Wireless Networking
Graph Theory Unplugged

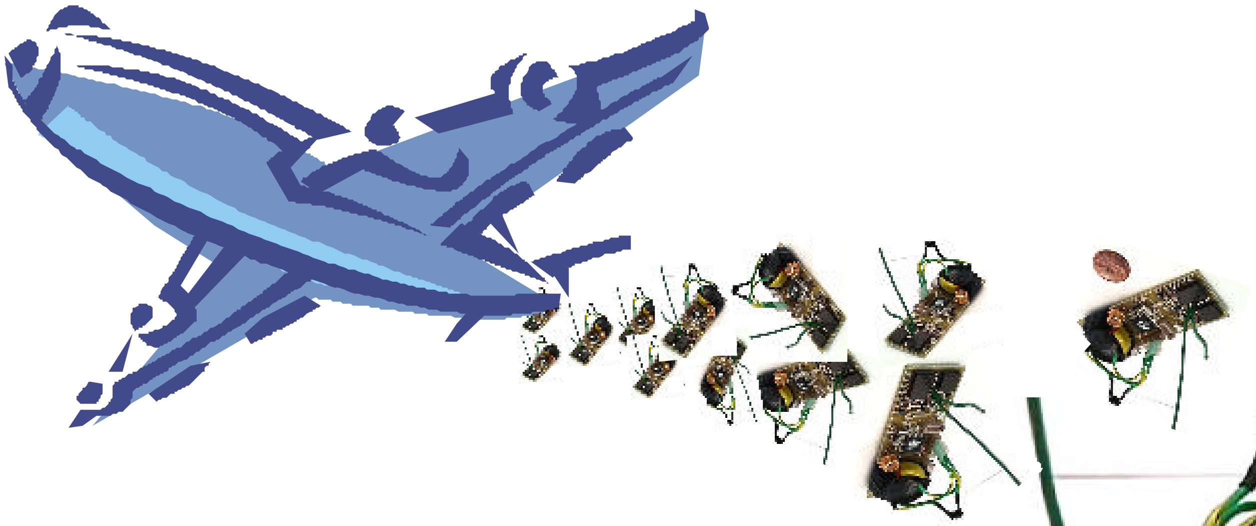
Distributed
Computing
Group
Roger Wattenhofer

WG 2004

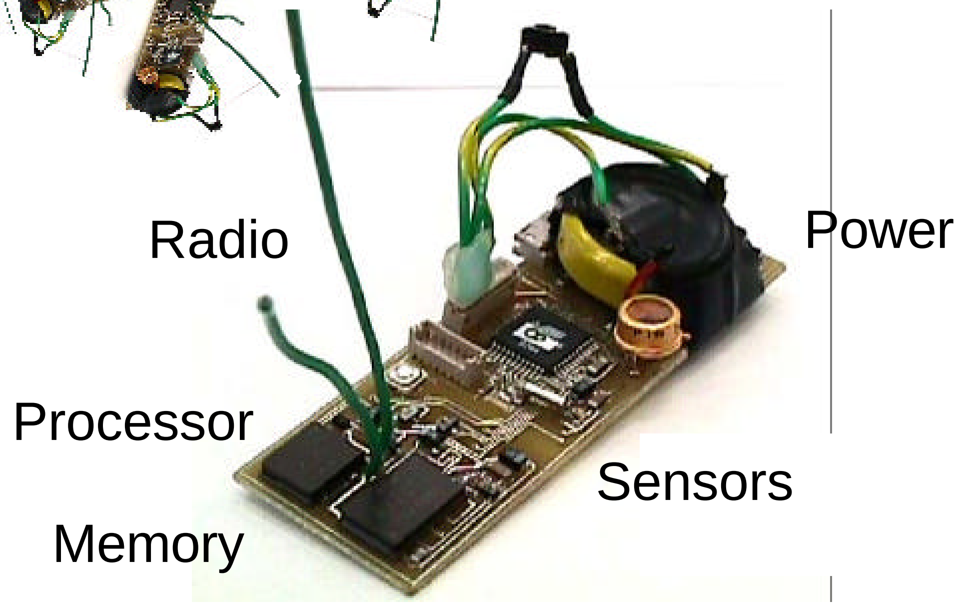
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

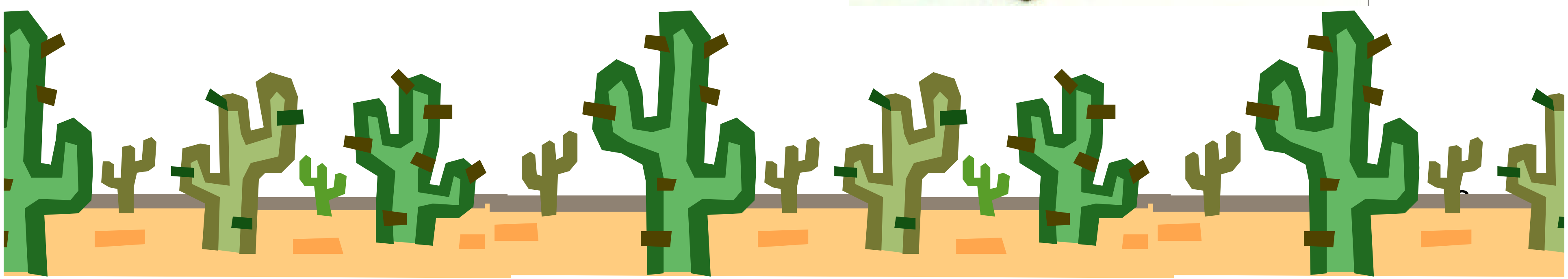
Overview

- Introduction
  - Ad-Hoc and Sensor Networks
  - Routing / Broadcasting
- Clustering
- Topology Control
- Conclusions

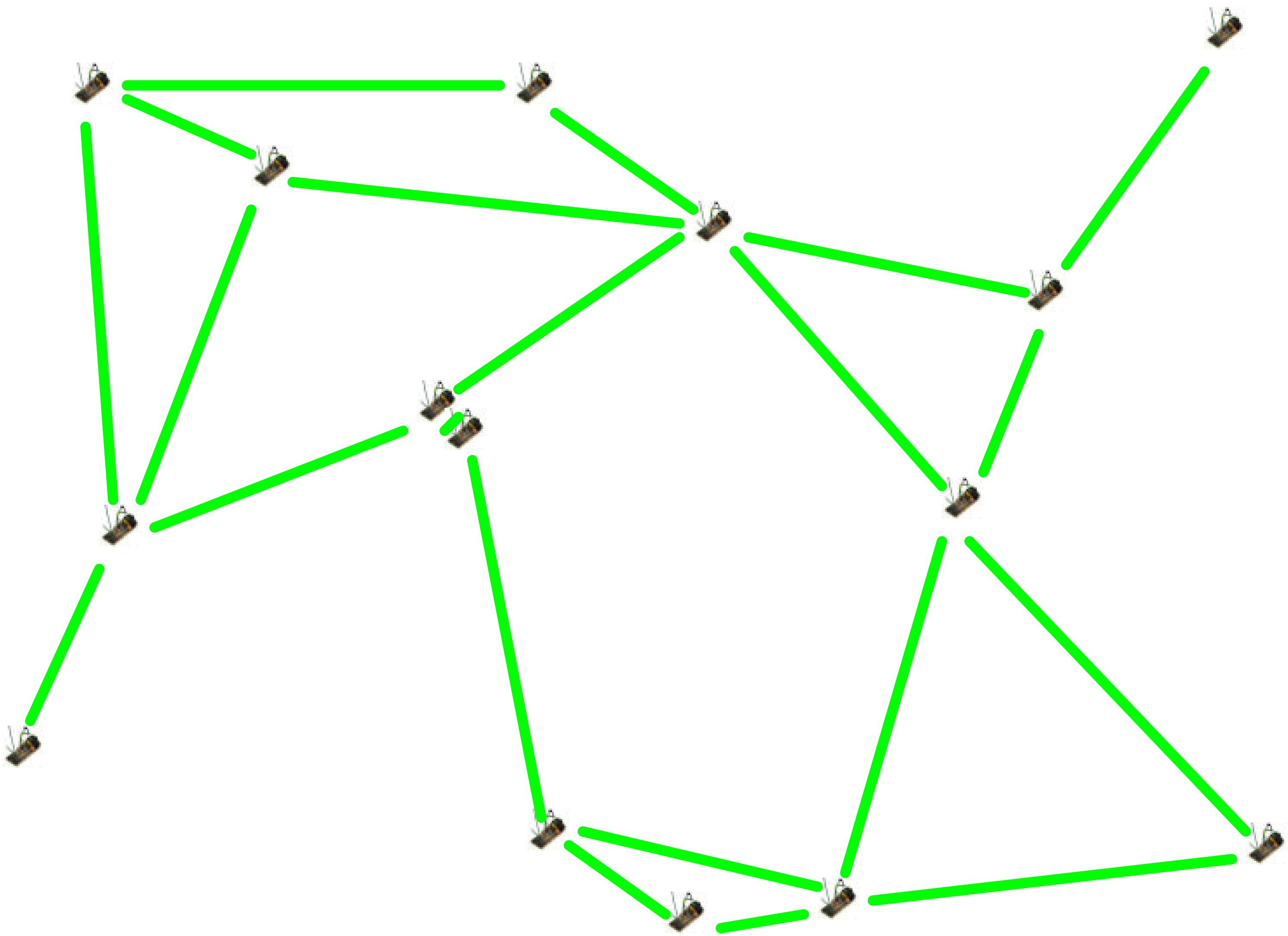


Wireless ad-hoc nodes ("terminodes") are distributed





What are Ad-Hoc/Sensor Networks?



Ad-Hoc Networks vs. Sensor Networks

Ad-Hoc Networks	Sensor Networks
Laptops, PDA’s, cars, soldiers	Tiny nodes: 4 MHz, 32 kB, …
All-to-all routing	Broadcast/Echo from/to sink
Often with mobility (MANET’s)	Usually no mobility – but link failures
Trust/Security an issue – No central coordinator	One administrative control
Maybe high bandwidth	Long lifetime → Energy

Open Problem #1: Positioning and Virtual Coordinates

- Unit Disk Graph: Link if and only if Euclidean distance at most 1.
- Positioning: Some nodes know their position (“anchor nodes”).
- Virtual Coordinates: Unit Disk Graph Embedding
  - Graph Drawing? (Edge crossings → no problem)
  - Known to be NP-hard [Breu & Kirkpatrick 1998]
  - There is no PTAS [Kuhn et al., 2004]
  - Approximation algorithms?
    - Minimize ratio of longest edge over shortest non-edge.
    - Polylogarithmic approximation ratio [Moscibroda et al., 2004]
  - Mobile/dynamic nodes → Local updates, stability

Routing in Ad-Hoc Networks

- Multi-Hop Routing
  - Moving information through a network from a source to a destination if source and destination are not within mutual transmission range
- Reliability
  - Nodes in an ad-hoc network are not 100% reliable
  - Algorithms need to find alternate routes when nodes are failing
- Mobile Ad-Hoc Network (MANET)
  - It is often assumed that the nodes are mobile (“Moteran”)

Simple Classification of Ad-hoc Routing Algorithms

- Proactive Routing

Distance Vector Routing:
as in a fixnet nodes
maintain routing tables
using update messages

- Small topology changes trigger a lot of updates, even when there is no communication
→ does not scale

- Reactive Routing

Flooding:
when node received
message the first time,
forward it to all neighbors

- Flooding the whole network does not scale

no mobility — critical mobility — mobility very high

Source Routing (DSR, AODV):
flooding, but re-use old routes

Discussion

- Lecture “Mobile Computing”: 10 Tricks → 2^{10} routing algorithms
- In reality there are almost that many!

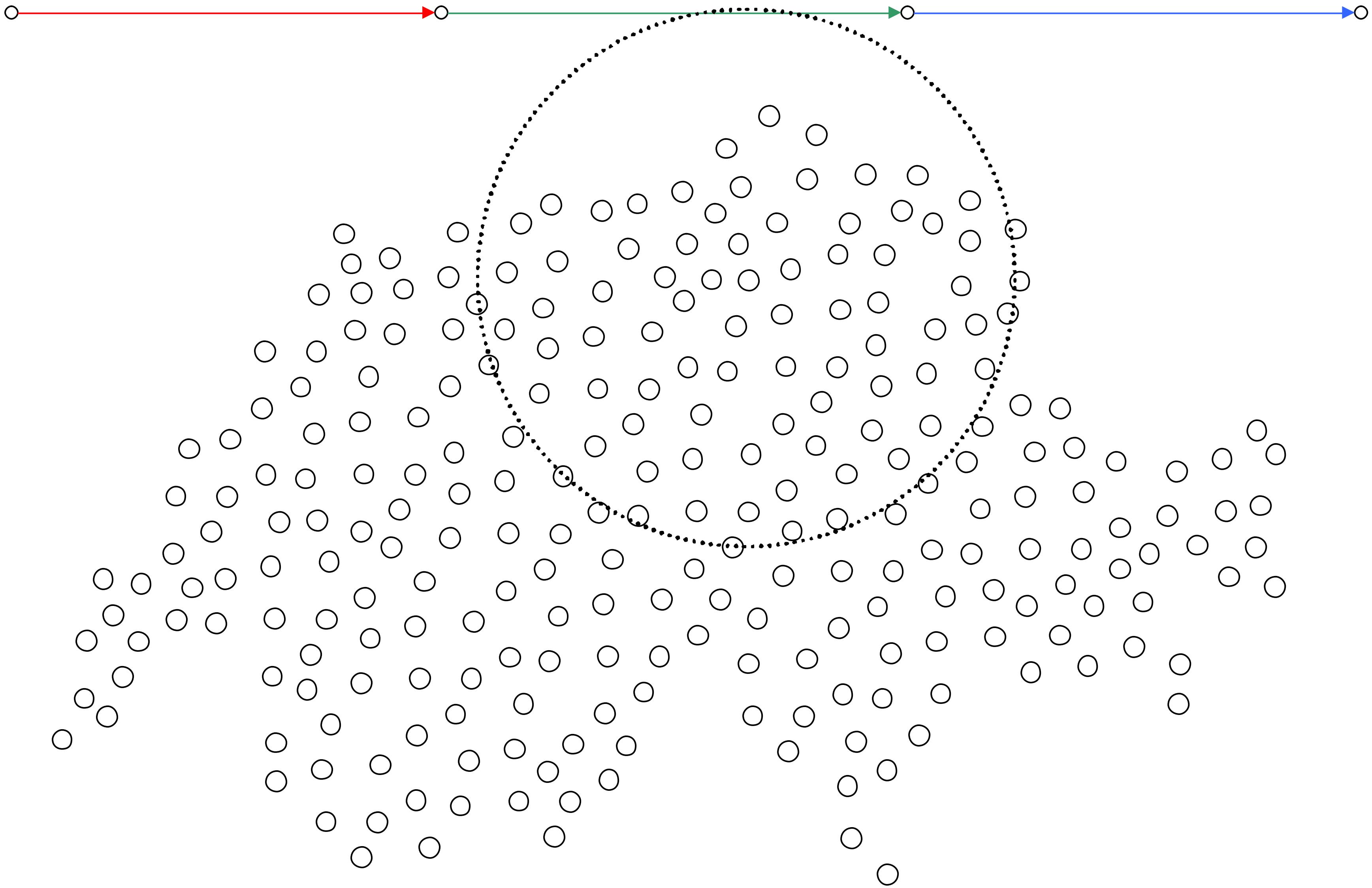
- Q: How good are these routing algorithms?!? Any hard results?
- A: Almost none! Method-of-choice is simulation…
- Perkins: “if you simulate three times, you get three different results”

- Flooding is key component of (many) proposed algorithms
- At least flooding should be efficient

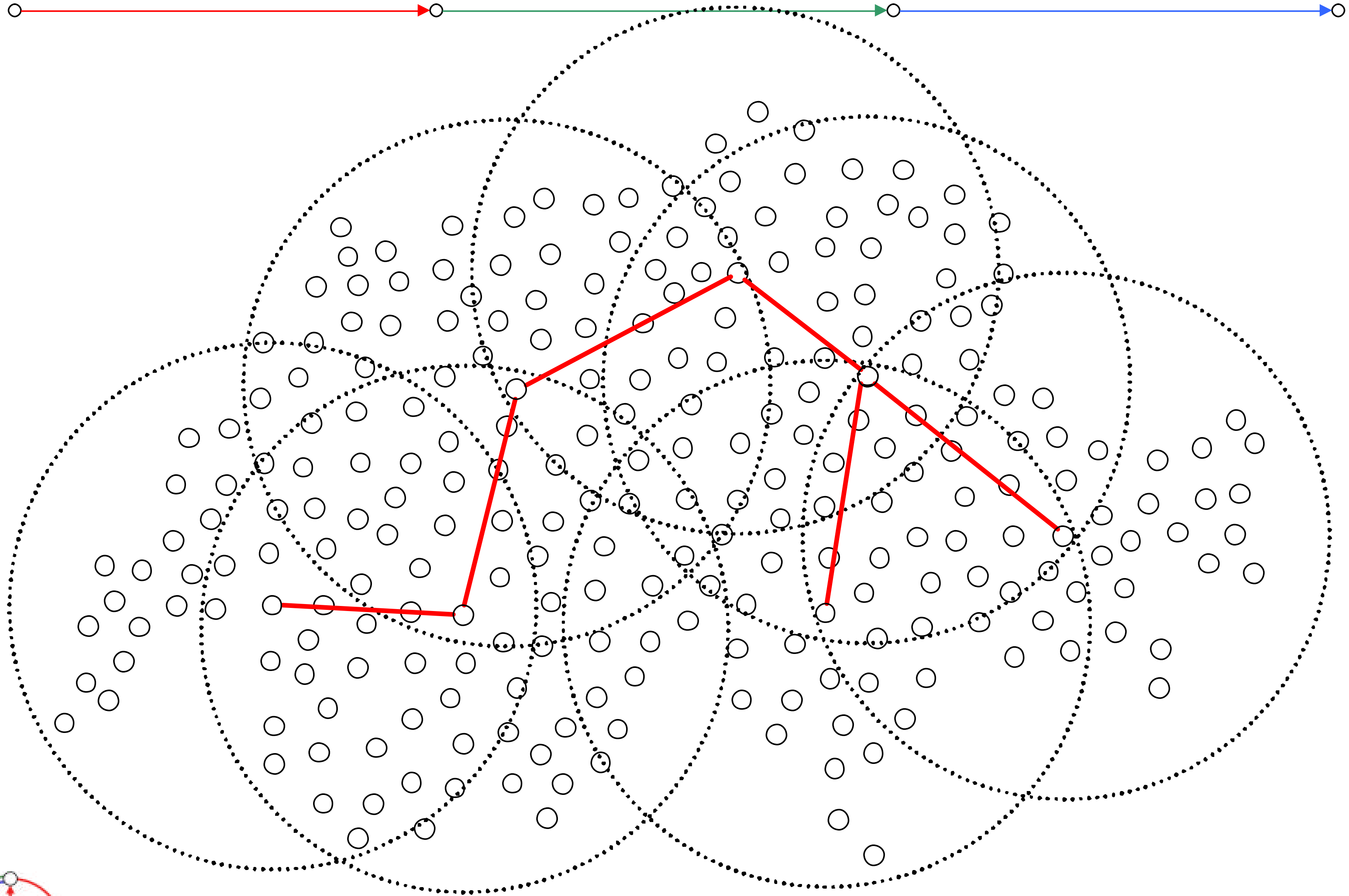
Overview

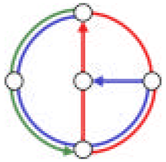
- Introduction
- Clustering
  - Flooding vs. Dominating Sets
  - Algorithm Overview
  - Phase A
  - Phase B
  - Lower Bounds
- Topology Control
- Conclusions

Finding a Destination by Flooding



Finding a Destination *Efficiently*



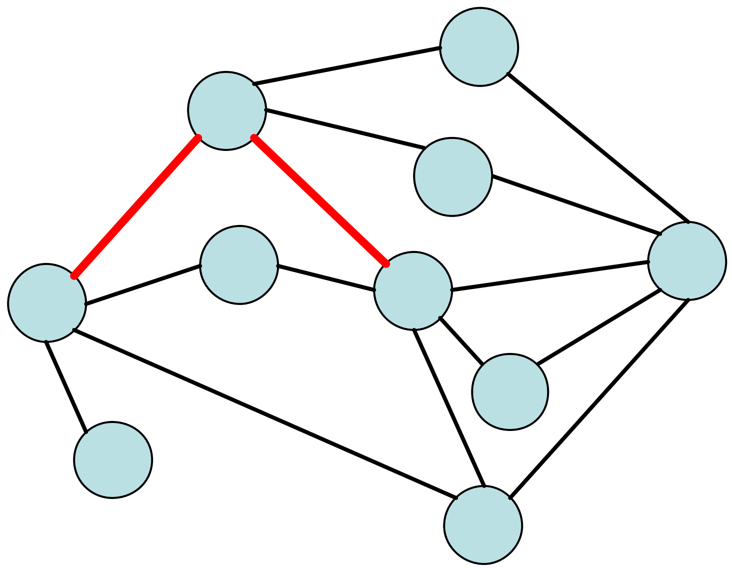


(Connected) Dominating Set

- A Dominating Set DS is a subset of nodes such that each node is either in DS or has a neighbor in DS.

- A Connected Dominating Set CDS is a connected DS, that is, there is a path between any two nodes in CDS that does not use nodes that are not in CDS.

- It might be favorable to have few nodes in the (C)DS. This is known as the Minimum (C)DS problem.



Formal Problem Definition: M(C)DS

- Input: We are given an (arbitrary) undirected graph.
- Output: Find a Minimum (Connected) Dominating Set, that is, a (C)DS with a minimum number of nodes.
- Problems
  - M(C)DS is NP-hard
  - Find a (C)DS that is “close” to minimum (approximation)
  - The solution must be local (global solutions are impractical for mobile ad-hoc network) – topology of graph “far away” should not influence decision who belongs to (C)DS

Overview

- Introduction
- Clustering
  - Flooding vs. Dominating Sets
  - Algorithm Overview
  - Phase A
  - Phase B
  - Lower Bounds
- Topology Control
- Conclusions

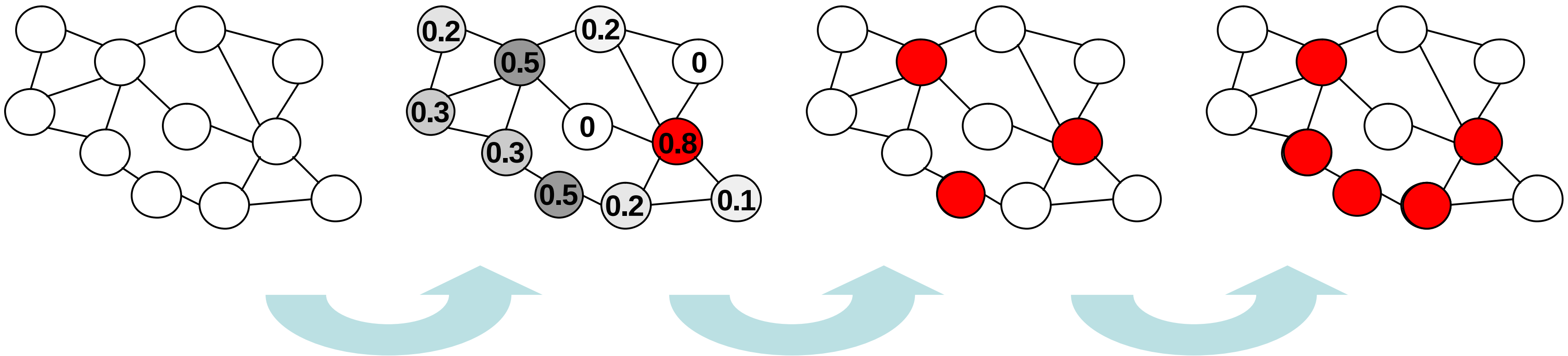
Algorithm Overview

Input: Local Graph

Fractional Dominating Set

Dominating Set

Connected Dominating Set



Phase A:
Distributed linear program
rel. high degree gives high value

Phase B:
Probabilistic algorithm

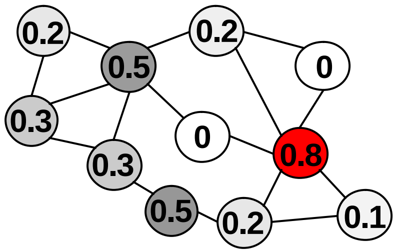
Phase C:
Connect DS by “tree” of “bridges”

Overview

- Introduction
- Clustering
  - Flooding vs. Dominating Sets
  - Algorithm Overview
  - Phase A
  - Phase B
  - Lower Bounds
- Topology Control
- Conclusions

Phase A is a Distributed Linear Program

- Nodes 1, …, *n*: Each node *u* has variable x_u with $x_u \geq 0$
- Sum of *x*-values in each neighborhood at least 1 (local)
- Minimize sum of all *x*-values (global)



0.5+0.3+0.3+0.2+0.2+0 = 1.5 ≥ 1

Linear Program

$$\min \sum_{i=1}^{n} x_i$$
$$\text{subject to} \quad N \cdot \underline{x} \geq \underline{1}$$
$$\underline{x} \geq \underline{0}$$

Adjacency matrix with 1's in diagonal

- Linear Programs can be solved optimally in polynomial time
- But not in a distributed fashion! That's what we do here…

Phase A Algorithm

$x_i := 0;$
calculate $\delta_i^{(2)}$; (* 2 communication rounds *)
$\gamma^{(2)}(v_i) := \delta_i^{(2)} + 1;\ \tilde{\delta}(v_i) := \delta_i + 1;$
for $\ell := k-1$ to 0 by -1 **do**
  (* $\tilde{\delta}(v_i) \lesssim (\Delta + 1)^{(\ell+1)/k}$, $z_i := 0$ *)
  for $m := k-1$ to 0 by -1 **do**
    if $\tilde{\delta}(v_i) \geq \gamma^{(2)}(v_i)^{\frac{\ell}{\ell+1}}$ **then**
      send 'active node' to all neighbors
    fi;
    $a(v_i) := |\{j \in N_i | v_j \text{ is 'active node'}\}|;$
    if color_i = 'gray' **then** $a(v_i) := 0$ **fi**;
    send $a(v_i)$ to all neighbors;
    $a^{(1)}(v_i) := \max_{j \in N_i}\{a(v_j)\};$
    (* $a(v_i), a^{(1)}(v_i) \lesssim (\Delta + 1)^{(m+1)/k}$ *)
    if $\tilde{\delta}(v_i) \geq \gamma^{(2)}(v_i)^{\frac{\ell}{\ell+1}}$ **then**
      $x_i := \max\left\{x_i, a^{(1)}(v_i)^{-\frac{m}{m+1}}\right\}$
    fi;
    send x_i to all neighbors;
    if $\sum_{j \in N_i} x_j \geq 1$ **then** color_i := 'gray' **fi**;
    send color_i to all neighbors;
    $\tilde{\delta}(v_i) := \left|\{j \in N_i \mid \text{color}_j = \text{'white'}\}\right|$
  od;
  (* $z_i \lesssim (1 + (\Delta + 1)^{1/k})/\gamma^{(1)}(v_i)^{\ell/(\ell+1)}$ *)
  send $\tilde{\delta}(v_i)$ to all neighbors;
  $\gamma^{(1)}(v_i) := \max_{j \in N_i}\{\tilde{\delta}(v_j)\};$
  send $\gamma^{(1)}(v_i)$ to all neighbors;
  $\gamma^{(2)}(v_i) := \max_{j \in N_i}\{\gamma^{(1)}(v_j)\}$
od

Result after Phase A

- Distributed Approximation for Linear Program
- Instead of the optimal values x_i^* at nodes, nodes have $x_i^{(\alpha)}$, with

$$\sum_{i=1}^{n} x_i^{(\alpha)} \leq \alpha \cdot \sum_{i=1}^{n} x_i^*$$

- The value of α depends on the number of rounds k (the locality)

$$\alpha \leq \sqrt{k} \cdot (\Delta + 1)^{2/\sqrt{k}}$$

Overview

- Introduction
- Clustering
  - Flooding vs. Dominating Sets
  - Algorithm Overview
  - Phase A
  - Phase B
  - Lower Bounds
- Topology Control
- Conclusions

Dominating Set as Integer Program

- What we have after phase A

$$\begin{array}{lll} \text{min} & \sum_{i=1}^{n} x_i & \\ \text{subject to} & N \cdot \underline{x} \geq \underline{1} & (\text{LP}_{\text{MDS}}) \\ & \underline{x} \geq \underline{0} & \end{array}$$

- What we want after phase B

$$\begin{array}{lll} \text{min} & \sum_{i=1}^{n} x_i & \\ \text{subject to} & N \cdot \underline{x} \geq \underline{1} & (\text{IP}_{\text{MDS}}) \\ & \underline{x} \in \{0,1\}^n & \end{array}$$

Phase B Algorithm

Each node applies the following algorithm:

1. *Calculate* $\delta_i^{(2)}$ *(= maximum degree of neighbors in distance 2)*
2. *Become a dominator (i.e. go to the dominating set) with probability*

$$p_i := \min\{1,\ x_i^{(\alpha)} \cdot \ln(\delta_i^{(2)} + 1)\}$$

From phase A

Highest degree in distance 2

3. *Send status (dominator or not) to all neighbors*
4. *If no neighbor is a dominator, become a dominator yourself*

Result after Phase B

- Randomized rounding technique

- Expected number of nodes joining the dominating set in step 2 is bounded by $\alpha \log(\Delta+1) \cdot |DS_{OPT}|$.

- Expected number of nodes joining the dominating set in step 4 is bounded by $|DS_{OPT}|$.

Theorem: $E[|DS|] \leq O(\alpha \ln \Delta \cdot |DS_{OPT}|)$

Related Work on (Connected) Dominating Sets

- Global algorithms
  - Johnson (1974), Lovasz (1975), Slavik (1996): Greedy is optimal
  - Guha, Kuller (1996): An optimal algorithm for CDS
  - Feige (1998): $\ln \Delta$ lower bound unless $NP \in n^{O(\log \log n)}$

- Local (distributed) algorithms
  - "Handbook of Wireless Networks and Mobile Computing": All algorithms presented have no guarantees
  - Gao, Guibas, Hershberger, Zhang, Zhu (2001): "Discrete Mobile Centers" O(loglog n) time, but nodes know coordinates
  - MIS-based algorithms (e.g. Alzoubi, Wan, Frieder, 2002) that only work on unit disk graphs.
  - Kuhn, Wattenhofer (2003): Tradeoff time vs. approximation

Recent Improvements

- Improved algorithms (Kuhn, Wattenhofer, 2004):
  - $O(\log^2\Delta / \varepsilon^4)$ time for a $(1+\varepsilon)$-approximation of phase A with logarithmic sized messages.
  - If messages can be of unbounded size there is a constant approximation of phase A in $O(\log n)$ time, using the graph decomposition by Linial and Saks.
  - An improved and generalized distributed randomized rounding technique for phase B.
  - Works for quite general linear programs.

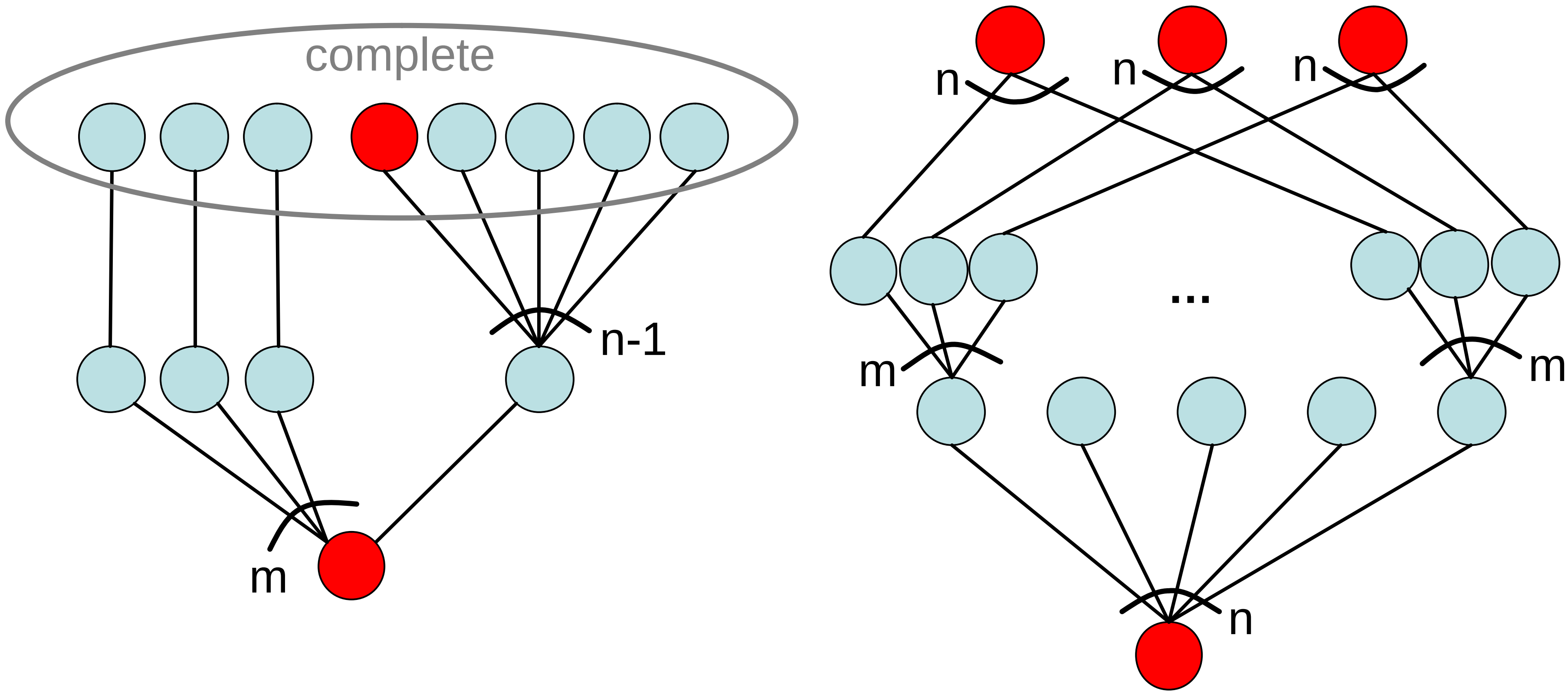
- Is it any good…?

Overview

- Introduction
- Clustering
  - Flooding vs. Dominating Sets
  - Algorithm Overview
  - Phase A
  - Phase B
  - Lower Bounds
- Topology Control
- Conclusions

Lower Bound for Dominating Sets: Intuition…

- Two graphs (m << n). Optimal dominating sets are marked red.

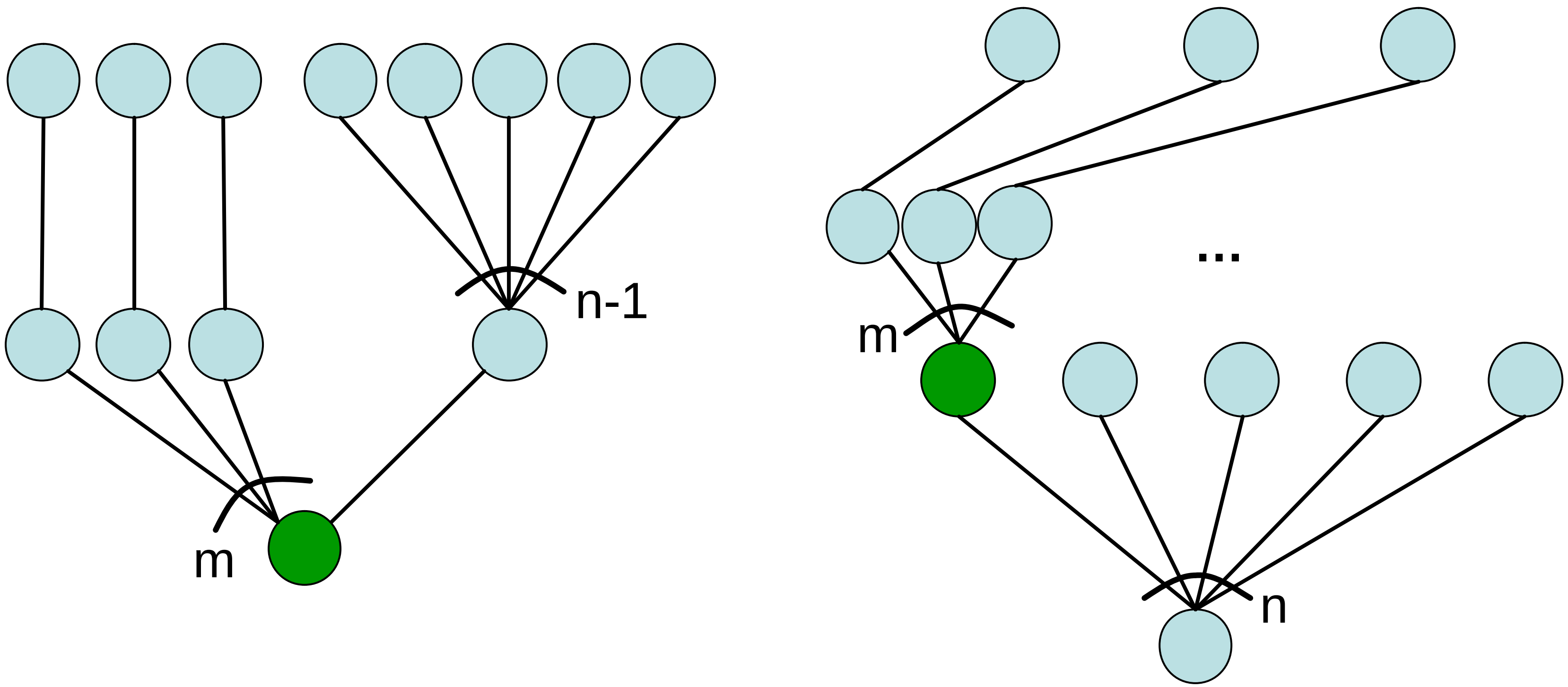


$|DS_{OPT}| = 2.$

$|DS_{OPT}| = m+1.$

Lower Bound for Dominating Sets: Intuition…

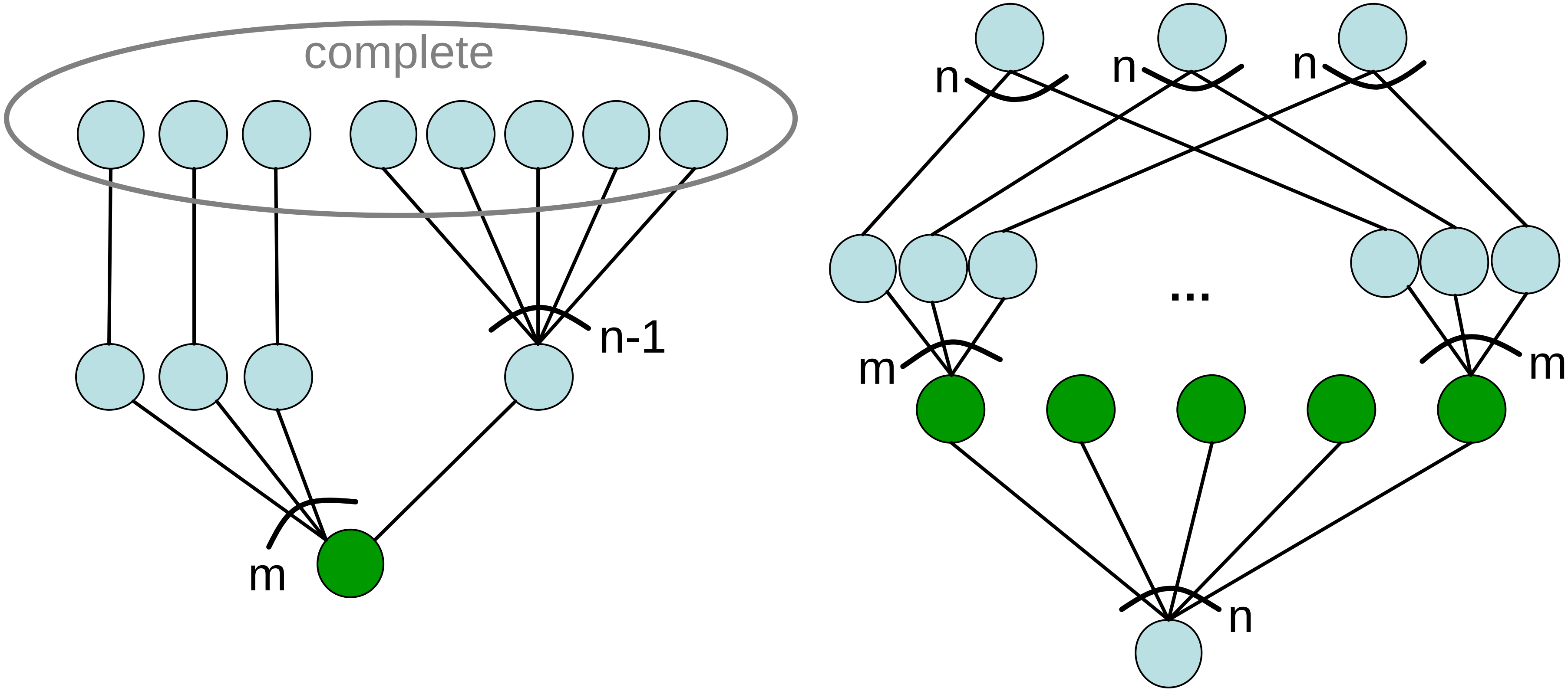
- In local algorithms, nodes must decide only using local knowledge.
- In the example green nodes see exactly the same neighborhood.



- So these green nodes must decide the same way!

Lower Bound for Dominating Sets: Intuition…

- But however they decide, one way will be devastating (with $n = m^2$)!



$|DS_{OPT}| = 2.$

$|DS_{OPT\ without\ green}| \geq m.$

$|DS_{OPT}| = m+1.$

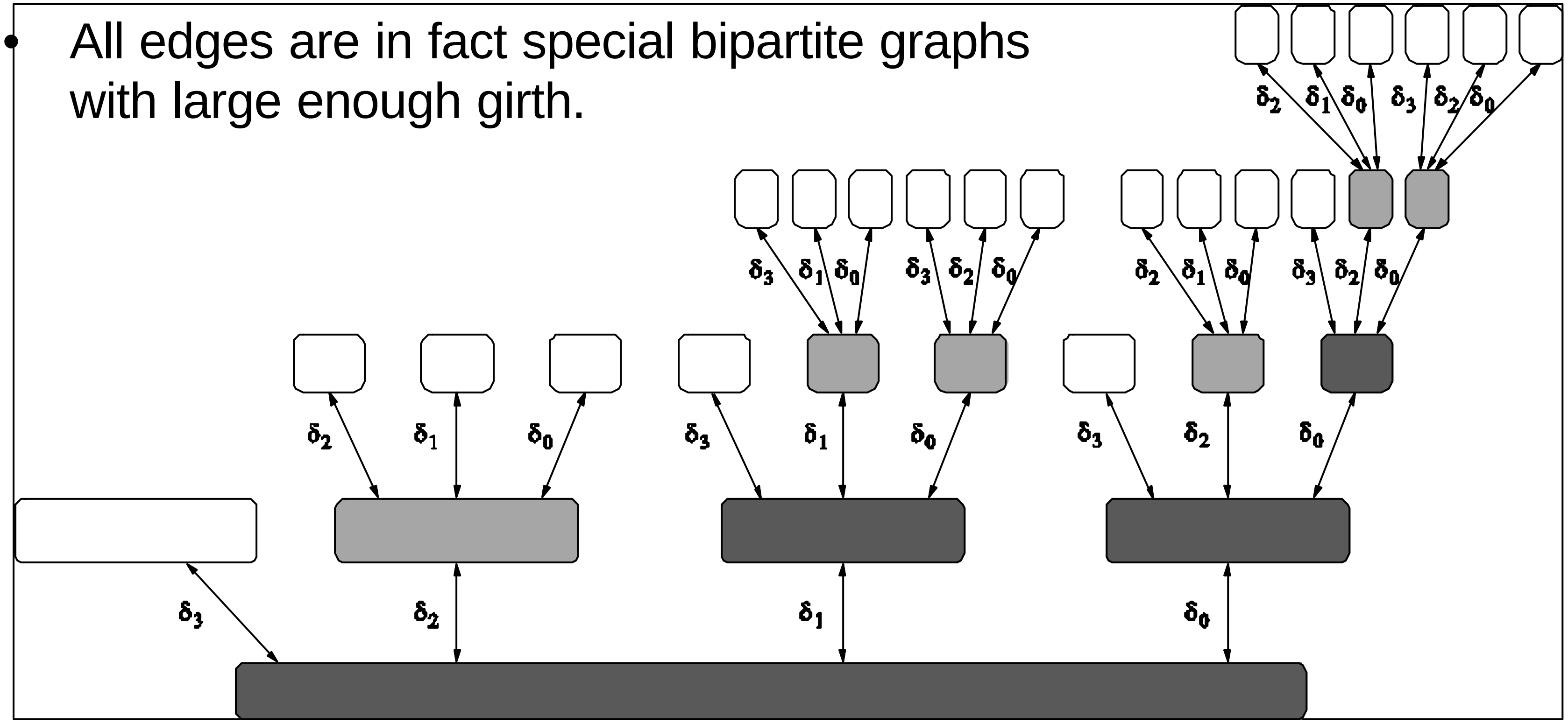
$|DS_{OPT\ with\ green}| > n$

The Lower Bound

- Lower bounds (Kuhn, Moscibroda, Wattenhofer, 2004):
  - Model: In a network/graph G (nodes = processors), each node can exchange a message with all its neighbors for k rounds. After k rounds, node needs to decide.
  - We construct the graph such that there are nodes that see the same neighborhood up to distance k. We show that node ID's do not help, and using Yao's principle also randomization does not.
  - Results: Many problems (vertex cover, dominating set, matching, etc.) can only be approximated $\Omega(n^{c/k^2} / k)$ and/or $\Omega(\Delta^{1/k} / k)$.
  - It follows that a polylogarithmic dominating set approximation (or maximal independent set, etc.) needs at least $\Omega(\log \Delta / \log\log \Delta)$ and/or $\Omega((\log n / \log\log n)^{1/2})$ time.

Graph Used in Dominating Set Lower Bound

- The example is for k = 3.
- All edges are in fact special bipartite graphs with large enough girth.



Clustering for Unstructured Radio Networks

- "Big Bang" (deployment) of a sensor and/or ad-hoc network:
  - Nodes wake up asynchronously (very late, maybe)
  - Neighbors unknown
  - Hidden terminal problem
  - No global clock
  - No established MAC protocol
  - No reliable collision detection
  - Limited knowledge of the number of nodes or degree of network.
- We have randomized algorithms that compute DS (or MIS) in polylog(n) time even under these harsh circumstances, where n is an upper bound on the number of nodes in the system.
- [Kuhn, Moscibroda, Wattenhofer, 2004]

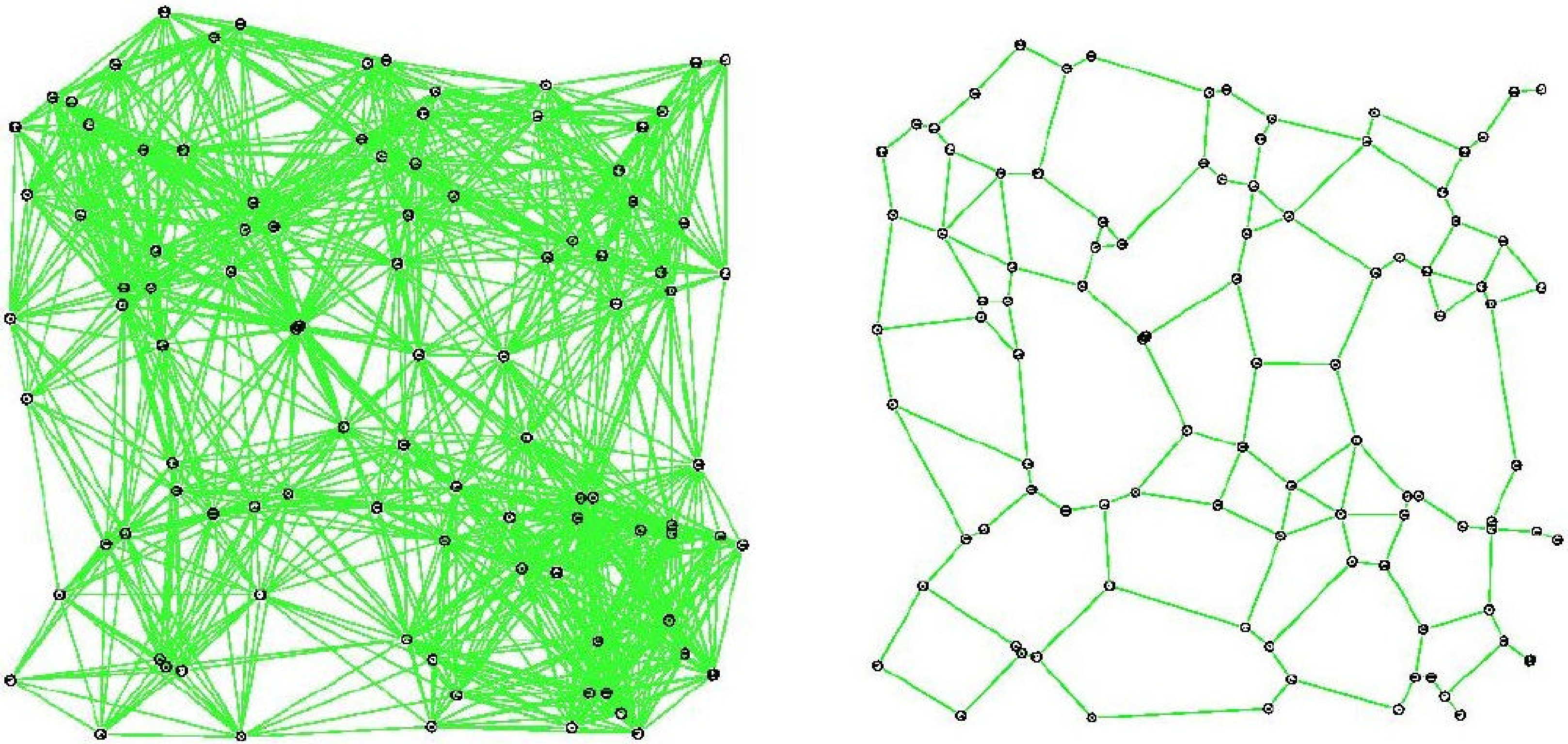
Overview

- Introduction
- Clustering

- Topology Control
  - What is it? What is it good for?
  - Does Topology Control Reduce Interference?
  - Cellular Networks, Sensor Networks, etc.

- Conclusions

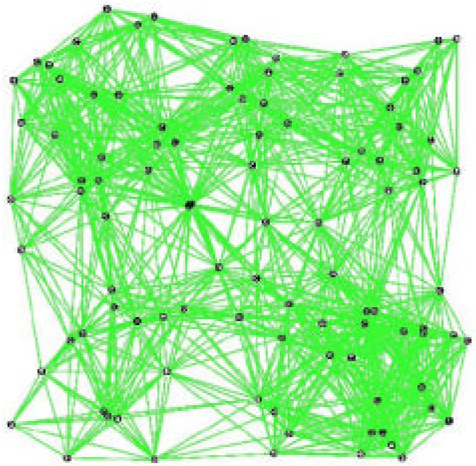
Topology Control



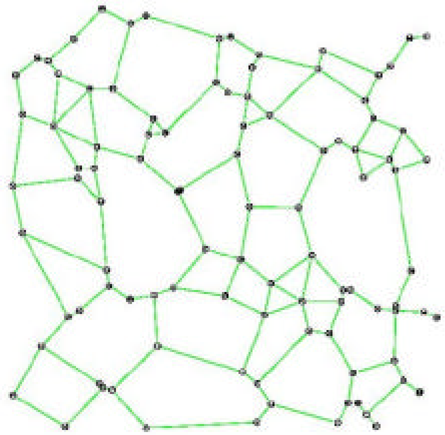
- Drop long-range neighbors: Reduces interference and energy!
- But still stay connected (or even spanner)

Topology Control as a Trade-Off

Sometimes also clustering (first part of the talk) is called topology control



Topology Control



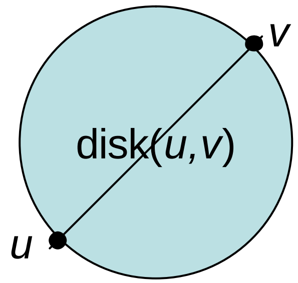
Network Connectivity
Spanner Property

$d(u,v) \cdot t \geq d_{TC}(u,v)$

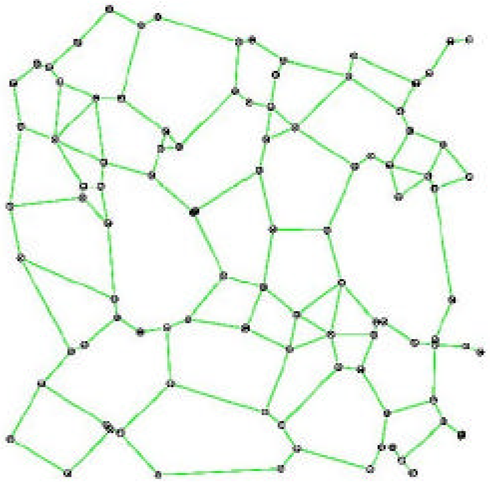
Conserve Energy
Reduce Interference

Classic Solution: Gabriel Graph

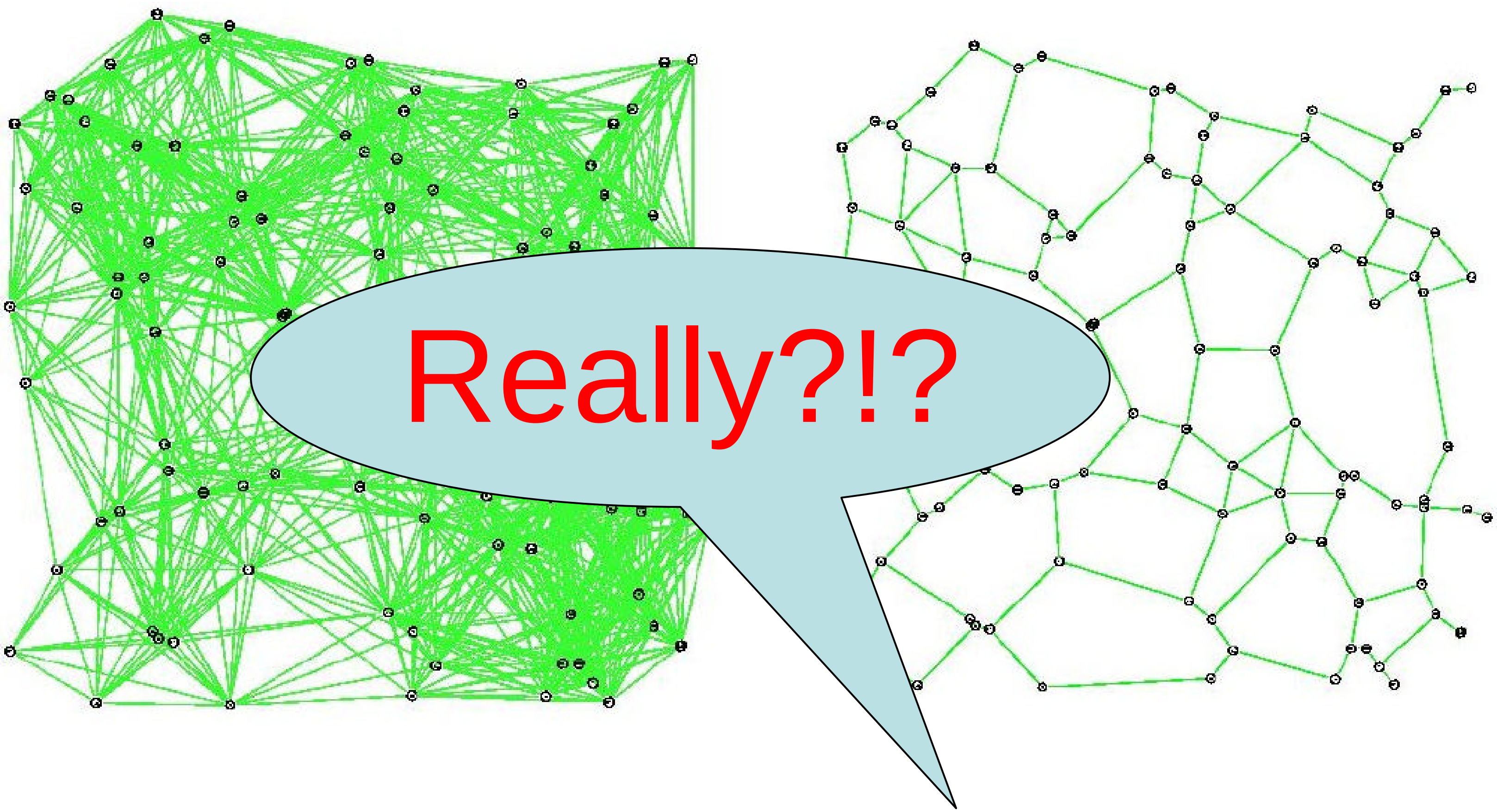
- Let disk(*u*,*v*) be a disk with diameter (*u*,*v*) that is determined by the two points *u*,*v*.
- The Gabriel Graph GG(*V*) is defined as an undirected graph (with *E* being a set of undirected edges). There is an edge between two nodes *u*,*v* iff the disk(*u*,*v*) including boundary contains no other points.



- Gabriel Graph is planar
- Gabriel Graph is energy optimal [energy of link is at least distance squared]



Topology Control



- Drop long-range neighbors: Reduces interference and energy!
- But still stay connected (or even spanner)

Related Work

- Mid-Eighties: randomly distributed nodes [Takagi & Kleinrock 1984, Hou & Li 1986]
- Second Wave: constructions from computational geometry, Delaunay Triangulation [Hu 1993], Minimum Spanning Tree [Ramanathan & Rosales-Hain INFOCOM 2000], Gabriel Graph [Rodoplu & Meng J.Sel.Ar.Com 1999]
- Cone-Based Topology Control [Wattenhofer et al. INFOCOM 2000]; explicitly prove several properties (energy spanner, sparse graph), locality. Collecting more and more properties [Li et al. PODC 2001, Jia et al. SPAA 2003, Li et al. INFOCOM 2002] (e.g. local, planar, distance and energy spanner, constant node degree [Wang & Li DIALM-POMC 2003])
- Explicit interference [Meyer auf der Heide et al. SPAA 2002]. Interference between edges, time-step routing model, congestion; trade-offs; however, interference model based on network traffic

Overview

- Introduction
- Clustering

- Topology Control
  - What is it? What is it good for?
  - Does Topology Control Reduce Interference?
  - Cellular Networks, Sensor Networks, etc.

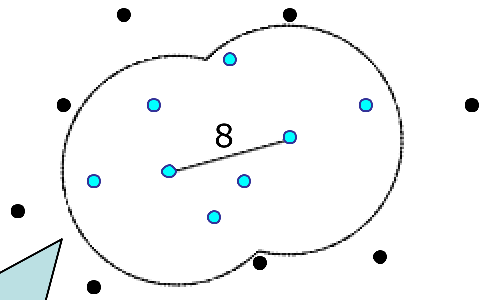
- Conclusions

What Is Interference?

- Model
  - Transmitting edge *e* = (*u*,*v*) disturbs all nodes in vicinity
  - Interference of edge *e* =

    # Nodes covered by union of the two circles
    with center u and v, respectively, and radius |*e*|

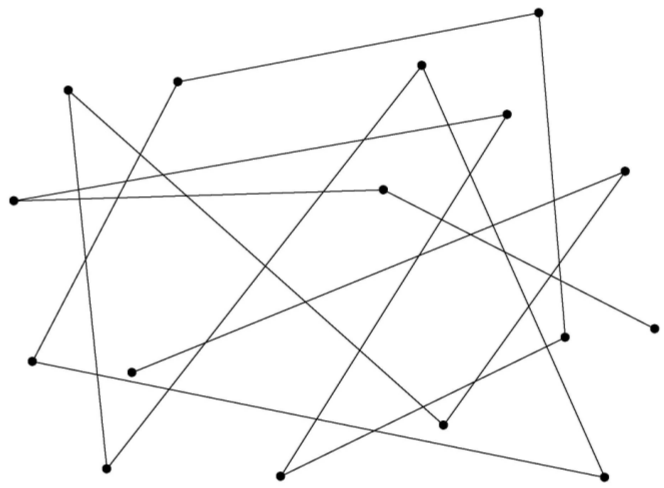
- Problem statement
  - We want to minimize maximum interference!
  - At the same time topology must be connected or a spanner etc.



Exact size of interference range does not change the results

Low Node Degree Topology Control?

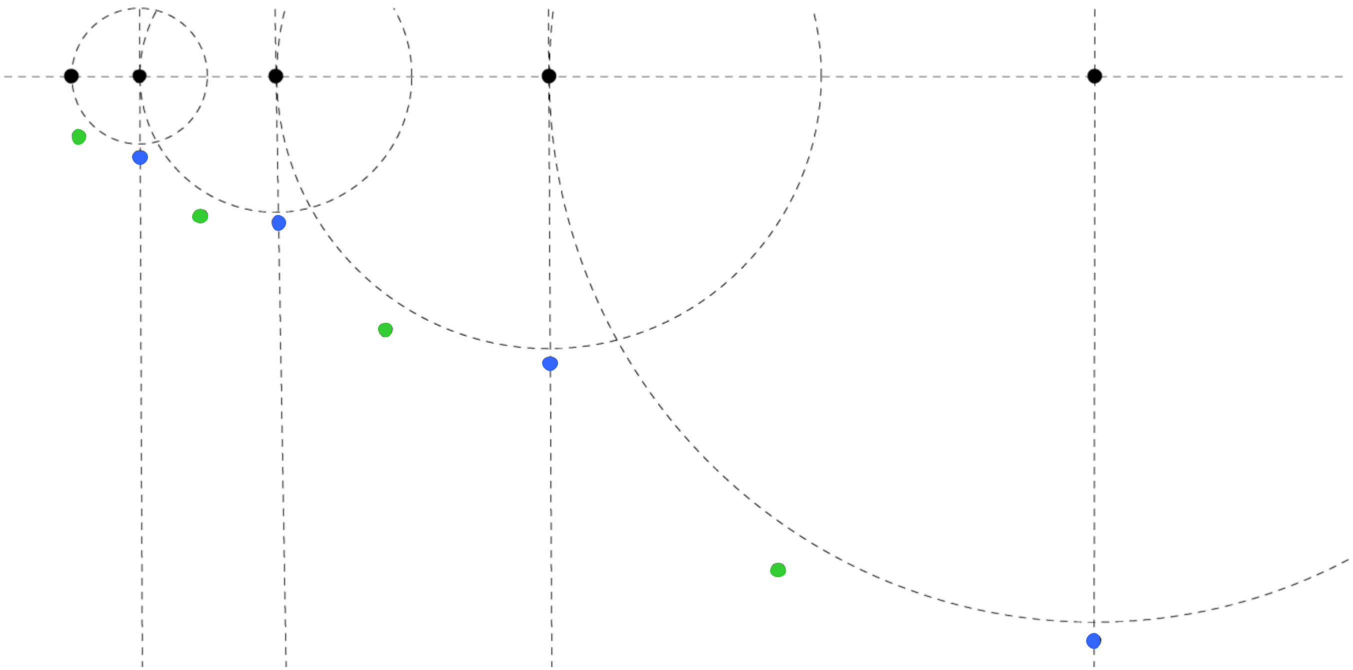
Low node degree does not necessarily imply low interference:



Very low node degree
but huge interference

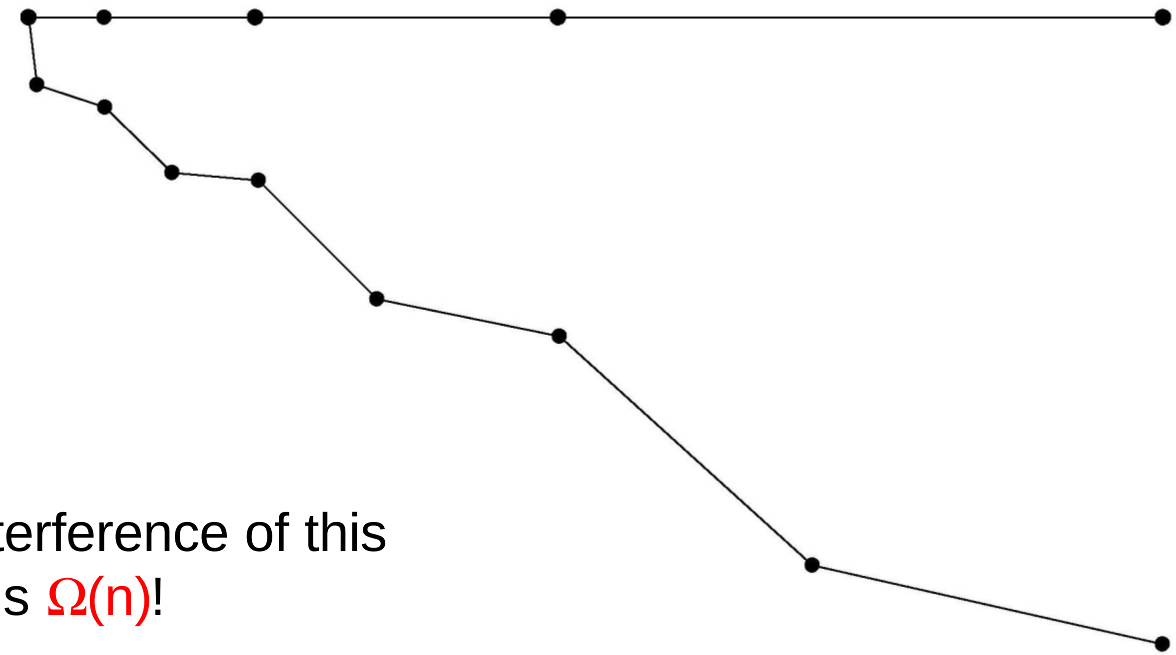
Let's Study the Following Topology!

…from a worst-case perspective



Topology Control Algorithms Produce…

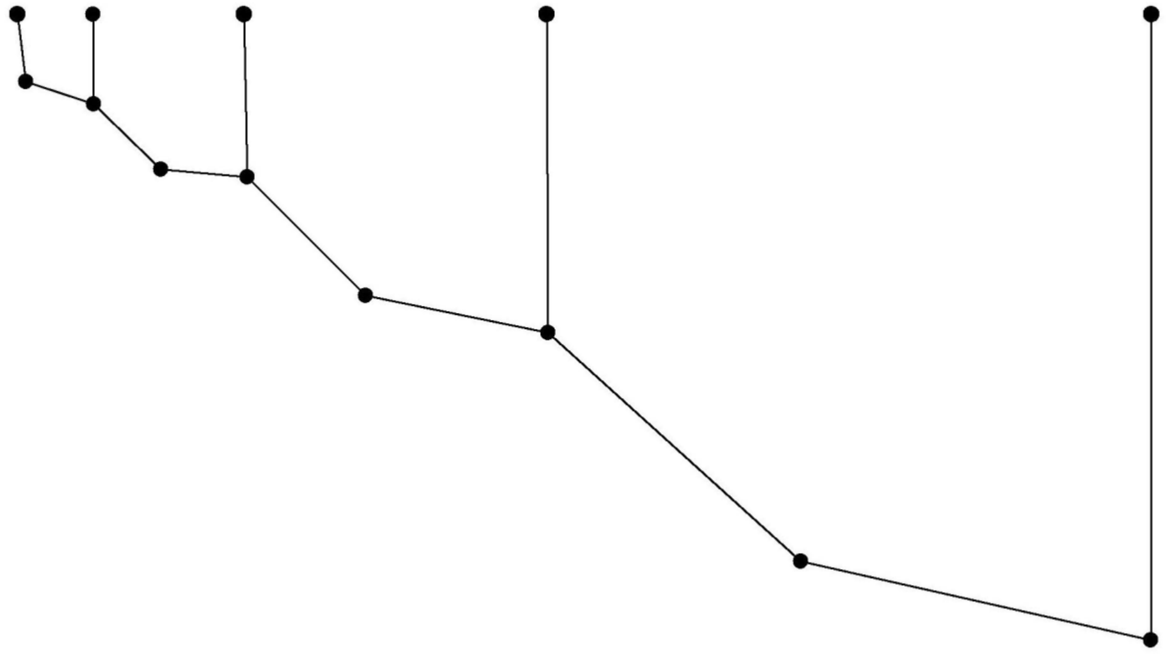
- All known topology control algorithms (with symmetric edges) include the nearest neighbor forest as a subgraph and produce something like this:



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

But Interference…

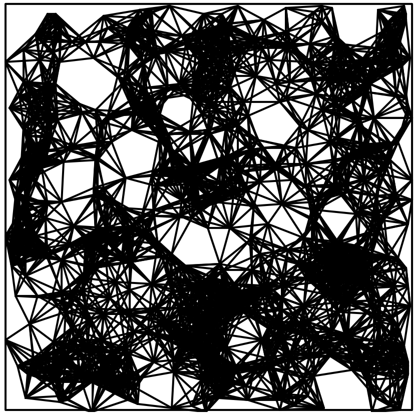
- Interference does not need to be high…



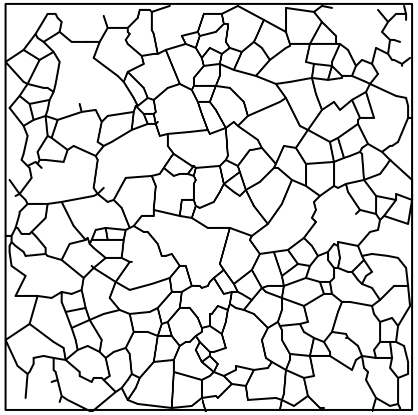
- This topology has interference O(1)!!

Algorithms and Lower Bounds

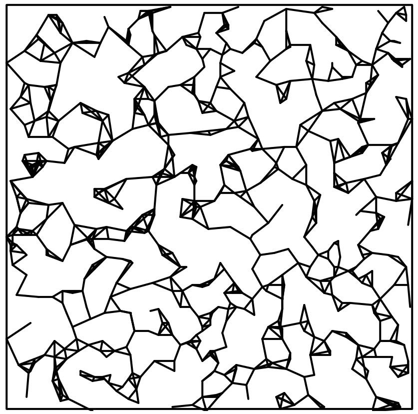
- [Burkhart, von Rickenbach, Wattenhofer, Zollinger, 2004]
- Interference-optimal connectivity-preserving topology
- Local interference-optimal spanner topology
- Algorithms also work if interference radius >> transmission radius
- No local algorithm can find a good topology
- Optimal topology is not planar



UDG, I = 50



RNG, I = 25



LLISE_{10}, I = 12

Overview

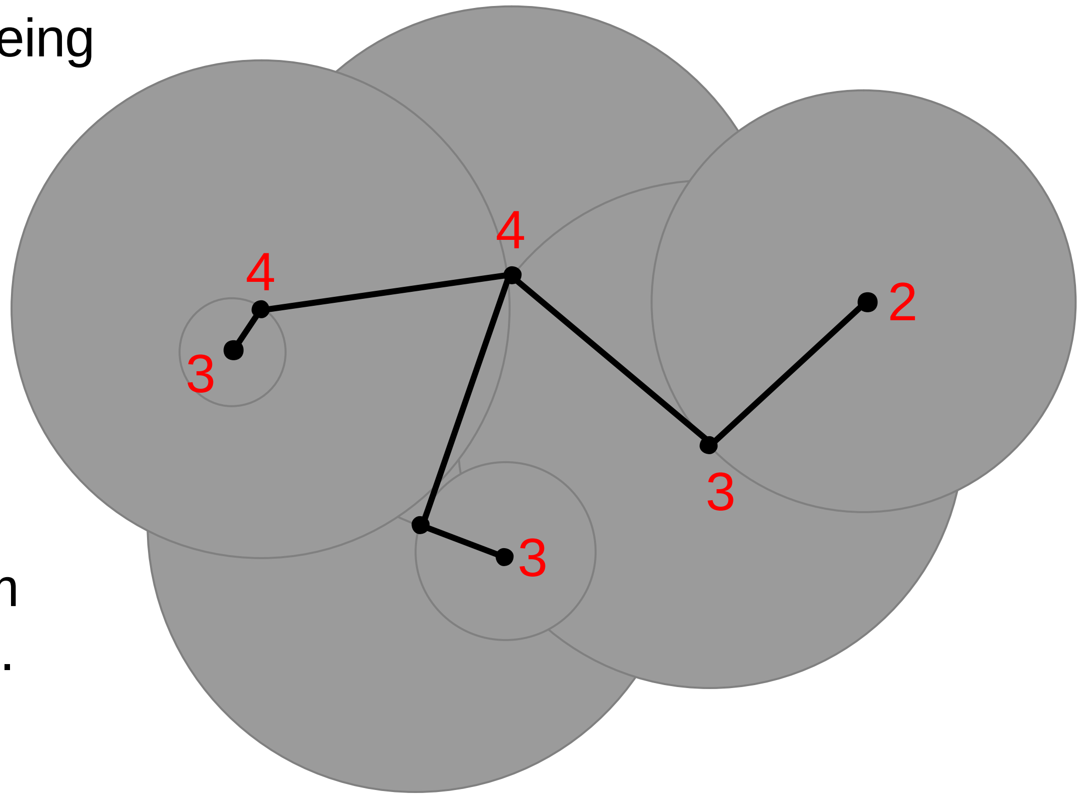
- Introduction
- Clustering
- Topology Control
  - What is it? What is it good for?
  - Does Topology Control Reduce Interference?
  - Cellular Networks, Sensor Networks, etc.
- Conclusions

New Results…

- Interference-driven topology control is exciting new paradigm…
- We have a few other upcoming results:
- For cellular networks: minimize number of base stations a mobile station overhears by reducing the transmission power of the base stations → “minimum membership set cover” problem
- For sensor networks: data gathering without listening to lots of unwanted traffic…

Open Problem #2: In-Interference

- Given ad-hoc network represented by nodes in a plane.
- Connect nodes by spanning tree.
- Circle of each node centered at node with the radius being the length of longest adjacent edge in spanning tree.
- Coverage of node is the number of circles node falls into.
- Minimize the maximum (or average) coverage.



Overview

- Introduction
- Clustering
- Topology Control

- Conclusions
  - Clustering vs. Topology Control
  - More realism, more realism, more realism, …
  - … Practice!

Clustering vs. Topology Control

- Clustering
- (Connected) Dominating Set
- (Connected) Domatic Partition

- Topology Control
- Interference-Driven T.C.

Both approaches sparsen the graph in order to reduce energy…

- … by turning off fraction of the nodes, and thus interference.

- … by turning off long-range links, and thus interference.

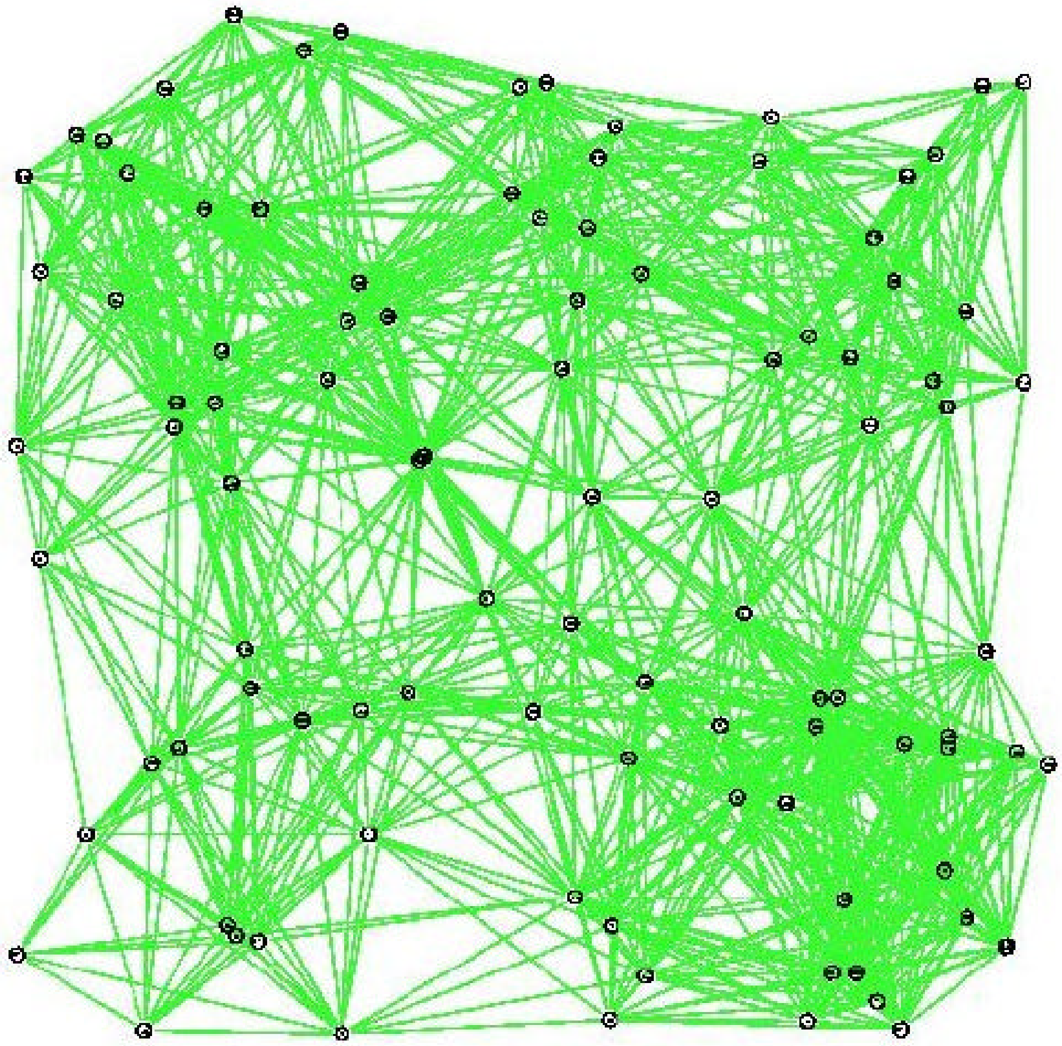
Two sides of the same medal?

Overview

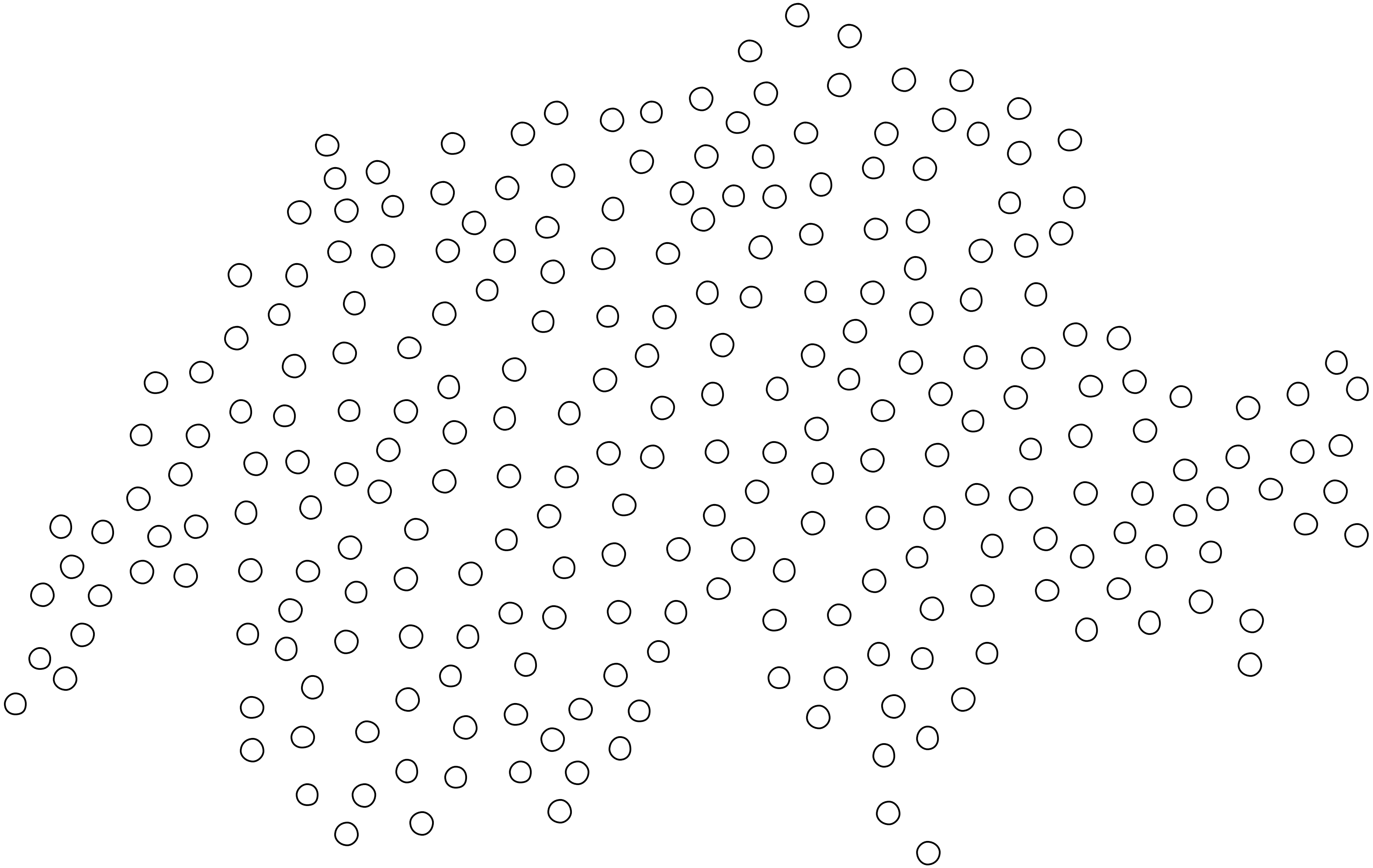
- Introduction
- Clustering
- Topology Control
- Conclusions
  - Clustering vs. Topology Control
  - More realism, more realism, more realism, …
  - … Practice!

What does a ***typical*** ad-hoc network look like?

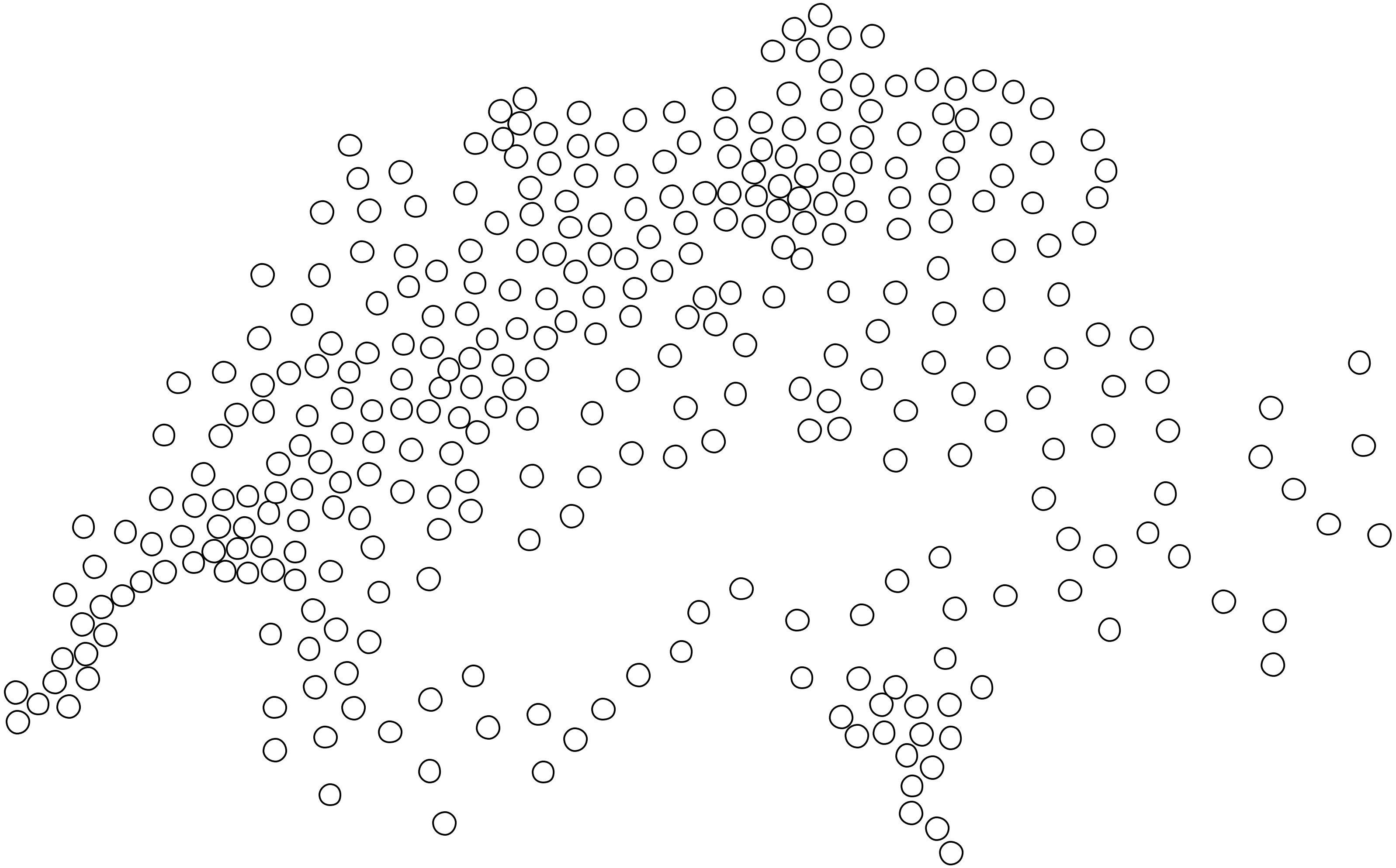
Like this?

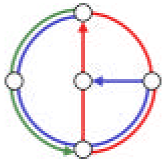


Like this?

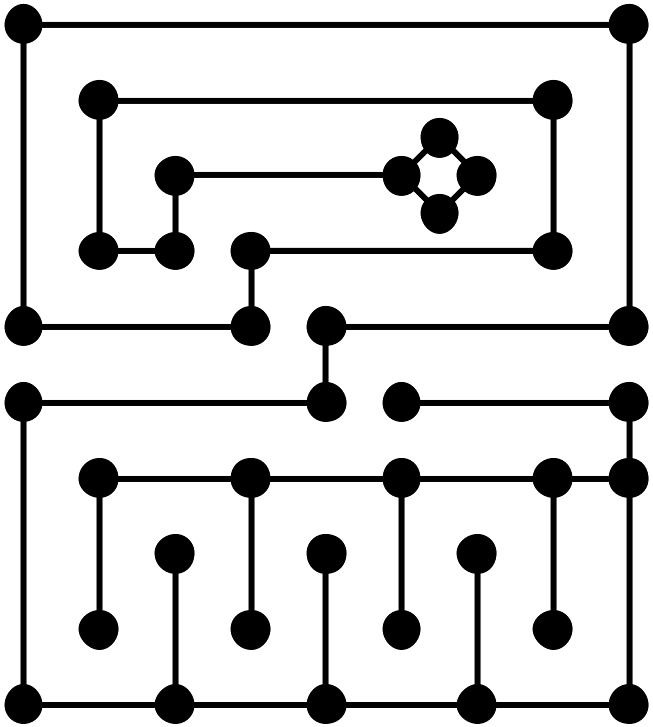


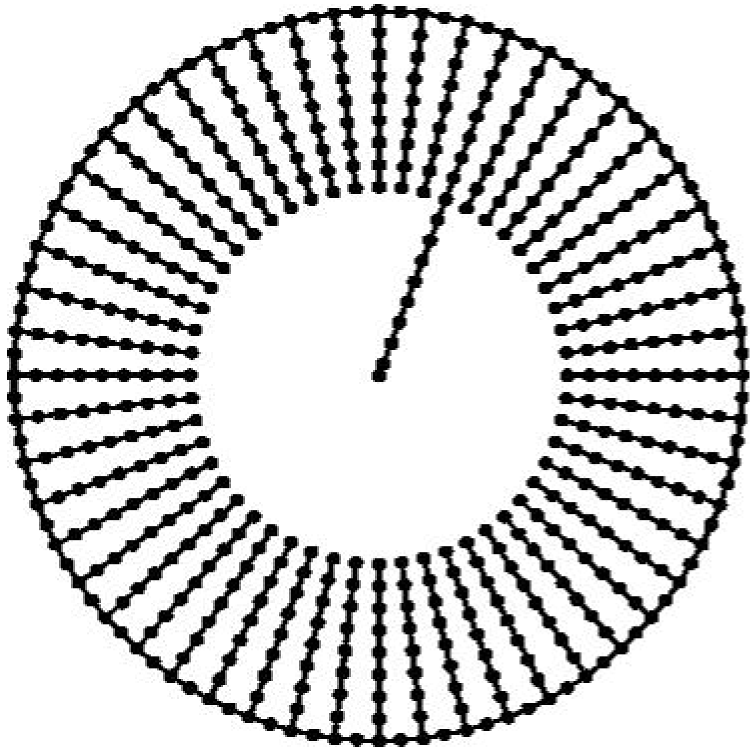
Or rather like this?





Or even like this?





What about ***typical*** mobility?

- Brownian Motion?
- Random Way-Point?
- Statistical Data Model?
- Maximum Speed Model?
- …?

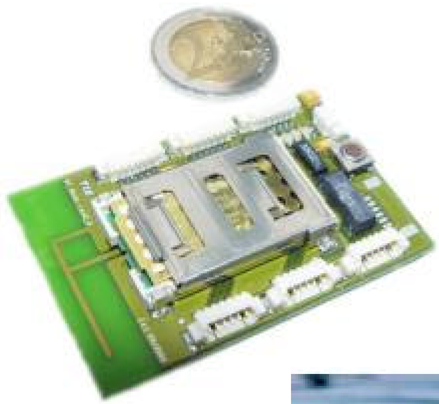
Overview

- Introduction
- Clustering
- Topology Control

- Conclusions
  - Clustering vs. Topology Control
  - More realism, more realism, more realism, …
  - … Practice!

Combine Theory with Practice

- Practical experiments…



btnodes of NCCR/MICS

Scatterweb

Shockfish

Some credit…

Problem (Algorithm)	Reference
Dominating Set	Kuhn, W. @ PODC 2003 Kuhn, Moscibroda, W. @ PODC 2004
Topology Control and Interference (LISE, XTC, MMSC, Sensor Networks)	Burkhart et al. @ MobiHoc 2004 W., Zollinger @ WMAN 2004 von Rickenbach et al. @ submitted Zollinger et al. @ submitted
Geo-Routing (GOAFR)	Kuhn, W., Zollinger @ MobiHoc 2003 Kuhn, W., Zhang, Zollinger @ PODC 2003
Positioning (GHoST)	Bischoff, W. @ PerCom 2004 Kuhn, Moscibroda, W. @ DIALM 2004 Moscibroda et al. @ DIALM 2004.
Data gathering	Cristescu et al. @ submitted von Rickenbach, W. @ DIALM 2004
Models: Quasi-UDG	Kuhn, W., Zollinger @ DIALM 2003
“Big Bang” problem	Moscibroda, Kuhn, W. @ ESA & MobiCom 2004

Questions?
Comments?

Distributed Computing Group
Roger Wattenhofer

Thanks to my students Fabian Kuhn, Aaron Zollinger, Regina Bischoff, Thomas Moscibroda, Pascal von Rickenbach, Martin Burkhart, etc.