Topology Control
Chapter 3
Inventory Tracking (Cargo Tracking)

• Current tracking systems require line-of-sight to satellite.
• Count and locate containers
• Search containers for specific item
• Monitor accelerometer for sudden motion
• Monitor light sensor for unauthorized entry into container
Rating

- Area maturity
  - First steps
  - Text book

- Practical importance
  - No apps
  - Mission critical

- Theory appeal
  - Booooorrring
  - Exciting
Overview – Topology Control

- Proximity Graphs: Gabriel Graph et al.
- Practical Topology Control: XTC
- Interference
Topology Control

- **Drop long-range neighbors**: Reduces interference and energy!
- But still stay **connected** (or even spanner)
Topology Control as a Trade-Off

Network Connectivity
Spanner Property

Conserve Energy
Reduce Interference
Sparse Graph, Low Degree
Planarity
Symmetric Links
Less Dynamics

\[ d_{TC}(u,v) \leq c \cdot d(u,v) \]
Spanners

- Let the distance of a path from node $u$ to node $v$, denoted as $d(u,v)$, be the sum of the Euclidean distances of the links of the shortest path.
  - Writing $d(u,v)^p$ is short for taking each link distance to the power of $p$, again summing up over all links.

- Basic idea: $S$ is spanner of graph $G$ if $S$ is a subgraph of $G$ that has certain properties for all pairs of nodes, e.g.
  - Geometric spanner: $d_S(u,v) \leq c \cdot d_G(u,v)$
  - Power spanner: $d_S(u,v)^\alpha \leq c \cdot d_G(u,v)^\alpha$, for path loss exponent $\alpha$
  - Weak spanner: path of $S$ from $u$ to $v$ within disk of diameter $c \cdot d_G(u,v)$
  - Hop spanner: $d_S(u,v)^0 \leq c \cdot d_G(u,v)^0$
  - Additive hop spanner: $d_S(u,v)^0 \leq d_G(u,v)^0 + c$
  - $(\alpha, \beta)$ spanner: $d_S(u,v)^0 \leq \alpha \cdot d_G(u,v)^0 + \beta$
  - The stretch can be defined as maximum ratio $d_S/d_G$
Gabriel Graph

- Let $\text{disk}(u,v)$ be a disk with diameter $(u,v)$ that is determined by the two points $u,v$.
- The Gabriel Graph $\text{GG}(V)$ is defined as an undirected graph (with $E$ being a set of undirected edges). There is an edge between two nodes $u,v$ iff the disk$(u,v)$ including boundary contains no other points.
- As we will see the Gabriel Graph has interesting properties.
Delaunay Triangulation

- Let $\text{disk}(u,v,w)$ be a disk defined by the three points $u,v,w$.

- The Delaunay Triangulation (Graph) $\text{DT}(V)$ is defined as an undirected graph (with $E$ being a set of undirected edges). There is a triangle of edges between three nodes $u,v,w$ iff the $\text{disk}(u,v,w)$ contains no other points.

- The Delaunay Triangulation is the dual of the Voronoi diagram, and widely used in various CS areas
  - the DT is planar
  - the DT is a geometric spanner
Other Proximity Graphs

• Relative Neighborhood Graph RNG(V)
  
  – An edge $e = (u,v)$ is in the RNG(V) iff there is no node $w$ in the “lune” of $(u,v)$, i.e., no node with $(u,w) < (u,v)$ and $(v,w) < (u,v)$.

• Minimum Spanning Tree MST(V)
  
  – A subset of $E$ of $G$ of minimum weight which forms a tree on $V$. 
Properties of Proximity Graphs

- Theorem 1:
  MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ DT

- Corollary:
  Since the MST is connected and the DT is planar, all the graphs in Theorem 1 are connected and planar.

- Theorem 2:
  The Gabriel Graph is a power spanner (for path loss exponent $\alpha \geq 2$). So is GG $\cap$ UDG.

- Remaining issue: either high degree (RNG and up), and/or no spanner (RNG and down). There is an extensive and ongoing search for “Swiss Army Knife” topology control algorithms.
More Proximity Graphs

- **β-Skeleton**
  - Disk diameters are $\beta \cdot d(u,v)$, going through $u$ resp. $v$
  - Generalizing GG ($\beta = 1$) and RNG ($\beta = 2$)

- **Yao-Graph**
  - Each node partitions directions in $k$ cones and then connects to the closest node in each cone

- **Cone-Based Graph**
  - Dynamic version of the Yao Graph. Neighbors are visited in order of their distance, and used only if they cover not yet covered angle
Lightweight Topology Control

- Topology Control commonly assumes that the node positions are known.

What if we do not have access to position information?
XTC: Lightweight Topology Control without Geometry

- Each node produces “ranking” of neighbors.
- Examples
  - Distance (closest)
  - Energy (lowest)
  - Link quality (best)
  - Must be symmetric!
- Not necessarily depending on explicit positions
- Nodes exchange rankings with neighbors
XTC Algorithm (Part 2)

- Each node locally goes through all neighbors in order of their ranking.
- If the candidate (current neighbor) ranks any of your already processed neighbors higher than yourself, then you do not need to connect to the candidate.
XTC Analysis (Part 1)

- **Symmetry**: A node $u$ wants a node $v$ as a neighbor if and only if $v$ wants $u$.

- **Proof**:
  - Assume 1) $u \rightarrow v$ and 2) $u \not\leftrightarrow v$
  - Assumption 2) $\exists w$: (i) $w <_v u$ and (ii) $w <_u v$

  *In node $u$’s neighbor list, $w$ is better than $v*

  **Contradicts Assumption 1**
XTC Analysis (Part 1)

- **Symmetry**: A node $u$ wants a node $v$ as a neighbor if and only if $v$ wants $u$.

- **Connectivity**: If two nodes are connected originally, they will stay so (easy to show if rankings are based on symmetric link-weights).

- If the ranking is energy or link quality based, then XTC will choose a topology that routes **around walls** and obstacles.
XTC Analysis (Part 2)

- If the given graph is a **Unit Disk Graph** (no obstacles, nodes homogeneous, but **not** necessarily uniformly distributed), then …

- The **degree** of each node is at most 6.
- The topology is **planar**.
- The graph is a subgraph of the **RNG**.

- **Relative Neighborhood Graph** **RNG(V)**:
  - An edge \( e = (u,v) \) is in the RNG(V) iff there is no node \( w \) with \( (u,w) < (u,v) \) and \( (v,w) < (u,v) \).
XTC Average-Case

Unit Disk Graph

XTC
XTC Average-Case (Degrees)

Network Density [nodes per unit disk]

- **UDG max**
- **GG max**
- **XTC max**

- **UDG avg**
- **GG avg**
- **XTC avg**
XTC Average-Case (Stretch Factor)

XTC vs. UDG – Geometric

GG vs. UDG – Geometric

XTC vs. UDG – Power

GG vs. UDG – Power
Implementing XTC, e.g. BTnodes v3
Implementing XTC, e.g. on mica2 motes

- **Idea:**
  - XTC chooses the reliable links
  - The quality measure is a moving average of the received packet ratio
  - Source routing: route discovery (flooding) over these reliable links only
  - (black: using all links, grey: with XTC)
Topology Control as a Trade-Off

Network Connectivity
Spanner Property

Conserve Energy
Reduce Interference
Sparse Graph, Low Degree
Planarity
Symmetric Links
Less Dynamics

Really?!?
What is Interference?

Link-based Interference Model

```
How many nodes are affected by communication over a given link?
```

Node-based Interference Model

```
By how many other nodes can a given network node be disturbed?
```

Problem statement

- We want to minimize maximum interference
- At the same time topology must be connected or spanner
Low Node Degree Topology Control?

Low node degree does not necessarily imply low interference:

Very low node degree but huge interference
Let’s Study the Following Topology!

…from a worst-case perspective
Topology Control Algorithms Produce…

- All known topology control algorithms (with symmetric edges) include the nearest neighbor forest as a subgraph and produce something like this:

- The interference of this graph is $\Omega(n)!$
But Interference…

- Interference does not need to be high…

- This topology has interference $O(1)$!!
Link-based Interference Model

There is no local algorithm that can find a good interference topology.

The optimal topology will not be planar.
Link-based Interference Model

- LIFE (Low Interference Forest Establisher)
  - Preserves Graph Connectivity

LIFE

- Attribute interference values as weights to edges
- Compute minimum spanning tree/forest (Kruskal’s algorithm)

LIFE constructs a minimum-interference forest
Average-Case Interference: Preserve Connectivity

Network Density [nodes per unit disk] vs. Interference for different network models:
- UDG
- GG
- RNG
- LIFE
Node-based Interference Model

- Already **1-dimensional node distributions** seem to yield inherently high interference...

- ...but the **exponential node chain** can be connected in a better way
Connecting linearly results in interference $O(n)$

- Already 1-dimensional node distributions seem to yield inherently high interference...

- ...but the **exponential node chain** can be connected in a better way

Interference $\in O(\sqrt{n})$

Matches an existing lower bound
Node-based Interference Model

- Arbitrary distributed nodes in one dimension
  - Approximation algorithm with approximation ratio in $O\left(\frac{4}{\sqrt{n}}\right)$

- Two-dimensional node distributions
  - Simple randomized algorithm resulting in interference $O(\sqrt{n \log n})$
  - Can be improved to $O(\sqrt{n})$
Open problem

- On the theory side there are quite a few open problems. Even the simplest questions of the node-based interference model are open:

- We are given n nodes (points) in the plane, in arbitrary (worst-case) position. You must connect the nodes by a spanning tree. The neighbors of a node are the direct neighbors in the spanning tree. Now draw a circle around each node, centered at the node, with the radius being the minimal radius such that all the nodes’ neighbors are included in the circle. The interference of a node u is defined as the number of circles that include the node u. The interference of the graph is the maximum node interference. We are interested to construct the spanning tree in a way that minimizes the interference. Many questions are open: Is this problem in P, or is it NP-complete? Is there a good approximation algorithm? Etc.