Greedy Routing in Wireless Networks

Jie Gao
Stony Brook University
A generic sensor node

- CPU.
- On-board flash **memory** or external memory
- Sensors: thermometer, camera, motion, light sensor, etc.
- **Wireless** radio.
- **Battery**.
• Environment Monitoring
• Cyber Physical Systems
• Participatory Sensing
Greedy Routing

• Assign coordinates to nodes
• Message moves to neighbor closest to destination
• Simple, compact, scalable
Our objective

1. Examine a realistic setting of dense networks with holes

2. Greedy routing w/ nice properties
   - Guaranteed delivery
   - Load balancing
   - Multipath routing
   - Link dynamics
   - Node mobility

• Tool: computational geometry.
Large-scale sensor field

• large-scale, dense deployment.
  – Sufficient sensor coverage.
• w/ holes/obstacles.
• Prominent feature: *shape* of the network.
Local view vs. global view

Local view
• dense graphs
• links are dynamic

Global view
• geometric model
• topology is stable
Part I: Guaranteed delivery

- Can get stuck

- 1st idea: take detour to get around the local minimum.

- 2nd idea: assign new coordinates & remove local minima.
Deform the holes to be circular

- Greedy routing does not get stuck at circular holes.

Ricci flow
Curvature

Curvature: $\kappa = 1/R$

Flatter curves have smaller curvature.
Changing the Curvature

Flatten a curve: \( \kappa \to 0 \)

Make a cycle round \( \kappa \to 1/R \)
Discrete Curvature

Turning angle

Sum of curvature = $2\pi$
Discrete Curvature in 2D Triangulated Surface

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
\[ \kappa(v) = \pi - \sum_{i} \alpha_i \]
What they look like

Negative Curvature

Positive Curvature
Discrete Curvature in 2D Triangulated Surface

• 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: $2 - 2(# \text{ handles}) - (# \text{ holes})$

• Ricci flow: diffuse uneven curvatures to constant
Sanity check: total curvature

- $h$ holes, Euler characteristics = $2-(h+1)$
- Outer boundary: curvature $2\pi$
- Each inner hole: $-2\pi$
- Interior vertex: 0
- Total curvature: $(1-h)2\pi$
Ricci Flow in our case

• Target: a metric with pre-specified curvature
Ricci flow on sensor networks

• Curvature flow (Ricci flow) is a natural distributed algorithm.
  – Each node computes its current curvature & compare with target curvature
  – Each node changes the length of its adjacent edges ~ the curvature difference
  – Iterate

• Ricci energy is strictly convex ➔ unique solution
Routing
Deforming into a circular domain

• Deform a complex shape to a simple one, making it easy to explore the space of paths.
  – Multi-path routing
  – Recover from link failure
Circular Domains are not Unique

• Embedding into a circular domain is not unique, they differ by a Möbius transformation.
Möbius Transform

• Möbius transform
  – Conformal: maps circles to circles
  – Four basic elements: translation, dilation, rotation, inversions.

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

a, b, c, d are 4 complex numbers, \( ad \neq bc \)
Greedy Paths in Different Circular Domains

• Generate disjoint paths using different transformations
Greedy Paths in Different Circular Domains

• Generate disjoint paths using different transformations
Recovery from Temporary Link Failure

- We compute a Möbius transformation
  - S.t. the broken link is **NOT** on the greedy path.
  - Möbius transform attached to packet.
  - Make big jumps
Our objective

1. Examine a realistic setting of dense networks with holes

2. Greedy routing w/ nice properties
   – Guaranteed delivery
   – Load balancing
   – Multipath routing
   – Link dynamics
   – Node mobility
Part II: Load Balancing
Boundary nodes get overloaded
Soln: Reduce boundary load

- Use covering space to tile up the domain;
- Boundary “disappears”.
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 \( r \)

\[ f(z) - c = \frac{r^2}{\bar{z} - \bar{c}} \]
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
Reflections

• Theorem:
  – Total area of the holes shrinks exponentially fast.
  – After $O(\log 1/\varepsilon)$ reflections, size of holes $\sim \varepsilon$.

• In practice
  – $\leq 5$ levels of reflections
  – Reflections are computed on demand.
Routing in the covering space

• Equivalently, reflect on the hole boundary
Our objective

1. Examine a realistic setting of dense networks with holes
2. Greedy routing w/ nice properties
   – Guaranteed delivery
   – Load balancing
   – Multipath routing
   – Link dynamics
   – Node mobility
Circular domain embedding for mobile networks?

• What if nodes move around:
  – Run Ricci flow again to get the new coordinates
  – Location management scheme to update and maintain the current location information

• Our goal:
  – Avoid running Ricci flow again when nodes move
  – Compute the new virtual coordinates efficiently
Our Solution

• Assuming mobile nodes densely covering the domain
• Compute a compact map for the geometric domain in advance
• Preload the map to all nodes.
• Apply greedy routing based on virtual coordinates directly computed from the map
Compute the Compact Map

• Compute the continuous conformal map from the geometric domain to a circular domain
• Encode the map by the Schwarz-Christoffel Transformation
Schwarz-Christoffel Transformation

• Describe a conformal map of a disk onto the interior of a simple polygon

\[ f(z) = A \int_{z}^{\bar{z}} \prod_{k=1}^{n} (\zeta - z_k)^{-\beta_k} \, d\zeta + B \]
Schwarz-Christoffel Transformation

• We need the inverse: mapping a simple polygon onto a disk

\[ g(w) = f^{-1}(w) = \int_{w_0}^{w} \frac{dg(w)}{dw} dw + 0 = \int_{w_0}^{w} \prod_{k=1}^{n} (g(\tau) - z_k)^{\beta_k} d\tau + 0. \]
An Example

<table>
<thead>
<tr>
<th>$w$</th>
<th>$\beta$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 + 1i</td>
<td>0.5</td>
<td>-0.98993 + 0.14155i</td>
</tr>
<tr>
<td>20 + 1i</td>
<td>0.5</td>
<td>-0.99478 + 0.10202i</td>
</tr>
<tr>
<td>20 + 8i</td>
<td>0.5</td>
<td>-0.99480 + 0.10184i</td>
</tr>
<tr>
<td>5 + 8i</td>
<td>-0.5</td>
<td>-0.99978 - 0.02078i</td>
</tr>
<tr>
<td>5 + 13i</td>
<td>-0.5</td>
<td>0.97883 - 0.20466i</td>
</tr>
<tr>
<td>20 + 13i</td>
<td>0.5</td>
<td>0.99872 - 0.05063i</td>
</tr>
<tr>
<td>20 + 20i</td>
<td>0.5</td>
<td>0.99873 - 0.05041i</td>
</tr>
<tr>
<td>1 + 20i</td>
<td>0.5</td>
<td>1.00000 + 0.00000i</td>
</tr>
</tbody>
</table>

$A = -0.16623648 + 2.5343952i$

$B = 3.15 + 10.65i$

(a) vertices on $w$—plane

(b) prevertices on $z$—plane
Greedy Routing with Geographical Coordinates
Greedy Routing with Virtual Coordinates
Extensions

• Multiply connected domain
  – Similar ideas
  – Use Laurent series
Summary

• Deformation of network metric by changing local curvatures.
• “Simplify” the network shape to benefit network management.
Acknowledgement

• Joint work with Xianfeng David Gu, Feng Luo, Rik Sarkar, Wei Zeng, Xiaomeng Ban, Ruirui Jiang, Mayank Goswami, Xiaotian Yin, Xiaokang Yu, Siming Li
• Greedy Routing with Guaranteed Delivery Using Ricci Flows, IPSN’09.
• Resilient Routing for Sensor Networks using Hyperbolic Embedding of Universal Covering Space, INFOCOM’10
• Covering Space for In-Network Sensor Data Storage, IPSN’10.
• Exploration of Path Space using Sensor Network Geometry, IPSN’11.
• Scalable Routing in 3D High Genus Sensor Networks Using Graph Embedding, INFOCOM’12.
• Compact Conformal Map for Greedy Routing in Wireless Mobile Sensor Networks, INFOCOM’13.
Questions?
Local view
Graph abstraction of sensor networks

- The “graph” view of a sensor network depends very much on the network connectivity, which is not stable: links come and go, nodes fail...

- Dynamic graph problems, in general, are difficult problems.
obstacles

Global view
Geometric View of Sensor Networks

• The **global geometry/topology** is stable

• Computational geometers know how to handle shapes
Greedy embedding

• An embedding in which greedy routing always succeeds. [Papadimitriou, Ratajczak 05]
  — Not all graphs have a greedy embedding.
  — 3-connected planar graph has a greedy embedding [Leighton, Moitra 08][Angelini etal 08]

• Very nice theory

• In practice: links go up and down; maintaining the greedy drawing is non-trivial.
Ricci Flow: diffuse curvature

- Riemannian metric $g$ on $M$
- We modify $g$ by curvature
- Curvature evolves:
  \[
  \frac{dg_{ij}(t)}{dt} = -2K(t)g_{ij}(t)
  \]
  \[
  \frac{dK(t)}{dt} = \Delta_{g(t)}K(t)
  \]
- Same equation as heat diffusion $\Delta$: Laplace operator
Ricci Flow: diffuse curvature

- Ricci energy is strictly convex $\rightarrow$ unique solution with constant curvature everywhere and equal surface area.
- **Conformal** map: angle preserving
Discrete Flow : Circle Packing

- Conformal deformation: Change edge lengths of the triangulation, keep angles same

Circle packing metric: Radius at each vertex + angle at intersection

Related to edge length by cosine law

Remark: no need of an embedding (vertex location)
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
Ricci Flow in Sensor Networks

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

• Compute virtual coordinates
  – Apply distributed Ricci flow algorithm to compute **edge lengths** \( d(u, v) \) with the target curvature (0 at interior vertices and \( 2\pi/|B| \) at boundary vertices \( B \))
  – Calculate the **virtual coordinates** with the edge lengths \( d(u, v) \)
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 triangle
  – Iteratively flatten the graph by triangulation.
Examples
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

<table>
<thead>
<tr>
<th>Method</th>
<th>Delivery</th>
<th>Avg Stretch</th>
<th>Max Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ricci Flow</td>
<td>100%</td>
<td>1.59</td>
<td>3.21</td>
</tr>
<tr>
<td>NoGeo</td>
<td>83.66%</td>
<td>1.17</td>
<td>1.54</td>
</tr>
</tbody>
</table>
Convergence rate

- Curvature error bound
- Ricci flow step size

![Graph: Iterations Vs Number of Nodes](Image)
Theoretical guarantee of delivery

Theoretically,
• Ricci flow is a numerical algorithm.
• Triangles can be skinny.
• Route on edges/triangles.

Experimentally,
• Low stretch

What about load balancing?
“Center” gets overloaded

- A simple case: disk shape network.
- Greedy routing (send to neighbor closer to dest)
  \[\approx\] Shortest path routing
- Uniform traffic: All pairs of node have 1 message.
- Center load is high!
Soln #2: Reduce center load, lift to a sphere

- Greedy routing on a sphere has perfect uniform traffic load
Reduce center load: map to sphere

- **Koebe-Andreev-Thurston Theorem**: Any 3-connected graph can be embedded as a convex polyhedron
  - Circle packing with circles on vertices.
  - all edges are tangent to a unit sphere.
  - vertices are lifted from the sphere.
Polyhedron Routing

- [Papadimitriou & Ratajczak] Greedy routing with
  \[ d(u, v) = -c(u) \cdot c(v) \]
  guarantees delivery.

- Route along the surface of a convex polytope.
More recent work

• Extension to 3D sensor networks. [INFOCOM’12]
  – Sensors monitoring complex tunnels.
  – Embed & route on a high-genus surface
More recent work

• Space-filling curves with progressive density for non-simple domains. [Infocom’13]
  – App: data mule planning