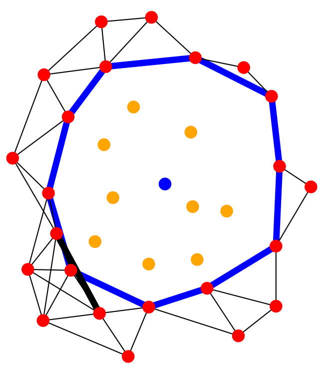
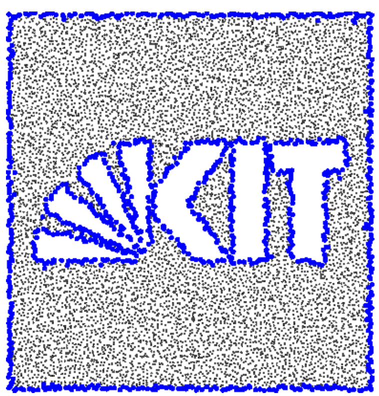
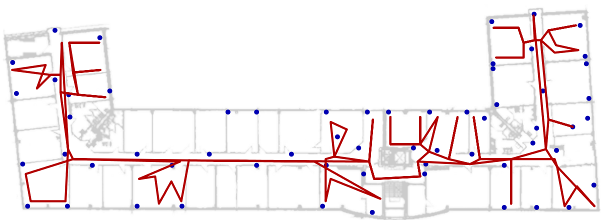
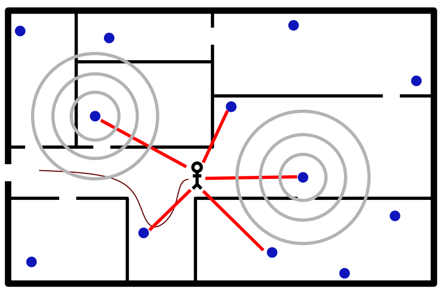
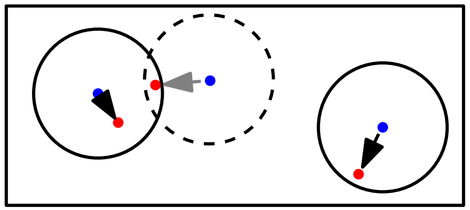
Scheduling, Boundary Detection, and Localization in Wireless Networks











Markus Völker

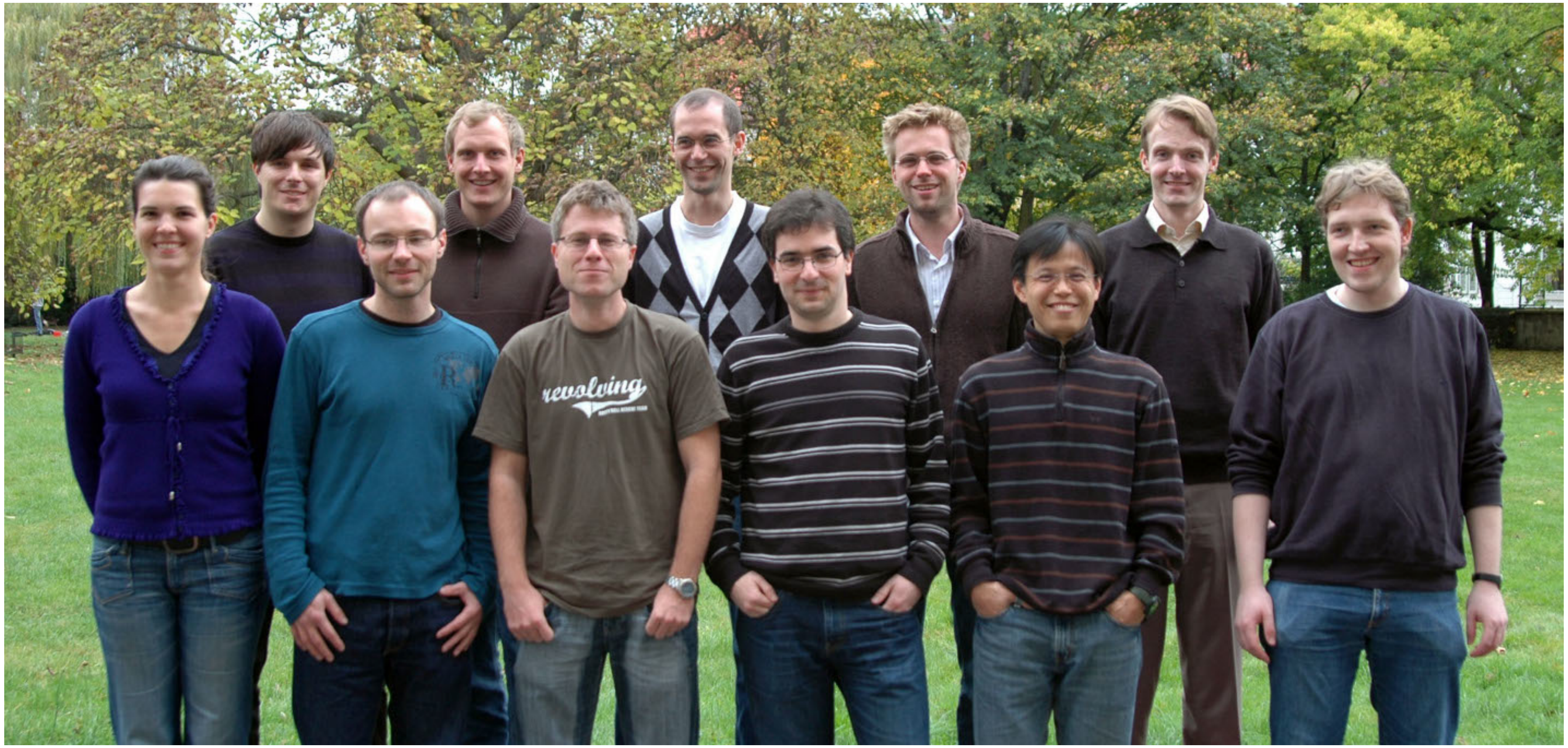
April 20, 2011

About me...

- PhD student at Institute for Theoretical Informatics at KIT
  - Algorithmics Group (Prof. Dorothea Wagner)

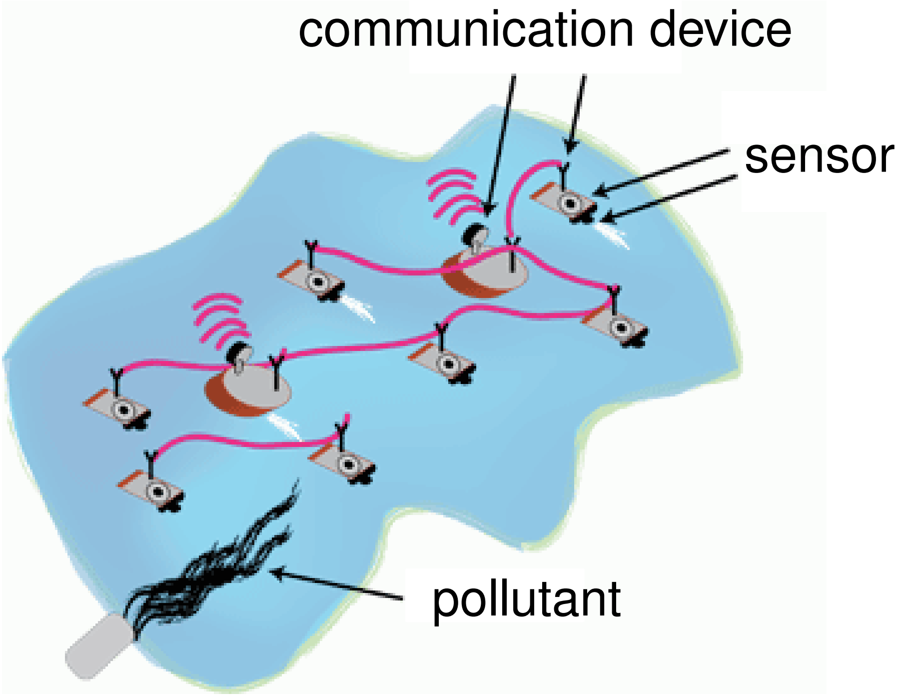
About me...

- Member of DFG Research Training Group 1194
  - Self-organizing Sensor Actuator Networks



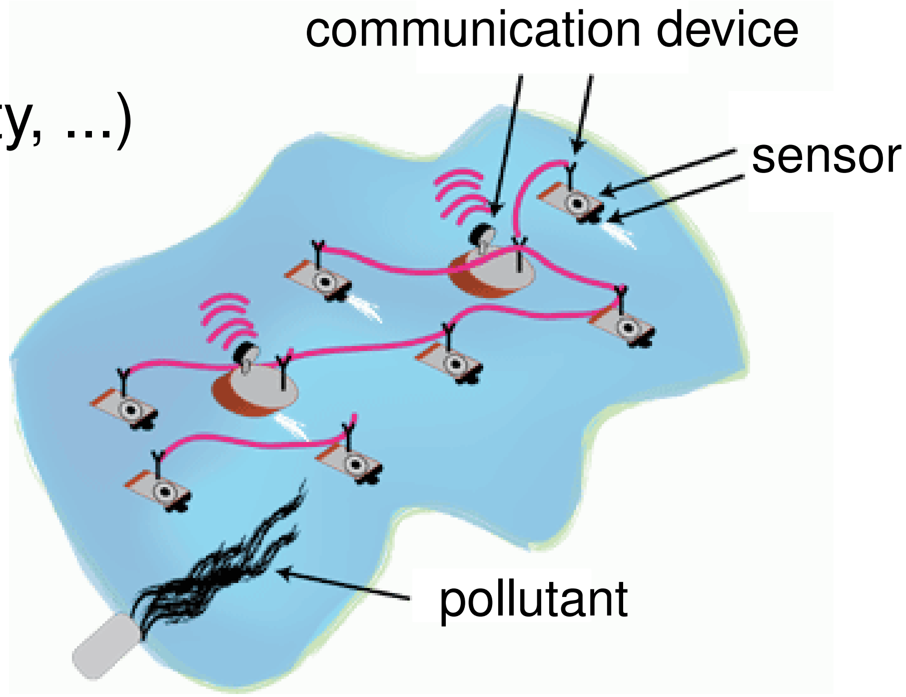
About me...

- Research Area: Algorithms for Wireless Sensor Networks
  - Wireless Communication
  - Localization



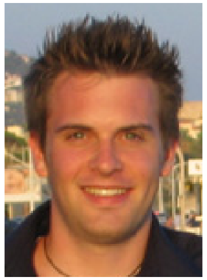
About Sensor Networks...

- Research Area: Algorithms for Wireless Sensor Networks
  - Wireless Communication
  - Localization
- Sensor network
  - network of small devices
  - equipped with sensors
  - collect data (temperature, humidity, ...)
  - communicate with each other
- Applications
  - detection of forest fires
  - intrusion detection
  - automatic watering of fields
  - pollutant detection



Topics of Today's Talk

- Scheduling of Wireless Transmissions
  (joint work with Bastian Katz)

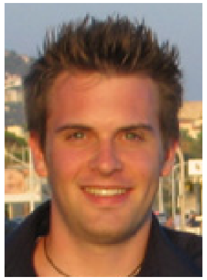


- Boundary Detection in Wireless Networks
  (joint work with Dennis Schieferdecker)

- Localization in Wireless Networks
  (joint work with Johannes Schmid)

Topics of Today’s Talk

- Scheduling of Wireless Transmissions
  (joint work with Bastian Katz)

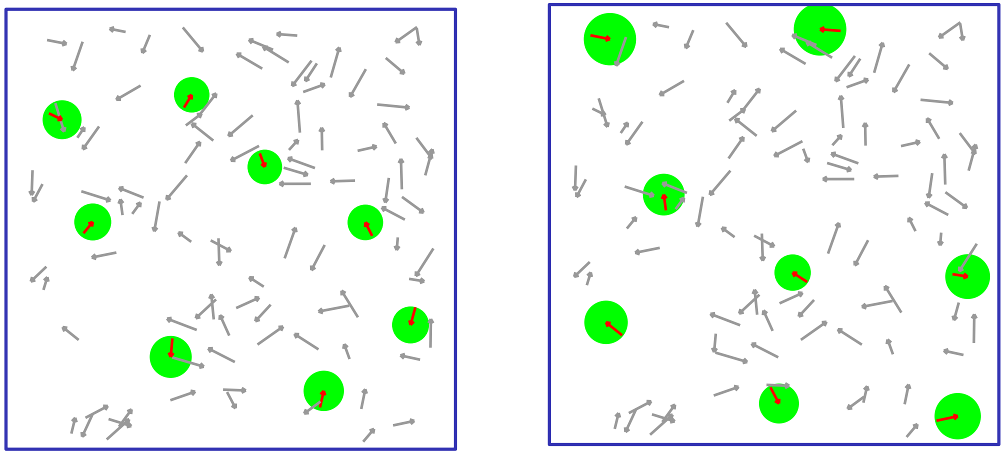


- Boundary Detection in Wireless Networks
  (joint work with Dennis Schieferdecker)

- Localization in Wireless Networks
  (joint work with Johannes Schmid)

Focus on main ideas, details are omitted

Transmission Scheduling

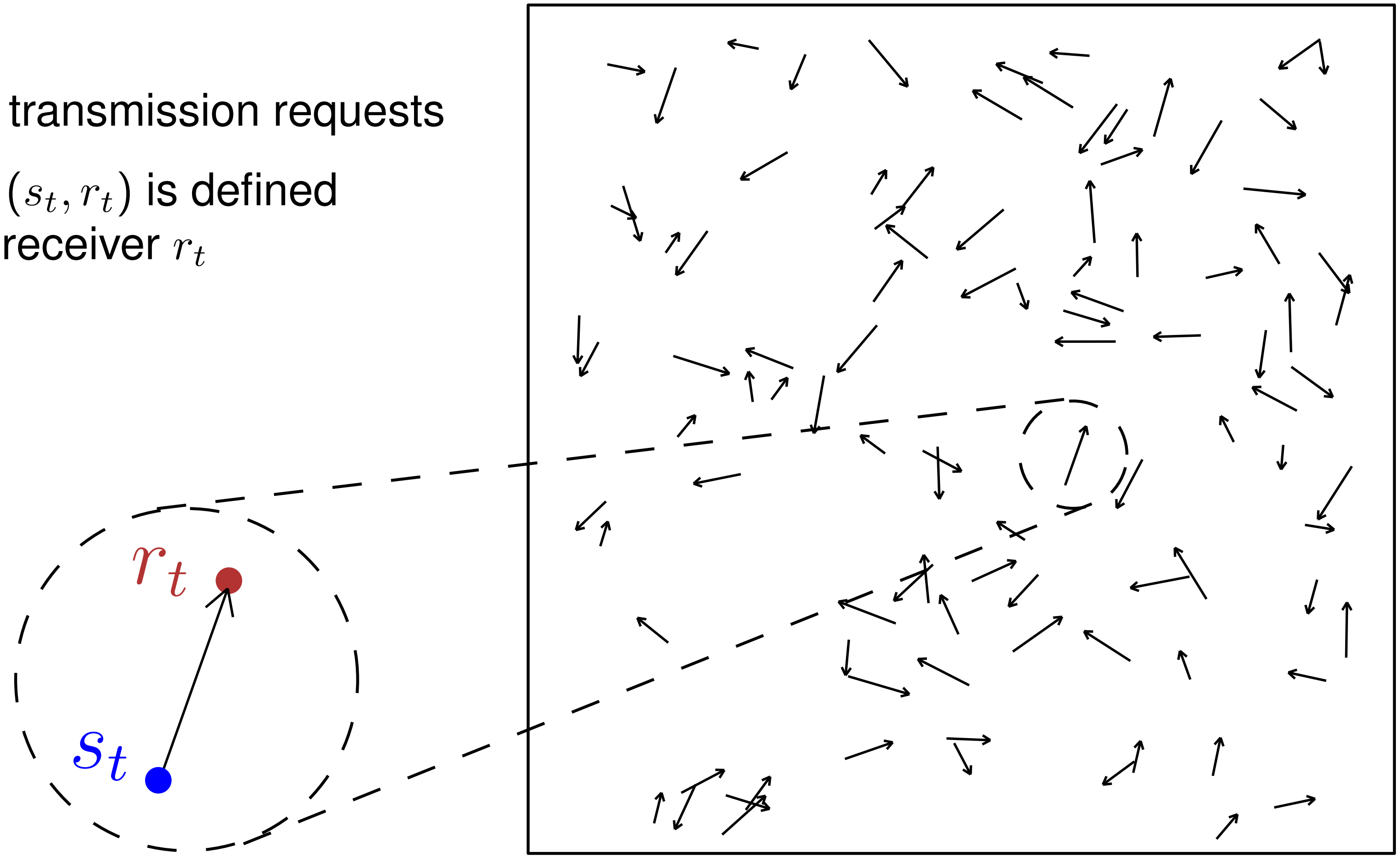


Transmission Scheduling

Problem Definition

Given:

- Set $\mathcal{T}$ of wireless transmission requests
- Transmission $t = (s_t, r_t)$ is defined by sender s_t and receiver r_t

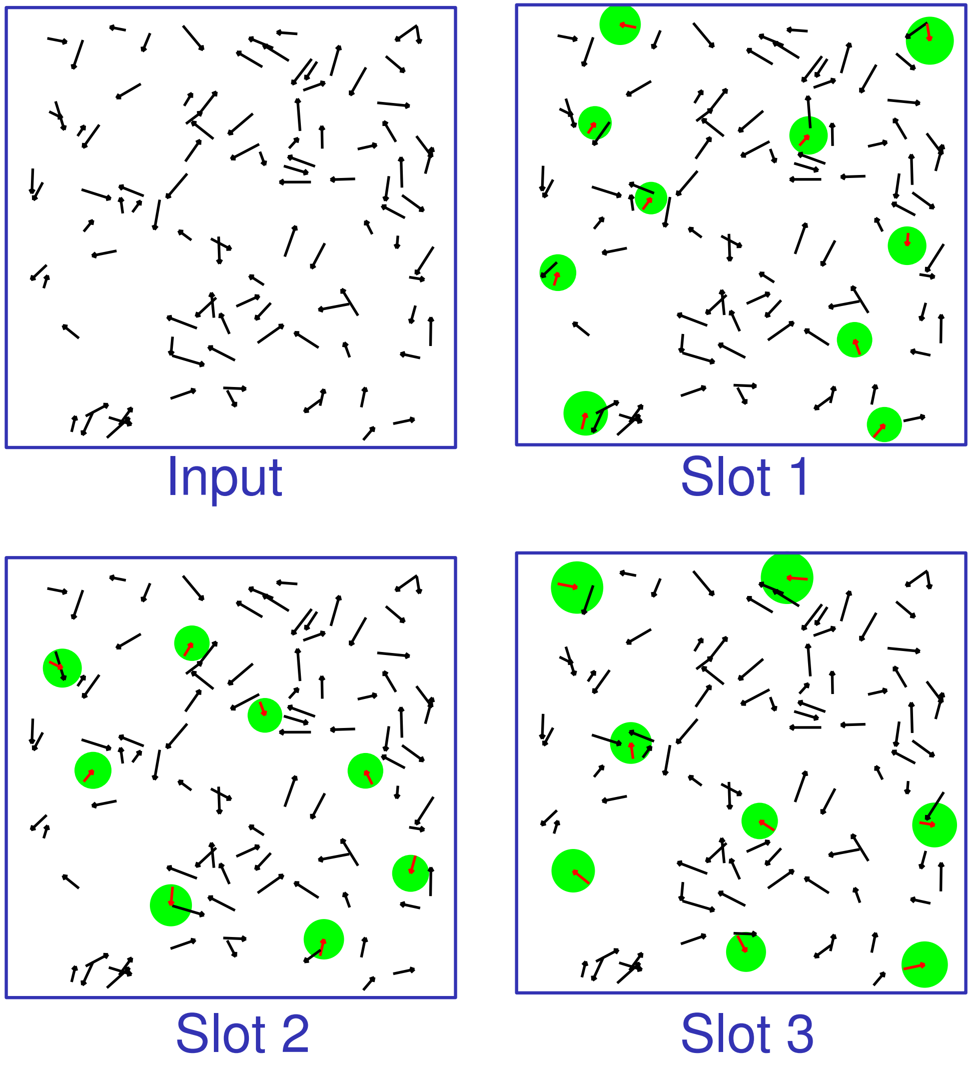


Transmission Scheduling

Problem Definition

Given:

- Set $\mathcal{T}$ of wireless transmission requests
- Transmission $t = (s_t, r_t)$ is defined by sender s_t and receiver r_t



Goal:

- Distribution of transmissions to time slots (TMDA schedule)
- No failures due to interference
- Minimum number of slots

Transmission Scheduling

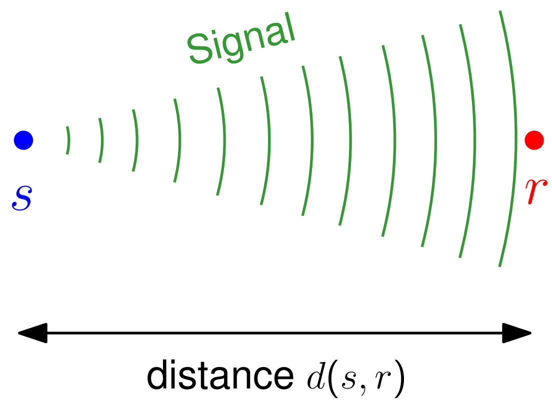
Problem Definition

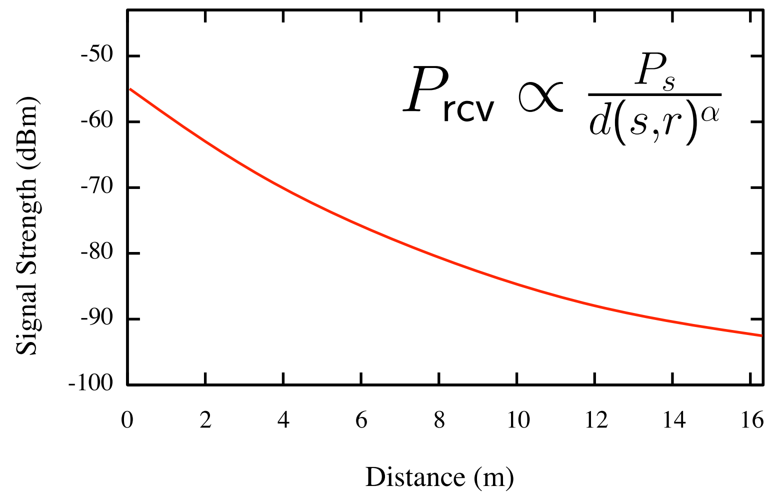
How do we know whether a transmission is successful?

Transmission Scheduling

Problem Definition

How do we know whether a transmission is successful?



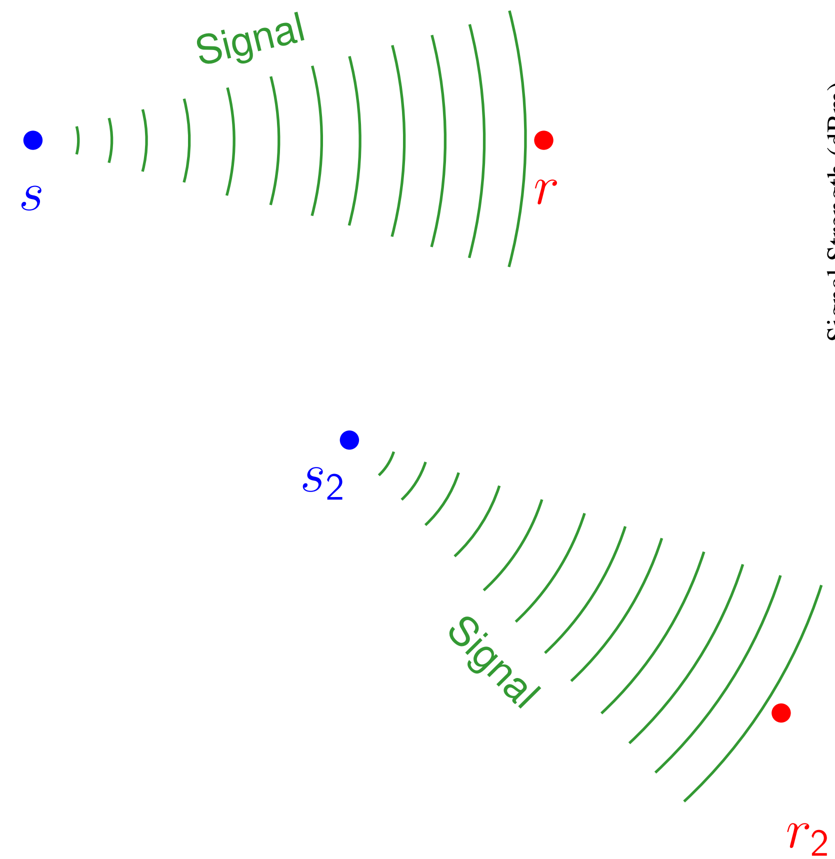


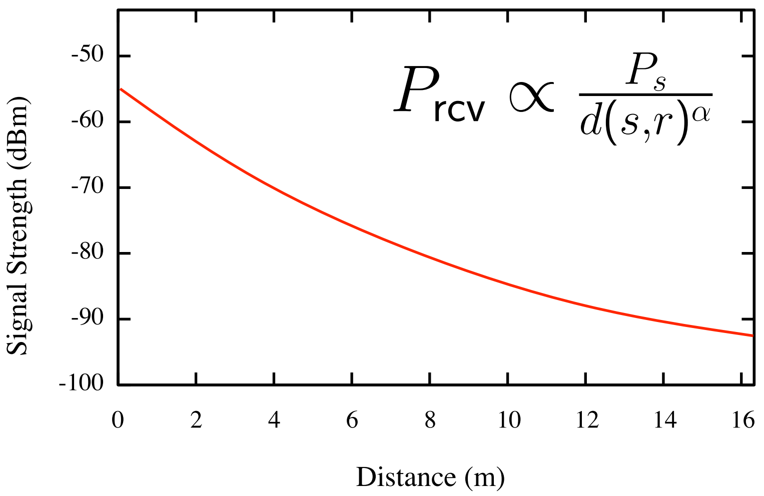
- Received signal strength depends on distance between s and r
- Distance dependence is given by a power law
- Path loss exponent α defines how fast the signal decays (≈ 2 in free space, between 2 and 5 in buildings)

Transmission Scheduling

Problem Definition

How do we know whether a transmission is successful?

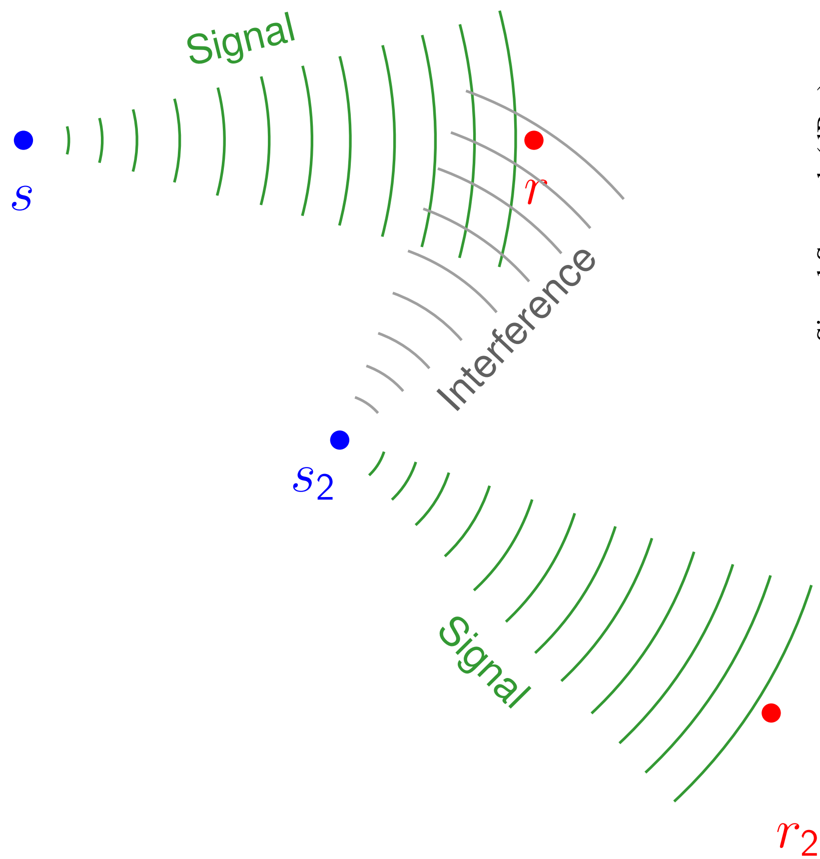


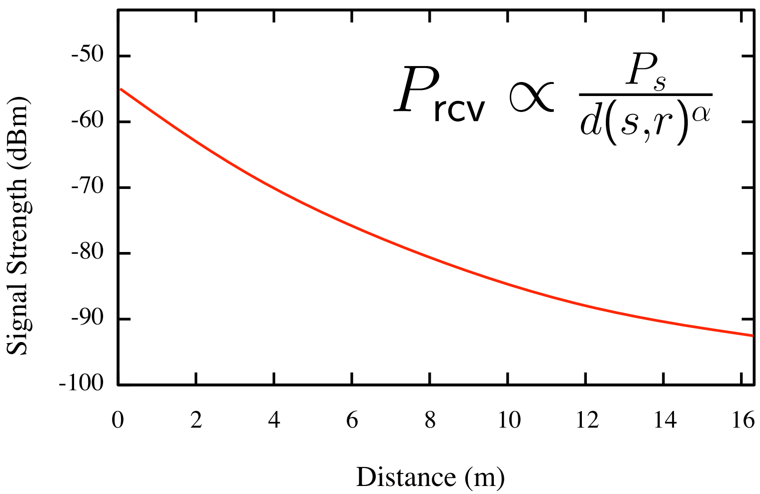


Transmission Scheduling

Problem Definition

How do we know whether a transmission is successful?

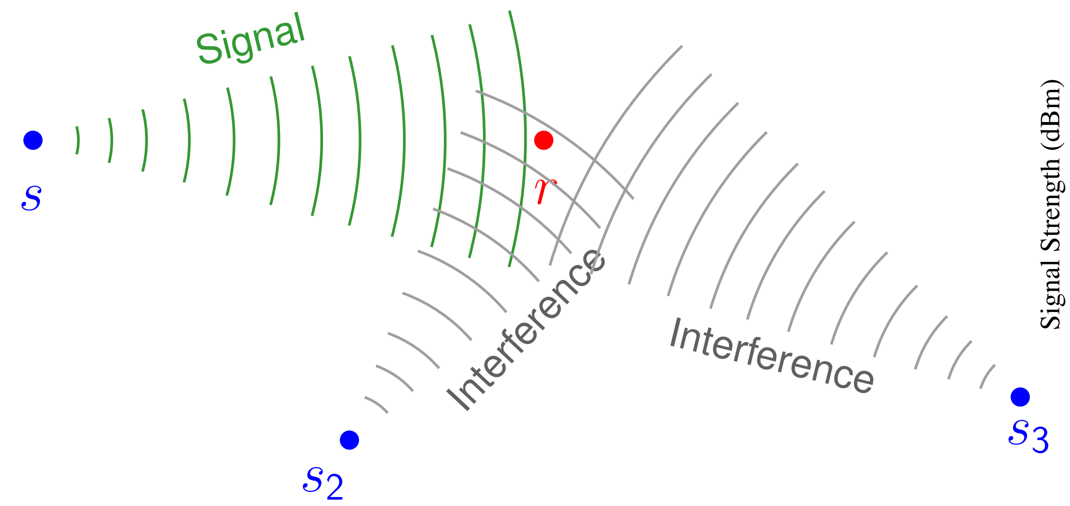


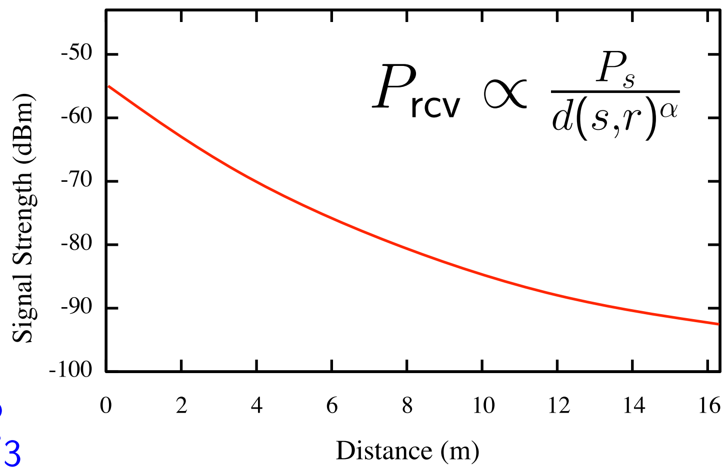


Transmission Scheduling

Problem Definition

How do we know whether a transmission is successful?

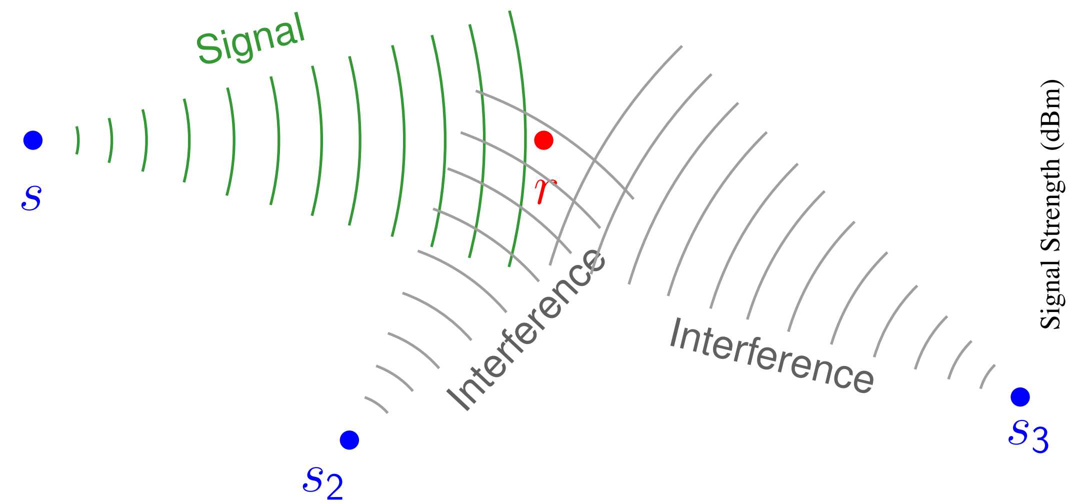


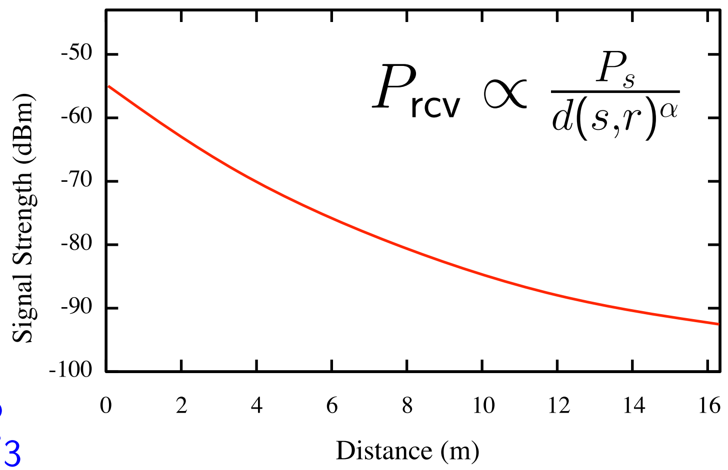


Transmission Scheduling

Problem Definition

How do we know whether a transmission is successful?





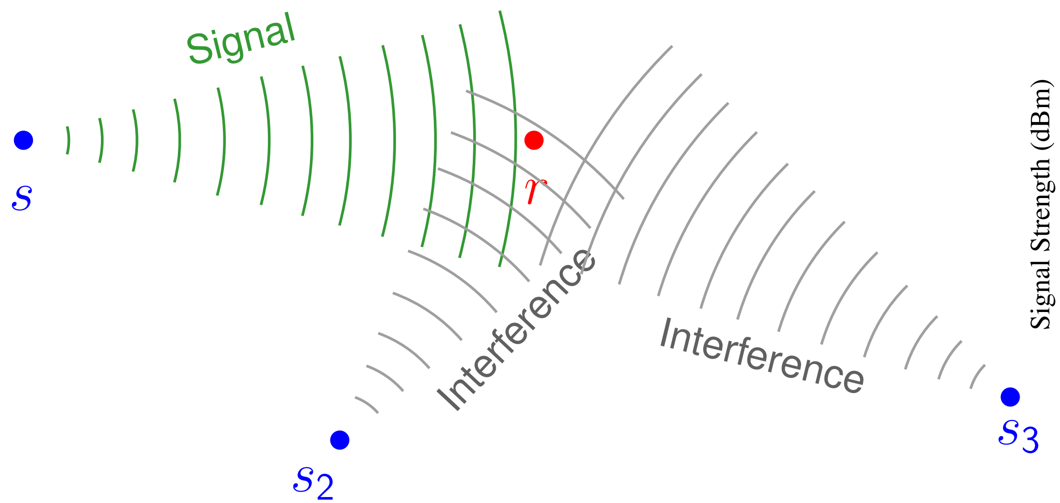
- Transmission is successful if SINR exceeds some threshold β

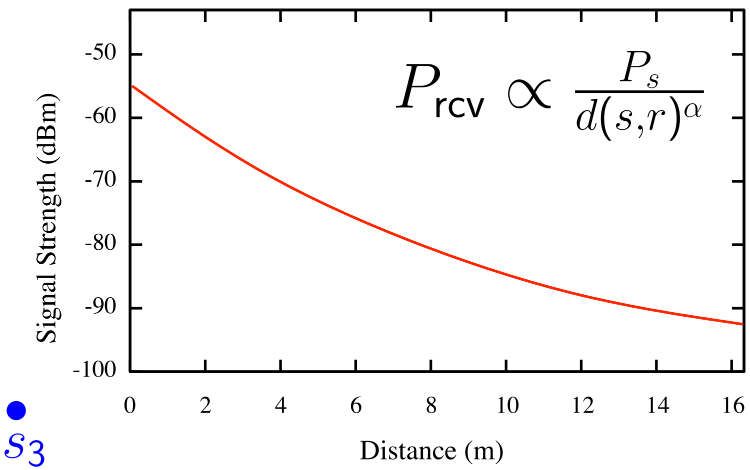
$$\frac{\text{Signal}}{\sum \text{Interferences} + \text{Background Noise}} \geq \beta \qquad (\beta \approx 10)$$

Transmission Scheduling

Problem Definition

Geometric SINR model   ($SINR_G$ model)





- Transmission is successful if SINR exceeds some threshold β

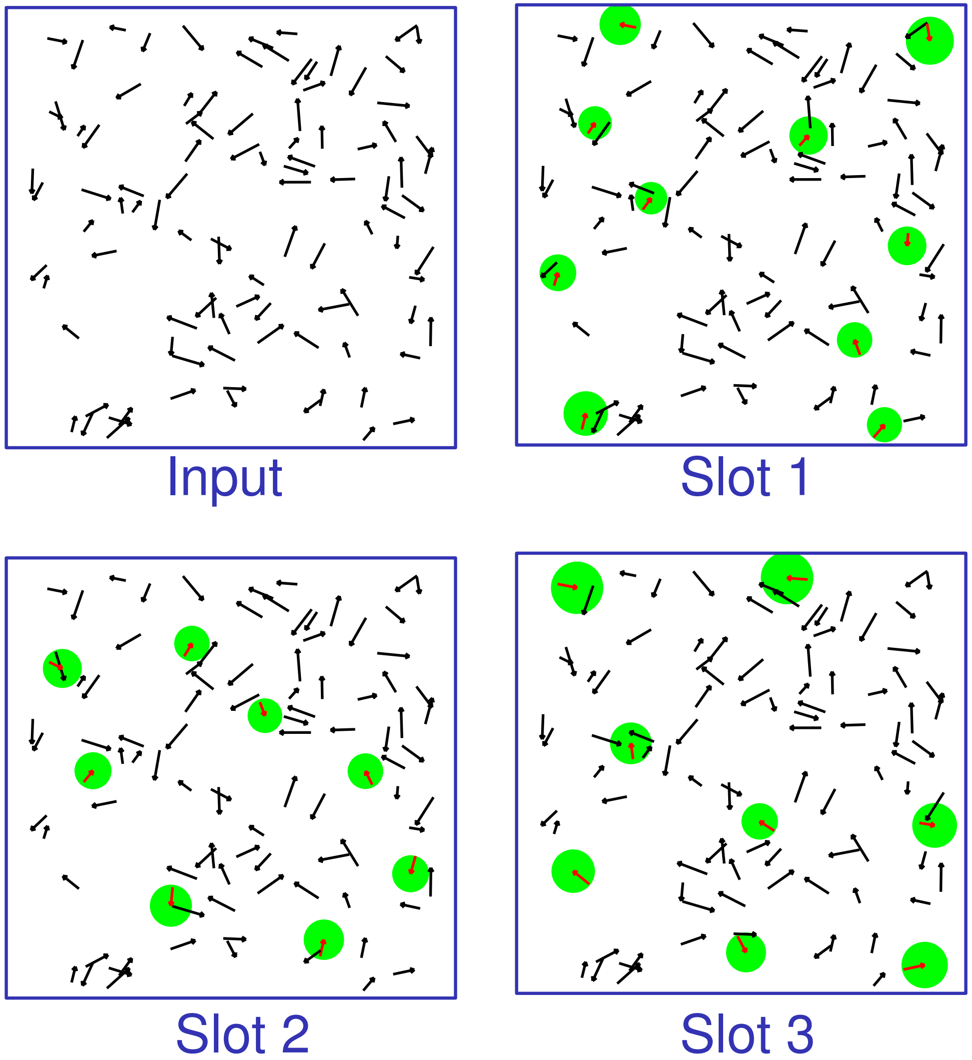
$$\frac{\text{Signal}}{\sum \text{Interferences + Background Noise}} \geq \beta \qquad (\beta \approx 10)$$

Transmission Scheduling

Problem Definition

SINR condition has
to be fulfilled!

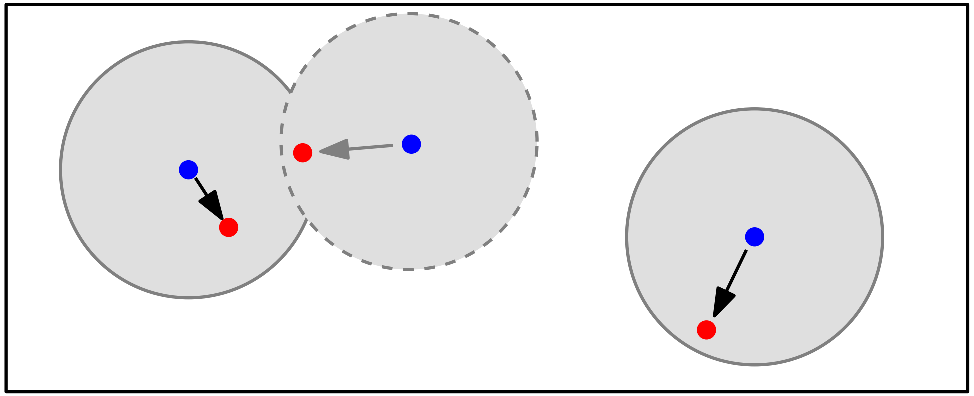
$$\frac{\text{Signal}}{\text{Interference} + \text{Background Noise}} \geq \beta$$



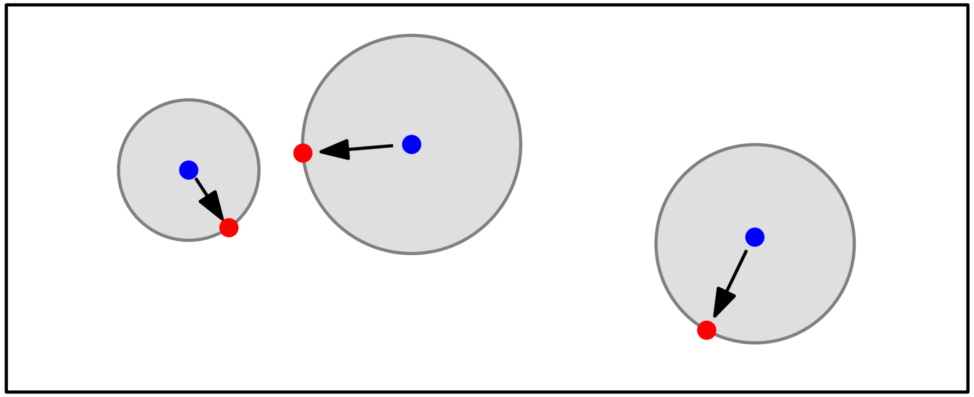
Transmission Scheduling

Problem Variants

- Scheduling with fixed transmission powers



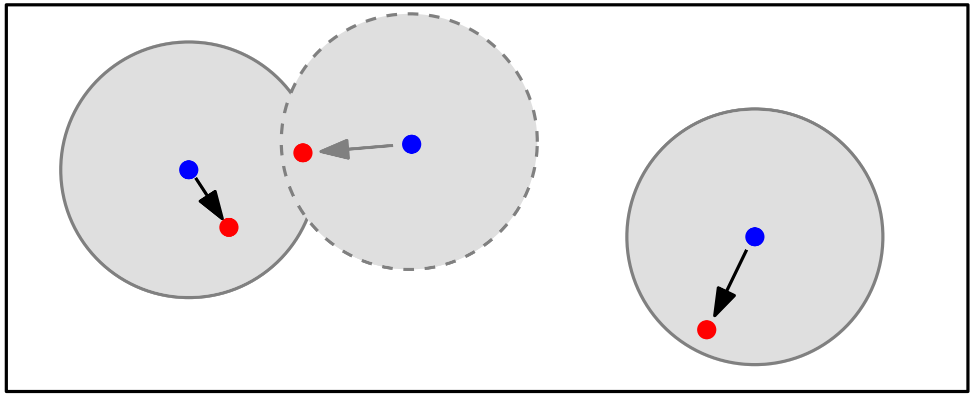
- Scheduling with power control



Transmission Scheduling

Complexity

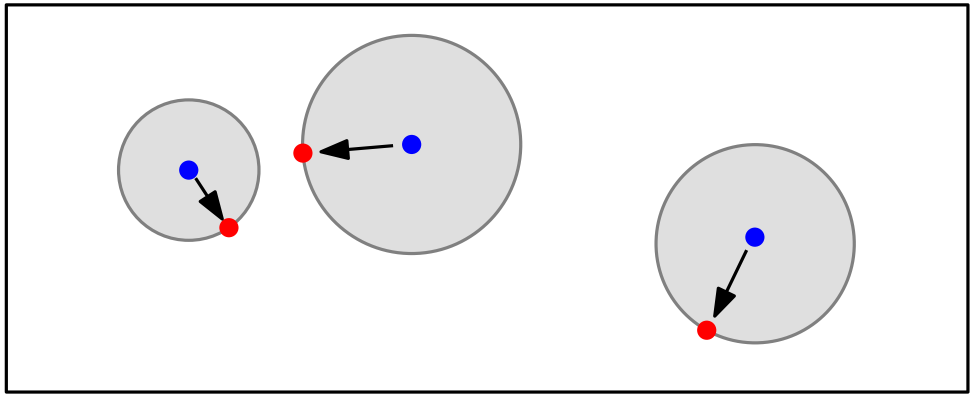
- Scheduling with fixed transmission powers



NP-hard

(Goussevskaia, Oswald, Wattenhofer, 2007)

- Scheduling with power control



NP-hard if powers are bounded

(Katz, Völker, Wagner, 2009)

Transmission Scheduling

Possible Approaches

- Exact solutions for small instances
- Heuristics
- Approximation algorithms
- Randomized algorithms with collision detection

Transmission Scheduling

Possible Approaches

- Exact solutions for small instances
  - Constraint Programming (CP)
  - Integer Linear Programming (ILP)
  - only up to $\approx$ 100 transmissions
- Approximation algorithms
  - Scheduling with local information
- Heuristics
  - Heuristic for scheduling with power control

Transmission Scheduling

Heuristic for Scheduling with Power Control

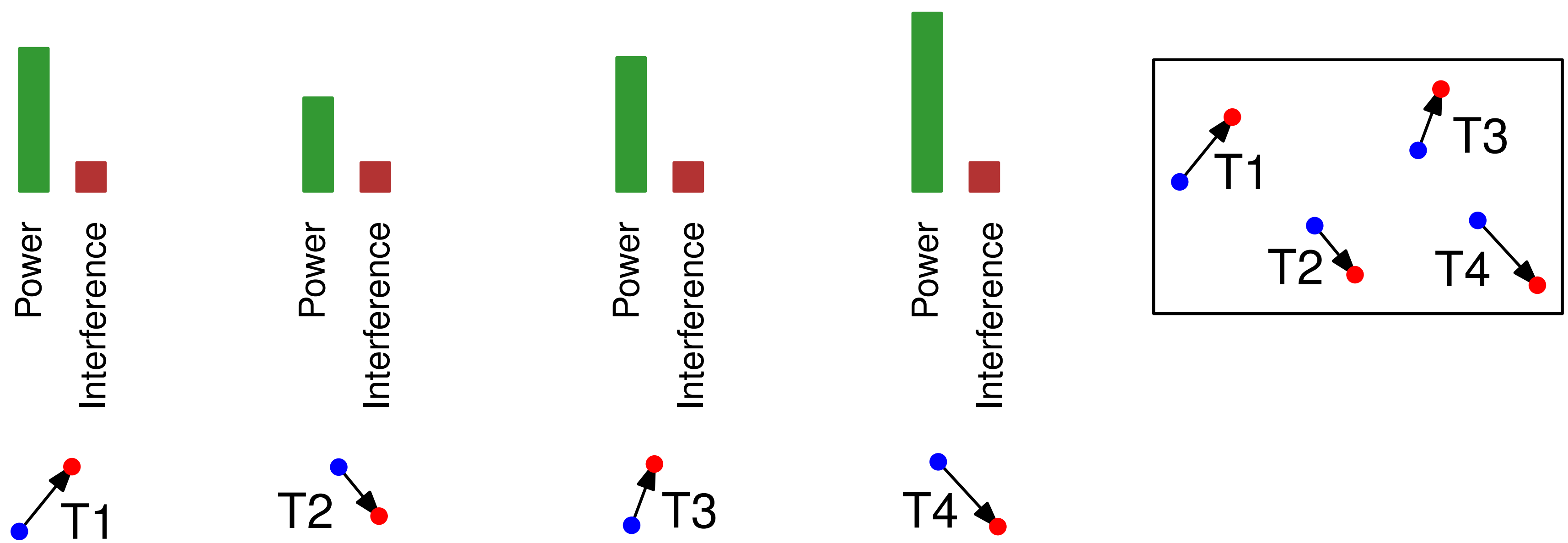
Usual approach of existing methods

- sort transmissions according to some measure (e.g., sender-receiver-distance or link gain)
- fill time slots greedily
- as soon as a slot is filled, open another slot
- to decide whether a transmission fits into a time slot, use an iterative method for power control

Transmission Scheduling

Heuristic for Scheduling with Power Control

Iterative Power Control

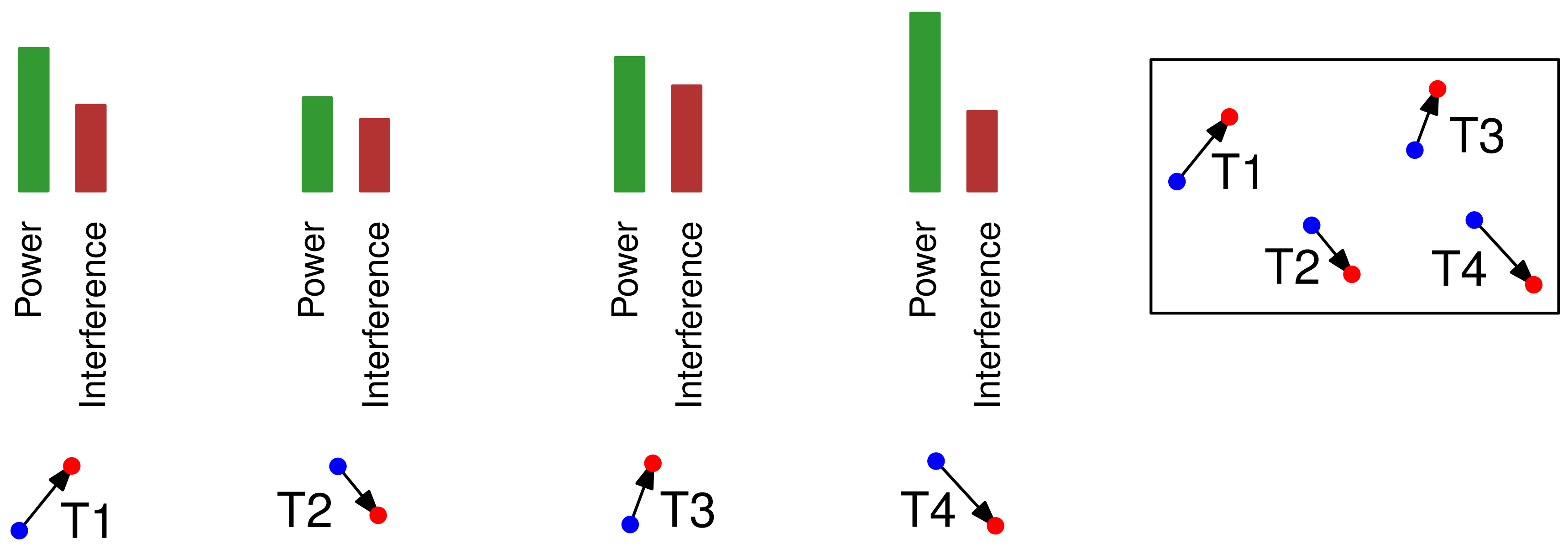


Complexity: $O(c \cdot n^2)$, with $c \approx 20$ number of iterations

Transmission Scheduling

Heuristic for Scheduling with Power Control

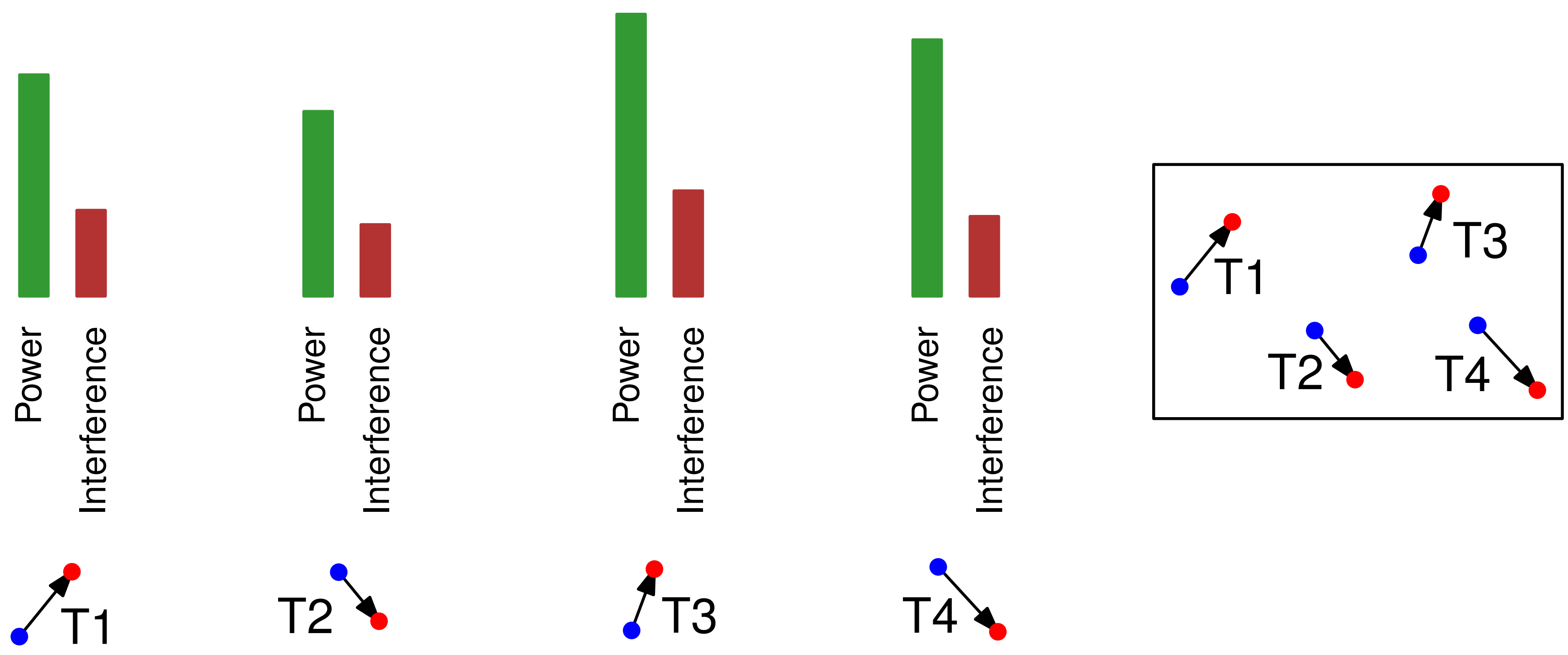
Iterative Power Control



Complexity: $O(c \cdot n^2)$, with $c \approx 20$ number of iterations

Heuristic for Scheduling with Power Control

Iterative Power Control

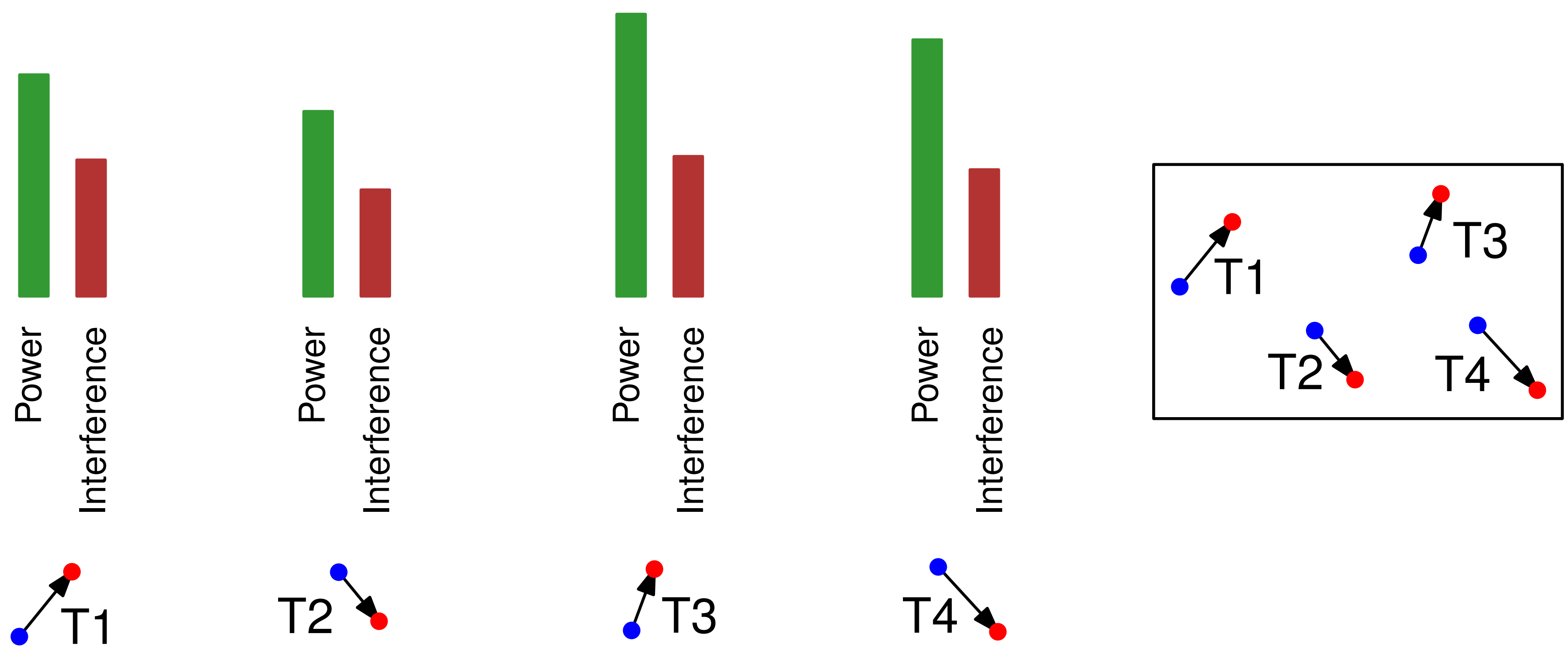


Complexity: $O(c \cdot n^2)$, with $c \approx 20$ number of iterations

Transmission Scheduling

Heuristic for Scheduling with Power Control

Iterative Power Control



Complexity: $O(c \cdot n^2)$, with $c \approx 20$ number of iterations

Transmission Scheduling

Heuristic for Scheduling with Power Control

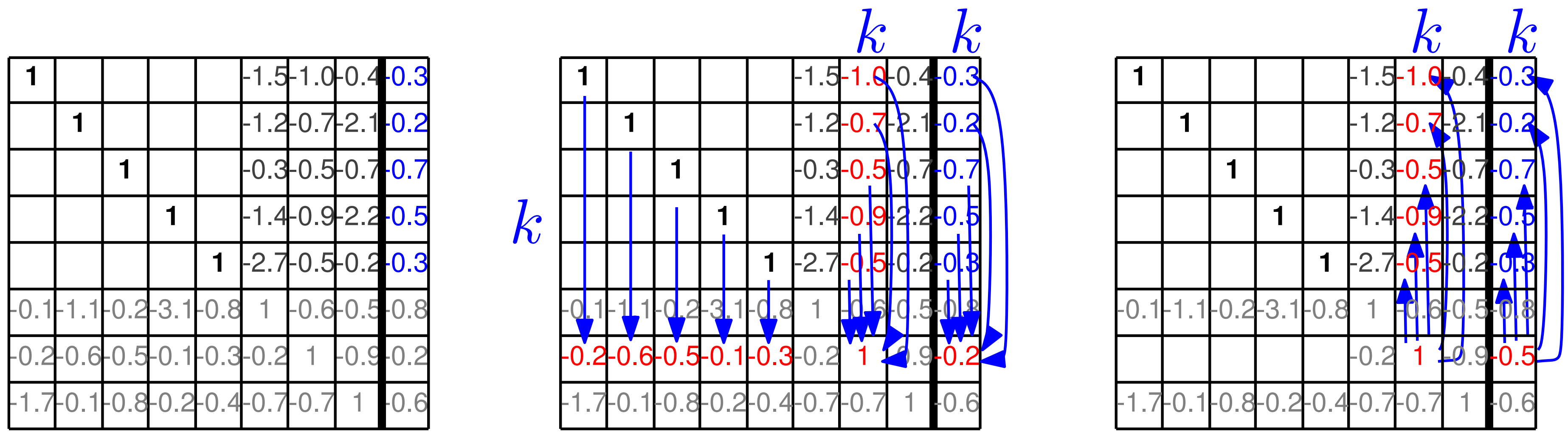
Our approach

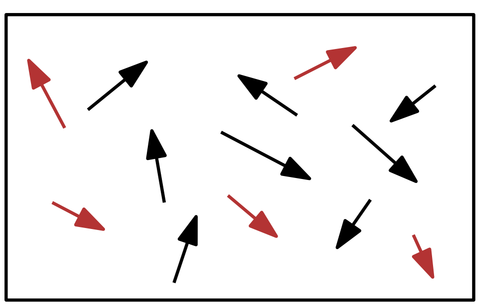
- fill several slots simultaneously
- do not add the transmissions in fixed order but choose always the transmission which fits best
  (= transmissions which minimizes additional power)
- requires frequent updates of transmission powers
  $\Rightarrow$ better solution to power control problem necessary

Transmission Scheduling

Heuristic for Scheduling with Power Control

New method for efficient power control





k active transmissions

Prediction of optimal transmission powers in $O(k)$ time instead of $O(ck^2)$ time.

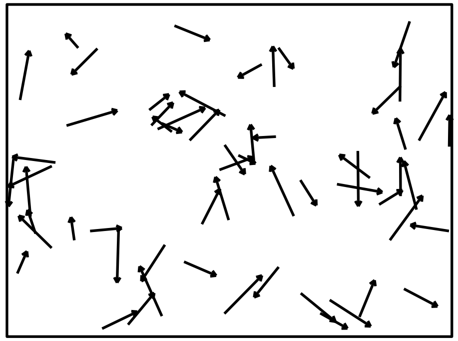
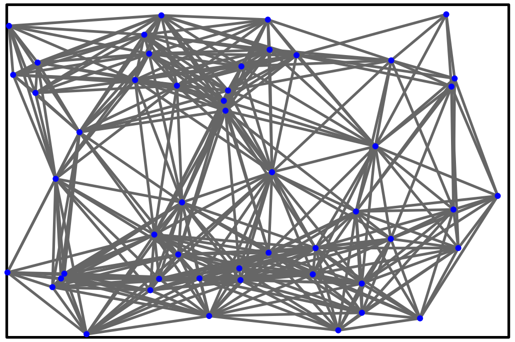
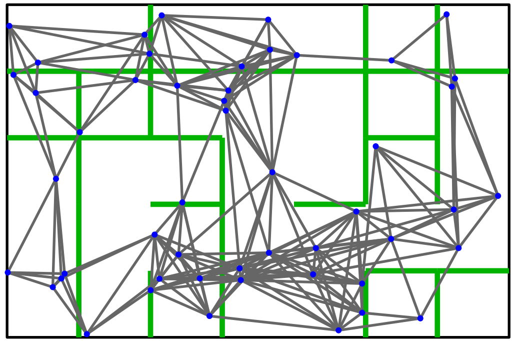
$\Rightarrow$ better heuristics with same time complexity

Transmission Scheduling

Heuristic for Scheduling with Power Control

Simulation results

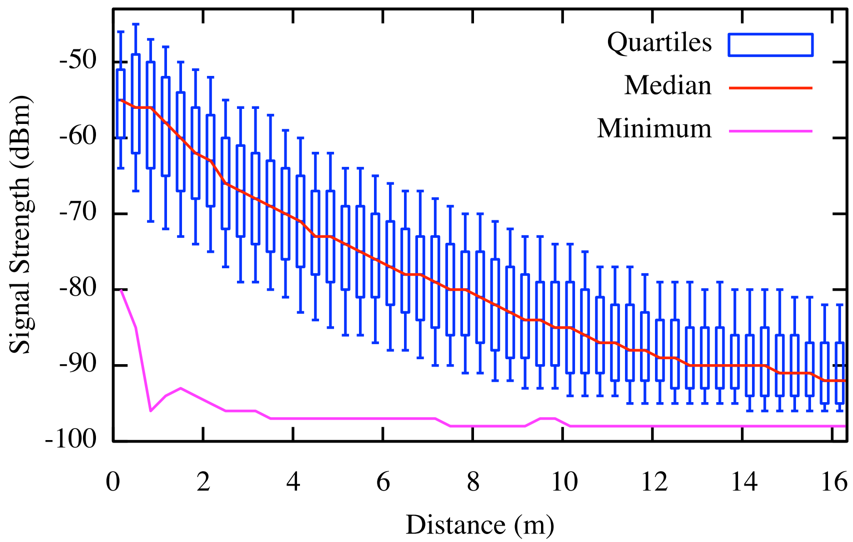
- up to 50% higher throughput than existing approaches
- same throughput with transmission power savings of about 30%
- same time complexity as existing approaches thanks to new power control algorithm

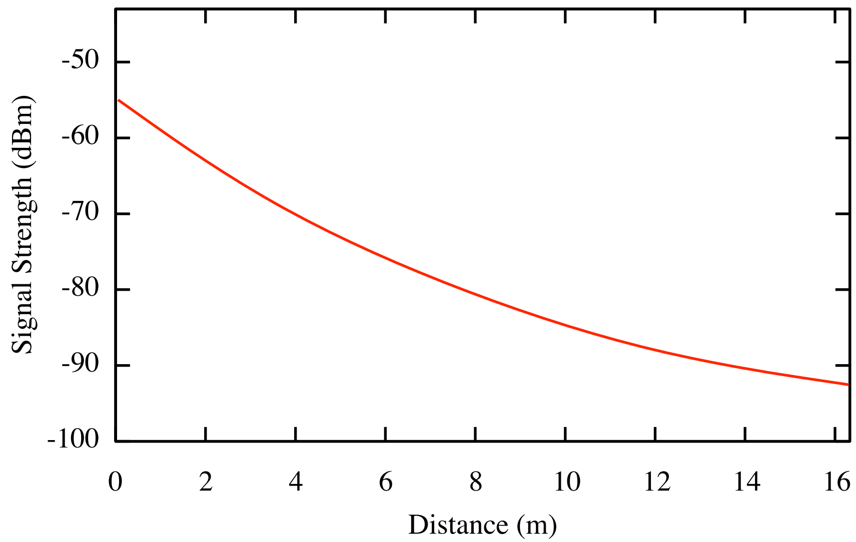
Transmission Scheduling

Shortcomings of the SINR$_G$ model

Comparison with Reality



Measurement in 60 node sensor network
(based on 300.000 packets)

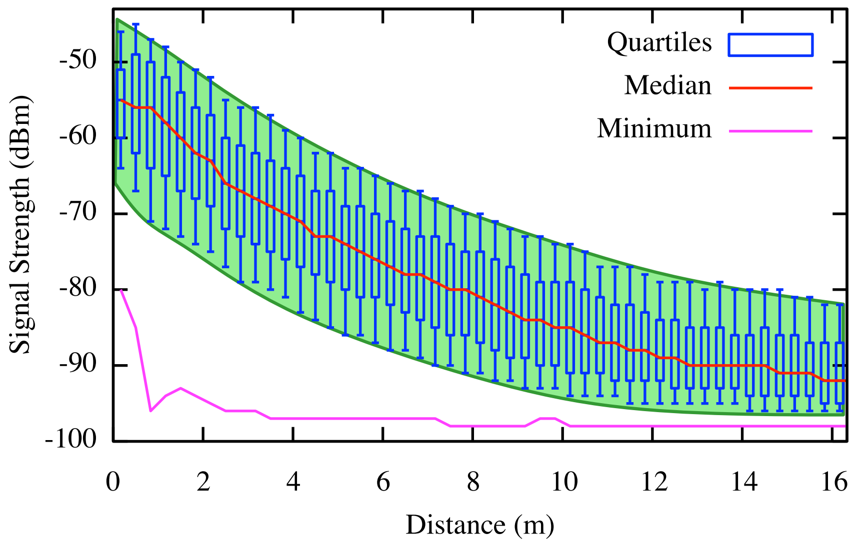


SINR$_G$ model

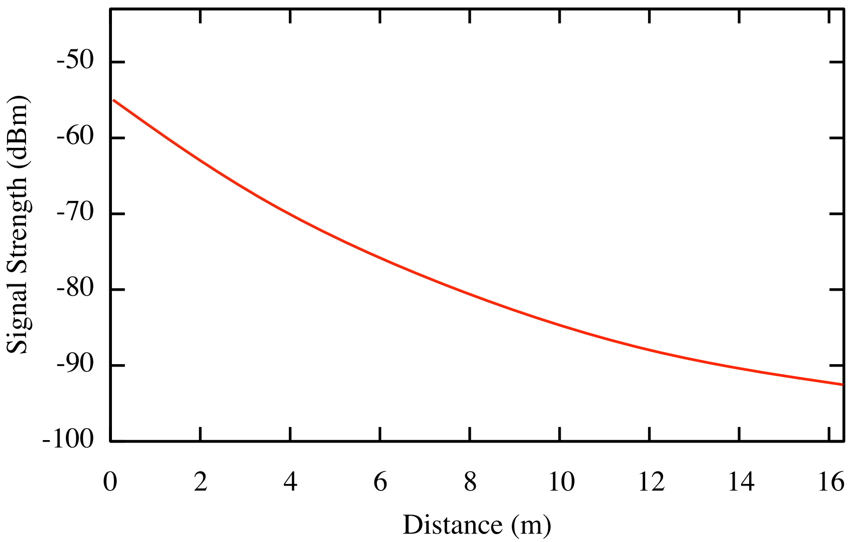
Transmission Scheduling

Shortcomings of the SINR_G model

Comparison with Reality



Measurement in 60 node sensor network
(based on 300.000 packets)



SINR_G model

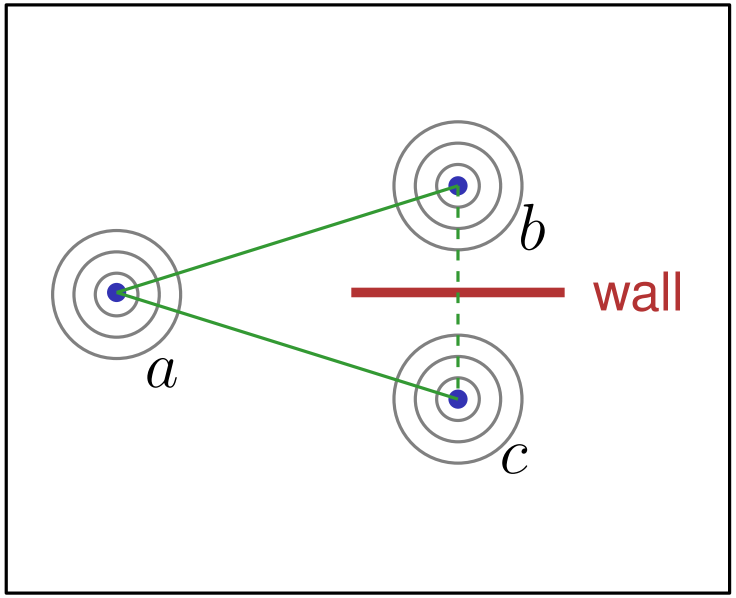
area between 10^{th} percentile and 90^{th} percentile contains 80% of the measurements

Factor of 200 in signal strength!

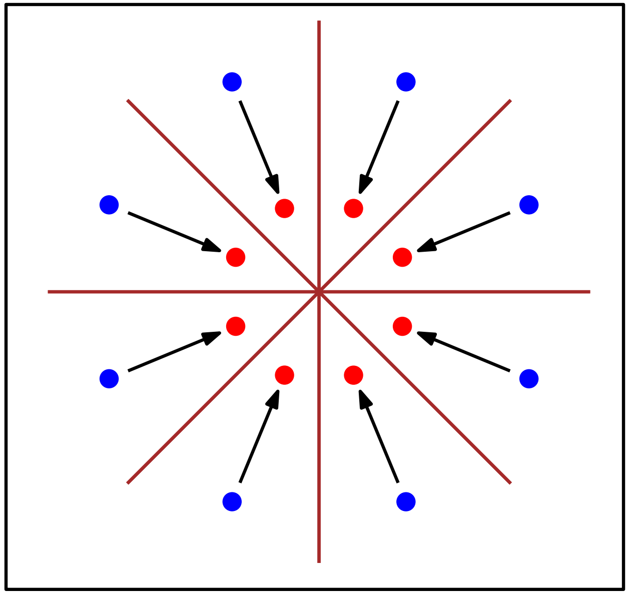
Transmission Scheduling

Shortcomings of the SINR_G model

Problems



$\Rightarrow$

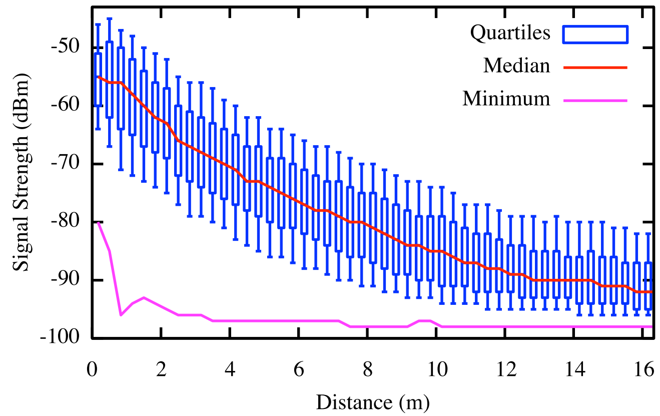


Individual walls cannot be modeled

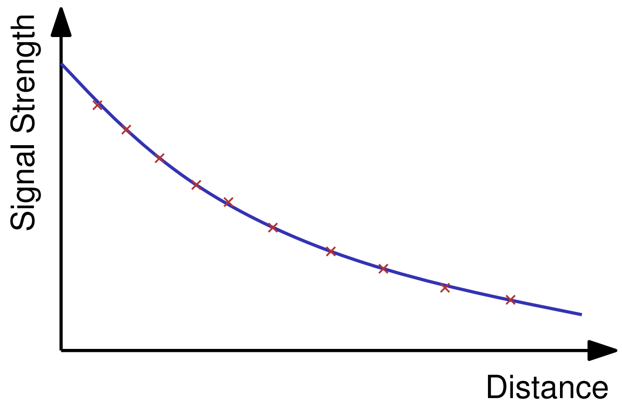
Can make results arbitrarily bad

Transmission Scheduling

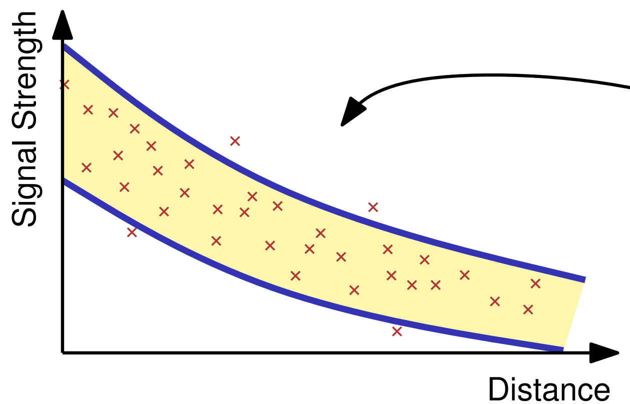
Possible alternatives



Reality



SINR_G



Alternative I

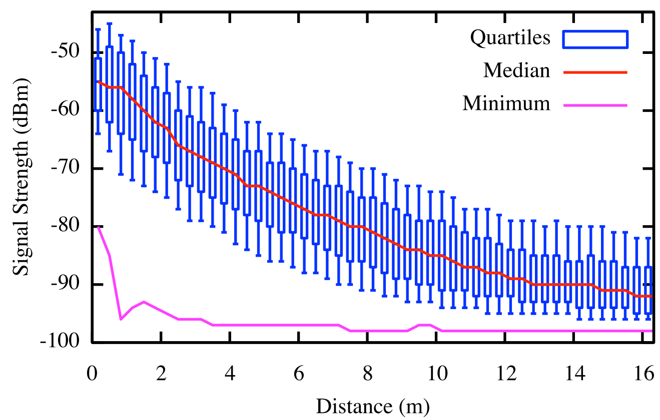
Example: Generalized Physical Model

$$\frac{1}{\theta}\frac{P_i}{d(s_i,r_j)} \leq P_{ij} \leq \theta\frac{P_i}{d(s_i,r_j)}$$

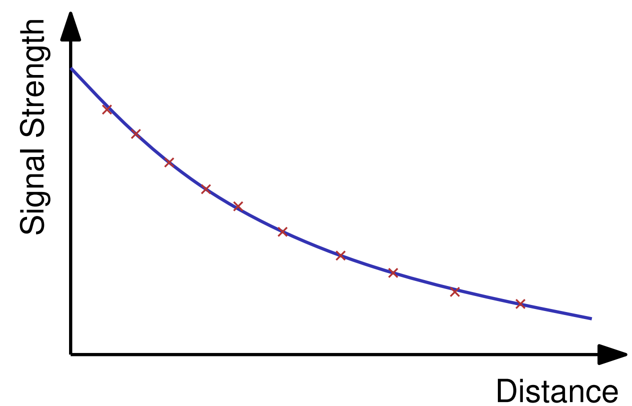
(Moscibroda, Wattenhofer, Zollinger, 2006)

Transmission Scheduling

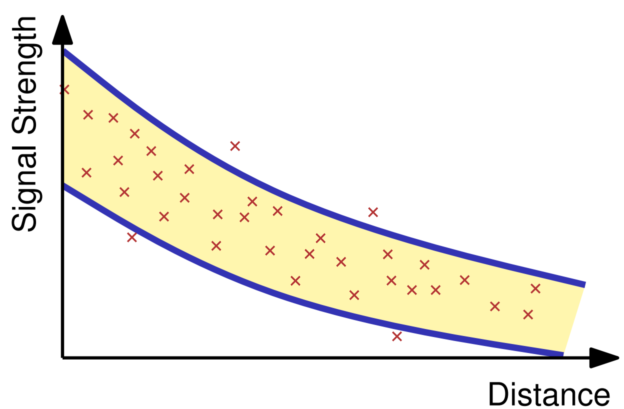
Possible alternatives



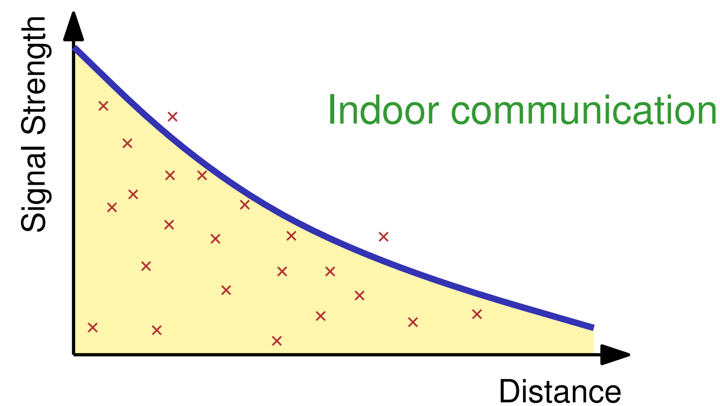
Reality



$SINR_G$

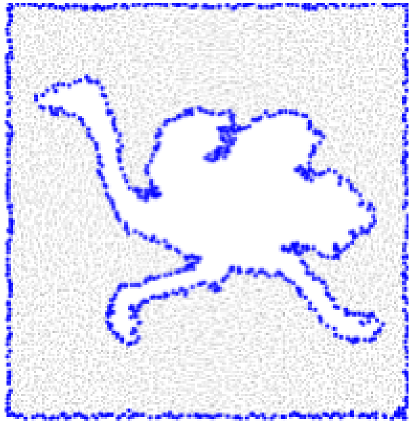


Alternative I



Alternative II

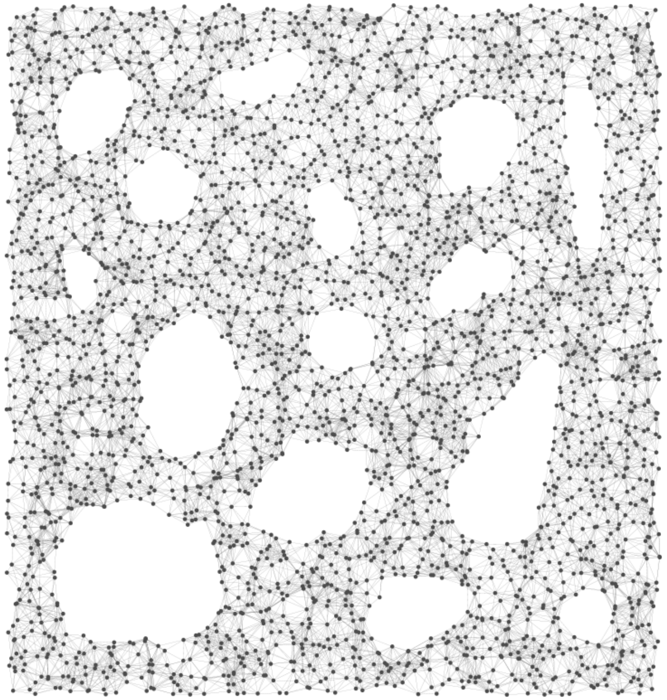
Boundary Detection



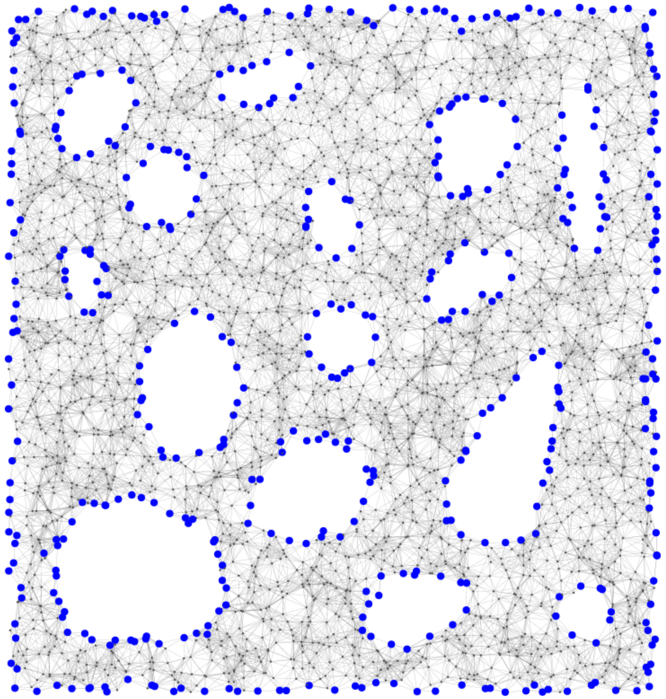
Boundary Detection

Problem Definition

- Recognition of holes and boundaries in WSNs
  - no knowledge about node positions
  - using only connectivity information



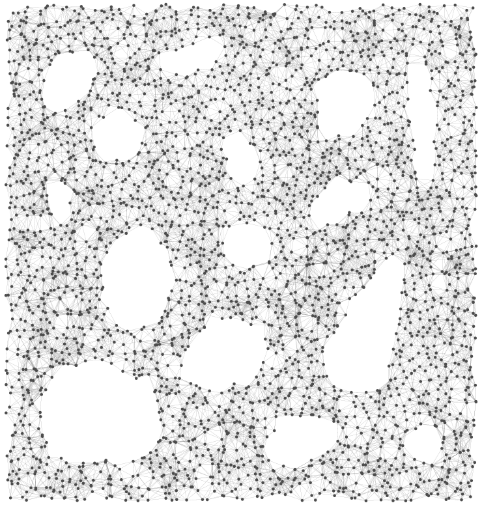
$\Rightarrow$



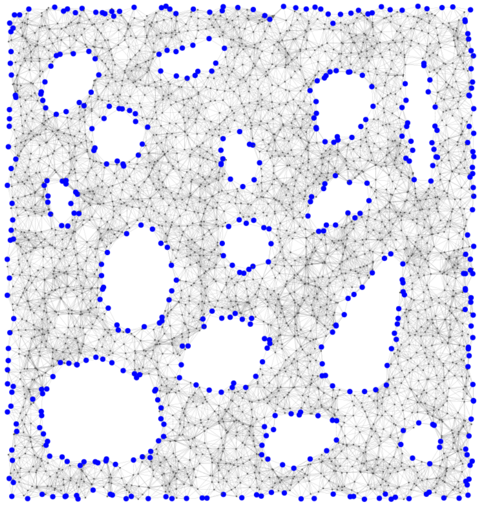
Boundary Detection

Problem Definition

- Recognition of holes and boundaries in WSNs
  - no knowledge about node positions
  - using only connectivity information
- Motivation
  - detection of areas of insufficient coverage
  - efficient routing
  - load balancing



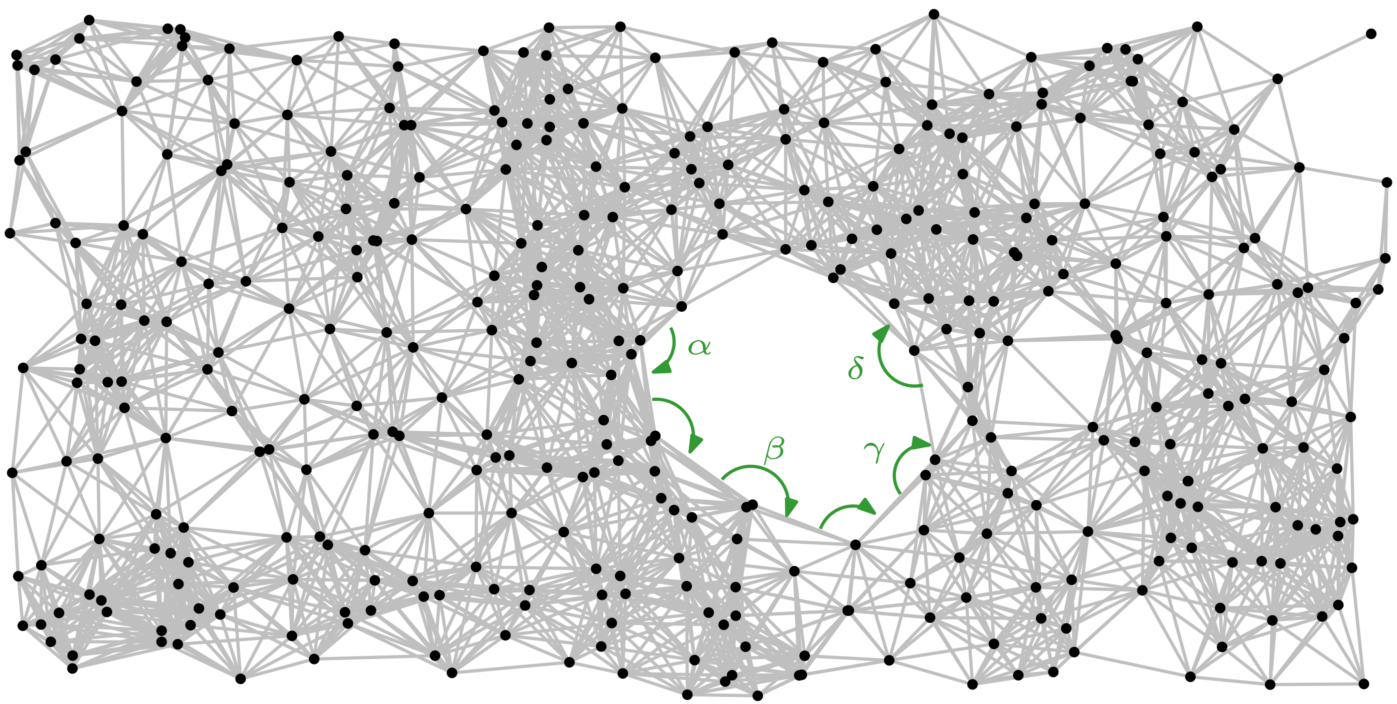
$\Rightarrow$



Boundary Detection

Existing Approaches

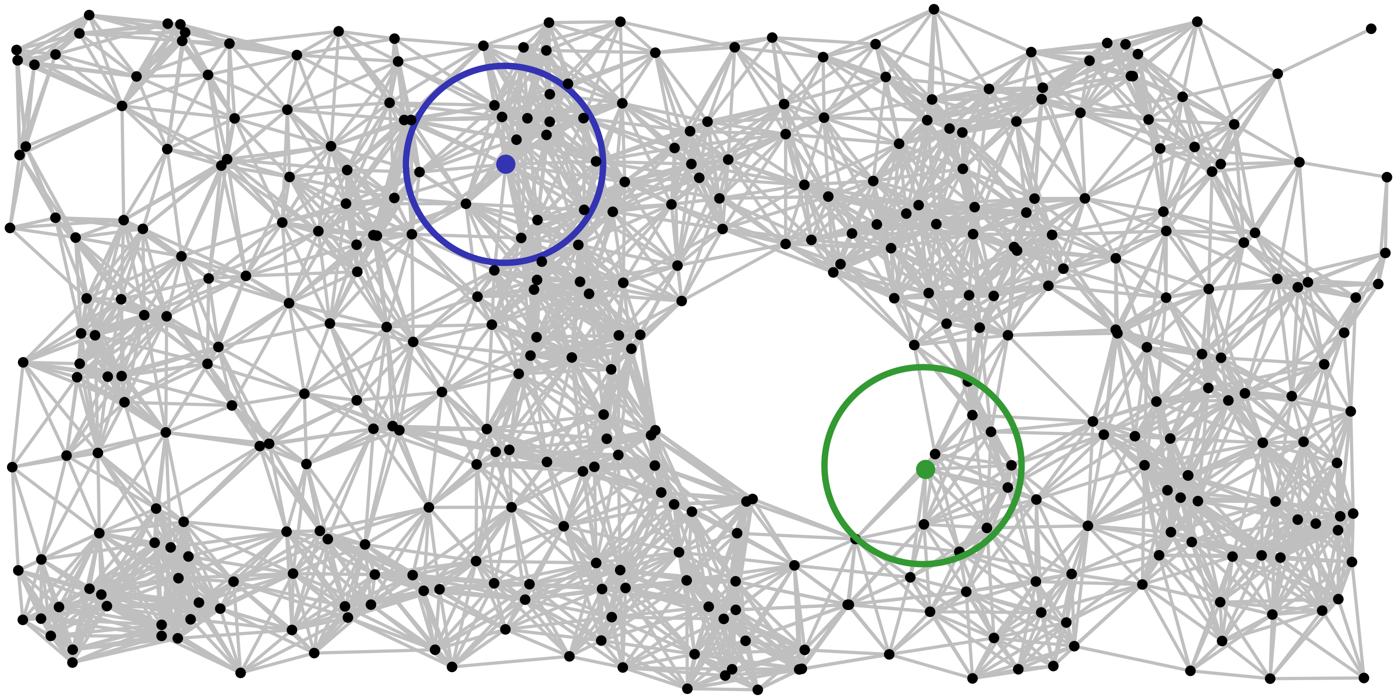
- Geometrical Approaches
  - use information about node positions, distances, or angular relationships



Boundary Detection

Existing Approaches

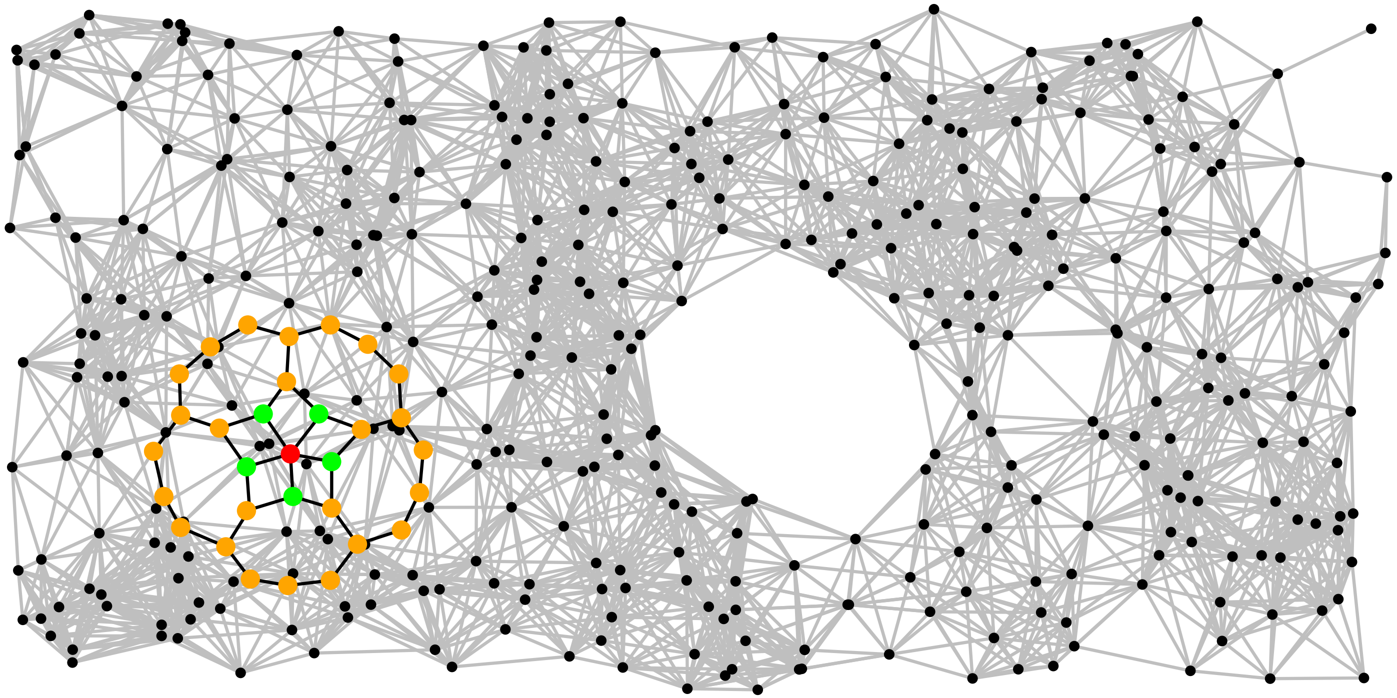
- Statistical Approaches
  - use statistical properties such as low node degree, etc.



Boundary Detection

Existing Approaches

- Topological Approaches
  - use topological structure of connectivity graph



Boundary Detection

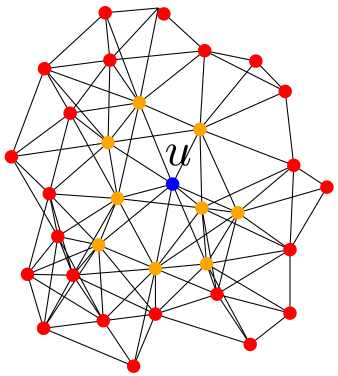
Design Goals

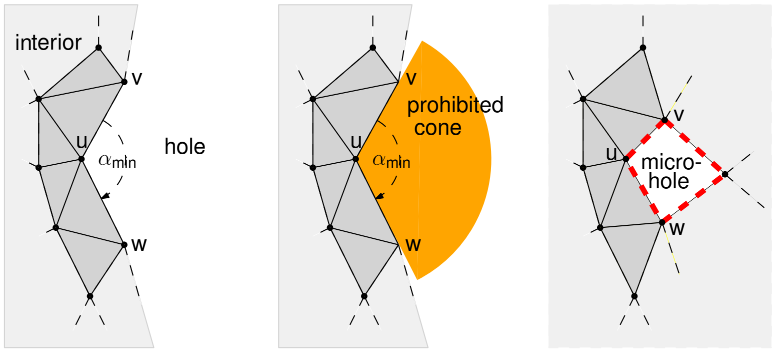
- simple (easy to understand and implement)
- only connectivity information
- little communication ($\Rightarrow$ dynamic networks)
- fast computation ($\Rightarrow$ sensor networks)
- stable with respect to different
  - node degrees
  - node distributions
  - communication models

Boundary Detection

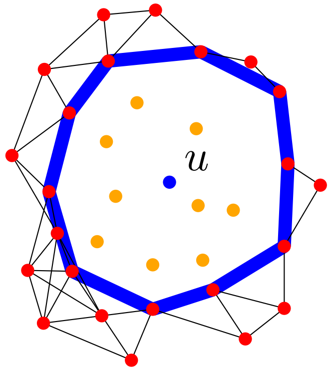
2 New Methods

- Multidimensional Scaling Boundary Recognition (MDS-BR)

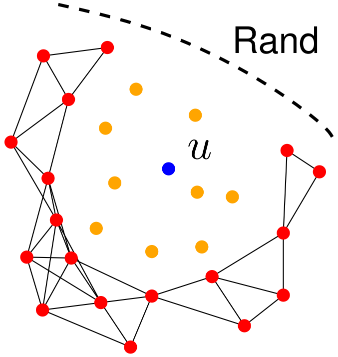




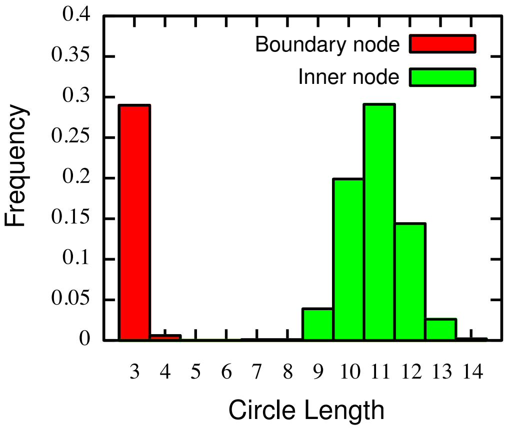
- Enclosing Circle Boundary Recognition (EC-BR)



Inner node



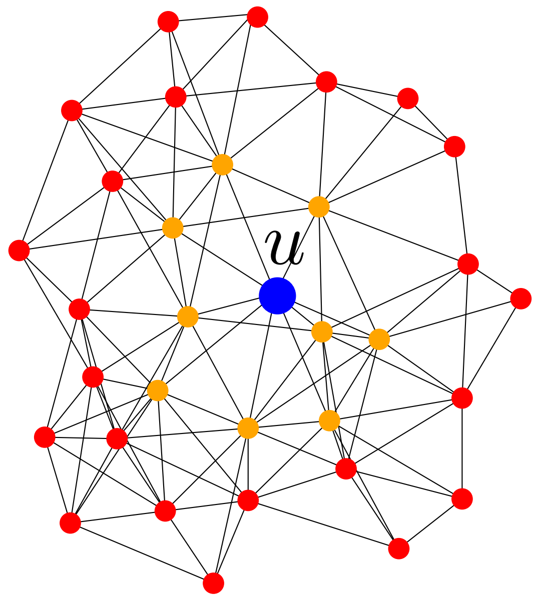
Boundary node



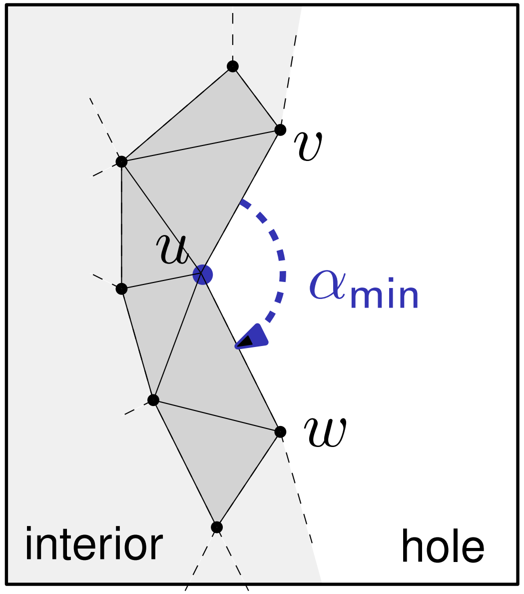
Boundary Detection

Multidimensional Scaling Boundary Recognition

Basic Idea



Estimate relative positions
of 2-hop neighborhood



Check some
angular conditions

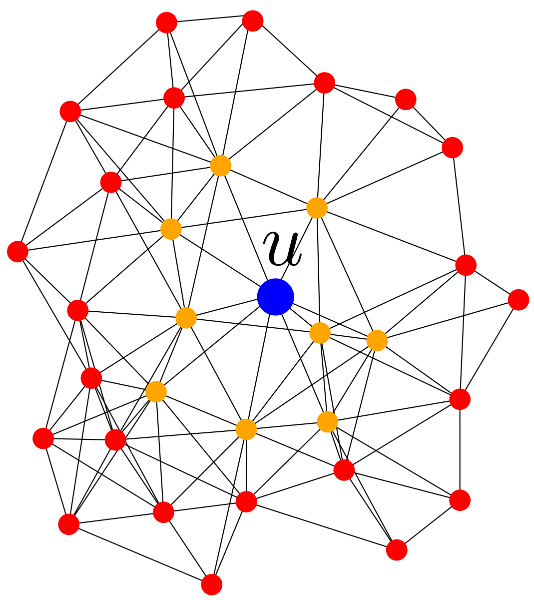
Boundary Detection

Multidimensional Scaling Boundary Recognition

- 1^{st} step: get connectivity information of 2-hop neighborhood

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & 1 & \cdots \\ 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Adjacency matrix of 2-hop neighborhood



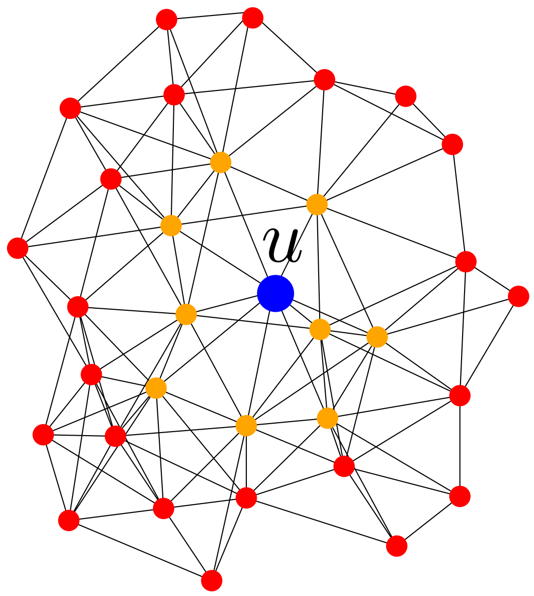
(2 messages per node)

Boundary Detection

Multidimensional Scaling Boundary Recognition

- 2nd step: approximate real distances by hop-distances

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$



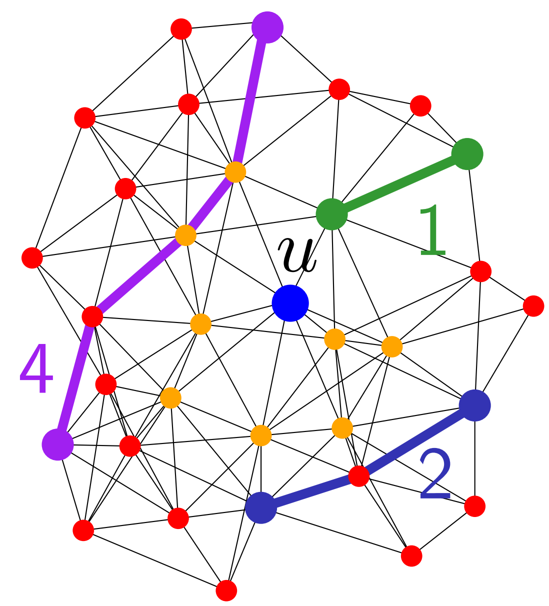
Hop-distance matrix of 2-hop neighborhood

Boundary Detection

Multidimensional Scaling Boundary Recognition

- 2nd step: approximate real distances by hop-distances

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$



Hop-distance matrix of 2-hop neighborhood

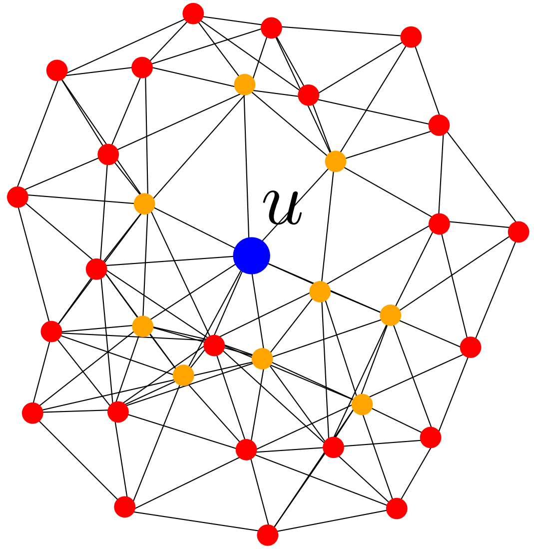
Boundary Detection

Multidimensional Scaling Boundary Recognition

- 3rd step: use MDS to compute embedding from hop-distances

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \Longrightarrow$$

Hop-distance matrix of 2-hop neighborhood



Estimated relative node positions

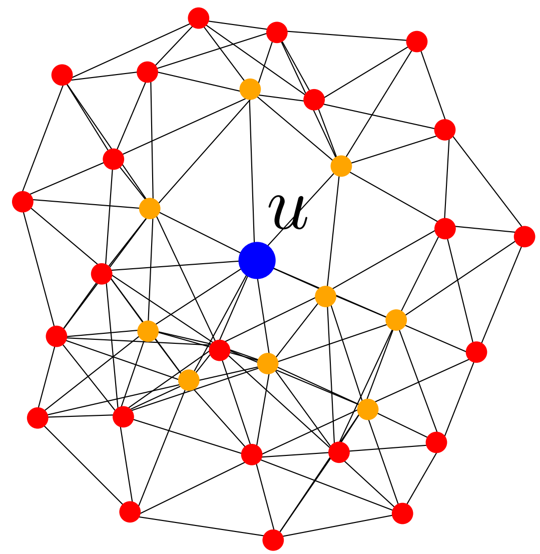
Boundary Detection

Multidimensional Scaling Boundary Recognition

- 3rd step: use MDS to compute embedding from hop-distances

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \xRightarrow{\text{MDS}}$$

Hop-distance matrix of 2-hop neighborhood



Estimated relative node positions

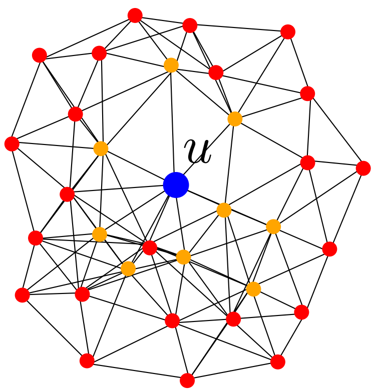
Boundary Detection

Multidimensional Scaling Boundary Recognition

- Multidimensional Scaling (MDS)

$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad \xRightarrow{\text{MDS}}$$

Pairwise distances δ_{ij}



Embedding
$\{x_1, ..., x_n\}$

$\sum_{i \neq j} (||x_i - x_j|| - \delta_{ij})^2$ minimal

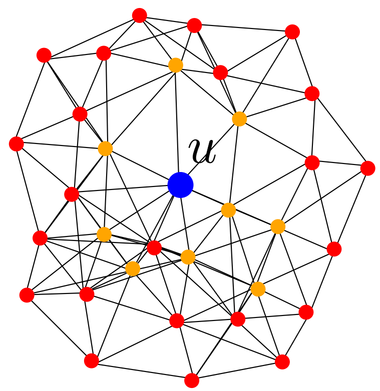
Boundary Detection

Multidimensional Scaling Boundary Recognition

- Multidimensional Scaling (MDS)

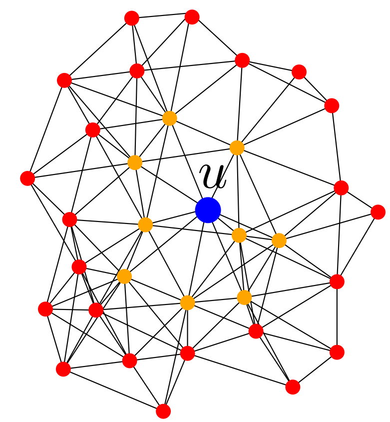
$$\begin{pmatrix} 0 & 1 & 1 & 2 & 3 & \cdots \\ 1 & 0 & 3 & 1 & 2 & \cdots \\ 1 & 3 & 0 & 2 & 1 & \cdots \\ 2 & 1 & 2 & 0 & 2 & \cdots \\ 3 & 2 & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \overset{\text{MDS}}{\Longrightarrow}$$

Pairwise distances δ_{ij}



Embedding $\{x_1, \ldots, x_n\}$

$\neq$



Real positions

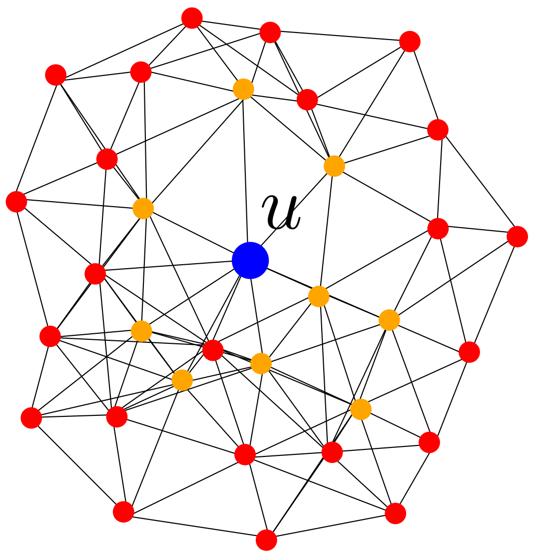
$$\sum_{i \neq j} (||x_i - x_j|| - \delta_{ij})^2 \text{ minimal}$$

hop distances

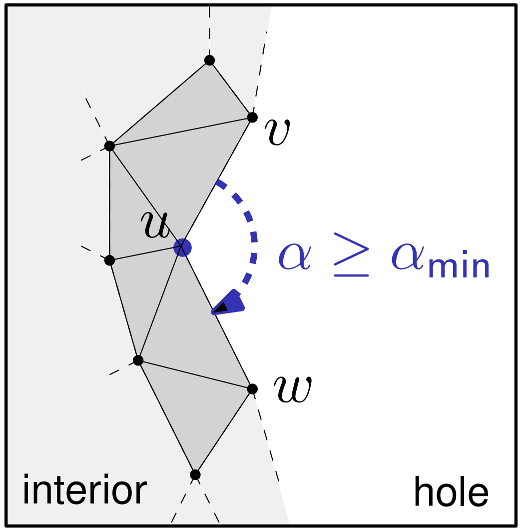
Boundary Detection

Multidimensional Scaling Boundary Recognition

- 4^{th} step: check angular conditions on computed embedding



Estimated relative node positions

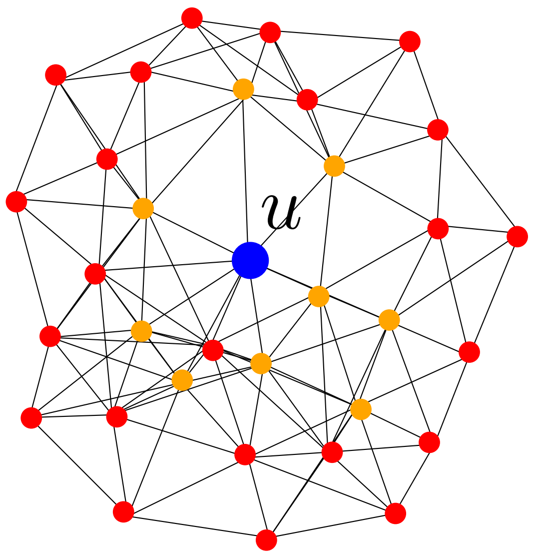


1^{st} condition:
$\alpha \geq \alpha_{\text{min}}$

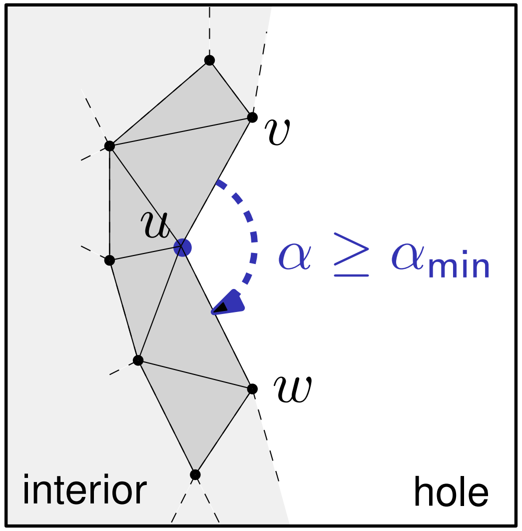
Boundary Detection

Multidimensional Scaling Boundary Recognition

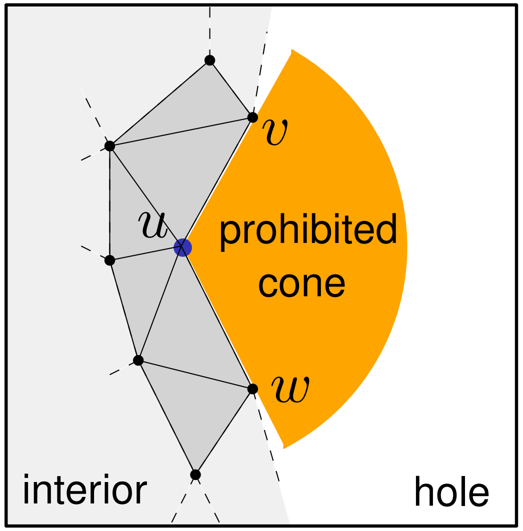
- 4th step: check angular conditions on computed embedding



Estimated relative node positions



1st condition: $\alpha \geq \alpha_{\mathsf{min}}$

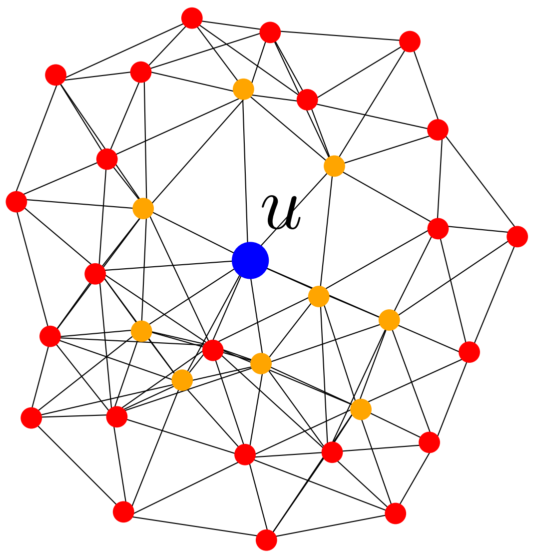


2nd condition: prohibited cone is empty

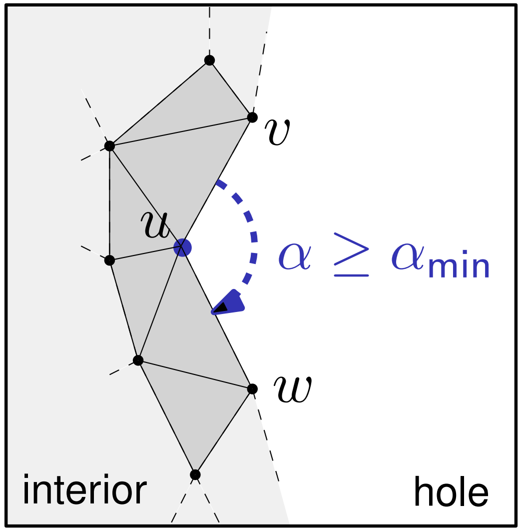
Boundary Detection

Multidimensional Scaling Boundary Recognition

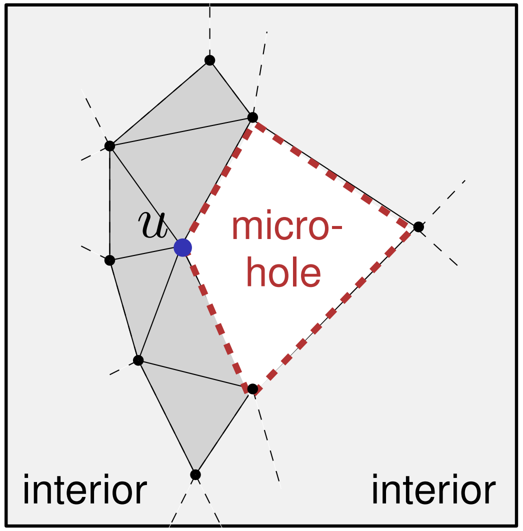
- 4^{th} step: check angular conditions on computed embedding



Estimated relative node positions



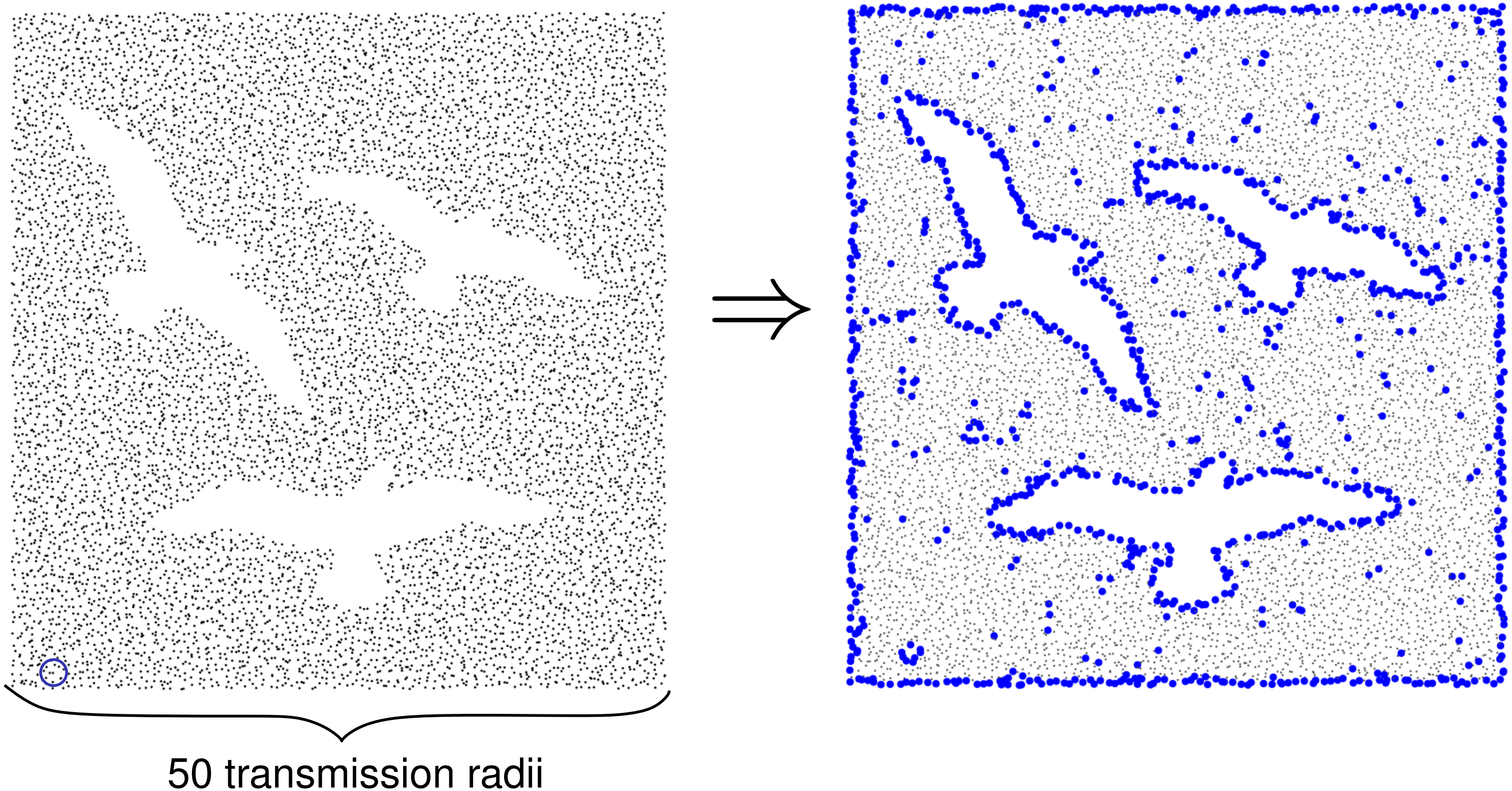
1^{st} condition:
$\alpha \geq \alpha_{\text{min}}$



2^{nd} condition: prohibited cone is empty

Multidimensional Scaling Boundary Recognition

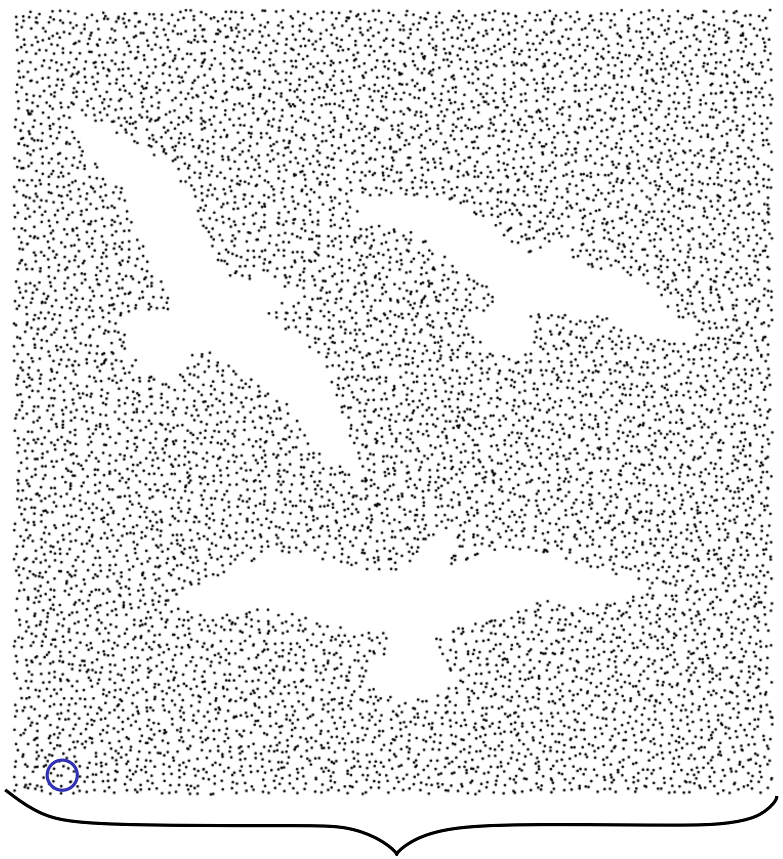
- Example



Boundary Detection

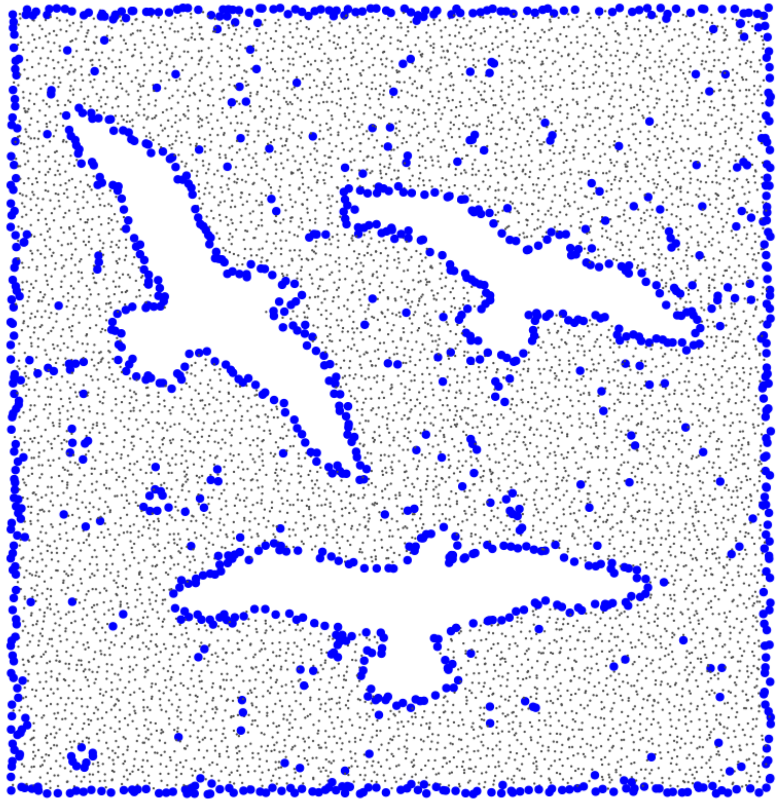
Multidimensional Scaling Boundary Recognition

- Example



50 transmission radii

$\Rightarrow$

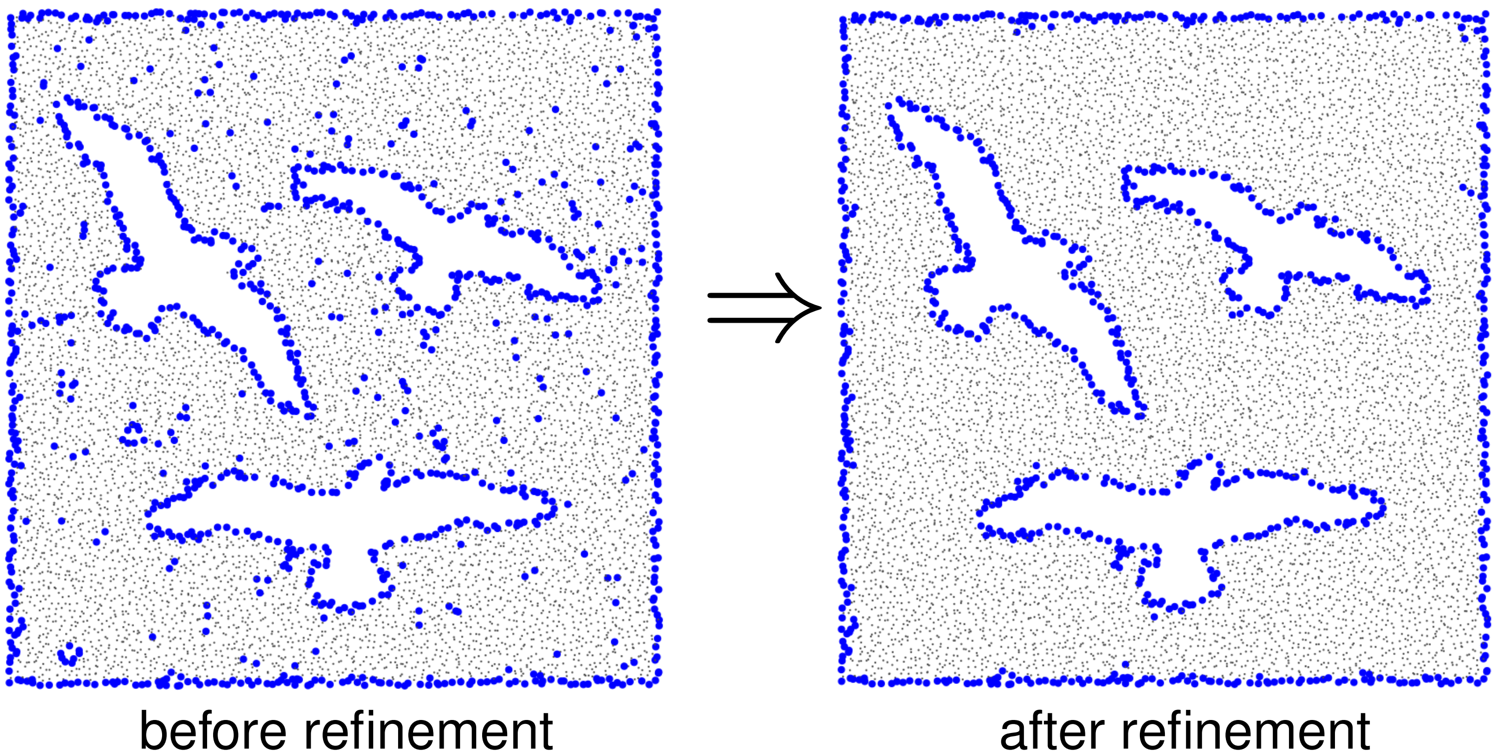


noisy $\Rightarrow$ filtering step

Boundary Detection

Multidimensional Scaling Boundary Recognition

- 5^{th} step: Refinement

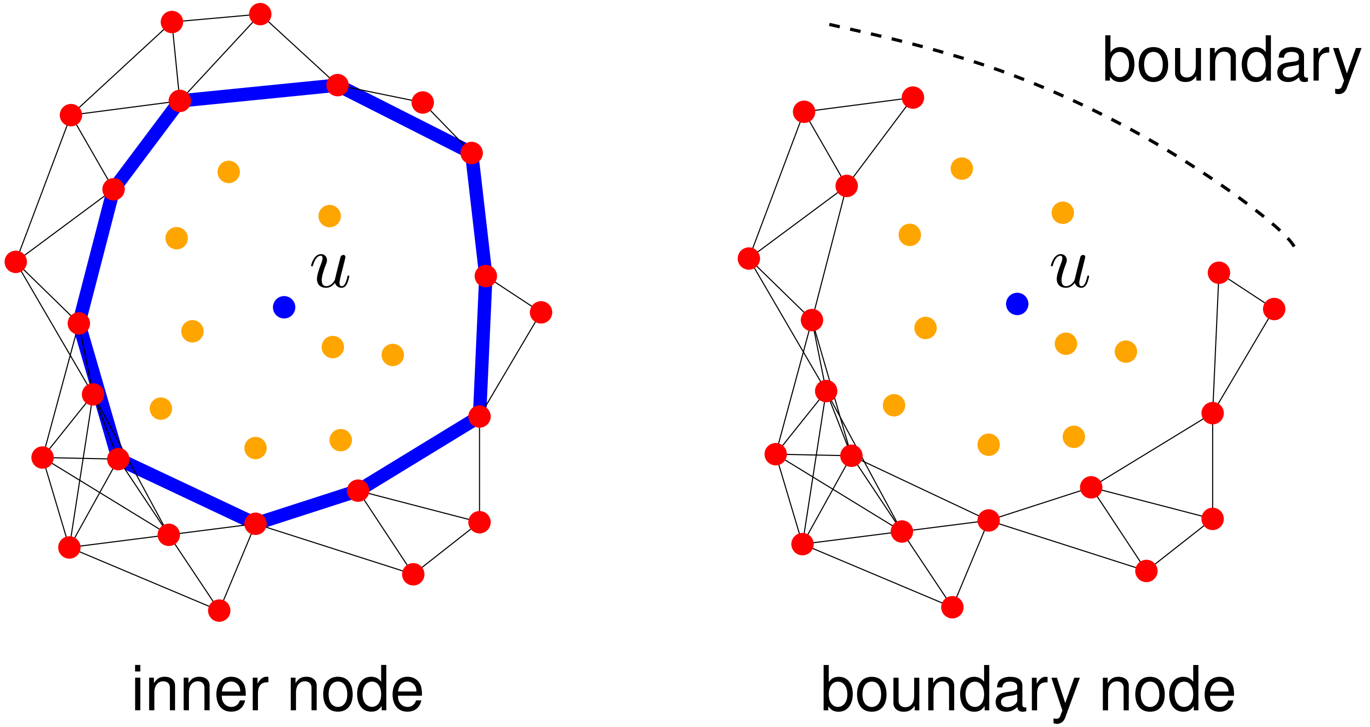


reclassify isolated nodes

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

Basic Idea

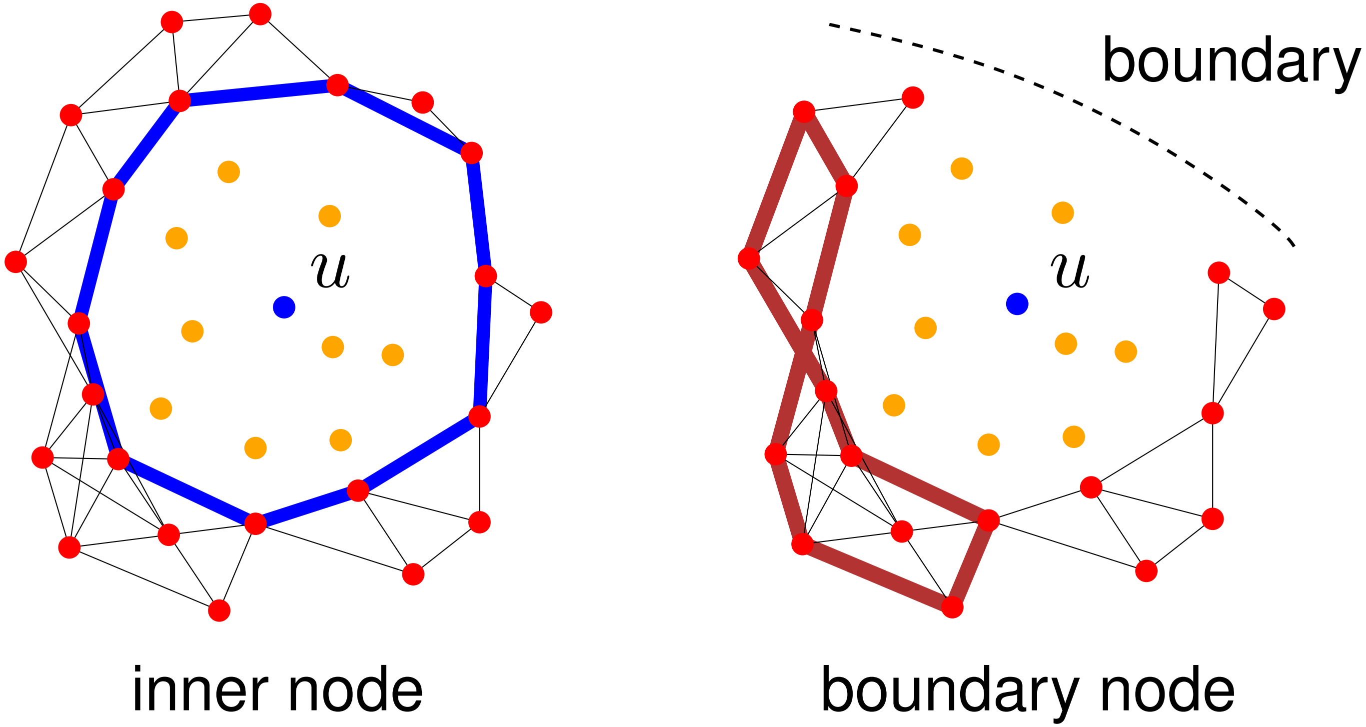


Node checks wheter it is surrounded by a closed circle

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

Basic Idea

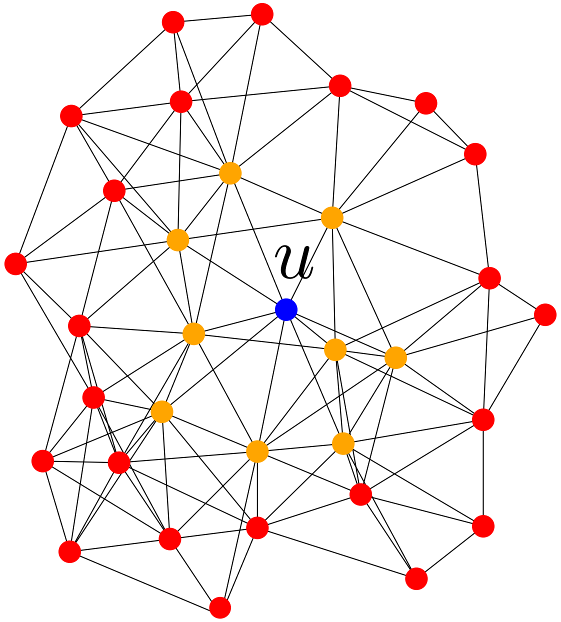


Problem: how to detect enclosing circles?

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 1^{st} step: collect information about 2-hop neighborhood

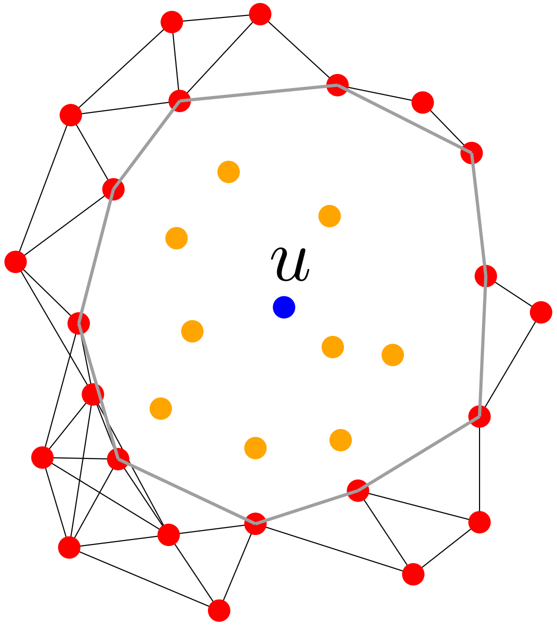


(2 messages per node)

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

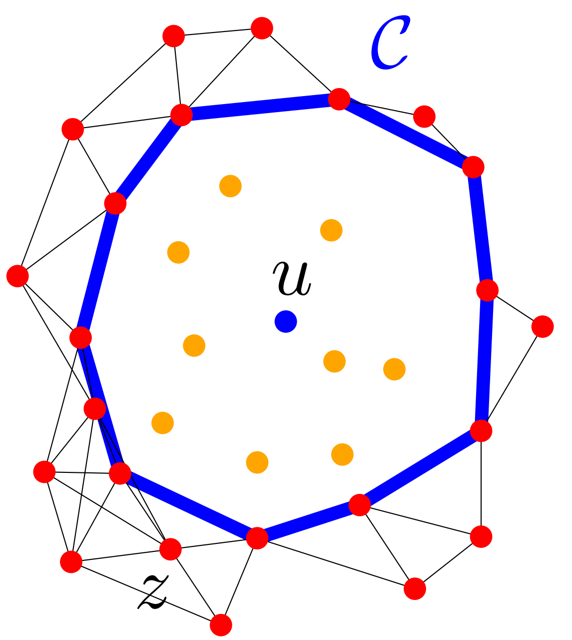
- 2[nd] step: remove 1-hop neighborhood



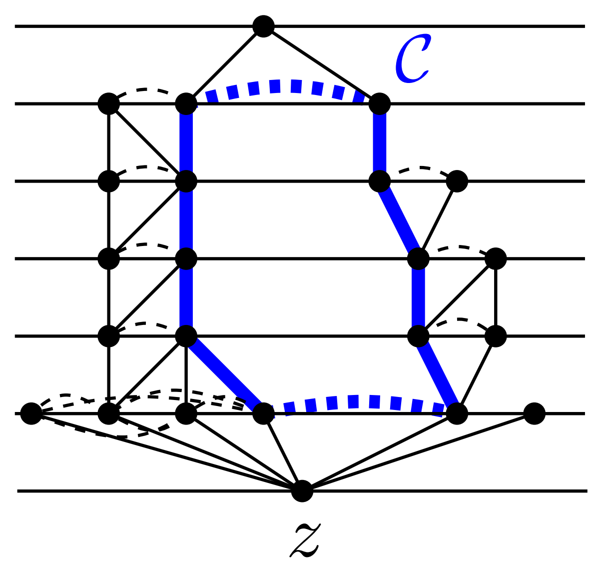
Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 3rd step: circle detection



Find "tight" cycle of maximum length

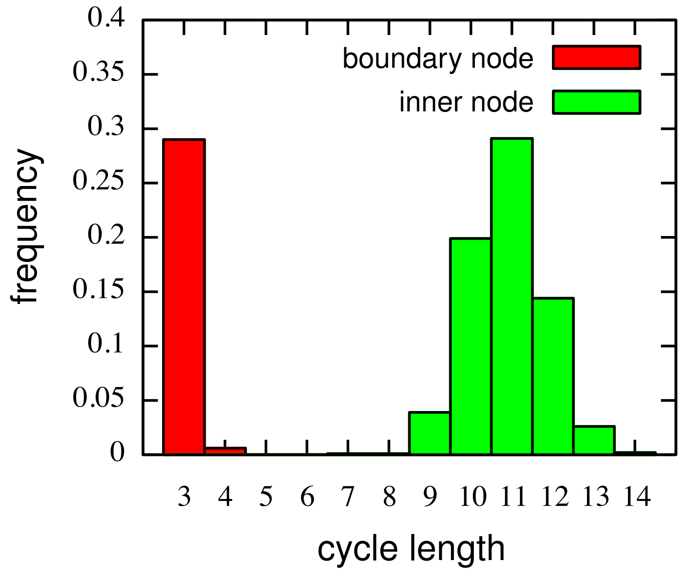


modified breadth-first search
(in $\mathcal{O}(|E|)$ time)

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 4^{th} step: classification according to length of longest cycle found

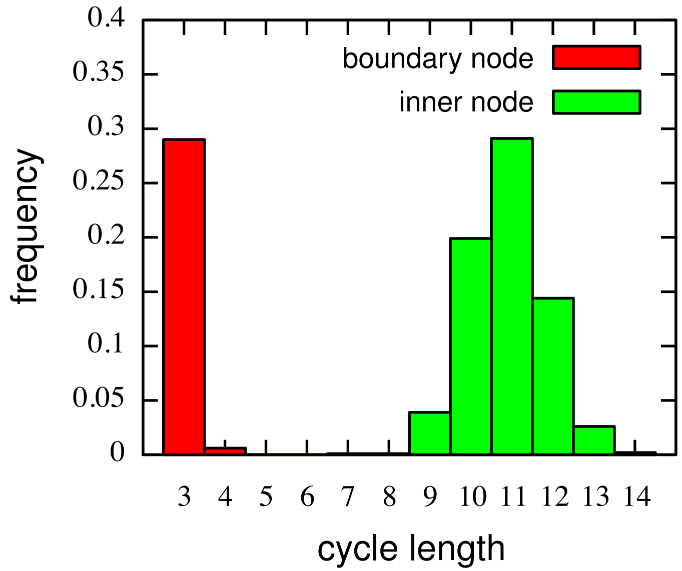


Unit Disk Graph Model

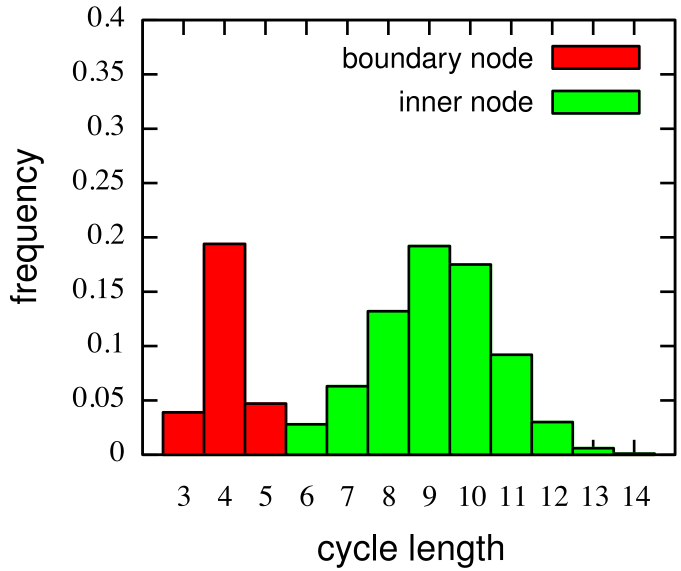
Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 4th step: classification according to length of longest cycle found



Unit Disk Graph Model

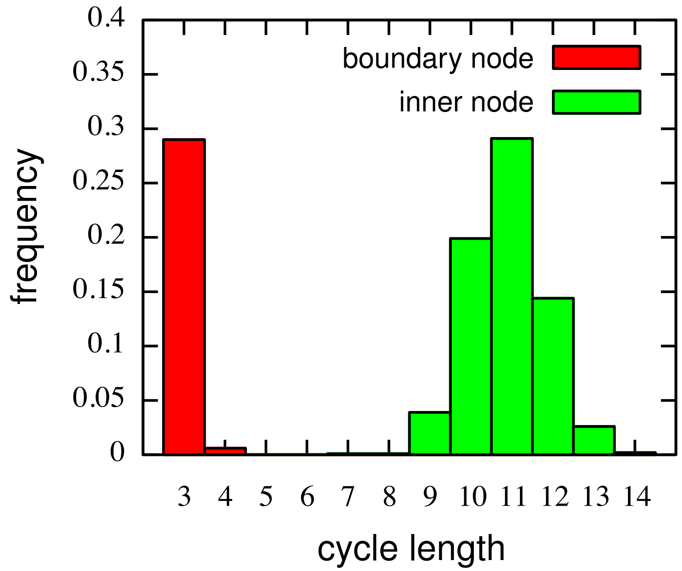


Quasi-UDG model

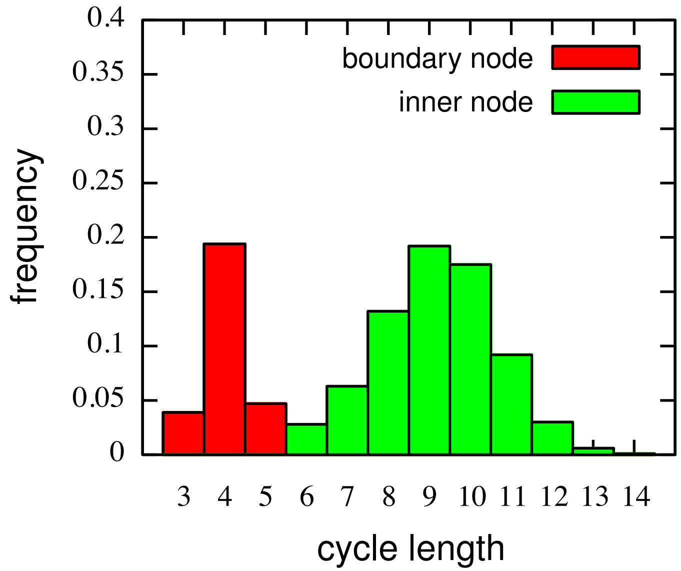
Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 4th step: classification according to length of longest cycle found



Unit Disk Graph Model



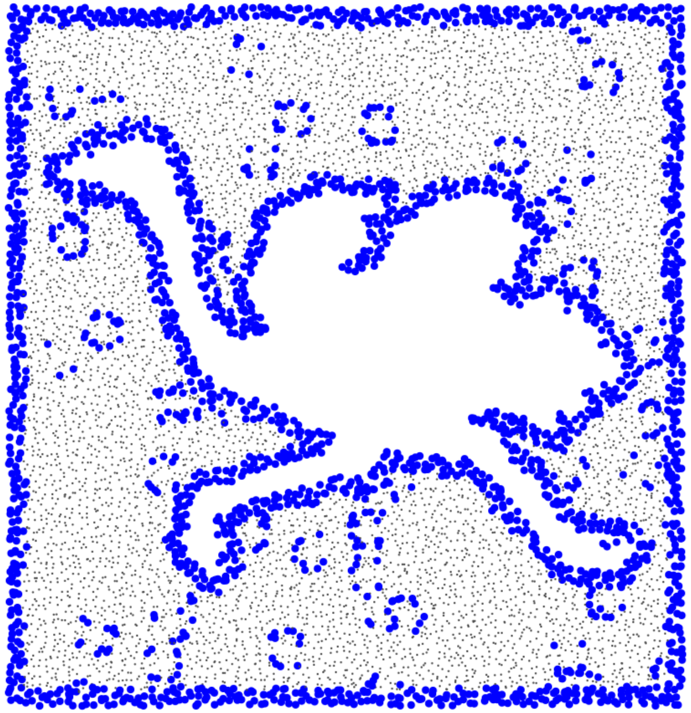
Quasi-UDG model

Almost independent of node degree!

Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

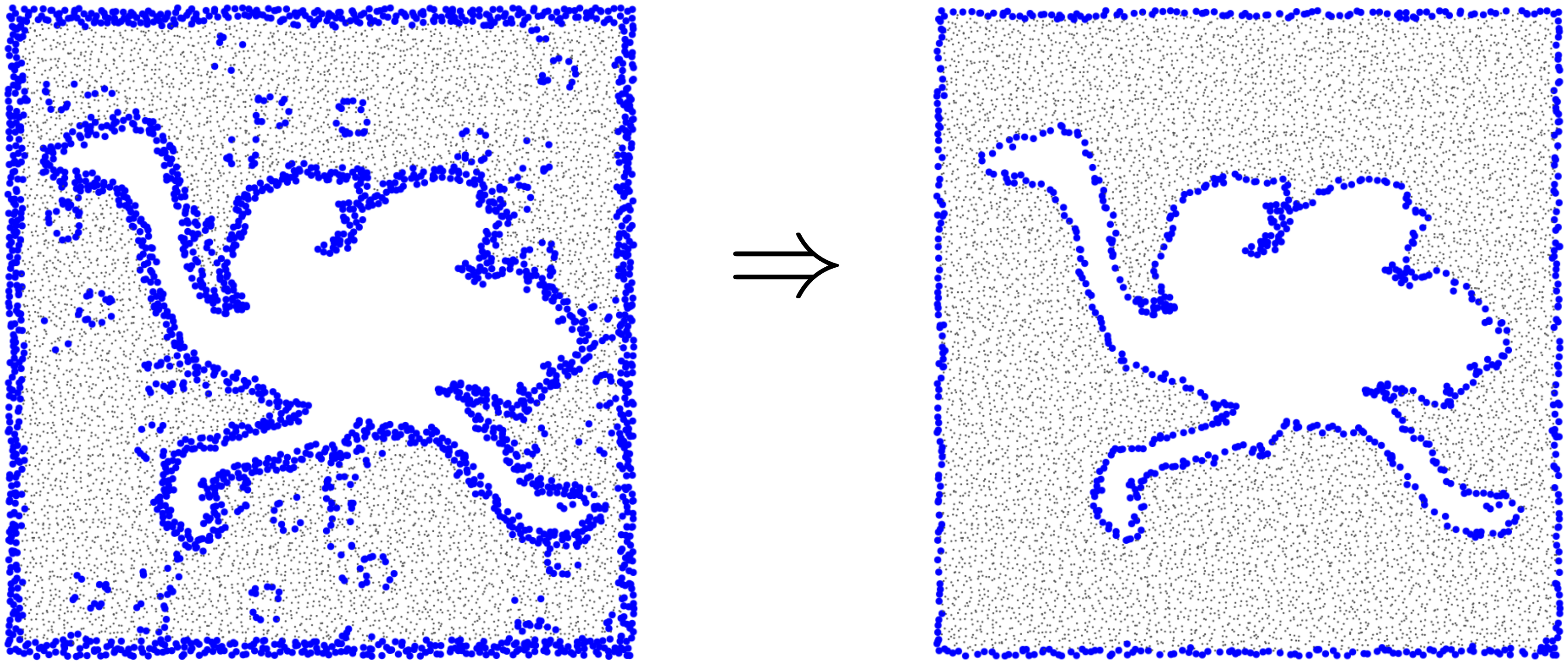
- 4^{th} step: classification according to length of longest cycle found



Boundary Detection

Enclosing Circle Boundary Recognition (EC-BR)

- 5^{th} step: refinement (optional)



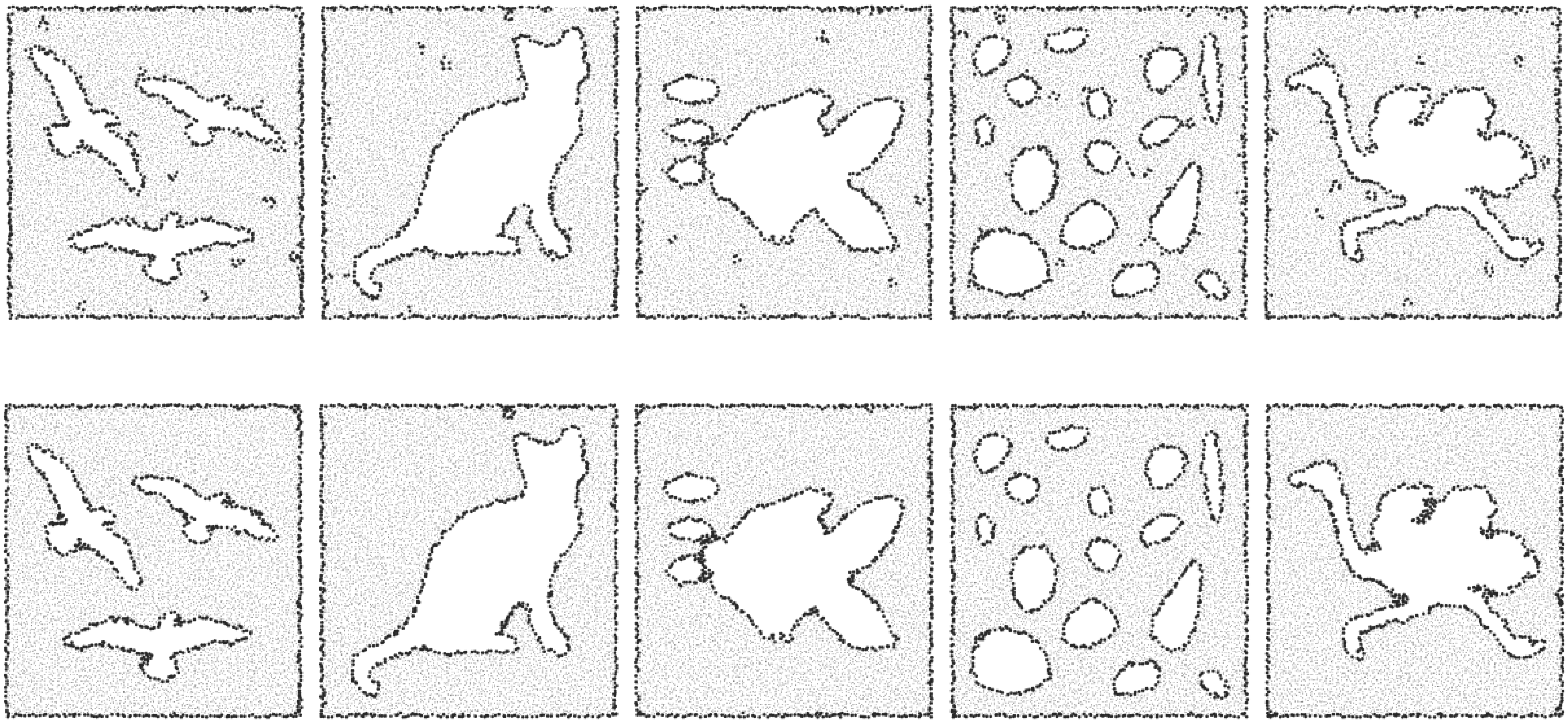
reclassify nodes which are not surrounded by other candidates

Boundary Detection

Some examples

MDS-BR

EC-BR

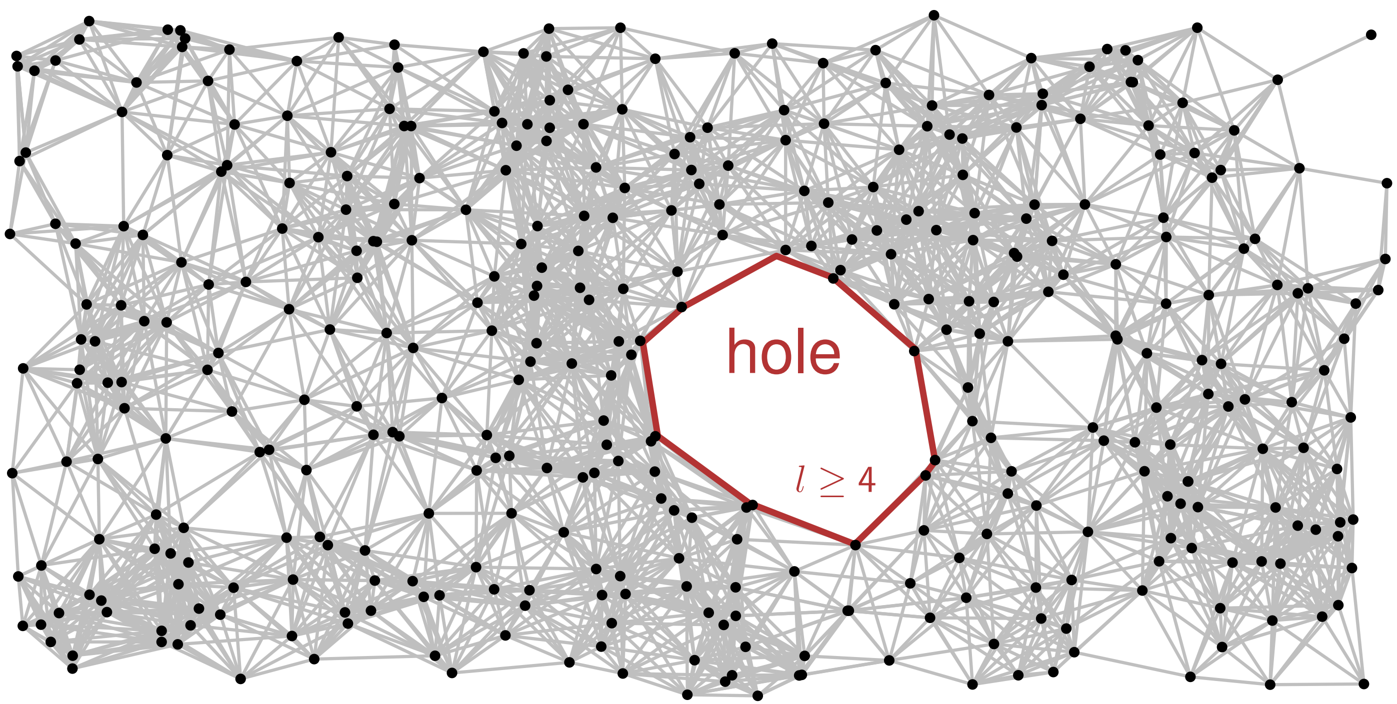


avg. node degree: 12 area: 50x50 transmission radii

Boundary Detection

Quantitative Evaluation

Hole Definition

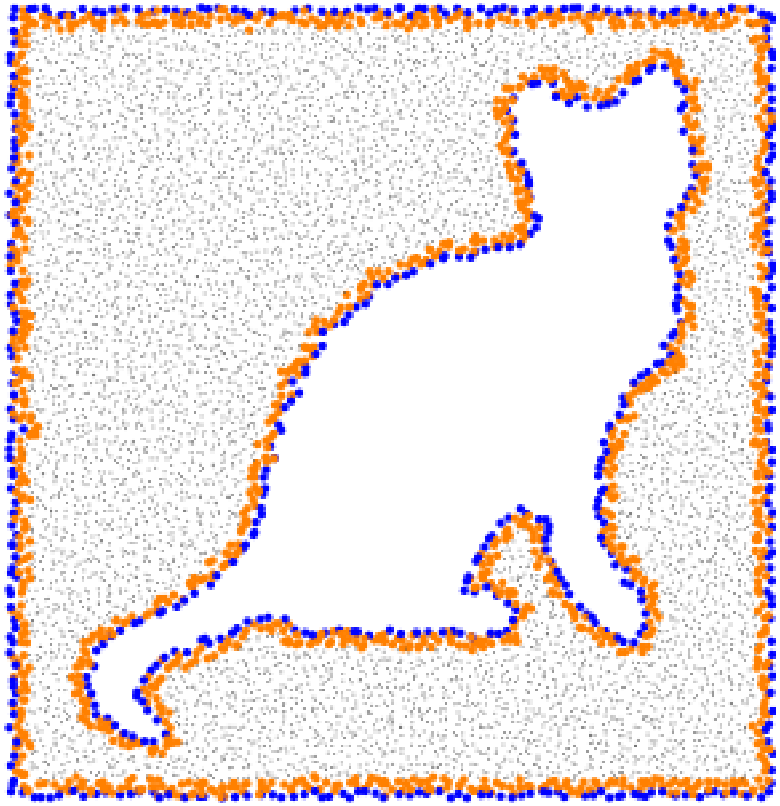


holes are faces of the connectivity graph
with circumference $l \geq 4$

Boundary Detection

Quantitative Evaluation

Hole Definition



- 3 categories
  - Mandatory boundary nodes
  - Optional boundary nodes
  - Inner nodes

Boundary Detection

Quantitative Evaluation

Performed Experiments

- network density (avg. degree 9, 12, 15, 18, 21)
- distribution methods (random vs. perturbed grid)
- communication models (UDG, QUDG)
- algorithm variants (e.g., inclusion of signal strength information)
- parameter analysis

Boundary Detection

Quantitative Evaluation

Included Algorithms

- [Fekete, Kröller, Pfisterer, Fischer, Buschmann, 2004]
- [Funke, 2005]
- [Funke, Klein, 2006]

- [Wang, Gao, Mitchell, 2006]
- [Saukh, Sauter, Gauger, Marrón, Rothermel, 2010]

In progress

Boundary Detection

Quantitative Evaluation

Example: Network Density

	Boundary Nodes					Inner Nodes				
	9	12	15	18	21	9	12	15	18	21
MDS-BR	**3.3**	3.0	3.7	4.1	4.2	17.3	0.3	0.1	0.1	**0.0**
EC-BR	4.4	**0.4**	**0.6**	**1.0**	**1.3**	**7.1**	**0.0**	**0.0**	**0.0**	**0.0**
Fekete 04	34.7	14.2	6.7	3.4	1.9	9.8	3.5	7.2	6.9	2.5
Funke 05	16.6	6.3	5.7	5.1	5.0	21.7	3.5	2.0	1.3	0.9
Funke 06	39.7	13.8	16.6	18.9	20.9	13.0	3.4	1.4	0.6	0.3

percentage of false negatives

Boundary Detection

Quantitative Evaluation

Example: Network Density

	Boundary Nodes					Inner Nodes				
	9	12	15	18	21	9	12	15	18	21
MDS-BR	**3.3**	3.0	3.7	4.1	4.2	17.3	0.3	0.1	0.1	**0.0**
EC-BR	4.4	**0.4**	**0.6**	**1.0**	**1.3**	**7.1**	**0.0**	**0.0**	**0.0**	**0.0**
Fekete 04	34.7	14.2	6.7	3.4	1.9	9.8	3.5	7.2	6.9	2.5
Funke 05	16.6	6.3	5.7	5.1	5.0	21.7	3.5	2.0	1.3	0.9
Funke 06	39.7	13.8	16.6	18.9	20.9	13.0	3.4	1.4	0.6	0.3

percentage of false negatives

Boundary Detection

Quantitative Evaluation

Example: Network Density

	Boundary Nodes					Inner Nodes				
	9	12	15	18	21	9	12	15	18	21
MDS-BR	**3.3**	3.0	3.7	4.1	4.2	17.3	0.3	0.1	0.1	**0.0**
EC-BR	4.4	**0.4**	**0.6**	**1.0**	**1.3**	**7.1**	**0.0**	**0.0**	**0.0**	**0.0**
Fekete 04	34.7	14.2	6.7	3.4	1.9	9.8	3.5	7.2	6.9	2.5
Funke 05	16.6	6.3	5.7	5.1	5.0	21.7	3.5	2.0	1.3	0.9
Funke 06	39.7	13.8	16.6	18.9	20.9	13.0	3.4	1.4	0.6	0.3

percentage of false negatives

all considered algorithms start having problems

for avg. degrees below 10

Boundary Detection

Quantitative Evaluation

Performed Experiments

- network density (avg. degree 9, 12, 15, 18, 21)
- distribution methods (random vs. perturbed grid)
- communication models (UDG, QUDG)
- algorithm variants (e.g., inclusion of signal strength information)
- parameter analysis

only EC-BR works well with QUDG

Boundary Detection

Quantitative Evaluation

Performed Experiments

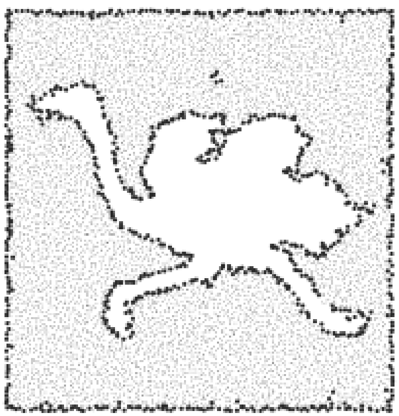
- network density (avg. degree 9, 12, 15, 18, 21)
- distribution methods (random vs. perturbed grid)
- communication models (UDG, QUDG)
- algorithm variants (e.g., inclusion of signal strength information)
- parameter analysis

only EC-BR works well with QUDG

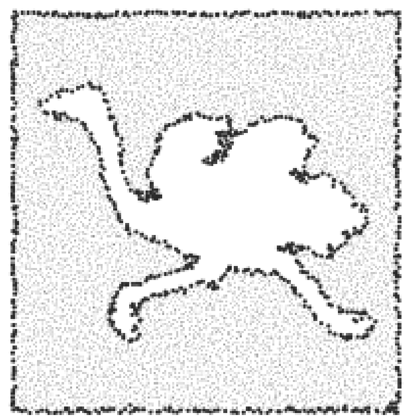
Now: visual comparison

Boundary Detection

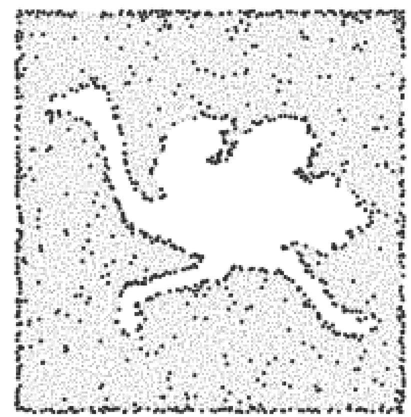
Comparison with existing approaches



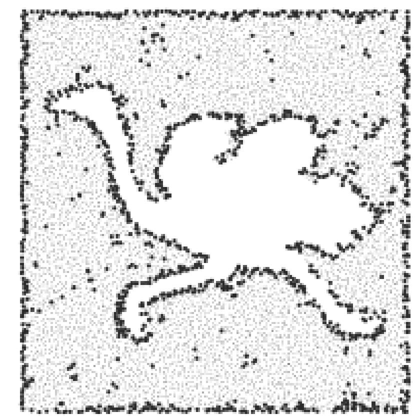
MDS-BR



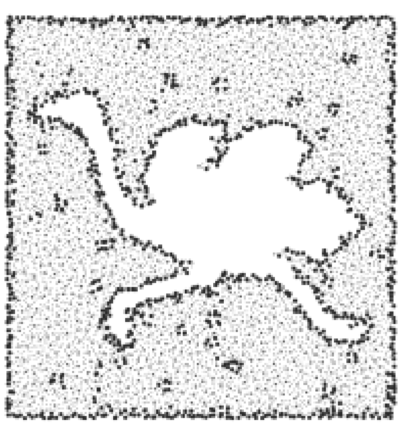
EC-BR



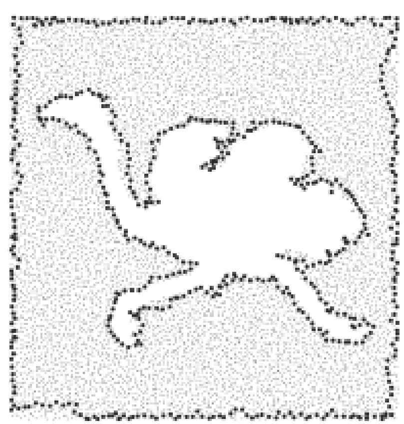
Fekete 04



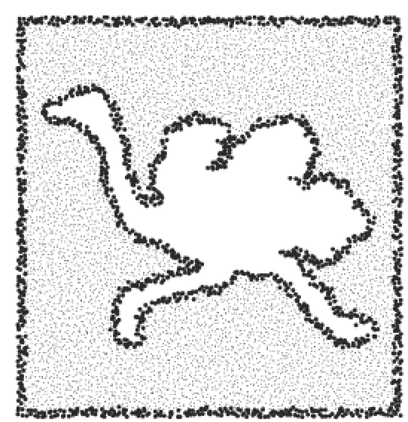
Funke 05



Funke 06



Wang 06

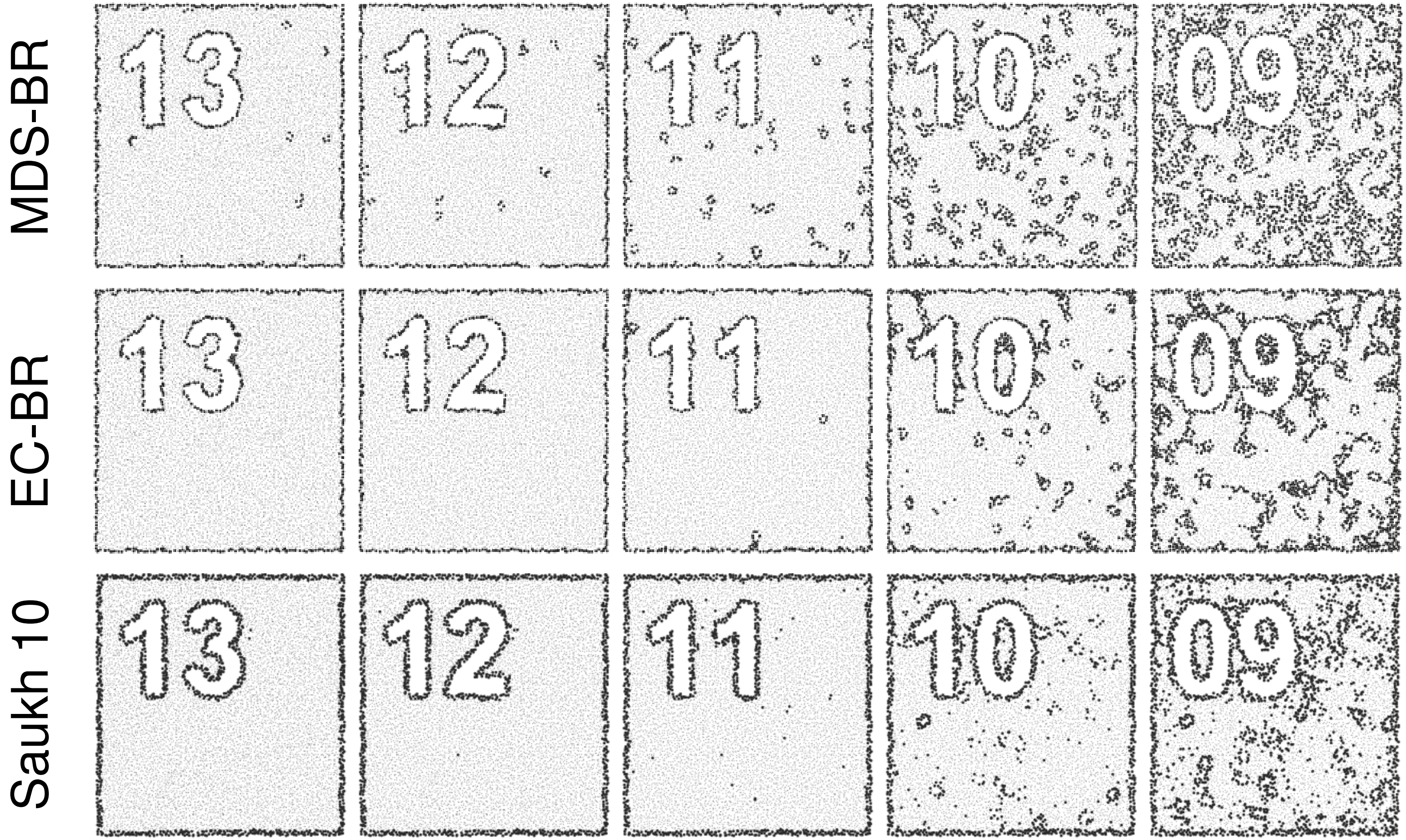


Saukh 10

(avg. node degree 12)

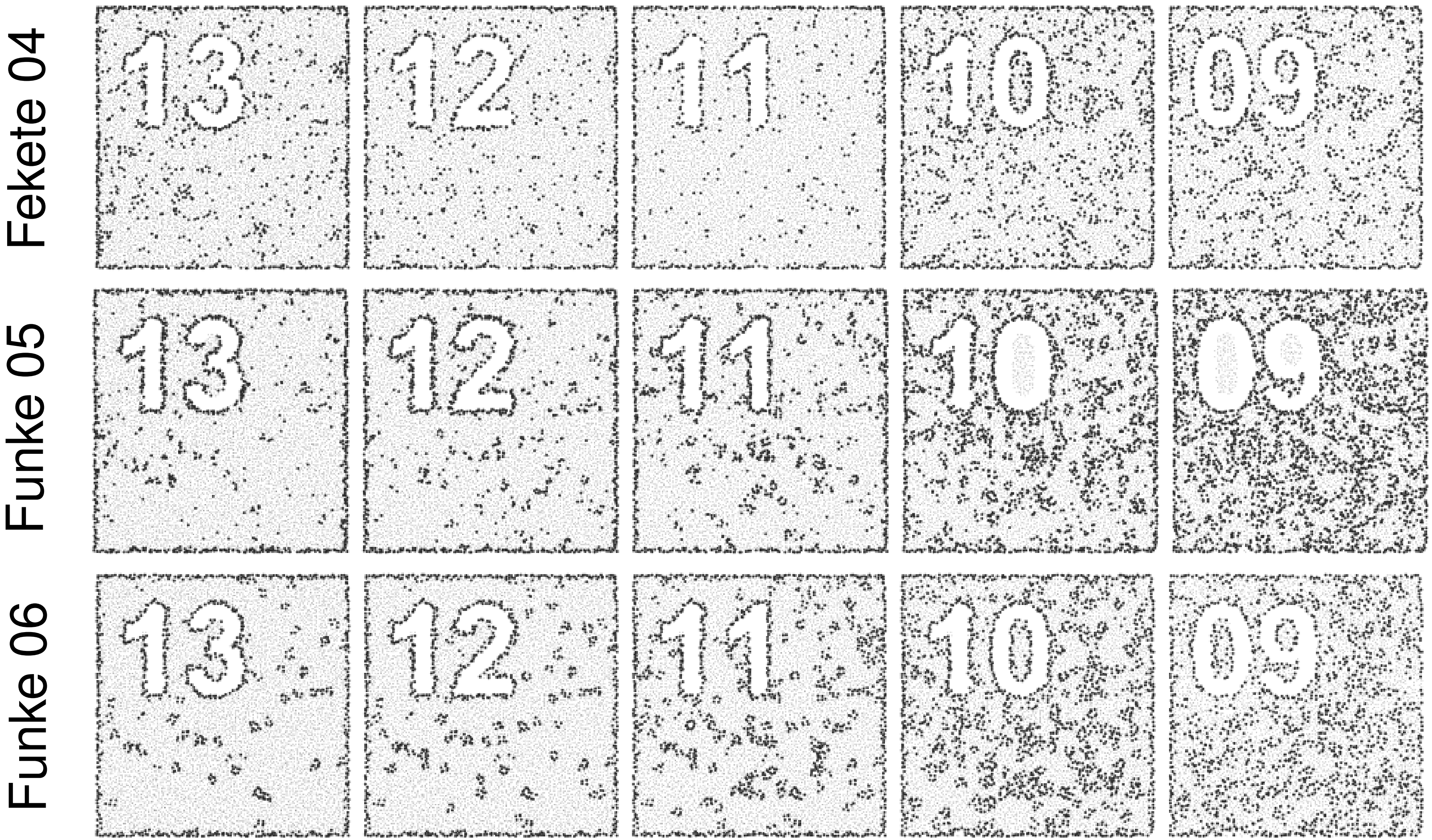
Boundary Detection

Effect of node degree



Boundary Detection

Effect of node degree

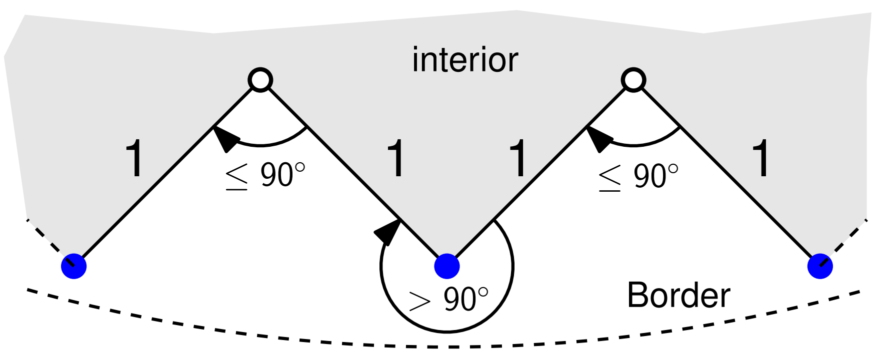


Boundary Detection

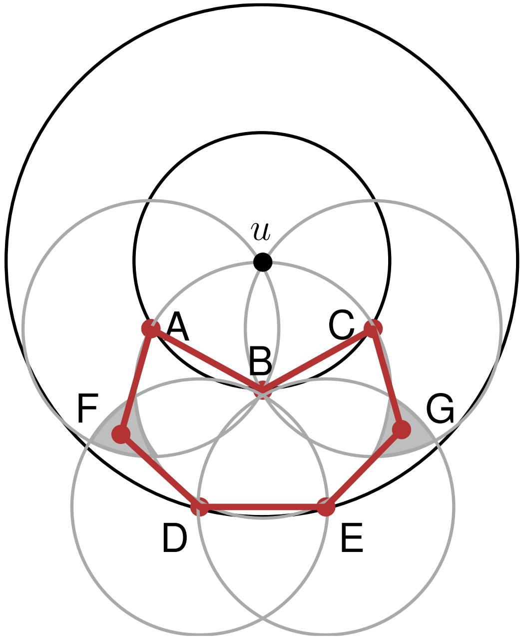
Classification Guarantees

- no guarantees for correct classification
- would require adjusted hole definition or larger comm. radius
- but one can proove that it can go wrong ;-)

MDS-BR



EC-BR



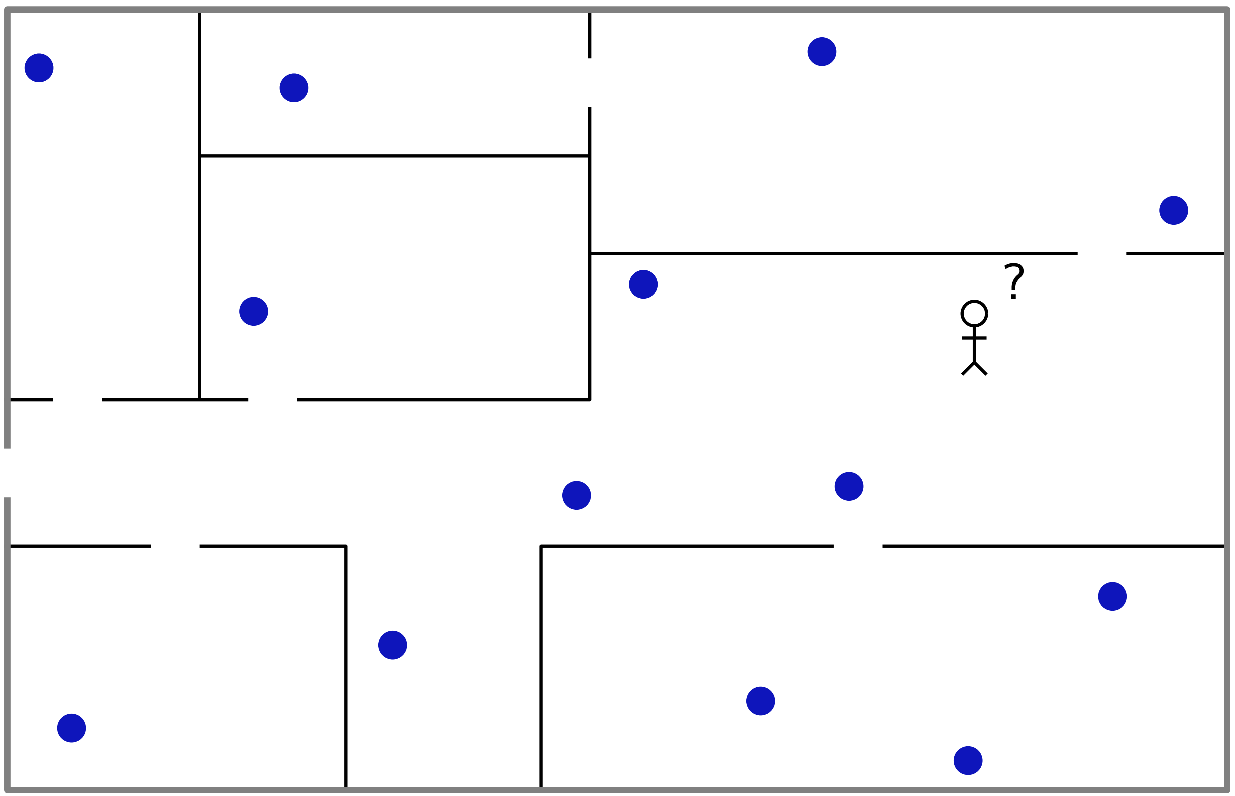
Happens only rarely for random networks

RSS Based Localization

(work in progress, only intermediate results)

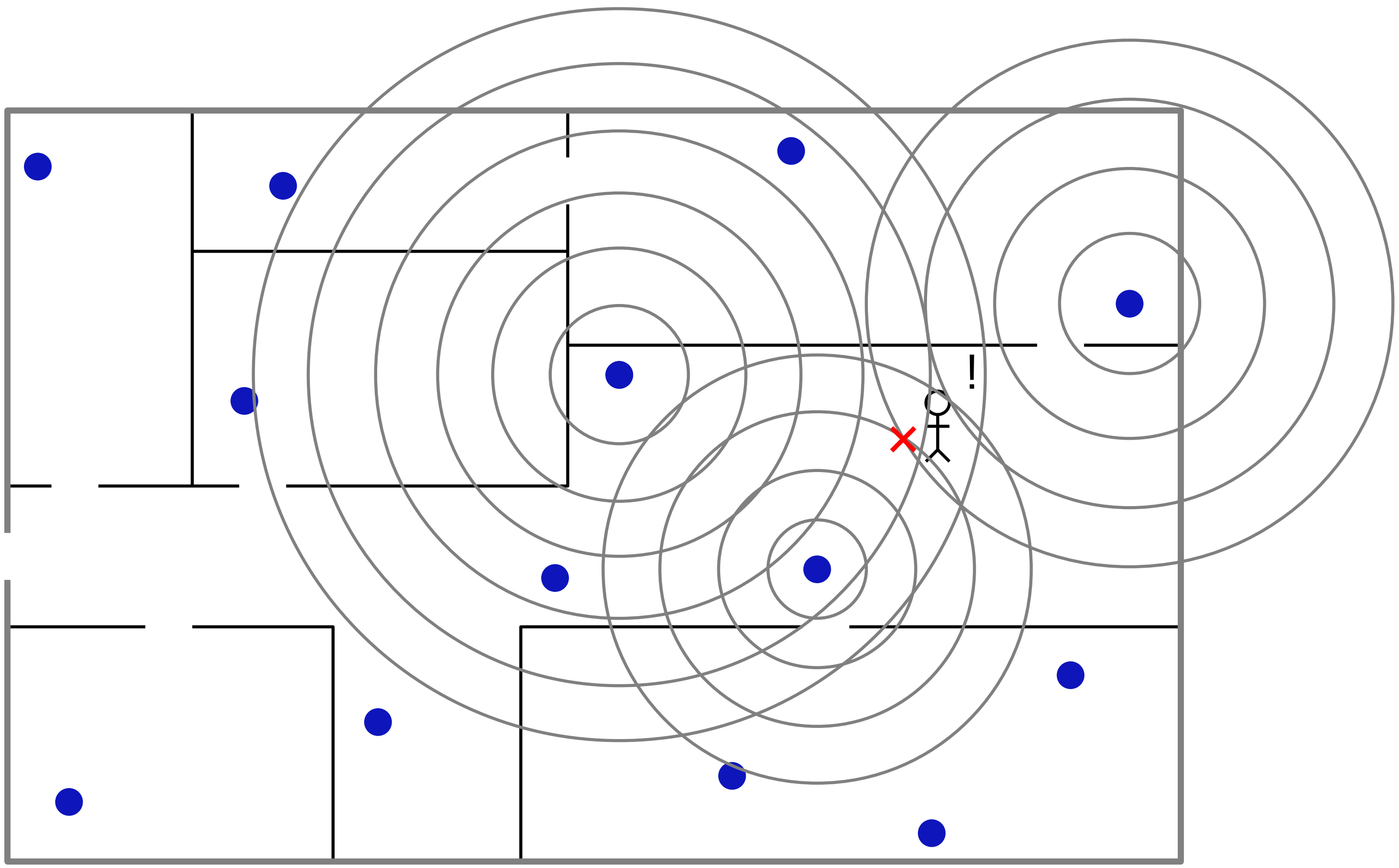
RSS Based Localization

Basic Idea



RSS Based Localization

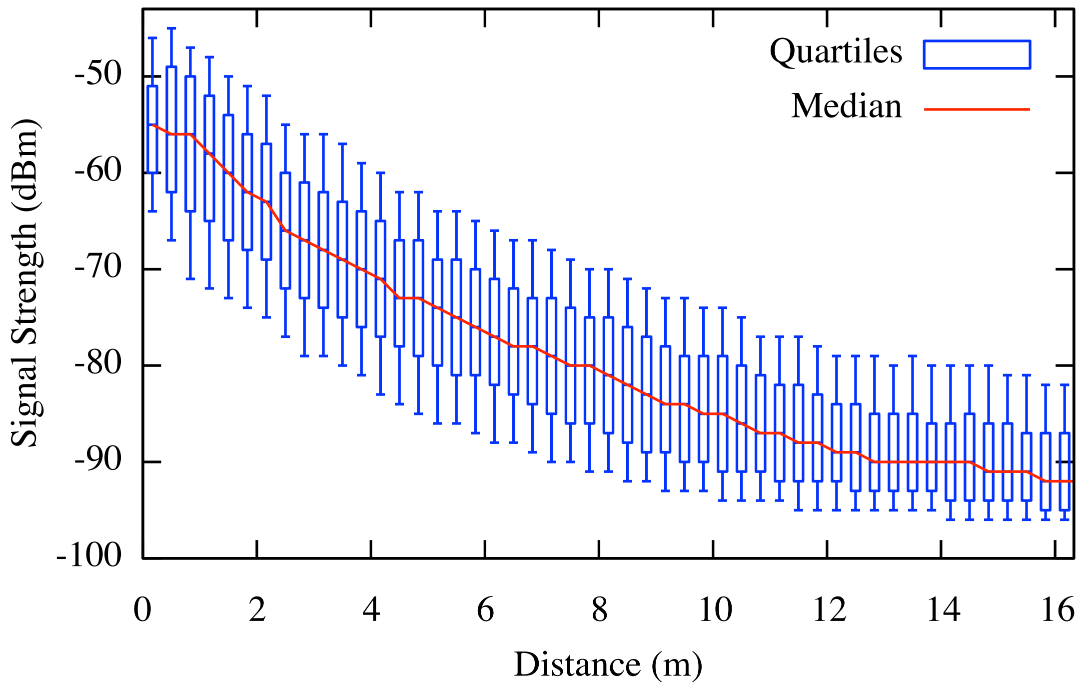
Basic Idea



RSS Based Localization

Background

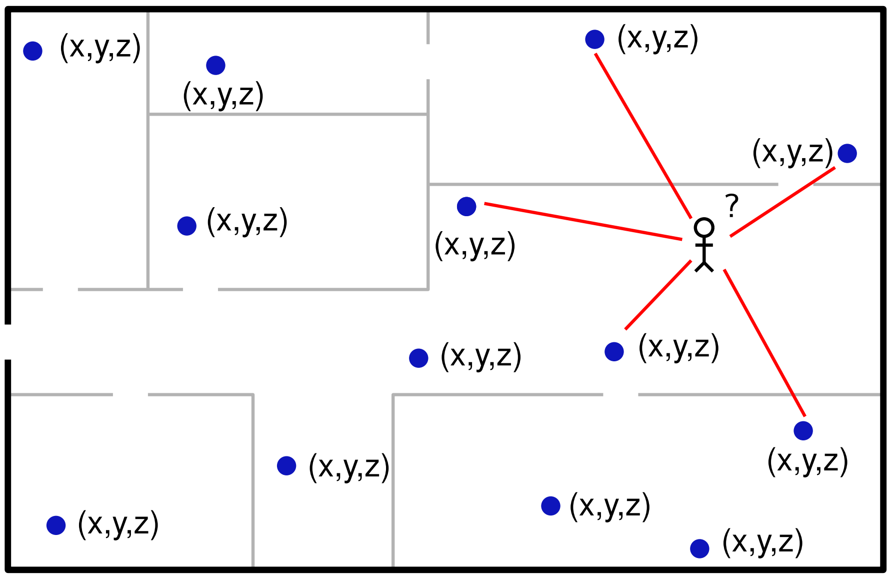
Distance dependance of signal strength



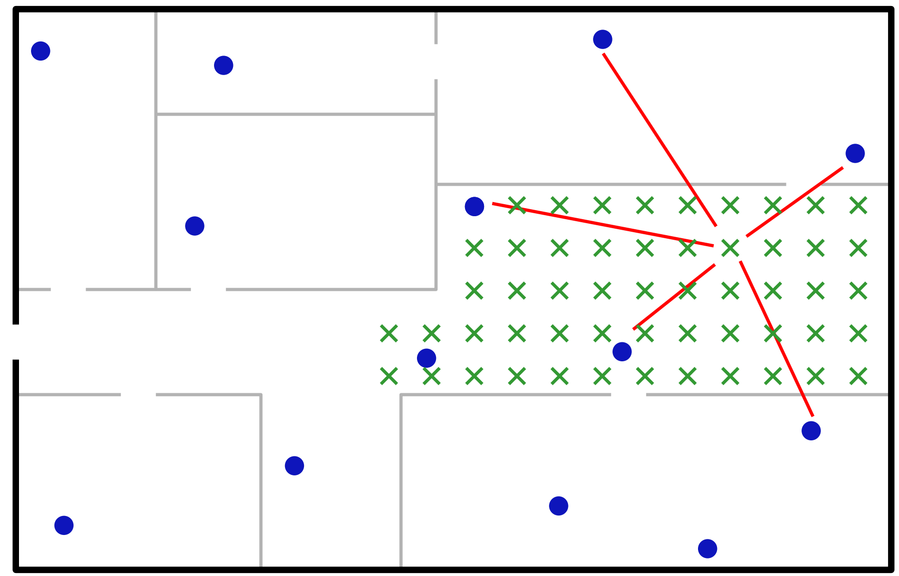
very noisy, but possible

RSS Based Localization

Usual Approaches



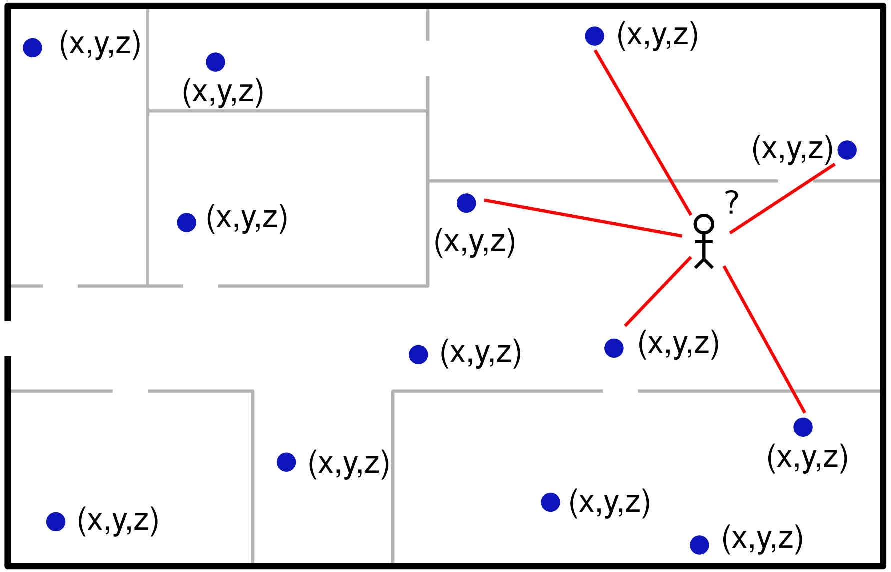
1st variant: *Trilateration, Kalman Filter*



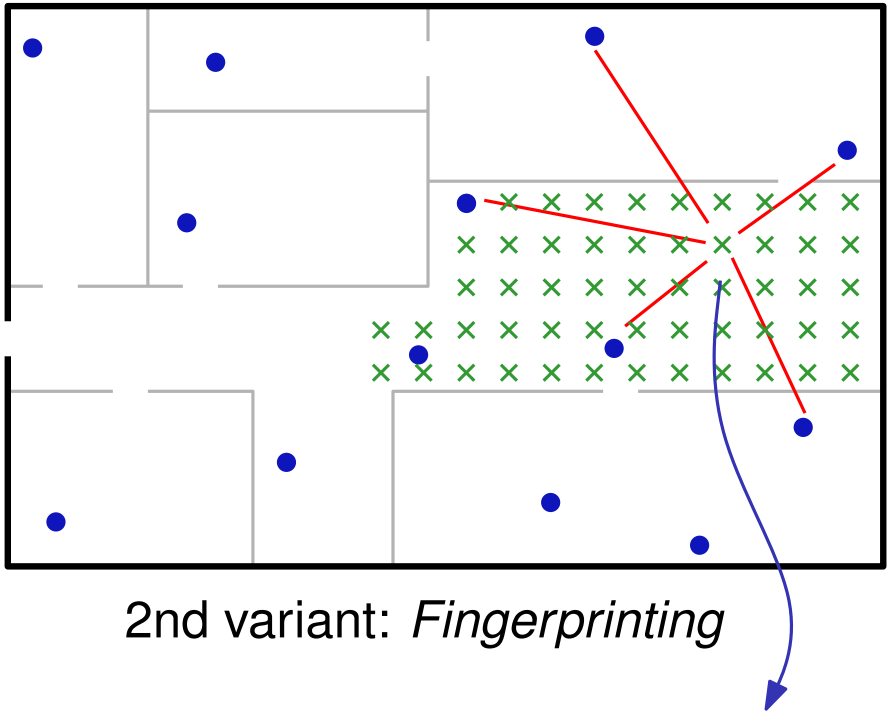
2nd variant: *Fingerprinting*

RSS Based Localization

Usual Approaches



1st variant: *Trilateration, Kalman Filter*

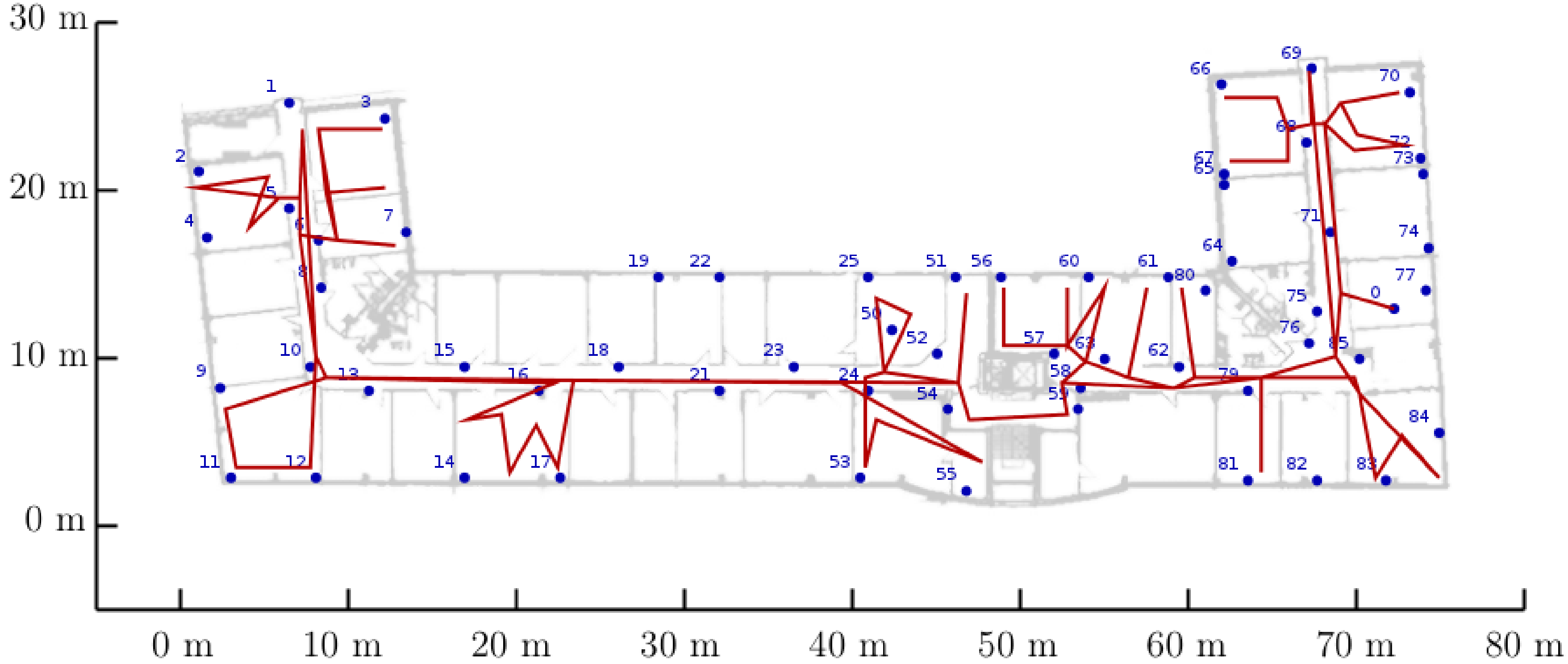


2nd variant: *Fingerprinting*

lookup table for RSS values

RSS Based Localization

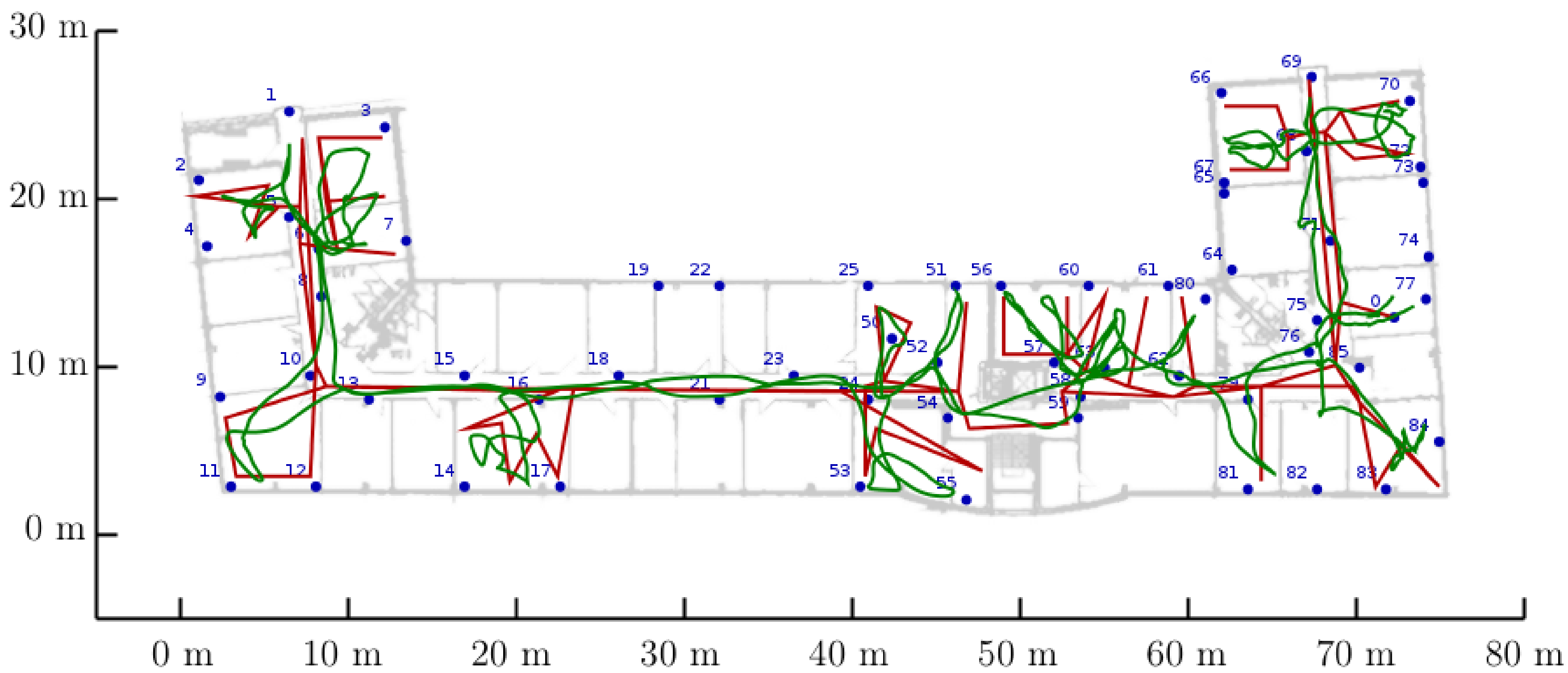
Real-World Examples



(network with 60 sensor nodes)

RSS Based Localization

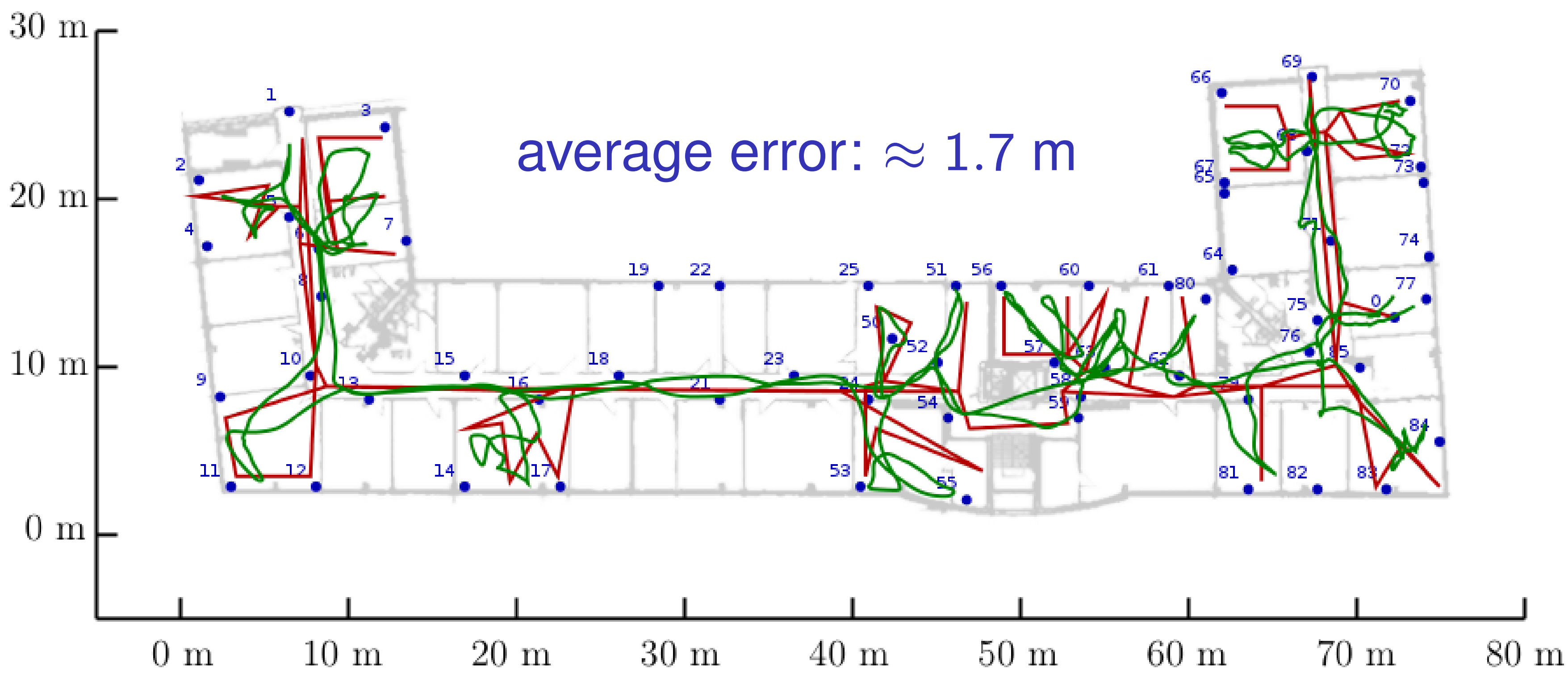
Real-World Examples



(network with 60 sensor nodes)

RSS Based Localization

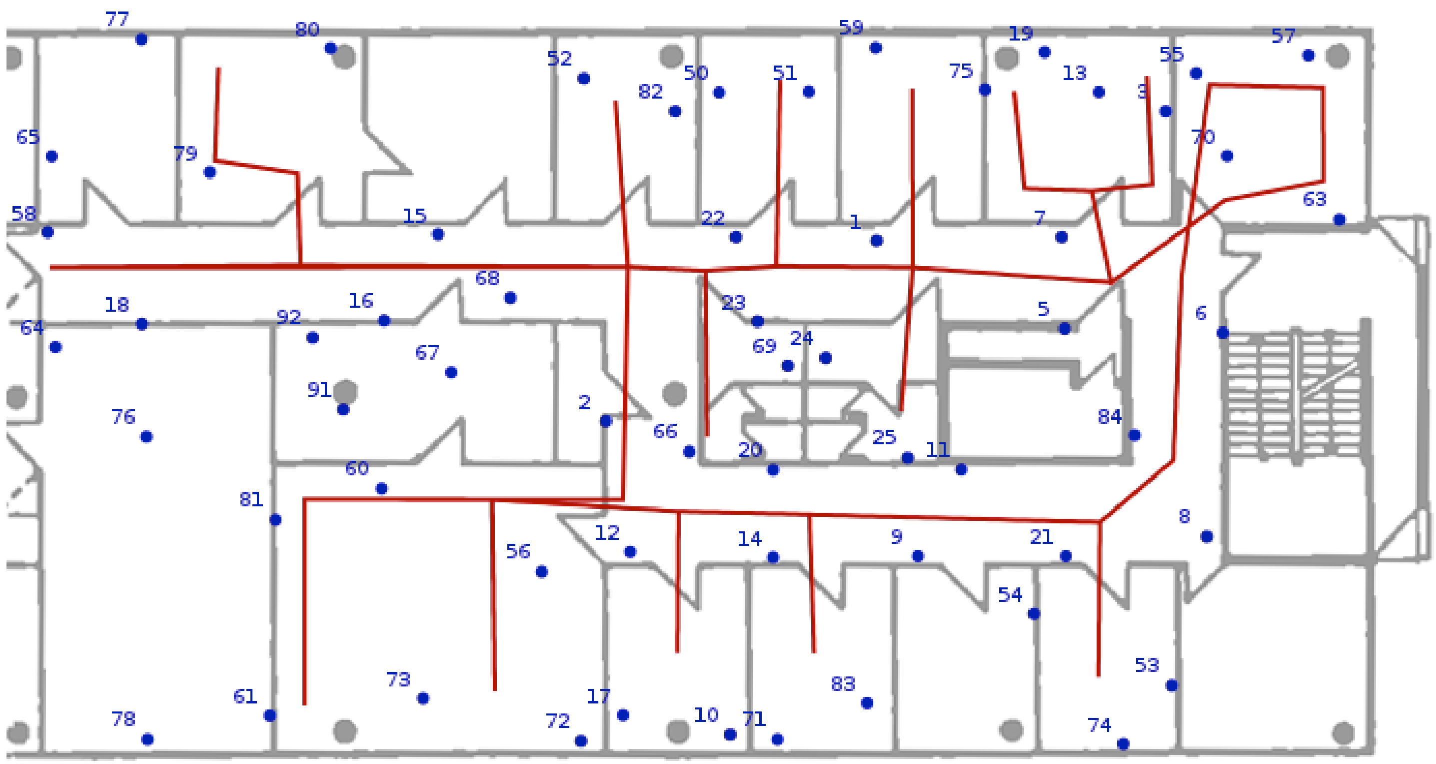
Real-World Examples



(network with 60 sensor nodes)

RSS Based Localization

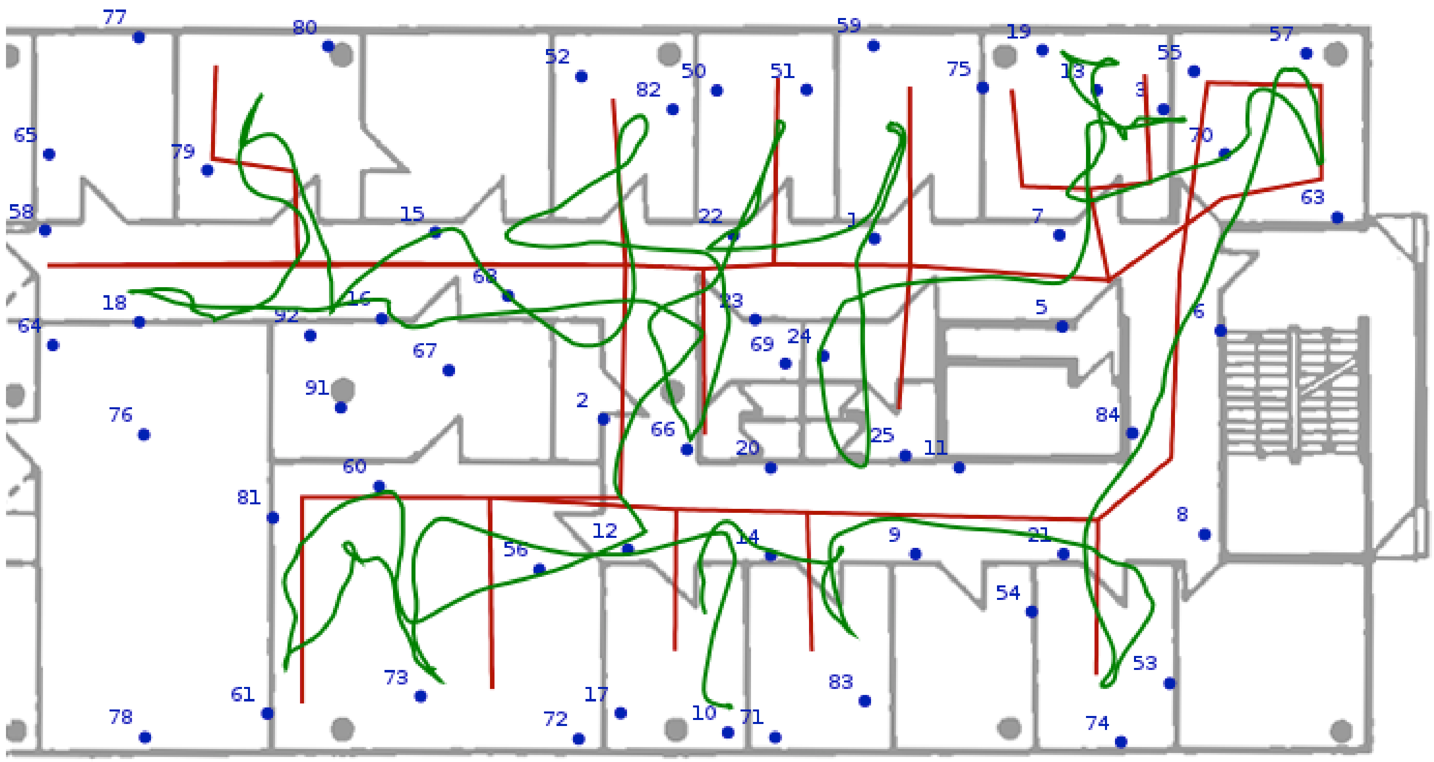
Real-World Examples



(network with 60 sensor nodes)

RSS Based Localization

Real-World Examples



(network with 60 sensor nodes)

RSS Based Localization

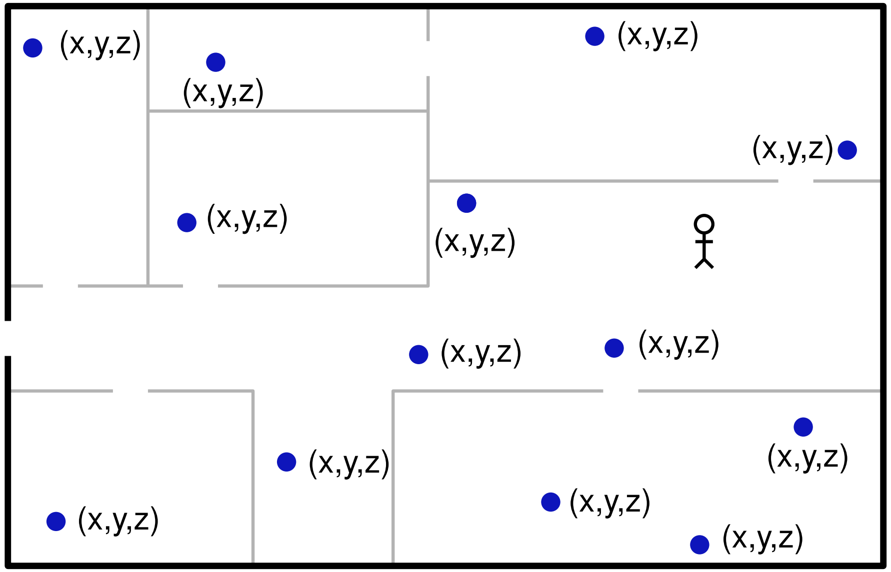
Real-World Examples

Used sensor nodes

- 65 sensor nodes (build by Johannes Schmid)
- Texas Instruments MSP430 low power MCU
- 2.4 GHz IEEE 802.15.4 compliant CC2520 radio chip
- 5.5 x 2.0 x 2.5 cm^3 casing

Sensor Network Localization

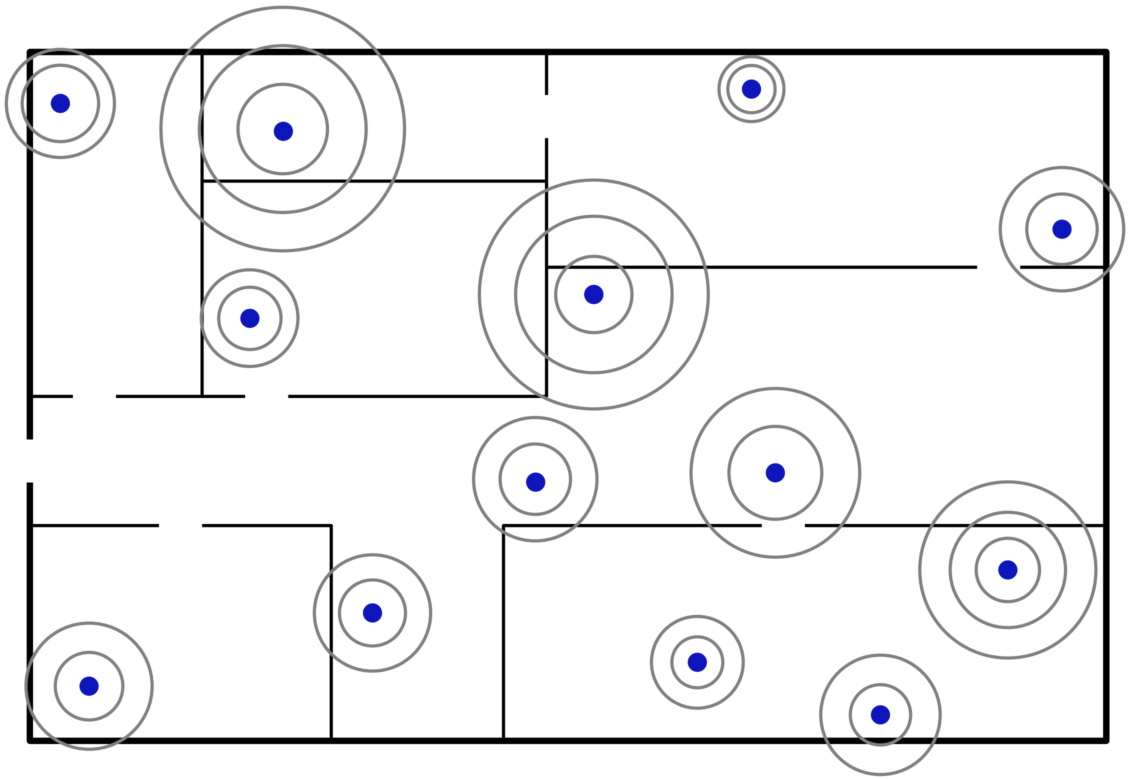
Considered Problem



Automatic Localization of Sensor Network?

Sensor Network Localization

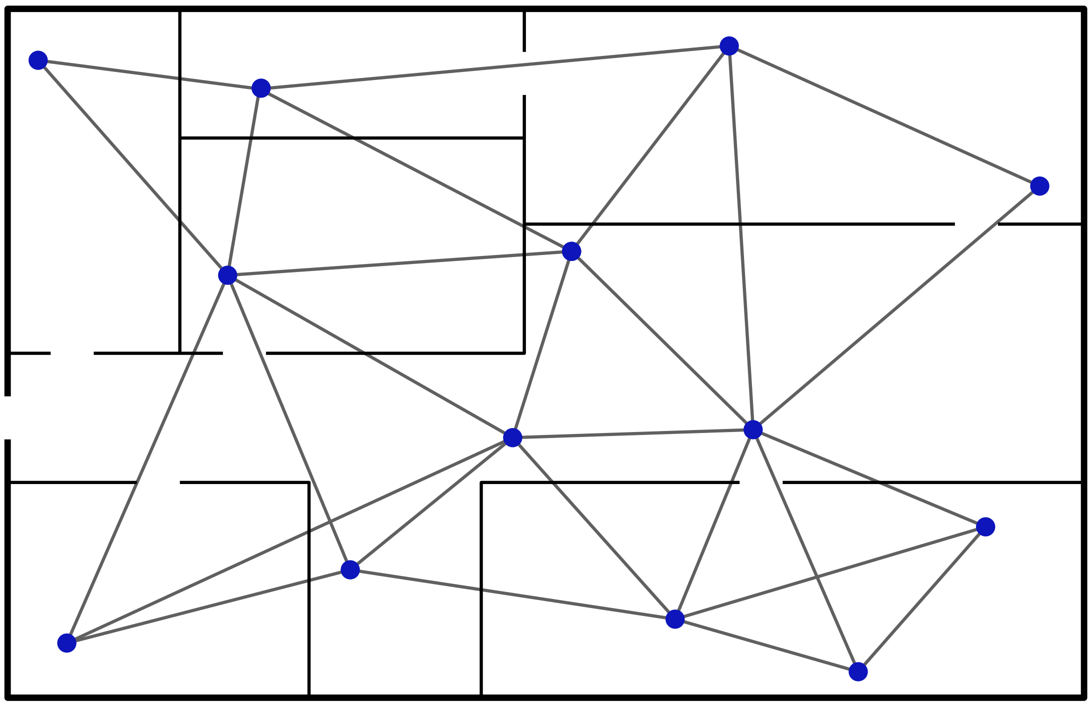
Existing Approach



Nodes communicate with neighbors

Sensor Network Localization

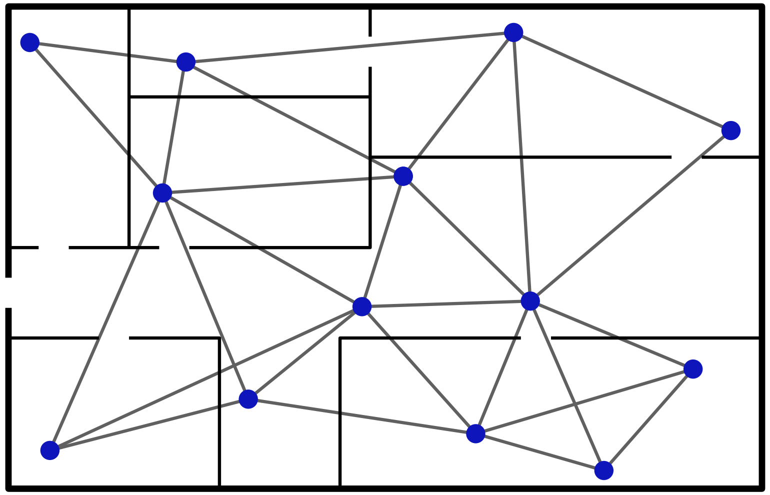
Existing Approach



Distance estimation from signal strength

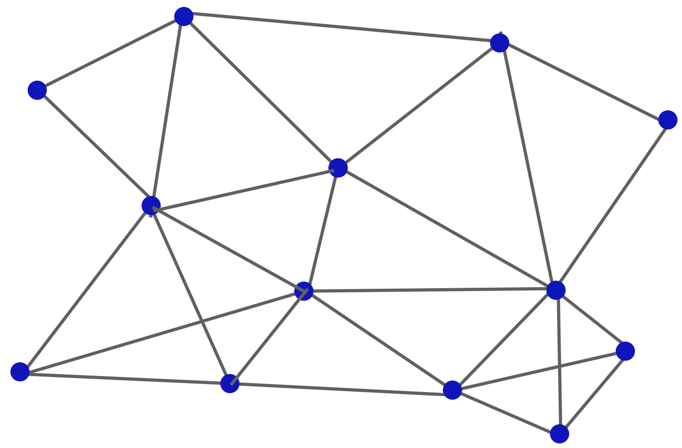
Sensor Network Localization

Existing Approach



MDS

$\Rightarrow$



Real Network

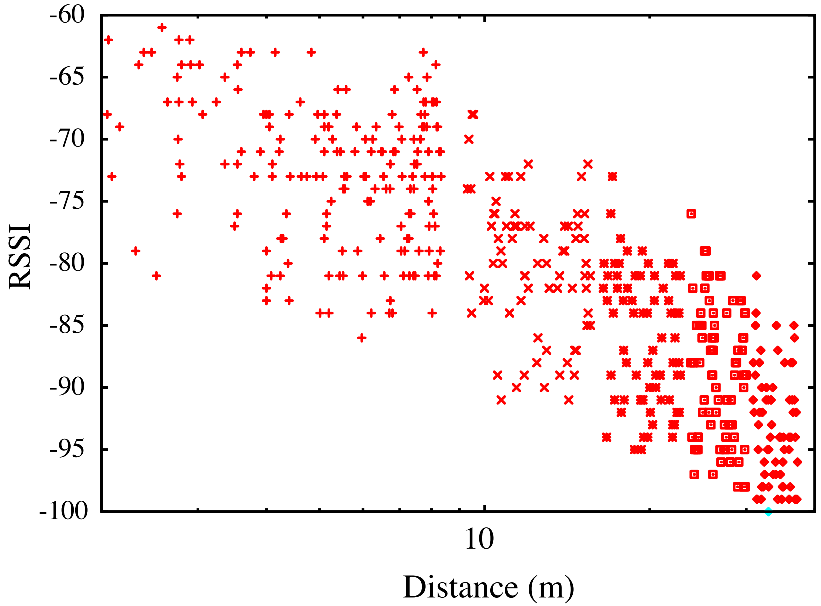
Computed Embedding

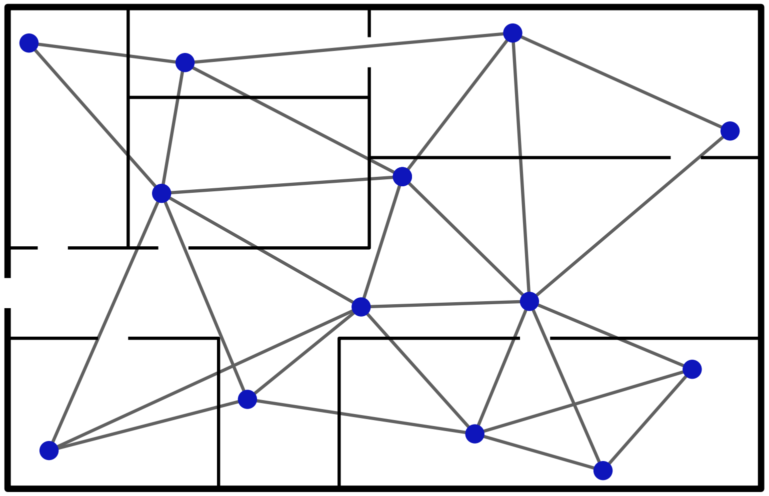
Compute embedding for est. distances with Multidimensional Scaling

(MDS-MAP, 2004)

Sensor Network Localization

Existing Approach - Problems

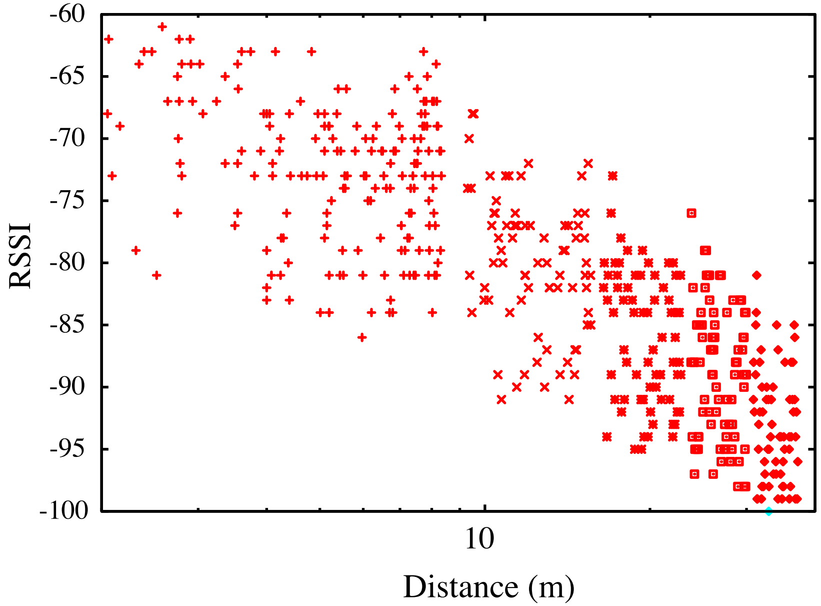


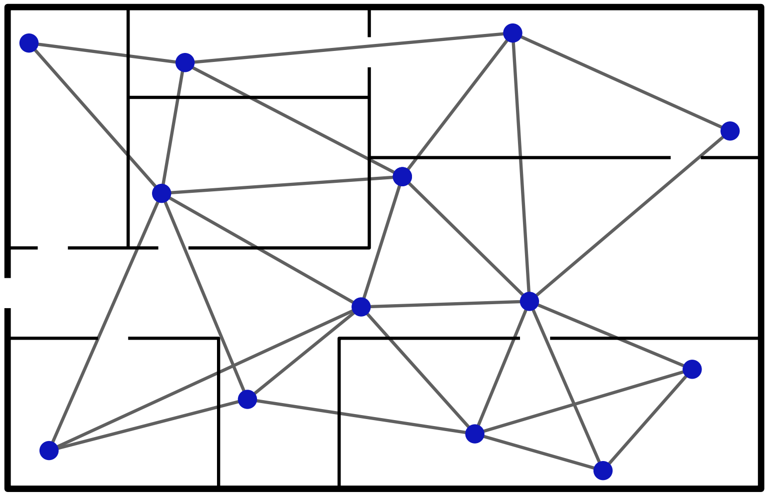


- signal strength fluctuates significantly
- small movements cause large change
- *but:* for fixed position almost constant
- obstacles, walls

Sensor Network Localization

Existing Approach - Problems



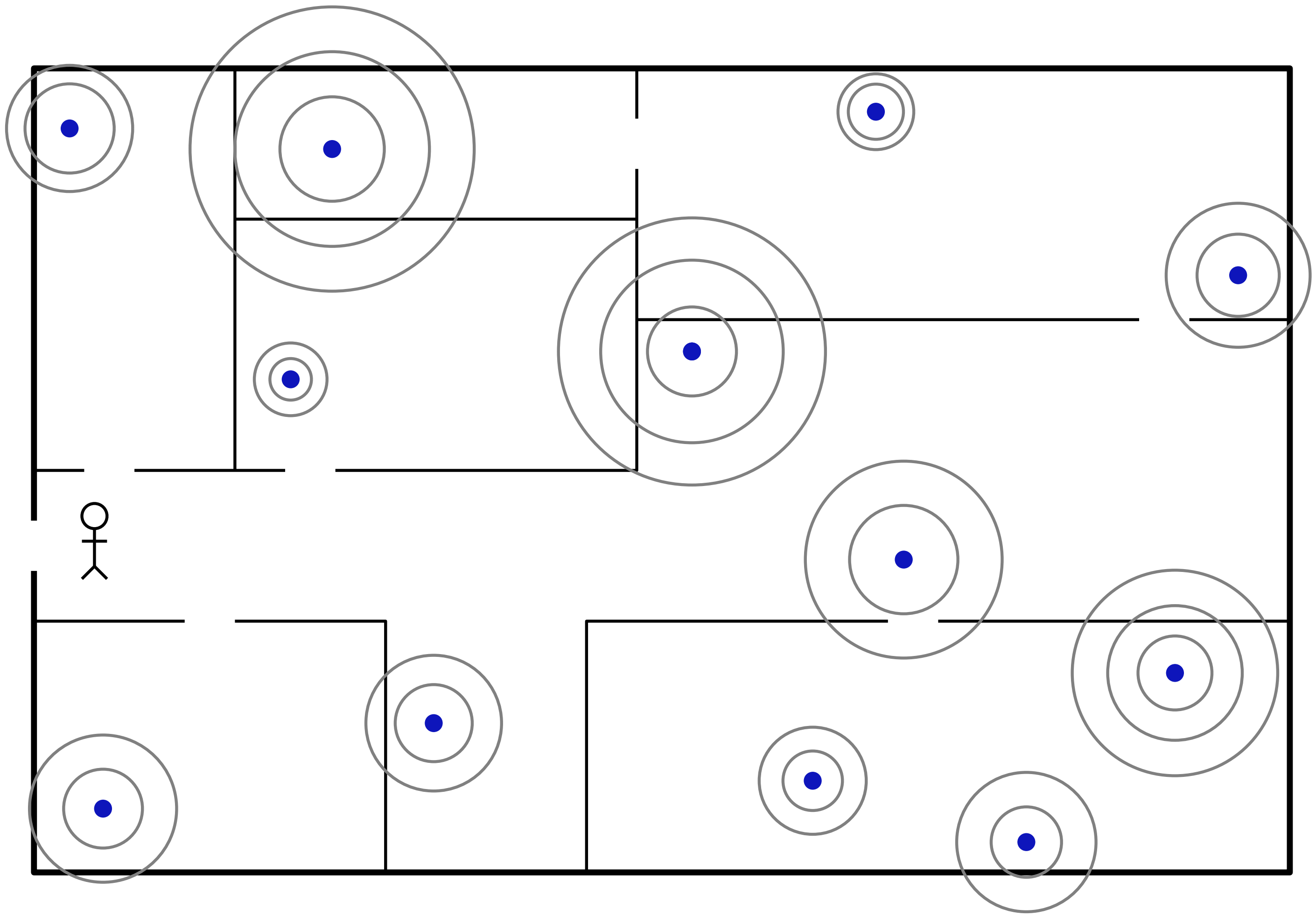


- signal strength fluctuates significantly
- small movements cause large change
- *but:* for fixed position almost constant
- obstacles, walls

Plan: use mobility to deal with this problems

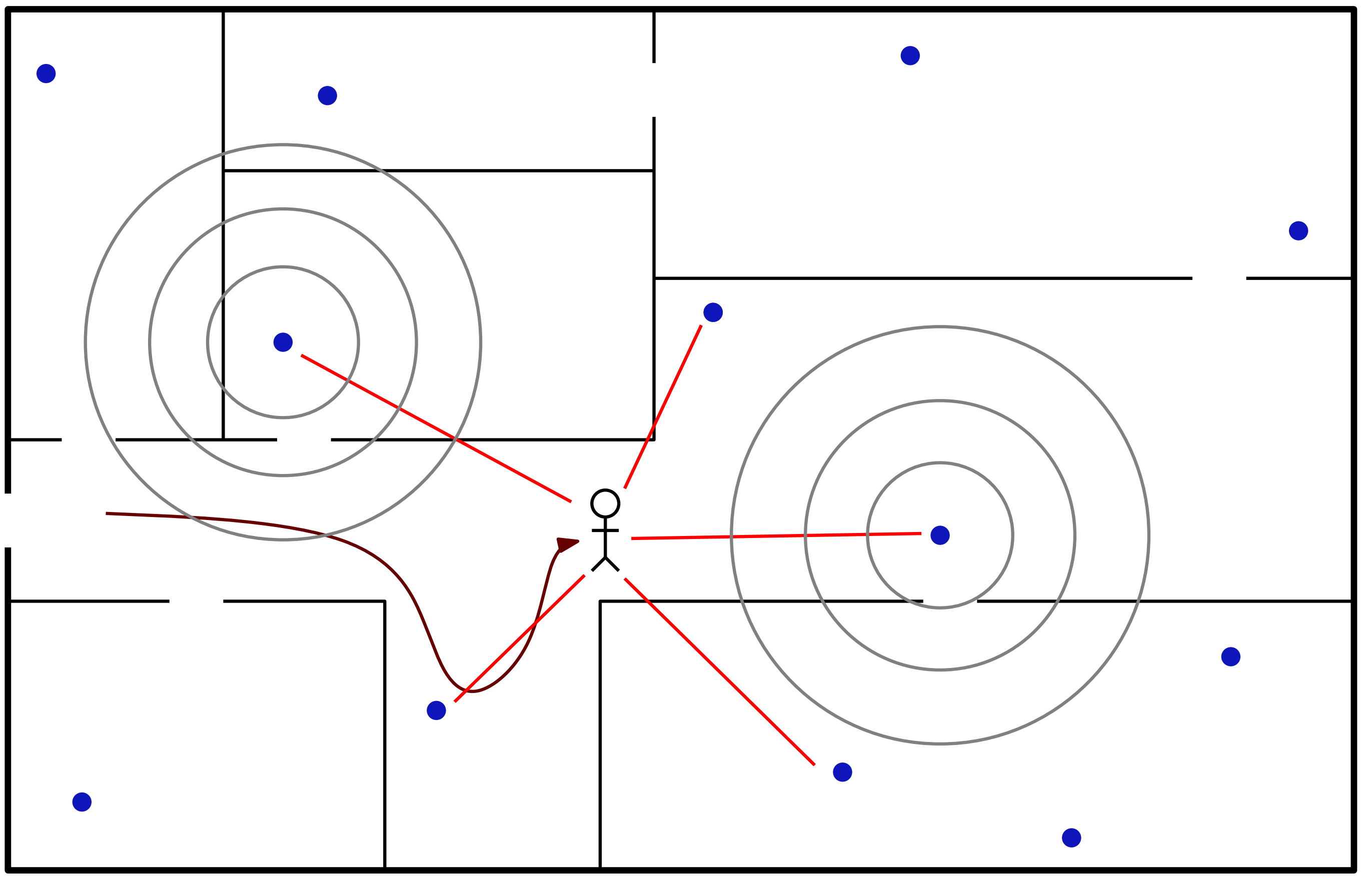
Sensor Network Localization

Indirect Network Localization



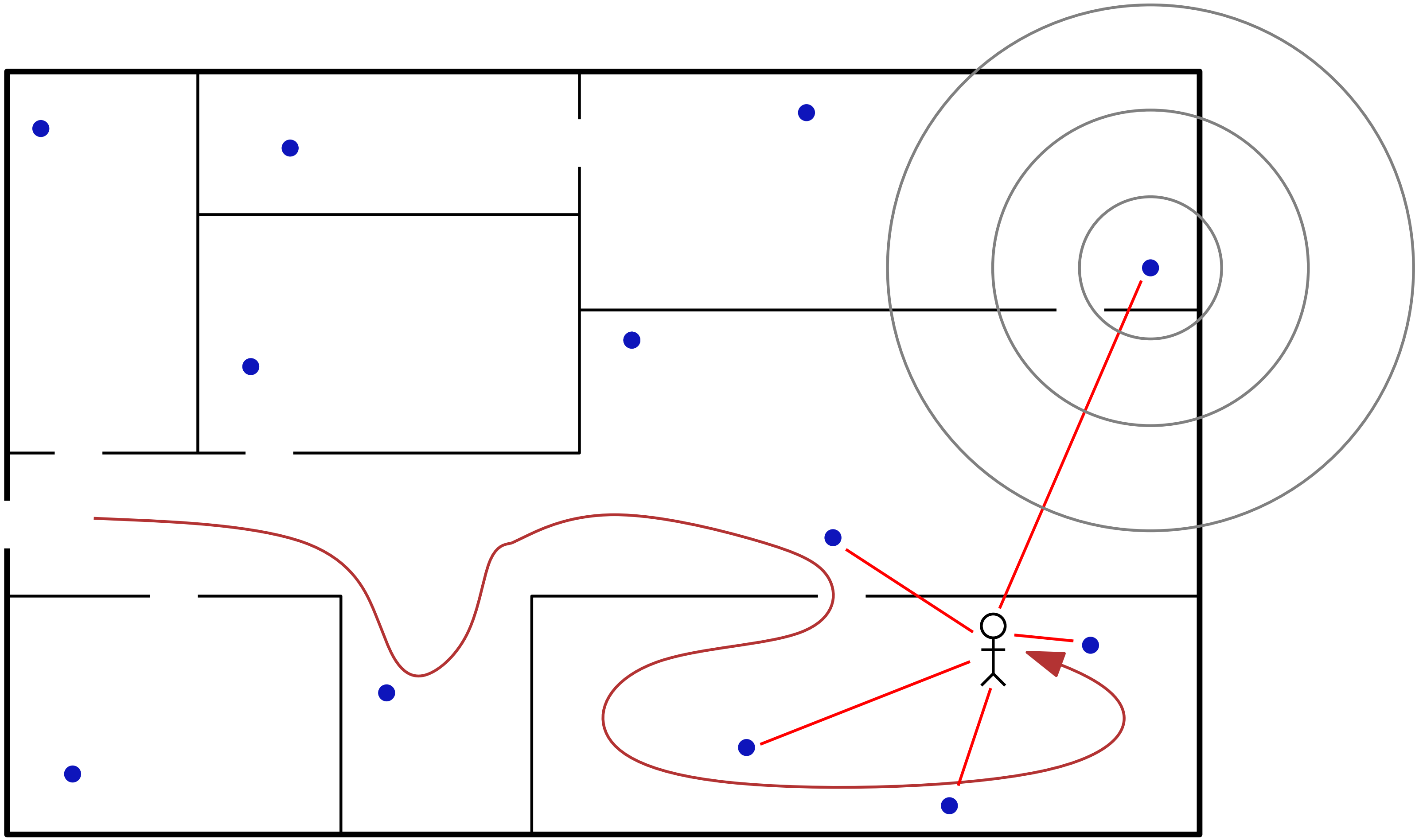
Sensor Network Localization

Indirect Network Localization



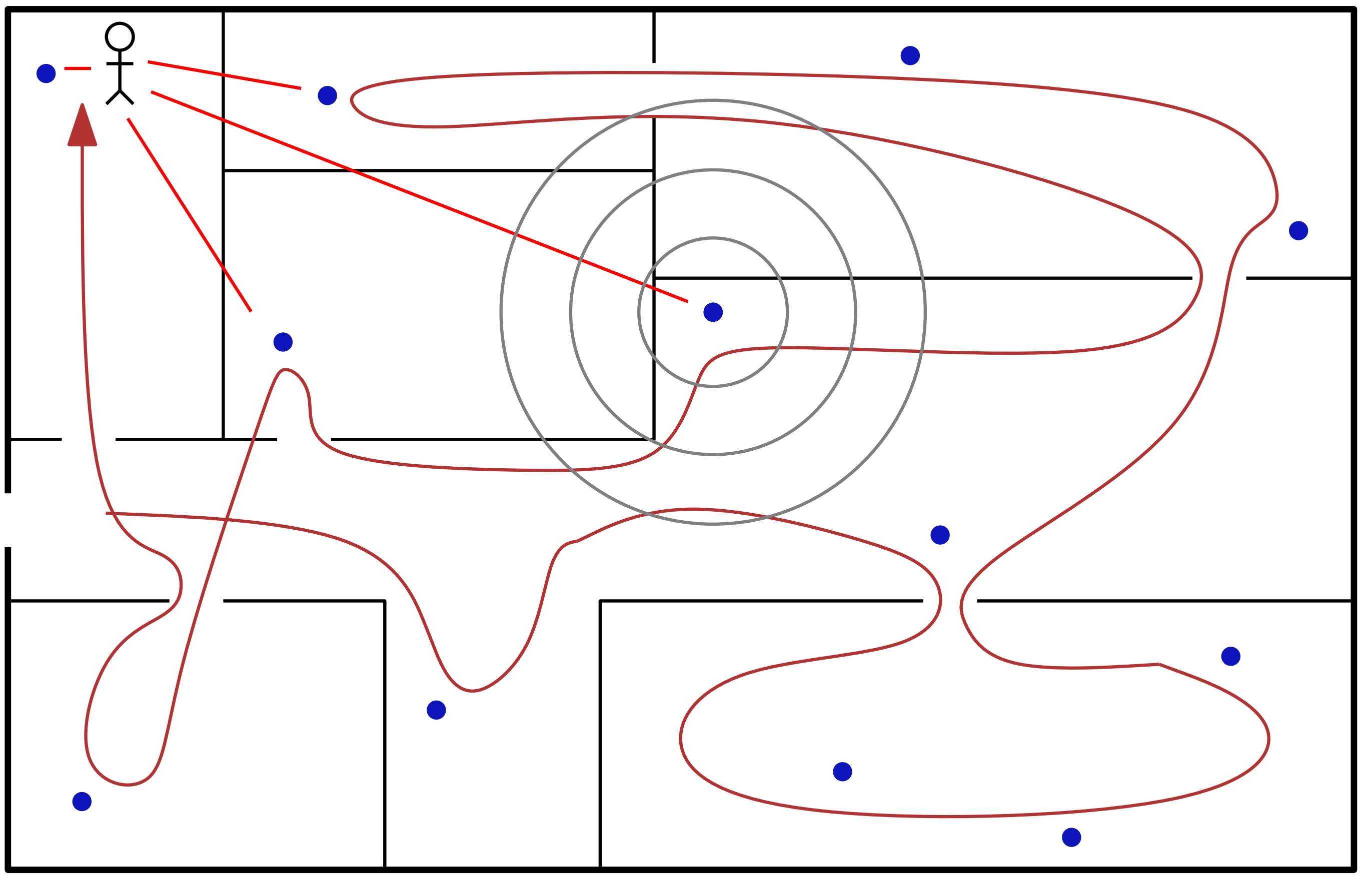
Sensor Network Localization

Indirect Network Localization



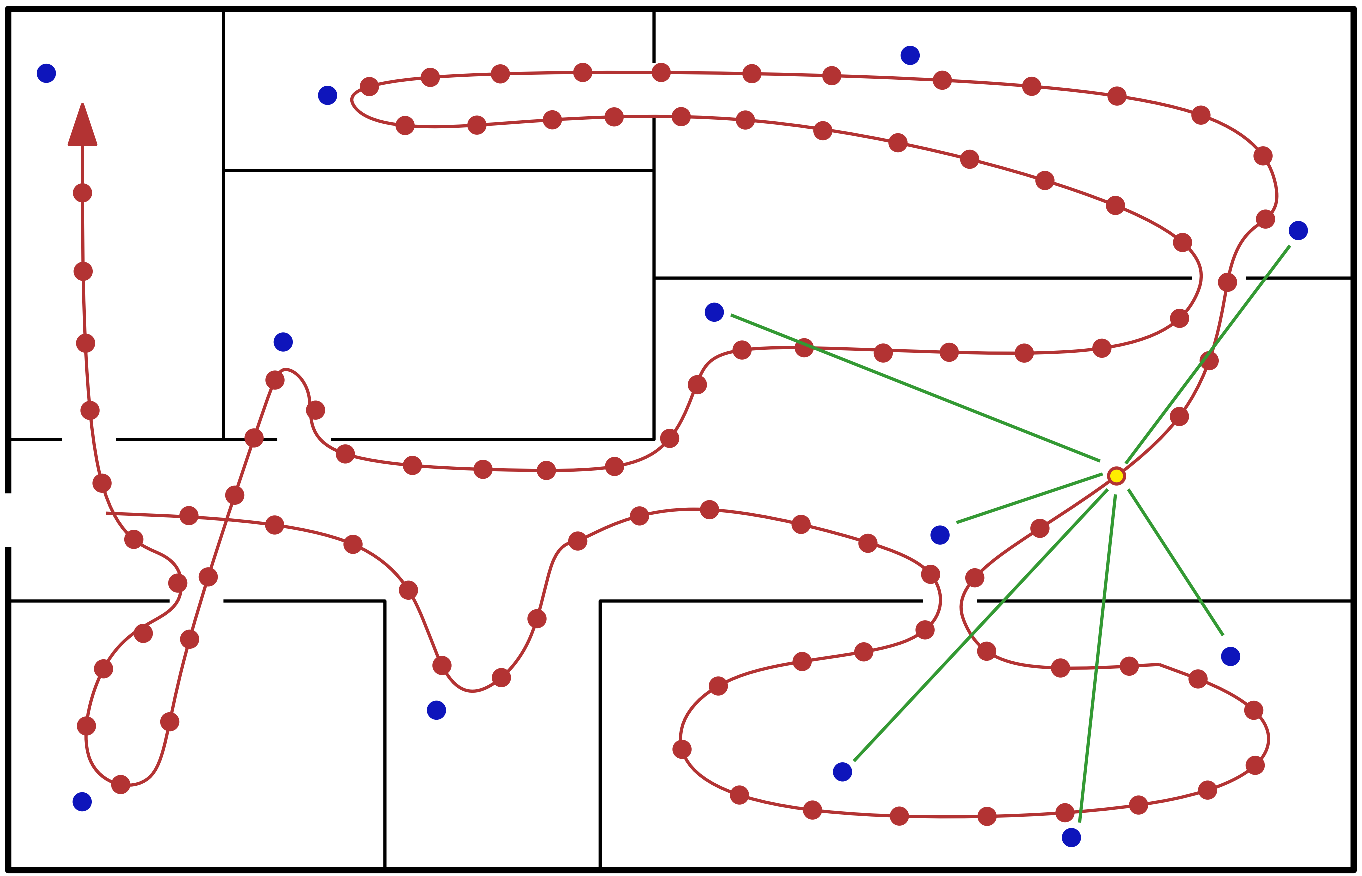
Sensor Network Localization

Indirect Network Localization



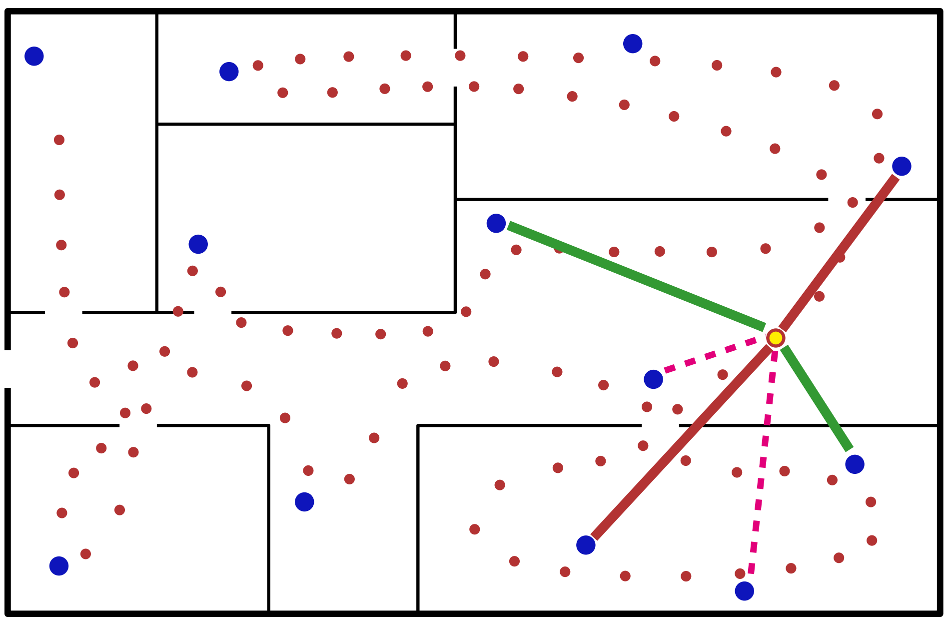
Sensor Network Localization

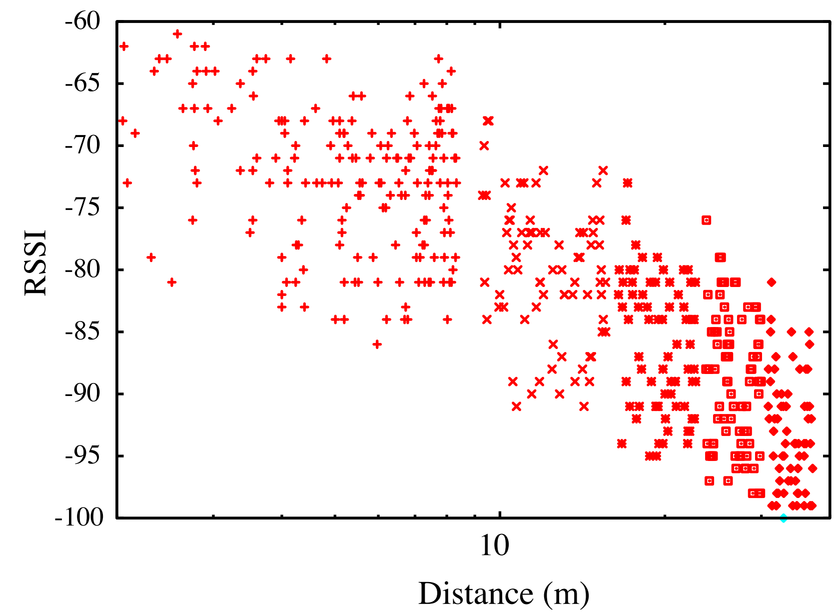
Indirect Network Localization



Sensor Network Localization

Indirect Network Localization - Advantages

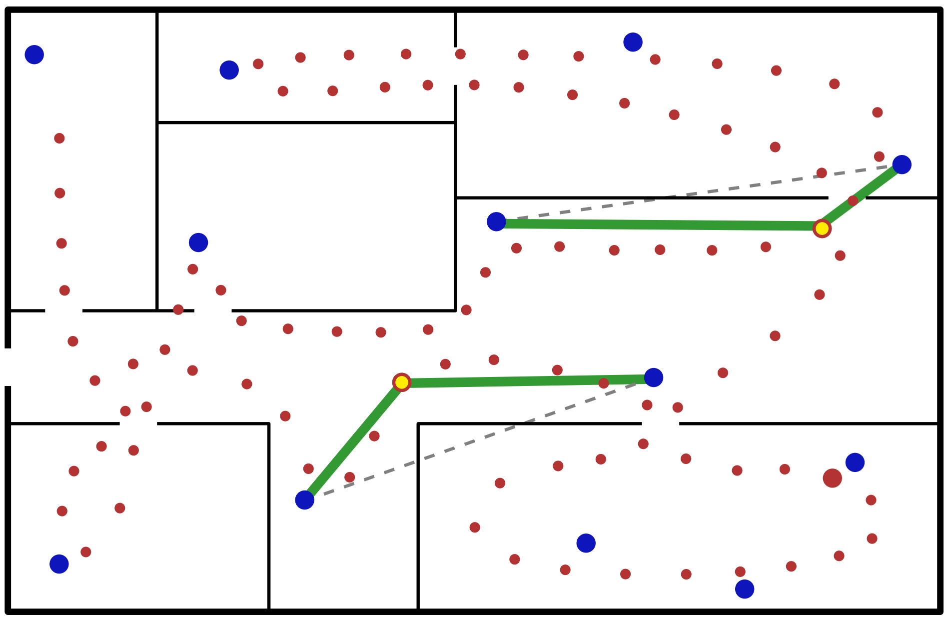




Measurements at different positions
$\Rightarrow$ averaging reduces errors

Sensor Network Localization

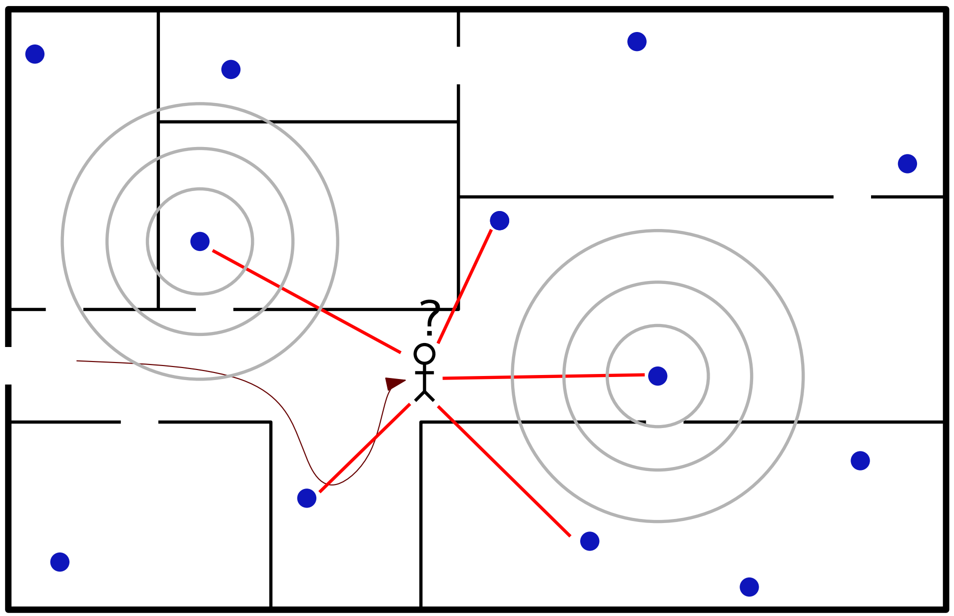
Indirect Network Localization - Advantages



Sometimes walls can be avoided

Sensor Network Localization

Indirect Network Localization - Advantages



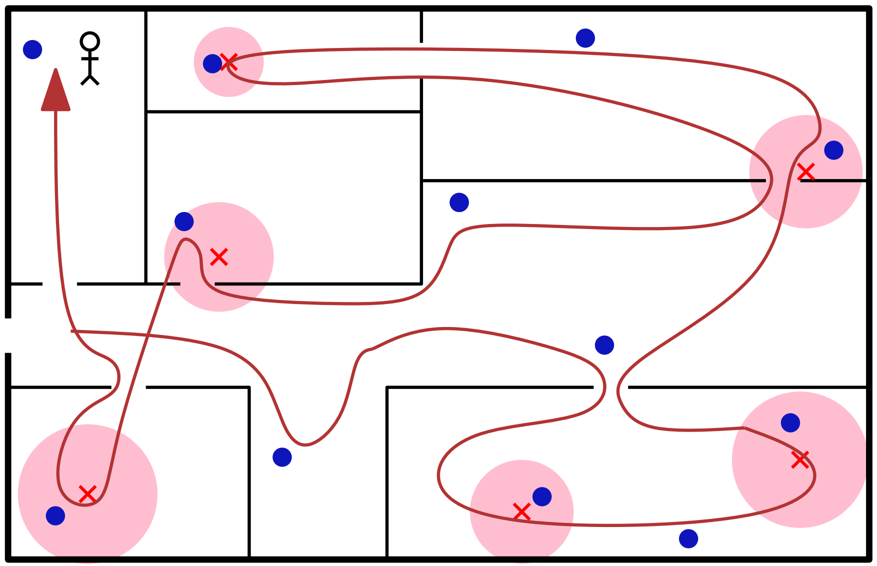
Stationary nodes only have to transmit
$\Rightarrow$ smaller, cheaper
$\Rightarrow$ many nodes possible

Sensor Network Localization

Existing approach

Information about (some) node positions or walked trajectory available

$\Rightarrow$ Use SLAM methods such as Kalman filter
(simultaneous localization and mapping)

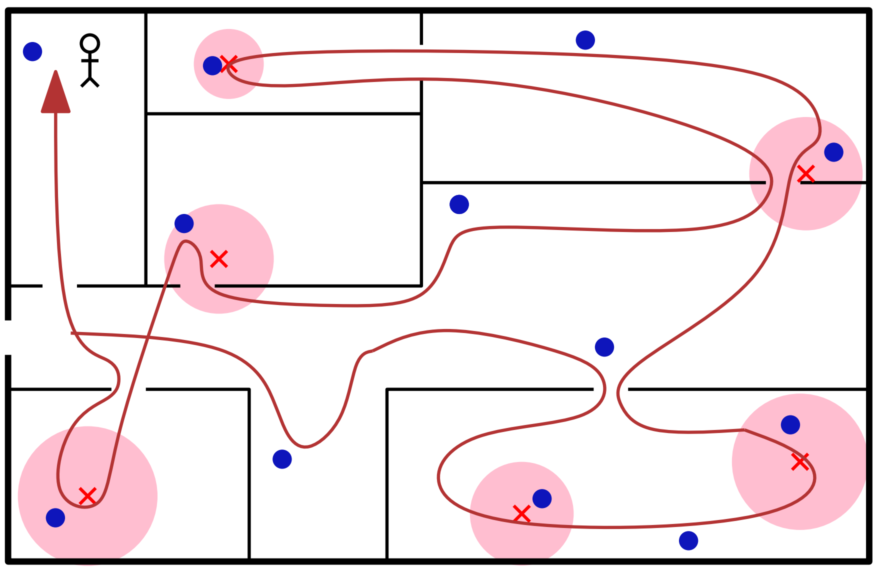


Sensor Network Localization

Existing approach

Information about (some) node positions or walked trajectory available

$\Rightarrow$ Use SLAM methods such as Kalman filter
(simultaneous localization and mapping)

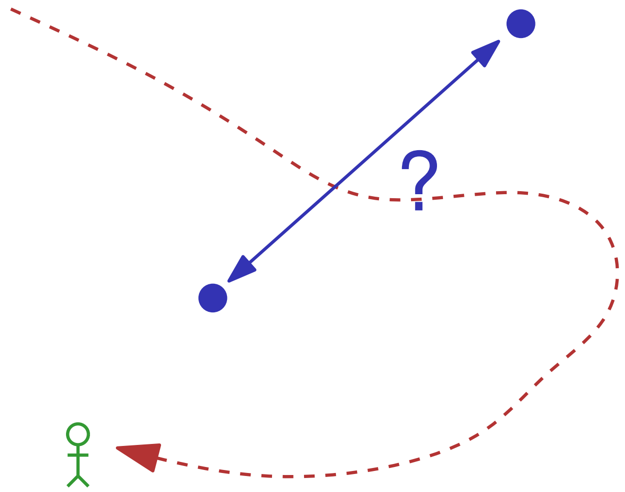


We assume that this kind of information is not available.

Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances



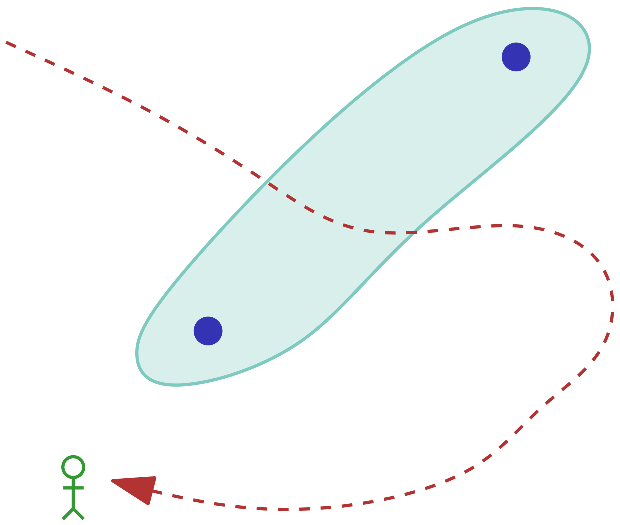
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

Possibility 1:

Two-way measurement



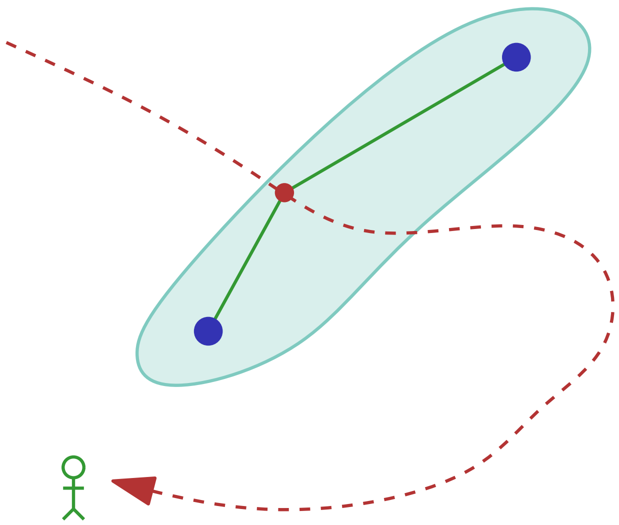
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

Possibility 1:

Two-way measurement



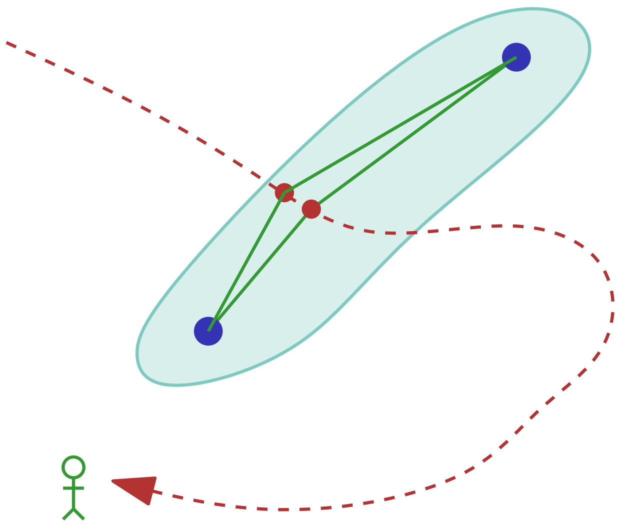
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

Possibility 1:

Two-way measurement



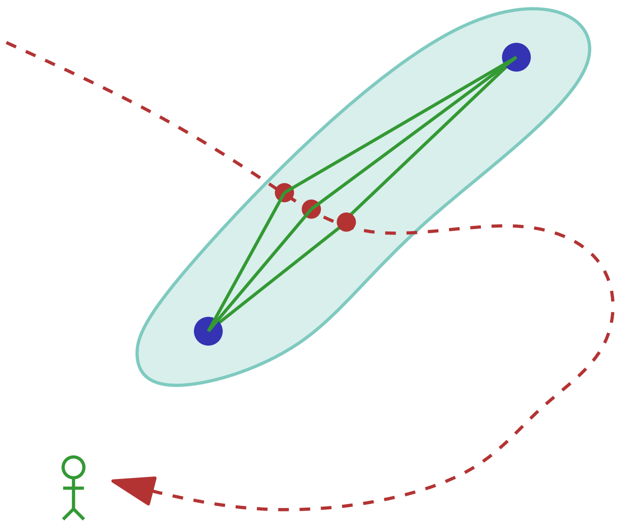
Sensor Network Localization

Planned Approach

- 1^{st} step: try to determine pairwise node distances

Possibility 1:

Two-way measurement



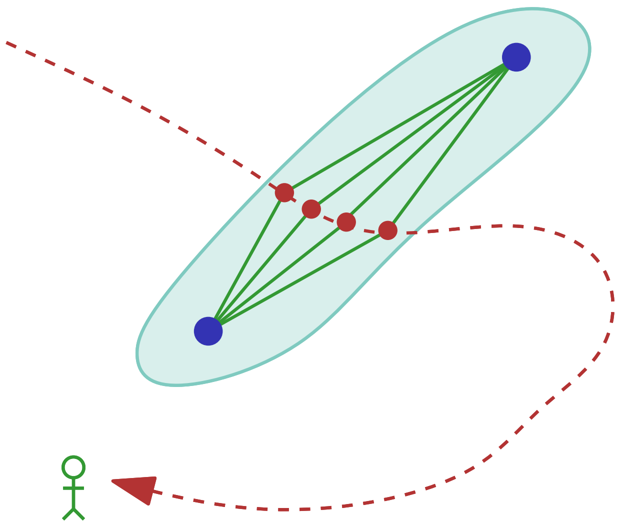
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

Possibility 1:

Two-way measurement



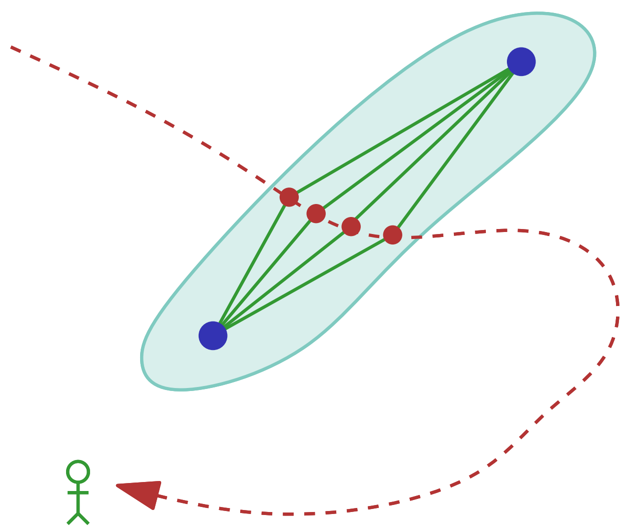
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

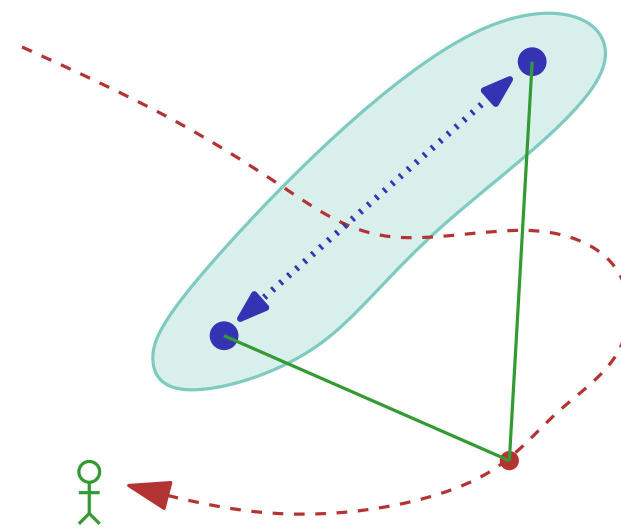
Possibility 1:

Two-way measurement



Problem

How to detect if measurement was made in the middle?



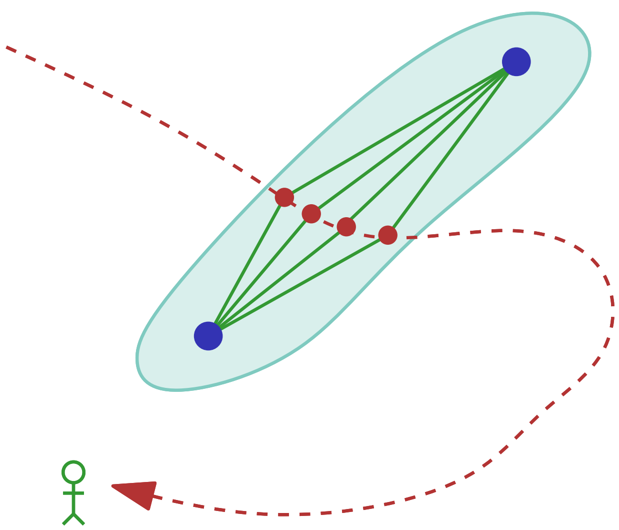
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

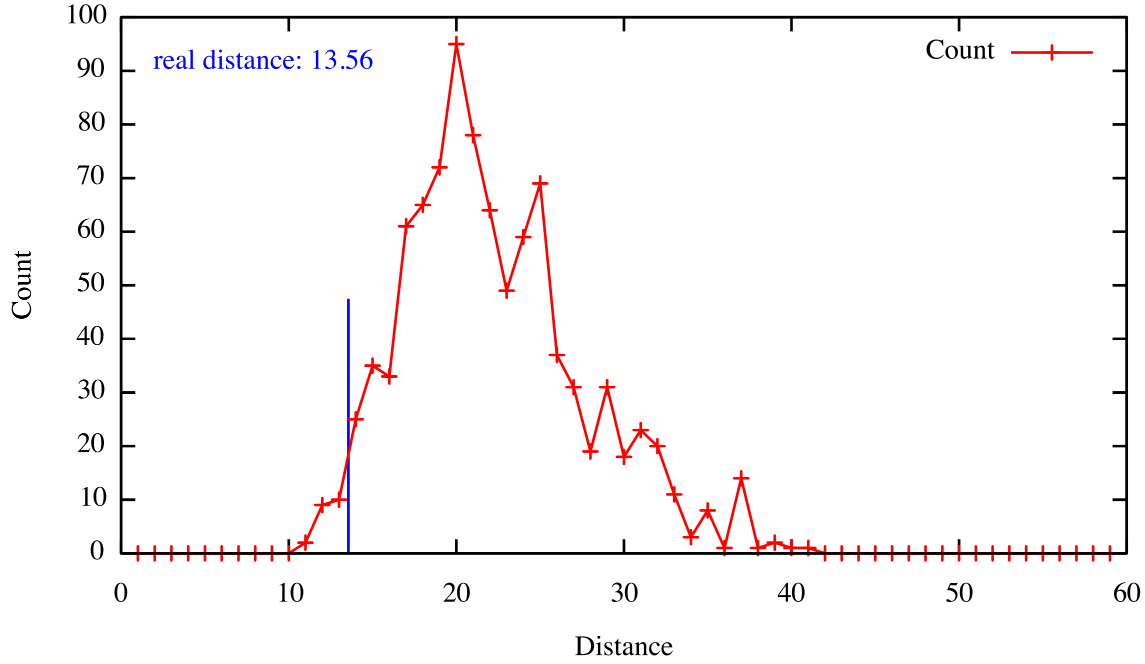
Possibility 1:

Two-way measurement



How to detect if measurement was made in the middle?

No averaging possible!

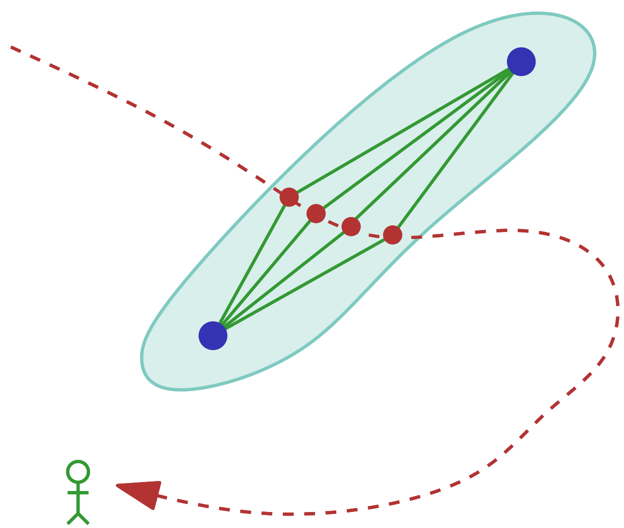


Sensor Network Localization

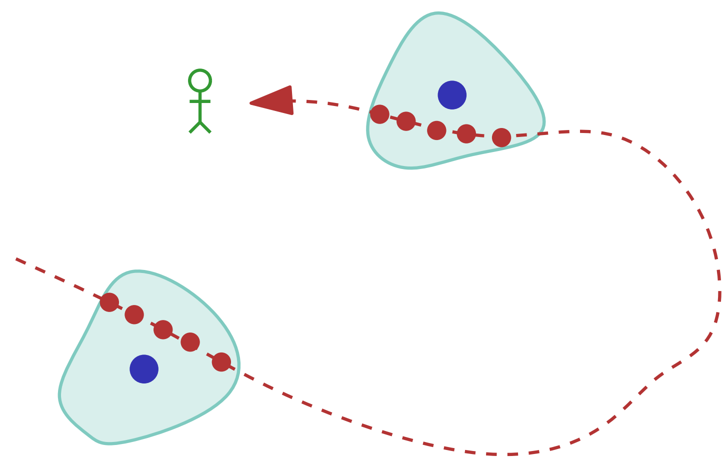
Planned Approach

- 1st step: try to determine pairwise node distances

Possibility 1:
Two-way measurement



Possibility 2:
One-way measurement



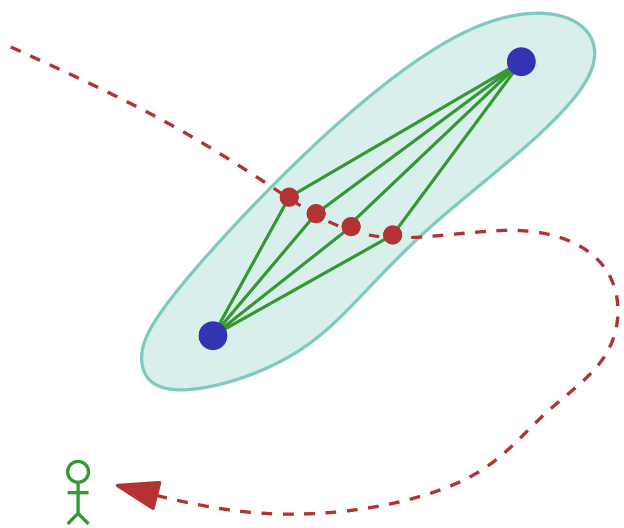
Sensor Network Localization

Planned Approach

- 1^{st} step: try to determine pairwise node distances

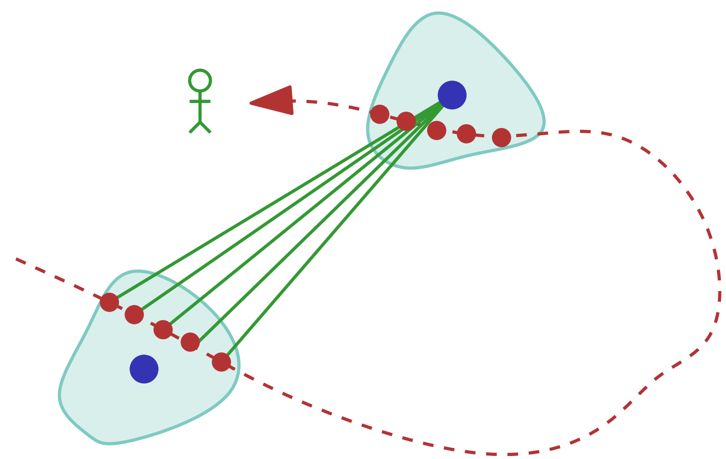
Possibility 1:

Two-way measurement



Possibility 2:

One-way measurement



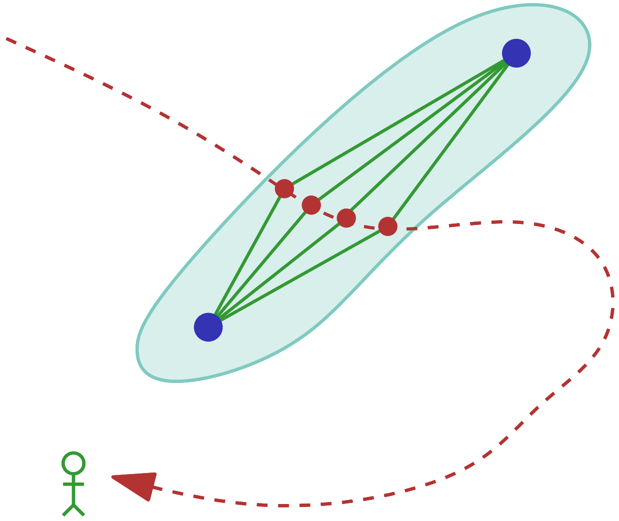
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

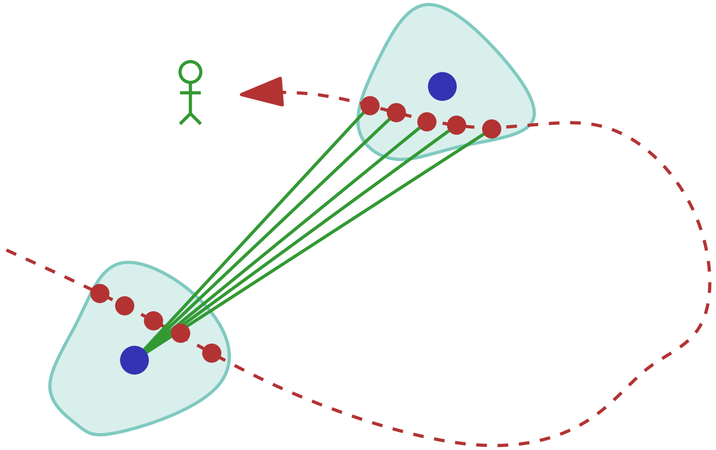
Possibility 1:

Two-way measurement



Possibility 2:

One-way measurement



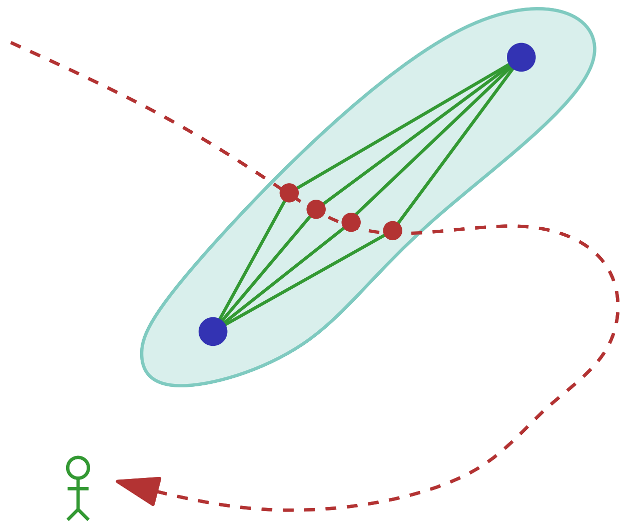
Sensor Network Localization

Planned Approach

- 1st step: try to determine pairwise node distances

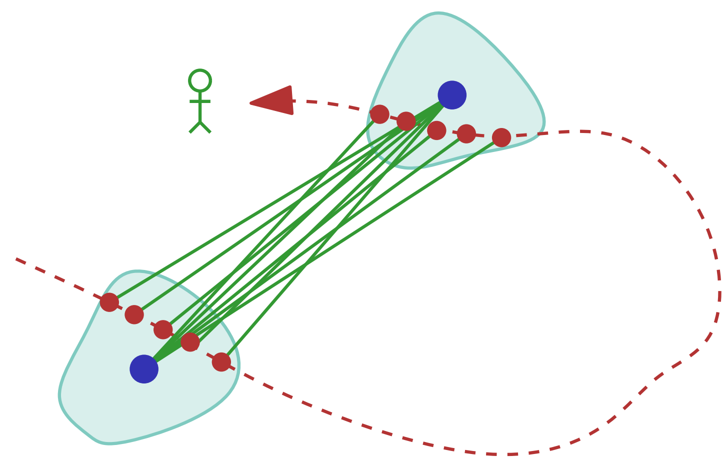
Possibility 1:

Two-way measurement



Possibility 2:

One-way measurement

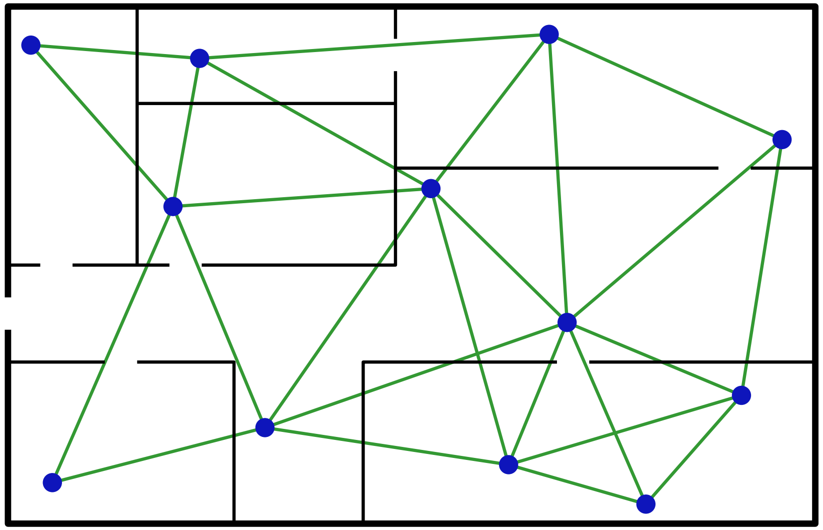


All measurements are equal!
Averaging possible!

Sensor Network Localization

Planned Approach

- 2nd step: completing the distance matrix

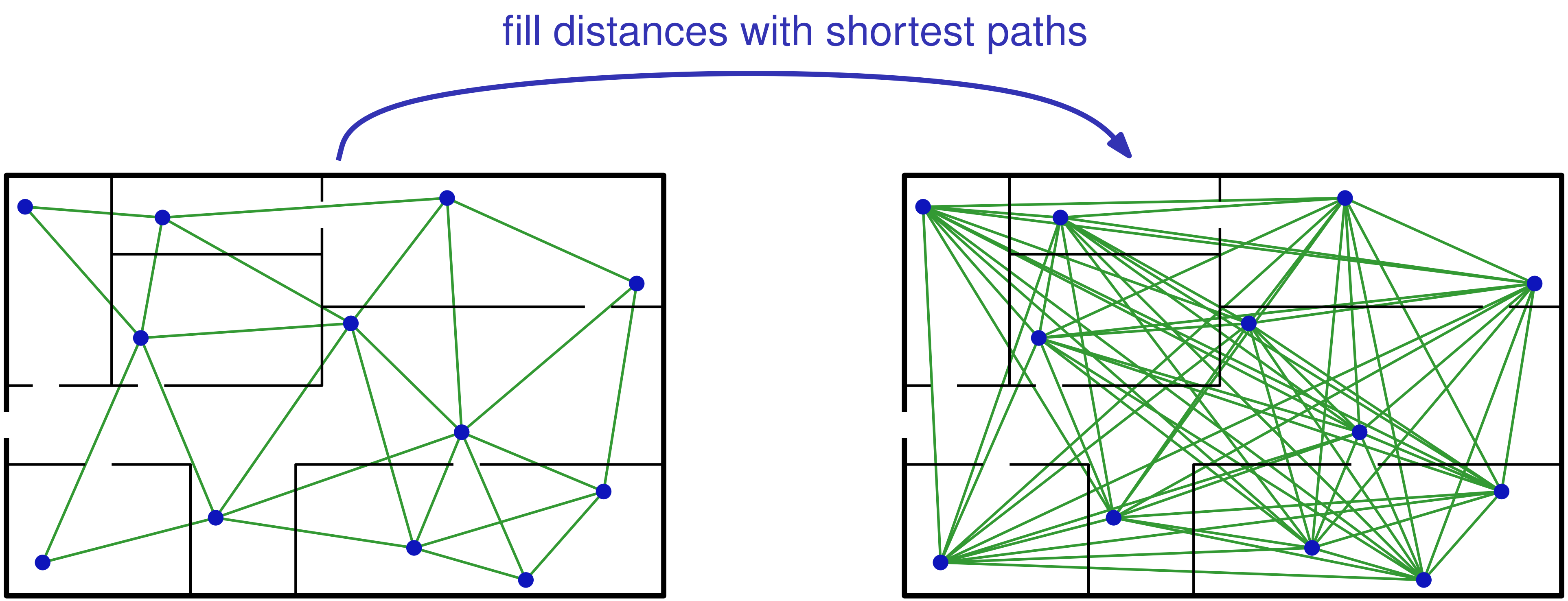


Pairwise distance estimate
between some nodes

Sensor Network Localization

Planned Approach

- 2nd step: completing the distance matrix



Pairwise distance estimate between some nodes

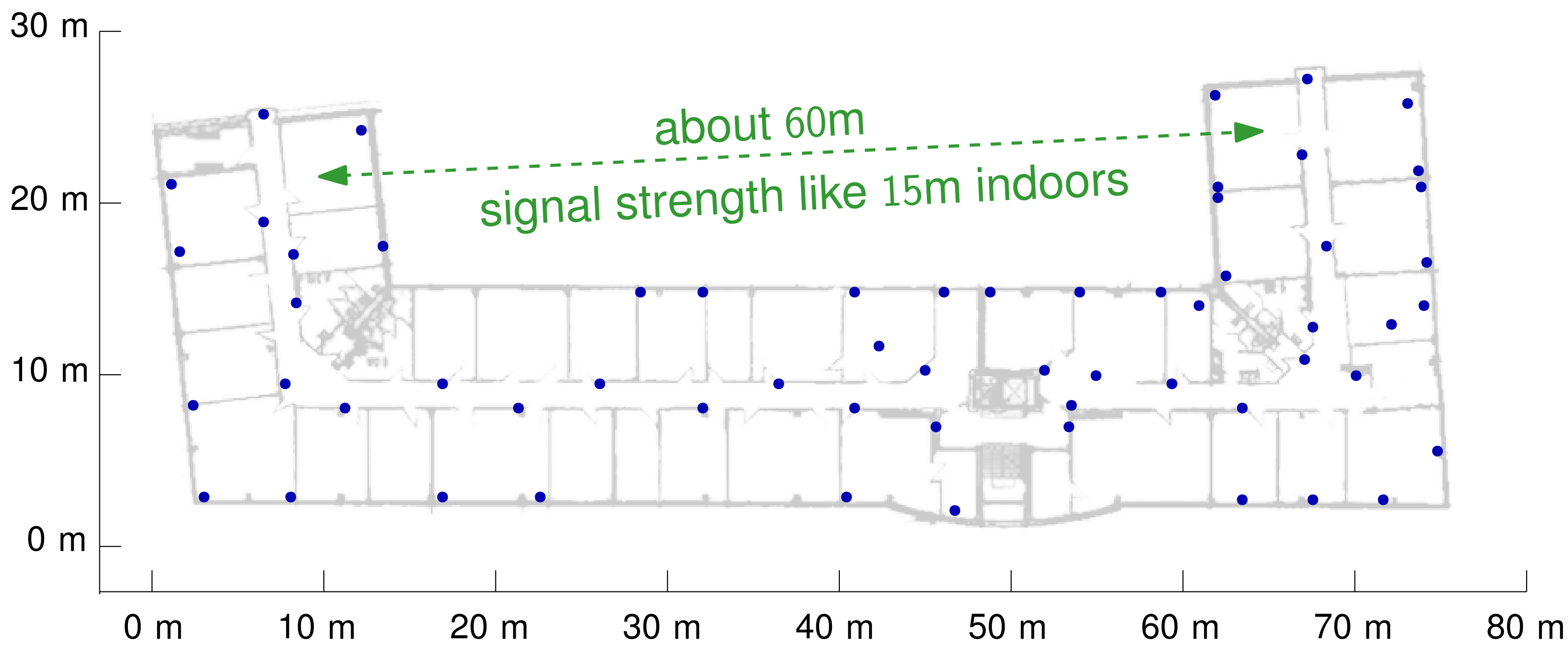
Pairwise distance estimate between all nodes

Sensor Network Localization

Planned Approach

- 2nd step: completing the distance matrix

Problem 1: some distances are extremely underestimated

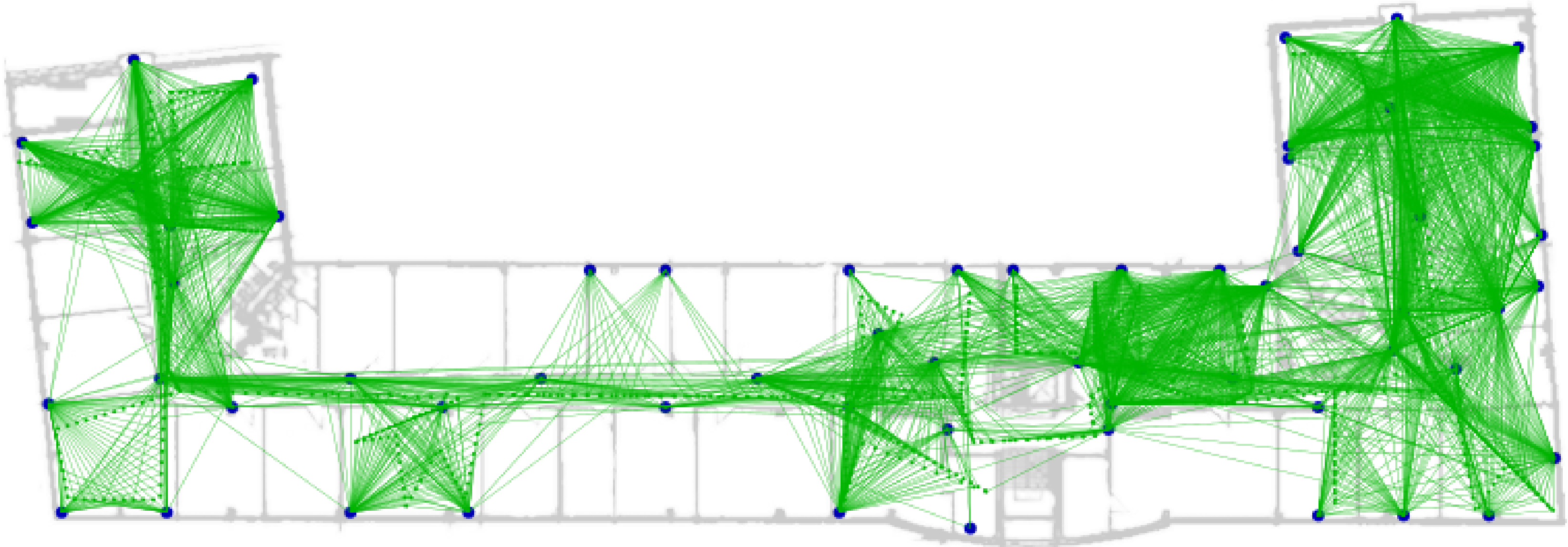


Sensor Network Localization

Planned Approach

- 2^{nd} step: completing the distance matrix

  Possible solution: use only strong signals



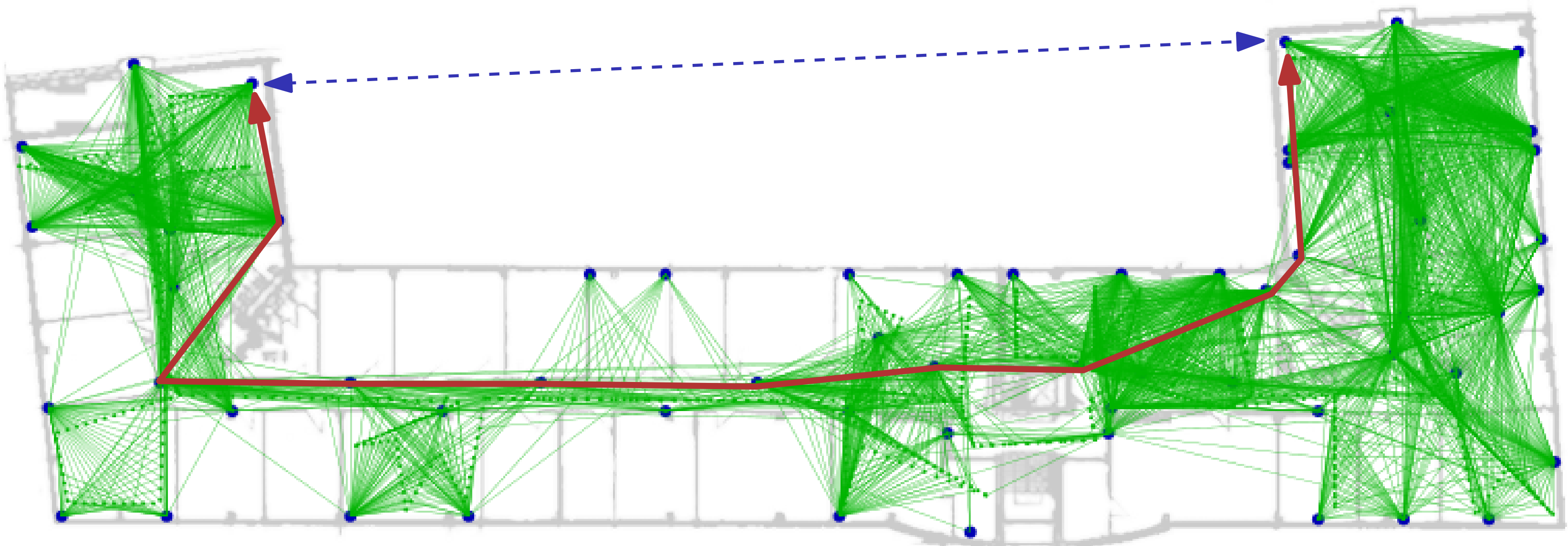
(measurements with RSS < -75 dBm are filtered)

Sensor Network Localization

Planned Approach

- 2nd step: completing the distance matrix

Problem 2: real distance often overestimated by shortest paths

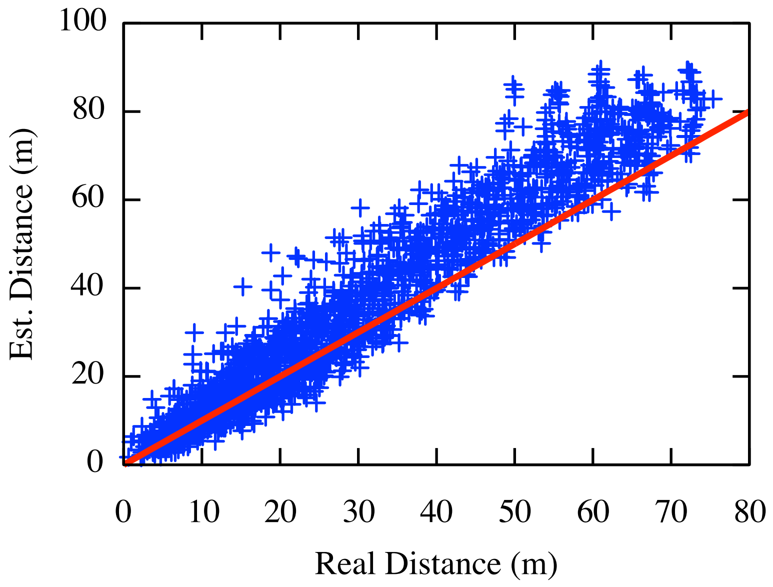


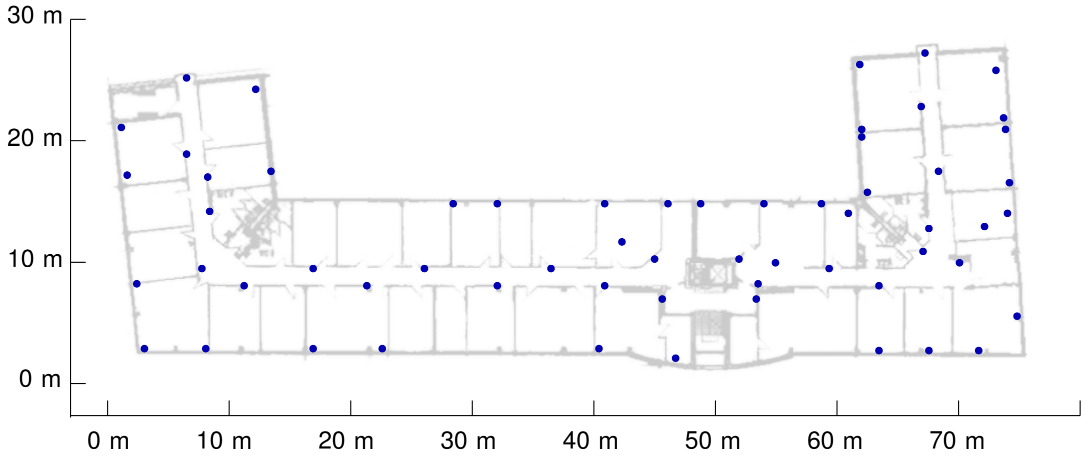
Sensor Network Localization

Planned Approach

- 2nd step: completing the distance matrix

Possible result





Sensor Network Localization

Planned Approach

- 3rd step: computing an embedding

Naive approach: Standard Multidimensional Scaling

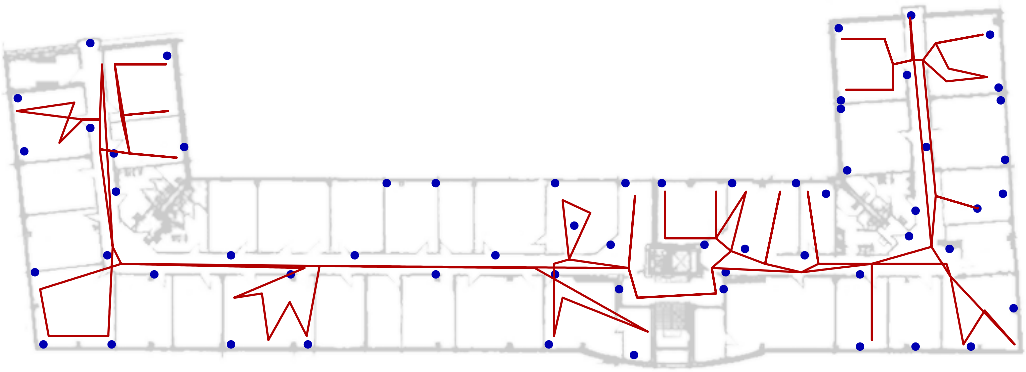
Institut für Theoretische Informatik
Algorithmik I

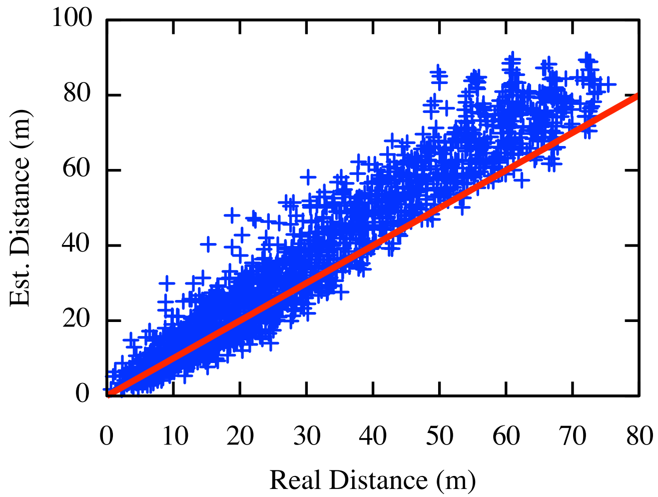
Sensor Network Localization

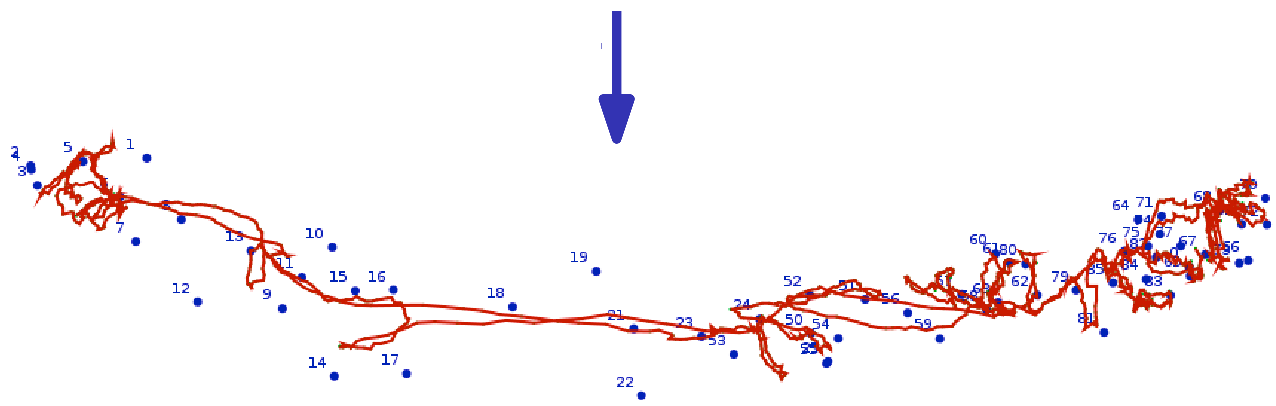
Planned Approach

- 3rd step: computing an embedding

Naive approach: Standard Multidimensional Scaling







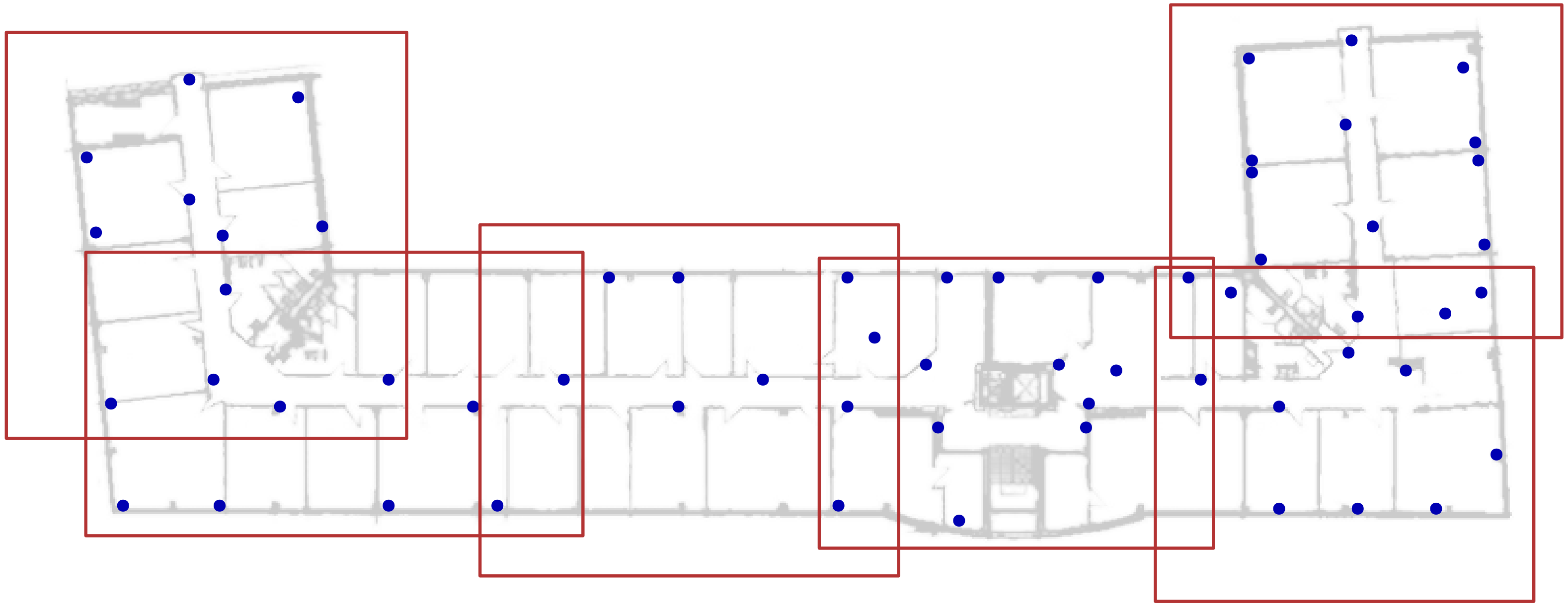
known problem for MDS-MAP

Sensor Network Localization

Planned Approach

- 3rd step: computing an embedding

Possible solution: Local MDS



TODO

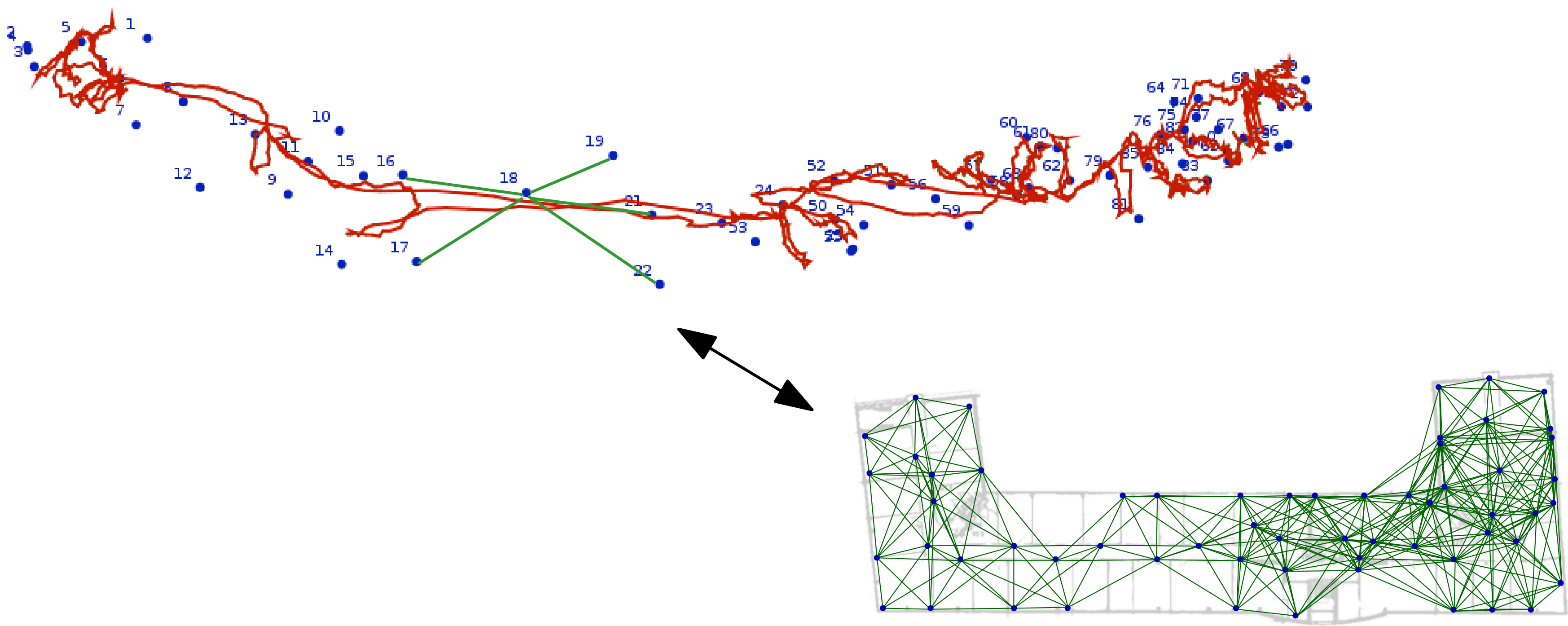
Institut für Theoretische Informatik
Algorithmik I

Sensor Network Localization

Planned Approach

- 3rd step: computing an embedding

Possible solution: Force-based methods (like spring embedder)



original pairwise distances

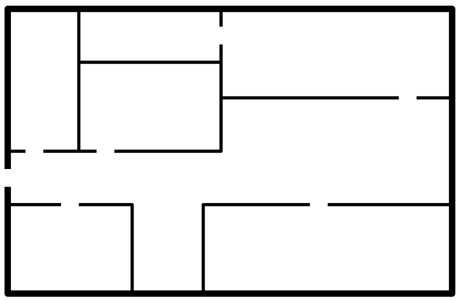
TODO

Sensor Network Localization

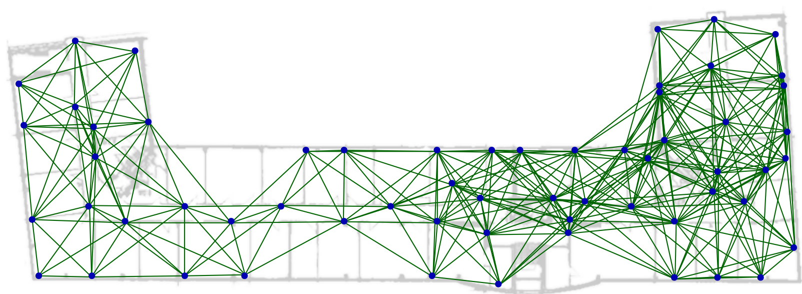
Comparison with simulations

Why do simulations of MDS-MAP usually look so good?

- usually 5%-10% avg. distance estimation error assumed
  $\Rightarrow$ RSS based in reality rather 20% to 50%
- usually smooth distribution of obstacles assumed
  $\Rightarrow$ in reality inhomogeneous (walls)

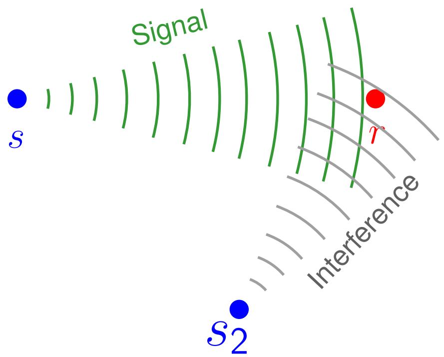


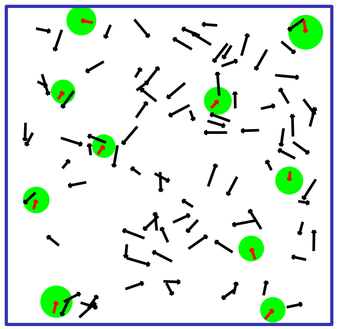
- usually rectangular area assumed
  $\Rightarrow$ in reality more complicated shapes possible

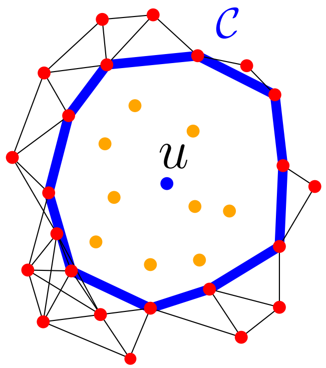


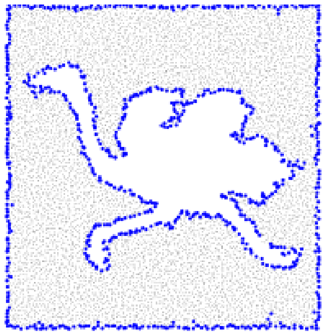
Karlsruhe Institute of Technology

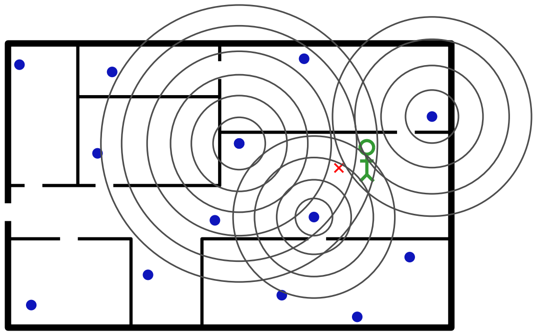
Thank you for your attention!

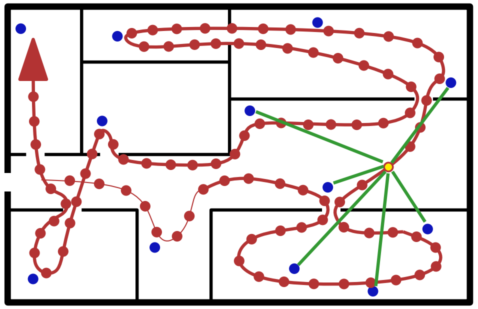












Do you have any questions?