# MAP: Medial Axis Based Geometric Routing in Sensor Networks

Jehoshua Bruck Jie Gao Anxiao (Andrew) Jiang

*California Institute of Technology*

# Point to Point Routing in SensorNet

- *Point to Point routing is useful for:*

  Sensor tasking and control

  Content-based data storage and retrieval

  Target tracking and detection

- *Goal: load balance, light-weight, localized routing.*

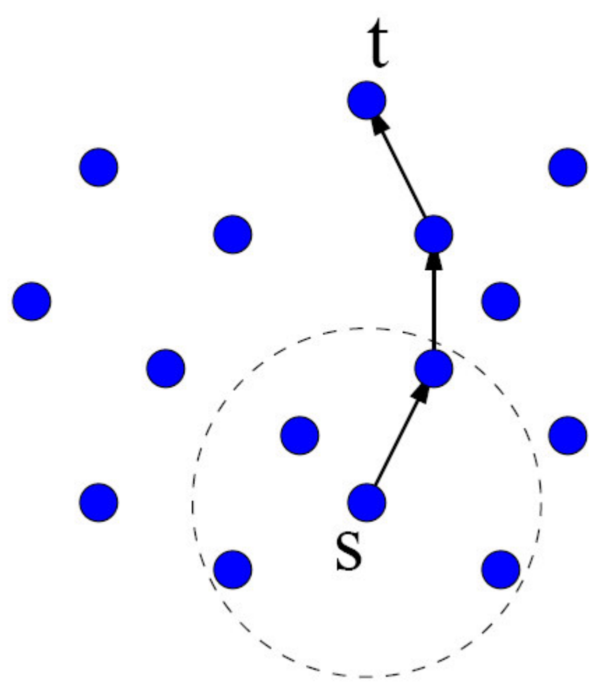
# Using Geometric Information for Routing

- *Sensor networks are closely related to the geometric environment where they are deployed.*

- *Geometric information helps!*

# Using Geometric Information for Routing

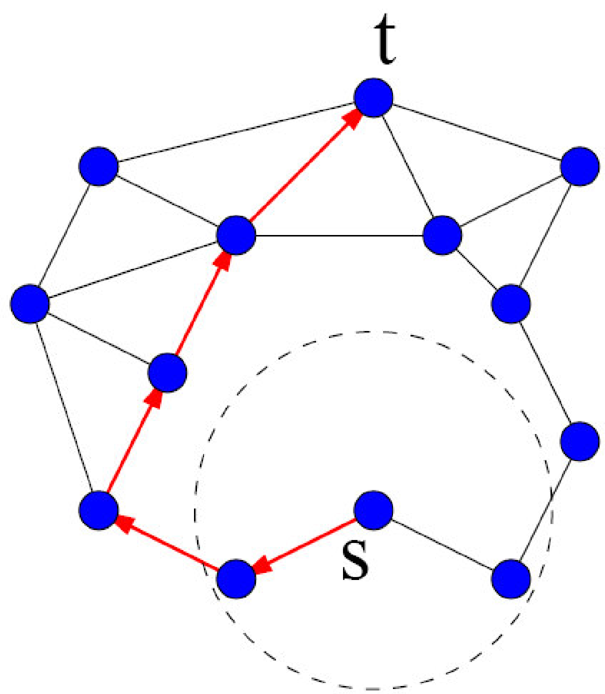
- *Geographical Forwarding:*
  - Significantly simplifies the routing protocol, low routing overhead.
  - Good for uniform and dense sensor deployment in a flat and regular region.



# Using Geometric Information for Routing

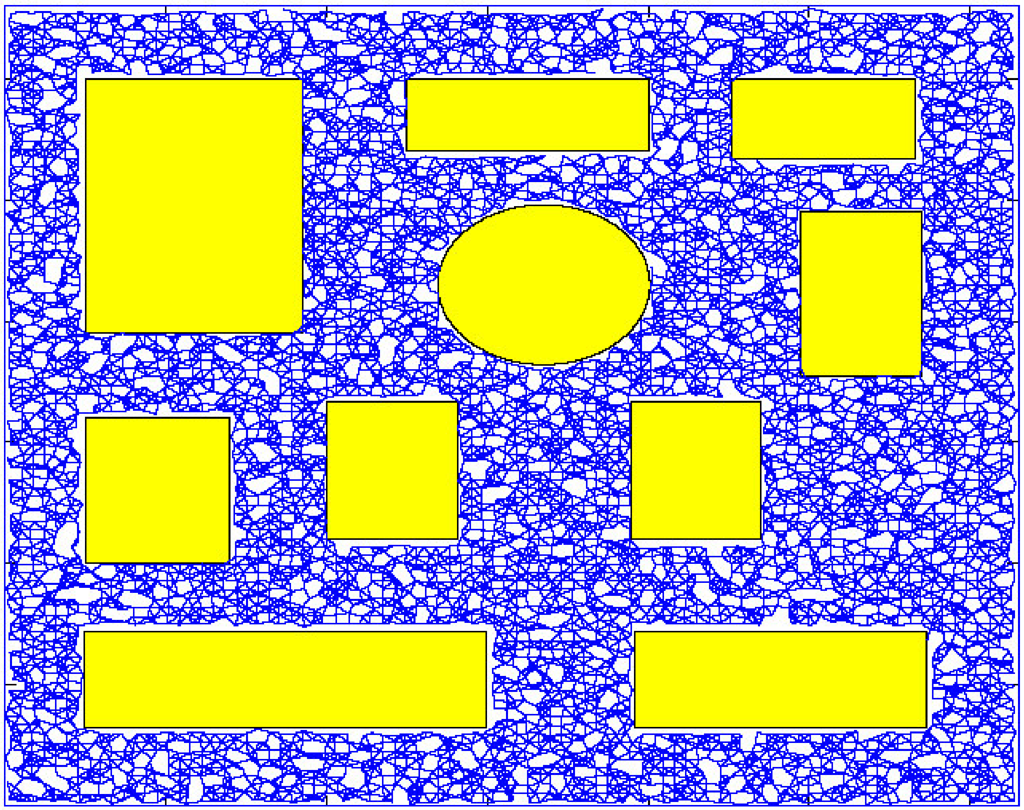
- *Geographical Forwarding may get stuck at local minima.*

  *Additional mechanisms to cope with dead-ends: Face Routing, Perimeter Routing, etc.* [Bose, et.al 01][Karp, Kung 00][Kuhn et.al.03]



# Face routing creates unbalanced loads

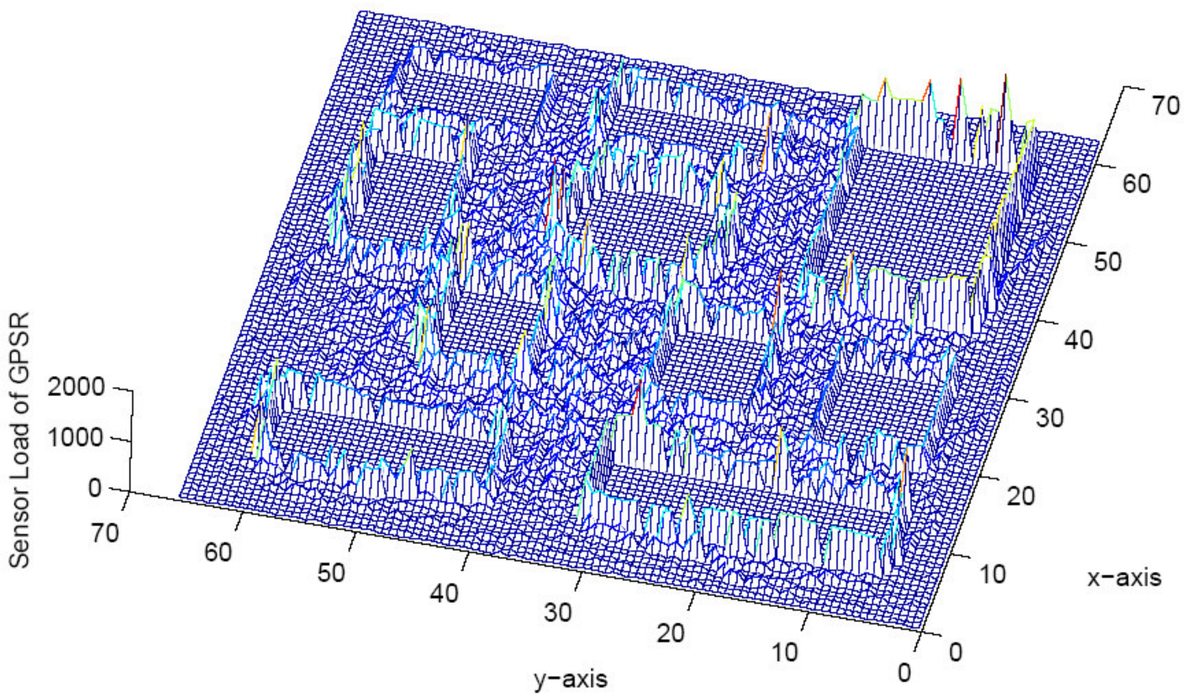
- *Poor performance in sensor fields with complex geometry: Nodes on the boundary are heavily loaded.*



A campus with buildings.

# Face routing creates unbalanced loads

- *Poor performance in sensor fields with complex geometry: Nodes on the boundary are heavily loaded.*




Distribution of traffic load for 12000 random source and destinations.

# Additional cost of geographical routing

- *Geographical routing comes with the price of localization.*

  Accurate location information is either expensive or hard to obtain.

- *Subtraction of a planar subgraph fails when*

  Location information is inaccurate;

  Unit disk graph assumption fails.

# Routing holes

- *Geographical forwarding uses the Euclidean coordinates as routing guidance, but routing is done on the connectivity graph.*

  *When there are holes in the sensor field, they mismatch.*

- *Thus the essential problem is, how to route around holes?*

# How to get around obstacles?

- *Face routing says: "walk along the bank of the lake", thus the bank is crowded. In fact, a rough idea on how to get around is sufficient to escape from local minima and alleviates overloading the boundaries of holes.*

- *Use an abstraction of the geometry/topology of the sensor field.*

  *E.g., GLIDER by Fang et.al uses a combinatorial Delaunay Graph to capture the global topology (holes, etc).*

- *We use the medial axis of the underlying geometric domain as a compact representation of global geometric/topological features.*

# General methodology – 2-level infrastructure

- *When the sensor field has complex geometric shape or nontrivial topology,*

- *Top level: a compact abstraction of the geometry/topology of the sensor field.*

  *E.g., there is a hole in the middle of the sensor field.*

- *Bottom level: a naming scheme with respect to the global topology that enables local gradient routing.*

# General methodology – Routing

- *When the sensor field has complex geometric shape or nontrivial topology,*

- *Top level: a compact abstraction of the geometry/topology of the sensor field.*

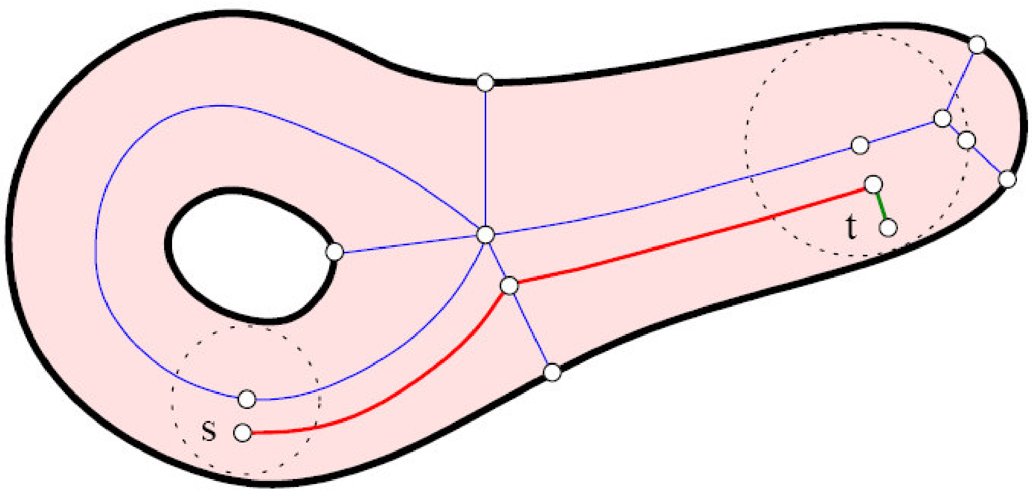
  *Check the compact abstract graph to get a global guidance on how to get around obstacles.*

- *Bottom level: a naming scheme with respect to the global topology that enables local gradient routing.*

  *The actual routing is performed by using local information to do gradient descending.*

# MAP – Medial Axis based Naming/Routing

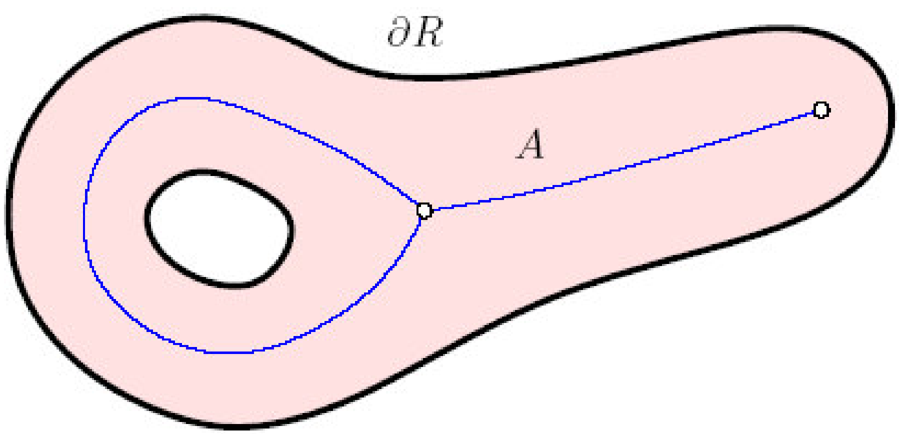
- *We propose MAP --- Medial Axis based Naming/Routing Protocol*




- *Properties:*
  - *MAP utilizes useful geometry information*
  - *Location free*
  - *Expressive*
  - *Compact*
  - *Lightweight*
  - *Efficient*
  - *Load balancing*
  - *Robust to network model*

# Medial Axis --- Definitions

*Given a bounded region* $\boldsymbol{R}$*, the medial axis of its boundary* $\partial\boldsymbol{R}$ *is the collection of points with two or more closest points in* $\partial\boldsymbol{R}$ *.*

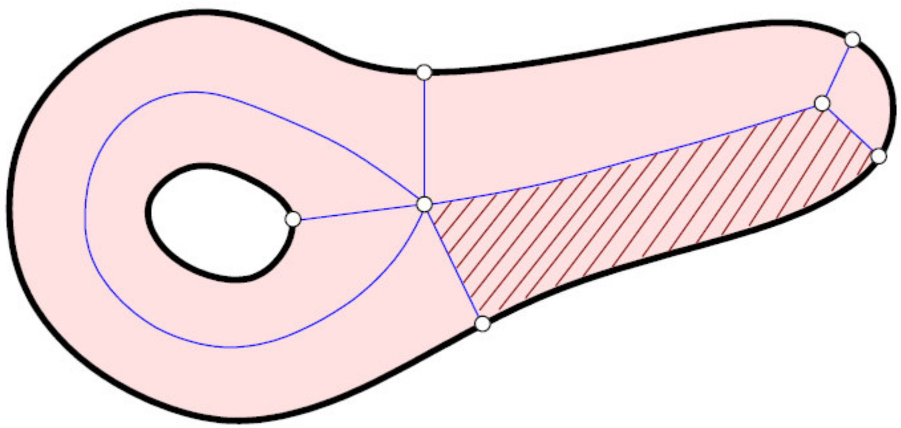



*The medial axis of a piecewise analytic curve is a finite number of continuous curves.*

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

# Partitioning into Canonical Regions

*A chord is a line segment connecting a point on the medial axis and its closest point on $\partial \boldsymbol{R}$. A point on the medial axis with 3 or more closest points on $\partial \boldsymbol{R}$ is called a medial vertex.*

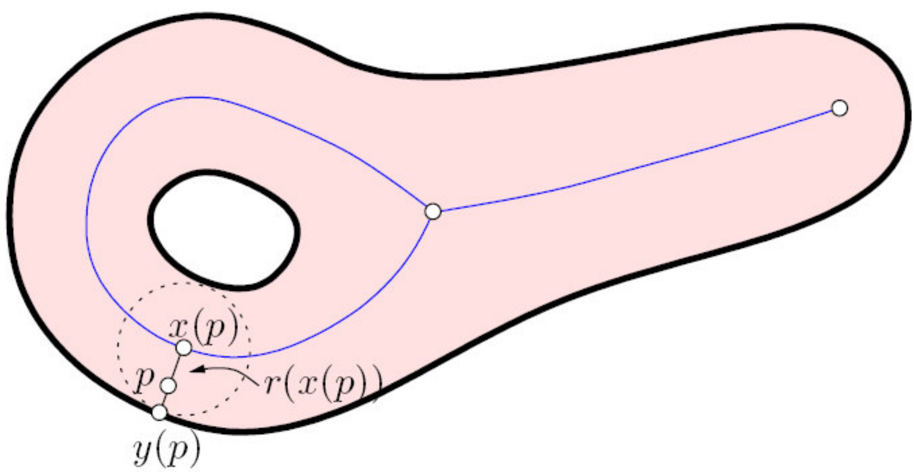


*We can partition the region $\boldsymbol{R}$ by the medial axis and the chords of medial vertices into canonical pieces, each resembling a stretched rectangular region.*

# 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 $\partial \boldsymbol{R}$.
d(p) is height, i.e., relative distance from x(p): |px(p)|/|x(p)y(p)|.

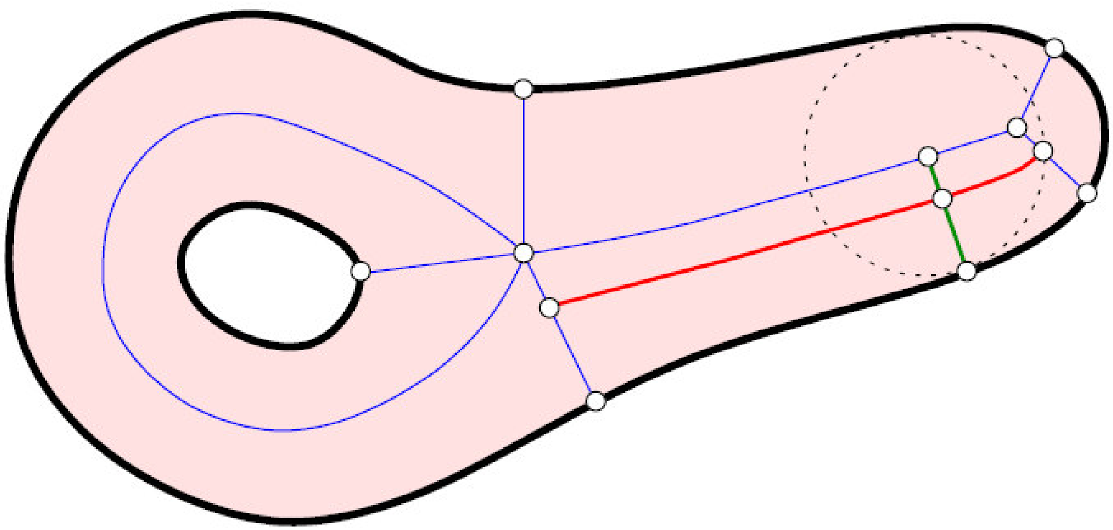



Theorem: each point is given a unique name.

# Routing inside a canonical piece

*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.*

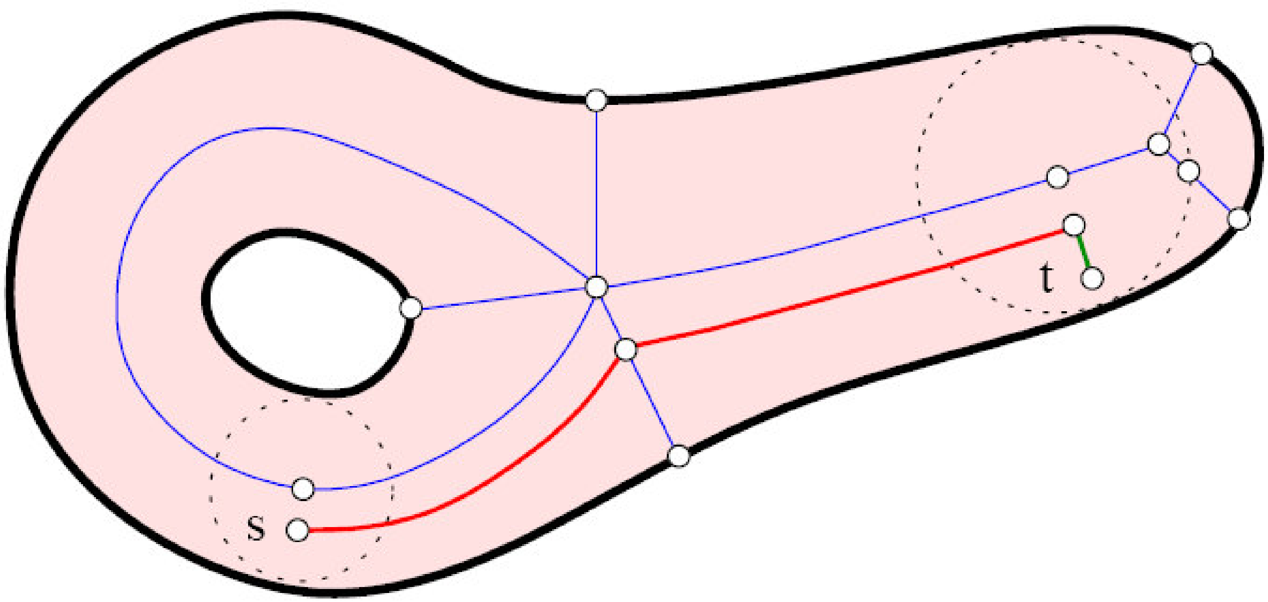


*Inside a canonical piece, we just do Manhattan routing!*

# Routing between canonical pieces

*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.*

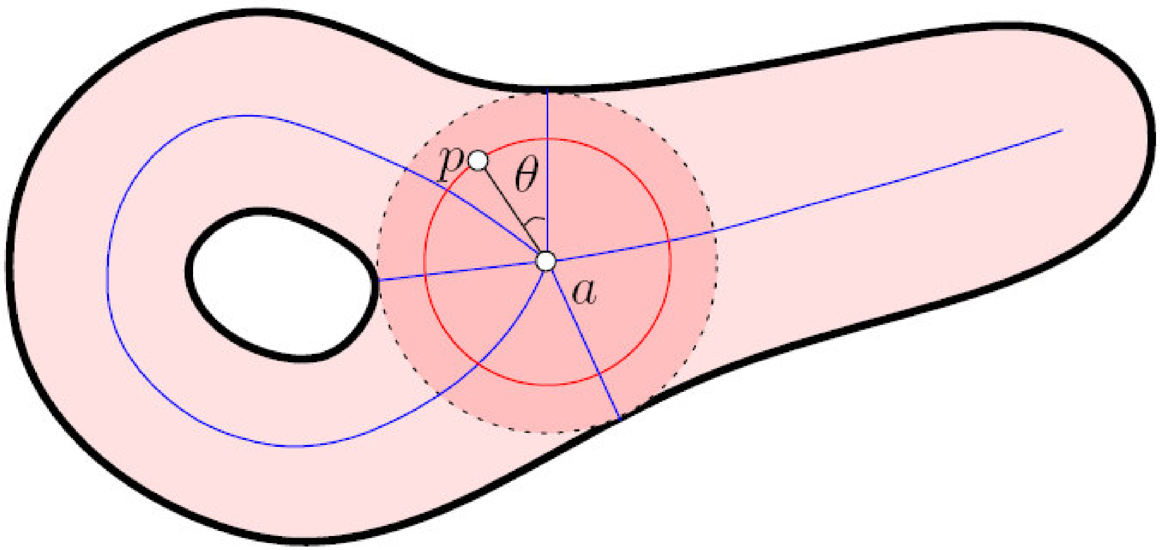


# Routing between canonical pieces

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

A fix: build rotary systems around medial vertices.

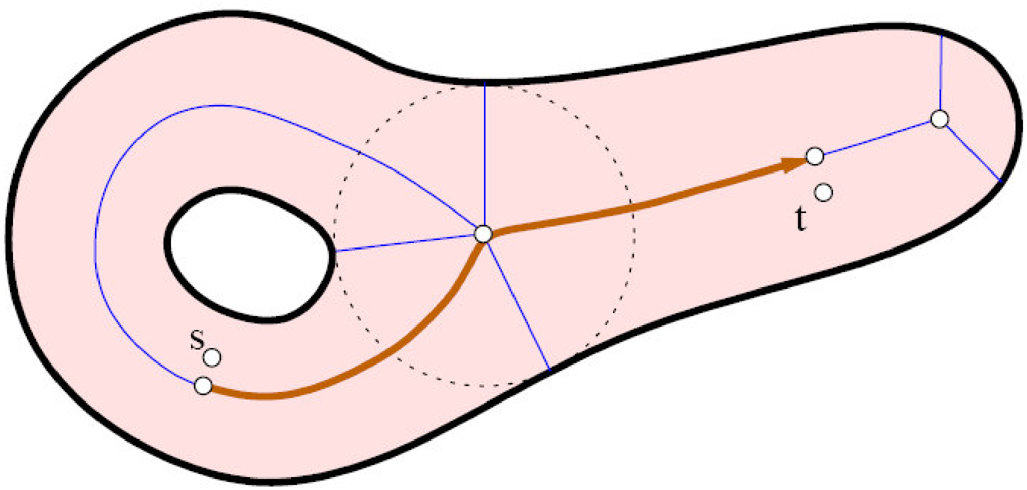
Polar coordinate system: $(|ap|/r, \theta)$, r is the maximum radius of a empty ball centered at a medial vertex a.



# Routing between canonical pieces

*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.*

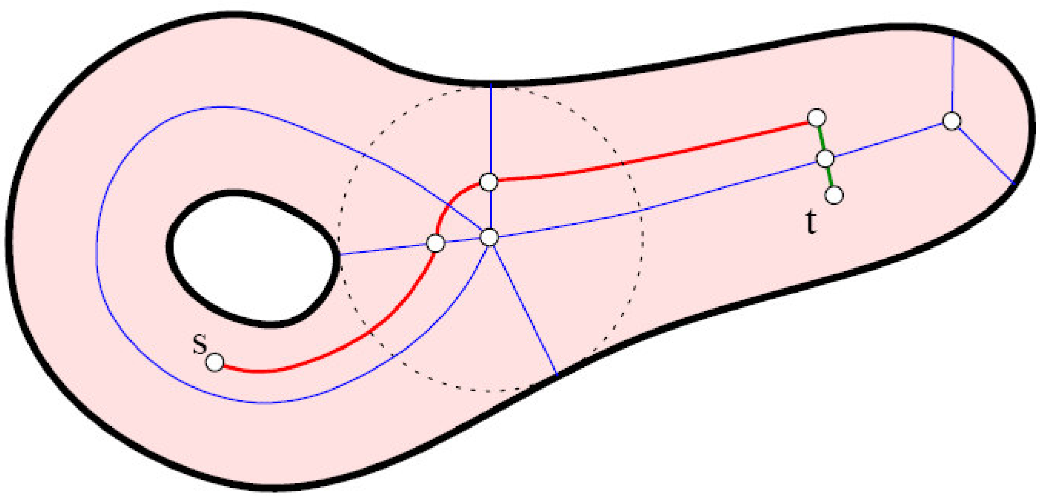


# Routing between canonical pieces

*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.*



# Challenges of MAP in discrete networks

- *The hop count is only a rough approximation to the Euclidean distance.*
- *The exact medial axis is sensitive to noises.*
- *Low cost and distributed construction of a robust medial axis is desirable.*

*We use the same intuition as in the continuous case but keep these challenges in mind.*

# MAP in discrete networks --- a sketch

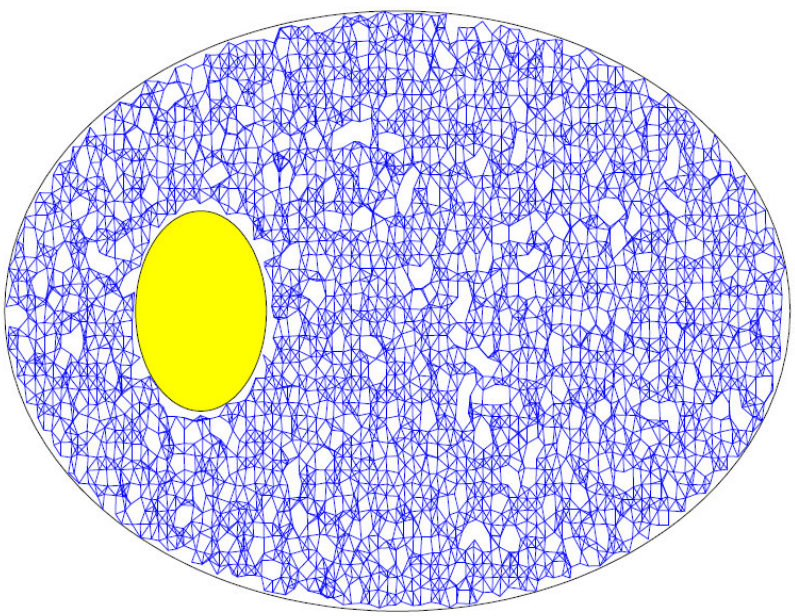
- *Sketch of Naming Protocol*
  - Detect boundaries of the sensor field.
  - Construct the medial axis graph.
  - Assign names to sensors.

- *Sketch of Routing Protocol*
  - Mimic Manhattan routing.
  - Guarantee delivery.

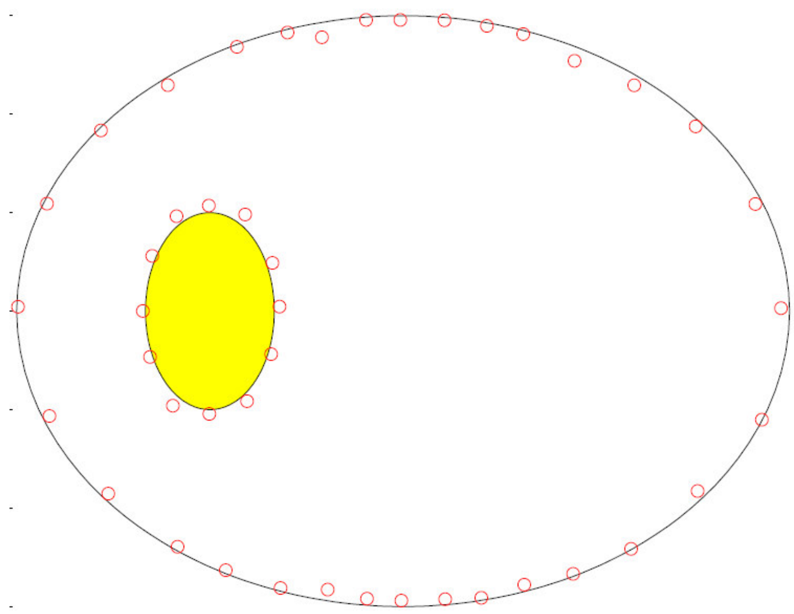
# MAP in discrete networks --- naming

- Detect boundaries of the sensor field.
  - *Find sample nodes on boundaries.*

    *By manual identification, or*
    *automatic detection [Fekete'04, Funke'05]*



*Network*

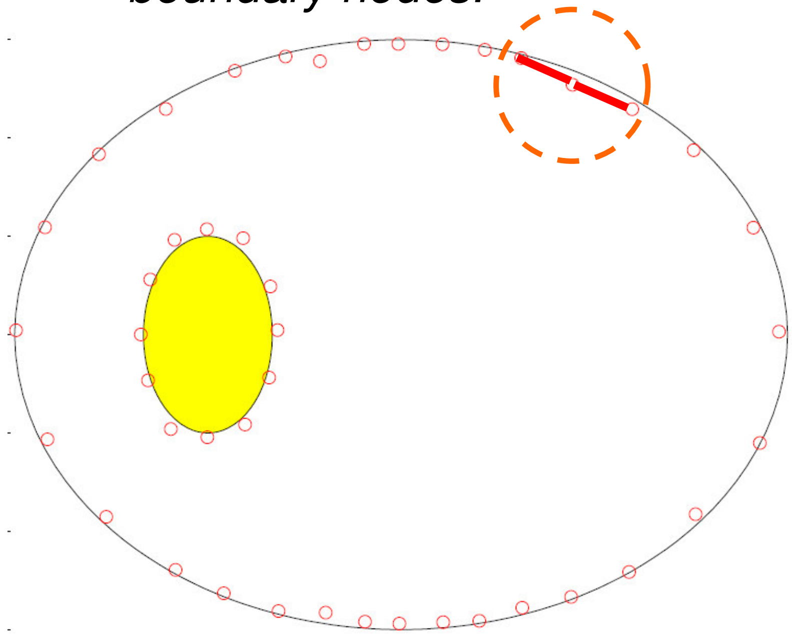


*Boundary nodes*

# MAP in discrete networks --- naming

- Detect boundaries of the sensor field.
  - *Detect boundaries (a curve reconstruction problem).*

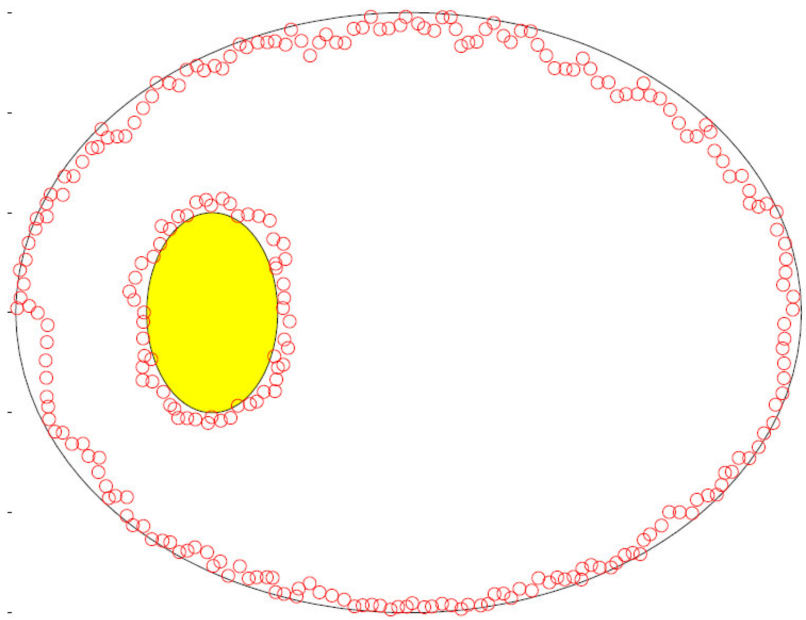
  *Method: use local flooding to connect nearby boundary nodes, and include nodes on the shortest path between them as boundary nodes.*



# MAP in discrete networks --- naming

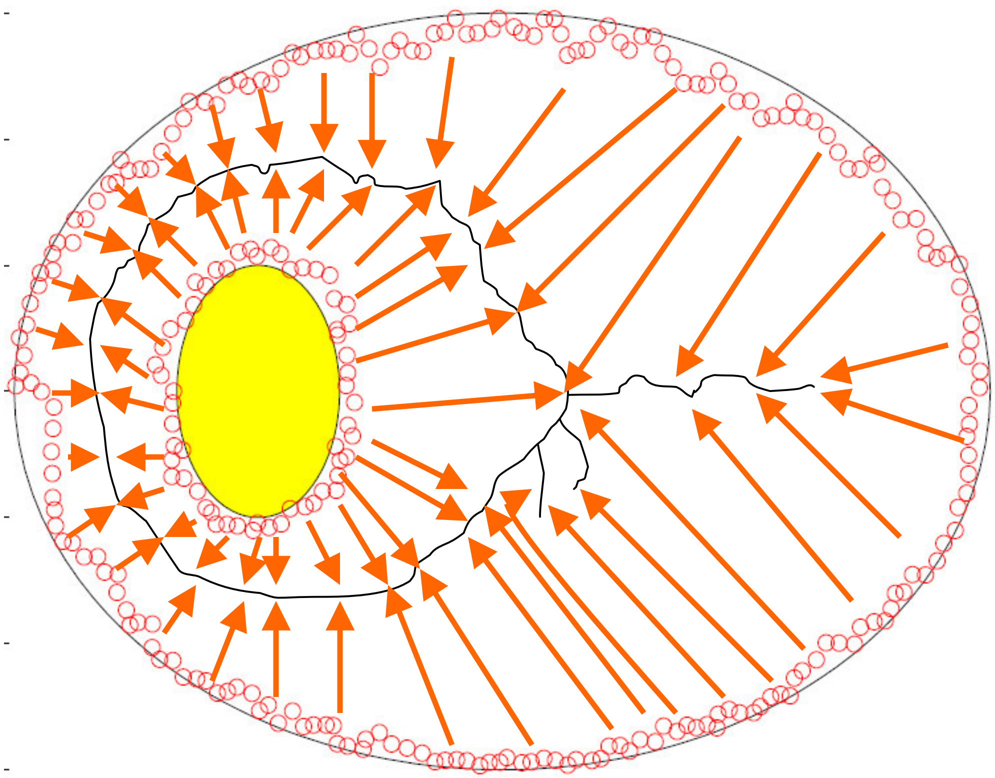
- Detect boundaries of the sensor field.
  - *Detect boundaries (a curve reconstruction problem).*

  *Method: use local flooding to connect nearby boundary nodes, and include nodes on the shortest path between them as boundary nodes.*



# MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Detect medial nodes (the sensors with 2 or more closest boundary nodes) by restricted flooding.*

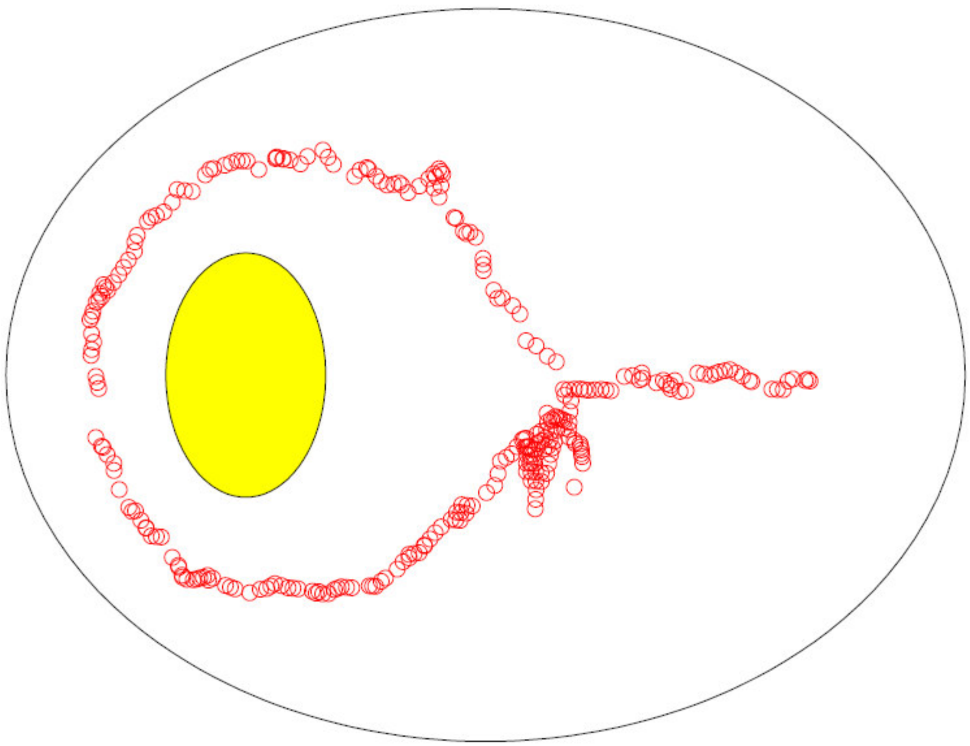


The flooding is in fact a Voronoi partition of the network. So every node receives only one or a few flooded messages.

To suppress noise, for those nodes whose closest boundary nodes are on the same boundary and are very close to each other, we do not consider them to be medial nodes.

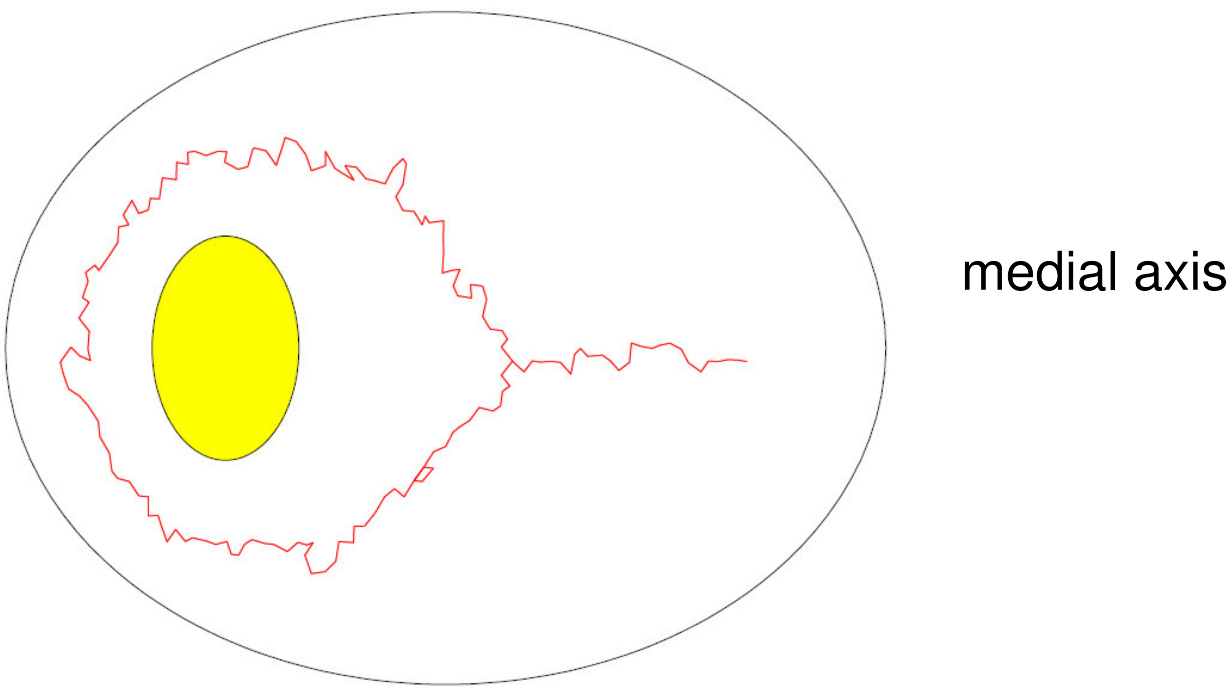
# MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Detect medial nodes (the sensors with 2 or more closest boundary nodes) by restricted flooding.*



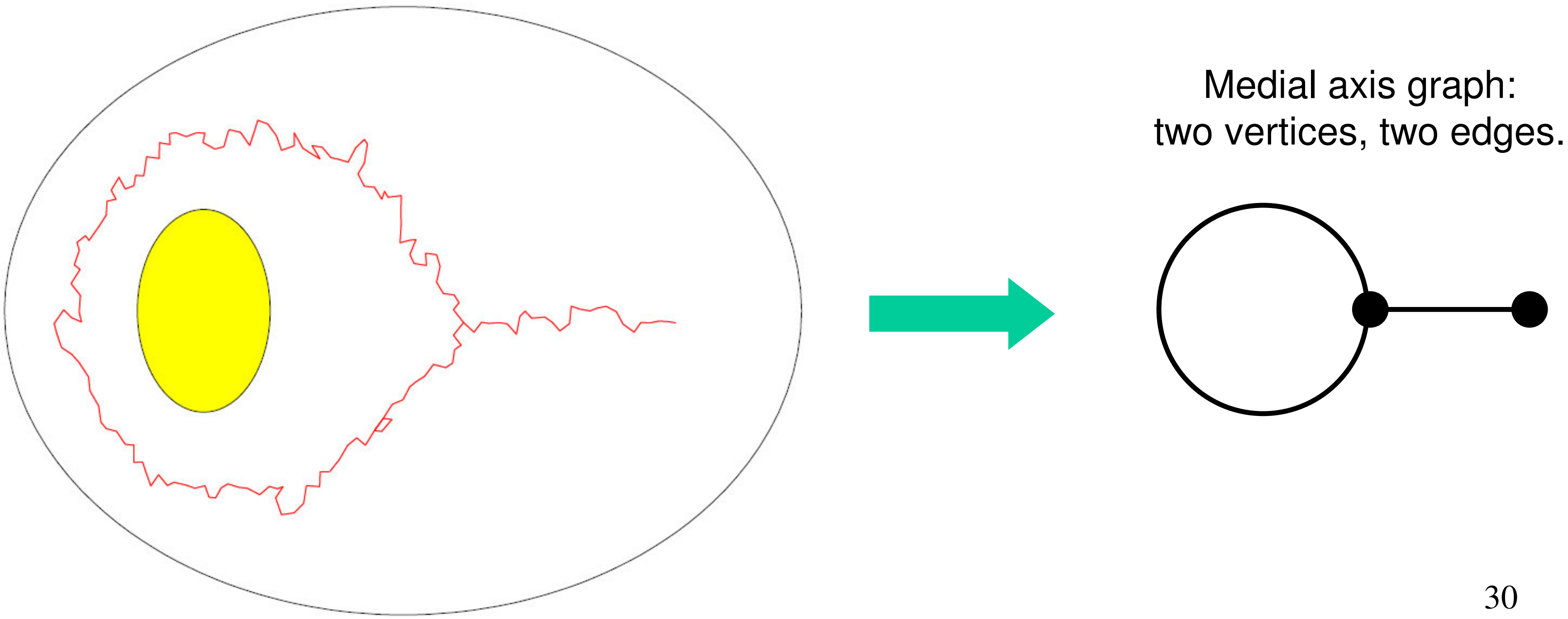
# MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Connect medial nodes into a graph and clean it up (remove very short branches).*



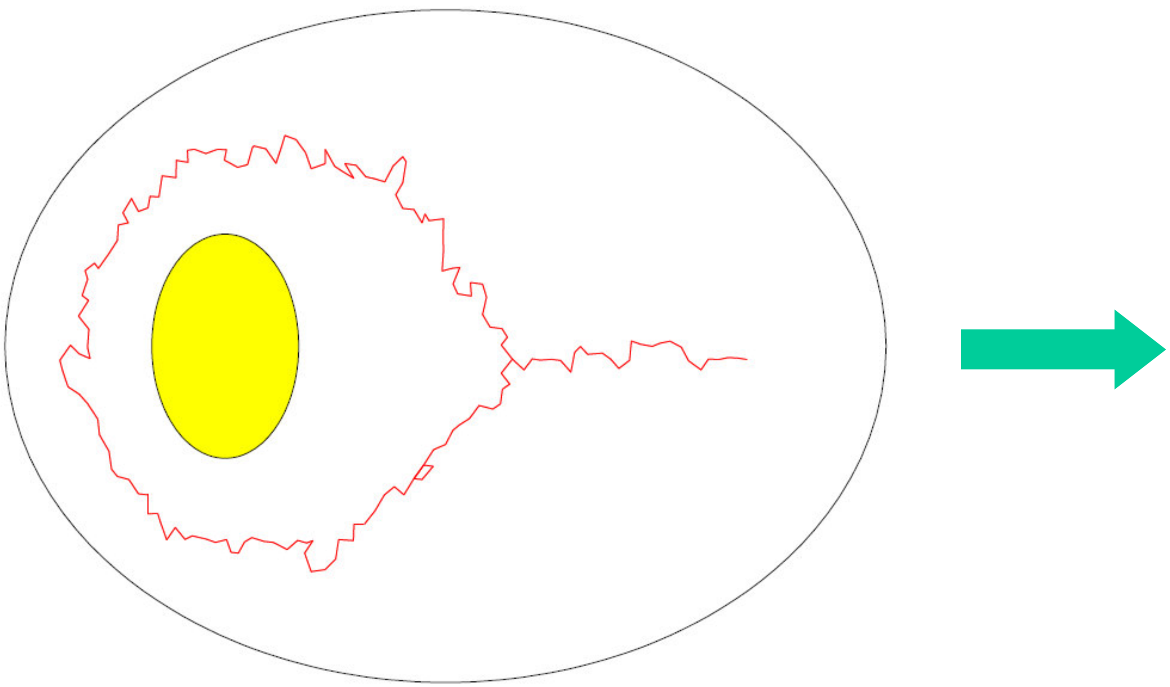
# MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Connect medial nodes into a graph and clean it up (remove very short branches).*

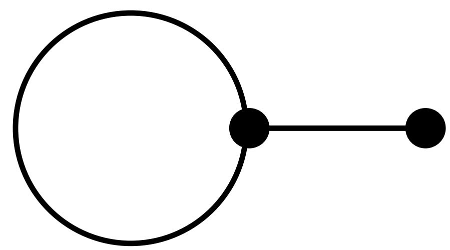


# MAP in discrete networks --- naming

- Construct the medial axis graph.
  - *Connect medial nodes into a graph and clean it up (remove very short branches).*



Medial axis graph:
two vertices, two edges.

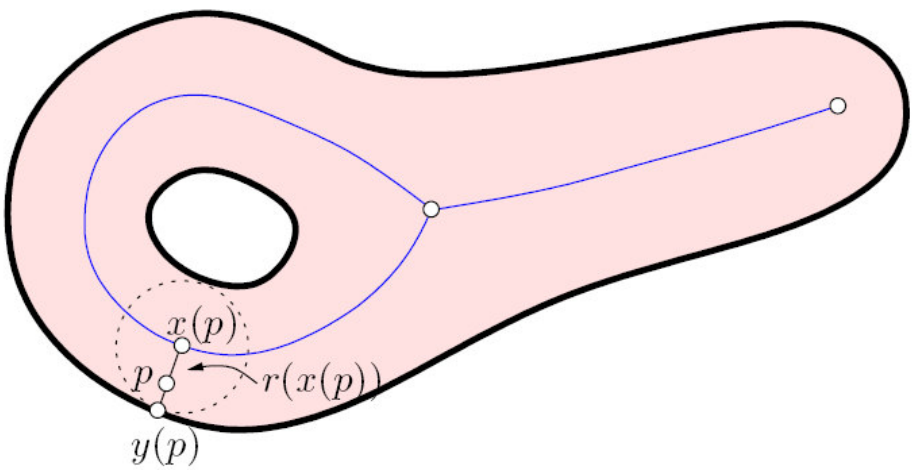


Broadcast this simple graph to all sensors.

# MAP in discrete networks --- naming

- Assign names to sensors.

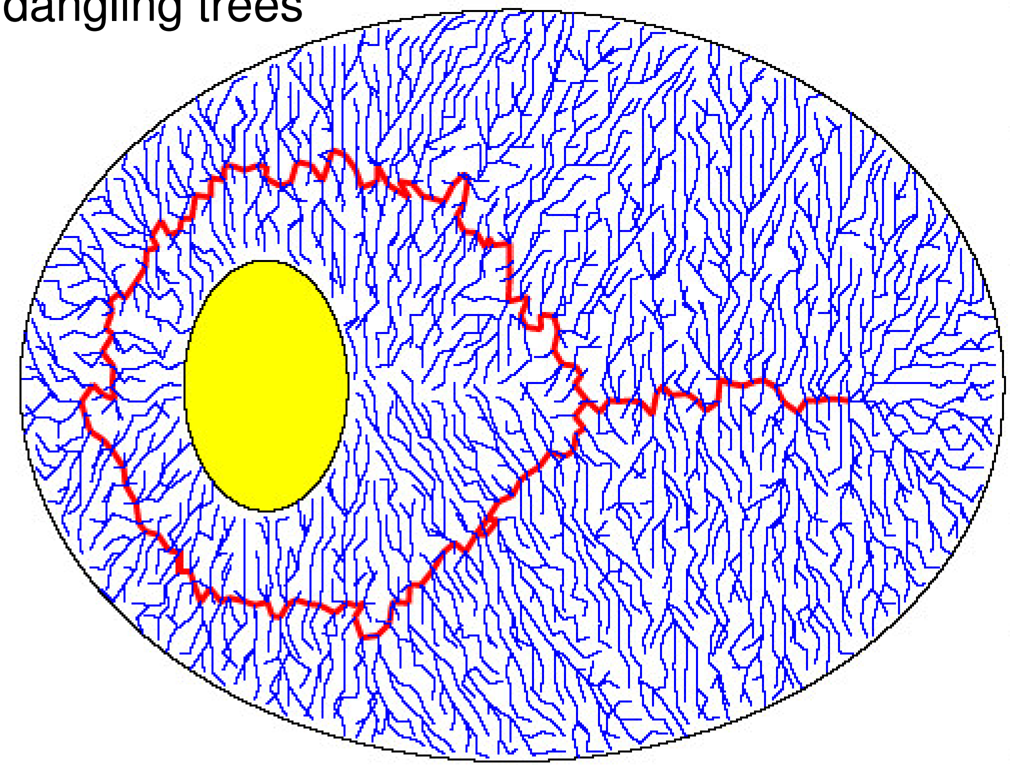
*Recall: in the continuous case, a point is named based on the medial axis graph and the corresponding chord.*



# MAP in discrete networks --- naming

- Assign names to sensors for a discrete network:
  - *Replace chords by (approximate) shortest path trees.*

"Medial axis with dangling trees"



# MAP in discrete networks --- naming

- Assign names to sensors for a discrete network:
  - *Replace chords by (approximate) shortest path trees.*
  - *Nodes are assigned names w.r.t. where it lies in its tree.*
  - *Take advantage of the discreteness, assign names in a way to make it easy for insertion / deletion of nodes and edges.*

*All the computation is simple and local.*

# MAP in discrete networks --- routing

- *Medial Axis based Routing Protocol*
  - Mimic Manhattan routing.
  - Guaranteed delivery:

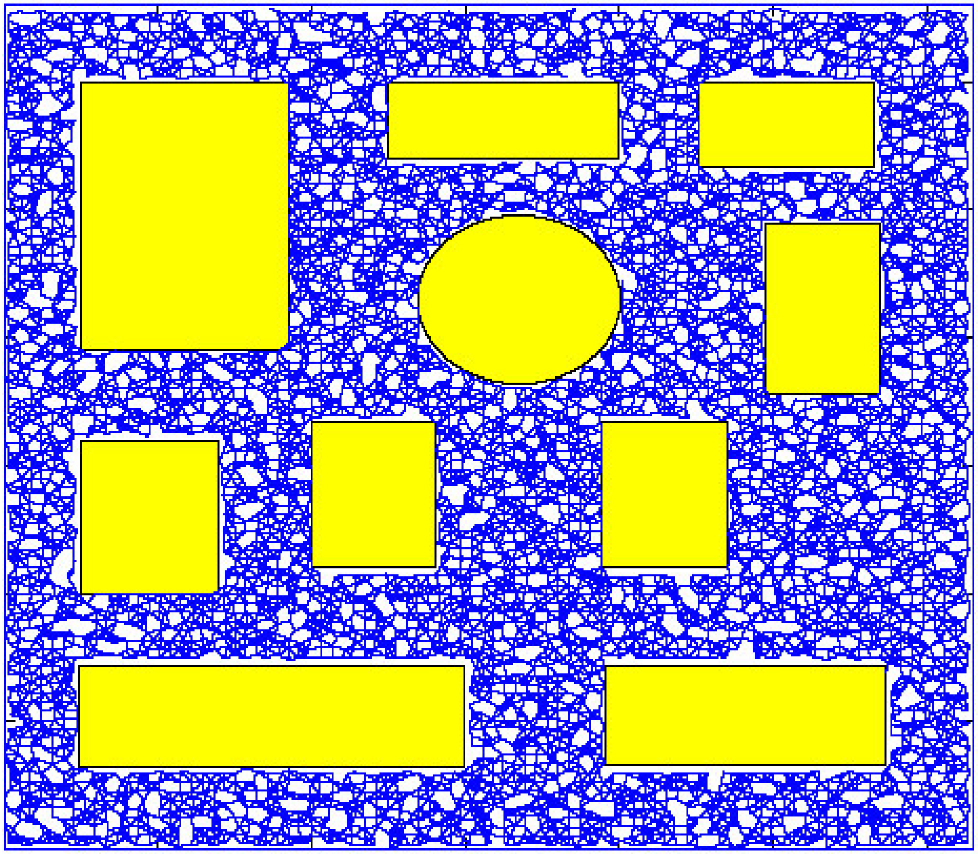
    If there is no better choice, route toward the medial axis.
  - Maintain balanced load:

    Try to route in parallel with the medial axis as much as possible, to avoid overloading nodes near the medial axis.
  - Building a small neighborhood routing table (e.g., a table for nodes within 3 hops) improves routing performance.

    Due to the discreteness of hop count distance.

# Simulation Examples

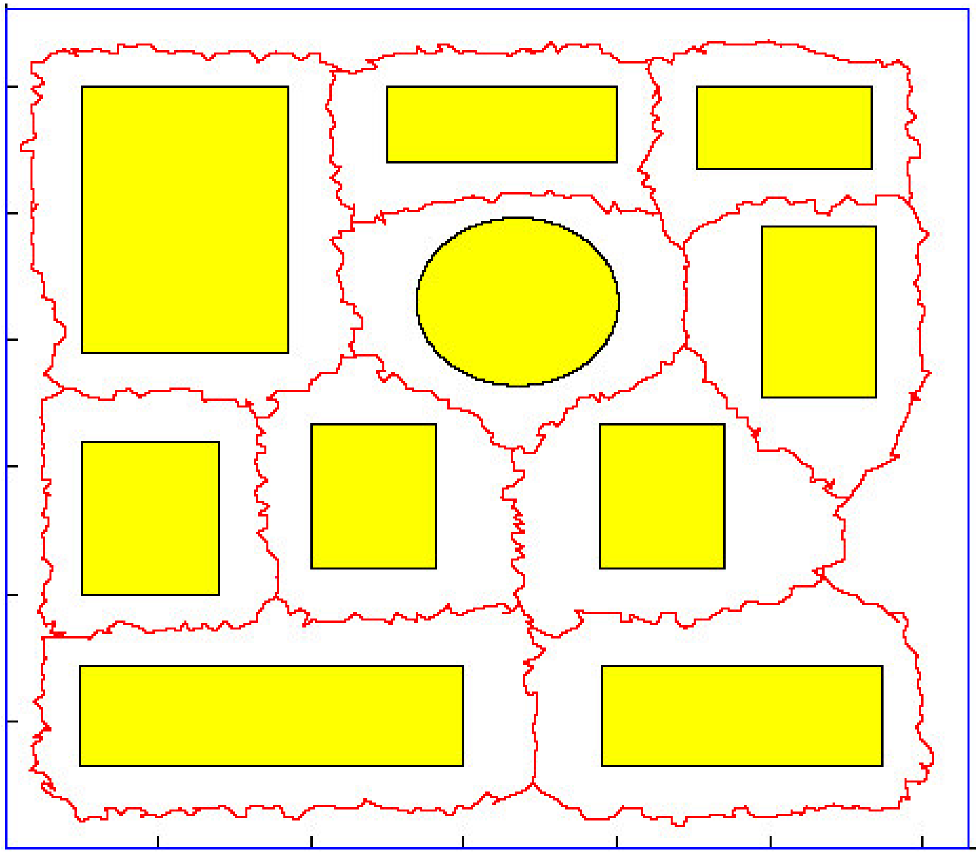
- *Outdoor sensor field: Campus*



5735 nodes in
the sensor network

# Simulation Examples

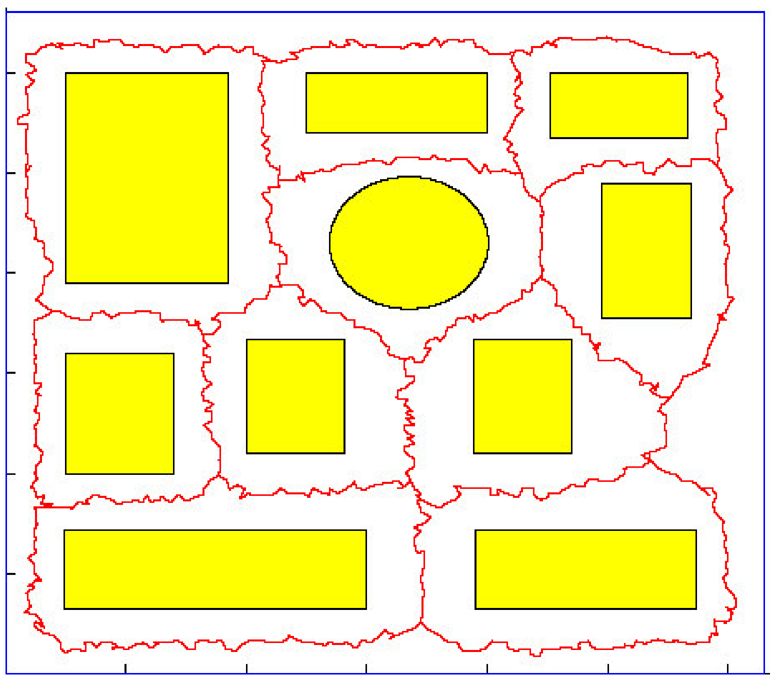
- *Outdoor sensor field: Campus*

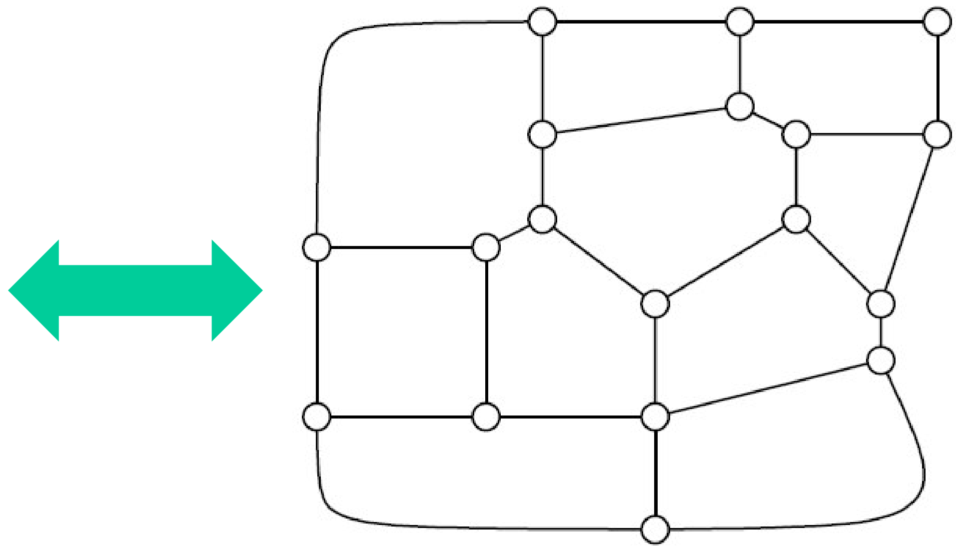


Medial Axis

# Simulation Examples

- *Outdoor sensor field: Campus*





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

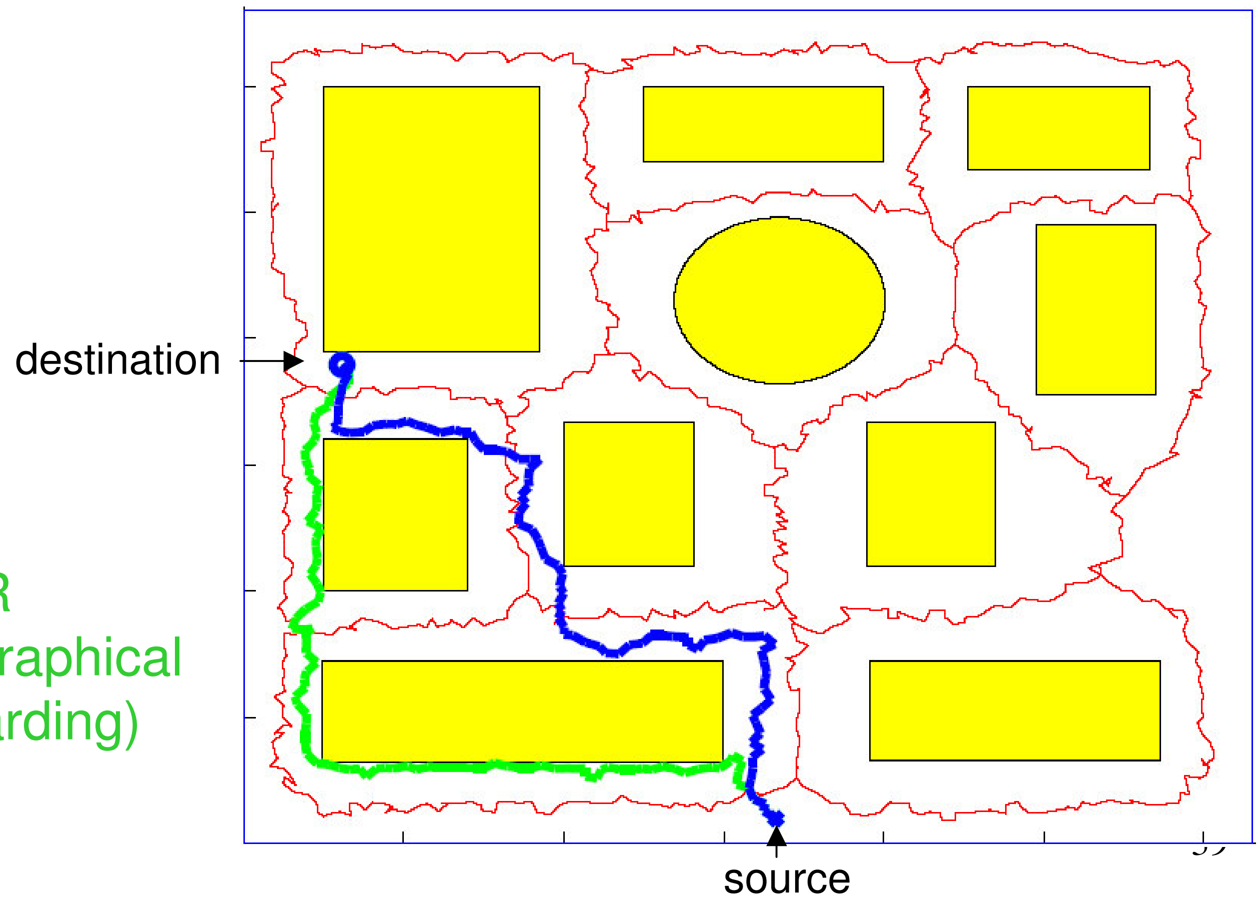
# Simulation Examples

- *Outdoor sensor field: Campus*

Routing path comparison:

Blue: MAP

Green: GPSR (geographical forwarding)

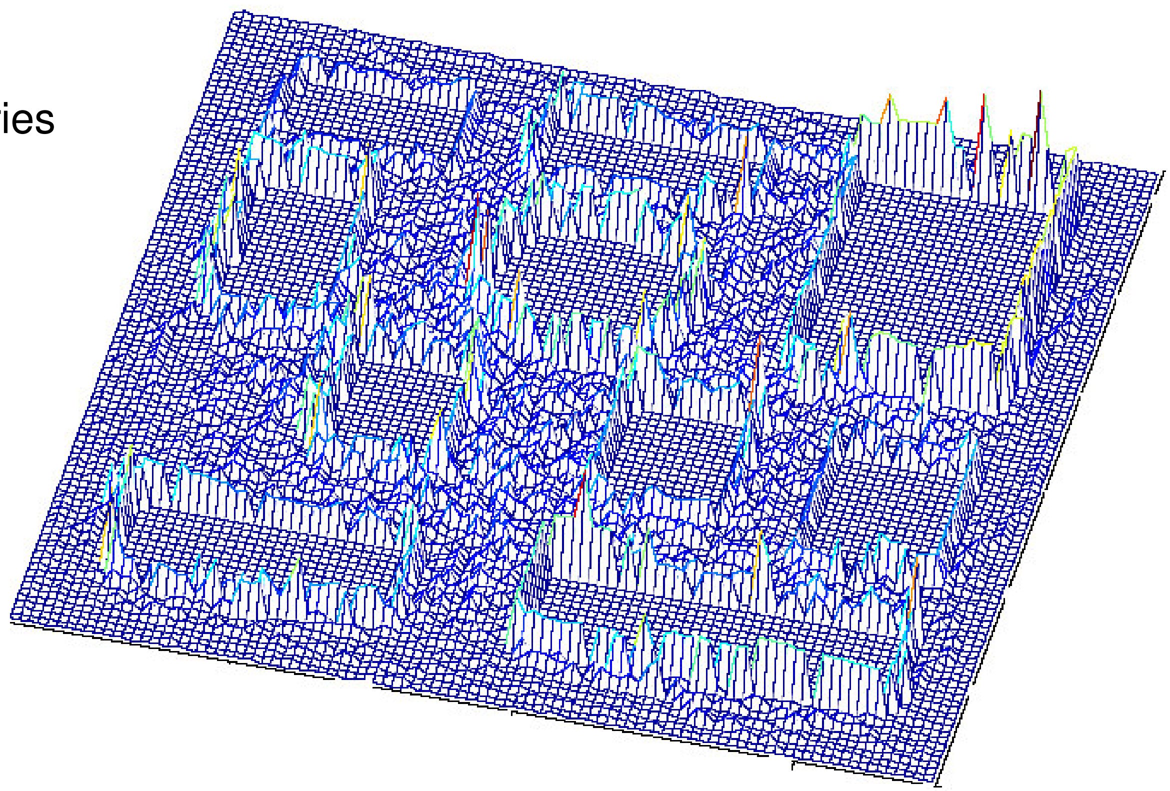


# Simulation Examples

- *Outdoor sensor field: Campus ------ Load Balance Comparison*

GPSR (Geographical Forwarding): Unbalanced Load

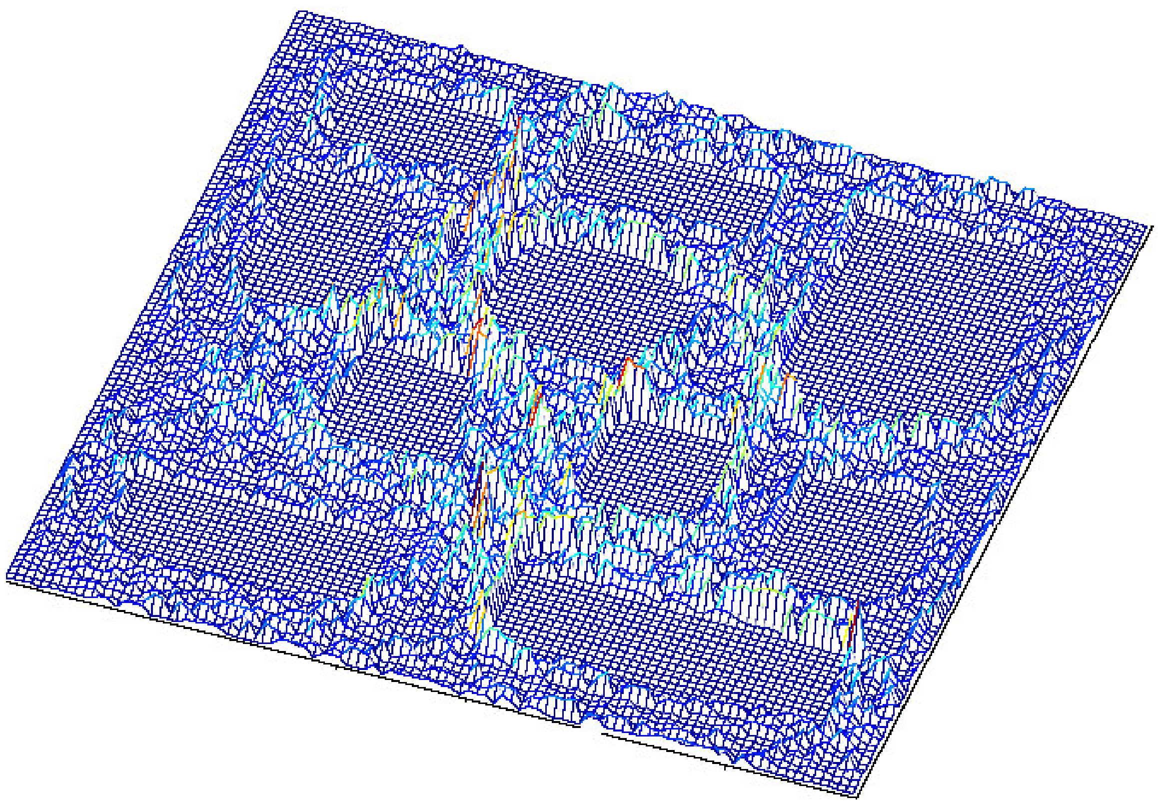
Nodes on boundaries are overwhelmed.



# Simulation Examples

- *Outdoor sensor field: Campus ------ Load Balance Comparison*

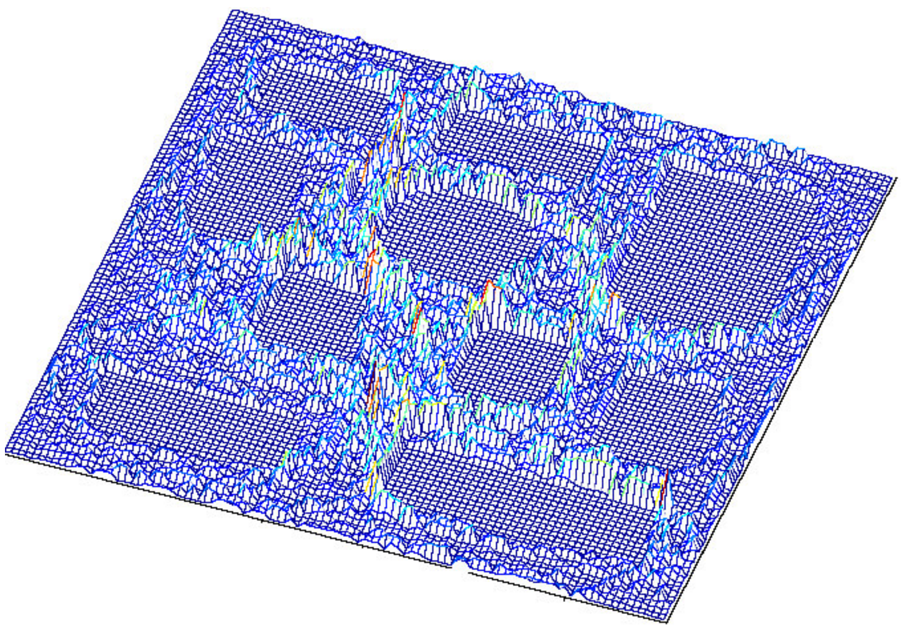
MAP: Well Balanced Load



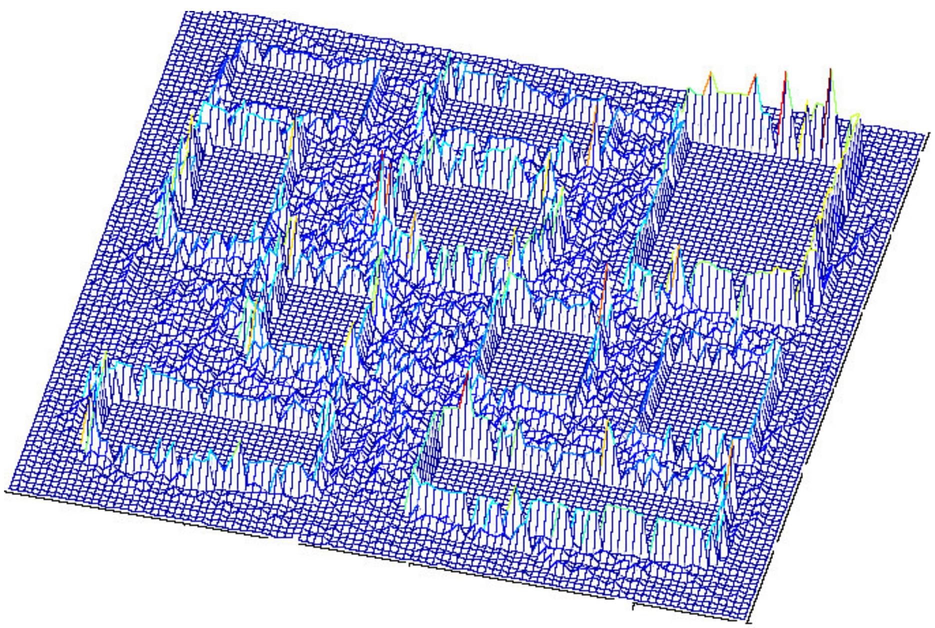
# Simulation Examples

- *Outdoor sensor field: Campus ------ Load Balance Comparison*

MAP:



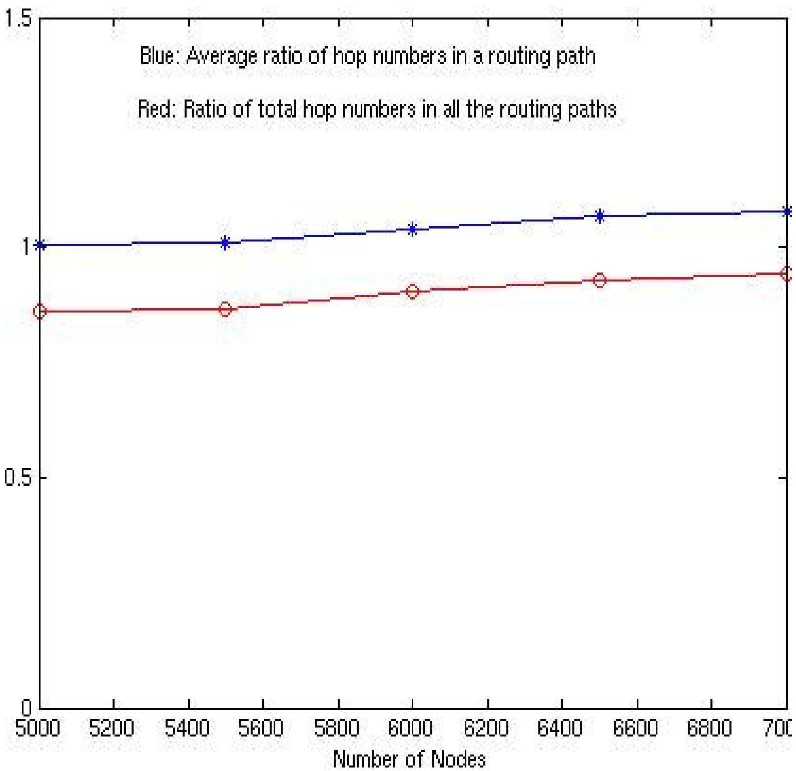
GPSR (Geographical Forwarding)

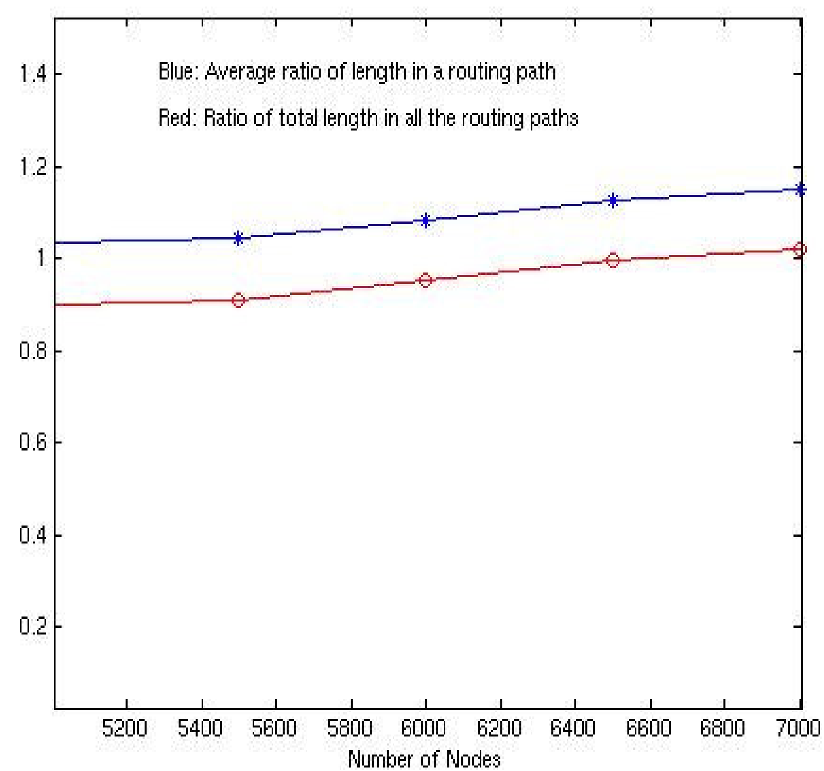


# Simulation Examples

- *Outdoor sensor field: Campus ---- Routing Distance Comparison*

| | Number of hops | | Euclidean length | |
|---|---|---|---|---|
| For the *i*-th path: | MAP: $h_i$ | GPSR: $H_i$ | MAP: $\ell_i$ | GPSR: $L_i$ |



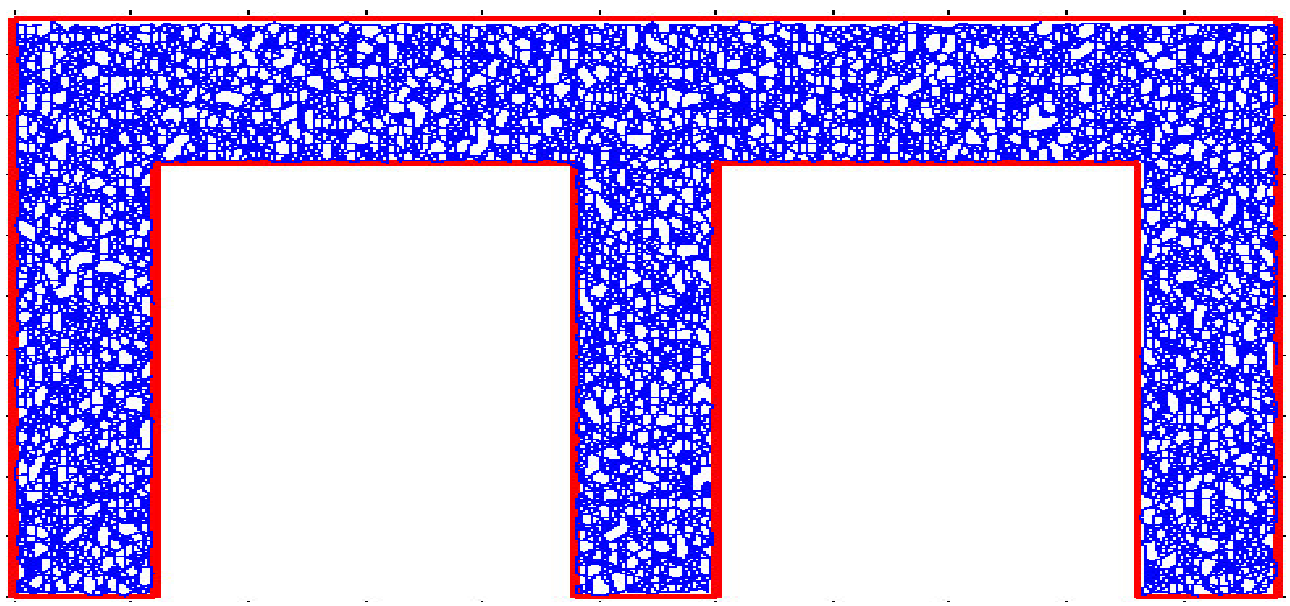




Blue: $\frac{1}{N}\sum_{i=1}^{N} h_i/H_i$ Red: $\frac{\sum_i h_i}{\sum_i H_i}$

Blue: $\frac{1}{N}\sum_{i=1}^{N} \ell_i/L_i$ Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$

# Simulation Examples

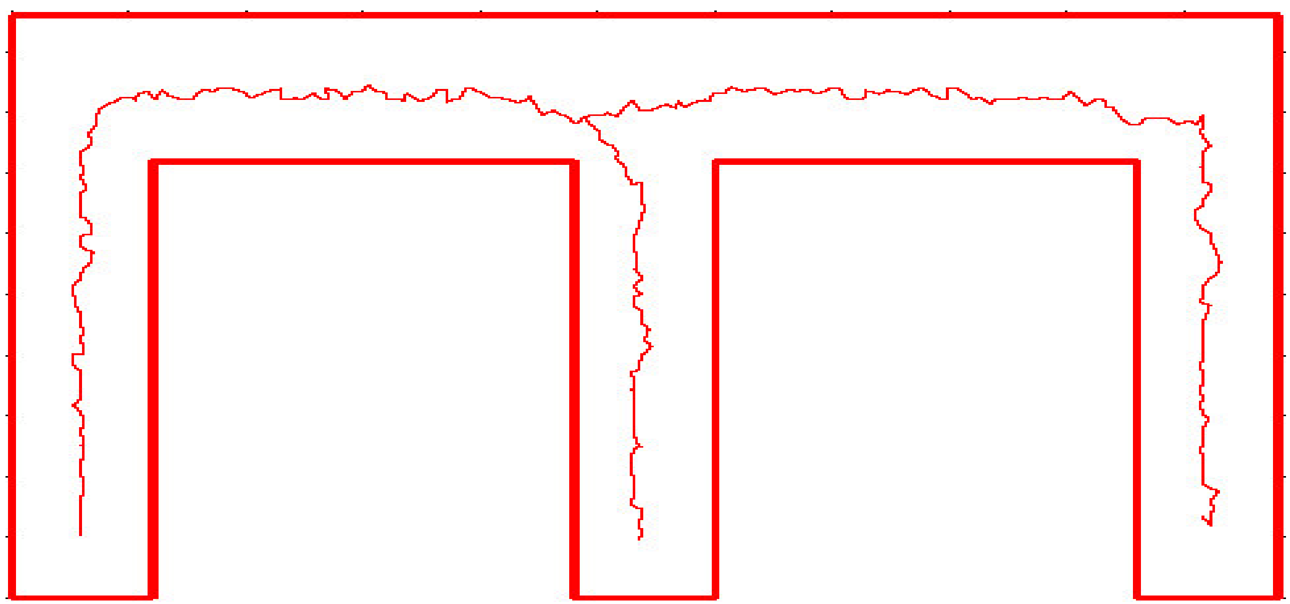
- *Indoor sensor field: Airport Terminals*



5204 nodes in
the sensor network

# Simulation Examples

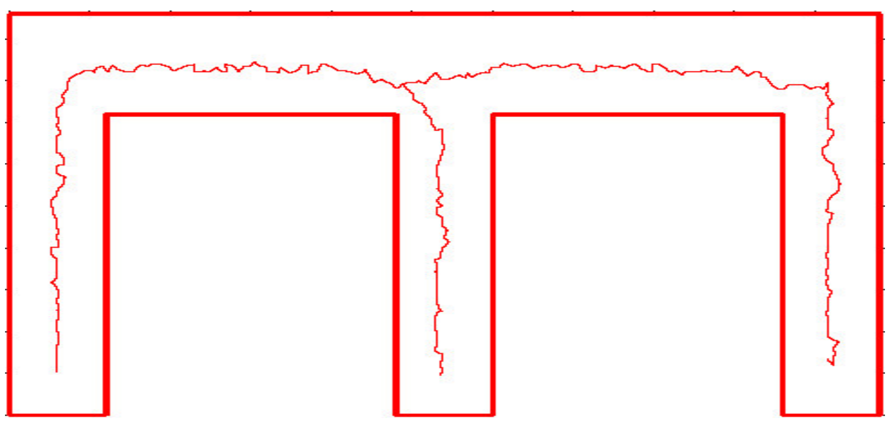
- *Indoor sensor field: Airport Terminals*

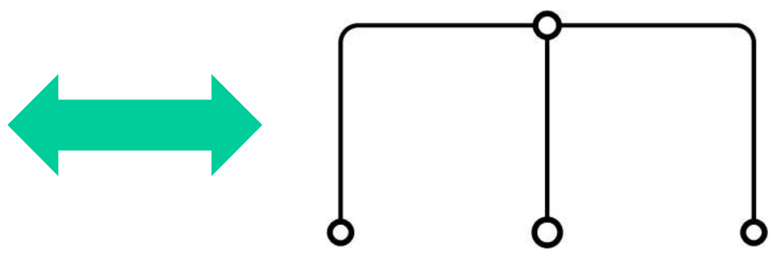


Medial Axis

# Simulation Examples

- *Indoor sensor field: Airport Terminals*





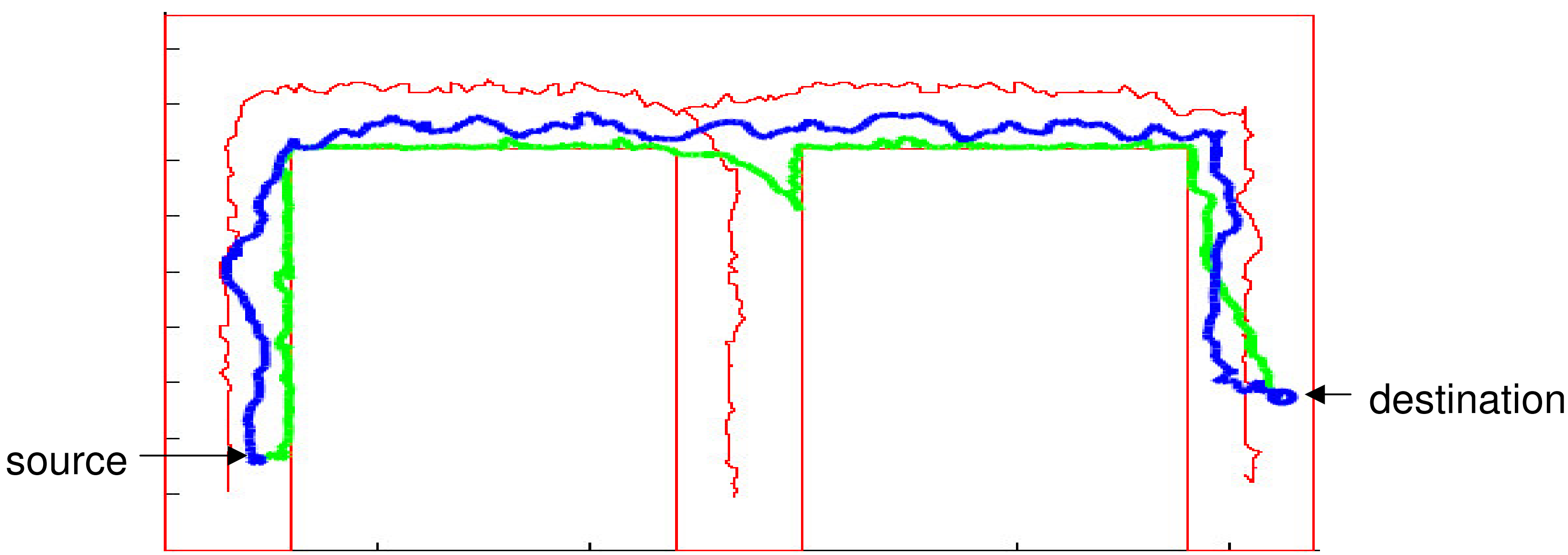
The simple medial axis graph:
4 nodes, 3 edges.

# Simulation Examples

- *Indoor sensor field: Airport Terminals*

Routing path comparison:

Blue: MAP　　　Green: GPSR (geographical forwarding)

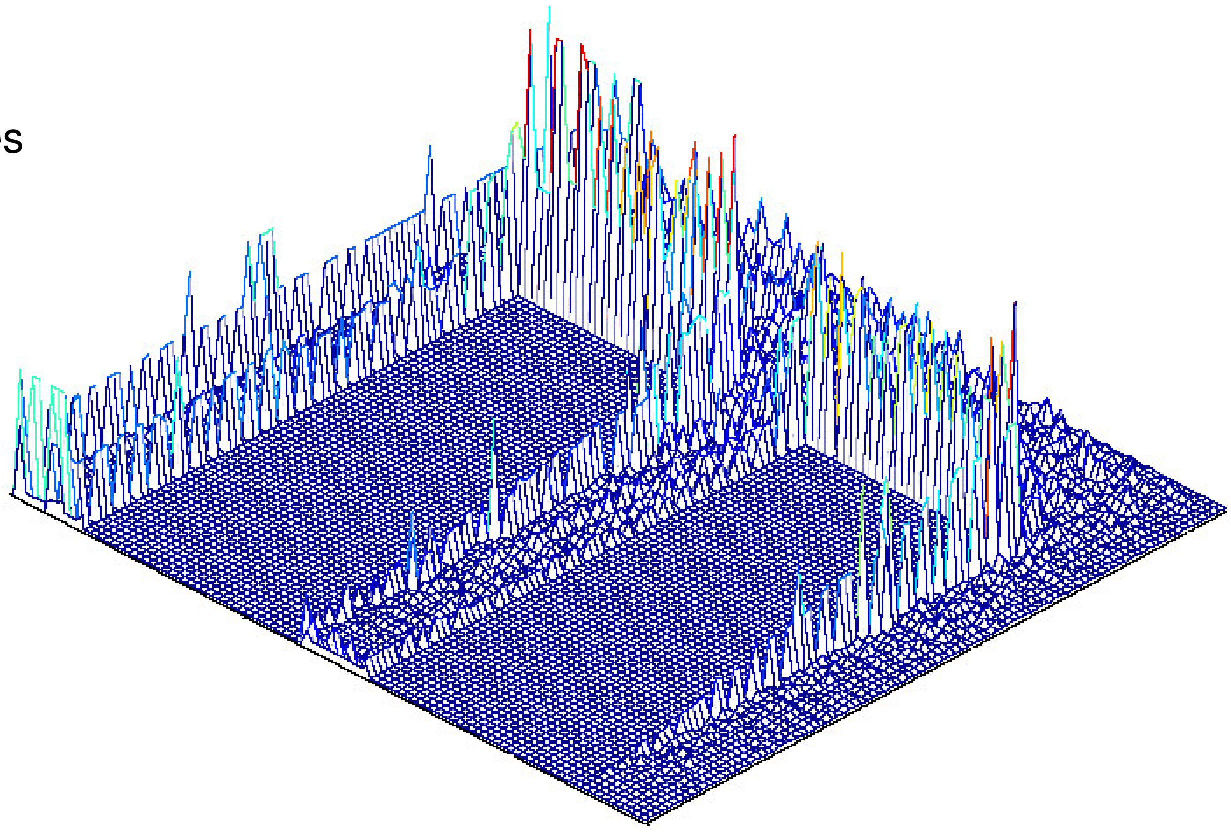


# Simulation Examples

- *Indoor sensor field: Airport Terminals  ------ Load Balance Comparison*

GPSR (Geographical Forwarding): Unbalanced Load

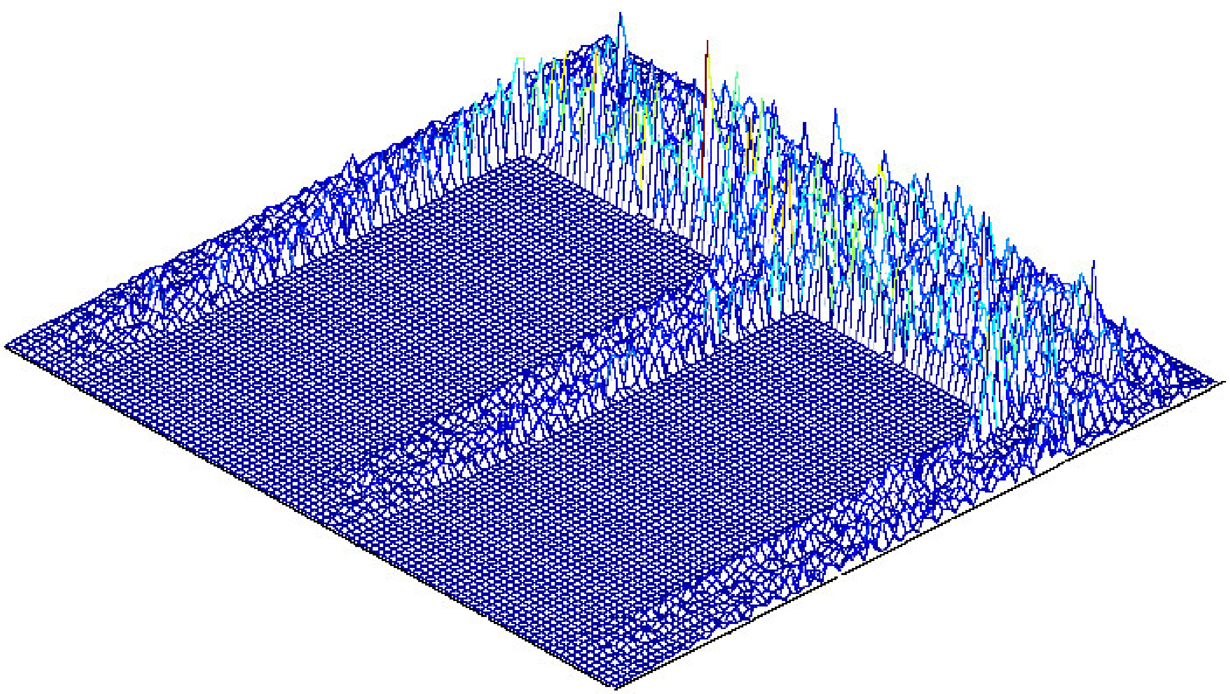
Nodes on boundaries
are overwhelmed.



# Simulation Examples

- *Indoor sensor field: Airport Terminals  ------ Load Balance Comparison*

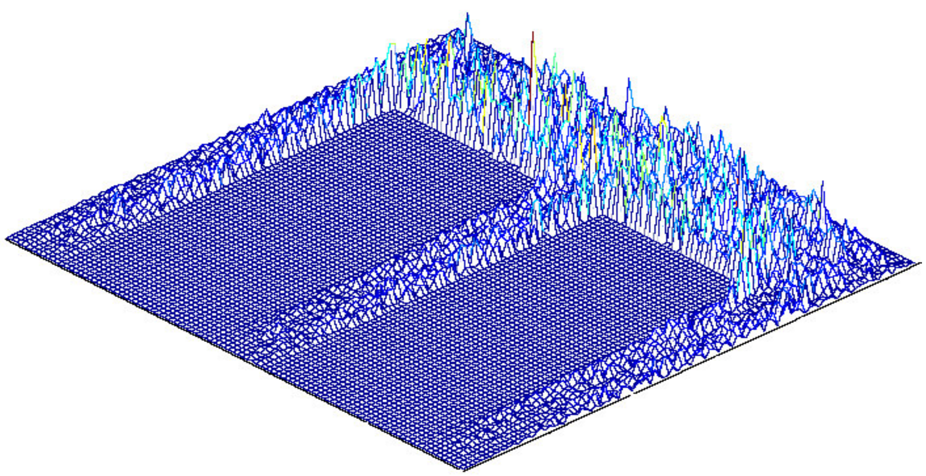
MAP: Well Balanced Load



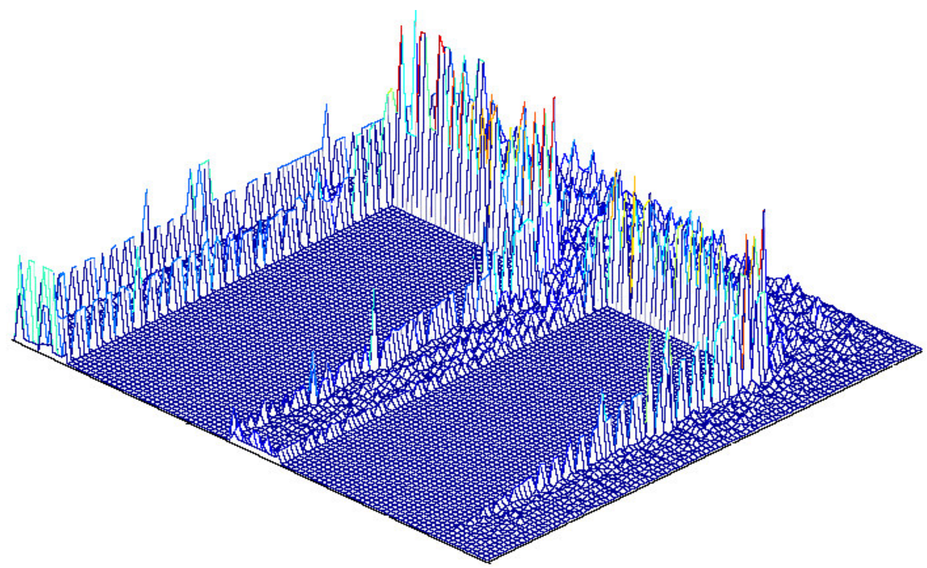
# Simulation Examples

- *Indoor sensor field: Airport Terminals  ------ Load Balance Comparison*

MAP:



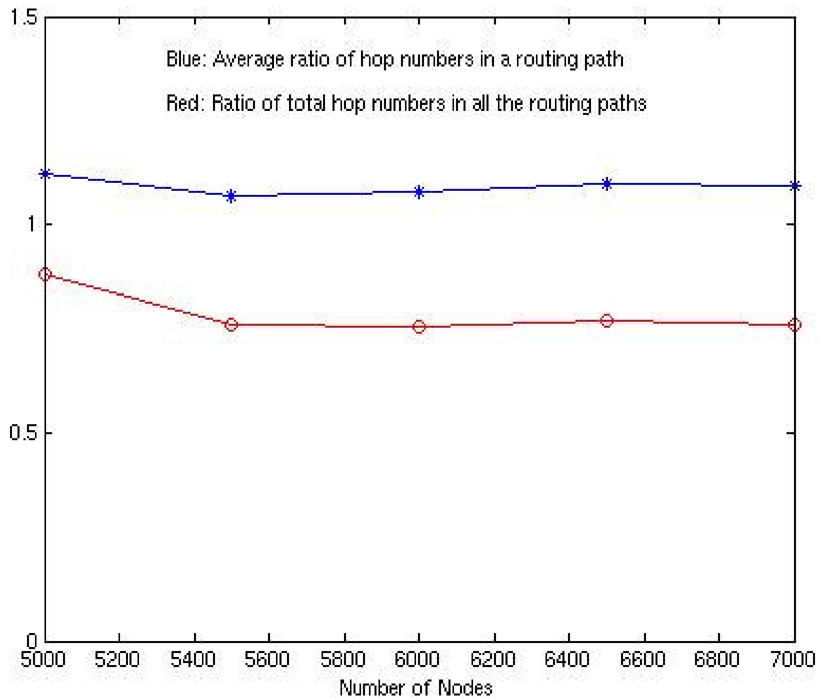
GPSR (Geographical Forwarding)

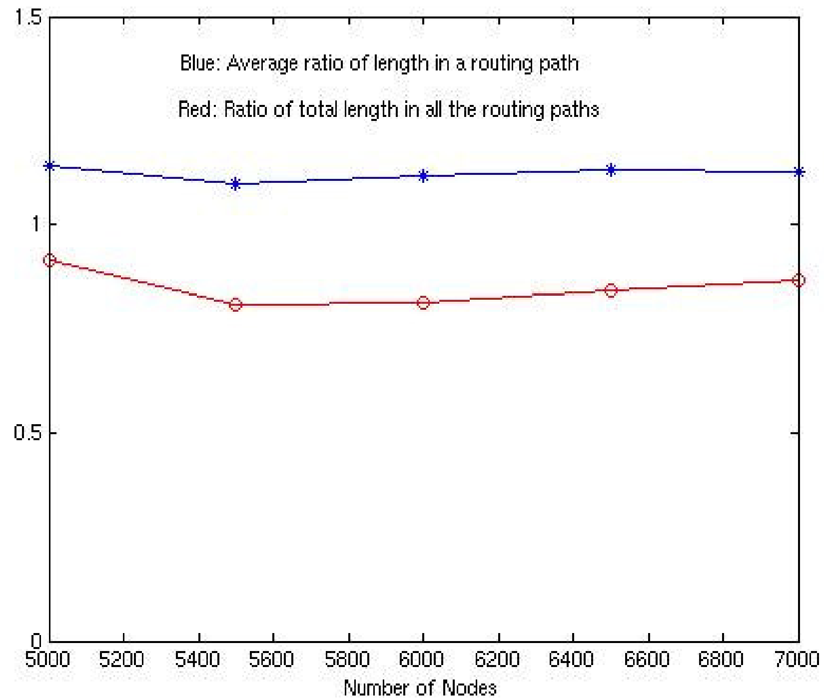


# Simulation Examples

- *Indoor sensor field: Airport Terminals ---- Routing Distance Comparison*

| | Number of hops | | Euclidean length | |
|---|---|---|---|---|
| For the *i*-th path: | MAP: $h_i$ | GPSR: $H_i$ | MAP: $\ell_i$ | GPSR: $L_i$ |




Blue: $\frac{1}{N}\sum_{i=1}^{N} h_i/H_i$ Red: $\frac{\sum_i h_i}{\sum_i H_i}$

Blue: $\frac{1}{N}\sum_{i=1}^{N} \ell_i/L_i$ Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$

# Simulation Examples

*So far, we have shown that MAP has better load balancing than geographical forwarding, and very similar routing distance (in terms of both hops and length) ……*

*What's more, MAP is very robust to network models. It does not require the network to be a unit disk graph.*

# 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

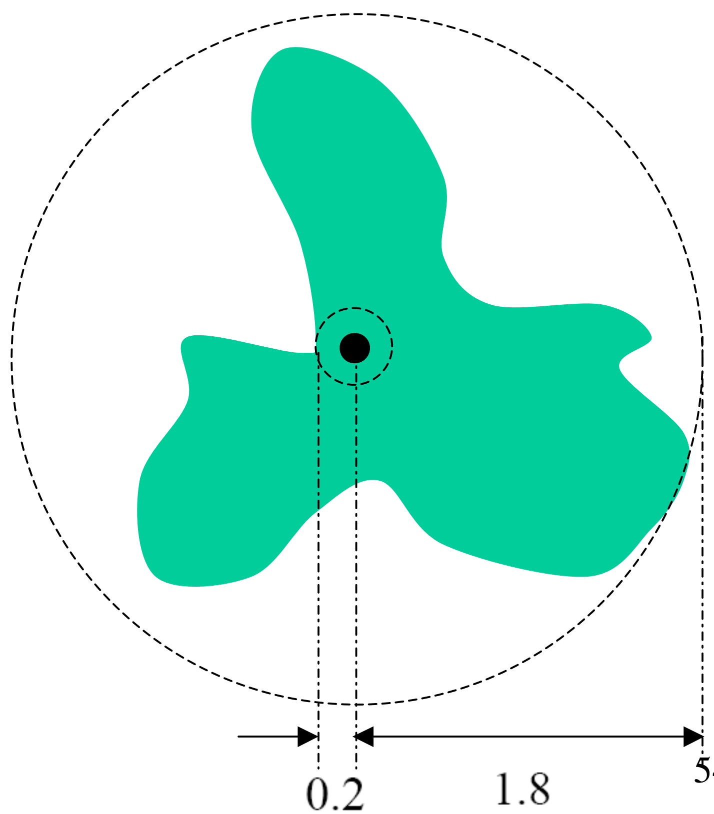
**Test MAP on networks modeled by quasi unit disk graphs**

Example: $\alpha = 0.8$

Maximum coverage range: $1.8$

Minimum coverage range: $0.2$

An example coverage area of a node:

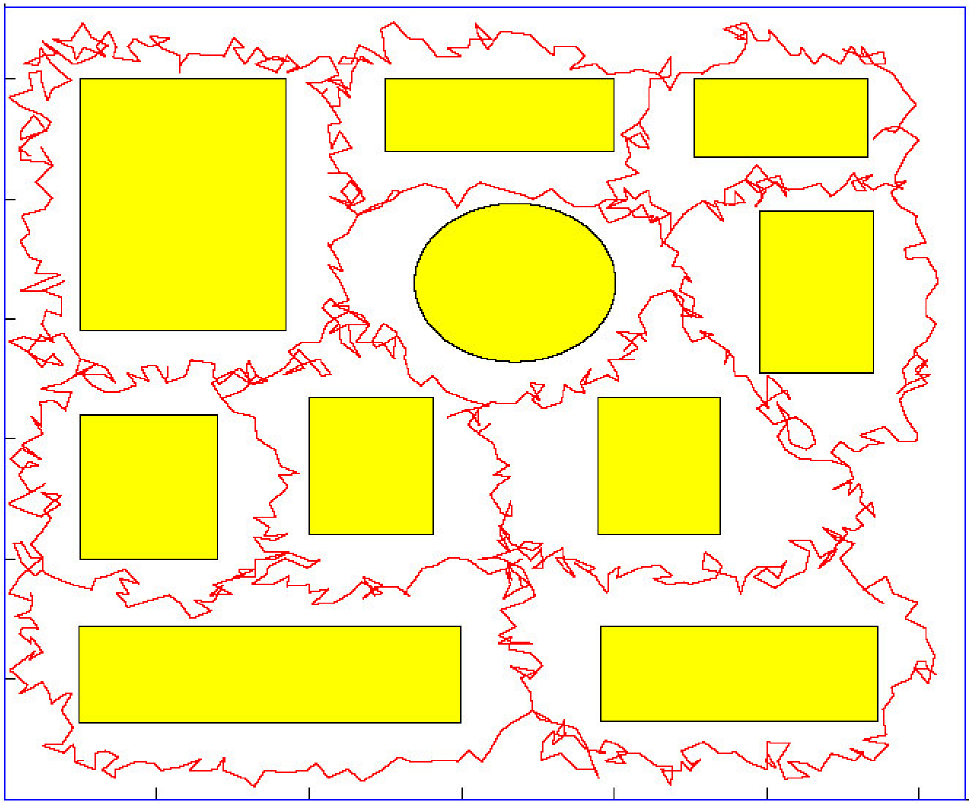


# Simulation Examples

**Test MAP on networks modeled by quasi unit disk graphs**

Example: $\alpha = 0.8$

Medial Axis (for campus):

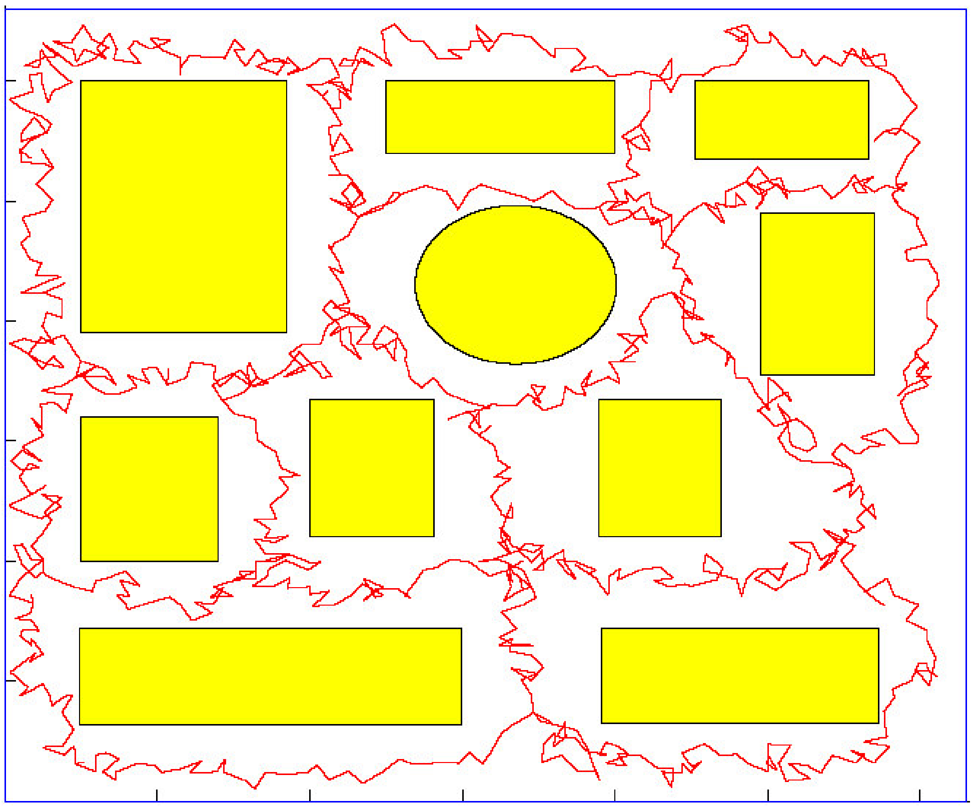


# Simulation Examples

**Test MAP on networks modeled by quasi unit disk graphs**

Example: $\alpha = 0.8$

Medial Axis
(for campus):



Although the network is very different from unit disk graph, the construction of medial axis is very robust.

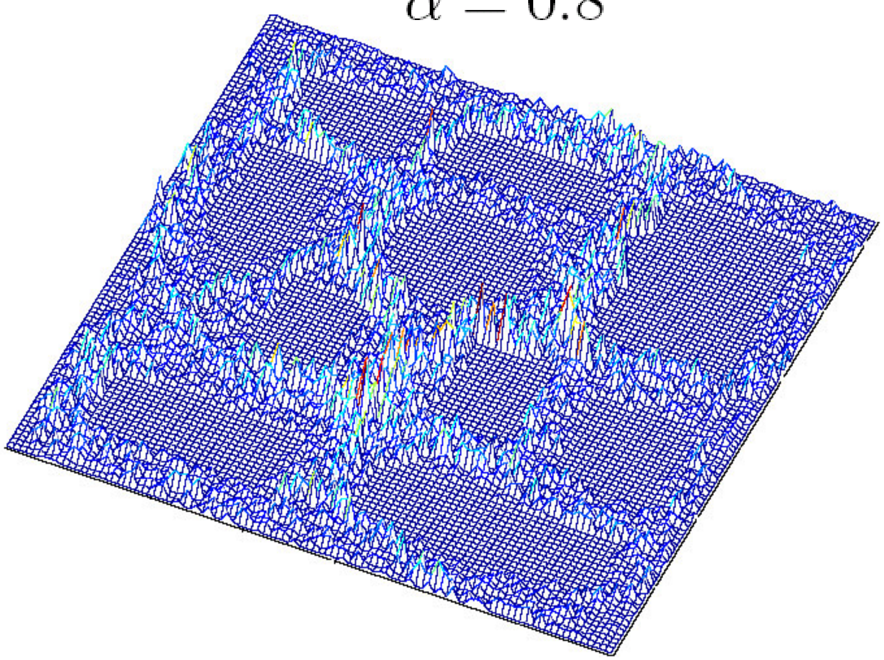
# Simulation Examples

**Test MAP on networks modeled by quasi unit disk graphs**

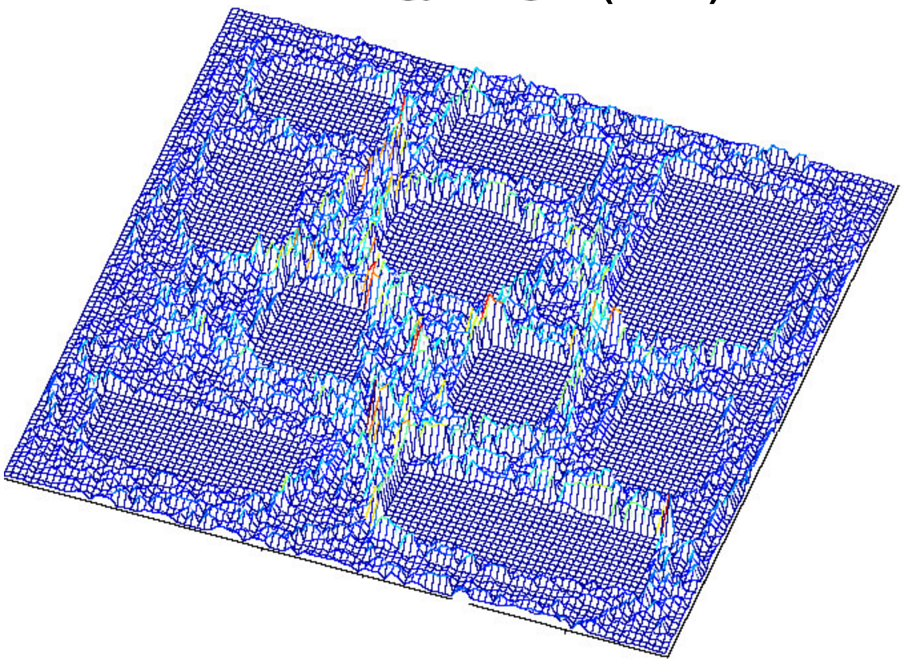
Example: $\alpha = 0.8$

**Compare MAP Load: both well balanced**

$\alpha = 0.8$



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



# Simulation Examples

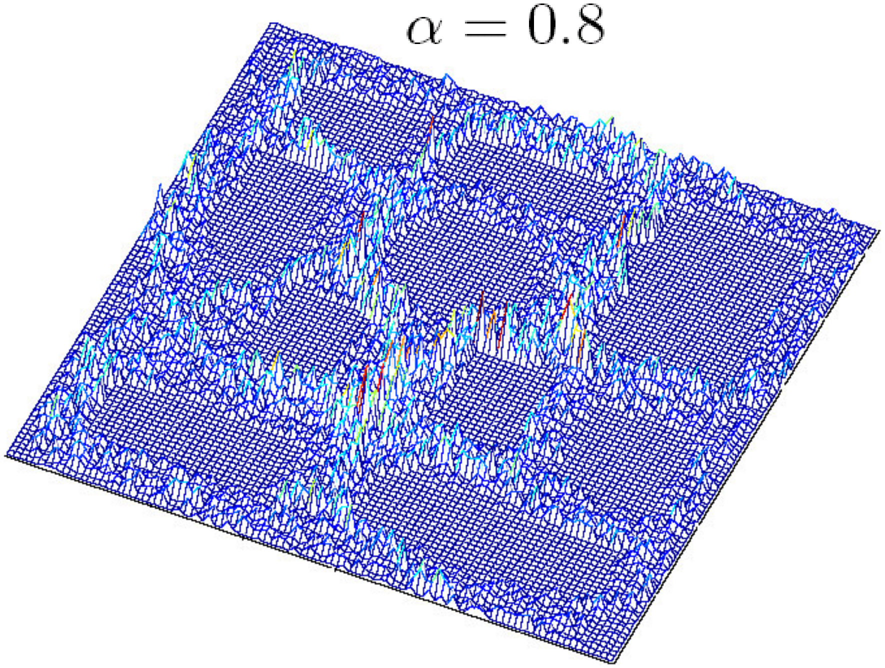
**Test MAP on networks modeled by quasi unit disk graphs**

Example: $\alpha = 0.8$

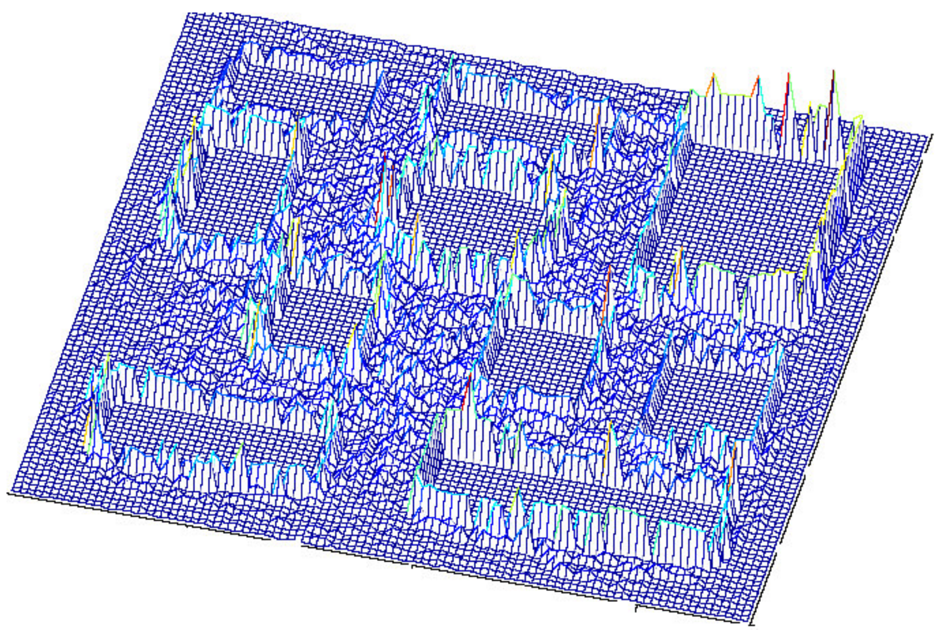
**Compare Load Balance:**

**MAP: Well Balanced**

$\alpha = 0.8$



**GPSR (geographical forwarding): Unbalanced**

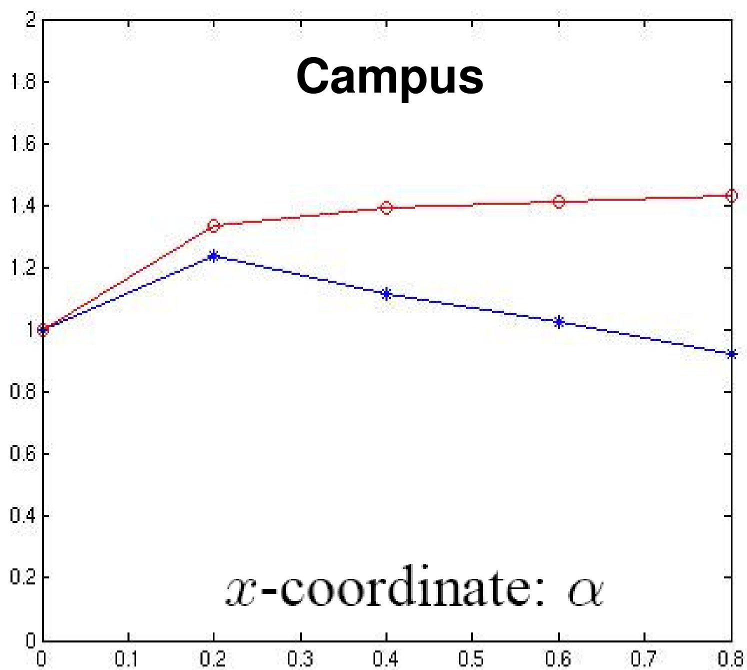


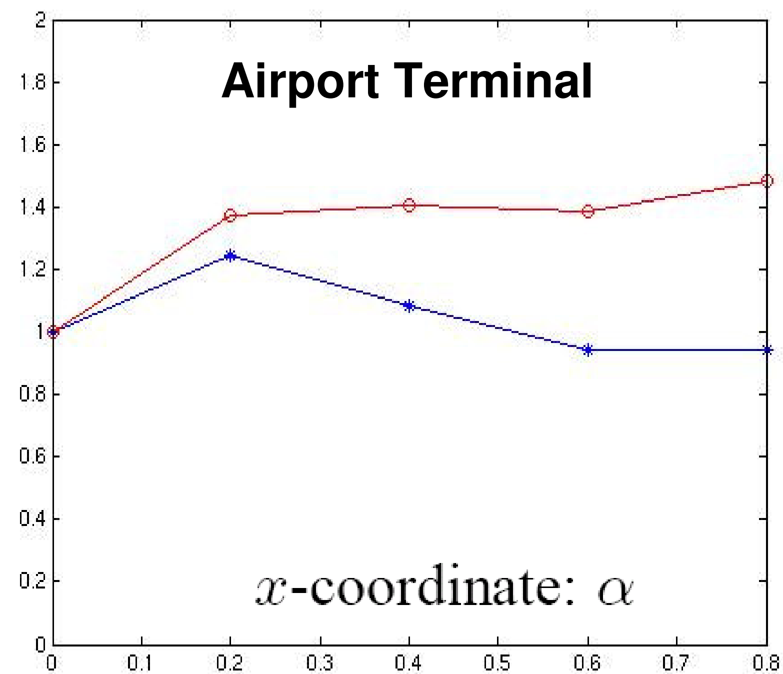
# Simulation Examples

**MAP on quasi unit disk graphs: Routing Distance**

Number of hops / Euclidean length

For the *i*-th path: Quasi-UDG: $h_i$ UDG: $H_i$ Quasi-UDG: $\ell_i$ UDG: $L_i$

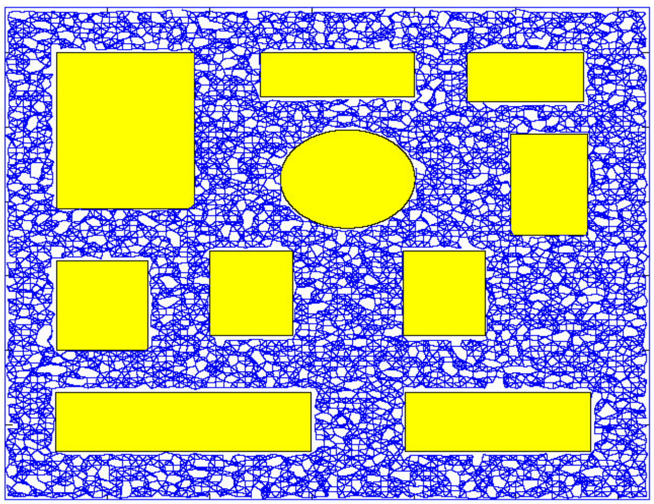


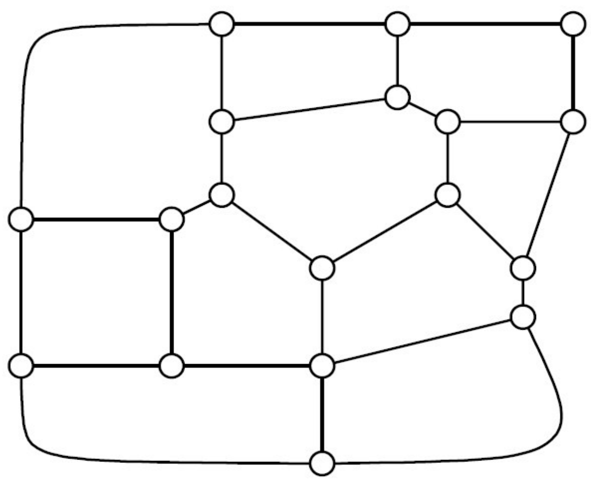

Blue: $\frac{\sum_i h_i}{\sum_i H_i}$ Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$ Blue: $\frac{\sum_i h_i}{\sum_i H_i}$ Red: $\frac{\sum_i \ell_i}{\sum_i L_i}$

# Summary of MAP

1. *Medial axis captures the shape of the sensor field.*
2. *It is compact.*
3. *No location-information is needed.*
4. *No Unit disk graph assumption.*
5. *Good load balancing.*



*Sensor field*



*Medial axis graph*