Topology Control

01204525 Wireless Sensor Networks and Internet of Things

Chaiporn Jaikaeo
(chaiporn.j@ku.ac.th)

Department of Computer Engineering
Kasetsart University
Overview

• Motivation, basics
• Power control
• Backbone construction
• Clustering
• Adaptive node activity
Motivation: Dense networks

• Too many collisions/too complex operation for a MAC protocol, too many paths to chose from for a routing protocol, ...

• Idea: Make topology less complex
  ◦ Topology: Which node is able/allowed to communicate with which other nodes
  ◦ Topology control needs to maintain invariants, e.g., connectivity
Options for topology control

- **Topology control**
  - Control **node** activity – deliberately turn on/off nodes
  - Control **link** activity – deliberately use/not use certain links

Flat network – all nodes have essentially same role

Hierarchical network – assign different roles to nodes; exploit that to control node/link activity

Power control

Backbones

Clustering
Flat networks

• Main option: Control transmission power

• Alternative: Selectively discard some links
  ◦ Usually done by introducing hierarchies
Hierarchical Networks

• Construct a **backbone** network
  ◦ Some nodes form a (minimal) **dominating set**
  ◦ Controlling nodes have to be connected (backbone)

• Formally: Given graph $G=(V,E)$, construct $D \subseteq V$ such that

$$\forall v \in V : v \in D \lor \exists d \in D : (v, d) \in E$$
Hierarchical Networks

• Construct **clusters**
  ◦ Each node in exactly one group
    ◦ Except “bridging” nodes
    ◦ Groups can have **clusterheads**

• Typically: all nodes in a cluster are direct neighbors of their clusterhead
  ◦ Clusterheads form both **Independent set** and **Dominating Set**
  ◦ Clusterheads + bridges → Connected Dominating set
Cluster Formulation

• Formally: Given graph $G=(V,E)$, construct $C \subseteq V$ such that

$$\forall v \in V - C : \exists c \in C : (v, c) \in E$$

$$\forall c_1, c_2 \in C : (c_1, c_2) \notin E$$
Aspects of Topology Control

• **Connectivity**

• **Stretch factor** – should be small
  ◦ *Hop stretch factor*: how much longer are paths in $G^0$ than in $G$?
  ◦ *Energy stretch factor*: how much more energy does the most energy-efficient path need?

• **Throughput** – removing nodes/links can reduce throughput, by how much?

• Robustness to mobility

• Algorithm overhead
Example: Maintaining connectivity

- Maintaining connectivity can be very “costly” for a power control approach.
Overview

• Motivation, basics

• *Power control*

• Backbone construction

• Clustering

• Adaptive node activity
Power control – Magic Numbers?

• Idea: Controlling transmission power corresponds to controlling the number of neighbors for a given node

• Is there an “optimal” number of neighbors a node should have?

• Historically, k=6 or k=8 had been suggested as such “magic numbers”
  ◦ However, they optimize progress per hop – they do not guarantee connectivity of the graph!!
Controlling Number of Neighbors

• Assumption: Nodes randomly, uniformly placed, only transmission range is controlled, identical for all nodes, only symmetric links are considered

• Result: For connected network, required number of neighbors per node is $\Theta (\log |V|)$
  ◦ It is not a constant, but depends on the number of nodes!
  ◦ Constants can be bounded
Example: Power Control

• Basic idea for most of the following methods:
  ◦ Given a graph $G=(V,E)$
  ◦ Produce a graph $G^0=(V,E^0)$ that maintains connectivity with fewer edges
  ◦ Assume, e.g., knowledge about node positions
  ◦ Construction should be local (for distributed implementation)
Relative Neighborhood Graph (RNG)

- Edge between nodes $u$ and $v$ if and only if there is no other node $w$ that is closer to either $u$ or $v$

$$\forall u, v \in V : (u, v) \in E' \text{ iff } \nexists w \in V : \max\{d(u, w), d(v, w)\} < d(u, v)$$

- RNG maintains connectivity of the original graph
- Easy to compute locally
- Worst-case spanning ratio is $\Omega(|V|)$
- Average degree is 2.6

Connected only when this region is empty
Gabriel Graph (GG)

- Similar to RNG
- Difference: Smallest circle with nodes u and v on its circumference must only contain node u and v

\[ \forall u, v \in V : (u, v) \in E' \iff \exists w \in V : d^2(u, w) + d^2(v, w) < d^2(u, v) \]

- Properties: Maintains connectivity, worst-case spanning ratio \( \Omega(\sqrt{|V|}) \), worst-case degree \( \Omega(|V|) \)

Connected only when this circle is empty
Centralized Algorithm

• Goal: Find topology control algorithm minimizing the maximum power used by any node

• Idea: Use a centralized, greedy algorithm
  ◦ Initially, all nodes have transmission power 0
  ◦ Connect those two components with the shortest distance between them (raise transmission power accordingly)
  ◦ Remove redundant links

• Based on Kruskal's MST
Topology

1) Connect A-C and B-D

2) Connect A-B

3) Connect C-D

4) Connect C-E and D-F

5) Remove edge A-B
Overview

• Motivation, basics
• Power control
• *Backbone construction*
• Clustering
• Adaptive node activity
Hierarchical Networks – Backbones

• Idea: Select some nodes from the network/graph to form a **backbone**
  ◦ A connected, minimal, dominating set (MDS or MCDS)
  ◦ Dominating nodes control their neighbors
  ◦ Protocols like routing are confronted with a simple topology – from a simple node, route to the backbone, routing in backbone is simple (few nodes)

• Problem: MDS/MCDS is an NP-hard problem
Backbone by growing a tree

• Construct the backbone as a tree, grown iteratively

initialize all nodes’ color to white
pick an arbitrary node and color it grey

while (there are white nodes) {
    pick a grey node \( v \) that has white neighbors
    color the grey node \( v \) black
    \textbf{foreach} white neighbor \( u \) of \( v \) {
        color \( u \) grey
        add \((v, u)\) to tree \( T \)
    }
}
Growing a tree – Example

1:

2:

3:

4:
Start big, make lean

• Idea: start with some, possibly large, connected dominating set, reduce it by removing unnecessary nodes

• Initial construction for dominating set
  ◦ All nodes are initially white
  ◦ Mark any node black that has two neighbors that are not neighbors of each other (they might need to be dominated)
  → Black nodes form a connected dominating set (proof by contradiction); shortest path between ANY two nodes only contains black nodes
Overview

• Motivation, basics
• Power control
• Backbone construction
• Clustering
• Adaptive node activity
Clustering

• Partition nodes into *clusters*

• Many options for details
  ◦ Are there *clusterheads*?
  ◦ May clusterheads be neighbors?
    ◦ If not → **Independent Set**
      \[
      \forall c_1, c_2 \in C : (c_1, c_2) \not\in E
      \]
  ◦ May clusters overlap? Do they have nodes in common?
Further options
- How do clusters communicate?
  - Some nodes need to act as *gateways* between clusters
  - If clusters may not overlap, two nodes need to jointly act as a *distributed gateway*
- Is there a hierarchy of clusters?
Constructing Independent Sets

• Make each node a clusterhead that locally has the largest priority

• Once a node is dominated by a clusterhead, it abstains from local competition, giving other nodes a chance

Init:

Step 1:

Step 2:

Step 3:

Step 4:
Connecting Clusters

• Suppose: Clusterheads have been found

• How to connect the clusters, how to select gateways?

• It suffices for each clusterhead to connect to all other clusterheads that are at most three hops
  ◦ Resulting backbone is connected
Rotating Clusterheads

• Serving as a clusterhead can put additional burdens on a node
  ◦ For MAC coordination, routing, ...

• Let this duty rotate among various members
  ◦ Periodically reelect – useful when energy reserves are used as discriminating attribute
  ◦ LEACH – determine an optimal percentage $P$ of nodes to become clusterheads in a network
Overview

- Motivation, basics
- Power control
- Backbone construction
- Clustering
- *Adaptive node activity*
Adaptive node activity

- Remaining option: Turn some nodes off deliberately
- Only possible if remaining nodes can take over their duties
- Example duty: Packet forwarding
  - Approach: Geographic Adaptive Fidelity (GAF)
- Observation: Any two nodes within a square of length $r < \frac{R}{\sqrt{5}}$ can replace each other with respect to forwarding
  - $R = \text{radio range}$
- Keep only one such node active, let the other sleep
Conclusion

• Various approaches exist to trim the topology of a network to a desired shape

• Most of them bear some non-negligible overhead
  ◦ At least: Some distributed coordination among neighbors, or they require additional information
  ◦ Constructed structures can turn out to be somewhat brittle – overhead might be wasted or even counter-productive

• Benefits have to be carefully weighted against risks for the particular scenario at hand