Initializing Newly Deployed Ad Hoc and Sensor Networks

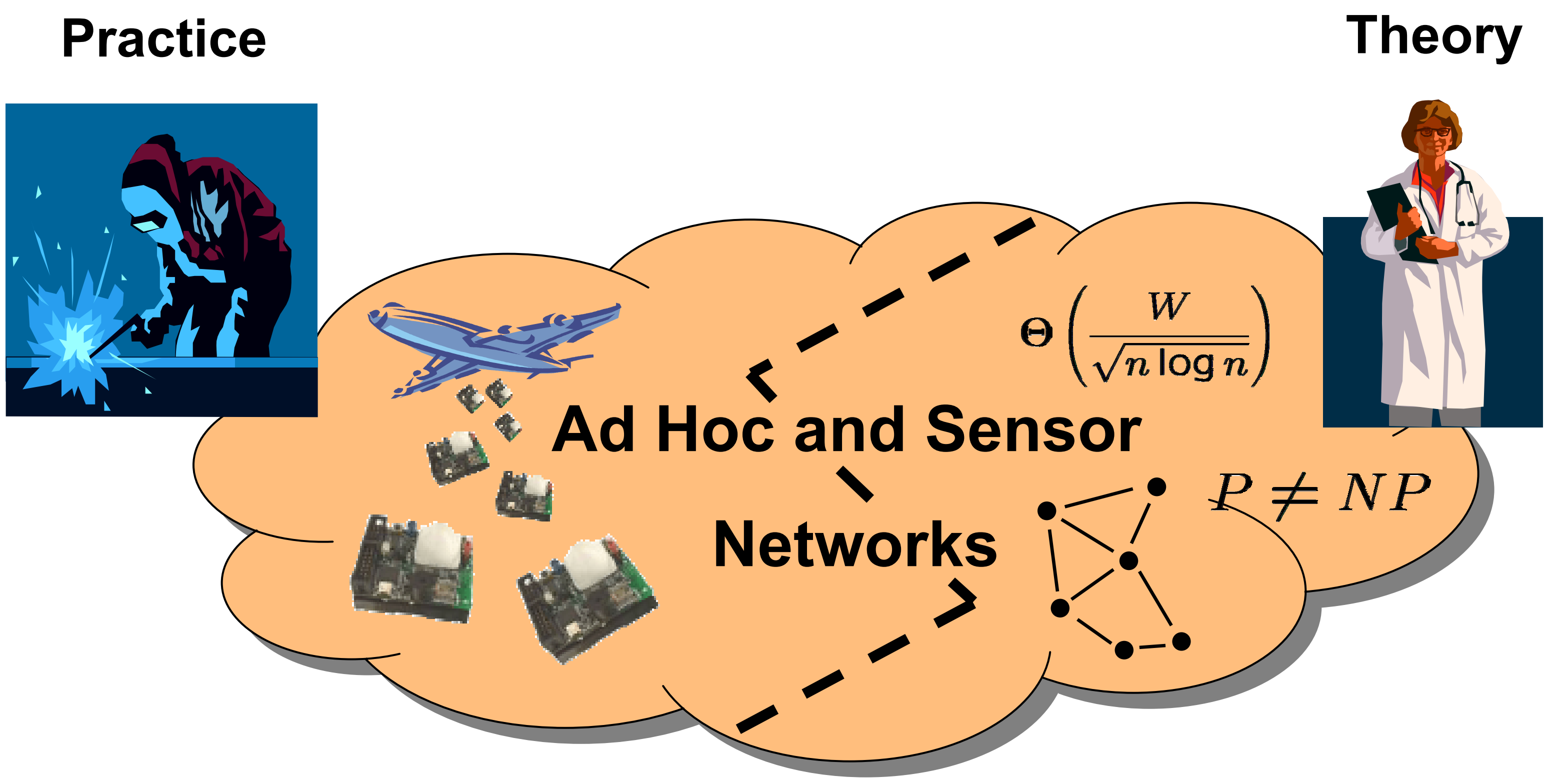
Fabian Kuhn
Thomas Moscibroda
Roger Wattenhofer

Distributed Computing Group

MOBICOM 2004

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

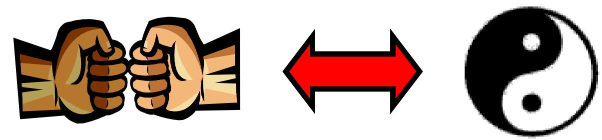
Of Theory and Practice...



There is often a big gap between theory and practice in the field of wireless ad hoc and sensor networks.

Of Theory and Practice...

- What is the reason for this chasm...?



- Theoreticians try to understand the fundamentals
- Need to abstract away a few technicalities...

What are technicalities...???

- Abstracting away too many „technicalities“ renders theory useless for practice!

Random Node Distribution

- Theoreticians often assume that, ...

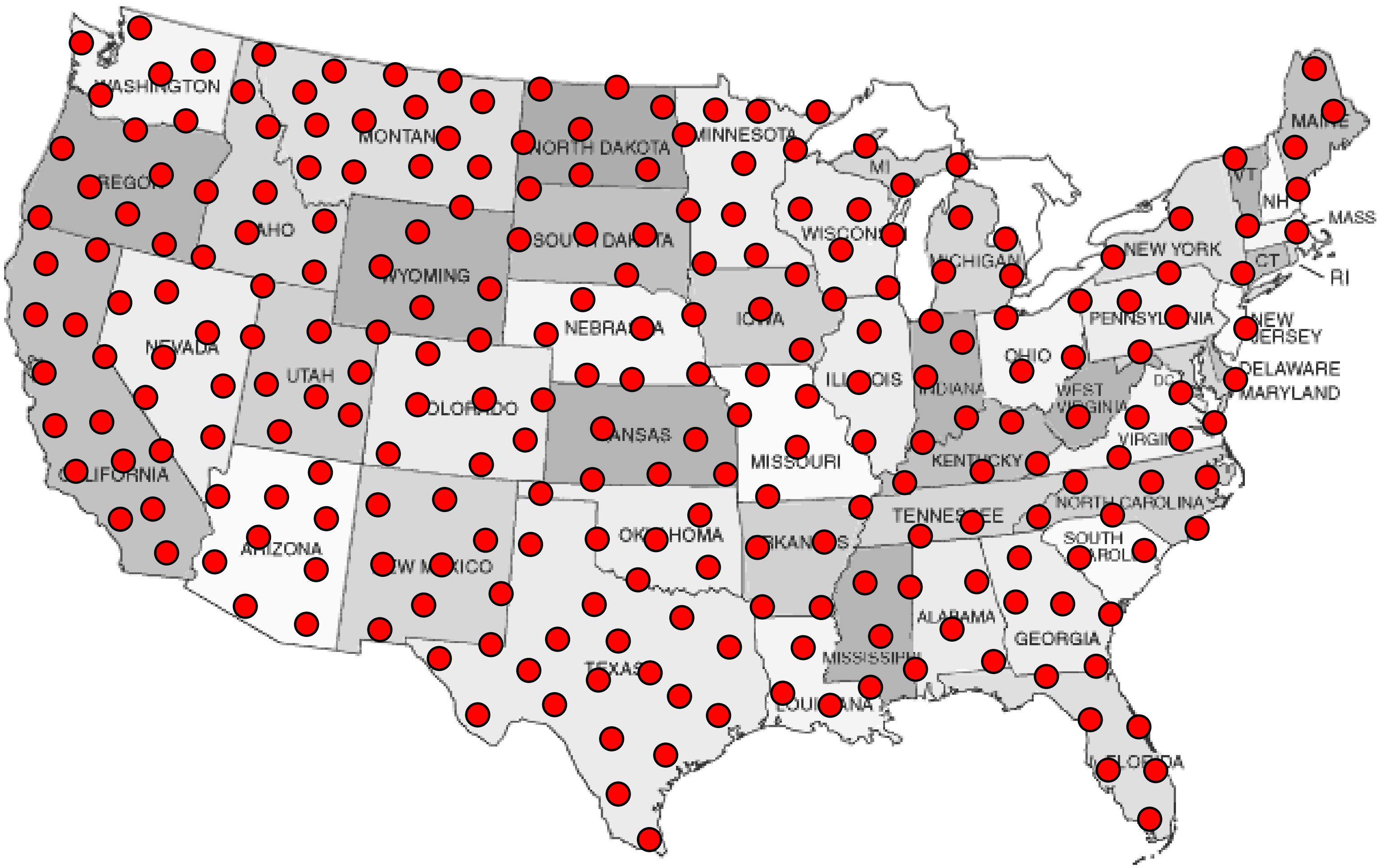
nodes are randomly, uniformly distributed in the plane.

This assumption allows for nice formulas

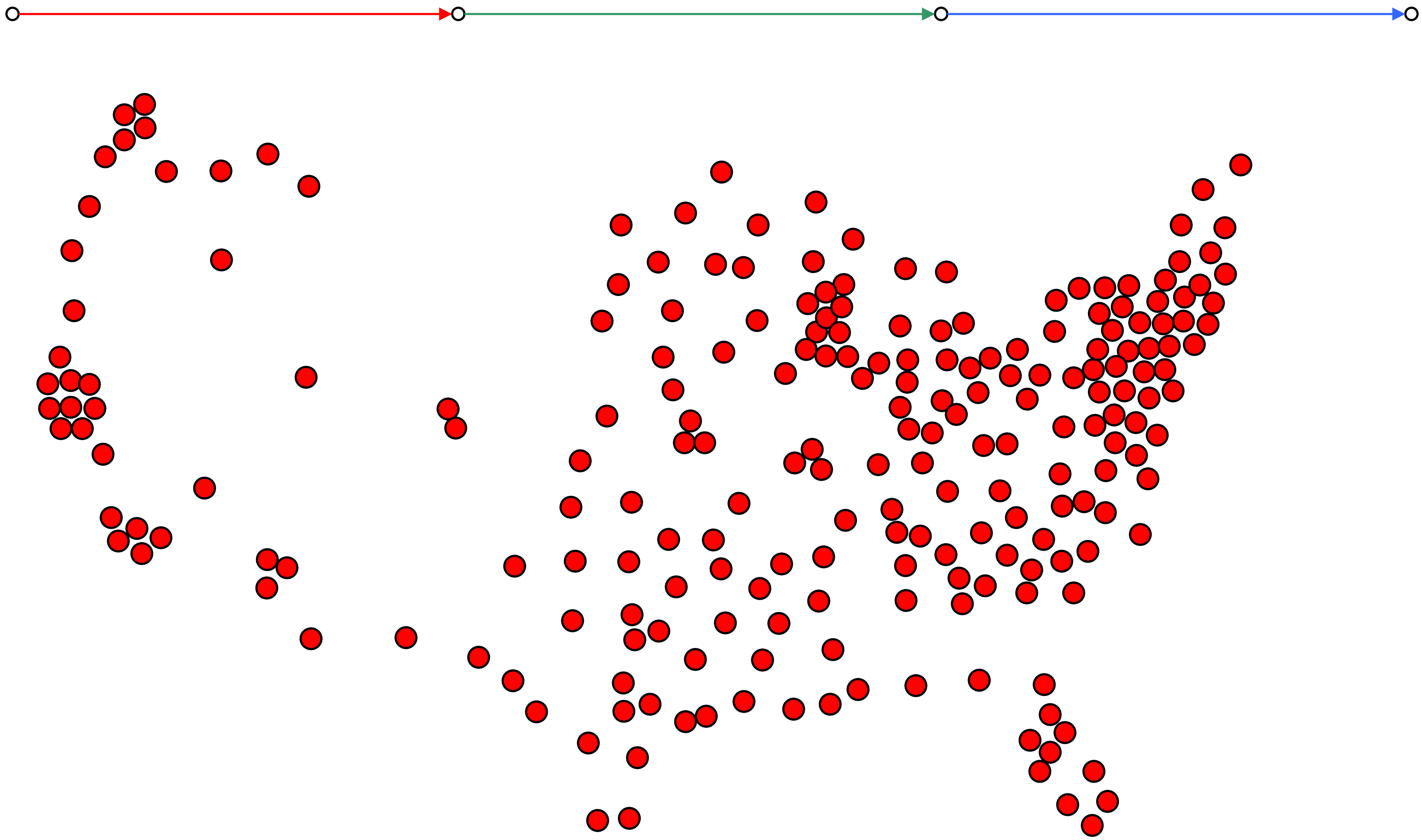
But is this really a „technicality“...?

How do real networks look like...?

Like this?


WASHINGTON
NORTH DAKOTA
MINNESOTA
MI
MICHIGAN
NEW YORK
VT
NH
MASS
CT
RI
PENNSYLVANIA
NEW JERSEY
DELAWARE
MARYLAND
OHIO
INDIANA
WEST VIRGINIA
DC
KENTUCKY
NORTH CAROLINA
TENNESSEE
GEORGIA
ALABAMA
MISSISSIPPI
MISSOURI
IOWA
NEBRASKA
KANSAS
OKLAHOMA
TEXAS
UTAH
WYOMING
NEVADA
ARIZONA
CALIFORNIA
FLORIDA

Or rather like this?



Random Node Distribution

- In theory, it is often assumed that, ...

nodes are randomly, uniformly distributed in the plane.

This assumption allows for nice formulas

Most small- and large-scale networks feature highly heterogenous node densities.

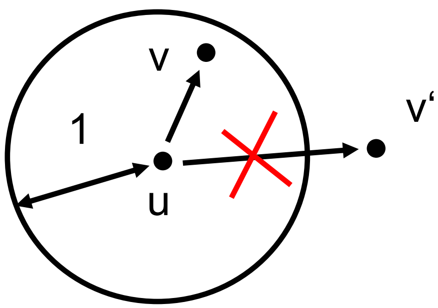
At high node density, assuming uniformity renders many practical problems trivial.

→ Not a technicality!

Unit Disk Graph Model

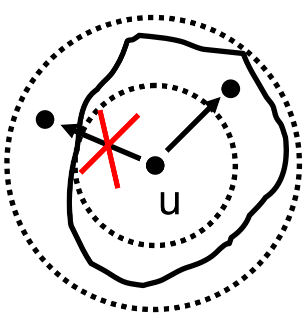
- In theory, it is often assumed that, ...

nodes form a unit disk graph!



Two nodes can communicate if they are within Euclidean distance 1.

This assumption allows for nice results



Signal propagation of real antennas not clear-cut disk!

Algorithms designed for unit disk graph model may not work well in reality. → Not a technicality!

More technicalities...

Excerpts from a typical paper

Algorithm 2 $\mathrm{LP}_{\mathrm{MDS}}$ approximation (Δ known)

$x_i := 0;$
for $\ell := k-1$ to 0 by -1 **do**
    $(*\ \tilde{\delta}(v_i) \leq (\Delta+1)^{(\ell+1)/k},\ z_i := 0\ *)$
    for $m := k-1$ to 0 by -1 **do**
        $(*\ a(v_i) \leq (\Delta+1)^{(m+1)/k}\ *)$
        send color_i to all neighbors;
        $\tilde{\delta}(v_i) := \left|\{j \in N_i \mid \mathrm{color}_j = \text{`white'}\}\right|;$
        if $\tilde{\delta}(v_i) \geq (\Delta+1)^{\ell/k}$ **then**
            $x_i := \max\left\{x_i, \frac{1}{(\Delta+1)^{m/k}}\right\}$
        fi;
        send x_i to all neighbors;
        if $\sum_{j \in N_i} x_j \geq 1$ **then** $\mathrm{color}_i :=$ 'gray' **fi**;
    od
    $(*\ z_i \leq 1/(\Delta+1)^{(\ell-1)/k}\ *)$
od

More technicalities...

6: **send** color$_i$ to all neighbors;

How do you know your neighbors ???

How can you exchange data with them ???
→ Collisions (Hidden-Terminal Problem)

Most papers assume that there is a MAC-Layer in place!

This assumption may make sense in well-established, structured networks,...

...but it is certainly invalid during and shortly after the deployment of ad hoc and sensor networks.

More technicalities...

Excerpts from a typical paper

Algorithm 2 LP_{MDS} approximation (Δ known)

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

More technicalities...

$$2{:}\ \textbf{for}\ \ell := k-1\ \text{to}\ 0\ \text{by}\ -1\ \textbf{do}$$

How do nodes know when to start the loop ???

What if nodes join in afterwards ???

→ Asynchronous wake-up!

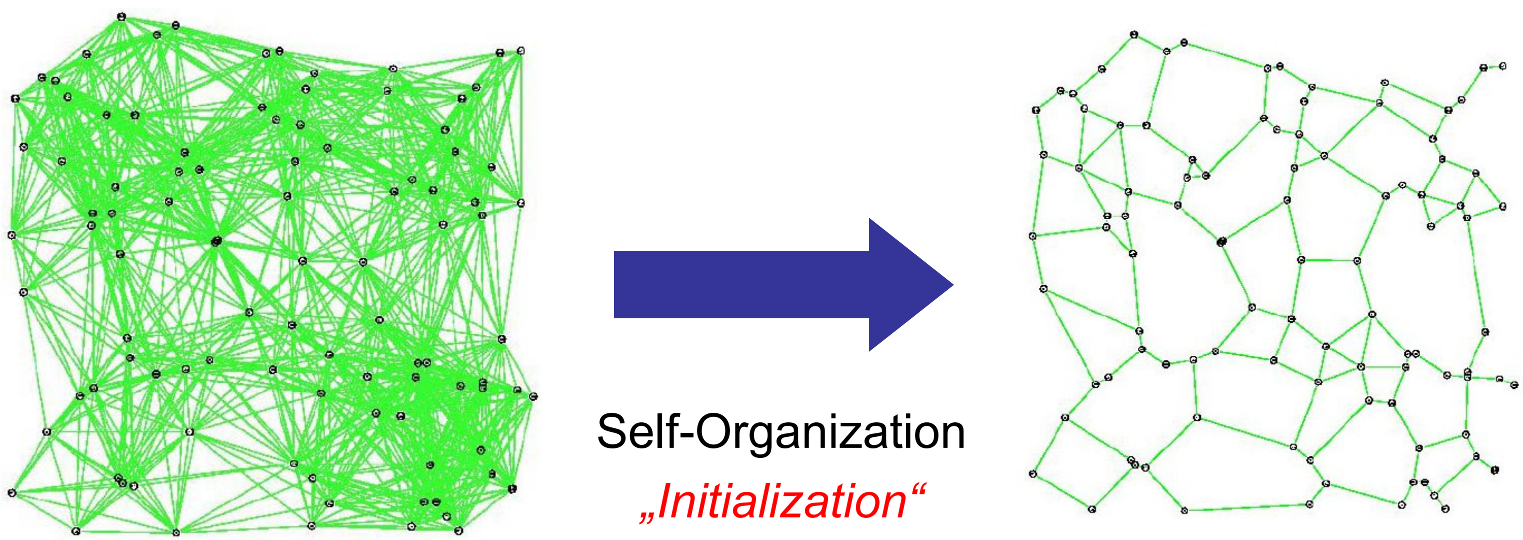
Paper assumes that there is a global clock and synchronous wake-up!

This assumption greatly facilitates the algorithm's analysis...

...but it is certainly invalid during and shortly after the deployment of ad hoc and sensor networks.

Deployment and Initialization

- Ad Hoc & Sensor Networks → no built-in infrastructure
- During and after the deployment → complete chaos
- Neighborhood is unknown
- There is no existing MAC-layer providing point-to-point connections!



Deployment and Initialization

- Initialization in current systems often very slow (Bluetooth)

- Ultimate Goal: Come up with an efficient MAC-Layer quickly.
- Theory Goal: Design a *provably* fast and reliable initialization algorithm.

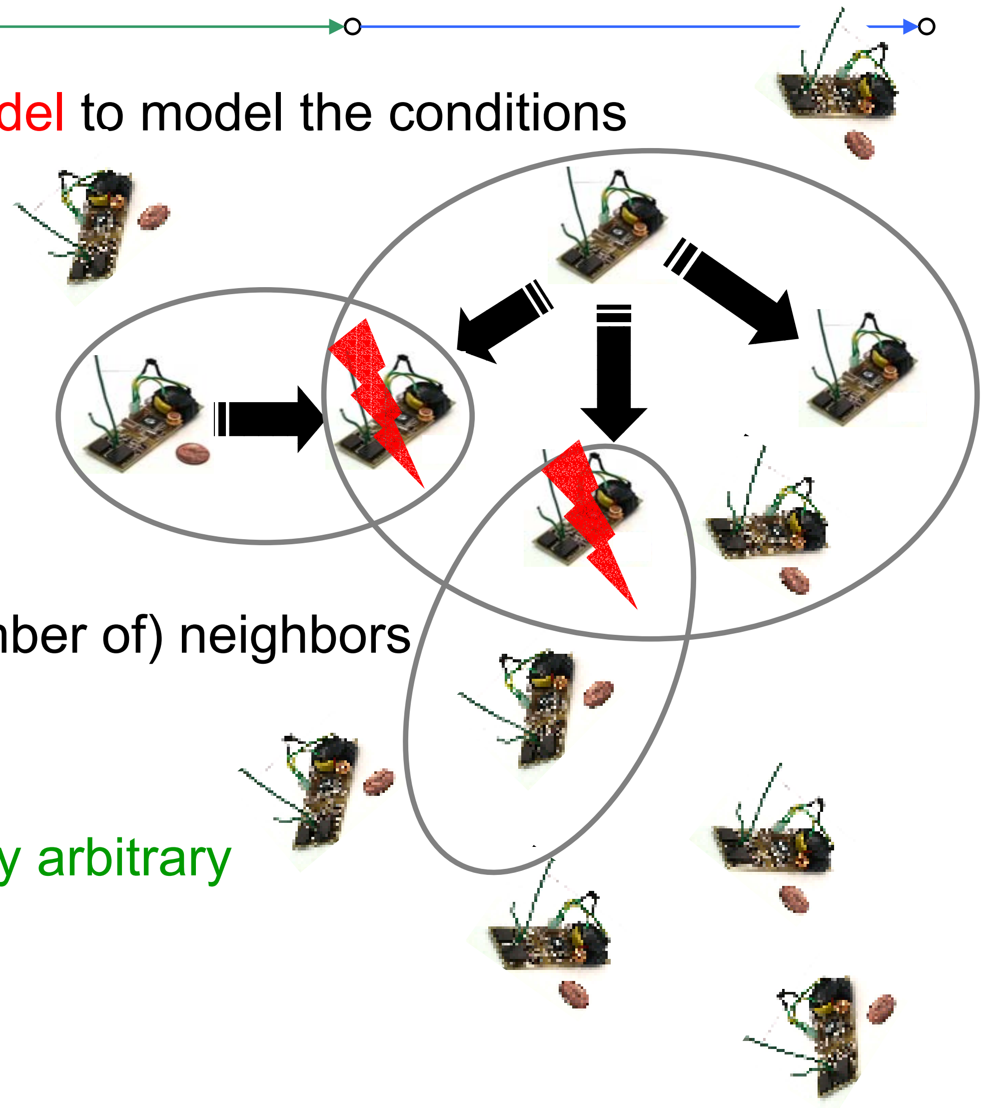
We have to consider the relevant technicalities!

We need to define a model capturing

the characteristics of the initialization phase.

Unstructured Multi-Hop Radio Networks – Model (1)

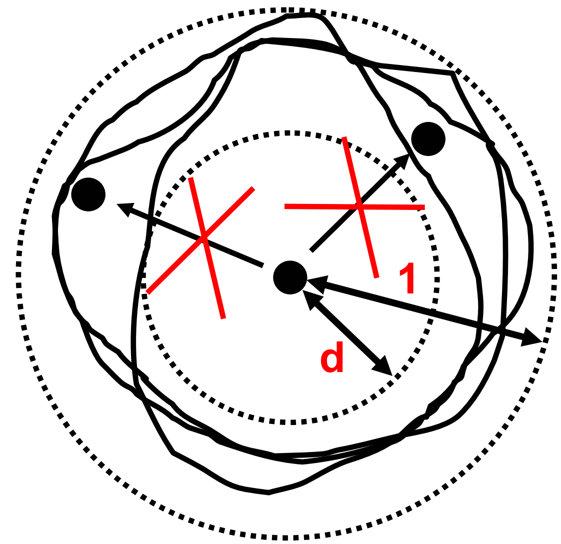
Adapt classic Radio Network Model to model the conditions immediately after deployment.



- Multi-Hop
  - Hidden-Terminal Problem
- No collision detection
  - Not even at the sender!
- No knowledge about (the number of) neighbors
- Asynchronous Wake-Up
  - No global clock!
- Node distribution is completely arbitrary
  - No uniform distribution!

Unstructured Multi-Hop Radio Networks – Model (2)

- Quasi Unit Disk Graph (QUDG) to model wireless multi-hop network
  - Two nodes can communicate if Euclidean distance is $\leq$ d
  - Two nodes cannot communicate if Euclidean distance is >1
  - In the range [d..1], it is unspecified whether a message arrives [Barrière, Fraigniaud, Narayanan, 2001]



- Upper bound N for number of nodes in network is known
  - This is necessary due to Ω(n / log n) lower bound [Jurdzinski, Stachowiak, 2002]

Q: Can we efficiently (and provably!) compute an *MAC-Layer* in this harsh model?

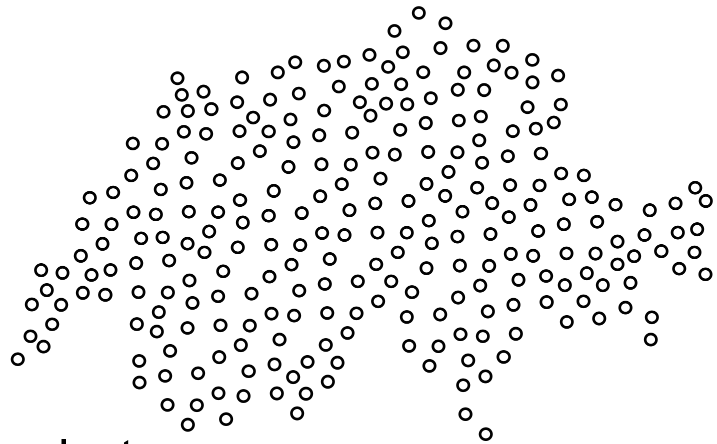
Q: Can we efficiently (and provably!) compute an *initial structure* in this model?

A: Yes, we can!

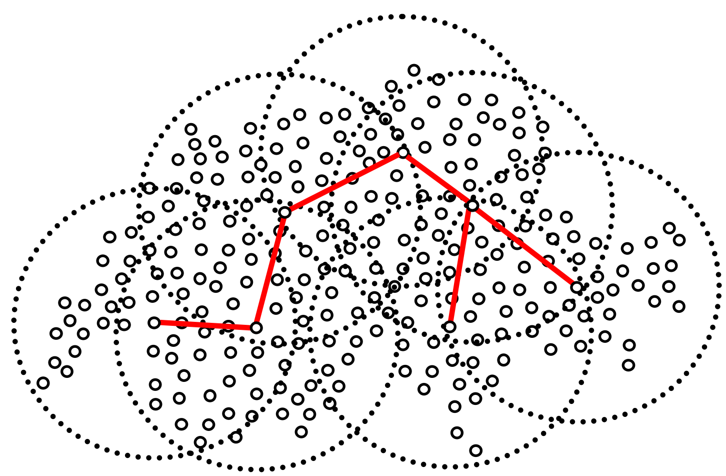
A: Hmm, ...

The Importance of Being Clustered...

- Clustering
  - Virtual Backbone for efficient routing
    → *Connected Dominating Set*
  - Improves usage of sparse resources
    → *Bandwidth, Energy, ...*
  - Spatial multiplexing in non-overlapping clusters
    → *Important step towards a MAC Layer*



Clustering



Clustering helps in bringing structure into Chaos!

Dominating Set

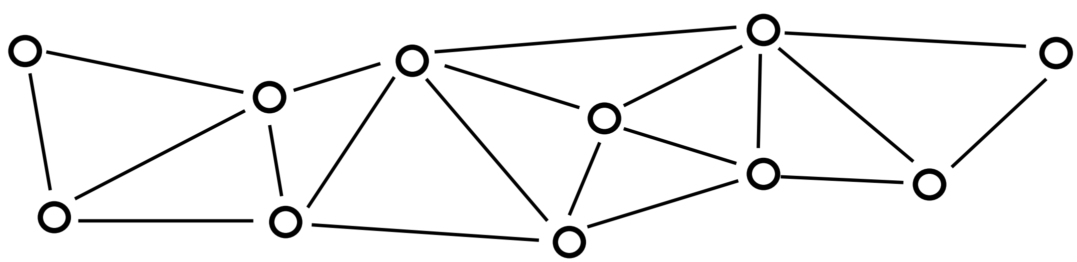
- Clustering:
  - Choose clusterhead such that:

    Each node is either a clusterhead or has a clusterhead in its communication range.

- When modeling the network as a graph G=(V,E), this leads to the well-known Dominating Set problem.

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.
- Minimum Dominating Set MDS is a DS of minimal cardinality.



Yet Another Dominating Set Algorithm...???

- There are many existing DS algorithms
  - [Kutten, Peleg, Journal of Algorithms 1998]
  - [Gao, et al., SCG 2001]
  - [Jia, Rajaraman, Suel, PODC 2001]
  - [Wan, Alzoubi, Frieder, INFOCOM 2002 & MOBIHOC 2002]
  - [Chen, Liestman, MOBIHOC 2002]
  - [Kuhn, Wattenhofer, PODC 2003]
  - .....

- Q: Why yet another clustering algorithm ?
- A: Other algorithms - with theoretical worst-case bounds - make too strong assumptions! (see previous slides...)

  → Not valid during initialization phase!

Clustering Algorithm - Results

- With three communication channels

In expectation, our algorithm computes a $O\left(\frac{1}{d^2}\right)$ approximation for MDS in time

$$O\left(\frac{\log N}{d^2}\left(\log \Delta + \frac{\log N}{\log\log N}\right)\right)$$

N : Upper bound on number of nodes in the network

Δ : Upper bound on number of nodes in a neighborhood (max. degree)

d : Quasi unit disk graph parameter

- Typically, d is a constant between 0.5 and 1.
  → Constant approximation!
- The time-complexity thus reduces to

$$O\left(\frac{\log^2 N}{\log\log N}\right) \text{ for } 1 \leq \Delta \leq N^{1/\log\log N}$$

$$O(\log N \log \Delta) \text{ for } N^{1/\log\log N} \leq \Delta \leq N$$

Overview

- Motivation
  Model

- Algorithm
  Analysis

- Conclusion
  Outlook

Clustering Algorithm – Basic Idea

- Use 3 independent communication channels Γ_1, Γ_2, and Γ_3.
  → Then, simulate these channels with a single channel.

- For the analysis: Assume time to be slotted
  → Algorithm does not rely on this assumption
  → Slotted analysis only a constant factor better than unslotted (ALOHA)

Clustering Algorithm – Basic Structure

Upon wake-up do:

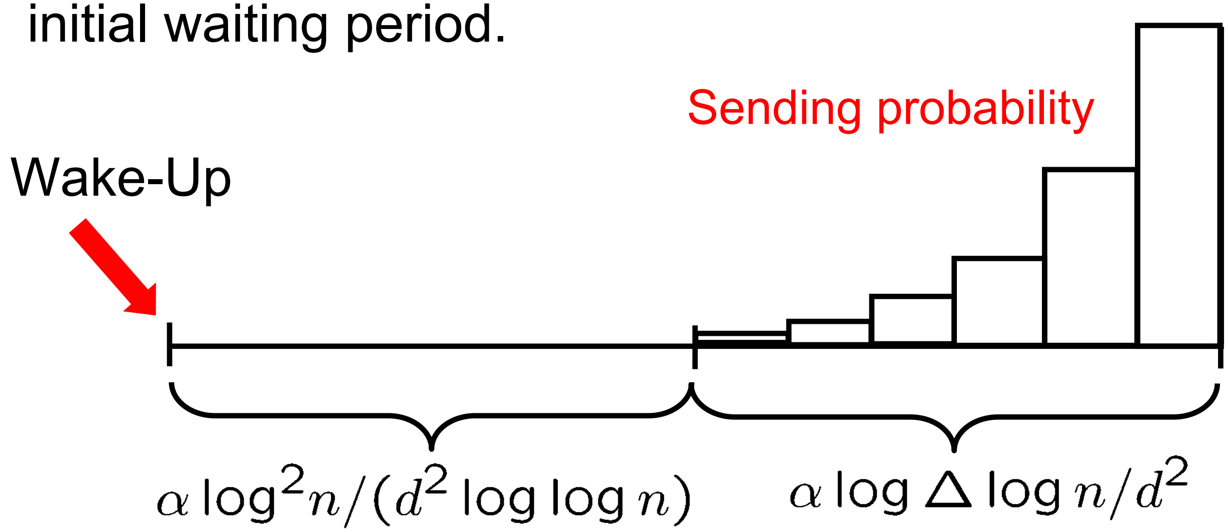
1) Listen for $\alpha \cdot \log^2 N/(d^2 \log\log N)$ time-slots on all channels
   upon receiving message → become dominated
   → stop competing to become dominator

2) For j=log Δ downto 0 do
   for $\alpha \cdot \log N/d^2$ slots, send with prob. $p_1 = \eta d^2 2^{-\log \Delta + j}$
   upon sending → become dominator
   upon receiving message → become dominated
   → stop competing to become dominator

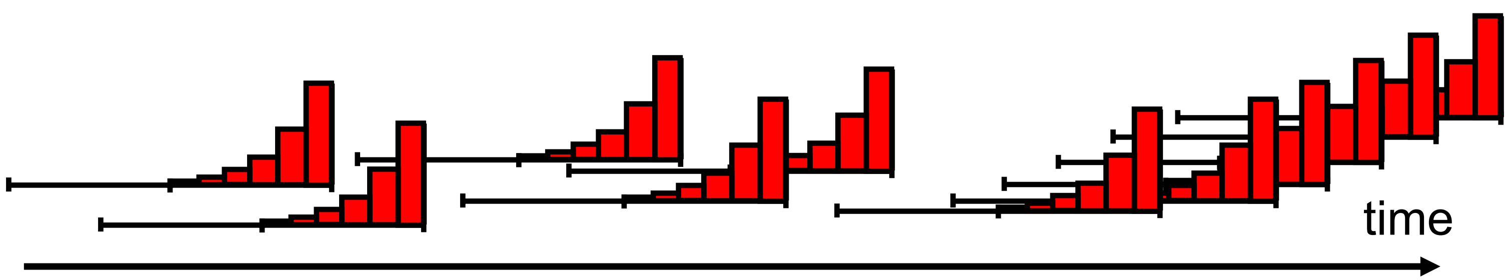
3) Additionally, dominators send on Γ_2 and Γ_3 with prob.
   $p_2 = \eta d^2 \log\log N/\log N$ and $p_3 = \eta d^2 \log\log N/\log^2 N$.

Clustering Algorithm – Basic Structure

- Each node‘s sending probability increases exponentially after an initial waiting period.

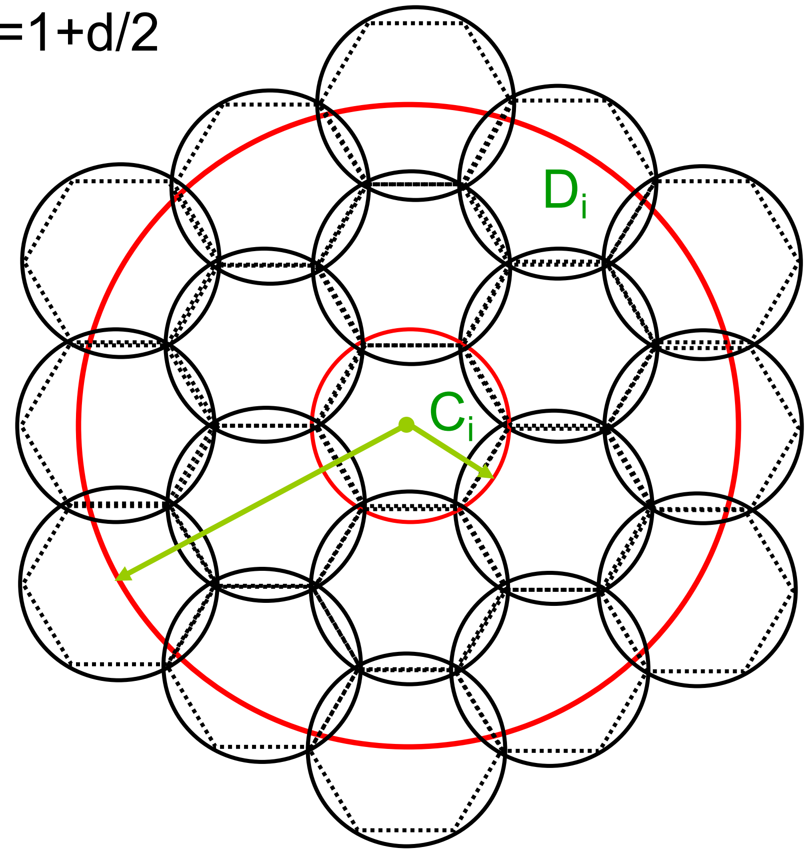


- Sequences are arbitrarily shifted in time (asynchronous wake-up)



Analysis - Outline

- Cover the plane with (imaginary) circles C_i of radius r=d/2
- Let D_i be the circle with radius R=1+d/2



- A node in C_i can hear all nodes in C_i
- Nodes outside of D_i cannot interfere with nodes in C_i

- We show: Algorithm has O(1) dominators in each C_i
- Optimum needs at least 1 dominator in D_i

Constant Approximation for constant d

Analysis - Outline

1. Bound the *sum of sending probabilities* in a circle C_i

   Remember: Due to asynchronous wake-up, every node may have a different sending probability

2. Bound the number of collisions in C_i before C_i becomes cleared

3. Bound the number of sending nodes per collision

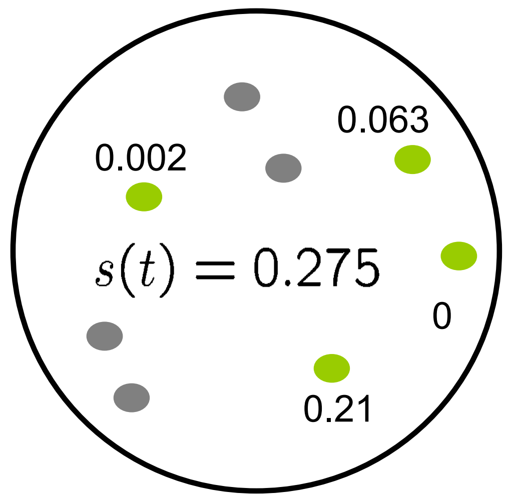
4. Newly awakened, already covered nodes will not become dominator

Analysis

Lemma 1: *Bound sum of sending probabilities in C_i*

- *Def:* Let $s(t)$ be the sum of sending probabilities of nodes in a circle C_i at time t, i.e.,

$$s(t) := \sum_{k \in C_i} p_k(t)$$

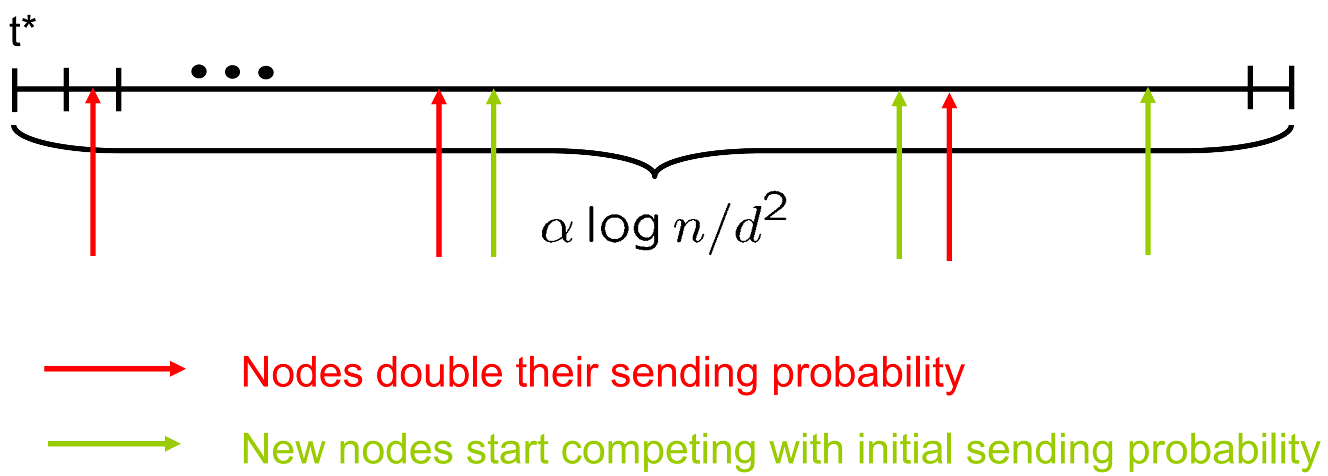


For all circles C_i and all times t, it holds that $s(t) \leq 3\eta d^2$ w.h.p.

Analysis

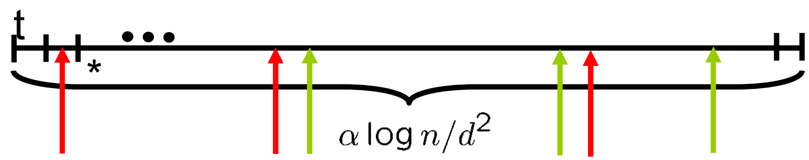
- Proof of Lemma 1:

- Induction over all time-slots when (for the first time) $s(t) > \eta d^2$ in a circle C_i. (Induction over multi-hop network!)
- Let t* be such a time-slot
- Consider interval $[t^*, \ldots, t^* + \alpha \log n / d^2 - 1]$



Analysis

- Proof of Lemma 1 (cont)
- Existing nodes can at most double
- New nodes send with very small probability



$$s(t + \alpha \log n/d^2 - 1) \leq 3\eta d^2$$

→ Next, we show in the paper that i$[t^*, \ldots, t^* + \alpha \log n/d^2 - 1]$ there will be at least one time-slot in which no node in $D_i \setminus C_i$, and exactly one node in C_i sends.

→ After this time-slot, C_i is *cleared,* i.e., all (currently awake) nodes are decided.

→ Sum of sending probabilities does not exceed $3\eta d^2$

Analysis

Lemma 1:
Bound on sum of sending probabilities

⇒

Lemma 2:
Bound number of collisions before C_i is cleared

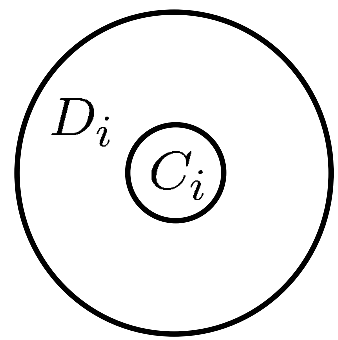
- Define events:

  X : More than one node in C_i is sending

  Y : At least one node in C_i is sending

  Z : Some node in $D_i \setminus C_i$ is sending

  A : Exactly one node in D_i is sending



- Compute probability:

$$P[A|Y] = \ldots = \left(1 - \frac{P[X]}{P[Y]}\right)(1 - P[Z])$$

Analysis

Lemma 2

Let C be the number of collisions in a circle C_i. The expected number of collisions in C_i before its clearance is $E[C] < 5$ and $C < 6 \log n$ with high probability.

Lemma 3 (Computed using Lemma 1)

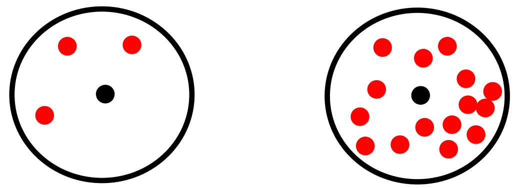
Let D be the number of nodes in a circle C_i sending in a time-slot. Let Φ be the event of a collision in C_i. Given a collision, the expected number of sending nodes is $E[D|\Phi] \in O(1)$ and $D|\Phi < 3 \log n / \log \log n$ with high probability.

Analysis

Lemma 4

- It remains to be shown that only O(1) nodes waking up after the clearance become dominator.
- D : Number of dominators in the range of a newly awakened node.
- Distinguish two cases:
  - Case 1: $1 \leq D \leq c \log n$
    → Consider channel Γ_2
  - Case 2: $c \log n \leq D \leq c' \log^2 n$
    → Consider channel Γ_3
- Compute probability, that one dominator sends alone on Γ_2 or Γ_3.

Case 1 Case 2



$$P_1 = D \cdot q \cdot (1-q)^{D-1} = \ldots \in \Omega(\log^{-1} n)$$

→ Since waiting-period is $O(\log^2 n / \log\log n)$ long, at least one message will eventually arrive at the node.

Analysis - Results

- For each circle C_i holds:
  - Number of dominators before a clearance in O(1) in expectation
  - Number of dominators after a clearance in O(1) w.h.p
  - → Number of dominators in C_i in O(1) in expectation

- Optimum has to place at least one dominator in D_i.

In expectation, the algorithm compute a $O(1/d^2)$ approximation.

- Reasonable values of d are constant → Constant approximation!

Three Channels → Single Channel

- Three independent communication channels not always feasible
- Simulation with a single channel is possible within O(polylog(n)).
- Idea:
  - Each node simulates each of its multi-channel time-slots with O(polylog(n)) single-channel time-slots.
  - It can be shown that result remains the same.

Algorithm compute a O(1/d^2) approximation for MDS in polylogarithmic time even with a single communication channel.

Overview

- Motivation
  Model

- Algorithm
  Analysis

- Conclusion
  Outlook

Simulation

- The hidden constants in the big-O notation are quite big.
- Simulation shows that this is an artefact of „worst-case“ analysis.
- In reality, it is sufficient to set $\alpha := 10$.

  → Running time is at most $t < 10 \cdot \log^2 n$

Example: Scatterweb, Embedded Sensor Nodes

TU Berlin

Transmission rate :	115 kb/s
Switch time trans → recv :	20 μs
Switch time recv → trans :	12 μs
Paketsize of algorithm :	~20 Byte

→ Lenght of one time-slot is < 3 ms

Initializing 1000 nodes takes time < 3 seconds!

Comparison: For 2 nodes, Bluetooth takes about 20 seconds!

Conclusion and Outlook

- Initialization of ad hoc and sensor network of great importance!
- Relevant technicalities must be considered!

In this work:

- A model capturing the characteristics of the initialization phase
- A fast algorithm for computing a good dominating set from scratch
- An application of the algorithm

In our MASS 2004 paper:

- A fast algorithm for computing more sophisticated structures (MIS)

GOAL

A fast algorithm for establishing a MAC Layer from scratch!