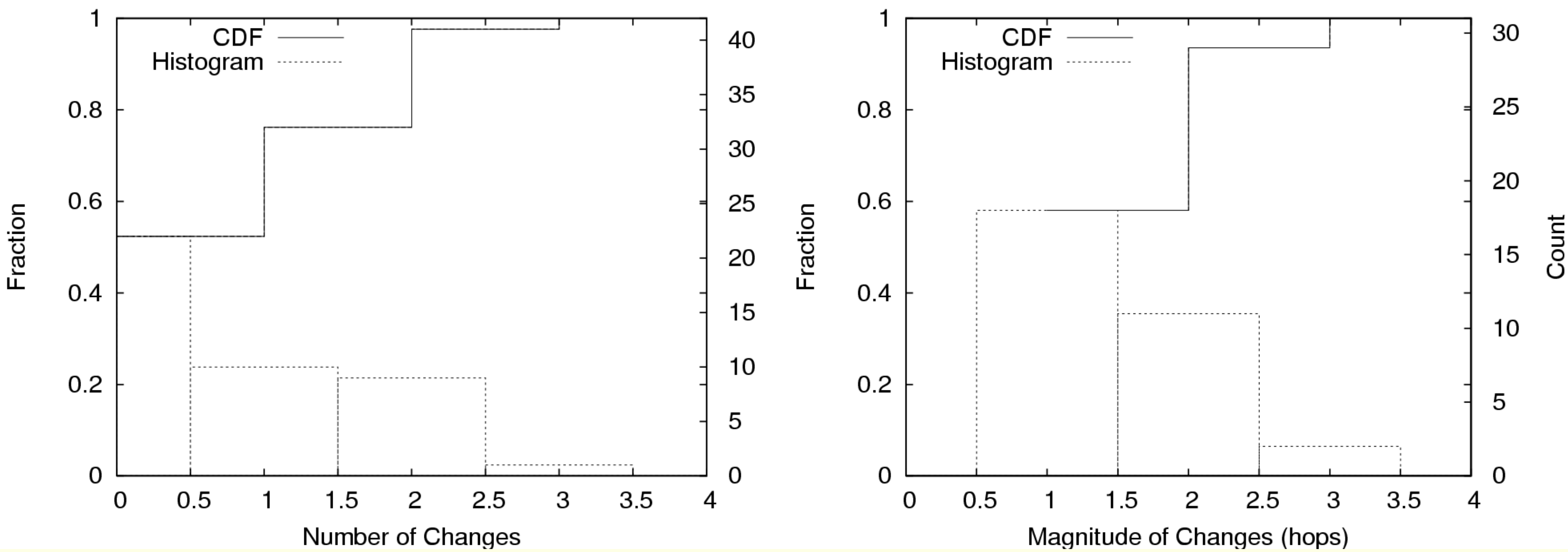
Routing performance

- Soda Testbed, 3100+ random pairs



•88.4% success w/o flood
•4.57% flood (avg. dist 2.6)
•0.5% stuck (no good neighbor to forward)
•6.5% drops (contention and radio drops)

Coordinate stability



- Coordinates were found to be very stable
  - E.g., almost 80% of the nodes had 2 or fewer changes, and over 90% of the changes were smaller than 3 hops

Implementation Results

- Success rates and flood scopes agree with simulation
- Sustained high throughput (in comparison to the network capacity)
- Coordinates were found to be stable
  - Few changes observed, small changes

BVR Lessons and Future Work

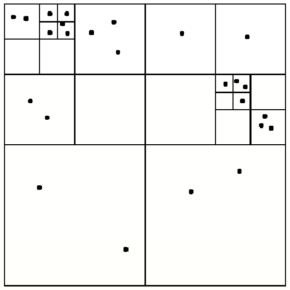
- BVR is simple, robust to node failures, scalable, and presents efficient routes
- Using connectivity for deriving routes is good for low density/obstacles
- The implementation results indicate that it can work in real settings
- We still need to
  - Better study how performance is linked to radio stability, and to high churn rates
  - Implement applications on top of BVR

Structure Discovery, Naming and Routing Based on Connectivity Information Only, No Global Embedding

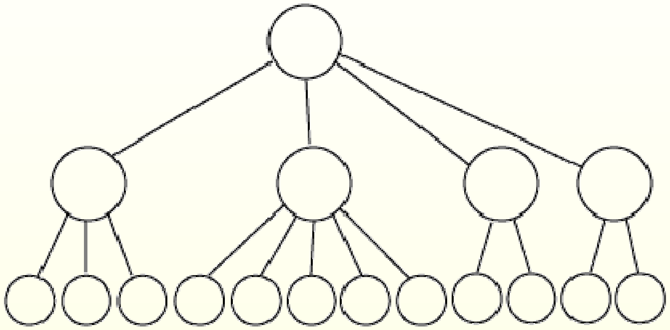
- III. A hierarchical approach using the Discrete Center Hierarchy (DCH)

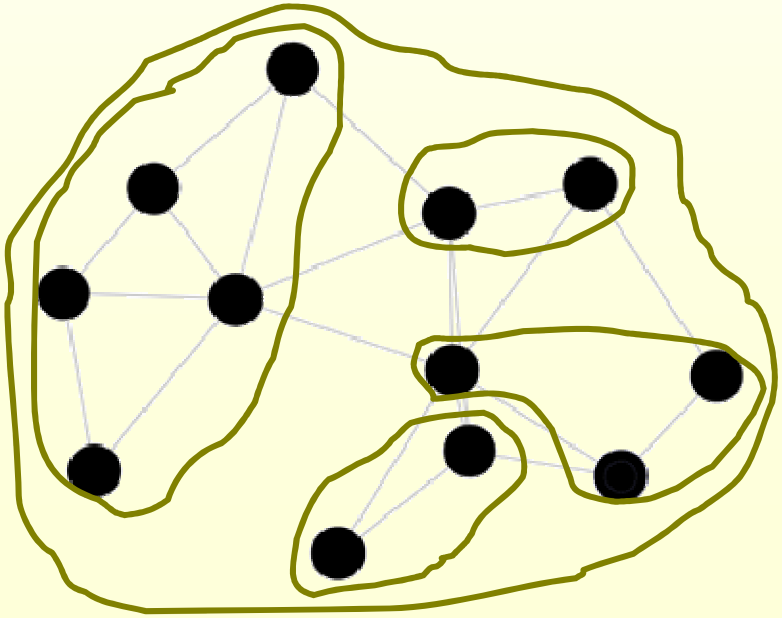
[Funke, G., Nguyen, Wang, DCOSS ’06]

Naming and Routing via Hierarchical Decompositions of Graphs



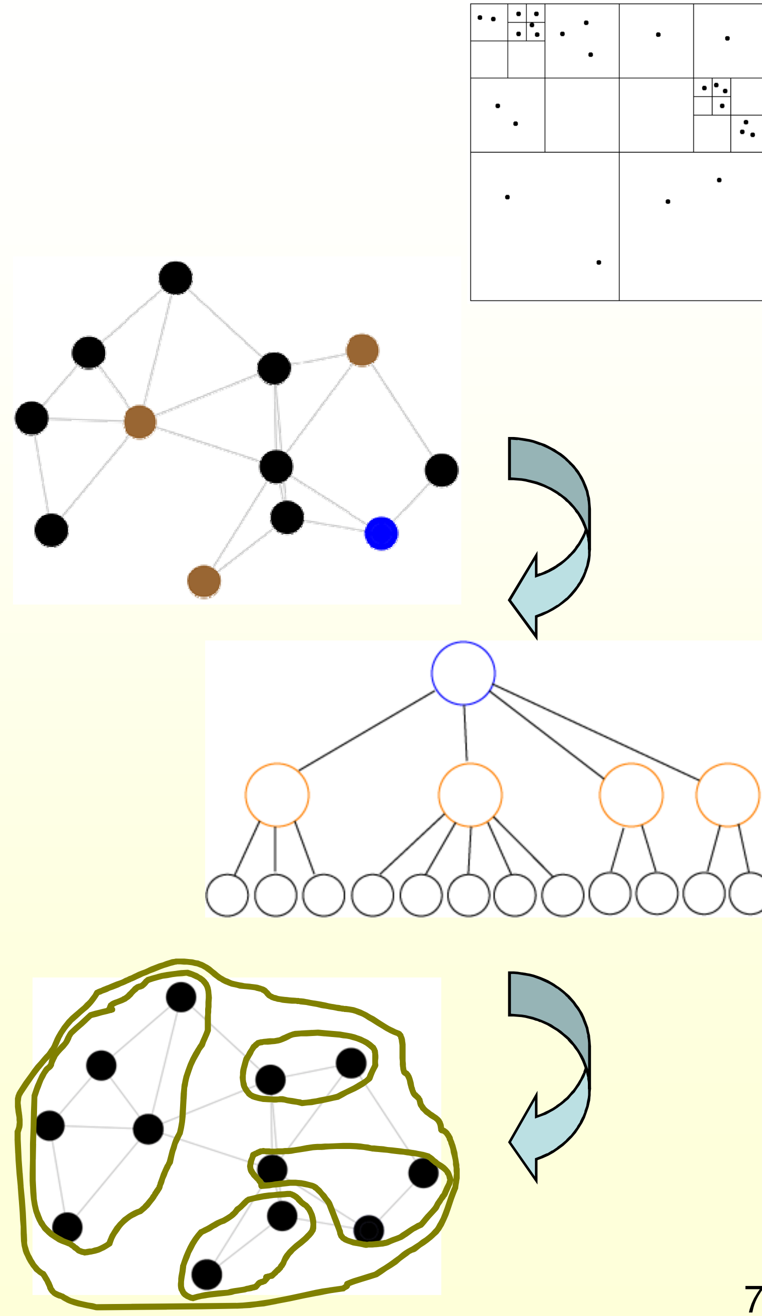
- Generalize quadtrees
- No geographic coordinates assumed for nodes
- Required properties:
  - Clusters in level i of the decomposition have diameter at most $\alpha \cdot 2^i$, where α is a constant
  - Each cluster in level i+1 contains a small (constant) number of clusters in level i





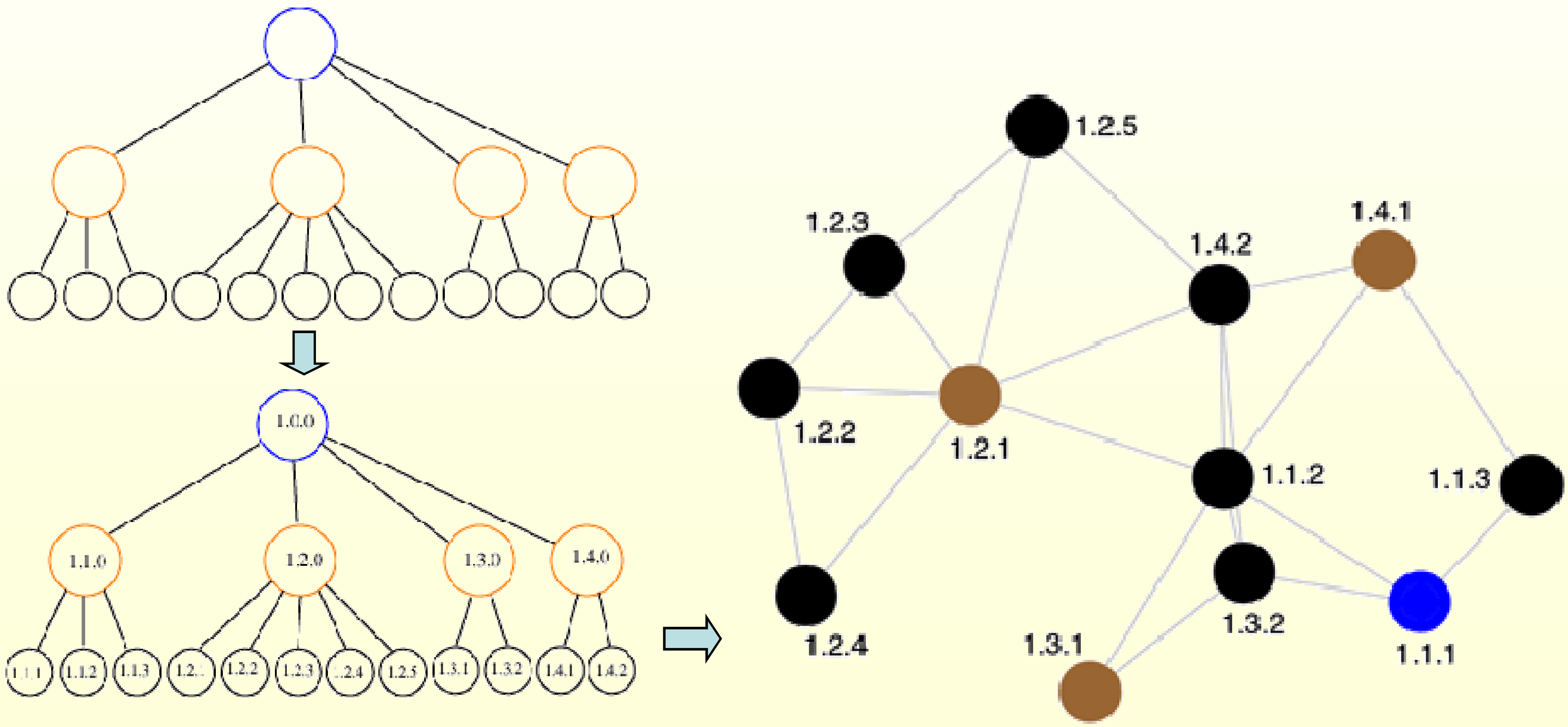
Examples of HDs

- A quad-tree induces a HD when the sensor field is dense and node coordinates are available.
- Discrete Center Hierarchy:
  - A hierarchical sampling of the nodes so that:
    - Nodes in level *i* are at least 2^i hops apart
    - Each node in level *i* is within 2^{i+1} hops from some node in level $i+1$



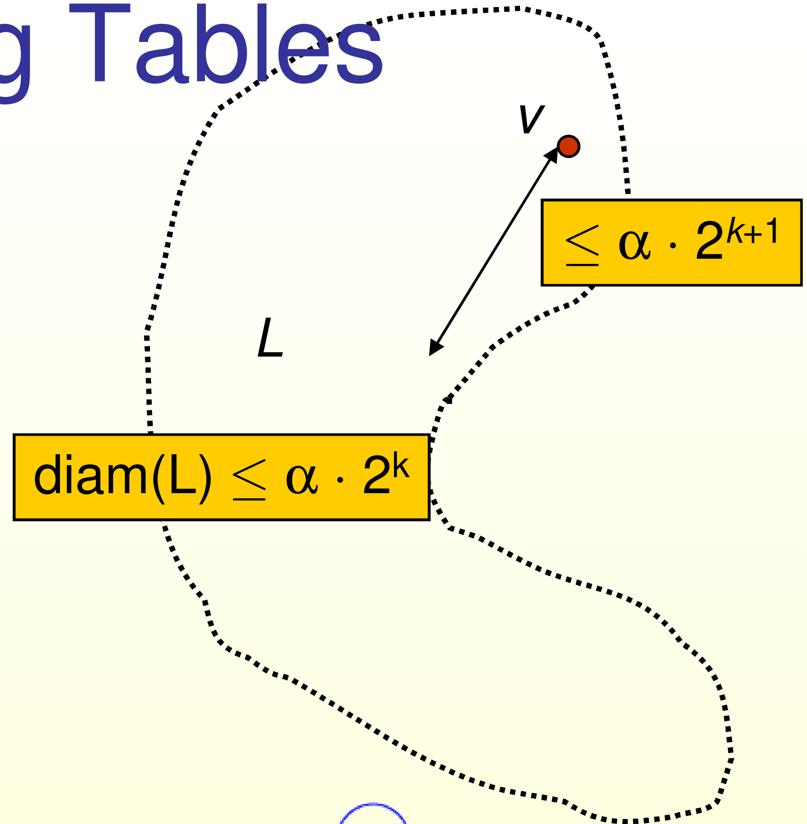
Addressing Scheme

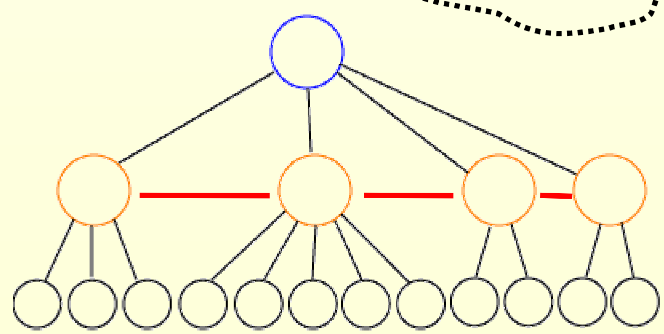
- A HD yields an IP-type addressing scheme for nodes
- Clusters are also assigned addresses



Neighboring Clusters and Local Routing Tables

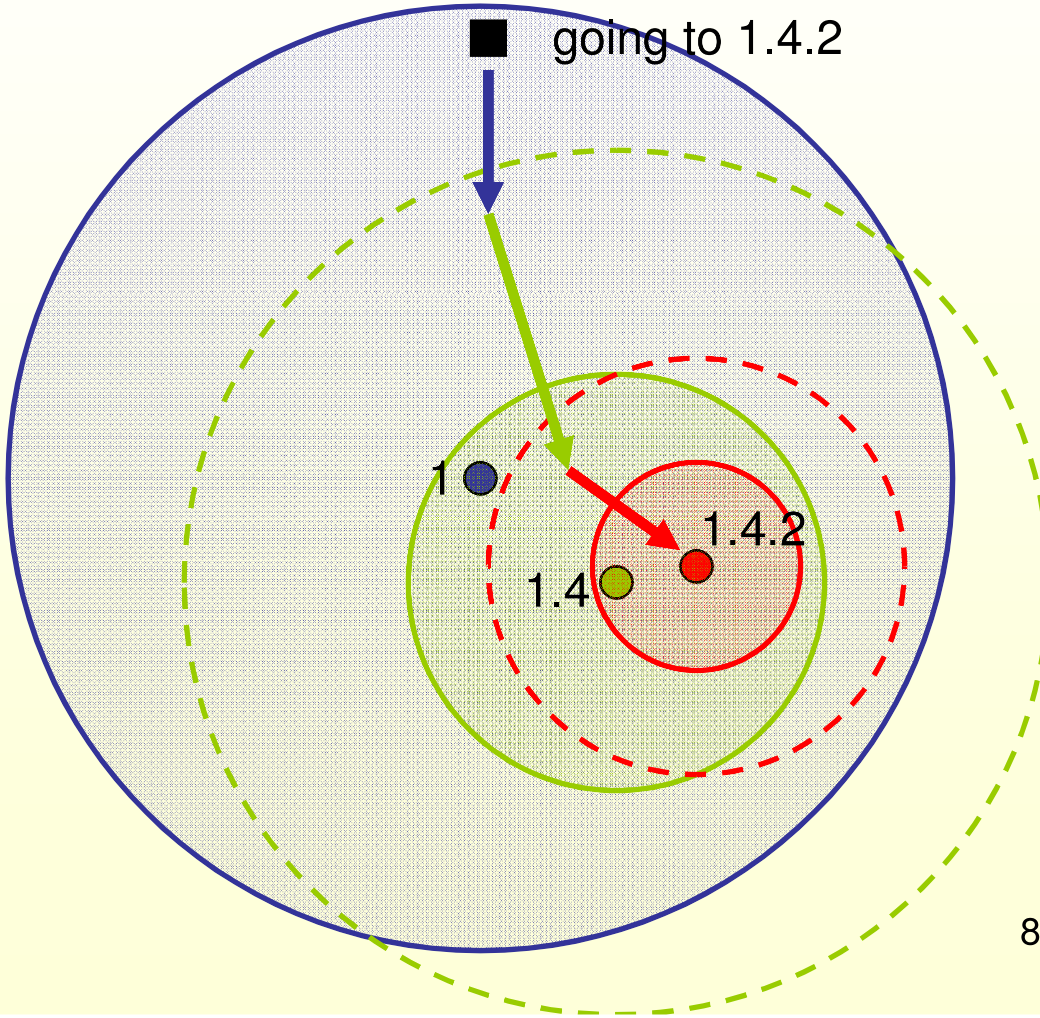
- Def: A cluster L at level k is a neighboring cluster of a node v if $\text{dist}(v, L) \leq \alpha \cdot 2^{k+1}$
- A routing table is stored at each node, providing hop distances to all its neighboring clusters
- Under mild assumptions, each node has $O(\log n)$ neighboring clusters





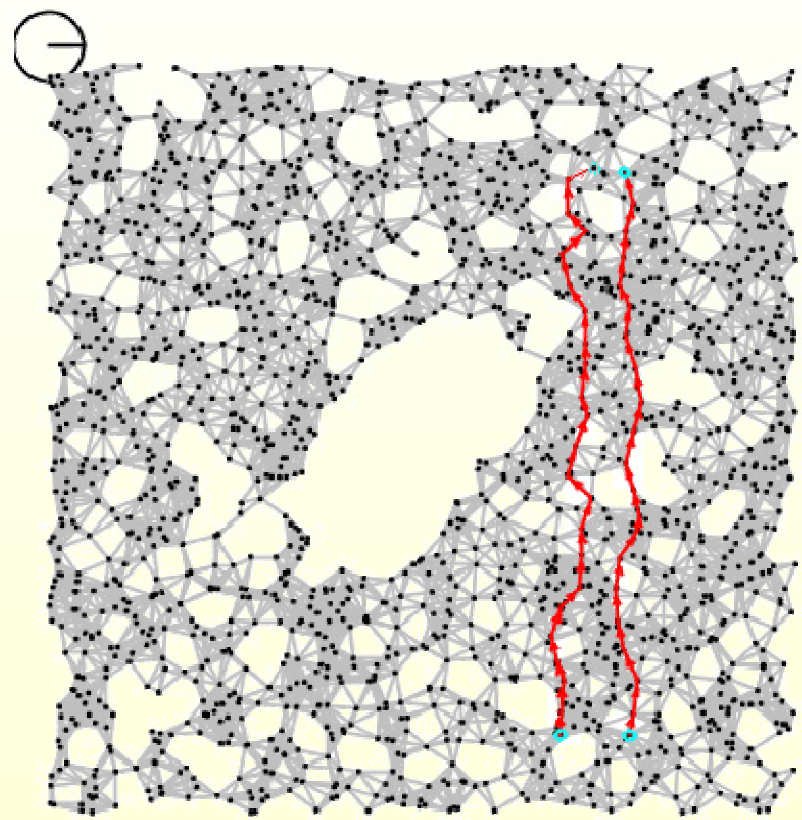
Getting to Your Destination

- Head in the direction of the cluster with the longest prefix that agrees with your destination address
- Use local routing tables to make the best local decision



Routing Scheme

- Routing quality:
  - By proof:
    - Efficient: $|\text{path}(u,v)| \leq 4 \cdot |d_{uv}|$
  - By simulation:
    - Balanced: Nodes high up in the hierarchy do not get overloaded
    - Robust: the failure of any given link does not affect many paths

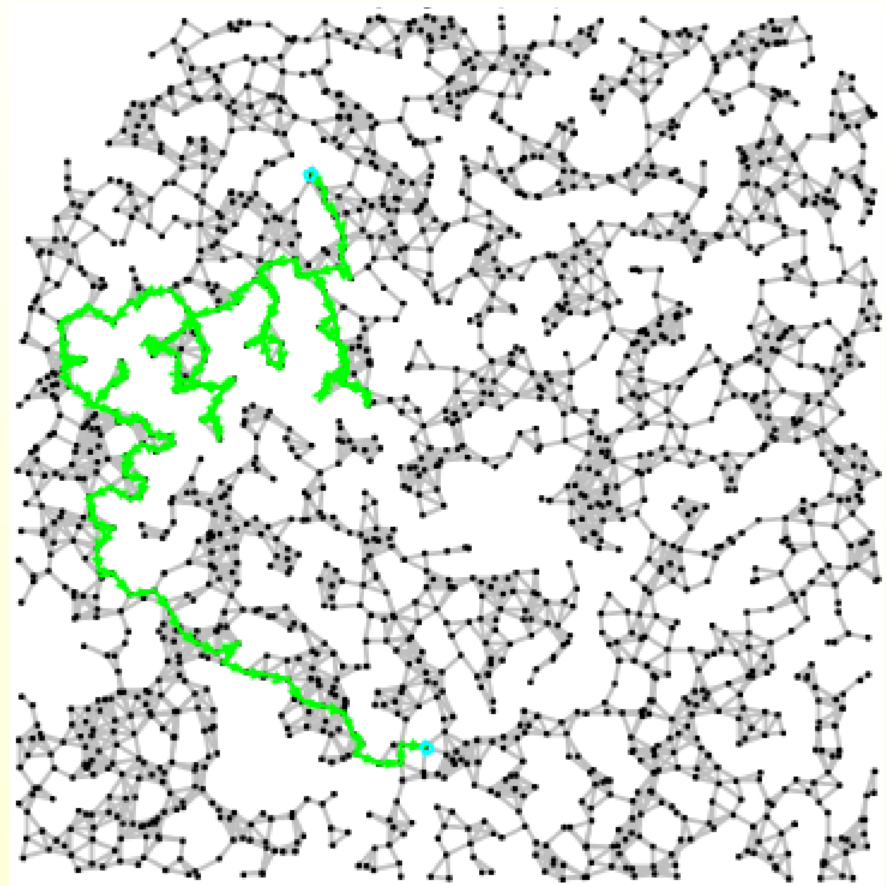


Experimental Results

degree 6.21

Routing quality (2000 nodes)

Avg. deg.	Qual. of HD avg	Qual. of HD std	Qual. of GPSR avg	Qual. of GPSR std
6.21	1.08	0.18	4.91	6.79
6.80	1.06	0.11	4.04	7.41
7.39	1.05	0.09	3.25	7.02
7.93	1.09	0.16	2.04	3.10
8.53	1.07	0.12	1.59	2.22
8.94	1.07	0.10	1.51	2.42
9.82	1.07	0.10	1.35	1.62
10.2	1.06	0.09	1.31	1.80
10.8	1.05	0.09	1.44	2.90
12.0	1.06	0.11	1.15	1.30



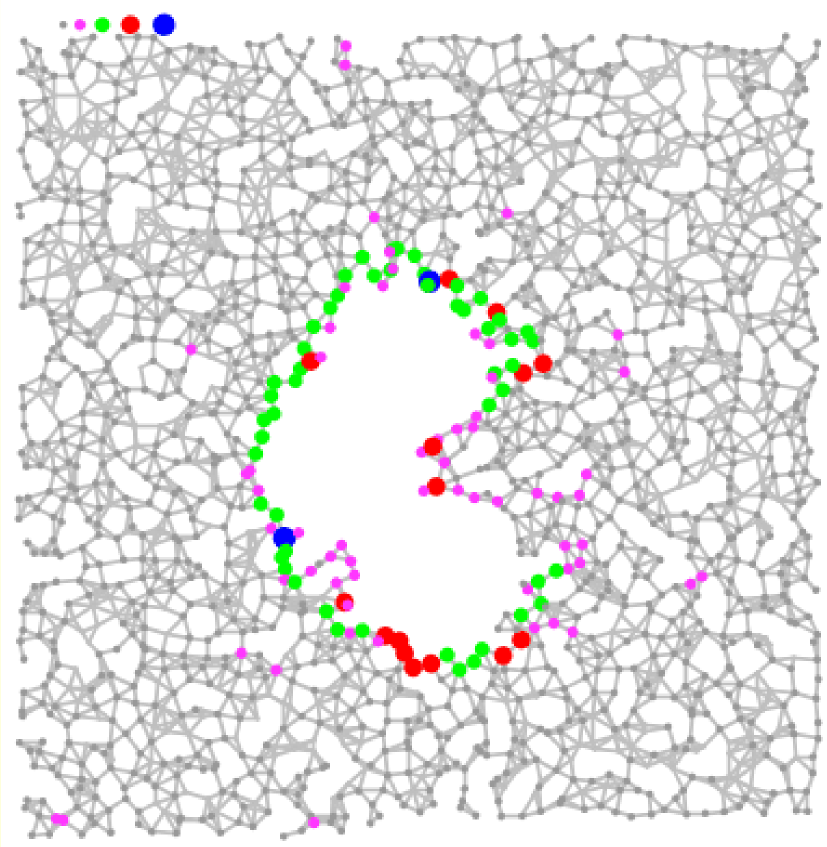
Paths generated are near optimal

$$\text{Quality} = \frac{\text{HD path length}}{\text{Shortest path length}}$$

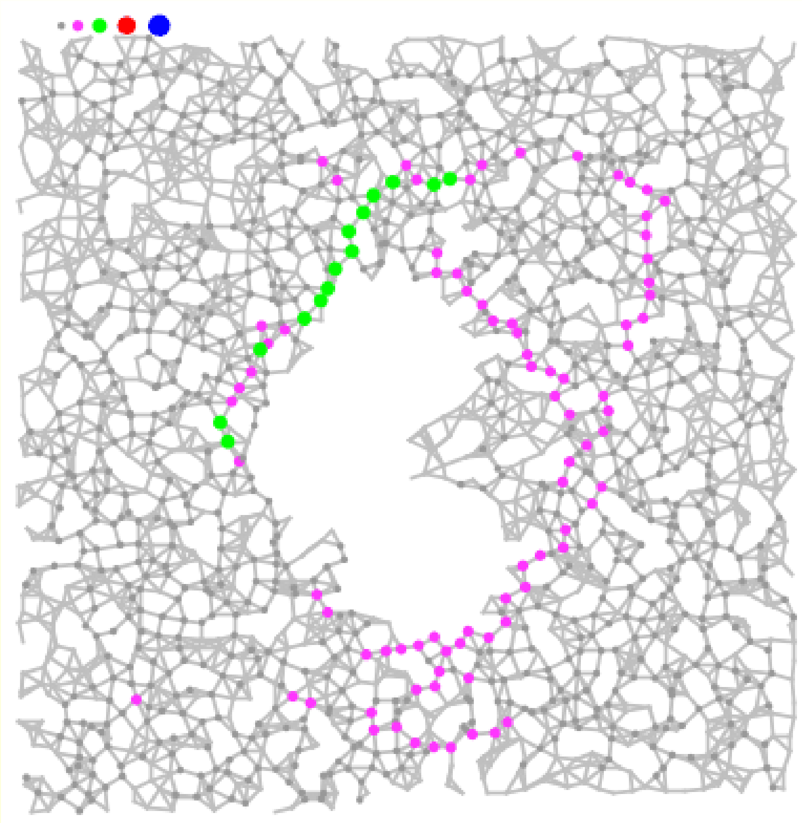
HD path – Shortest path
GPSR path

Hot Spots

2000 nodes, perturbed grid
100 random paths
max load = 32



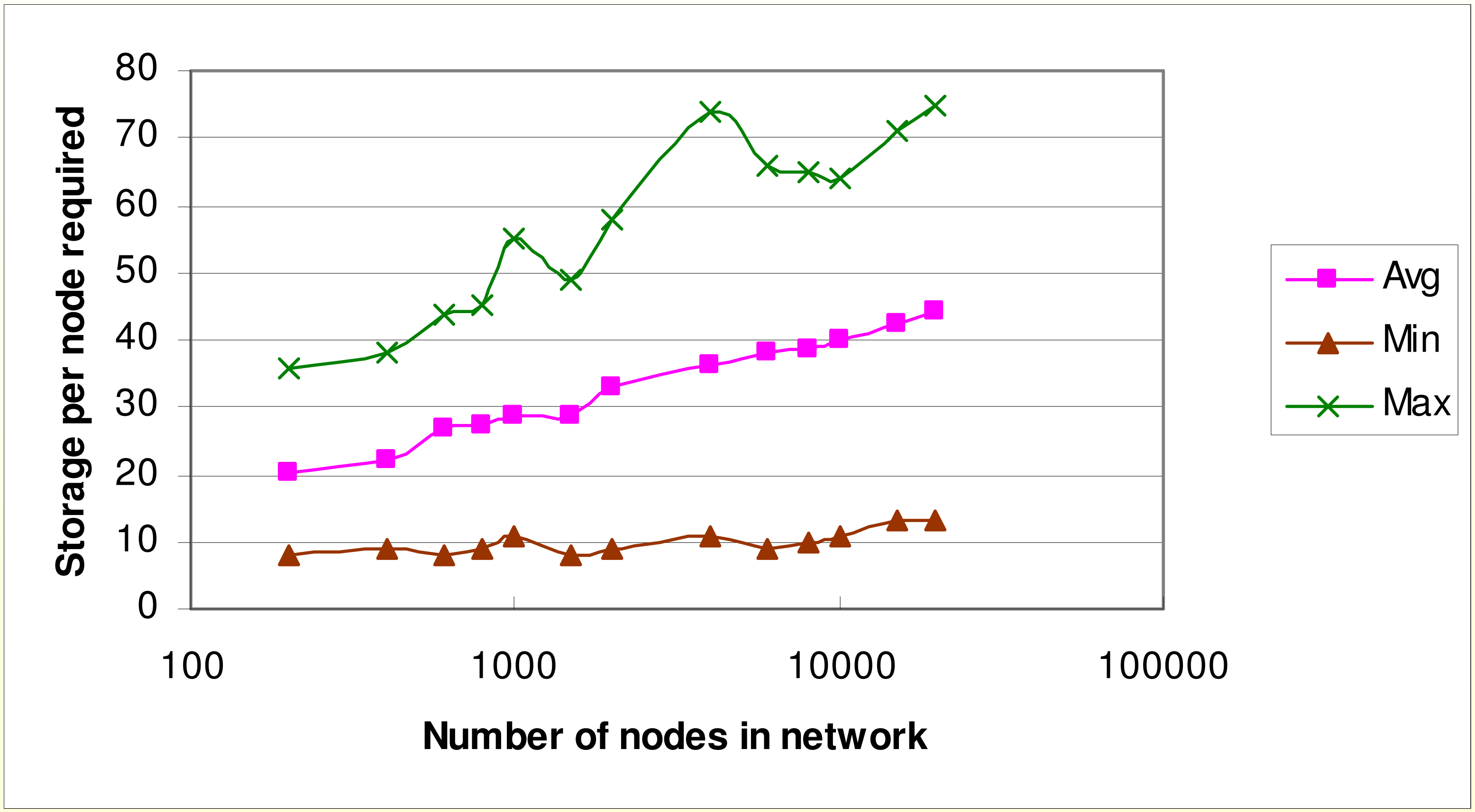
GPSR



HD

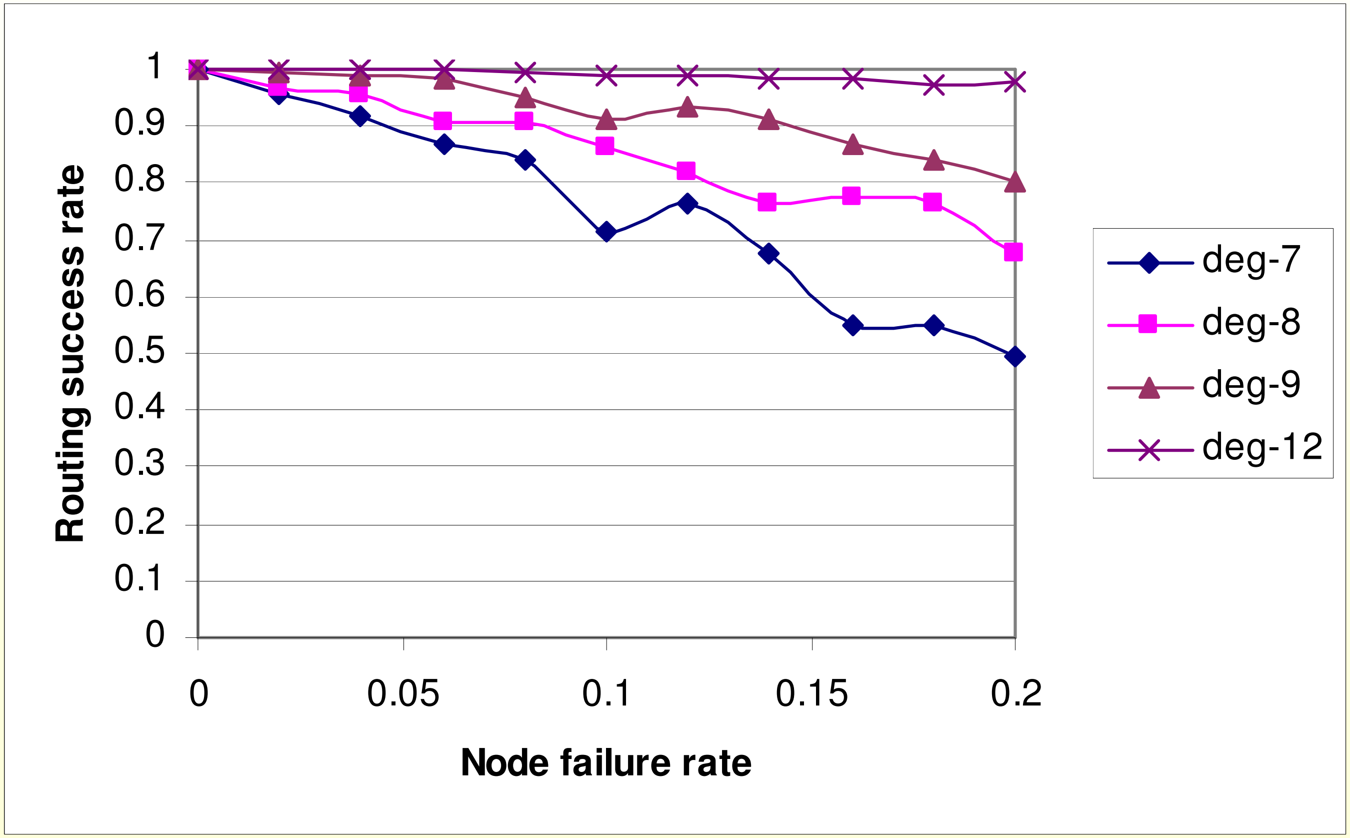
HD does not hug holes as much as GPSR

Routing Scalability



Storage used grows slowly (log)
Network initialization cost ~ storage used

Routing Robustness



Routing performance degrades gracefully
as node failure rate increases

HD Names and Routes Summary

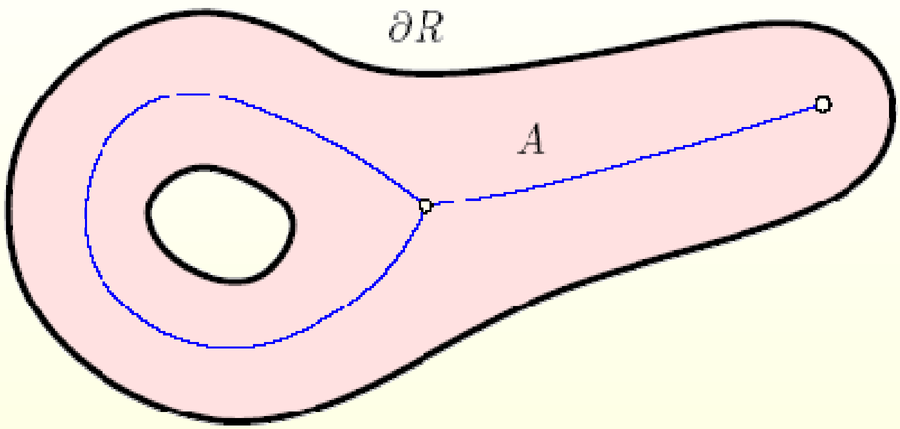
- HD effectively discovers the intrinsic geometry of the network
- Provides a hierarchy-based scheme with provable approximation quality on the routing paths
- Node/link failures affect mostly the low levels of the hierarchy

MAP:
Medial-Axis Based Geometric Routing in Sensor Networks

[Bruck, Gao, Jiang '05]

Medial Axis --- Definitions

Given a bounded region R, the medial axis of its boundary ∂R is the collection of points with two or more closest points in ∂R .

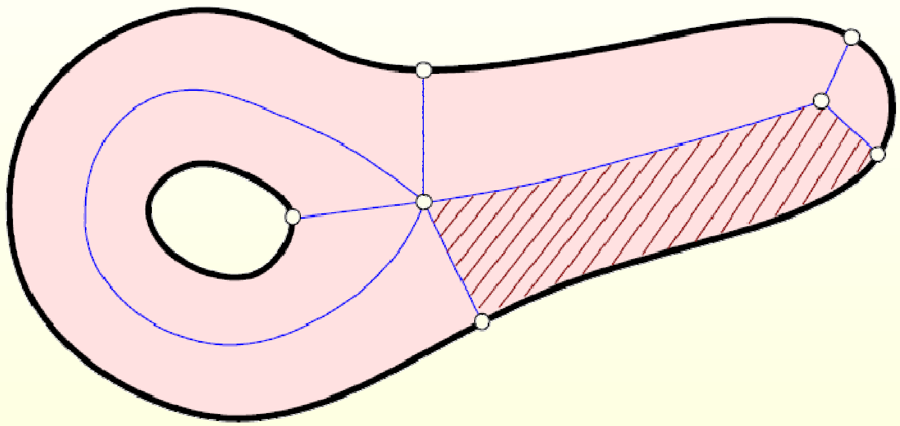


The medial axis of a piecewise analytic curve is a finite number of continuous curves.

Any bounded open subset in $\mathbf{R}^2$ is homotopy equivalent to its medial axis. Thus it has the same topological features of R.

Partitioning into Canonical Regions

A chord is a line segment connecting a point on the medial axis and its closest points on ∂R. A point on the medial axis with 3 or more closest points on ∂R is called a medial vertex.

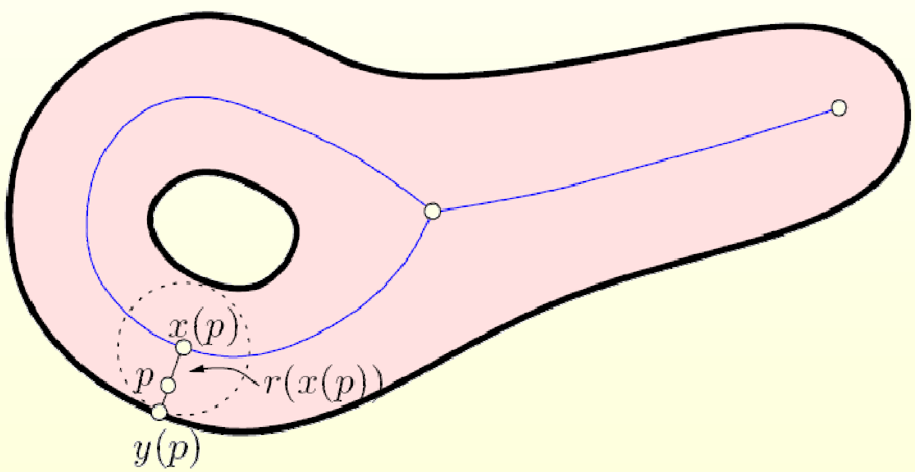


We can partition the region R by the medial axis and the chords of medial vertices into canonical pieces, each resembling a stretched rectangular region.

Naming w.r.t. Medial Axis

A point p is named by the chord x(p)y(p) it stays on. (x(p), y(p), d(p))

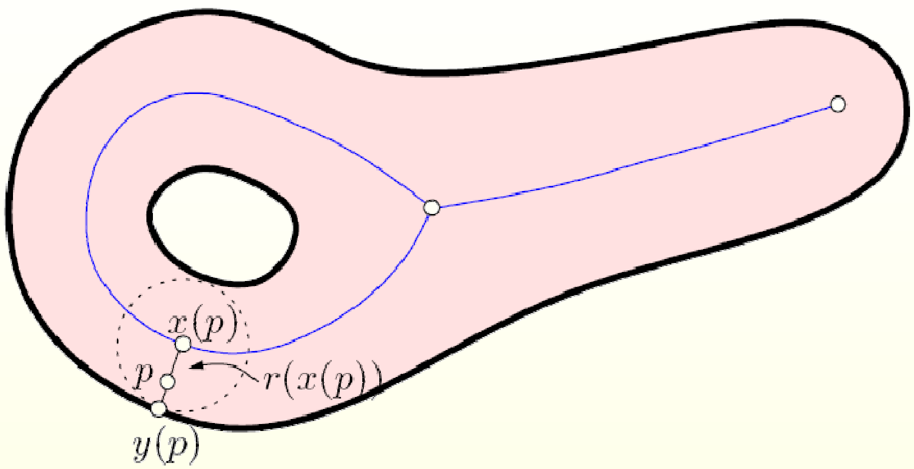
x(p) is a point on the medial axis.
y(p) is the closest point of x(p) on ∂R.
d(p) is height, i.e., relative distance from x(p): |px(p)|/|x(p)y(p)|.

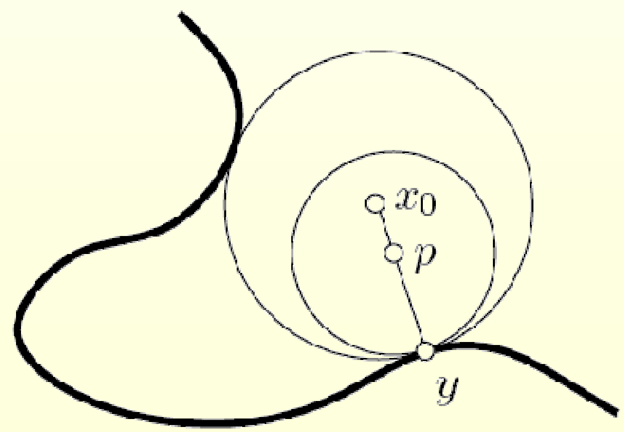


Theorem: each point is given a unique name.

Naming is Unique

- Lemma 1: for a point p not on the medial axis, if p is on a chord xy, then y is p's only closest point on ∂***R***.
- Say y' is p's closest point, then |xy'| ≤ |xp|+|py'| < |xy|.
- Lemma 2: If p is not on the medial axis, there is a unique chord through p.

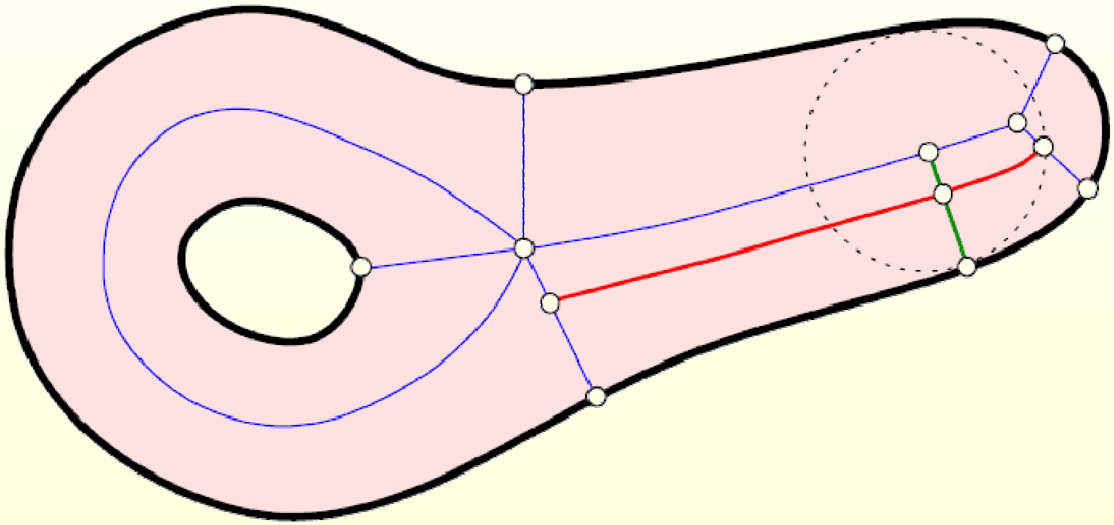




Routing Inside a Canonical Tile

The naming system naturally builds a Cartesian coordinate system:

x-longitude curve --- the chord attached to point x on the medial axis
h-latitude curve --- the points with the same height h.

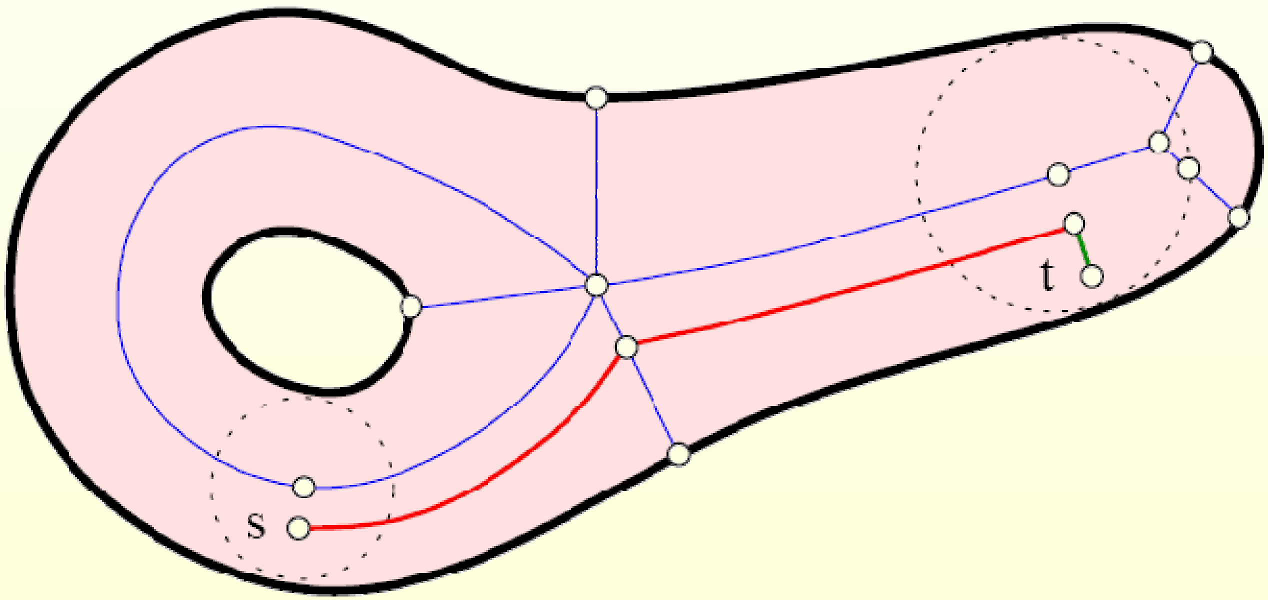


Inside a canonical piece, we just do Manhattan routing!

Routing Between Canonical Tiles

The canonical pieces are glued together by the medial axis.

With the knowledge of the medial axis – we can route from pieces to pieces by checking only local neighbor information.

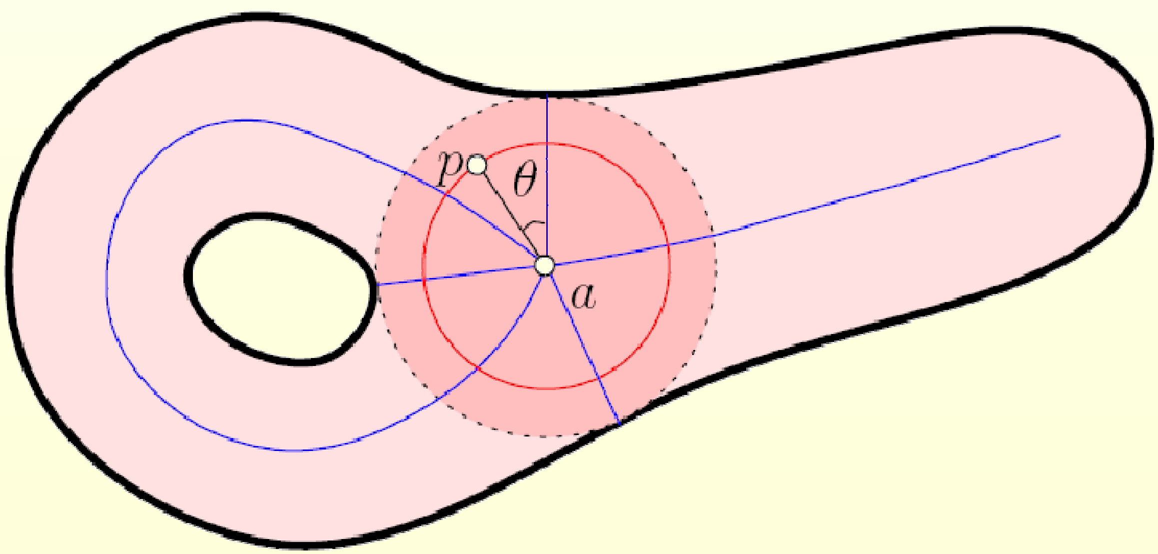


Routing Between Canonical Tiles

Two canonical pieces adjacent to the same medial vertex may not share a chord.

A fix: build rotary systems around medial vertices.

Polar coordinate system: (|ap|/r, θ), r is the maximum radius of a empty ball centered at a medial vertex a.

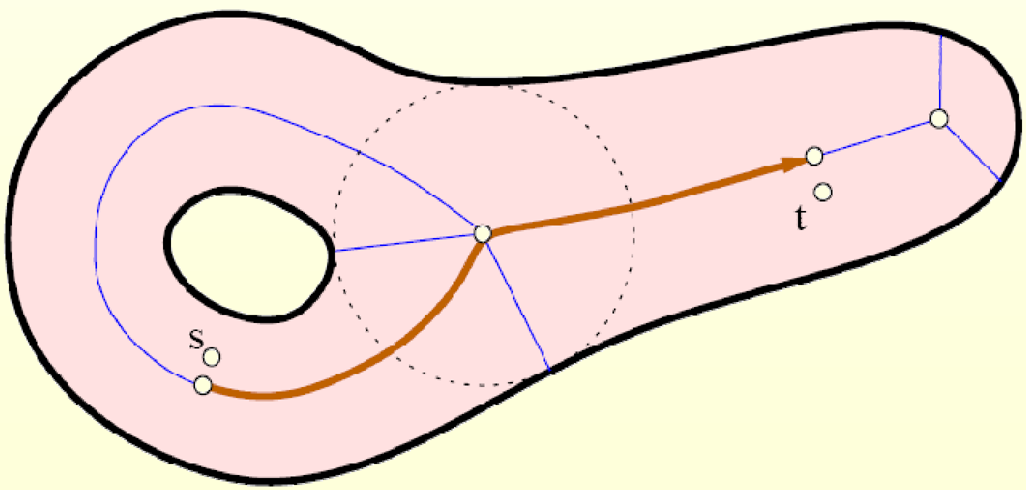


Routing Between Canonical Tiles

Routing is done in 2 steps:

1. Check the medial axis graph, find a route connecting the corresponding points on the medial axis as guidance.

2. Realize the route by local gradient descending, in either the Cartesian coordinate system inside a canonical piece, or a polar coordinate system around a medial vertex.

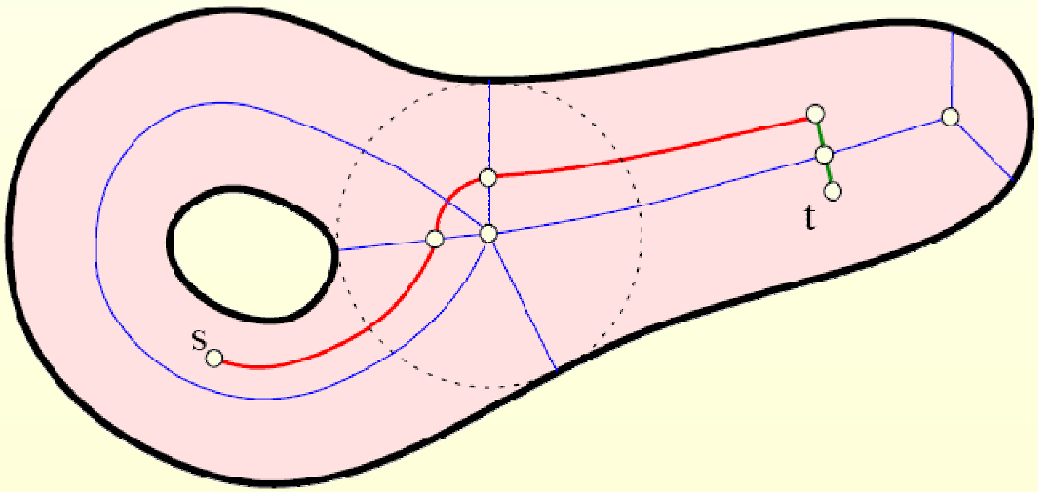


Routing Between Canonical Tiles

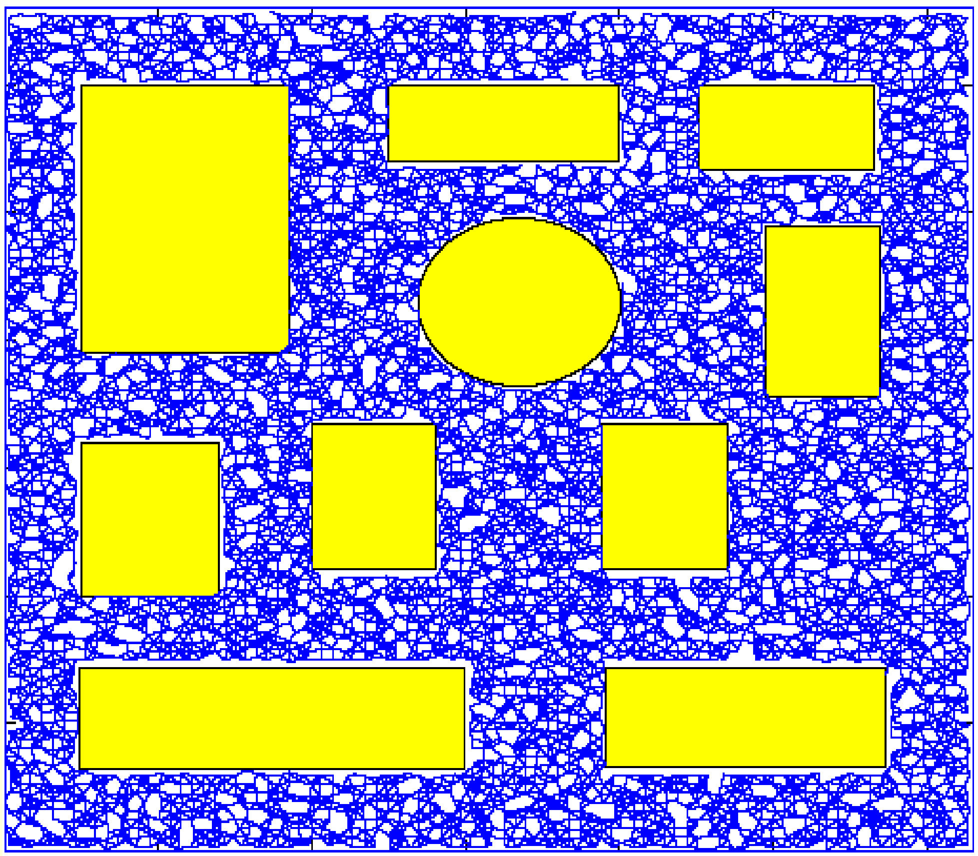
Routing is done in 2 steps:

1. Check the medial axis graph, find a route connecting the corresponding points on the medial axis as guidance.

2. Realize the route by local gradient descending, in either the Cartesian coordinate system inside a canonical piece, or a polar coordinate system around a medial vertex.

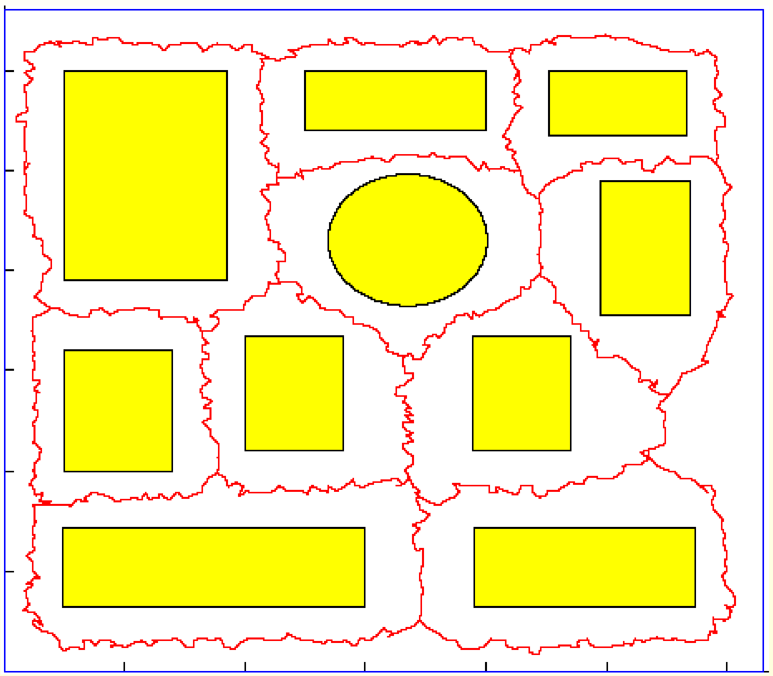


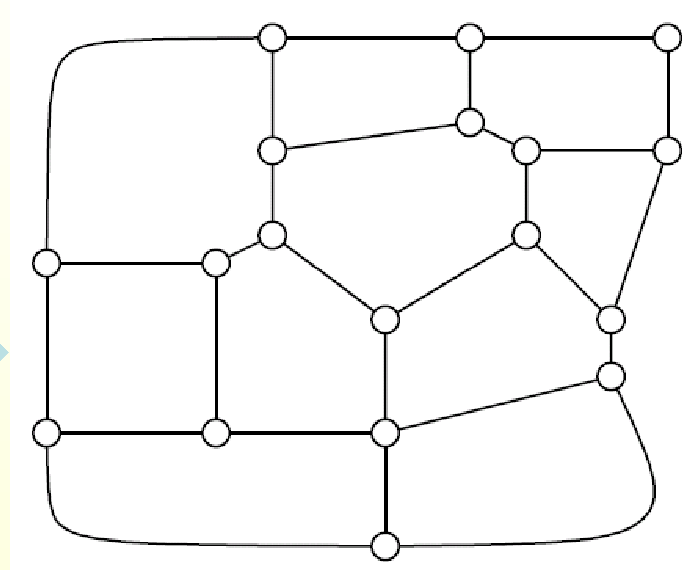
Simulation Examples



5735 nodes

Simulation Examples





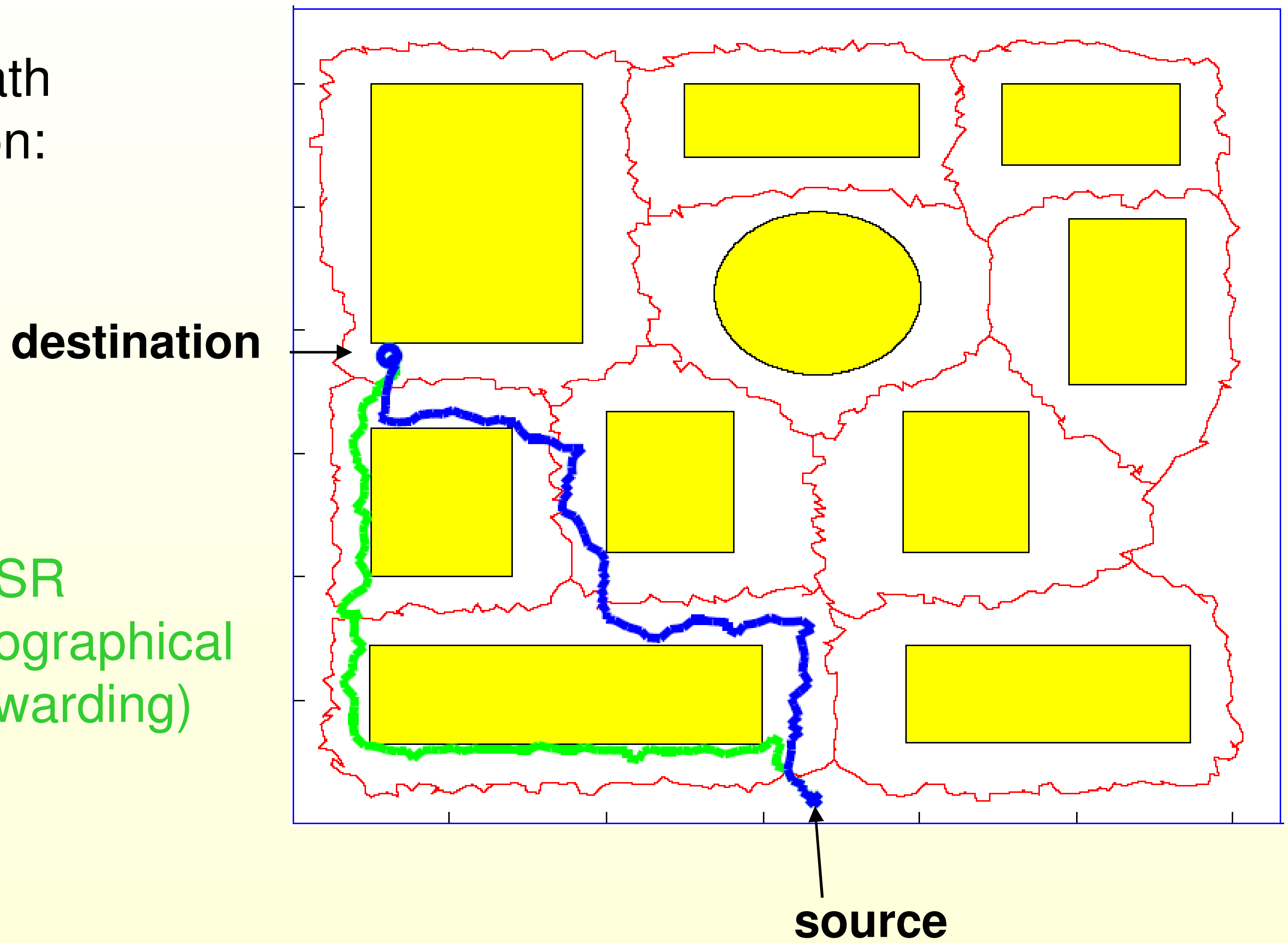
The simple medial axis graph: 18 nodes, 27 edges.

Simulation Examples

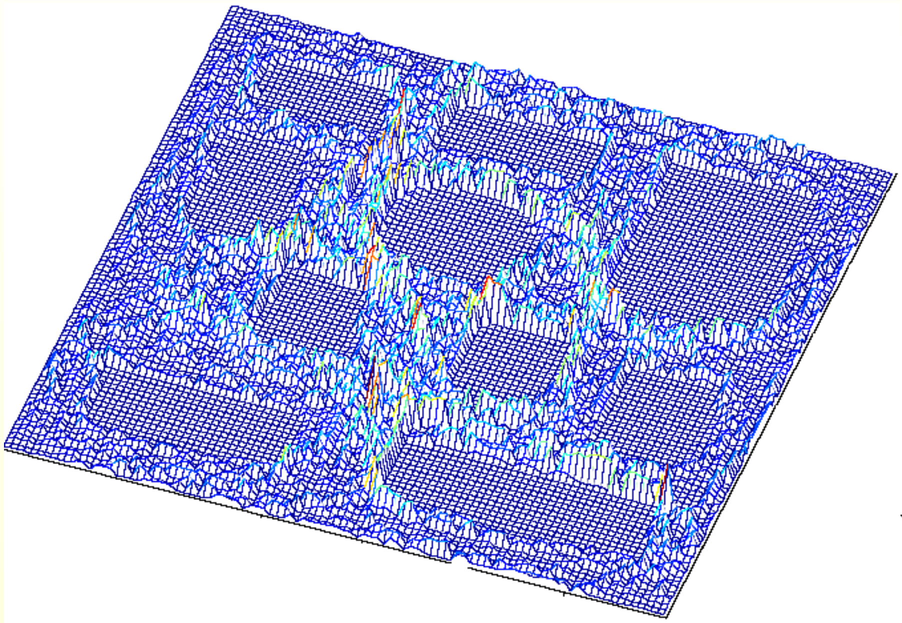
Routing path comparison:

Blue: MAP

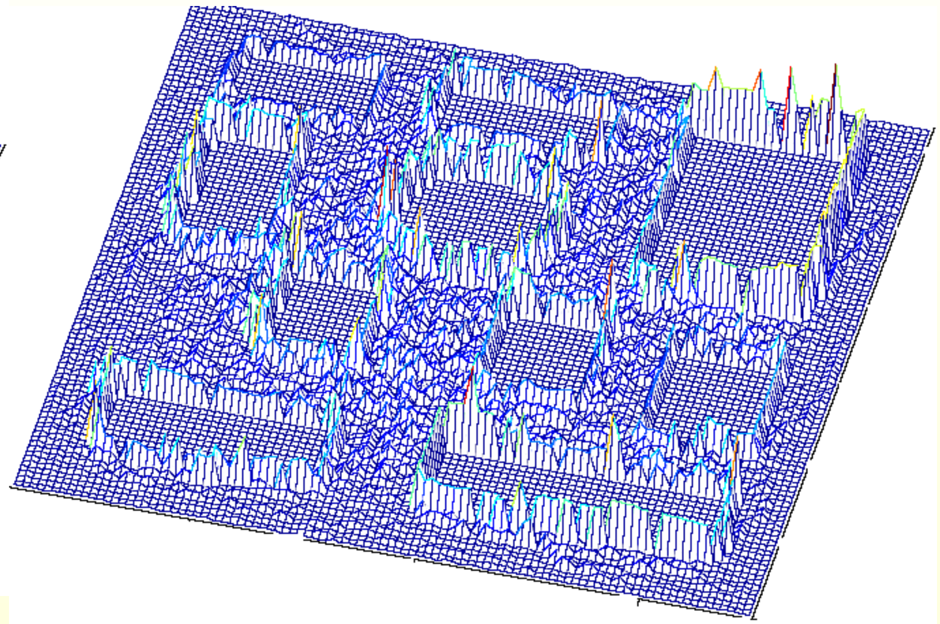
Green: GPSR (geographical forwarding)



Load Balancing



Map



GPSR

Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Quasi unit disk graph model:

- If two nodes are within distance $1-\alpha$, they are connected.
- If two nodes are more than $1+\alpha$ away, they are not connected.
- If the distance of two nodes is between $1-\alpha$ and $1+\alpha$, a link between them exists with probability p.

Note:

- Unit disk graph corresponds to the special case $\alpha = 0$.
- The ratio of the largest and the smallest coverage ranges is $\frac{1+\alpha}{1-\alpha}$.

Simulation Examples

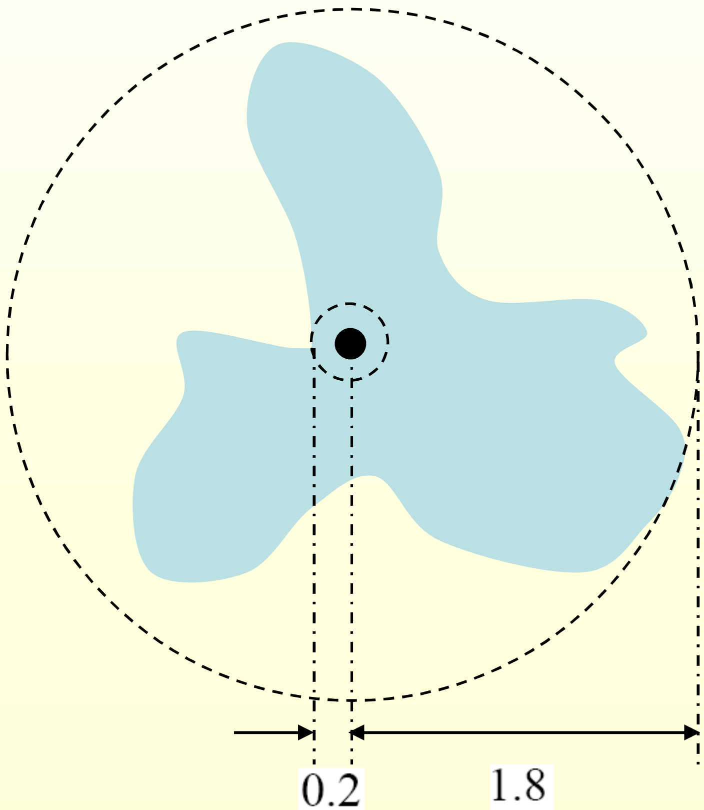
Test MAP on networks modeled by quasi unit disk graphs

Example: $\alpha = 0.8$

Maximum coverage range: 1.8

Minimum coverage range: 0.2

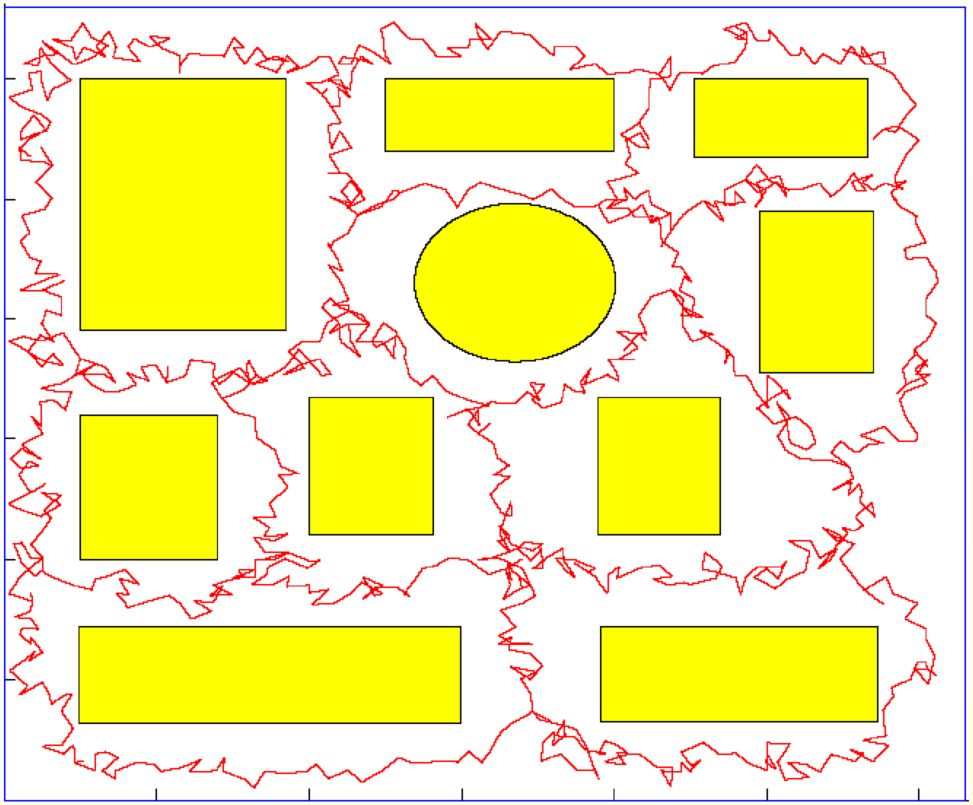
An example coverage
area of a node:



Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

Example: $\alpha = 0.8$



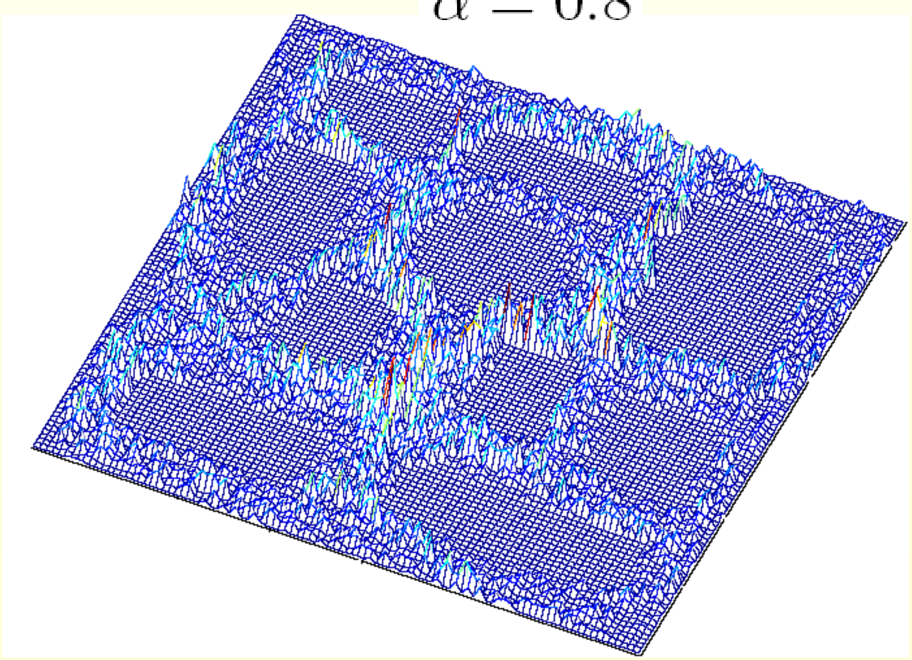
Simulation Examples

Test MAP on networks modeled by quasi unit disk graphs

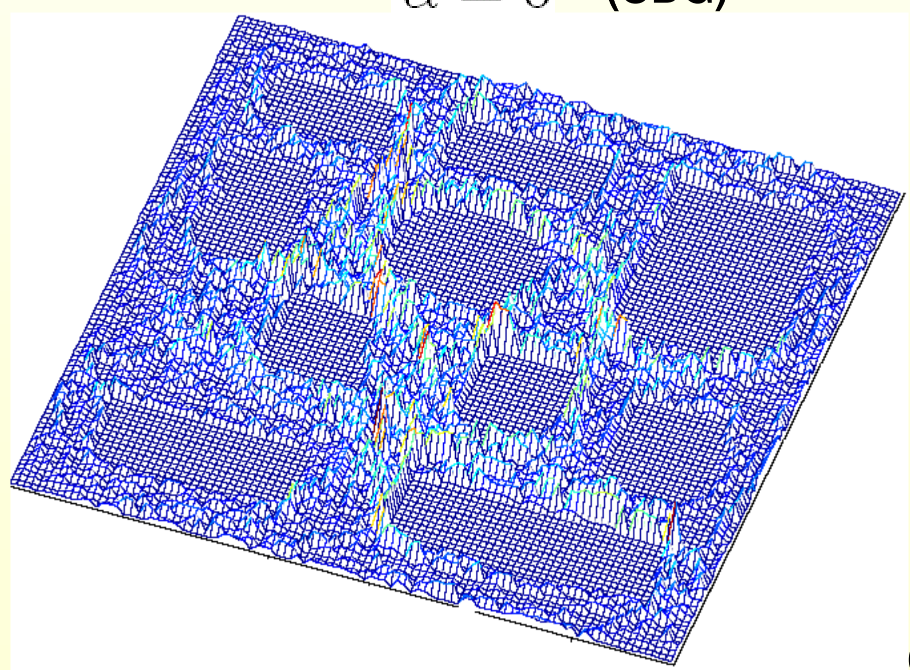
Example: $\alpha = 0.8$

Compare MAP Load: both well balanced

$\alpha = 0.8$

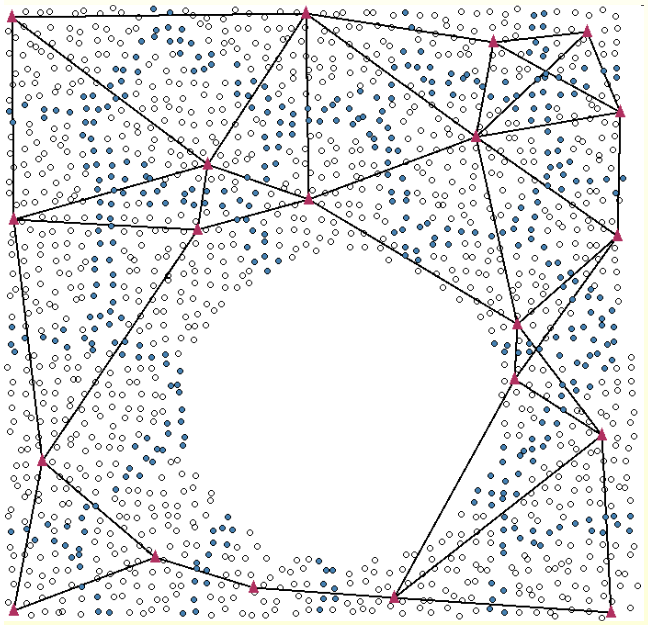


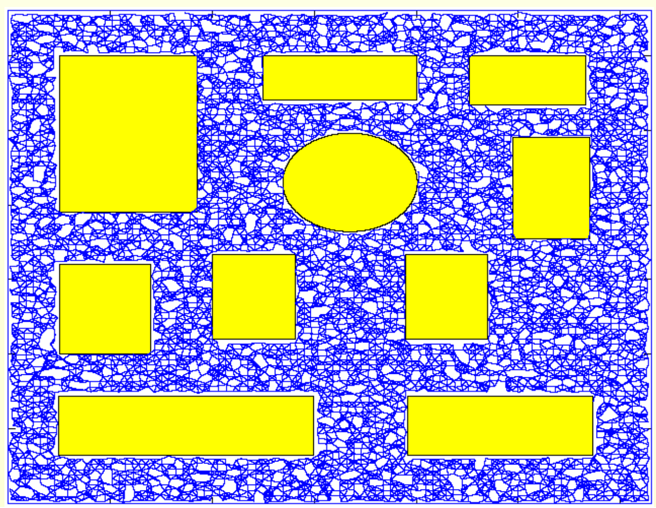
$\alpha = 0$ **(UDG)**

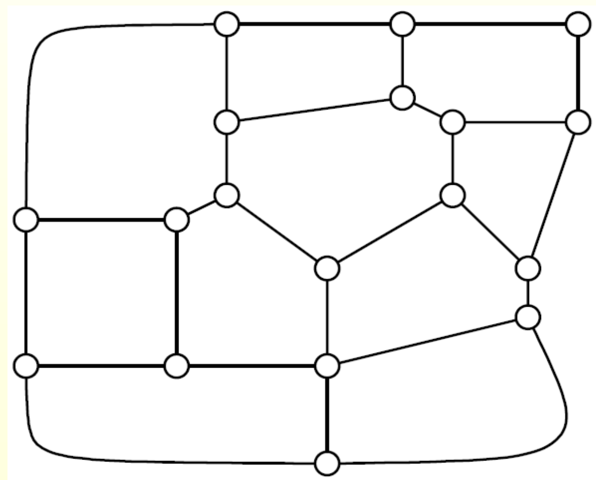


Comparison of GLIDER and MAP

1. Different ways to represent the global topology.
2. More understand of the performance comparison is necessary.
3. GLIDER works also for 3D sensor field, but landmark selection requires more study.







Summary

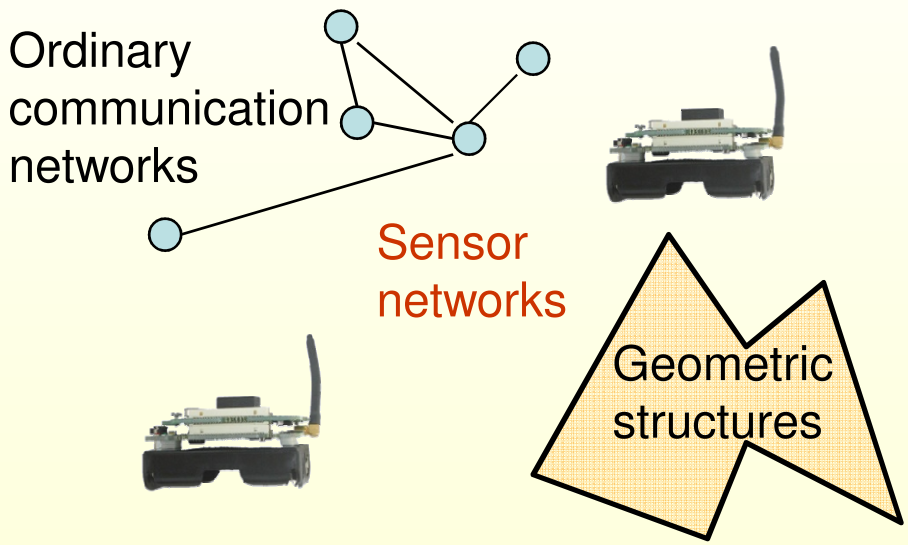
- Topology-enabled naming and routing schemes
- Separate the global topology and the local connectivity
  - Use topological information to build a routing infrastructure
  - Propose a new coordinate system for a node based on its hop distances to a subset of landmarks
- Advantages
  - No location info.
  - No unit disk graph assumption.
  - Local routing.

Lessons

- Structure discovery and information discovery are fundamental problems for WSNs
- With light preprocessing, we can compute certain local quantities that capture global aspects of the network which can significantly help with local decisions
- These quantities reflect an understanding of the geometry or topology of the sensor field, or of its signal landscape, and do not require localization
- The same quantities are also robust to local volatility in the network connectivity

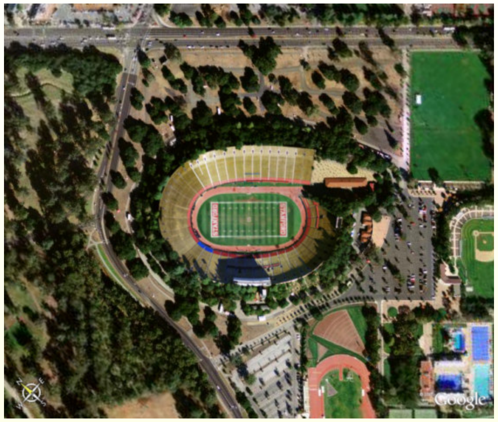
A Dilemma: Living Between Two Worlds

- 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?

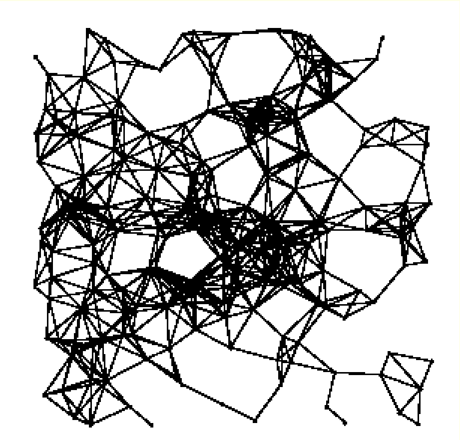


- What if the network is volatile?

Which Reality to Believe?



Continuous physical world,



or discrete network world,

or,
invented appropriate new continua —
— as the ancients did when looking at constellations ...

The End