Efficient Computation of Maximal Independent Sets in Unstructured Multi-Hop Radio Networks

Thomas Moscibroda
Roger Wattenhofer

MASS 2004

Distributed Computing Group

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithms for Ad Hoc and Sensor Networks...

Excerpts from a typical paper

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

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

Assumptions

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 ...

1) ... there is a MAC-Layer in place!

2) ... nodes know their neighbors!

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.

Assumptions...

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** $\text{color}_i := \text{`gray'}$ **fi**;

  od

  $(*\ z_i \leq 1/(\Delta+1)^{(\ell-1)/k}\ *)$

od

More technicalities...

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

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 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.

Deployment

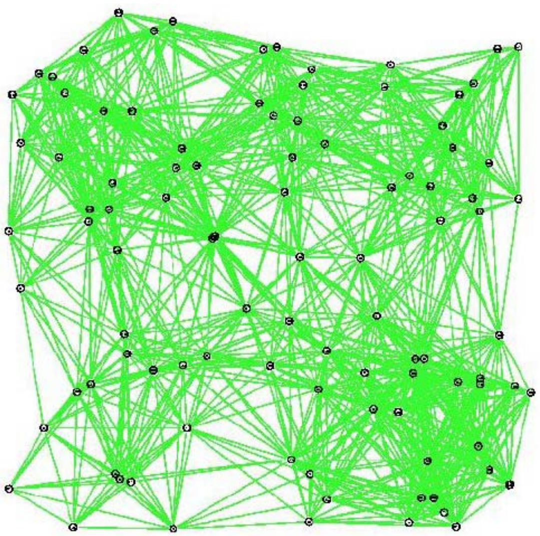
- Absence of a-priori knowledge
  - → Neighbors are unknown
  - → Even number of neighbors is unknown
- No built-in infrastructure
  - → No existing MAC-Layer
  - → No reliable point-to-point connections
  - → Collisions, Interference...
- Asynchronous wake-up

→ Nodes have to structure their network !

Initialization-Gap

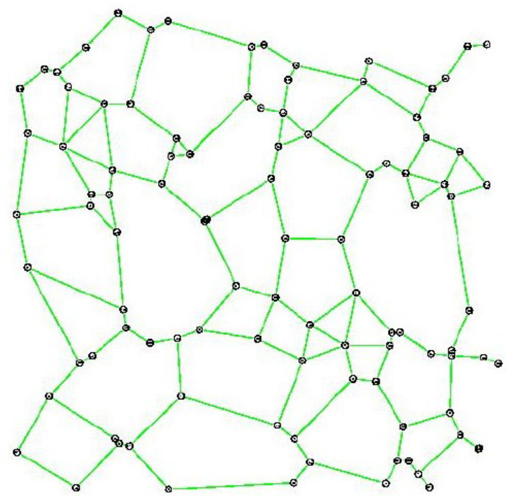
Reality starts like this:

Algorithms assume this:



Self-Organization

„Initialization“



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 must make realistic assumptions!

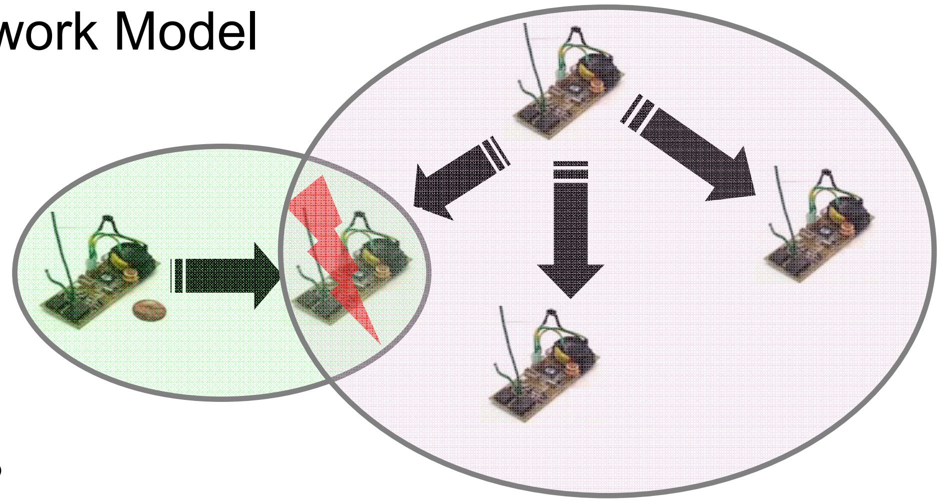
We need to define a model capturing the characteristics of the initialization phase.

Overview

- Introduction → Initialization-Gap
- Model
- Clustering
- Algorithm & Analysis
- Conclusions & Open Problems

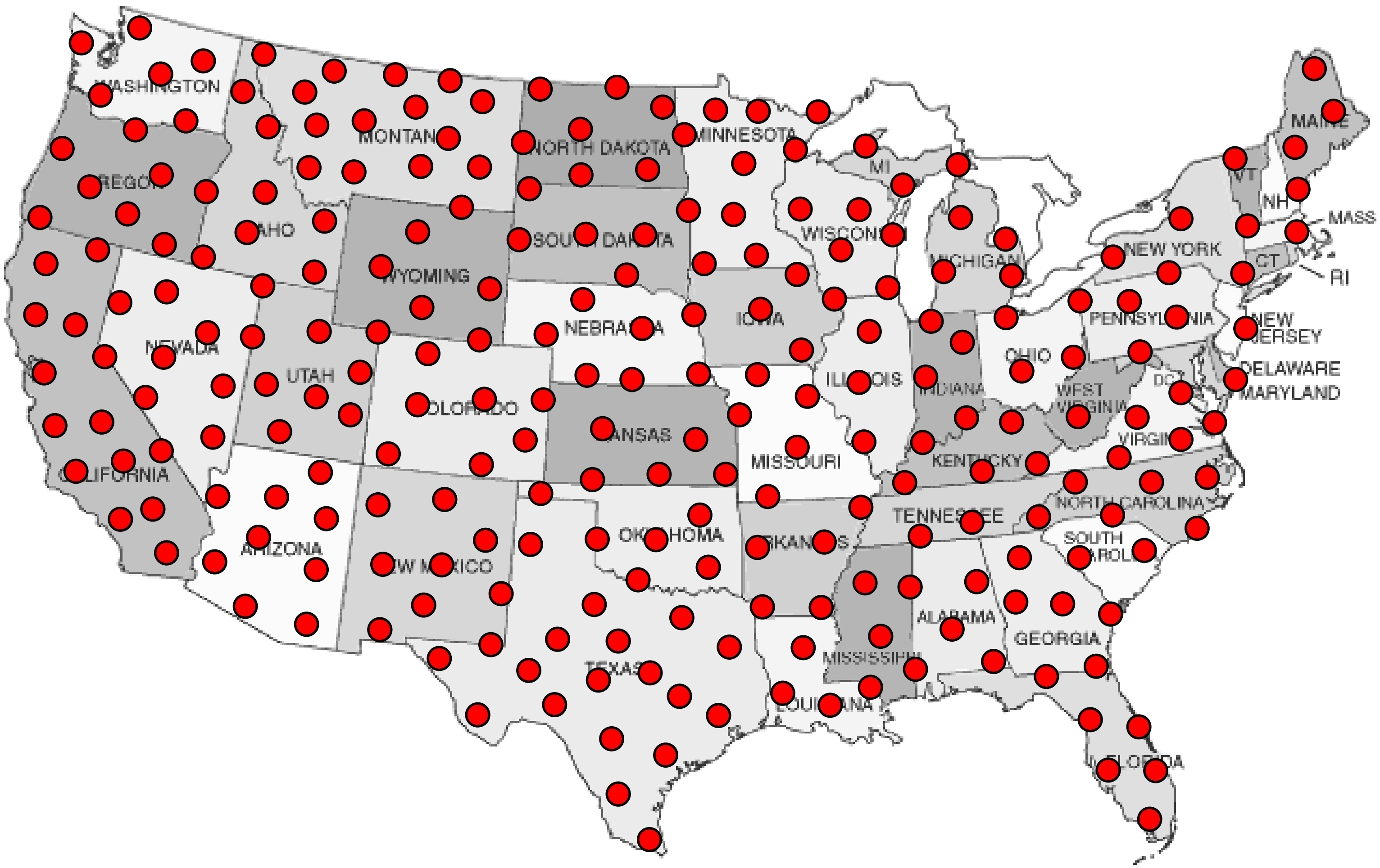
Unstructured Multi-Hop Radio Networks – Model (1)

- Introduced by [Kuhn, Moscibroda, Wattenhofer, MOBICOM 2004]
- Adaptation of classic Radio Network Model

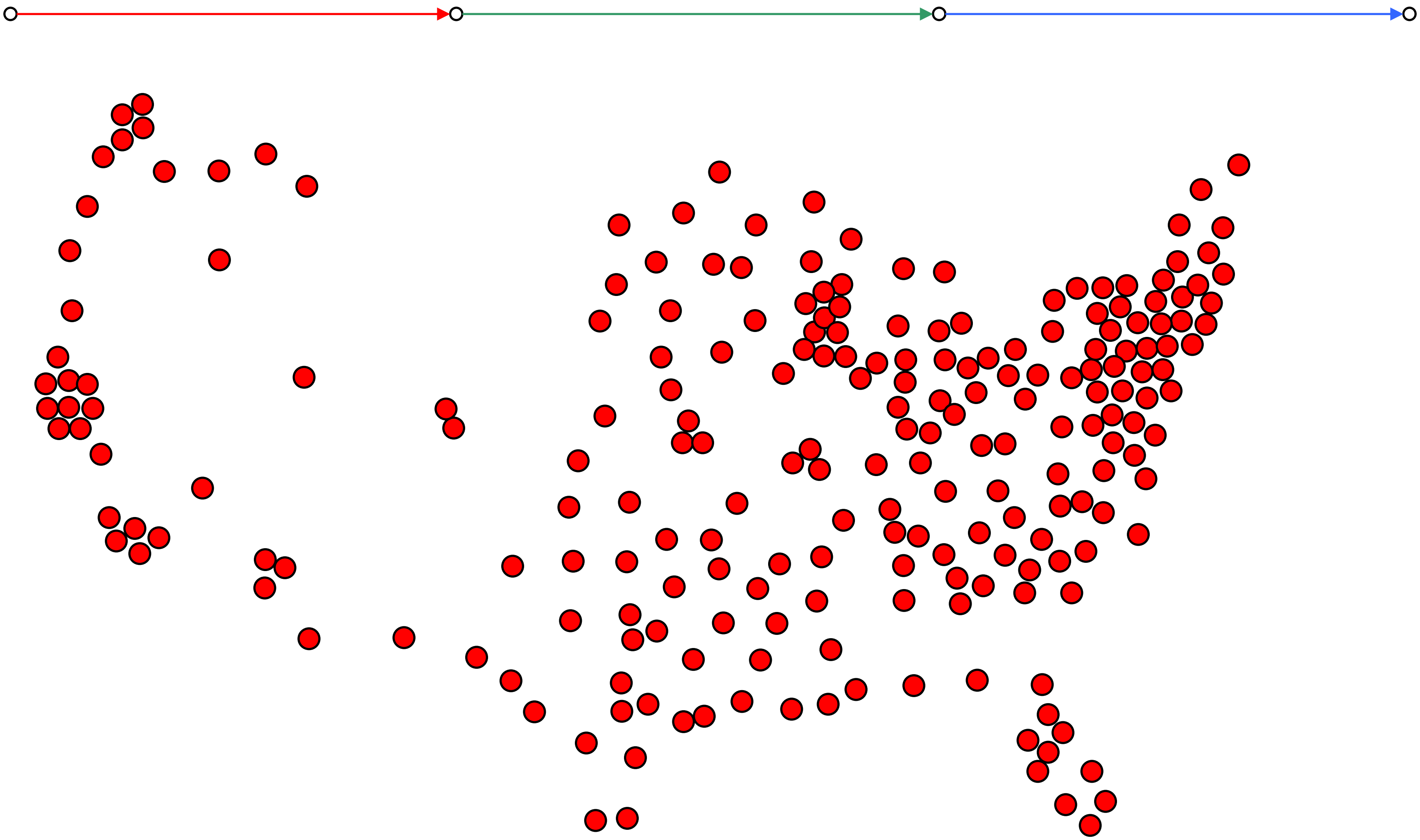


- Multi-Hop
  - Hidden-Terminal Problem
- Node distribution...?
  - How do real networks look like?

Like this?

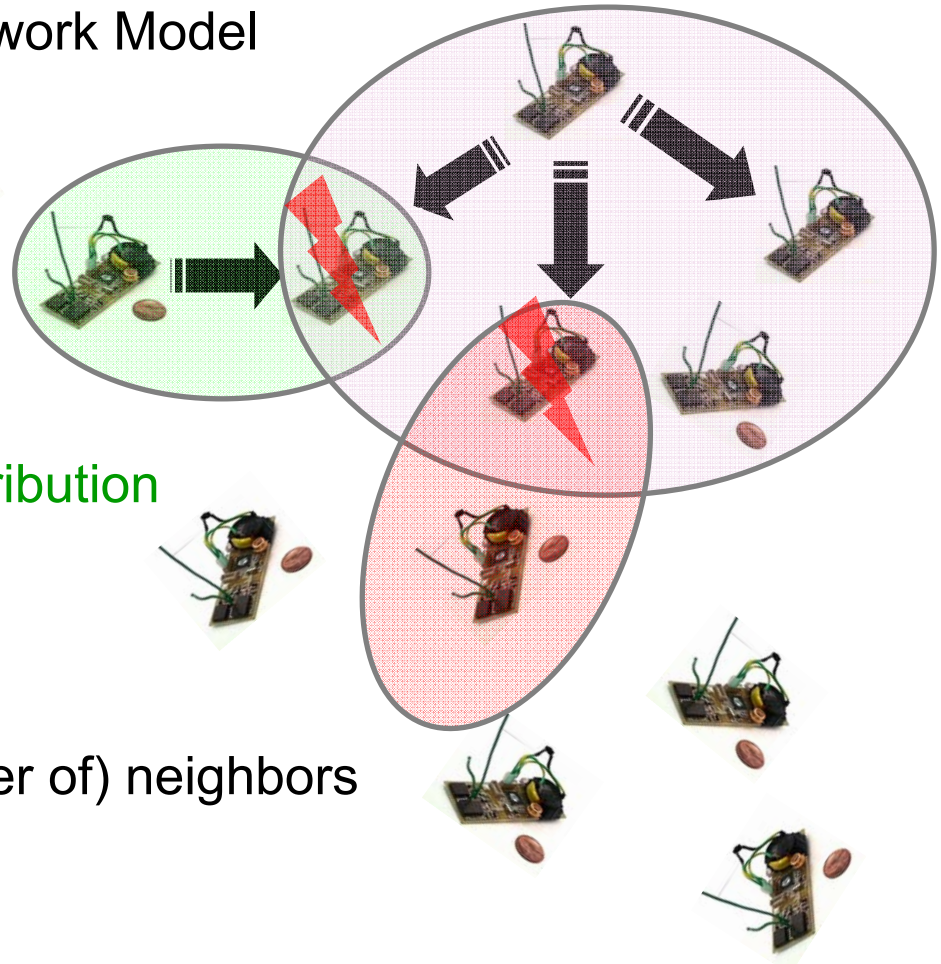

WYOMING
UTAH
MISSOURI
NEW YORK
MICHIGAN
OHIO
NORTH CAROLINA
GEORGIA
MAINE
VT
NH
MASS
CT
RI
NEW JERSEY
DELAWARE
MARYLAND
MI
DC

Or rather like this?



Unstructured Multi-Hop Radio Networks – Model (1)

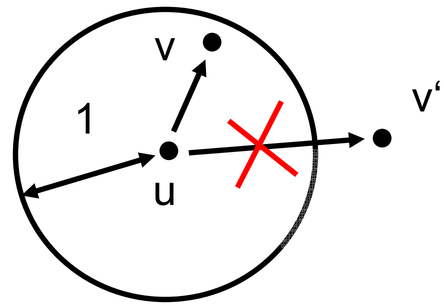
- Introduced by [Kuhn, Moscibroda, Wattenhofer, MOBICOM 2004]
- Adaptation of classic Radio Network Model



- Multi-Hop
  - Hidden-Terminal Problem
- Arbitrary (worst-case) node distribution
  - No assumption of uniformity!
- No collision detection
  - Not even at the sender!
- No knowledge about (the number of) neighbors
- Asynchronous Wake-Up
  - No global clock!

Unstructured Multi-Hop Radio Networks – Model (2)

- Unit Disk Graph (UDG) to model wireless multi-hop network
  - Two nodes can communicate iff Euclidean distance is at most 1



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

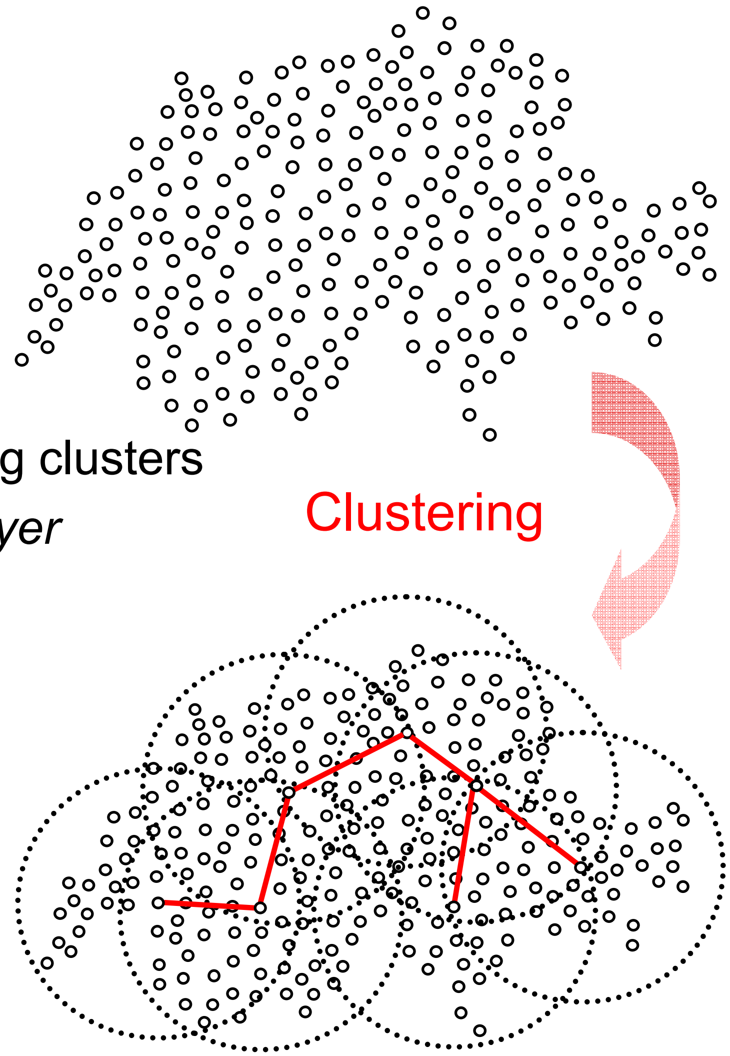
A: Hmm, we…
A: Yes, we can!

Overview

- Introduction → Initialization-Gap
- Model
- Clustering
- Algorithm & Analysis
- Conclusions & Open Problems

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 brings structure into Chaos!

Clustering: Dominating Set

- Choose clusterhead such that:

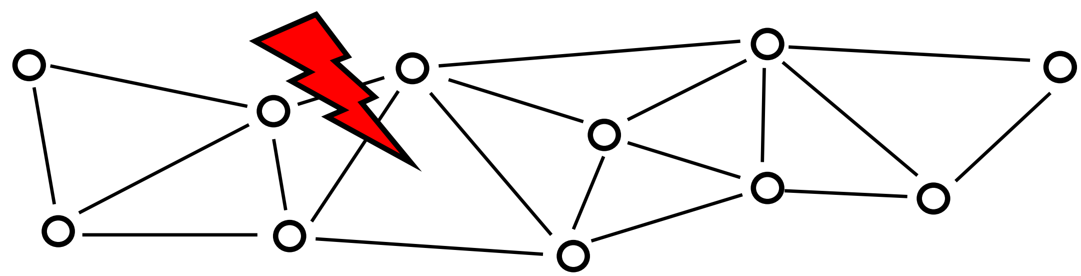
  Each node is either a clusterhead or has a clusterhead in its communication range.

- 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.

Interference between clusterheads



Clustering – Dominating Set

Dominating Set: provides a good initial structure !

Can be solved efficiently !

→ See our MOBICOM paper...

There may be interference between clusterheads!

→ We don't want neighboring clusterheads!

Clustering – Maximal Independent Set

- Choose clusterhead such that:

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

  2) *No two clusterheads are within each other's communication range.*

  → This avoids interference at clusterheads!

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

Maximal Independent Set:

– A Maximal Independent Set MIS is a dominating set where no two dominators are neighbors.

Yet Another Clustering Algorithm...???

Recently, we gave a O(1) approximation algorithm for dominating set in time $O(\log^2 n)$ in the same model.

[Kuhn, Moscibroda, Wattenhofer, MOBICOM 2004]

- There are many other existing clustering algorithms for ad hoc and sensor networks
  - [Kutten, Peleg, Journal of Algorithms 1998], [Jia, Rajaraman, Suel, PODC 2001]
  - [Gao, et al., SCG 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!

Overview

- Introduction → Initialization-Gap
- Model
- Clustering
- Algorithm & Analysis
- Conclusions & Open Problems

MIS Algorithm - Results

- With three communication channels

Algorithm computes a correct MIS in time $O(\log^3 n / \log\log n)$, w.h.p.

- Idea: Simulate three channels with a single channel.

Algorithm compute a correct MIS in polylogarithmic time even with a single communication channel, w.h.p.

MIS Algorithm – Basic Idea

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

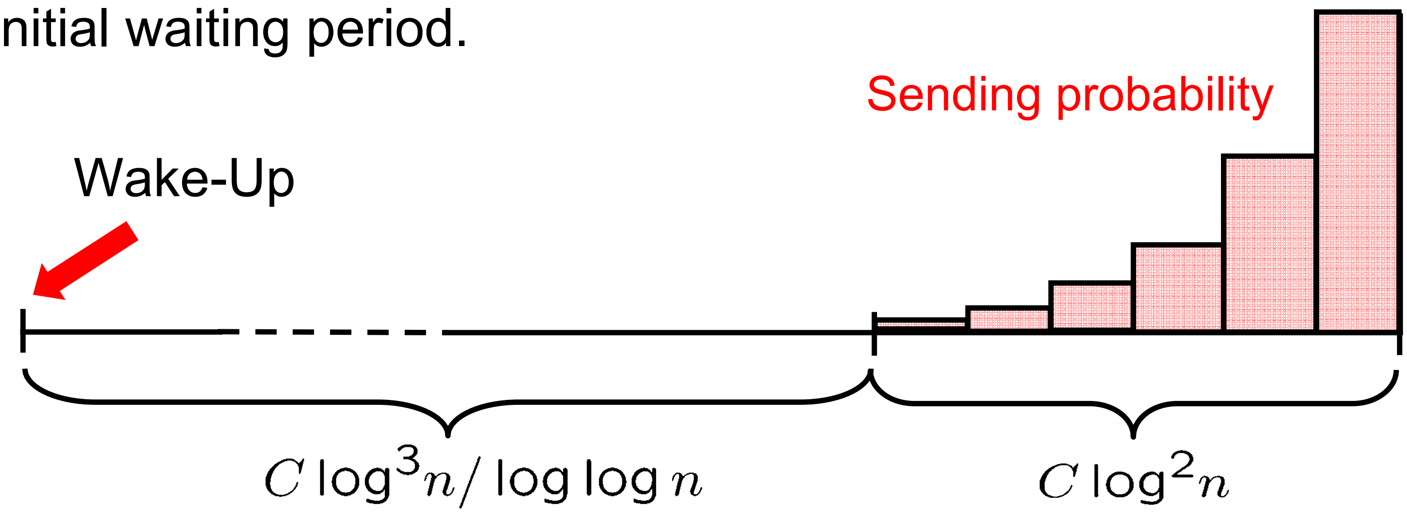
- Nodes compete for becoming candidates on Γ_1.
  → Dominating Set Algorithm → Candidate Election
  → Idea: Reduce number of potential MIS nodes
  from n to $O(\log^2 n / \log\log n)$ nodes w.h.p.

- Candidates compete for joining the MIS on Γ_2.
  → MIS Algorithm

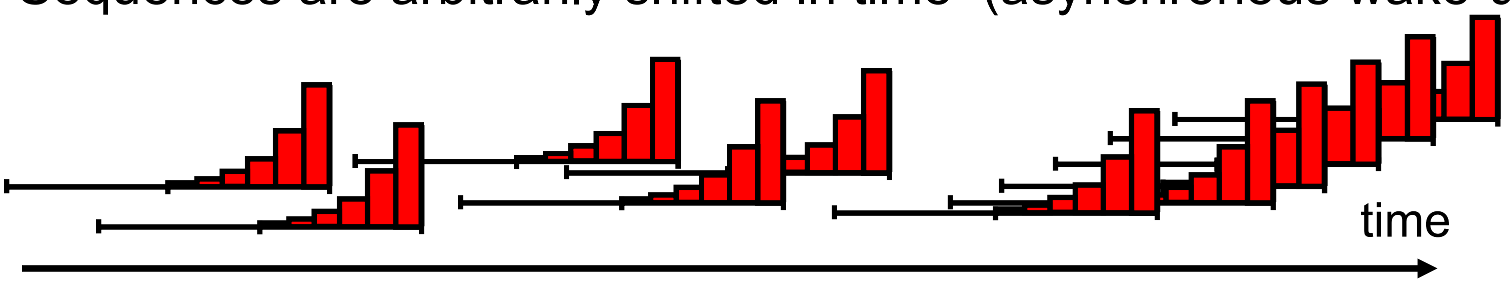
- MIS nodes send on Γ_3.
  → Inform nodes which have woken up late

Candidate Election – Basic Structure

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



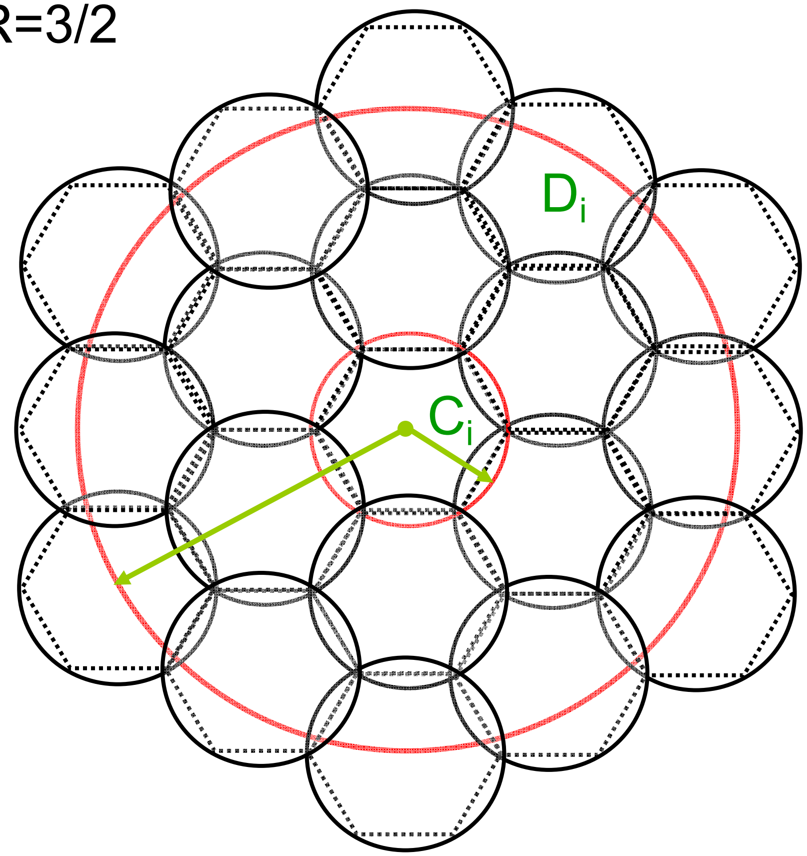
- When sending a message → become candidate
- When receiving a message (without collision!) → restart algorithm!
- Sequences are arbitrarily shifted in time (asynchronous wake-up)



Candidate Election - Analysis

- Cover the plane with (imaginary) circles C_i of radius r=1/2
- Let D_i be the circle with radius R=3/2

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



Result:

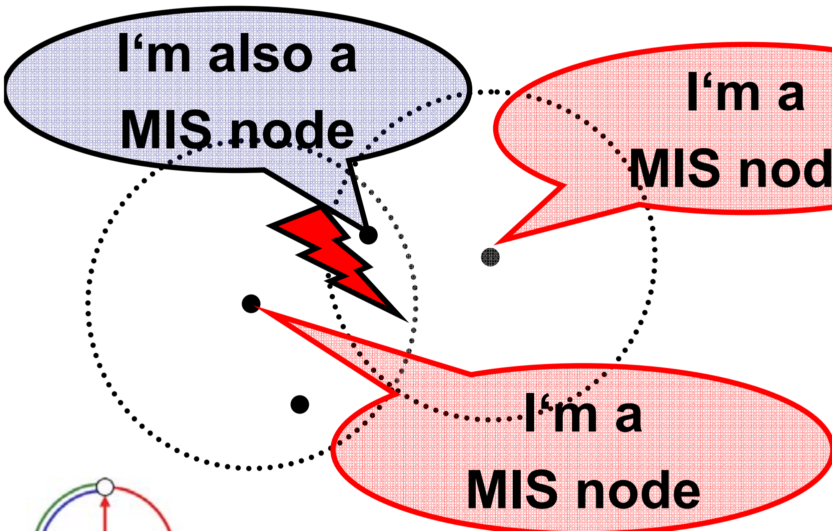
Algorithm produces at most $O(\log^2 n / \log\log n)$ candidates in each C_i w.h.p.

MIS Node Election

- Given the $O(\log^2 n / \log\log n)$ candidates per C_i.
- We want to select MIS nodes from candidates.

1.Simple Idea:

- Each candidate sends with certain probability
- Sender becomes MIS node
- Node that receives message stop competing



[Wan, Alzoubi, Frieder]

Doesn‘t work!

→ Collisions!

MIS Node Election

2. Better Idea:

- Each active candidate has counter c(v).
- Active candidates send with prob. $\Theta(\log\log n / \log^2 n)$.
- When sending first time

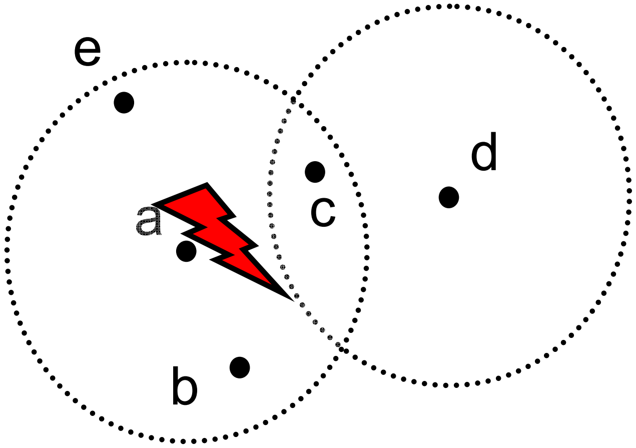
  → start incrementing counter in each time-slot
- Node that receives message, restart algorithm
- When counter reaches $\Theta(\log^3 n / \log\log n)$

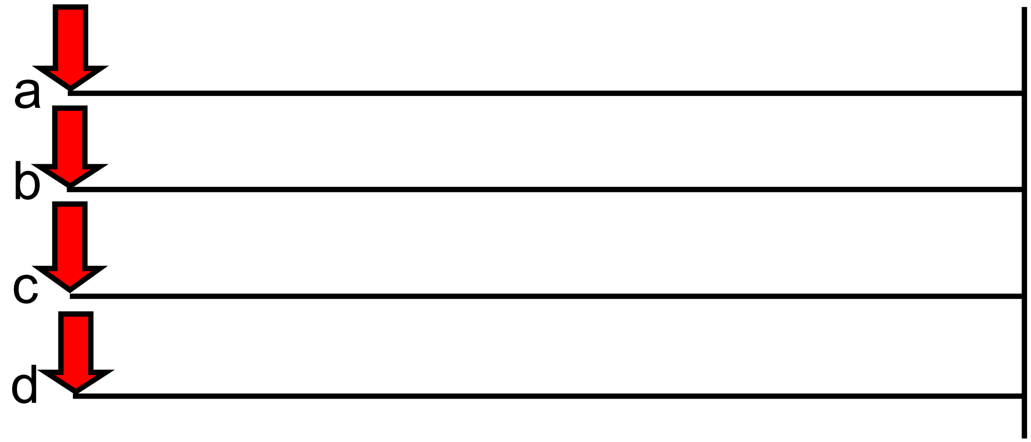
  → join MIS

  → start sending on channel Γ_3 with prob. 1/6

Intuition: Each node hears every other node (without collision) at least once!

MIS Node Election





$$\Theta\left(\frac{\log^3 n}{\log\log n}\right)$$

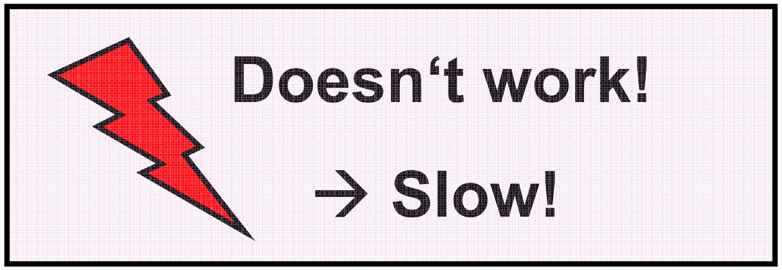
Positive:

→ No neighboring candidates join MIS

Negative:

→ New candidates can „kill“ so far successful candidates

→ Progres is too slow !!!



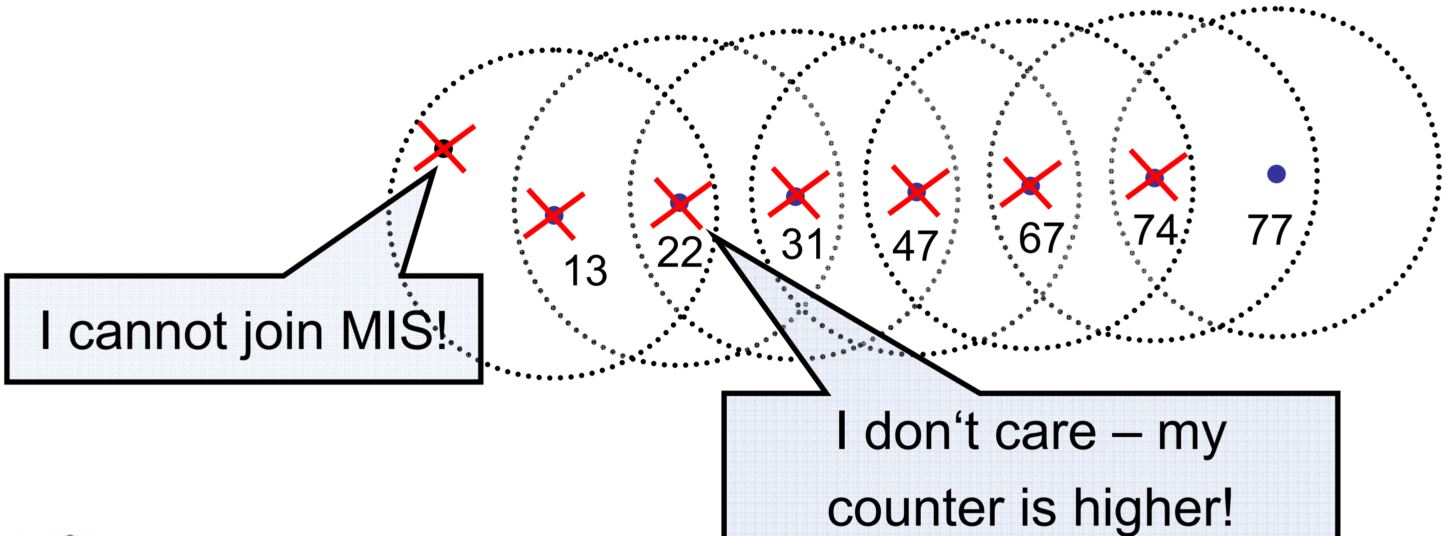
MIS Node Election

3. Improvement:

- Only nodes with **higher counter value** can „kill“ other active candidates!

Problem : Long cascading chains of restarts...

→ progress still slow in some regions!



MIS Node Election

- Goal: Every node restarts at most a constant number of times.
  → Must avoid long cascading chains of restarts!
- Intuition: Avoid that two candidates join MIS at the same time!
  → Then MIS nodes send on Γ_3 to inform candidates!

4. New Idea:

- Candidates that receive a message from a candidate with equal counter resets counter to 0.
  → Only candidates with **equal counter** can stop each other.

MIS Node Election

Let t* be the first time-slot when a candidate appears in C_i.
It can be shown that...

→ Within time t*+O(log n), a candidate sends without collision!

→ After that, this candidate cannot be stopped anymore!

→ **Only constant number of restarts!**

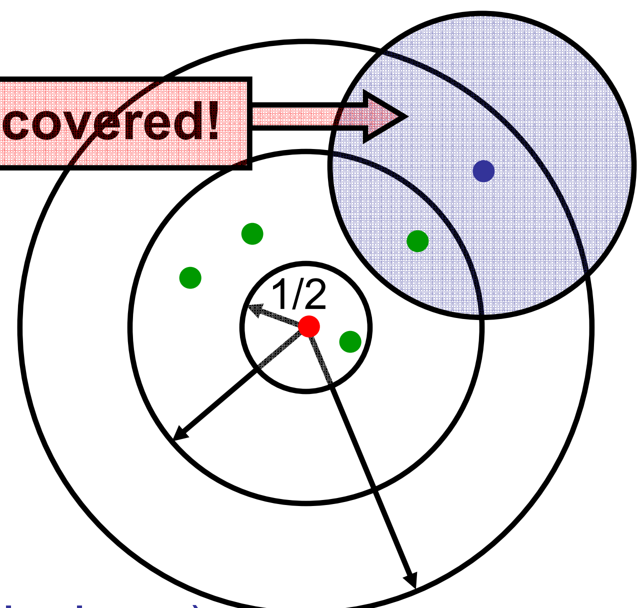
This region is covered!

t_0: u wakes up

$t_1 := t_0 + O(\log^2 n)$: at least one candidate emerges

$t_2 := t_1 + O(\log n)$: one candidate can send without collision...

... and joins MIS at $t_2 + O(\log^3 n / \log\log n)$



MIS Node Election

Candidates can have counter very close to each other!

When a node joins MIS

→ not enough time to inform neighboring candidates!

→ Feasibility not guaranteed anymore!

4. Final Idea:

- Candidates that receive a message from a candidate with similar counter resets counter to - $\Theta(\log n)$.

→ Only candidates with **similar counter** (within $\Theta(\log n)$) can stop each other

Enough time to inform neighbors!

This works!

Results

- Some nasty details have been left out...
- Running time of algorithm is $O\left(\frac{\log^3 n}{\log\log n}\right)$
- Solution 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 correct MIS in polylogarithmic time even with a single communication channel, w.h.p.

Conclusion and Outlook

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

MOBICOM 2004:

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

In this paper:

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




GOAL

A fast algorithm for establishing a MAC Layer from scratch!