Two problems remain

• Both RNG and GG remove some edges \(\Rightarrow\) a short path may not exist!

• The shortest path on RNG or GG might be much longer than the shortest path on the original network.

• Even if the planar subgraph contains a short path, can greedy routing and face routing find a short one?
Tackle problem I: Find a planar spanner
Find a good subgraph

- Goal: a **planar spanner** such that the shortest path is at most $\alpha$ times the shortest path in the unit disk graph.
  - Euclidean spanner: The shortest path length is measured in total Euclidean length.
  - Hop spanner: The shortest path length is measured in hop count.

- $\alpha$: spanning ratio.
  - Euclidean spanning ratio $\geq \sqrt{2}$
  - Hop spanning ratio $\geq 2$.

- Let’s first focus on Euclidean spanner.
Delaunay triangulation is an Euclidean spanner

- DT is a 2.42-spanner of the Euclidean distance.
- For any two nodes $uv$, the Euclidean length of the shortest path in DT is at most 2.42 times $|uv|$. 
Restricted Delaunay graph

- Keep all the Delaunay edges no longer than 1.
- Claim: RDG is a 2.42-spanner (in total Euclidean length) of the UDG.
- Proof sketch: If an edge in UDG is deleted in RDG, then it’s replaced by a path with length at most 2.42 longer.
Construction of RDG

- Easy to compute a superset of RDG: Each node computes a local Delaunay of its 1-hop neighbors.
  - A global Delaunay edge is always a local Delaunay edge, due to the empty-circle property.
  - A local Delaunay may not be a global Delaunay edges.

- What if the superset has crossing edges?
**Crossing Lemma**

- **Crossing lemma**: if two edges cross in a UDG, then one node has edges to the three other nodes in UDG.

\[
|uw| \leq |wp| + |up| \\
|vx| \leq |vp| + |xp| \\
|wu| + |vx| \leq 2 \\
|wu| + |vx| \leq 2 \\
|wv| + |ux| \leq 2 \\
\]

Also, \(|wv| + |ux| \leq |wx| + |ux| \leq 2\)

There must be 2 edges on the quad adjacent to the same node.
Detect crossings between local delaunay edges

- By the crossing Lemma: if two edges cross in a UDG, one of them has 3 nodes in its neighborhood and can tell which one is not Delaunay.

- Neighbors exchange their local DTs to resolve inconsistency.
  - A node tells its 1-hop neighbors the non-Delaunay edges in its local graph.
  - A node receiving a “forbidden” edge will delete it from its local graph.

- Completely distributed and local.
RDG construction

- 1-hop information exchange is sufficient.
  - Planar graph;
  - All the short Delaunay edges are included.
  - We may have some planar non-Delaunay edges but that does not hurt spanning property.
More on RDG construction

- RDG can be constructed without the full location information.

- Only local angle information suffices.

- Key operation: If two edges in the unit-disk graph cross, remove the one that is not in the Delaunay triangulation.

- How to tell that an edge is not in the Delaunay triangulation?
Removing non-Delaunay edges

If two edges $AB$, $CD$ cross, there are only three cases:

(i) 
(ii) 
(iii)
Removing non-Delaunay edges

If two edges $AB$, $CD$ cross, there are only three cases:

(i)

(ii)

(iii)

With angle info, the shape is fixed! Node C can tell which edge is not Delaunay.
Removing non-Delaunay edges

Case (i) : Use the “empty-circle” test of Delaunay triangulation

\[ |AC| > 1 \geq |CD| \]
\[ |BC| > 1 \geq |CD| \]

Conclusion: The edge AB is not a Delaunay edge.
Find a hop spanner

- Restricted Delaunay graph is not a hop spanner.
  - Take $n$ nodes uniformly in a segment of length 1. The hop count can be as large as $n-1$.
- Reduce the density of the sensors.
  - Use clustering to reduce density.
  - Compute RDG on the subset to get a hop spanner.
  - Clustering also reduce interference and enables efficient resource reuse such as bandwidth.
Reduce node density

- Find a subset of nodes, called **clusterheads**
  - Each node is directly connected to at least 1 clusterhead.
  - No two clusterheads are connected.
- Use a greedy algorithm. Pick a node as a clusterhead, remove all the 1-hop neighbors, continue.
- Constant density: $\leq 6$ clusterheads in any unit disk.
  - The angle spanned by two clusterheads is at least $\pi/3$. 

\[
\pi/3
\]
Connect clusterheads by gateways

- For two clusterheads, if their clients have an edge, then we pick one pair as **gateway** nodes.

- Notice that clusterheads \(x, y\) are within 3 hops to have a pair of gateways.

- There are constant clusterheads and gateways inside any unit disk.
Path on clusterheads and gateways

- For two nodes $u, v$ that are $k$ hops away, there is a path through clusterheads and gateways with at most $3k+2$ hops.

- Construct RDG on clusterheads and gateways, which have constant bounded density.

Shortest path
A Routing Graph Sample

Select clusterheads

Clusterheads select gateways

RDG on clusterheads & gateways
**Restricted Delaunay graph**

- **Claim:** (RDG on clusterheads and gateways + edges from clients to clusterheads) is a constant hop spanner of the original UDG.

- **Proof sketch:**
  - The shortest path $P$ in the unit disk graph has $k$ hops.
  - Through clusterheads and gateways $\exists$ a path $Q$ with $\leq 3k+2$ hops.
  - $Q$'s total Euclidean length is $\leq 3k+2$.
  - The shortest path on the RDG, $H$, has Euclidean length $\leq 2.42 \times (3k+2)$.
  - By constant density property a region with width 1 and length $2.42 \times (3k+2)$ has $O(k)$ nodes inside. So $\#$ hops of $H$ is $O(k)$.
  - This concludes the hop spanner property.
Restricted Delaunay graph

RNG

RDG
Restricted Delaunay graph

RNG

RDG

Clusterhead
Gateway
Tackle problem II: Improve face routing to find a short path & Geographic routing in practice
Overview of geographical routing

- Routing with geographical location information.
  - Greedy forwarding.
  - If stuck, do face routing on a planar sub-graph.
Overview

• How to find a planar subgraph?
  – Use distributed construction: relative neighborhood graph, Gabriel graph, etc.
  – A planar subgraph that contains a short path: restricted Delaunay graph: short Delaunay edges.

• Big problem: how is the performance of geo-routing?
  – Can we always find a short path?
Bad news: Lower bound of localized routing

- Any deterministic or randomized localized routing algorithm takes a path of length $\Omega(k^2)$, if the optimal path has length $k$.

- The adversary decides where the chain $w_t$ is. Since we store no information on nodes, in the worst case we have to visit about $\Omega(k)$ chains and pay a cost of $\Omega(k^2)$.
Good news: greedy forwarding is optimal

- If greedy routing gets to the destination, then the path length is at most $O(k^2)$, if the optimal path has length $k$.

- $|uv|$ is at most $k$. On the greedy path, every other node is not visible, so they are of distance at least 1 away. By a packing lemma, there are at most $O(k^2)$ nodes inside a disk of radius $k$.

How is face routing? How is greedy + face routing?
Performance of face routing
Performance of face routing

- What if we choose the wrong side?
Adaptive face routing

- Suppose the shortest path on the planar graph is bounded by L hops.
- Bound the search area by an ellipsoid \( \{ x \mid |xs| + |xt| \leq L \} \) → never walk outside the ellipsoid.
- Follow one direction, if we hit the ellipsoid; turn back.
- If we find a \textit{better} intersection \( p \) of the face with line \( st \), change to the face containing \( pt \).
- In the worst case, visit every node inside the ellipsoid: \( O(L^2) \) by the bounded density property (through clustering).
Adaptive face routing

- How to guess the upper bound $L$?
- Start from a small value say $|st|$; if we fail to find a path, then we double $L$ and re-run adaptive face routing.
- By the time we succeed, $L$ is at most twice the shortest path length $k$. The number of phases is $O(\log k)$.
- Total cost $= O(\sum_i (k/2^i)^2) = O(k^2)$. $\Leftarrow$ asymptotically optimal.
A simple worst-case optimal routing alg

• It’s easy to get a worst-case $O(k^2)$ bound.

• Do adaptive restricted flooding.
• Start with a small threshold $t$. Flood all the nodes within distance $t$ from the source.
• If the destination is not reached, double the radius and retry.

• On a network with bounded density, the total cost is $O(k^2)$ if the shortest path has length $k$.

Not quite efficient for most good cases.
Fall back to greedy

- When a node visits a node closer to the destination than that at which it enters the face routing mode, it returns to greedy mode.

- Other fall-back schemes are proposed. E.g., GOAFR+ considers falling back to greedy mode when considering a face change and when there are sufficient nodes closer to the destination than the local minimum.
Beyond point-to-point routing

- Multicast to a geographical region.
  - Use geographical forwarding to reach the destination region.
  - Restricted flooding inside the region.

- Routing on a curve.
  - Follow a parametric curve \(<x(t), y(t)>\).
  - Greedily select the nodes near the curve.
Geographical routing in practice
Revisit the assumptions of GPSR

- Nodes know their accurate locations.
- The network topology follows the unit disk graph model.

- These are 2 BIG assumptions.
- Localization is hard, both in theory and in practice.
- Unit disk graph model is simply not true in practice.
Sensor communication model

- Contour of probability of packet reception from a central node at two different transmit power settings.

Does not look like a disk to me.

Source: Ganesan, et.al
Each point represents a pair of nodes.
Sensor communication model

- How in-bound link quality varies with distance.

Source: Mark Paskin
Sensor communication model

- Link quality varies with time.

Source: Mark Paskin
Sensor communication model

- Experiments show that
  - Irregular transmission range: stable long links exist, links between two close by nodes might not exist.
  - Links are asymmetric (A talks to B, B can’t talk to A).
  - Localization errors.

- This makes planar graph construction fail.
Planar graph subtraction fails on irregular radio range

- Network is partitioned.
- Crossing links.

Edge AB is removed.

No crossing of line SD is closer than point p.
Testing GPSR on a real testbed

- GPSR only succeeds on 68.2% directed node pairs.

A 50-node testbed at Intel Berkeley Lab
Planarization partitions the network

- Planar graph subtraction disconnects the network.

Gabriel Graph
Directional link
Crossing links
A 50-node testbed at Berkeley Soda Hall
A small fix on the asymmetric links

- With irregular radio range the planar graph construction fails.

- A small fix by using mutual witness:
  - The link AB is removed only if there is witness that is seen by both A and B.
A small fix on the asymmetric links

- Leaves more crossing links.
- Only improves the success rate of GPSR to 87.8%.
Cross Link Detection Protocol

- Try to do face routing on a non-planar network.
- Eliminate not-OK crossings and keep the graph connected.
- Each node probes each of its links to see if it’s crossed by other links.
- How to probe? Record the link to be probed in packet, do face routing and mark all crossings.
Cross Link Detection Protocol

- Start from D and do face routing.

Remove either AD or BC

Can't remove BC

Can't remove AD

Can't remove either

Observation: a not-OK crossing is traversed twice, once in each direction.
Cross Link Detection Protocol

- A link is not removable, if it’s traversed twice.
- A crossing L and L’: remove the removable one. If none of them is removable, do nothing.
- Protocol: do the probing sequentially.
- For different probing sequences, one can get different graphs.
- Or, probe in a lazy fashion.
Multiple crossing links

- If a link is crossed by multiple other links, we probe it multiple times.

- Probing a pair of cross links may not find all the crossing, if they are obscured by other links.
Problems with CLDP

• How many probes? In what order?

• Can we probe the links concurrently?
  – Lock a link when it’s probed.

• Say we finish all the probes, and do face routing on the graph. Can we guarantee that the face routing always succeeds?
Summary on geographic routing

- Geographical routing is nice in terms of
  - No flooding
  - No routing table maintenance
  - Scalable

- Face routing: Nice in theory, big mess in practice.