Chapter 7
TOPOLOGY
CONTROL

Mobile Computing
Winter 2005 / 2006

Overview – Topology Control

• Gabriel Graph et al.
• XTC
• Interference
• SINR & Scheduling Complexity

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

Sometimes also clustering, Dominating Set construction (See later)
Gabriel Graph

- Let disk(u,v) be a disk with diameter (u,v) that is determined by the two points u,v.
- The Gabriel Graph 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 if the disk(u,v) including boundary contains no other points.
- As we will see the Gabriel Graph has interesting properties.

Delaunay Triangulation

- Let disk(u,v,w) be a disk defined by the three points u,v,w.
- The Delaunay Triangulation (Graph) 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 if the 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 distance of a path (s,…,t) on the DT is within a constant factor of the s-t distance.

Other planar graphs

- 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).
- Minimum Spanning Tree MST(V)
- A subset of E of G of minimum weight which forms a tree on V.

Properties of planar graphs

- Theorem 1:
  MST(V) ⊆ RNG(V) ⊆ GG(V) ⊆ DT(V)
- Corollary:
  Since the MST(V) is connected and the DT(V) is planar, all the planar graphs in Theorem 1 are connected and planar.
- Theorem 2:
  The Gabriel Graph contains the Minimum Energy Path (for any path loss exponent α ≥ 2)
- Corollary:
  GG(V) ∩ UDG(V) contains the Minimum Energy Path in UDG(V)
More examples

- \( \beta \)-Skeleton
  - Generalizing Gabriel (\( \beta = 1 \)) and Relative Neighborhood (\( \beta = 2 \)) Graph

- 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

XTC: Lightweight Topology Control

- Topology Control commonly assumes that the node positions are known.
- What if we do not have access to position information?
- XTC algorithm
- XTC analysis
  - Worst case
  - Average case

XTC: Lightweight Topology Control without geometry

- Each node produces “ranking” of neighbors.
- Examples
  - Distance (closest)
  - Energy (lowest)
  - Link quality (best)
- 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) $u \leftrightarrow v$
  - Assumption 2) $\exists w$ \((i) w \prec u \text{ and (ii) } w \prec v\)
  
  **Contradicts Assumption 1**

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 $\text{RNG}(V)$:
  - An edge $e = (u,v)$ is in the $\text{RNG}(V)$ 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)**

![Graph showing network density vs. node degree for XTC, UDG, and GG]

**XTC Average-Case (Stretch Factor)**

![Graph showing network density vs. stretch factor for XTC, UDG, and GG]

**XTC Average-Case (Geometric Routing)**

![Graph showing network density vs. geometric routing performance for GFG/GPSR on GG and GOAFR+ on G_{XTC}]

**k-XTC: More connectivity**

- A graph is k-(node)-connected, if k-1 arbitrary nodes can be removed, and the graph is still connected.

- In k-XTC, an edge \((u,v)\) is only removed if there exist k nodes \(w_1, \ldots, w_k\) such that the 2k edges \((w_1, u), \ldots, (w_k, u), (w_1, v), \ldots, (w_k, v)\) are all better than the original edge \((u,v)\).

- Theorem: If the original graph is k-connected, then the pruned graph produced by k-XTC is as well.

- Proof: Let \((u,v)\) be the best edge that was removed by k-XTC. Using the construction of k-XTC, there is at least one common neighbor \(w\) that survives the slaughter of k-1 nodes. By induction assume that this is true for the j best edges. By the same argument as for the best edge, also the \(j+1^{st}\) edge \((u',v')\), since at least one neighbor survives \(w\) survives and the edges \((u',w')\) and \((v',w')\) are better.
Implementing XTC, e.g. BTnodes v3

- 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

Topology Control as a Trade-Off

- Network Connectivity
- Spanner Property
- Conserve Energy
- Reduce Interference
- Sparse Graph, Low Degree
- 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 network nodes can a given node be disturbed?”

- Problem statement
  - We want to minimize maximum interference
  - At the same time topology must be connected or a spanner etc.
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

- Interference-optimal topologies:
  - There is no local algorithm that can find a good interference topology
  - The optimal topology will not be planar

  ![Interference-optimal topologies diagram]

- LIFE (Low Interference Forest Establisher)
  - Preserves Graph Connectivity
    - Attribute interference values as weights to edges
    - Compute minimum spanning tree/forest (Kruskal’s algorithm)
    - LIFE constructs a minimum-interference forest

  ![LIFE algorithm diagram]

- LISE (Low Interference Spanner Establisher)
  - Constructs a spanning subgraph
    - Add edges with increasing interference until spanner property fulfilled
    - LISE constructs a minimum-interference t-spanner
    - 5-hop spanner with Interference 7

  ![LISE algorithm diagram]

- LocaLISE
  - Constructs a spanner locally
    - Nodes collect \((t/2)\)-neighborhood
    - Locally compute interference-minimal paths guaranteeing spanner property
    - Only request that path to stay in the resulting topology
    - LocalISE constructs a minimum-interference t-spanner

  ![LocaLISE algorithm diagram]
Link-based Interference Model

- LocaLISE (Low Interference Spanner Estisher)
  - Constructs a spanner locally
  - Nodes collect \((t/2)\)-neighborhood
  - Locally compute interference-minimal paths guaranteeing spanner property
  - Only request that path to stay in the resulting topology

LocaLISE constructs a minimum-interference t-spanner

Average-Case Interference: Preserve Connectivity

Average-Case Interference: Spanners

Link-based Interference Model
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 $O(n)$

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

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 $O(\sqrt{n})$

- Two-dimensional node distributions
  - Randomized algorithm resulting in interference $O(\sqrt{n} \log n)$
  - No deterministic algorithm so far...

Towards a More Realistic Interference Model...

- Signal-to-interference and noise ratio (SINR)

  - Minimum signal-to-interference ratio
  - Power level of node $u$
  - Path-loss exponent
  - Noise
  - Distance between two nodes

  $N + \sum_{w \in V \setminus \{u\}} \frac{P_w}{d(w,v)^\alpha} \geq \beta$

- Problem statement
  - Determine a power assignment and a schedule for each node such that all message transmissions are successful

SINR is always assured
Quiz: Can these two links transmit simultaneously?

• Graph-theoretical models: No!
  – Neither in- nor out-interference

• SINR model: constant power: No!
  – Node B will receive the transmission of node C

• SINR model: power according to distance-squared: No!
  – Node D will receive the transmission of node A

Let’s try harder…

• Let’s forget about noise first…
  – Then the SINR at B is: \( \frac{P_A}{100^2} \div \frac{PC}{50^2} = \frac{\rho}{4} \), with \( \rho = \frac{PA}{PC} \).
  – And the SINR at D is: \( \frac{P_C}{12} \div \frac{PA}{50^2} = 2500 \div \rho \).
  – Making both SINR equal, \( \rho^2 = 10000 \) \( \Rightarrow \) \( \rho = 100 \).

• Let’s try with noise \( N = 1, P_A = 100000, PC = 1000 \):
  – Then SINR at B = \( \frac{P_A}{100^2} \div \frac{PC}{50^2} + 1 \) > 7
  – And SINR at D = \( \frac{P_C}{12} \div \frac{PA}{50^2} + 1 \) > 24
  – (Include noise directly, and you get both SINR’s above 10.)

Huge!

A Simple Problem

• Each node in the network wants to send a message to an arbitrary other node
  – Commonly assumed power assignment schemes
    – Constant power level
    – Uniform
    – Linear: Proportional to (receiver distance)^2

Both lead to a schedule of length \( \Theta(n) \)

– A clever power assignment results in a schedule of length \( \Theta(\log^2 n) \)

This has strong implications to MAC layer protocols