Greedy Routing with Guaranteed Delivery Using Ricci Flow

Jie Gao

Stony Brook University

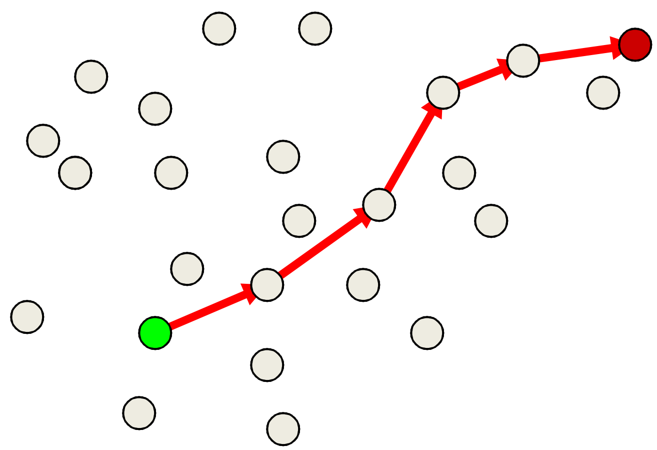
Joint work with

Rik Sarkar, Xiaotian Yin, Wei Zeng,

Feng Luo, Xianfeng David Gu

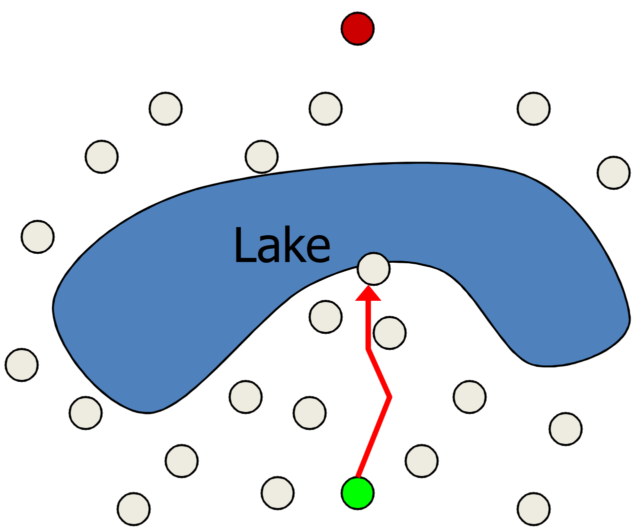
Greedy Routing

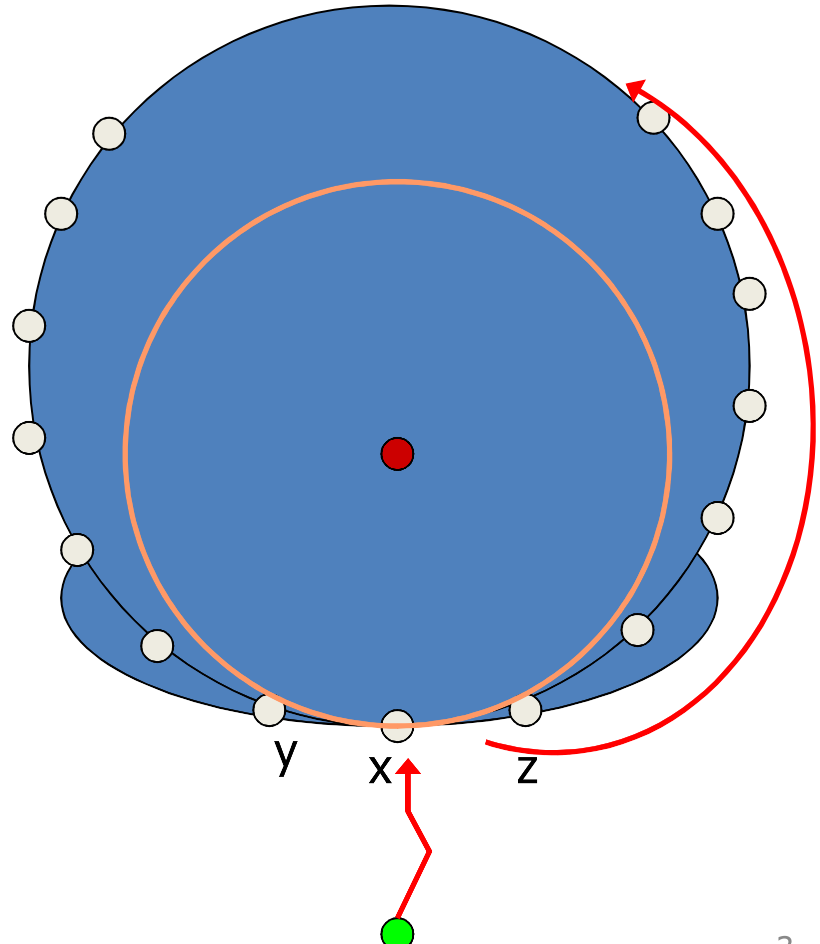
- Assign coordinates to nodes
- Message moves to neighbor closest to destination
- Simple, compact, scalable



Problem I: Greedy routing may get stuck

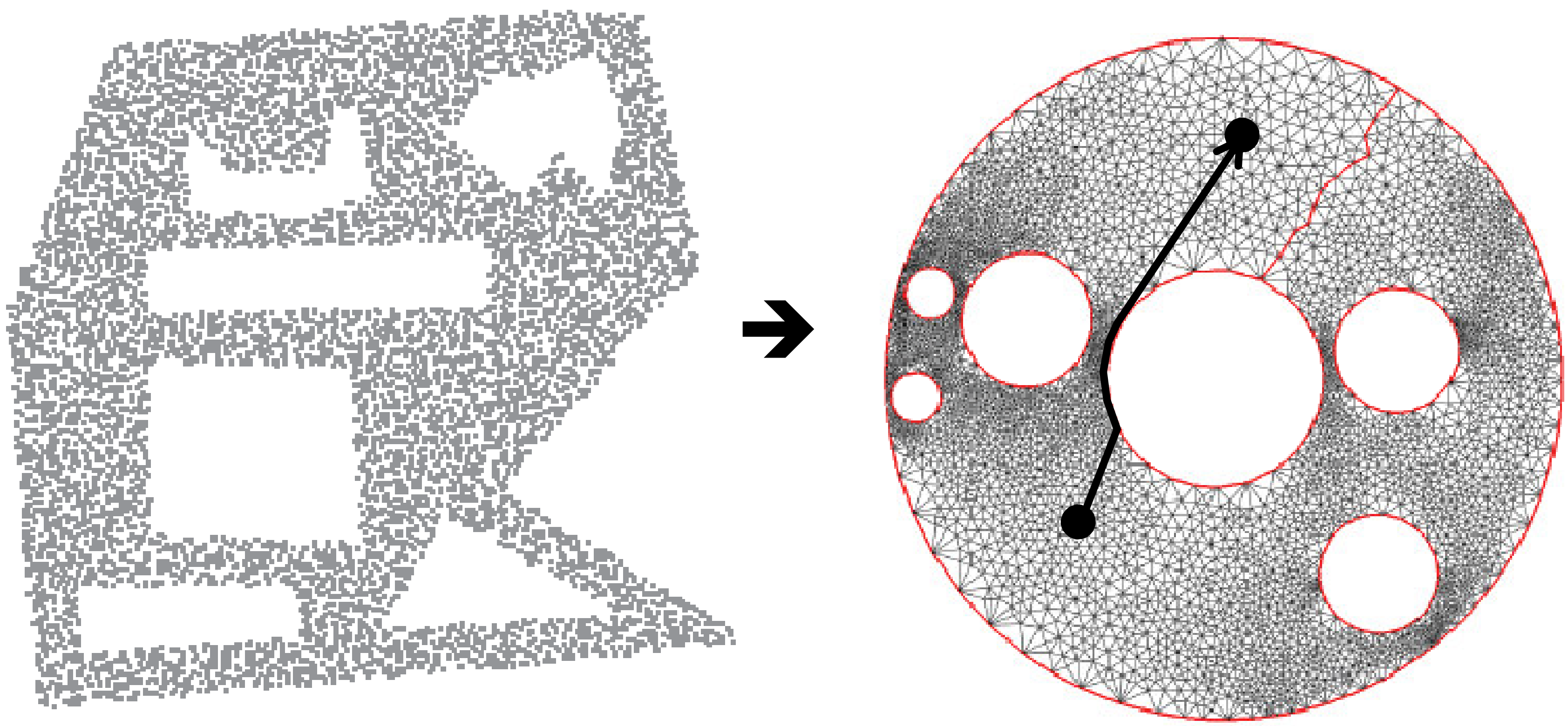
- Can get stuck





Solution: Use Ricci flow to make all holes circular

- Greedy routing does not get stuck at holes.



Talk Overview

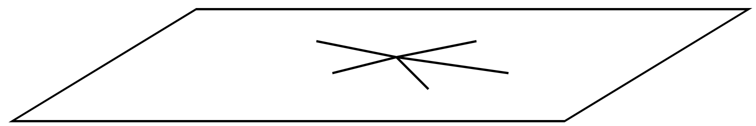
1. Theory on Ricci flow -- smoothing out surface curvature

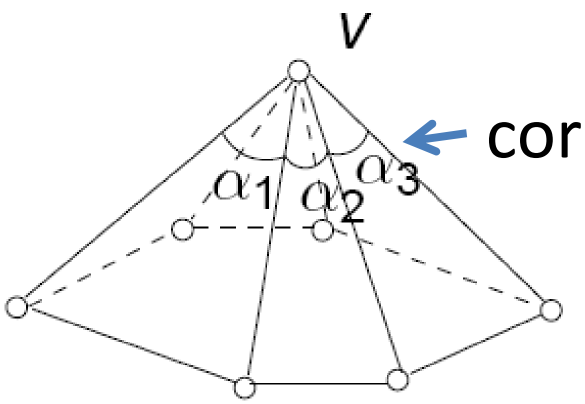
2. Distributed algorithm on sensor network

3. Addressing other issues in routing

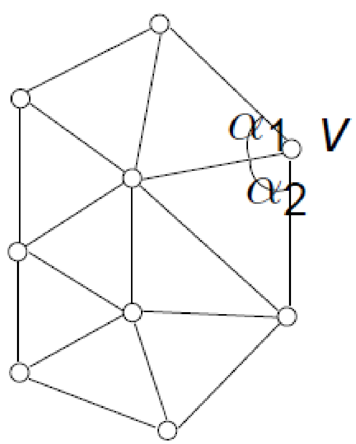
Part I: Ricci Flow, Discrete Curvature in 2D Triangulated Surface

Discrete Curvature in 2D Triangulated Surface





corner angle



For an interior vertex

Deviation from the plane

$$\kappa(v) = 2\pi - \sum_i \alpha_i$$

For a vertex on the boundary

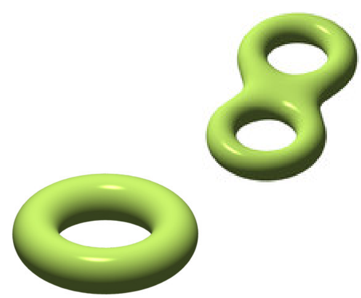
Deviation from straight line

$$\kappa(v) = \pi - \sum_i \alpha_i$$

Total Discrete Curvature

- Gauss Bonnet Theorem: total curvature of a surface M is a topological invariant:

$$\sum_{v_i \in V} K_i = 2\pi\chi(M)$$



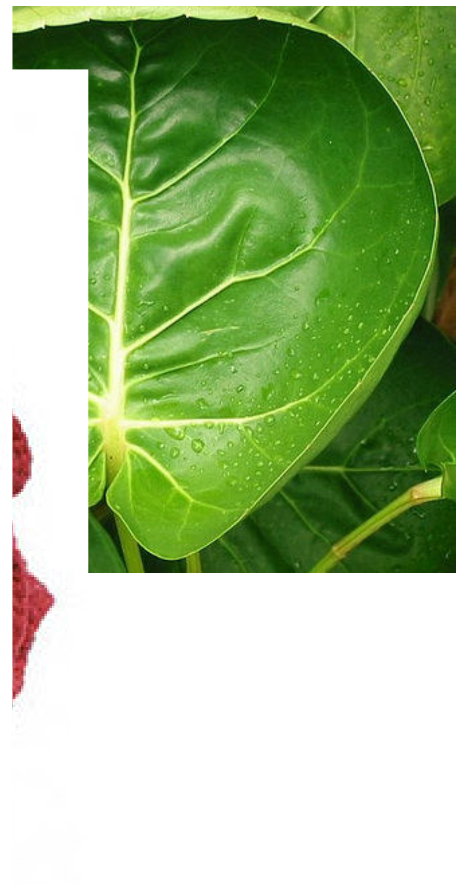
Euler Characteristic: 2 – 2(# handles) – (# holes)

- Ricci flow: diffuse uneven curvatures to be uniform

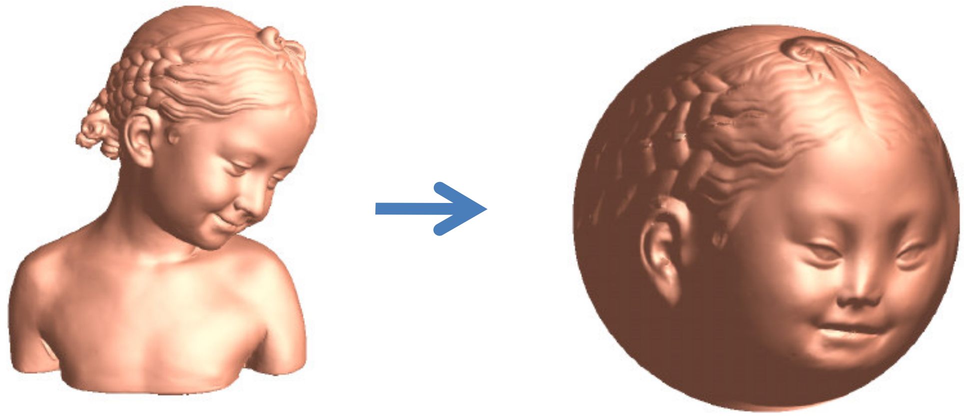
What they look like

Negative Curvature

Positive Curvature



Ricci Flow: diffuse curvature

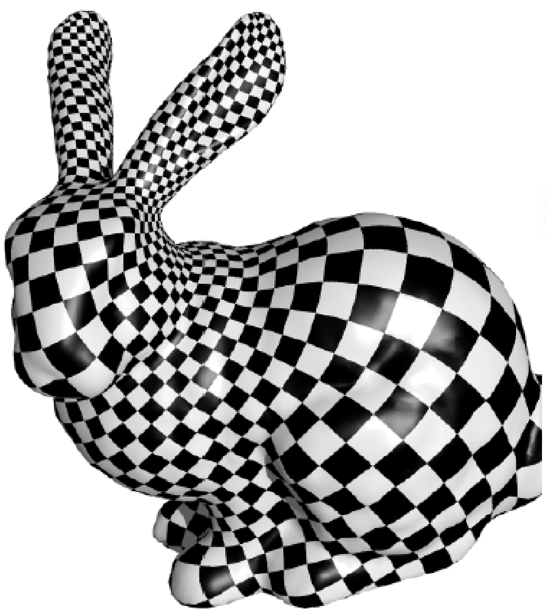


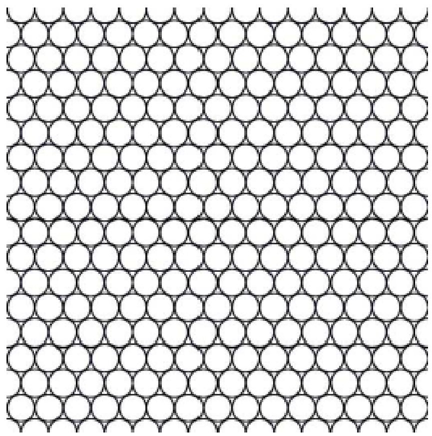
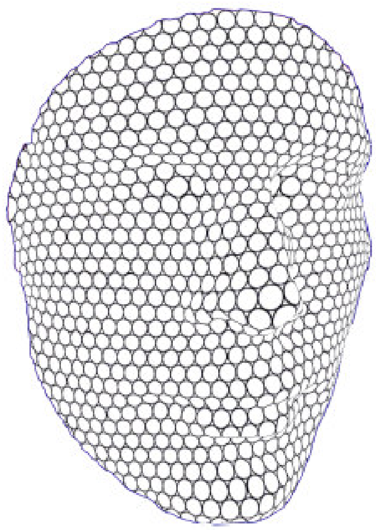
- Riemannian metric g on M: curve length
- We modify g by curvature $$\frac{dg_{ij}(t)}{dt} = -2K(t)g_{ij}(t)$$
- Curvature evolves: $$\frac{dK(t)}{dt} = -\Delta_{\mathbf{g}(t)} K(t)$$

  Δ: Laplace operator
- Same equation as heat diffusion

Ricci Flow: diffuse curvature

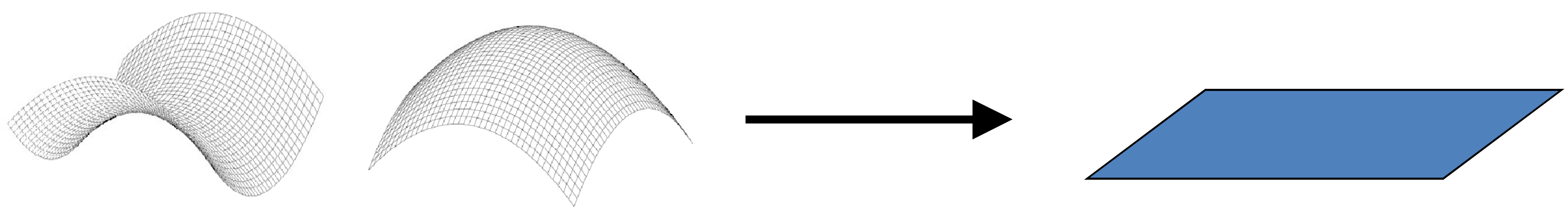
- "Ricci energy" is strictly convex ➔ unique surface with the same surface area s.t. there is constant curvature everywhere.
- Conformal map: angle preserving



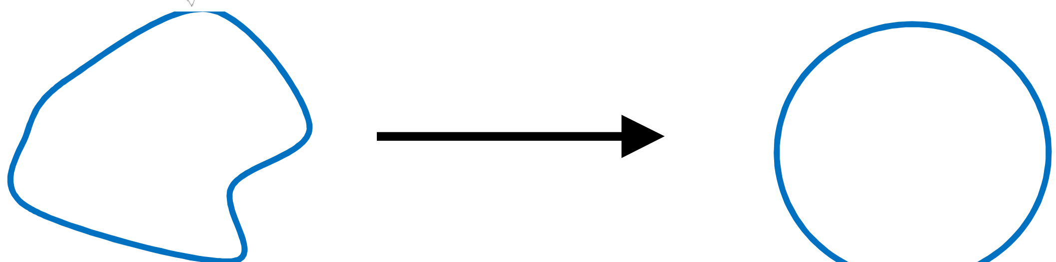



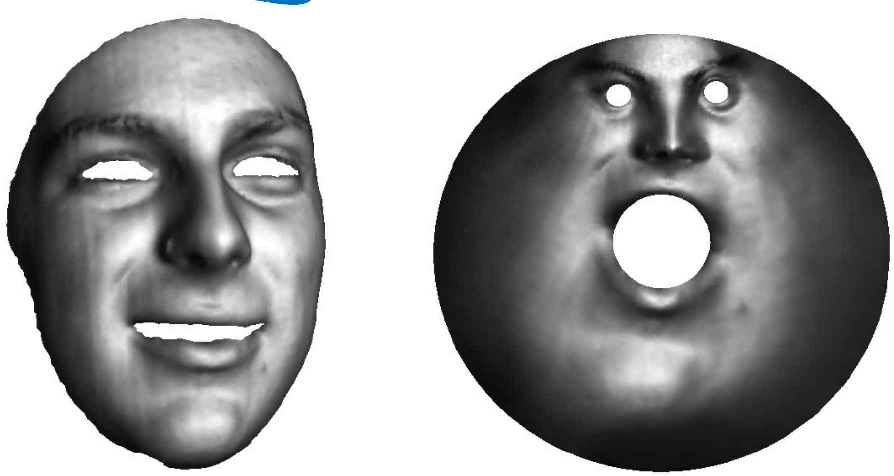
Ricci Flow in our case

- Target: a metric with pre-specified curvature



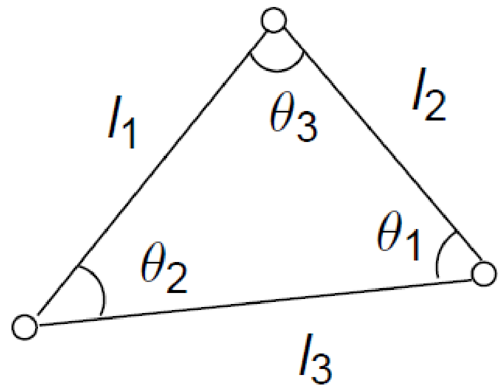
Boundary cycle





Discrete Ricci Flow

- Metric: edge length of the triangulation
  - Satisfies triangle inequality
- Metric determines the curvature

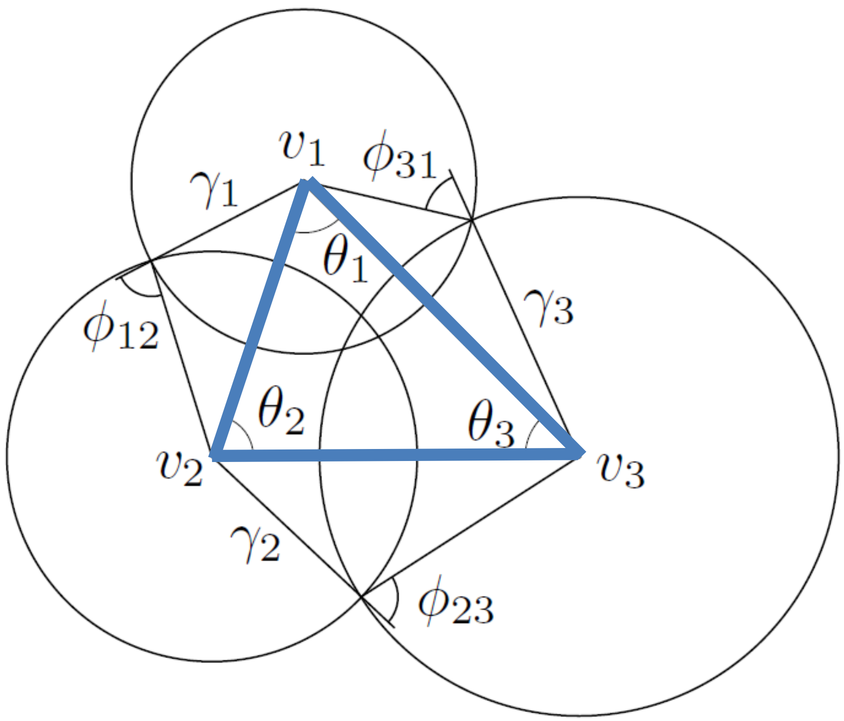


$$\cos \theta_i = \frac{l_j^2 + l_k^2 - l_i^2}{2 l_j l_k}$$

- Discrete Ricci flow: conformal map.
- What do we mean by “angles”?

Discrete Flow : Use Circle Packing Metric

- Circle packing metric: circle of radius γ_i at each vertex & intersection angle ϕ_{ij}.



With γ, ϕ one can calculate the edge length l and corner angle θ, by cosine law

Discrete Ricci flow modifies γ, preserves ϕ

Remark: no need of an embedding (vertex location)

Discrete Flow : Use Circle Packing Metric

- Circle packing metric: circle of radius γ_i at each vertex & intersection angle ϕ_{ij}.
- Take $u_i = \log \gamma_i$
- Discrete Ricci flow:

$$\frac{du_i(t)}{dt} = (\underbrace{\bar{K}_i}_{\text{Target curvature}} - \underbrace{K_i}_{\text{Current curvature}})$$

Background

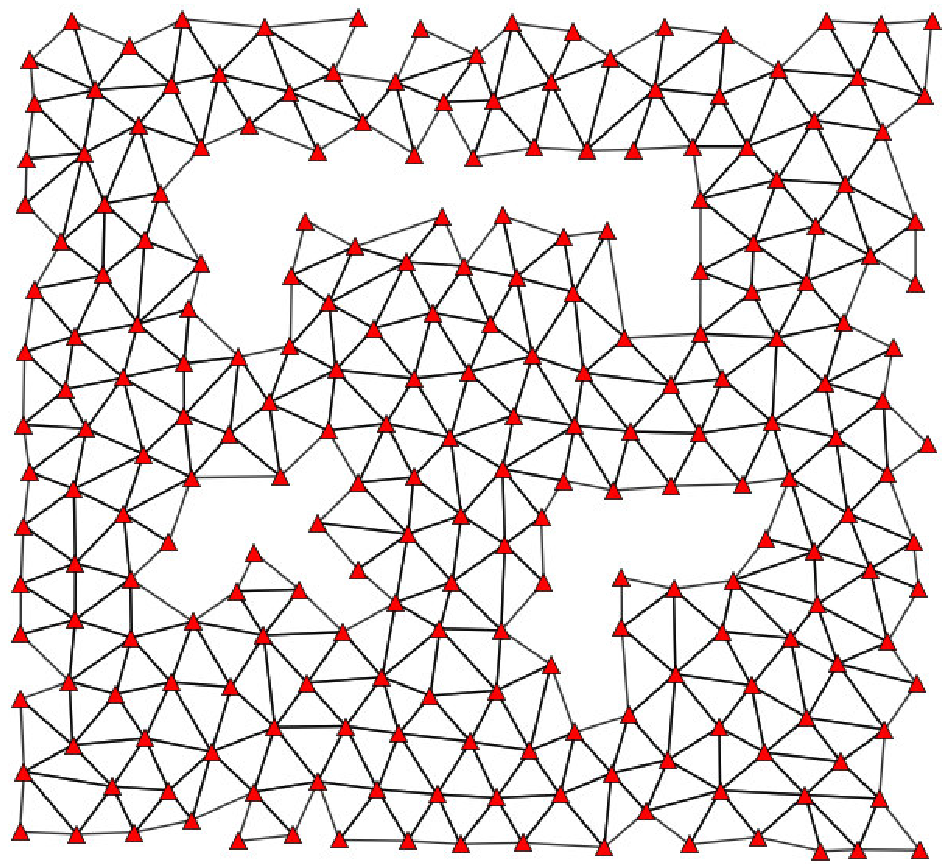
- [Hamilton 82, Chow 91] : Smooth curvature flow flattens the metric
- [Thurston 85, Sullivan and Rodin 87] : Circle packing – discrete conformal maps
- [He and Schramm 93, 96] : Discrete flow with non-uniform triangulations
- [Chow and Luo 03] : Discrete flow, existence of solutions, criteria, fast convergence

Part II: Ricci Flow in Sensor Networks

- Build a triangulation from network graph
  - Requirement: triangulation of a 2D manifold

• With node location: patch together the “local” Delaunay triangulations

• Without locations: use landmark-based combinatorial Delaunay graph



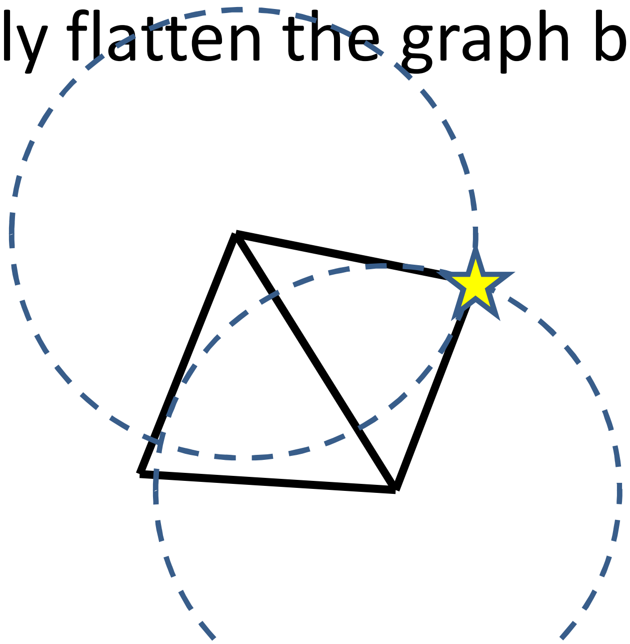
Ricci Flow in Sensor Networks

- All edge length = **1** initially
  - Introduces curvature in the surface
- Use Ricci flow to reach target curvature κ'
  - Take tangent circle packing metric: $\gamma=1/2$, $\phi=0$ initially
  - Interior nodes: target curvature = 0
  - Nodes on boundary C: target curvature = $-2\pi/|C|$
  - Modify $u_i = \log\gamma_i$ by $\delta\,(\kappa'-\kappa)$
  - Until the curvature difference < ε

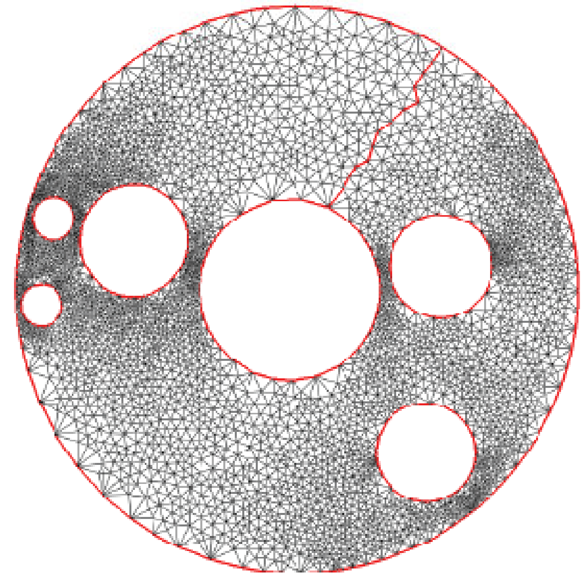
|C|: Length of the boundary cycle

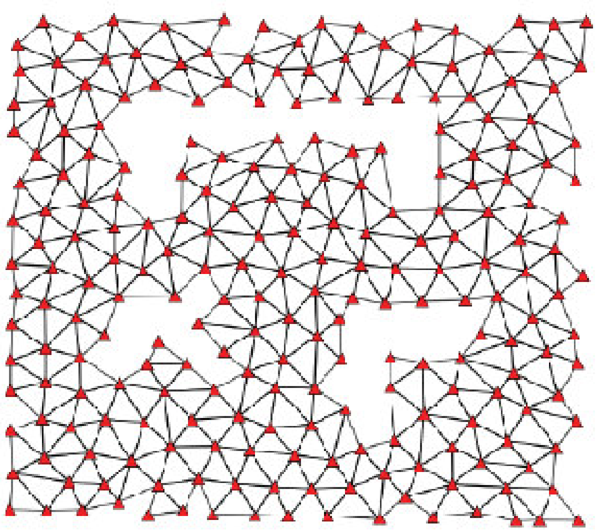
Ricci Flow in Sensor Networks

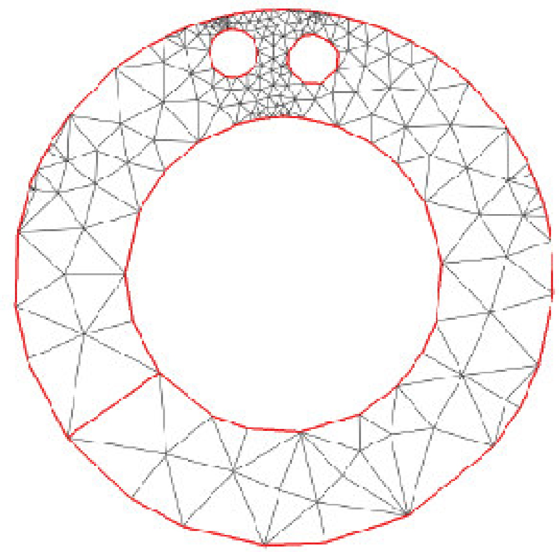
- Ricci flow is a distributed algorithm
  - Each node modifies its own circle radius.
- Compute virtual coordinates
  - Start from an arbitrary “seed” triangle
  - Iteratively flatten the graph by triangulation.



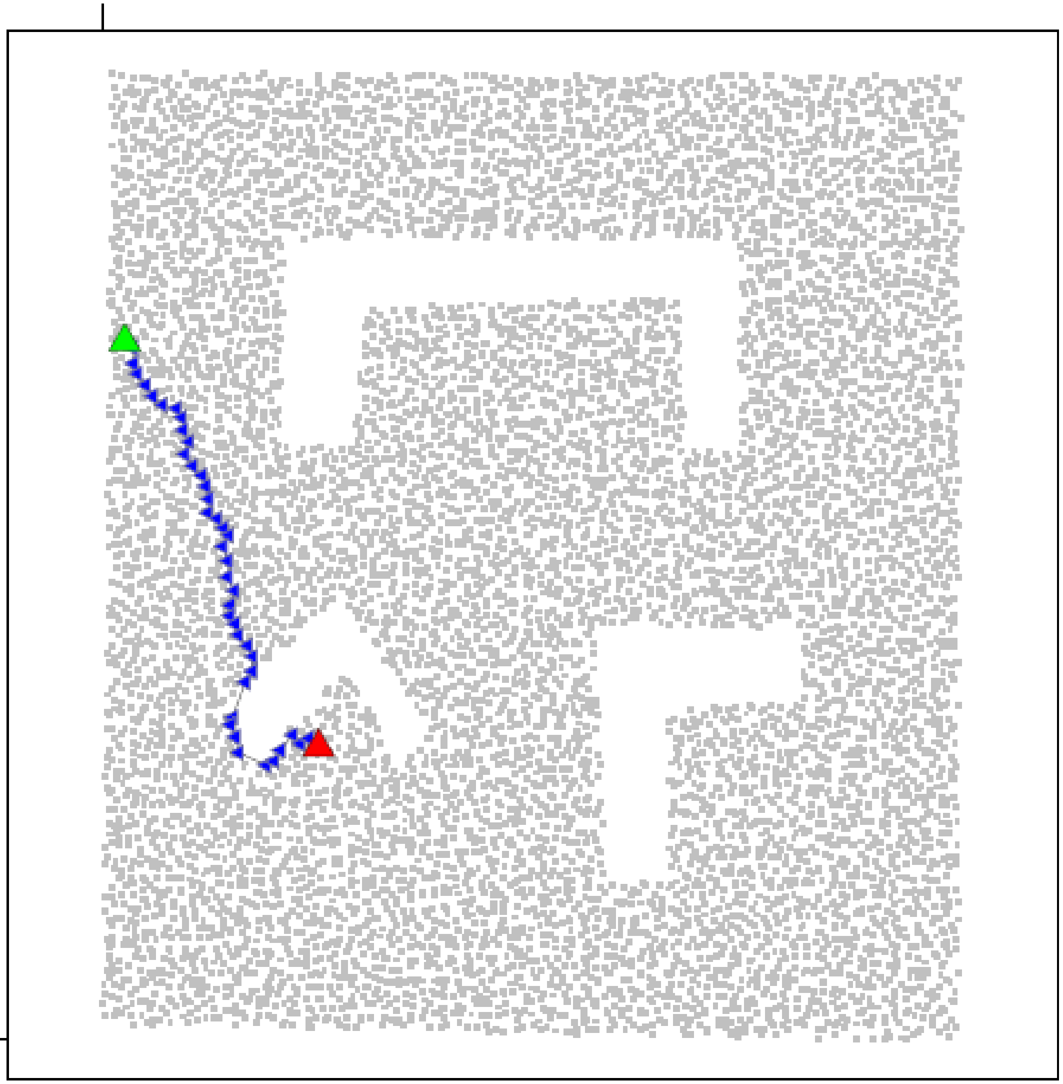
Examples







Routing

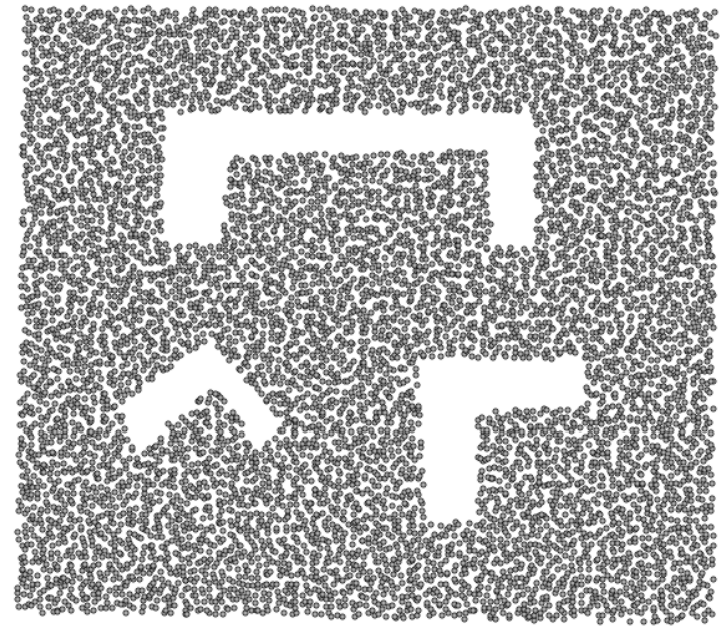


Experiments and Comparison

- NoGeo : Fix locations of boundary, replace edges by tight rubber bands : Produces convex holes

  [Rao, Papadimitriou, Shenker, Stoica – Mobicom 03]

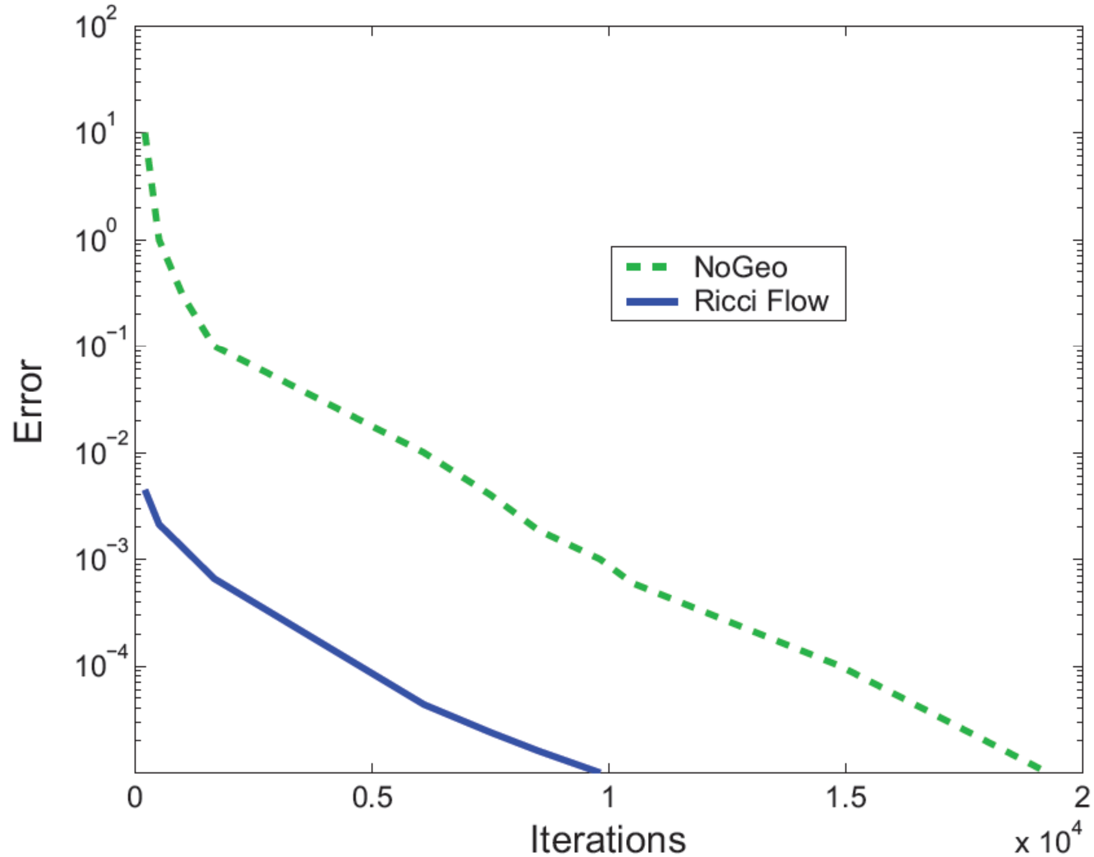
- Does not guarantee delivery: cannot handle concave holes well



Method	Delivery	Avg Path Stretch	Max Path Stretch
Ricci Flow	100%	1.59	3.21
NoGeo	83.66%	1.17	1.54

Convergence rate

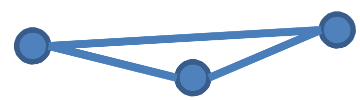
- Curvature error bound ε
- Step size δ
- # steps = O(log(1/ε)/δ)



Iterations Vs error

Theoretical guarantee of delivery

Theoretically,

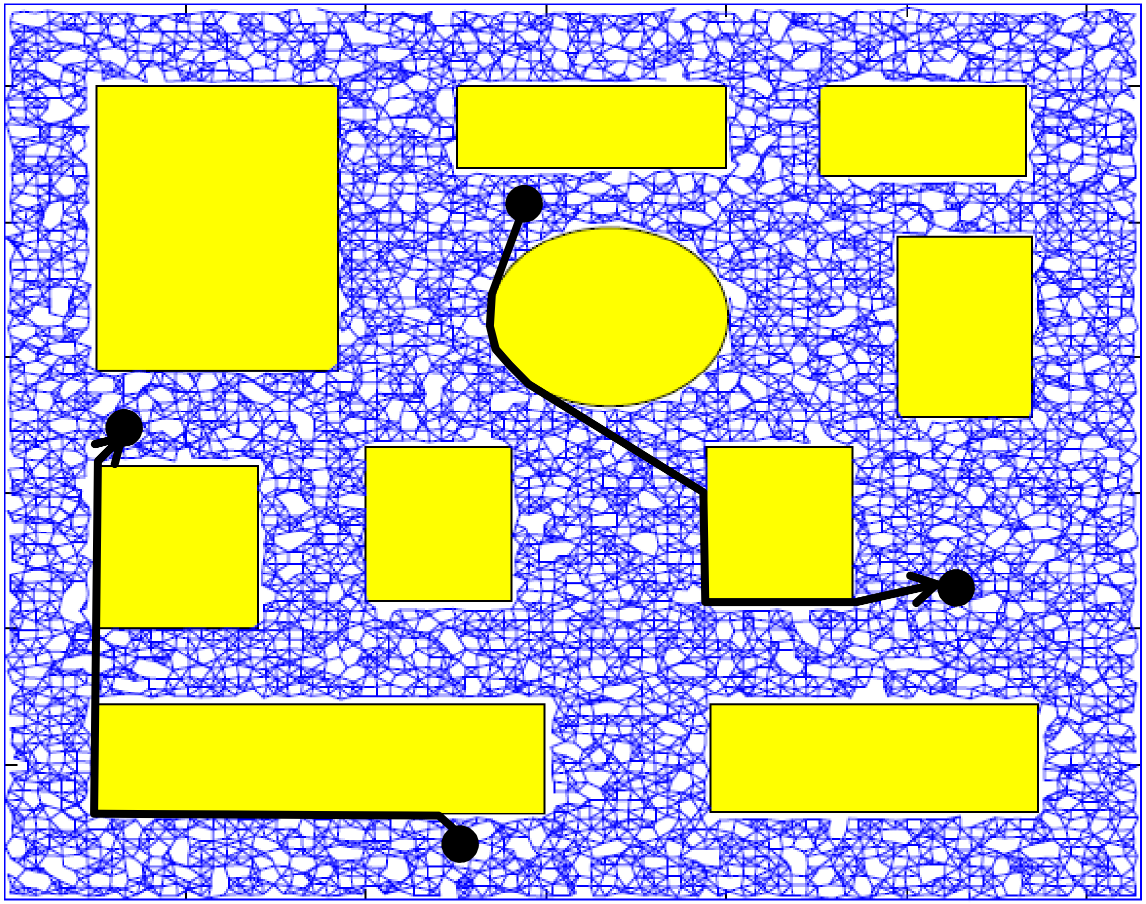
- Ricci flow is a numerical algorithm.
- Triangles can be skinny. 
- Solution: route on edges/triangles.

Outline

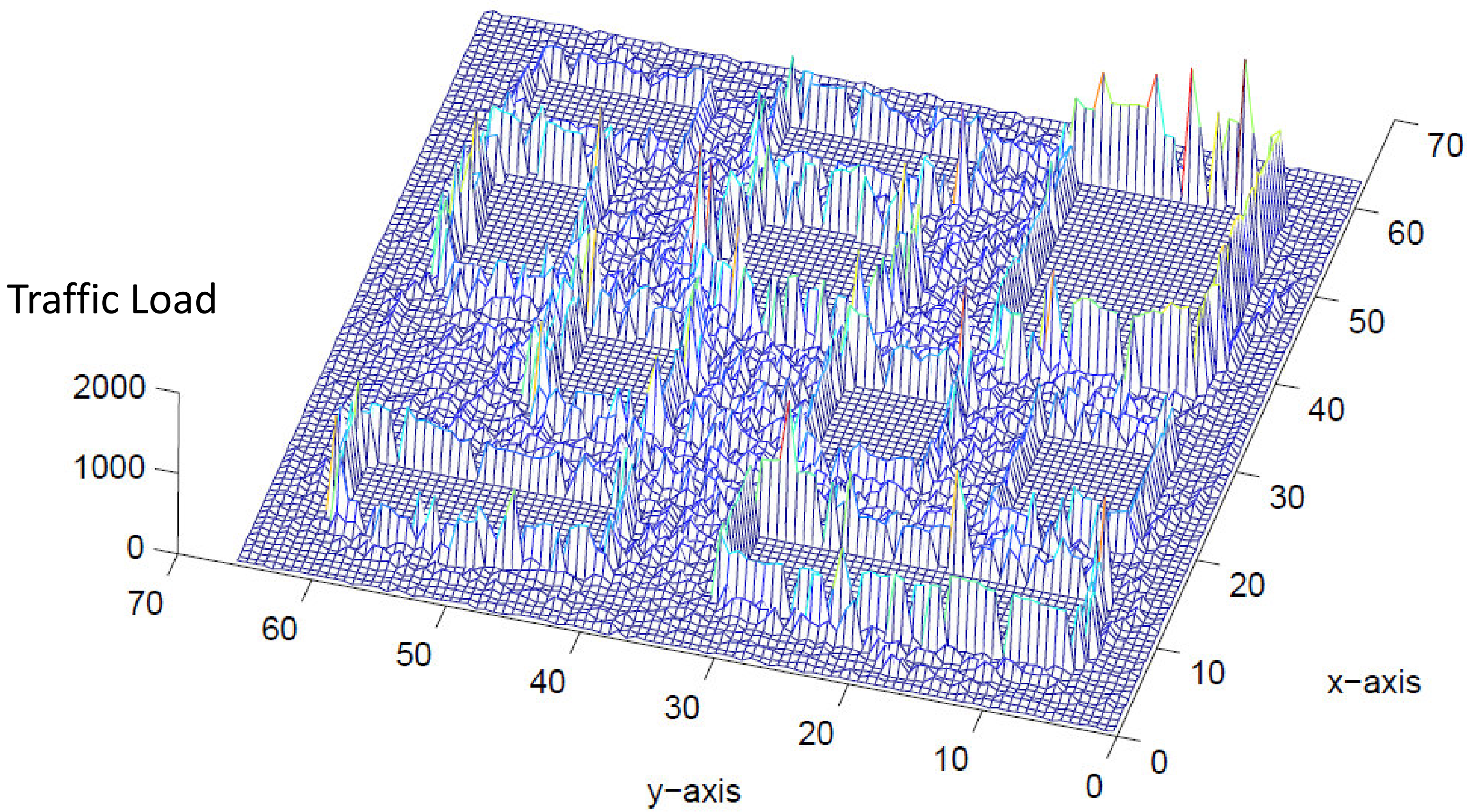
Greedy routing

1. Guaranteed delivery
2. Load balancing
3. Resilient to failures

Problem II: Boundary nodes get overloaded

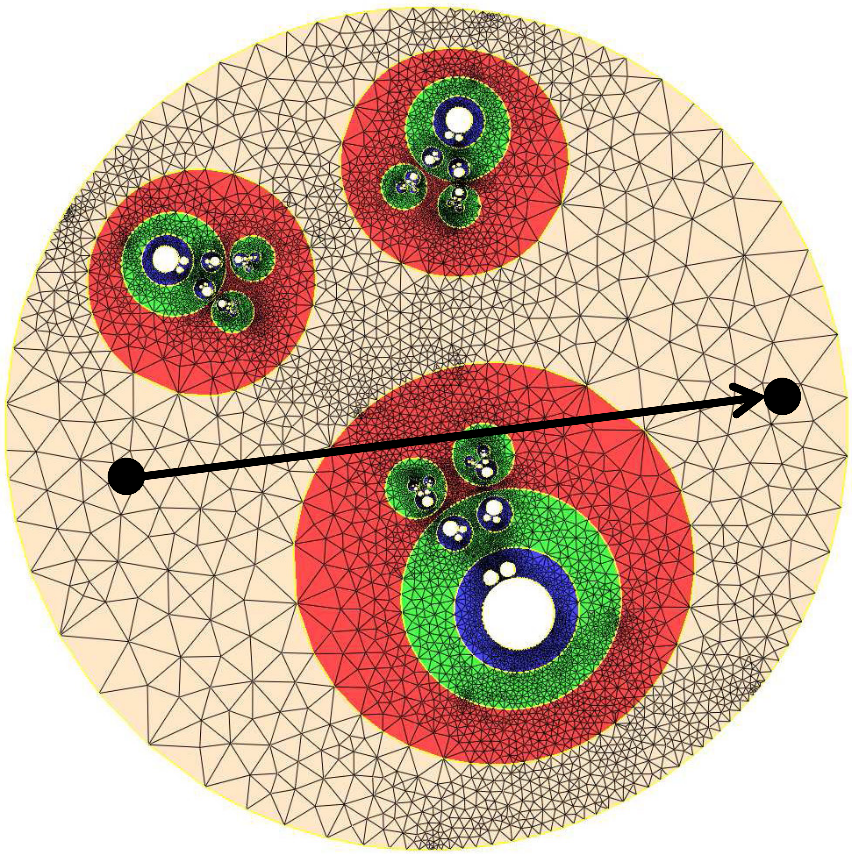


Boundary nodes get overloaded

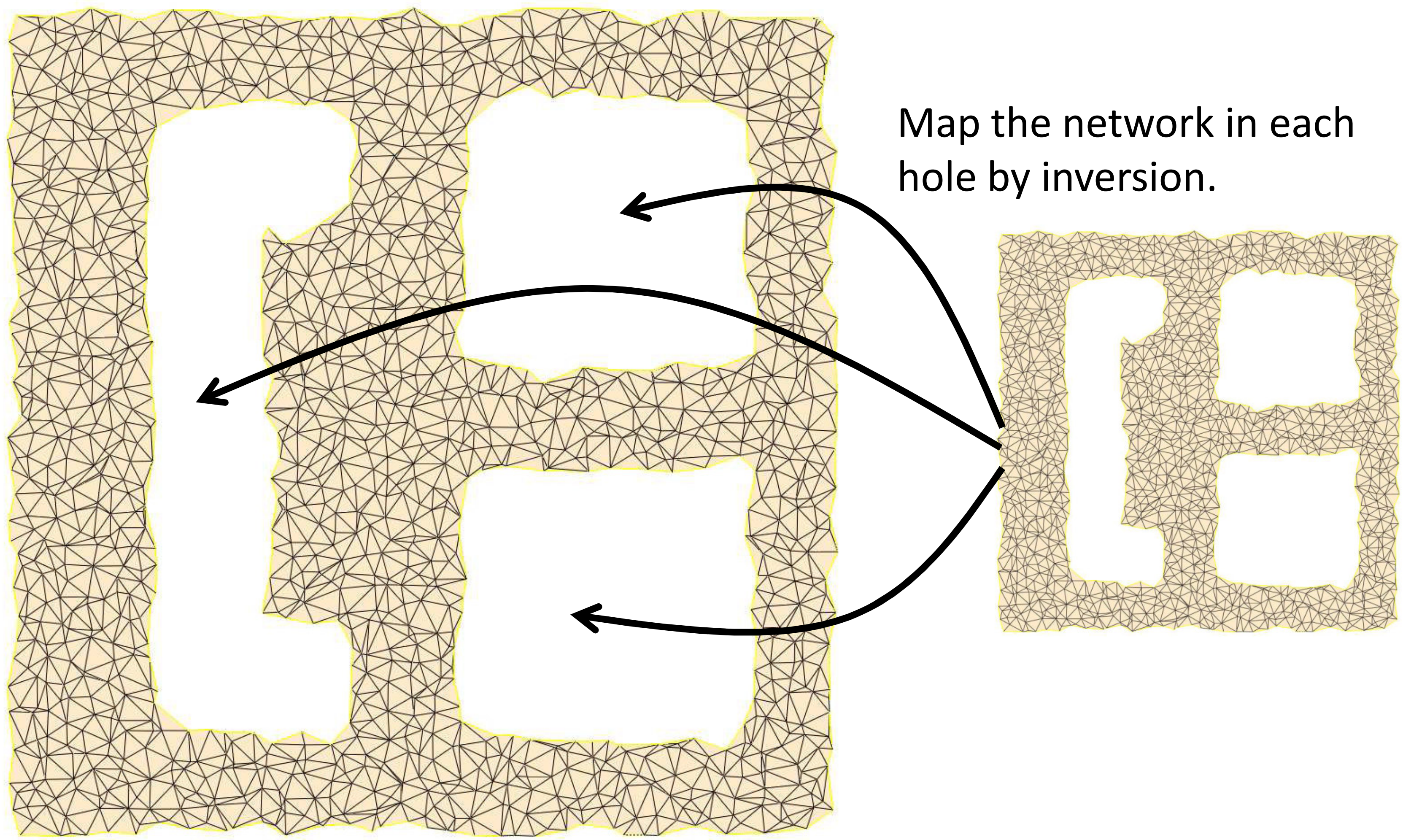


Solution II: Use covering space, tile up the domain

- Greedy routes cut through the holes and do not overload boundary.



Idea: Reflect the network to fill up the holes



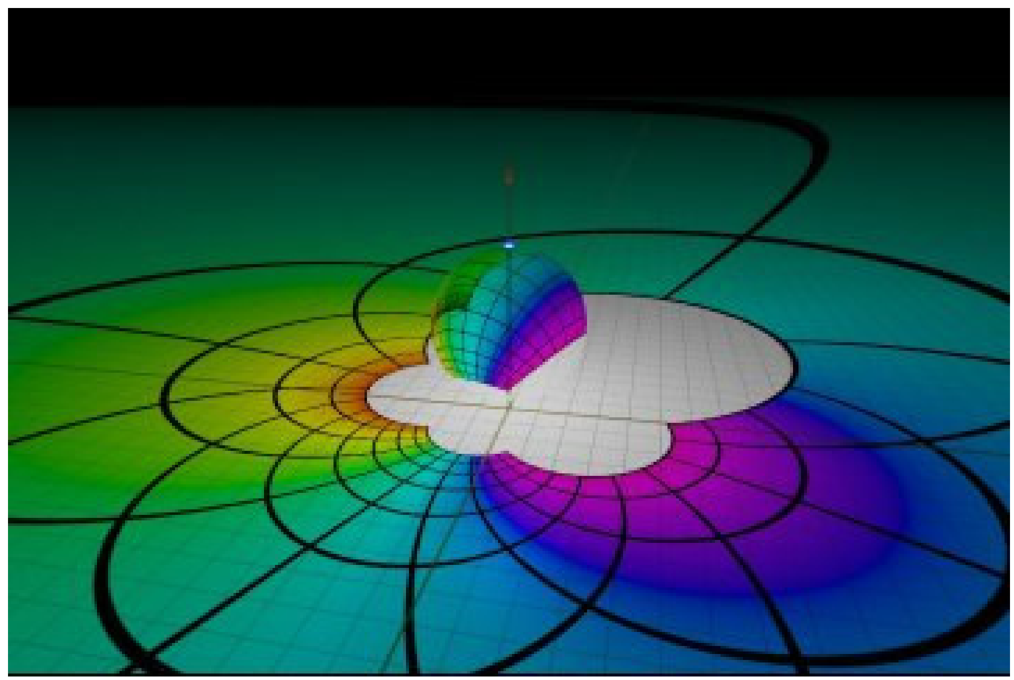
Map the network in each hole by inversion.

Möbius Transform

- Möbius transform
  - Conformal: maps circles to circles
  - Four basic elements: translation, dilation, rotation, inversions (reflection).

$$f(z) = \frac{az + b}{cz + d}$$

a, b, c, d are 4 complex numbers, ad ≠ bc



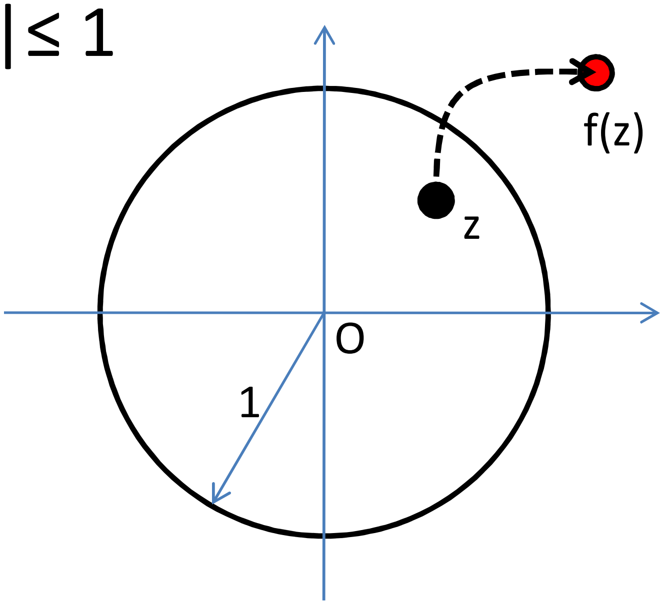
Reflections using Möbius Transform

- Map inside out

Unit circle at origin

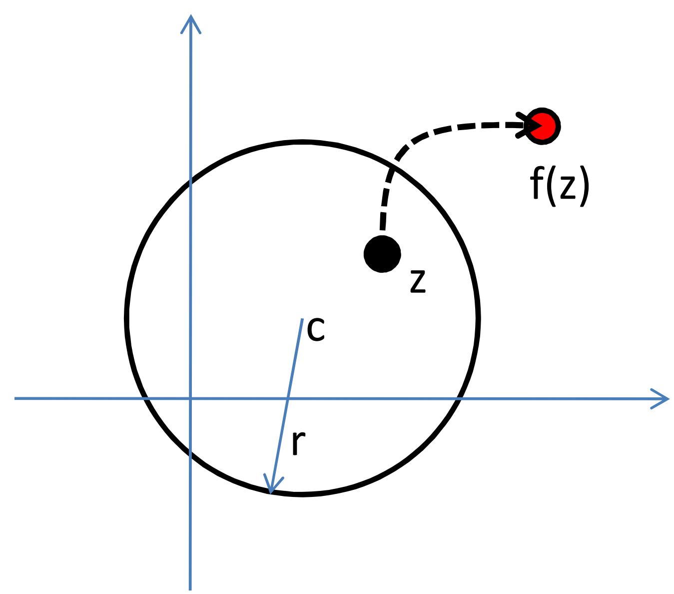
$$f(z) = \frac{1}{z}$$

$|z| \leq 1$

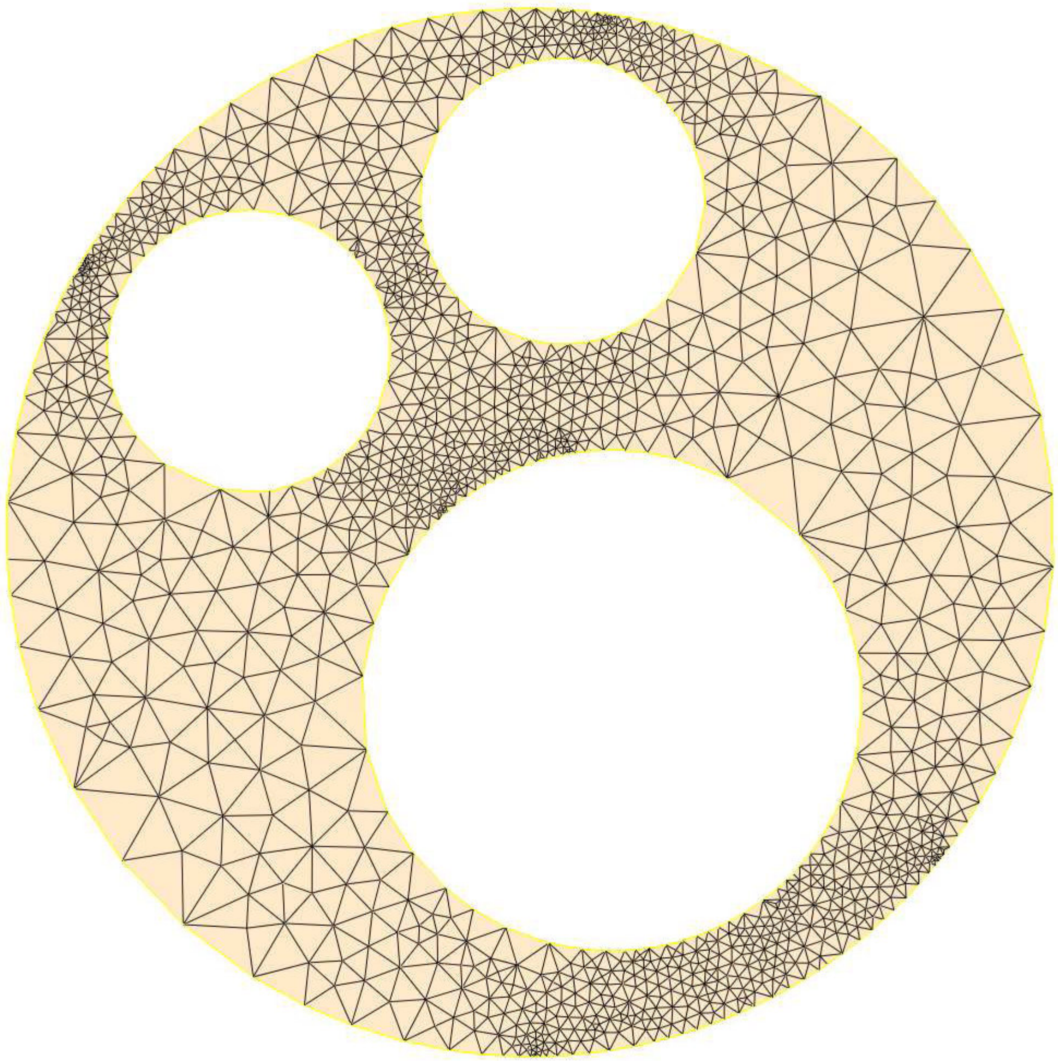


A circle at c with radius r

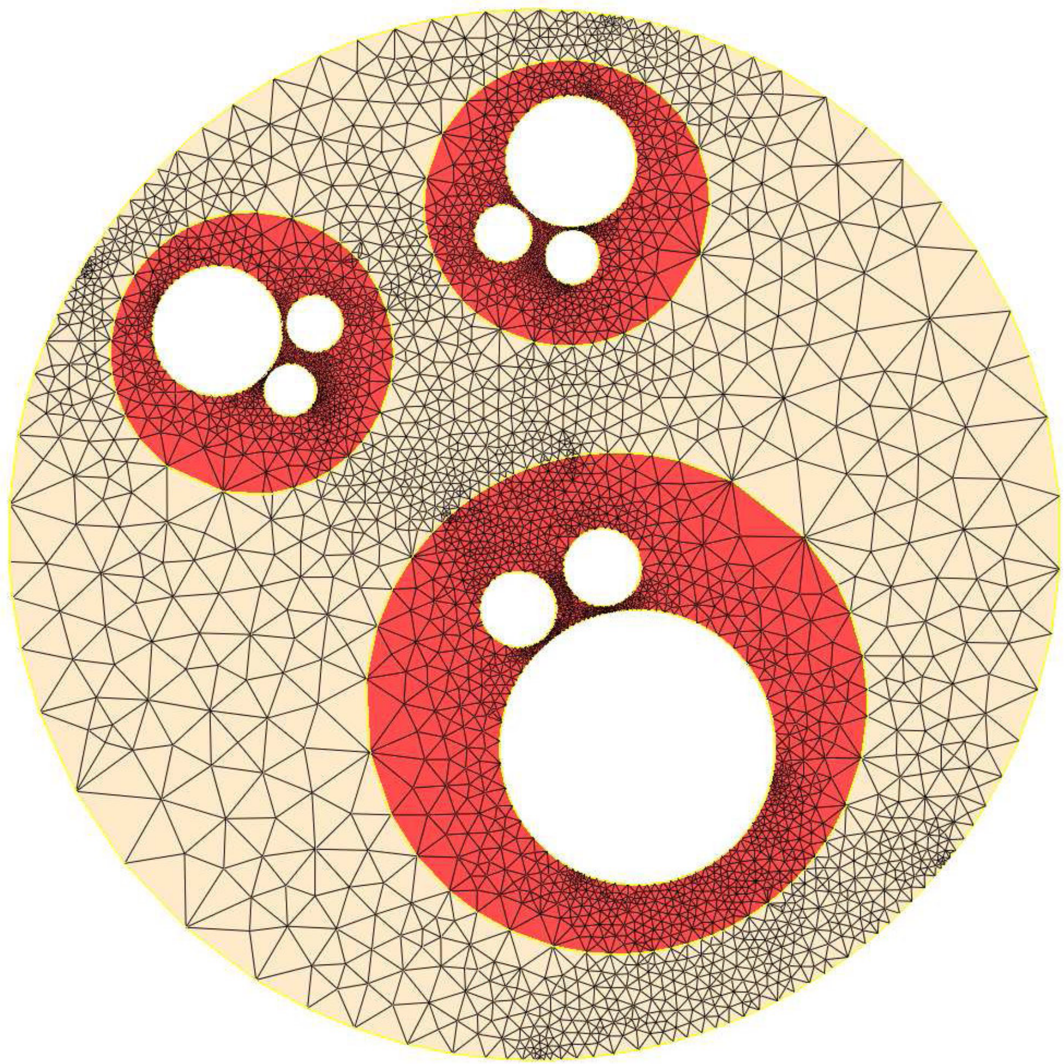
$$f(z) - c = \frac{r^2}{z - \overline{c}}$$



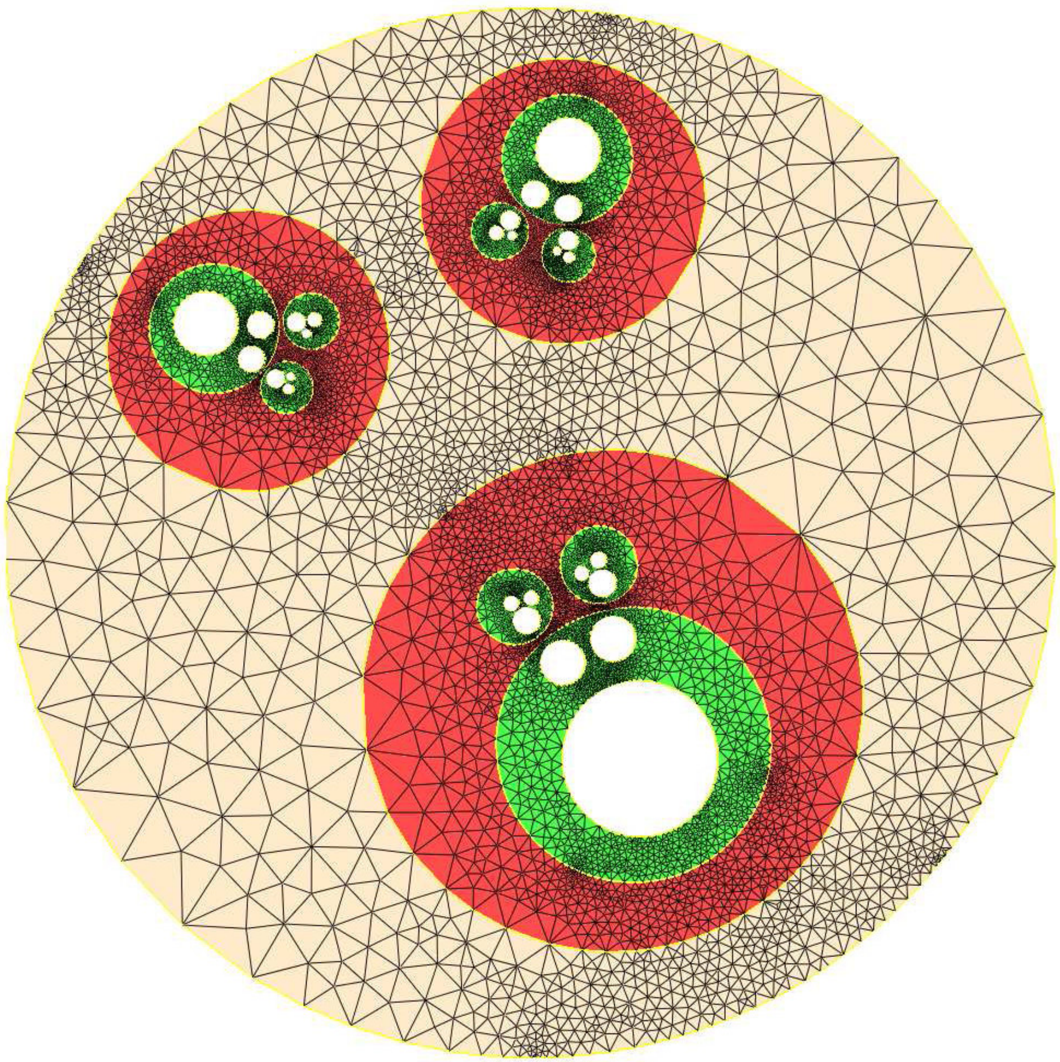
Step 1: Map All Holes Circular



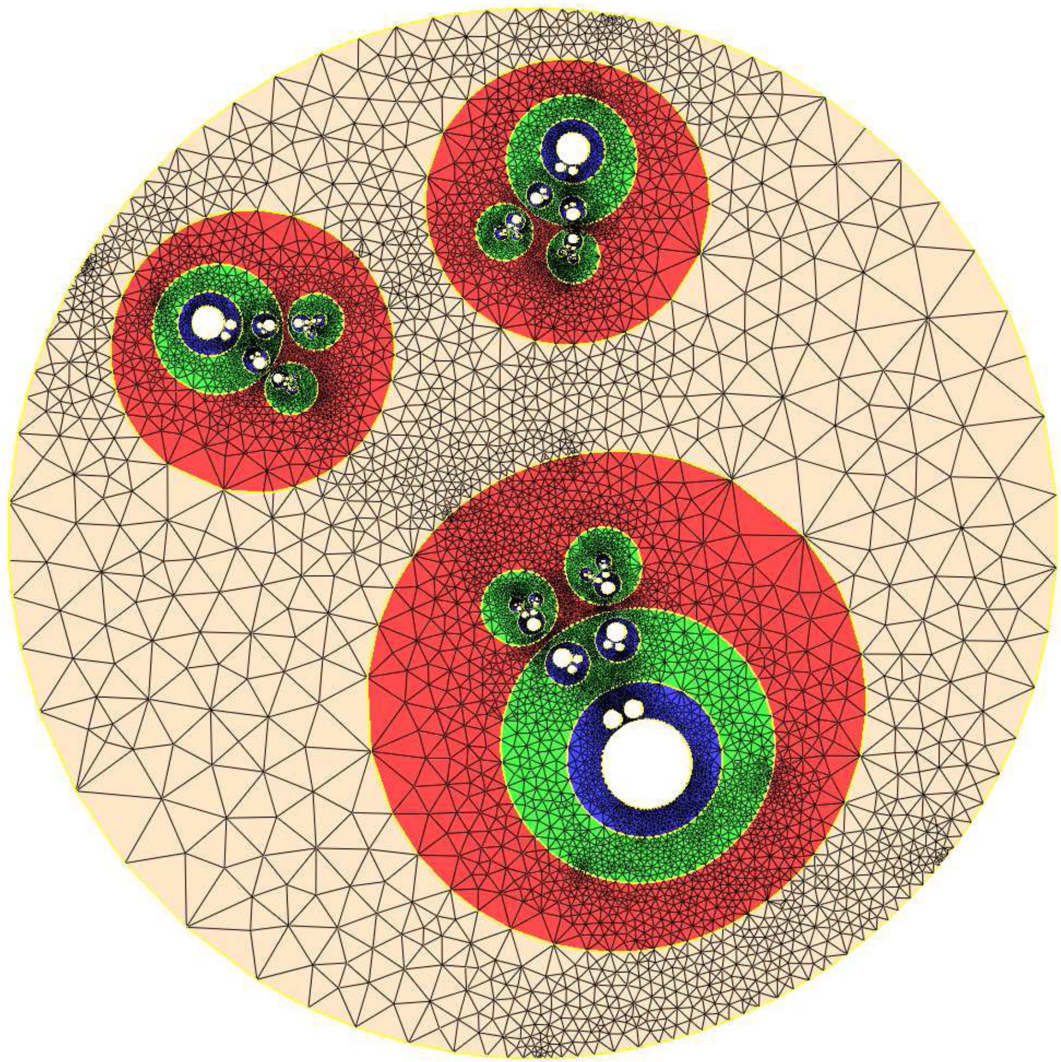
Step 2: Reflect Network for Each Hole



Step 3: Continue until all Holes are Small

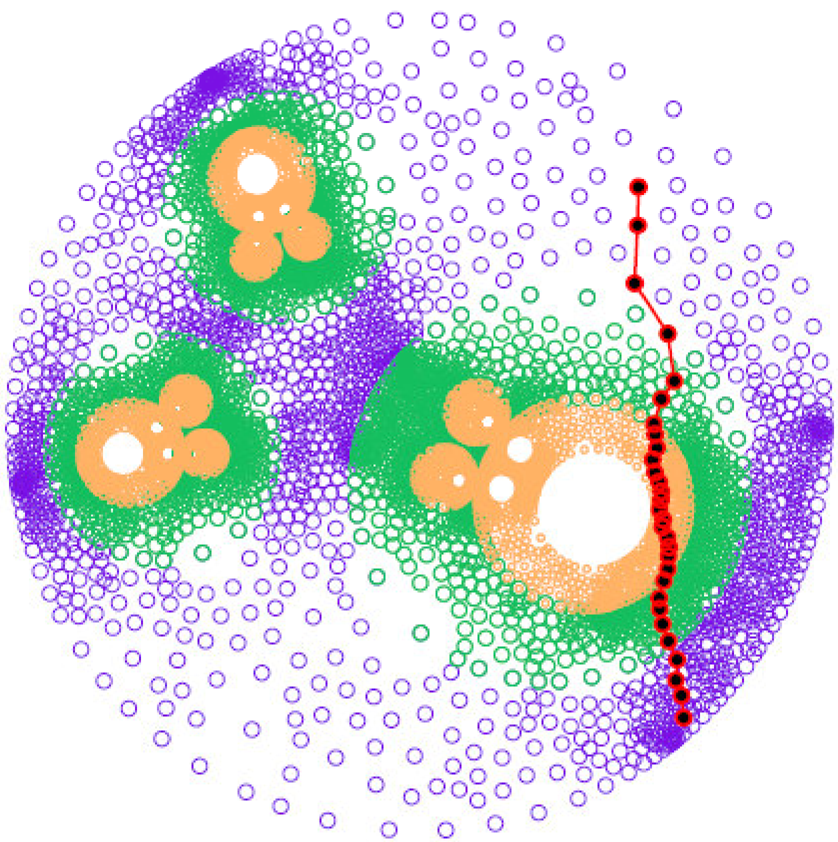


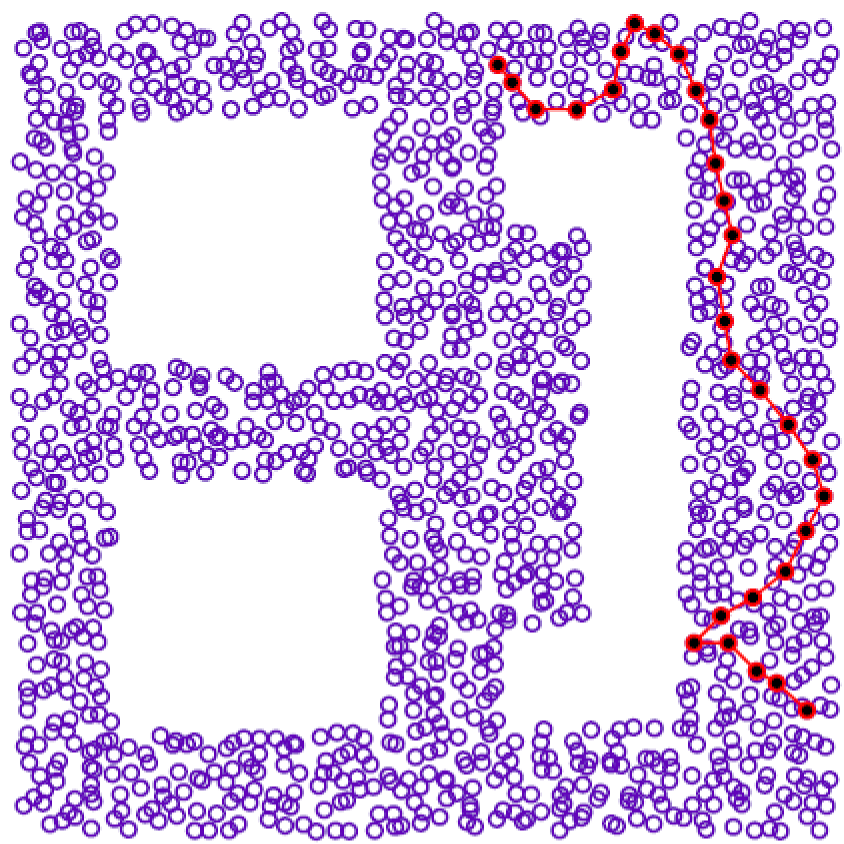
Covering Space: Tile up the domain



Routing in the covering space

- Equivalently, reflect on the hole boundary





Reflections

- Theorem:
  - Total area of the holes shrinks exponentially fast.
  - After O(log 1/ε) reflections, size of holes ~ ε.

- In practice
  - ≤ 5 levels of reflections
  - Reflections are computed on demand.

Max traffic load with greedy routing

- 1 or 2 reflections seem to be optimal.

Scheme	Avg. load	Max load	Avg. length	Max length
GPSR	33.6840	620.0	24.1915	92
1-ref	24.0682	319.0	17.571	42
2-ref	35.4960	190.0	25.439	117
3-ref	39.1742	241.0	27.9715	159
4-ref	43.9143	199.0	31.235	196
5-ref	46.3129	216.0	32.8865	228

Outline

Greedy routing

1. Guaranteed delivery
2. Load balancing
3. Resilient to failures

Problem III: Network Dynamics and Failures

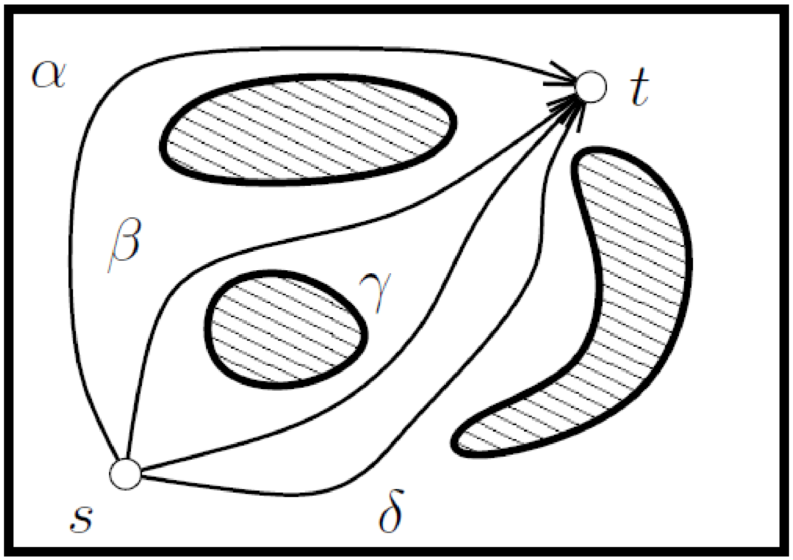
- Wireless links are unreliable
  - Interferences
  - Jamming
  - Nodes run out of battery, or are damaged

1. Select multiple paths that are "sufficiently different". → Paths of different homotopy types
2. Quickly switch to a different path upon link failure. → Use multiple routing metrics

Paths and Homotopy Types

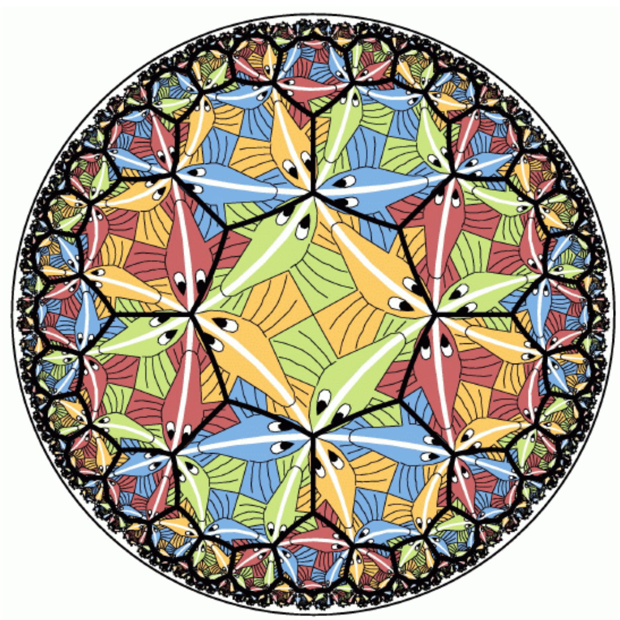
- Two paths are homotopy equivalent if they can be "locally deformed" to one another.
- Different homotopy types: bypass holes in different ways.

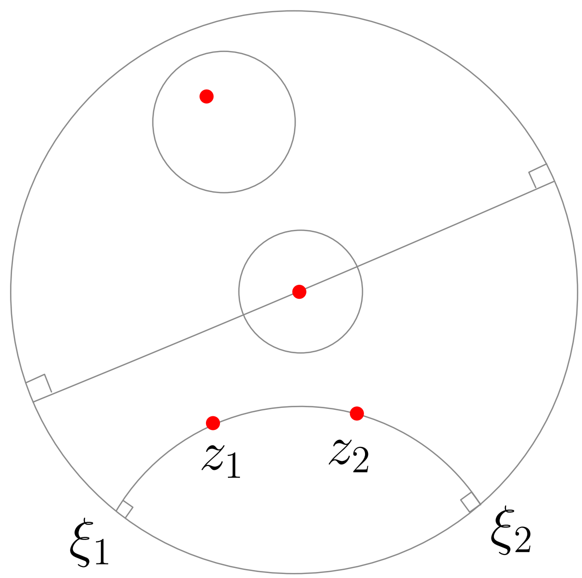
Goal: find a path of a given homotopy type using greedy routing.



Hyperbolic Space: Poincare Disk Model

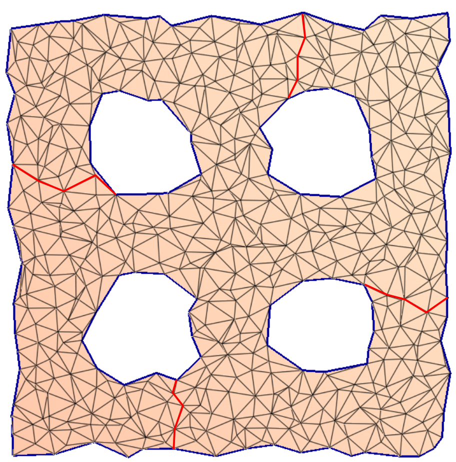
- Unit disk: entire hyperbolic space
- Hyperbolic line: circular arc (geodesic)
- Hyperbolic isometry up to Mobius transformation

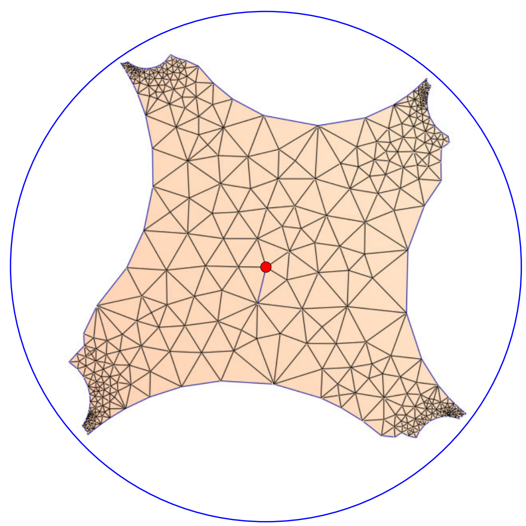




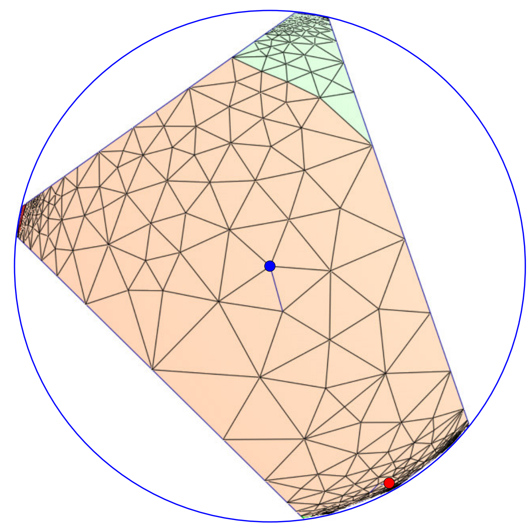
Hyperbolic Embedding by Ricci Flow

- Cut the holes open – a simply connected domain.
- Hyperbolic uniformization metric
  - Curvature: -1 on interior points, 0 on boundary points
- Use Ricci flow, embed to a convex piece → greedy routing guaranteed delivery.





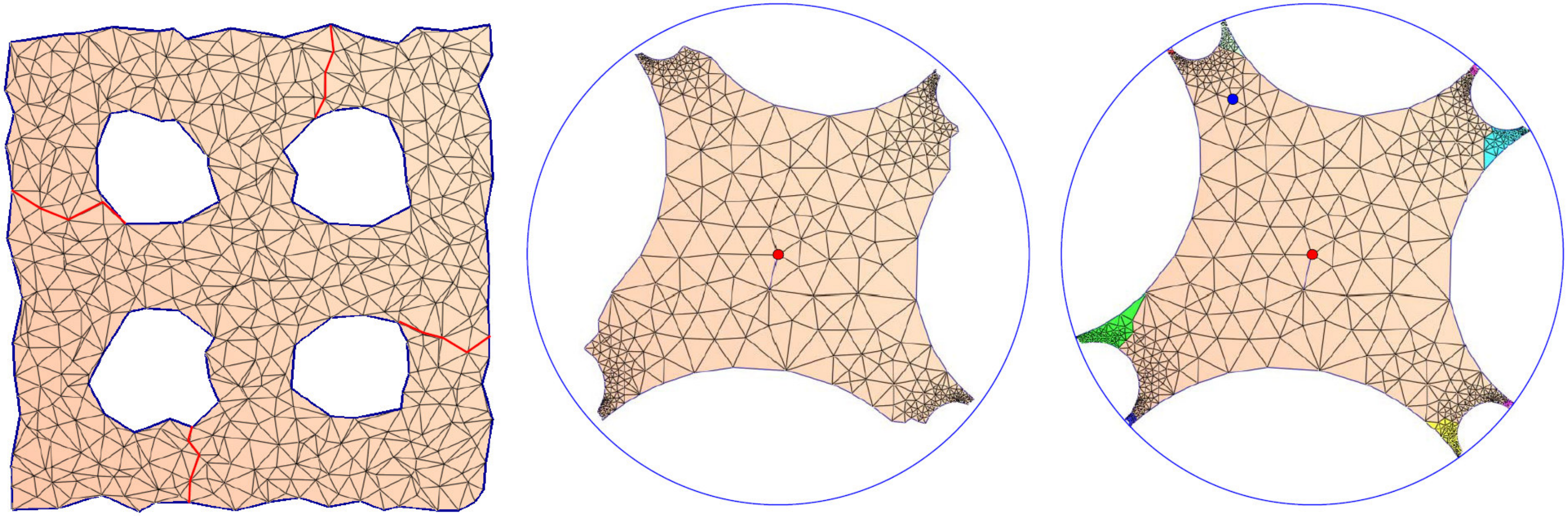
Boundary: geodesics



Klein projective model
Lines = Chords

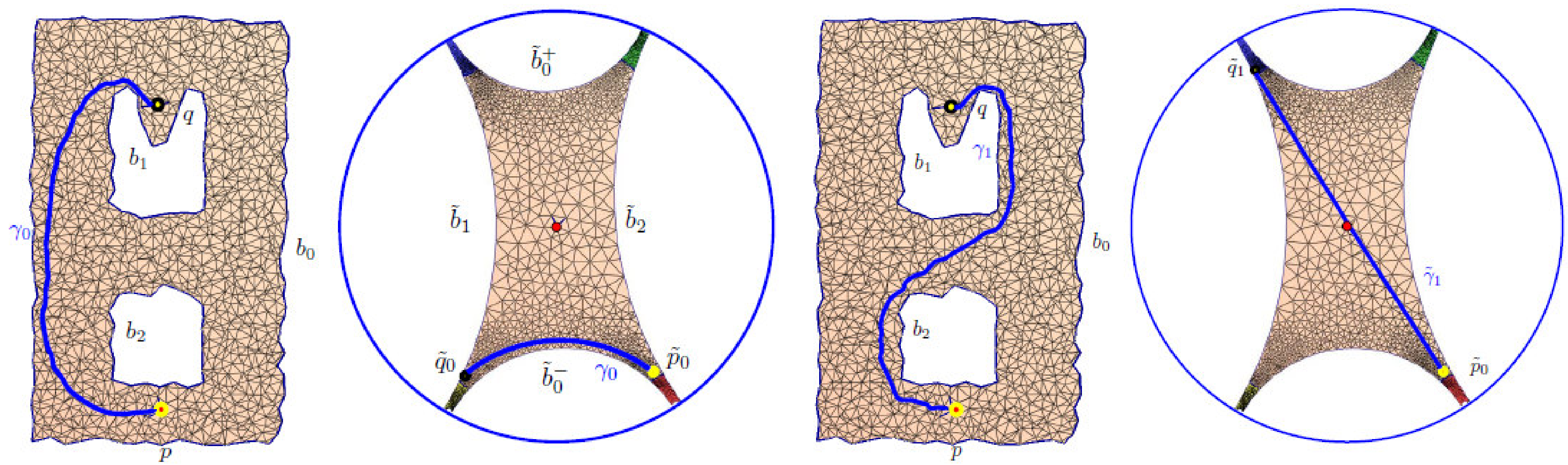
Universal Covering Space

- Embed into multiple such patches glued along boundaries.
- Patchs differ by a Möbius transformation.
- All such patches cover up the Poincare disk.



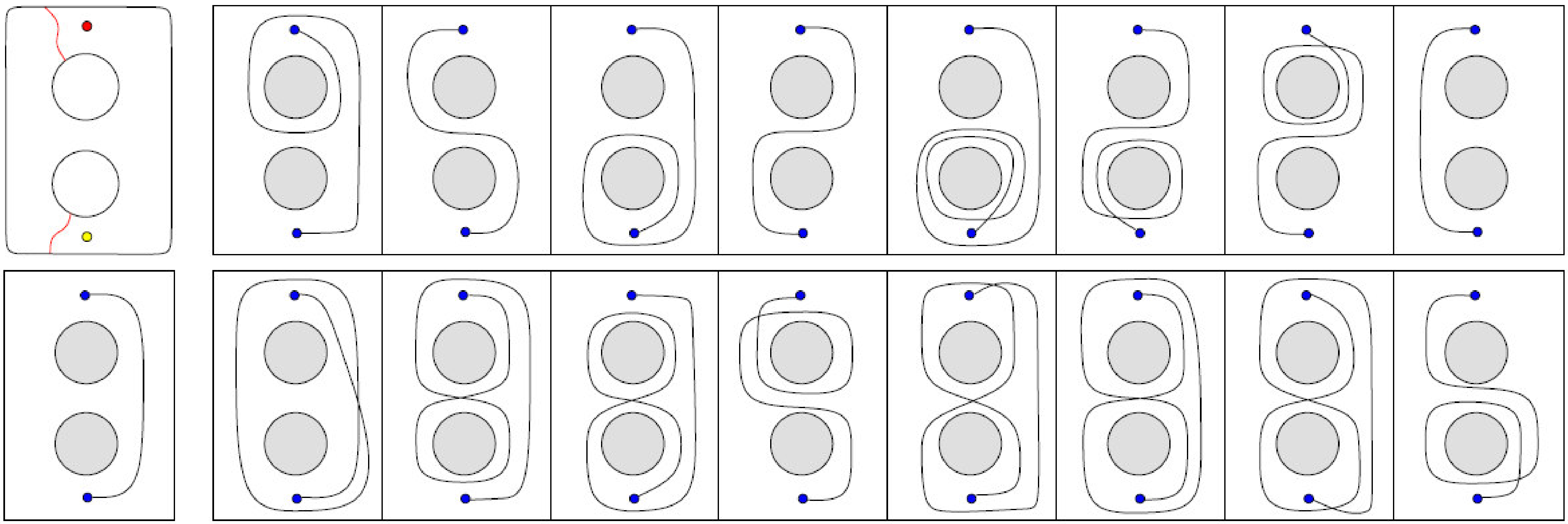
Universal Covering Space

- Greedy routing to the image of destination in different patches give paths of different homotopy types.



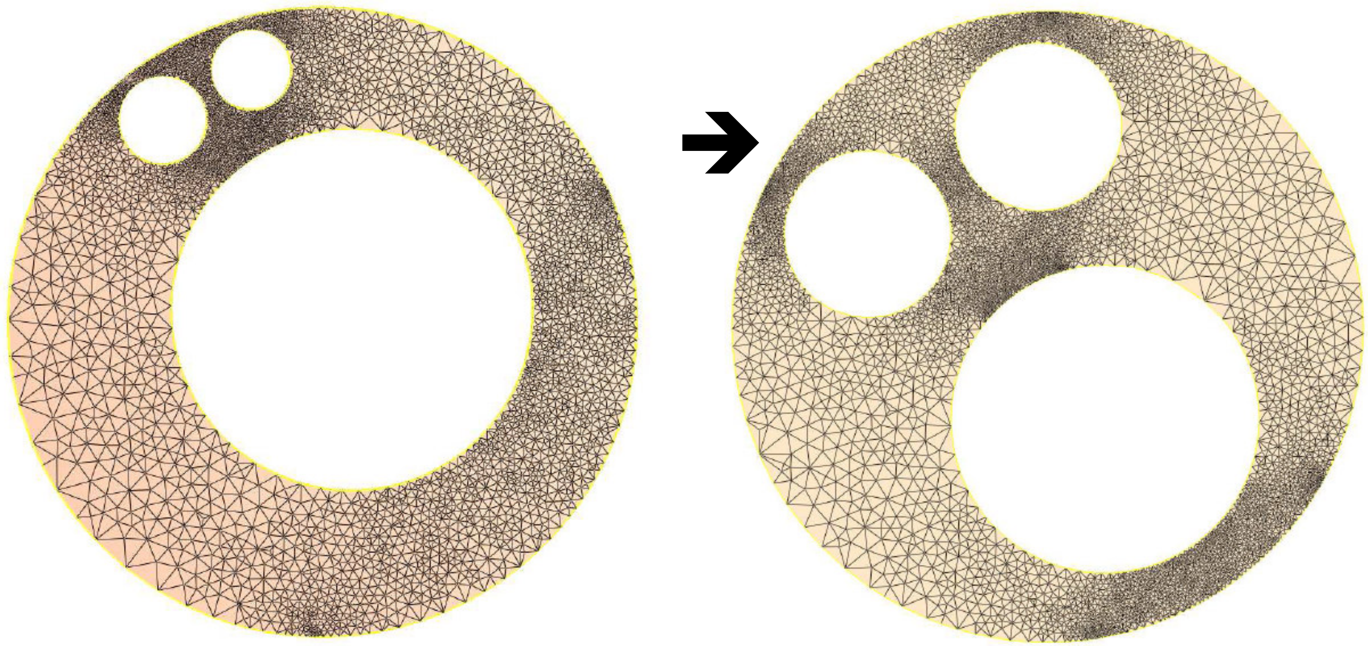
Universal Covering Space

- In practice, we only need a constant # patches
- Paths around a hole too many times are practically not interesting.



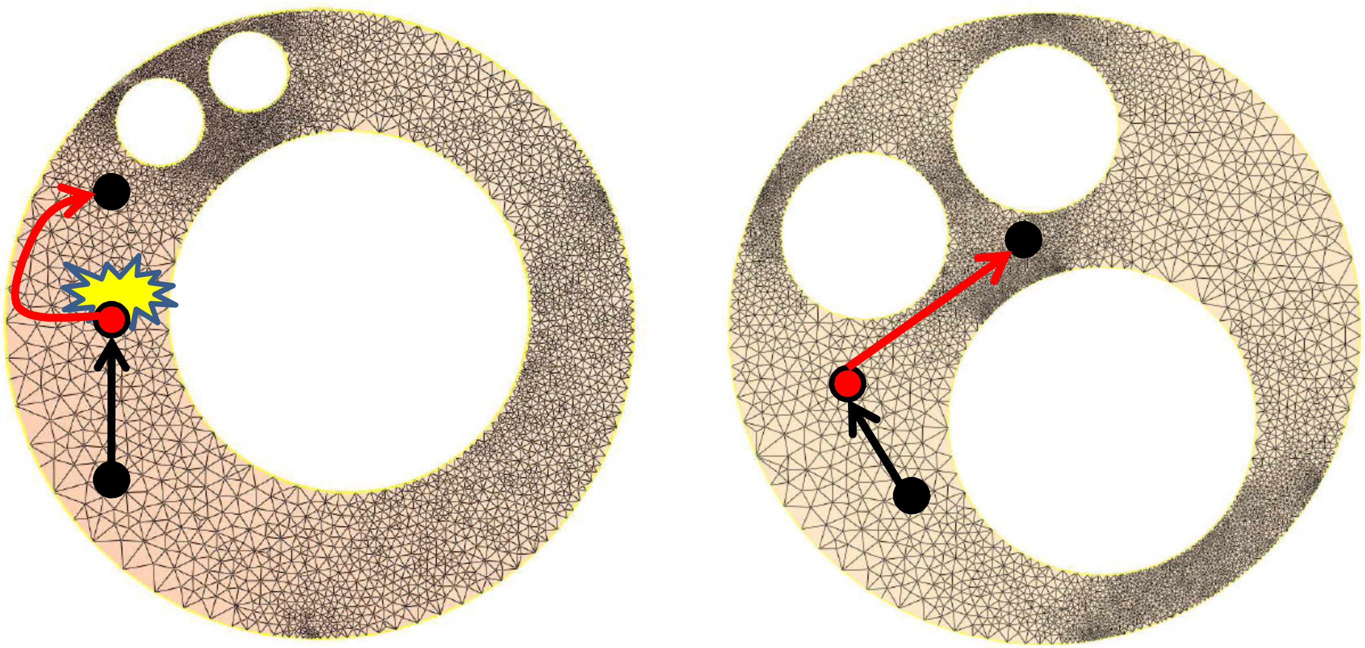
Recovery From Failures

- When a link fails, how to quickly find an alternative path?
- Recall: embedding into a circular domain is not unique, they differ by a Möbius transformation.



Multi-metric Routing

- If a link fails, we compute a Möbius transformation of the embedding & route in an alternative embedding.



Summary

- Deformation of network metric by changing local curvatures.
- Greedy routing with guaranteed delivery, good load balancing, resilience to failures, path diversity.

References

- Papers:
  – Greedy routing with guaranteed delivery using Ricci flows, IPSN'09.
  – Covering space for in-network sensor data storage, IPSN'10.
  – Resilient routing for sensor networks using hyperbolic embedding of universal covering space, INFOCOM'10.
- http://www.cs.sunysb.edu/~jgao

Questions?