“Spectral Methods, Sensor Nets and Self-organization”
Last time: spectral methods, eigen-spectrum

- If two distinct graphs have the same eigen-spectrum, they are likely isomorphic (esp for large graphs).

- Eigenvalues: degeneracy of $\lambda = 1$ tells us how many disconnected components in the graph.
Summary: spectral methods, measures

- **Mixing time** (time to forget where the walk started)
- **Relaxation time** (related to mixing time, gives bounds)
- **Cover time** (time to occupy each node)
- **Spectral gap**: the largest mixing time, \( t_{\text{max}} = -1/\ln(\lambda_2) \)
  - the larger \( t_{\text{max}} \) the longer it takes for a random walk to cover the graph.
  - the larger \( t_{\text{max}} \) the more accurately a graph can be partitioned into two pieces.
Applications: Wireless sensor networks

- Start with isolated sensor distributed at random.
- Is there a local way to build up global connectivity?
- Locality — why?
Locality

1. Locality $\sim$ distributed

2. Adapt quickly to changing environment

3. Minimal growth in overhead with increasing system size

4. “Self-organizing”
“self-organization”

- Not quantitatively defined.
- (Wikipedia:) Self-organization is a process in which the internal organization of a system, normally an open system, increases in complexity without being guided or managed by an outside source. Self-organizing systems typically (though not always) display emergent properties.
Beaconing
A geometric graph problem

One idea — percolation

Call the graph describing connectivity of nodes: $G_R$
Is this a local algorithm?

(How to determine $R_c$?)
How to determine $R_c$?

Keep increasing until only one eigenvalue $\lambda = 1$
Percolation

Why is it bad?

- Farthest away node sets operating power for all
- Need to *communicate* this value $R_c$ (critical operating range)
- Assumes wireless footprint a uniform disk

Why is in good?

- Guarantees full global connectivity. In the asymptotic limit ($N \to \infty$) know how $R_c$ scales with $N$. So for large $N$ can use theoretical estimate rather than $\lambda = 1$ construction.
- Want small range $R$ to conserve power and also reduce interference. Percolation is a “sweet spot” (full connectivity with out too much interference).
Refining percolation graph $G_R$ (also called the “unit disk graph”)

RNG & GG

Relative Neighbor Graph (RNG)
- An edge $(u, v)$ exist if $\forall w, d(u, v) \leq \max(d(u, w), d(w, v))$

Gabriel graph (GG)
- An edge $(u, v)$ exist, if no other vertex $w$ is present within the circle

$\forall w \neq u, v : d^2(u, v) \leq d^2(u, w) + d^2(w, v)$
• Preserve connectivity of $G_R$, but sparser (and also planar).

• There is an R-package that computes the disk, Gabriel and relative neighbor graph given an input set of coordinates (http://rss.acs.unt.edu/Rdoc/library/spdep/html/graphneigh.html)
Planar disk graphs $\Rightarrow$ greedy routing

- What is greedy forward routing?

- Packets are discarded if there is no neighbor which is nearer to the destination node than the current node; otherwise, packets are forwarded to the neighbor which is nearest to the destination node.

- Each node needs to know the locations of itself, its 1-hop neighbors and destination node.

- Pros: easy implement

- Cons: deliverability (stuck in local voids)
Examples

$w_6$ is a local minimum w.r.t. $v$
Theory versus reality
Wireless footprints

Monotonicity

Uniform-disk model

Experimental

[Ganesan, et. al., 2002]
Adaptive Power Topology control
(Local algorithms to build connectivity)

• Percolation (Common power) neglects natural clustering.
  – Too much power consumption and unnecessary interference.
  – Misses certain paths which could optimize traffic.

• How to build up a connected network using only local information?
  – Moreover, want to avoid uniform disk requirement
Adaptive Power Topology control

- Adaptive power topology control (APTC)
  [Wattenhofer, Li, Bahl, and Wang. Infocom 2001]
  [D’Souza, Ramanathan, and Temple Lang. Infocom 2003]

Each node *individually* increases power until it has a neighbor in every $\theta$ sector around it:

Call the graph describing connectivity of nodes: $G_\theta$
Sample topology

Topology control:

Percolation:

\[ \tau_{[\text{min}]} = 118 \ t_0 \]
\[ \tau_{[\text{med}]} = 335 \ t_0 \]
\[ \tau_{[\text{max}]} = 2920 \ t_0 \]

\[ \tau_{[\text{min}]} = 183 \ t_0 \]
\[ \tau_{[\text{med}]} = 459 \ t_0 \]
\[ \tau_{[\text{max}]} = 2590 \ t_0 \]

\[ R = 5.164 \]
\[ R = 4.727 \]
\[ R = 4.524 \]
Beyond the uniform disk model

[D’Souza, Galvin, Moore, Randall, IPSN 2006 ]

- Can use a local geometric $\theta$-constraint to ensure full network connectivity, independent of wireless footprint!

- Requires constraints on boundary nodes. (Carefully deploy boundary nodes so can communicate, or else have hard-wired boundary channel; then interior nodes can be scattered haphazardly).
Proof overview

Theorem 1. If $G_\theta$ satisfies the $\theta$-constraint at every internal node with $\theta < \pi$ and all of the boundary nodes are known to be connected, then $G_\theta$ is fully connected.

Proof: We need only show every internal node $v$ has a path in $G_\theta$ to some node on the boundary.
Comparison to percolation scheme

\[ \tau_{\text{min}} = 109 \ t_o \]
\[ R = 7.513 \]

\[ \tau_{\text{med}} = 604 \ t_o \]
\[ R = 4.61 \]

\[ \tau_{\text{max}} = 5314 \ t_o \]
\[ R = 5.315 \]

\[ \tau_{\text{min}} = 135 \ t_o \]

\[ \tau_{\text{med}} = 316 \ t_o \]

\[ \tau_{\text{max}} = 561 \ t_o \]
How to quantify “better”? 

- Need performance metrics 
- Direct measures: energy consumption
Power Control: Cross-Layer Design Issues

- **Physical Layer**
  - Power control affects quality of signal

- **Link Layer**
  - Power control affects number of clients sharing channel

- **Network Layer**
  - Power control affects topology/routing

- **Transport Layer**
  - Power control changes interference, which causes congestion

- **Application/OS Layer**
  - Power control affects energy consumption

The “protocol stack”
In general, networks layered

- Social networks
- Email networks
- Data networks
- Protocol networks
- Physical networks
Approximating interference

State transition matrix:

\[ M_{ii} = (k_i - 1)/k_i, \text{ for diagonal elements.} \]

\[ M_{ij} = 1/(k_i - 1)k_i, \text{ if an edge exists between } i \text{ and } j. \]
Network self-discovery time
(Using mixing time as a proxy)

\[ \tau = -1/\ln(\lambda_2) \]
Timescale, $\tau_1 = 5314 \ t_o$

Timescale, $\tau_2 = 1092 \ t_o$

Timescale, $\tau_3 = 157 \ t_o$
Comparison of timescales

Adaptive power

$-\frac{1}{\ln(\lambda)}$, compared to exponential density of Common Power

Common power

$-\frac{1}{\ln(\lambda)}$ and exponential density
Comparison of timescales

Scatterplot of timescales
common versus adaptive power
Regimes for routing

Two timescales:

1. $t_{info} = -1 / \ln(\lambda_2)$

2. $t_{network}$: time for network topology to change.

Routing:

- If $t_{network} \ll t_{info}$, network essentially static during packet routing, so build up routing information.

- If $t_{network} \gg t_{info}$, any info on the network topology will be immediately obsolete, so no routing strategy.

Is there a sharp threshold? Or even any way to bound these behaviors? (What routing protocols work best in which regimes?)
Other pressing issues: Sensor networks

- Deployed networks: optimal sensor placement
- Gossip algorithms: spreading shared information quickly through local exchanges.