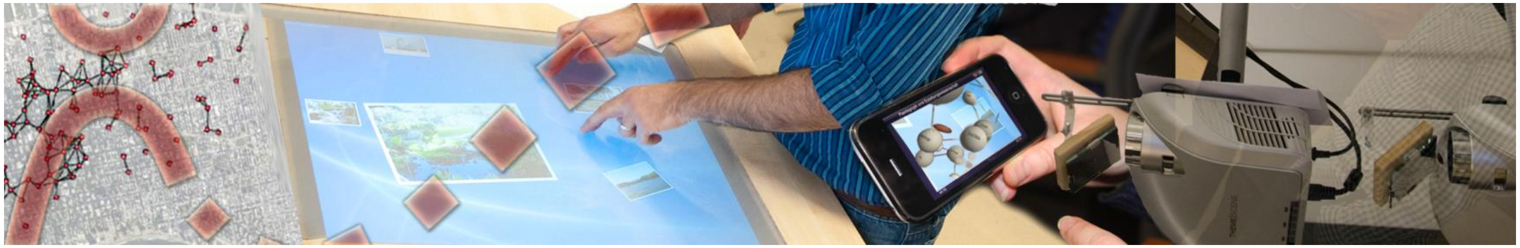
Technische Universität Darmstadt

Telecooperation Lab
Prof. Dr. Max Mühlhäuser

TK1: Distributed Systems - Programming & Algorithms

Chapter 3: Distributed Algorithms
Section 5: Local Algorithms
Lecturer: **Prof. Dr. Max Mühlhäuser**

Contributors: **Michael Stein (michael.stein@tk.informatik.tu-darmstadt.de), Immanuel Schweizer (schweizer@tk.informatik.tu-darmstadt.de)**

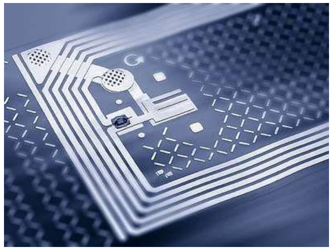


OUTLINE

- **Motivation**
- Local Algorithms in Theoretical Context
- Theory Meets Practice…
- Summary

Introduction

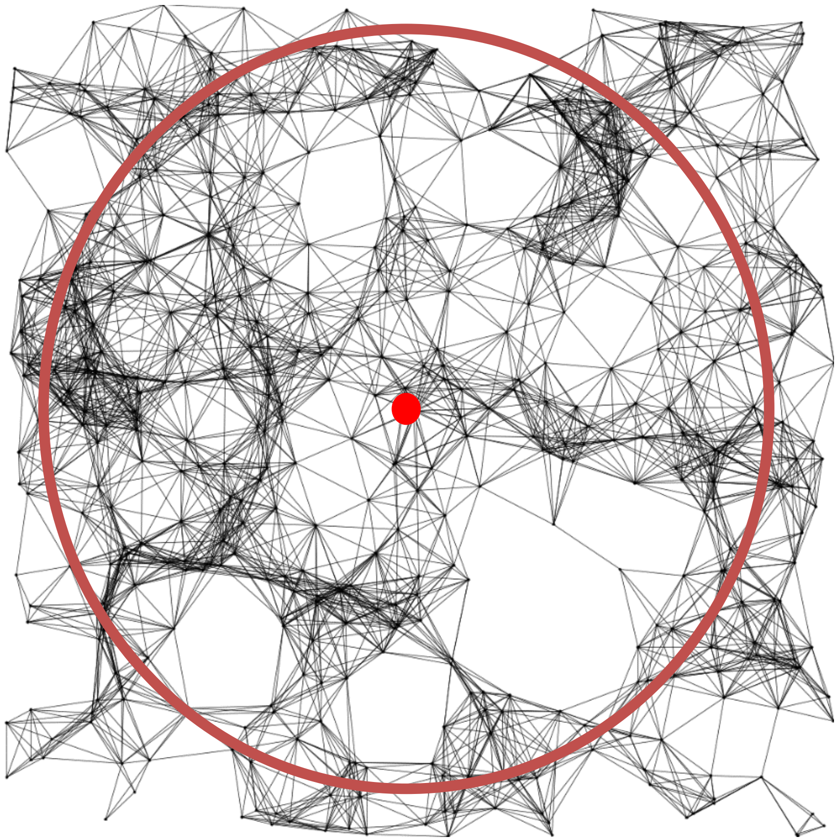
- The size of the Internet increases rapidly!
  - 2015: 15 Billion (Cisco, 2011)
  - 2020: 20 Billion (IMS Research, 2010)



- Internet of Things
  - RFID
  - Sensors
- One fundamental issue: **Scalability**

Reducing the Scope of Knowledge

- **Observation:** Considering scalability, having a global view on the network is too expensive



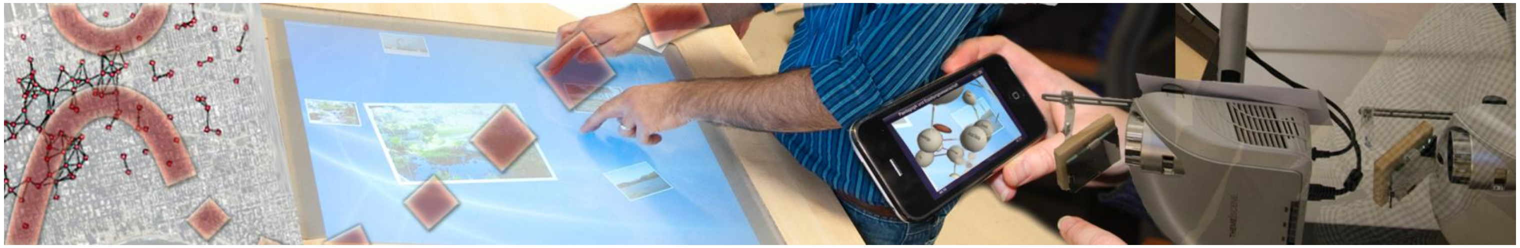
- **Intuition:** Limiting interactions to the local neighborhood of individual nodes improves scalability
- Other desired properties (e.g., robustness)
- **Problem:** Trade-Off (Performance vs. Scope)
- What can be done based only on local knowledge?

The Waspmote figure was taken from the Libelium Website http://www.libelium.com/products/waspmote/, February 2015

Fundamental Questions

- What is a local algorithm?

- What can be computed locally?



OUTLINE

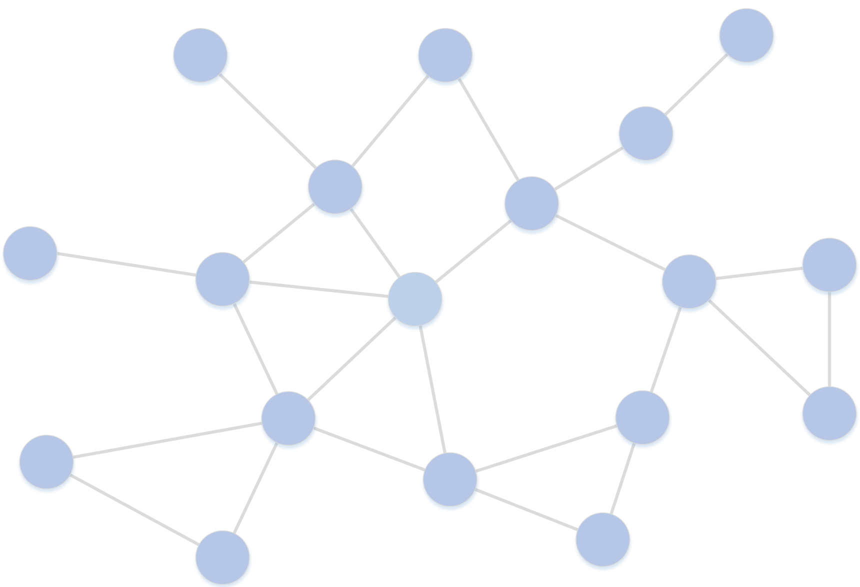
- Motivation
- **Local Algorithms in Theoretical Context**
- Theory Meets Practice…
- Summary

Graph Definitions

- We use a graph G = (V,E) do model communication topologies

- Vertices (V) (also called nodes)
  - Denote networking devices
  - $|V| = n$

- Edges (E)
  - Denote connectivity between devices
  - Might be directed or undirected



In Theory… The $\mathcal{LOCAL}$ Model

- Wide-spread deterministic model of distributed computing (Initially proposed by Linial [1992])
- The nodes perform globally synchronized communication rounds

Initially, each node has only task-specific input and no knowledge about other nodes in the network.

In each round, each node performs the following three consecutive steps:

1. Local computation
2. Send a message to each neighbor
3. Receive one message from each neighbor

After k rounds, the algorithm terminates and each node computes and announces its local output.

Properties of the $LOCAL$ model

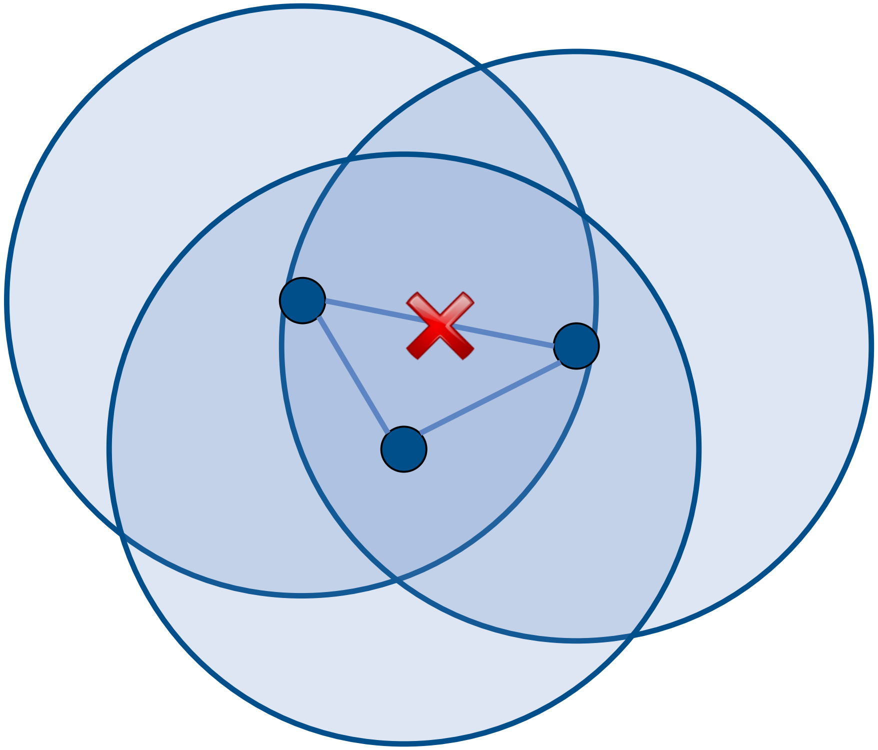
- This model focuses on the locality in distributed processing [Linial 1992] and abstracts away restricting factors [Peleg 2000]
  - Computation is free
  - Message size is unbounded

- **Observation:** In k synchronous communication rounds, information may travel at most k hops through the network
- Bounding k, we are able to ensure that nodes have only restricted knowledge on the network!

- **In the following, we focus on algorithms where k is a constant that is independent from the network size [Suomela 2013]**
- Such algorithms run in constant time independently from the network size!

kTC: An Example of an algorithm in the $\mathcal{LOCAL}$ Model

- **Related Work:** Local Algorithm **kTC** [Schweizer et al. 2012]
  - **Round 1:** Each node broadcasts its ID
  - **Round 2:** Each node broadcasts its 1-hop neighborhood
  - **After Round 2:** Remove the longest edge in each triangle (e.g., based on signal strength)

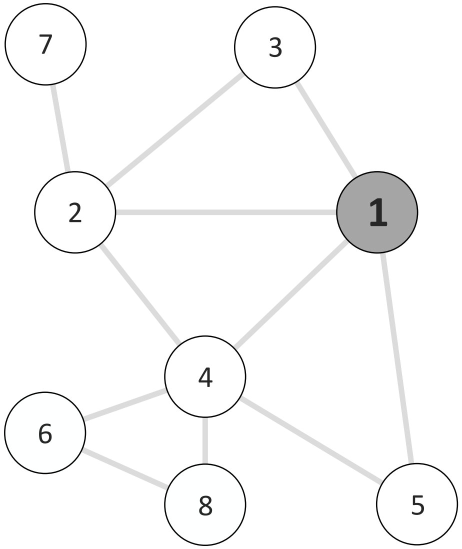


Definition: k-hop Neighborhood

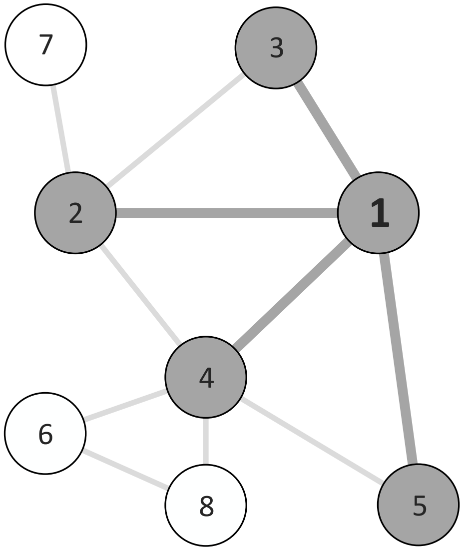
- When d(u,v) gives the hop count between u and v, the k-hop neighbors set of a node v is given by:

$$B_k(v) = \{u \in V_T : \min\{d(u,v), d(v,u)\} \leq k\}$$

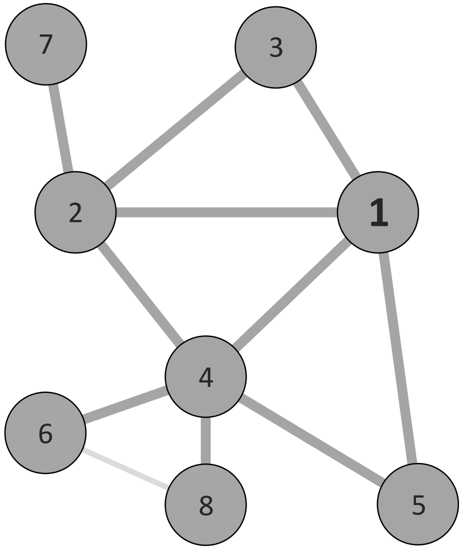
- $G_k(v)$ is the subgraph of G that can be constructed in the $\boldsymbol{\mathcal{LOCAL}}$ model by v in k rounds



$G_0(1)$



$G_1(1)$



$G_2(1)$

Limitations of the $\mathcal{LOCAL}$ Model

- What can be achieved in this model?

- In particular, there are two arguments that can be used to show that a certain problem can not be solved in this model [Suomela 2013]
  - Inherently non-local Problem
  - Impossibility of Symmetry Breaking

Inherently Non-Local Problems

- A problem is inherently non-local if the output of a node v depends on the initial input of a node outside of $G_k(v)$
- Example: Creating a spanning tree is inherently non-local

  [Suomela 2013]
- Illustration: An n-cycle where bold edges denote edges contained in the spanning tree
- The decision if (u_1,u_2) should be added to the tree depends on whether (v_1,v_2) is contained in the tree!

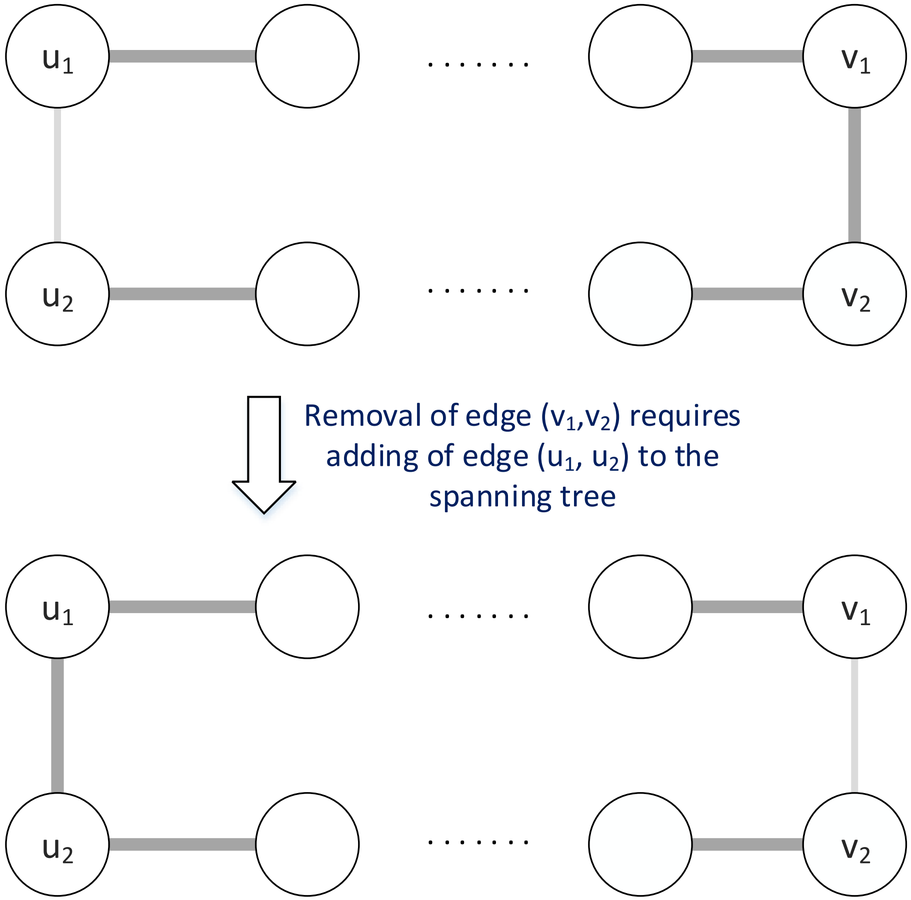
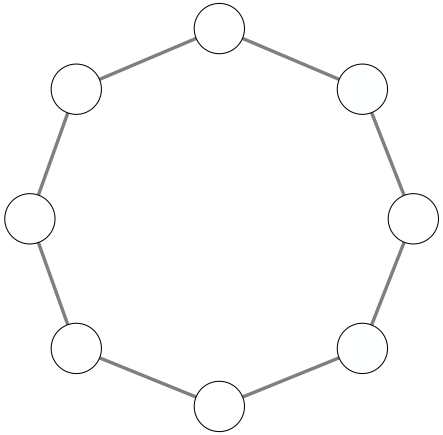


Figure adapted from [Suomela 2013]

Impossibility of Symmetry Breaking

- A distributed algorithm in the $LOCAL$ model is equivalent to a function that maps local neighborhoods G_k to local outputs [Fraigniaud et al. 2013]
- As the model is deterministic, two nodes with an equal view of the graph produce the same output!

- Example: Impossibility to assign addresses in an n-cycle



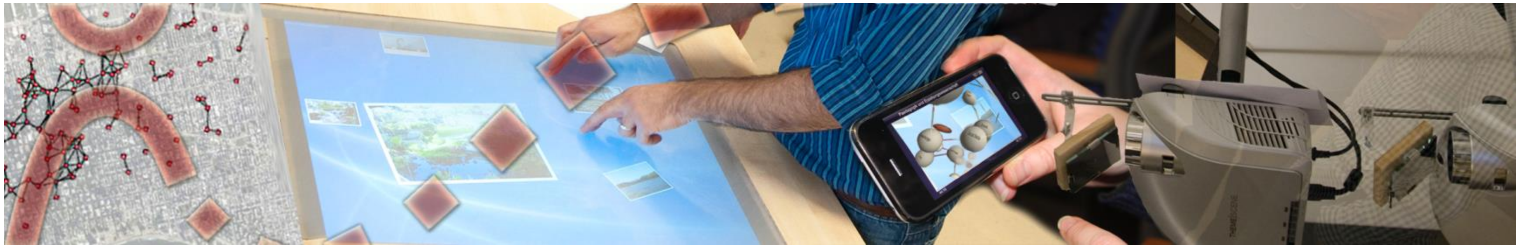
Each node assigns itself the same address!

Results for the $\mathcal{LOCAL}$ Model

- Many papers investigate usage of the model in the context of graph-theoretical problems

- **Possibility results regarding this model are rather pessimistic: Many problems can neither be solved exactly nor approximated in the $LOCAL$ model in constant time ☹**

- **A comprehensive collection of results is given in the survey by Suomela [2013]**



OUTLINE

- Motivation
- Local Algorithms in Theoretical Context
- **Theory Meets Practice…**
- Summary

Theory Meets Practice

- How do the $LOCAL$ model and corresponding results map to the real world?

- **Problems:**
  - The $LOCAL$ model is a synchronized round-based model. However, communication in practice is carried out asynchronously!
  - Nodes do not have access to information that may be available in practice (environment-specific knowledge, locally observable message flow, etc.)
  - The $LOCAL$ model terminates after k rounds. What about algorithms that do not terminate at all?

- So… what is a local algorithm in practice? And how can we check the practical relevance of the theoretical results?

→ Just take a look at practical algorithms that claim to be local!

Applications of Local Algorithms

- In the following application domains, there do exist algorithms that are considered to be local in corresponding publications
  - Addressing
  - Clustering
  - Connectivity
  - Load Balancing
  - Localization
  - Routing
  - Service Discovery
  - Service Placement
  - Sleep Control
  - Video Streaming
  - Graph-Theoretical Problems
  - Topology Control
  - Topology Mismatching

Next, we will give an exemplary insight into two of these domains

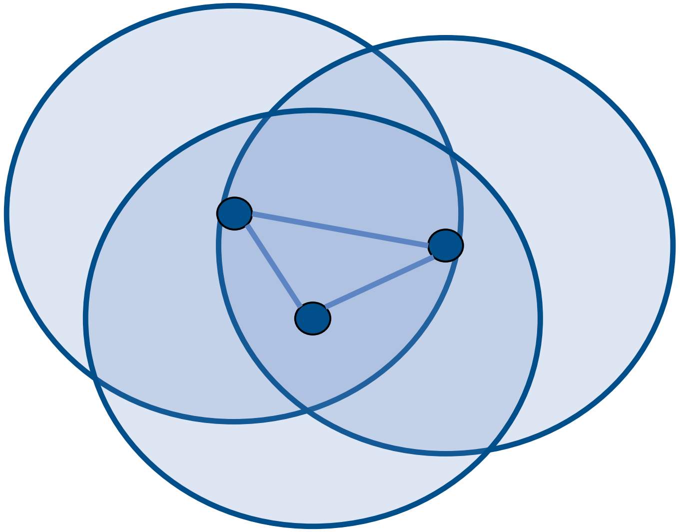
Excursus on Wireless Sensor Networks

- Large networks of small, strictly hardware-limited devices

  (e.g., only a few kBytes of RAM…)
- The devices are battery-powered

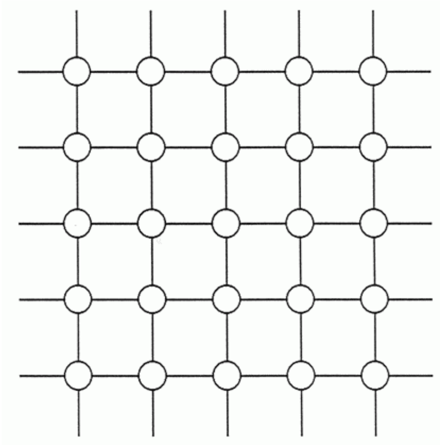
- Wireless communication interface with maximum transmission range
- The nodes aim at sensing their environment and send the data to a central base station

→ How to do routing?



Greedy Routing

- Greedy forwarding
  - Each node knows its own coordinates
  - Source knows coordinates of destination



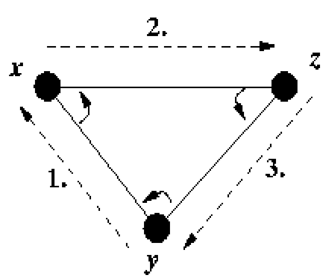
- Local decision is very easy
  - Take the „closest“ neighbor
  - Close might be defined different to enhance the algorithm

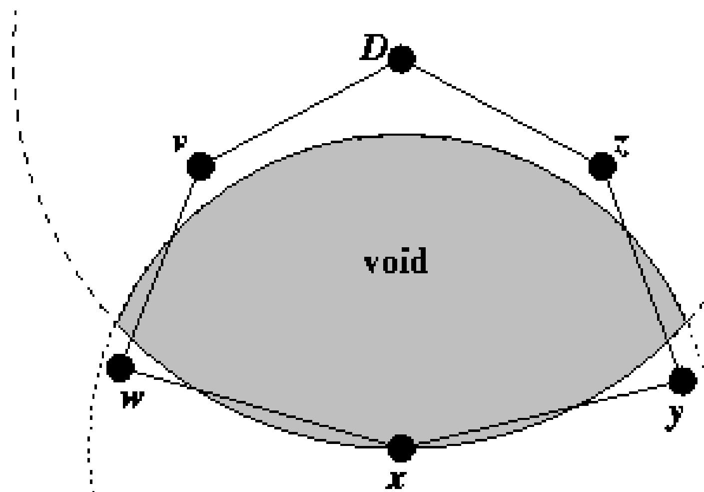
- Works well on some networks, e.g., grid networks

Greedy Perimeter Stateless Routing
[Karp and Kung 2000]

- Doesn‘t work with voids in the network
  - Use greedy when possible, change to right hand rule otherwise

- Right-hand rule
  - When arriving at a node x from node y, the next edge traversed is the next one sequentially counterclockwise from the edge (x,y)

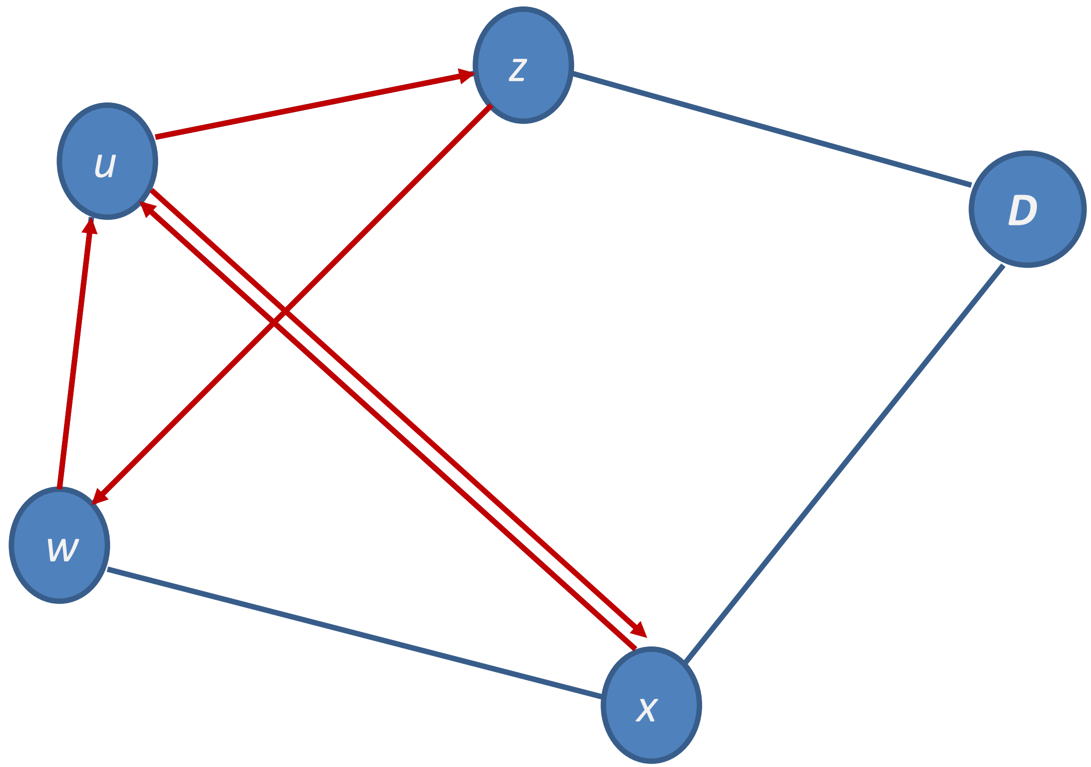




Figures on this slide from [Karp and Kung 2000]

Greedy Perimeter Stateless Routing
[Karp and Kung 2000]

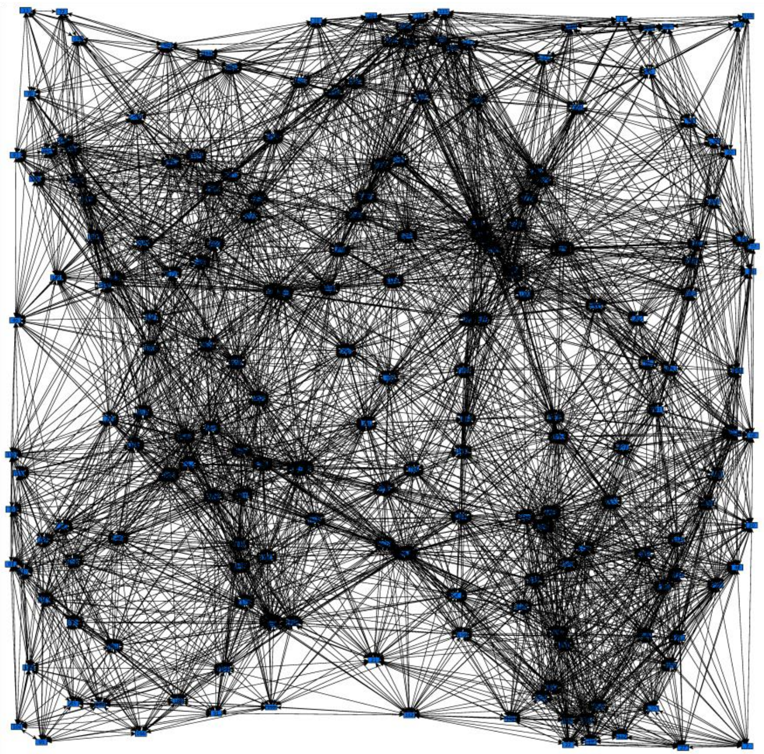
- Works very well in random graphs
  - But: No edges are allowed to cross, otherwise it runs into a loop

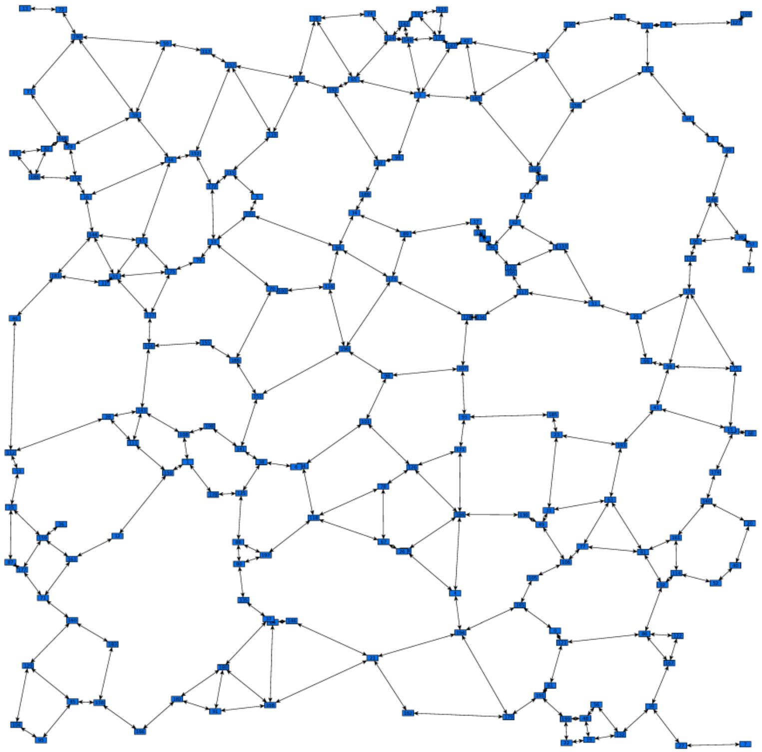


- Start with x to u
  - x-u-z-w-u-x
- Solutions:
  - No crossing heuristic
    - Remove an edge that is encountered again
    - Might partition the network
  - Planar graphs
    - No crossings allowed
    - Topology Control!

This example is based on a slide set by Karp and Kung

Topology Control





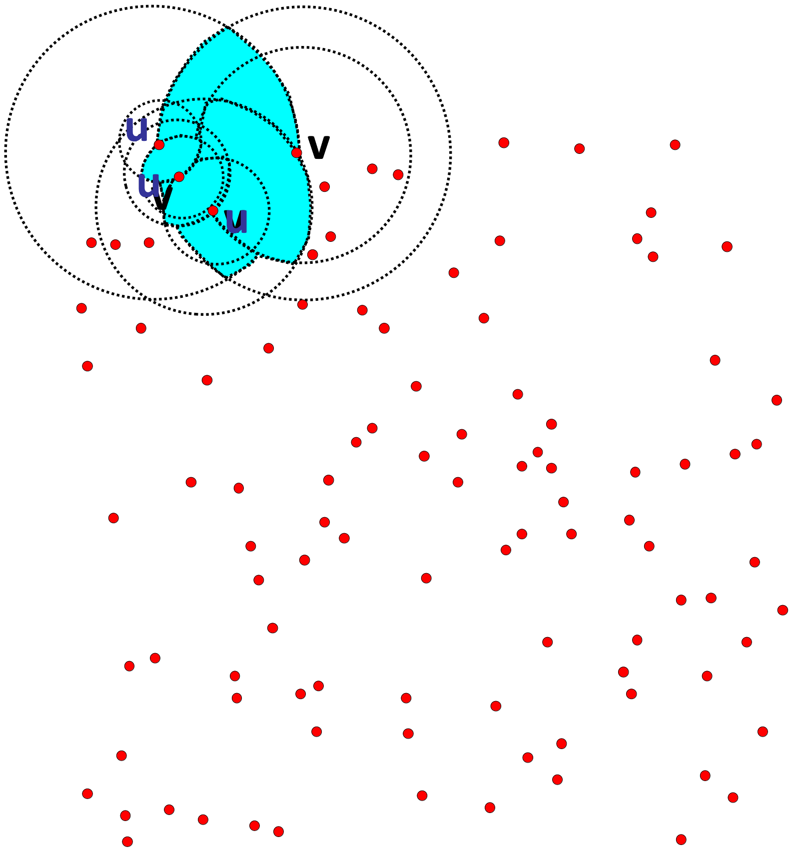
Topology Control

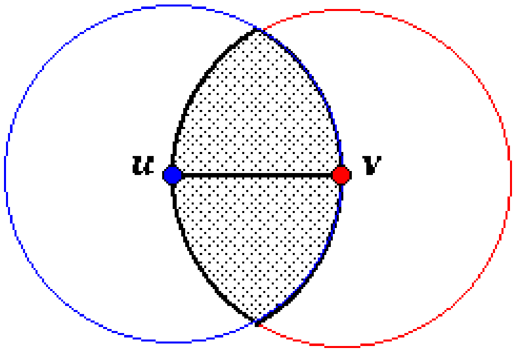
- Make the topology „better“ without:
  - Partitioning the network
  - Making the paths much longer

- What does „better“ mean
  - Planarity as , for example, required by Greedy Perimeter Stateless Routing
  - Less logical neighbors
  - Allows for shrinking the transmission power of nodes
  - Less physical neighbors
    - Less interference

Relative Neighborhood Graph (RNG)

- Edge (u, v) exists if the intersection of the disks centered at u and v is free of other nodes.

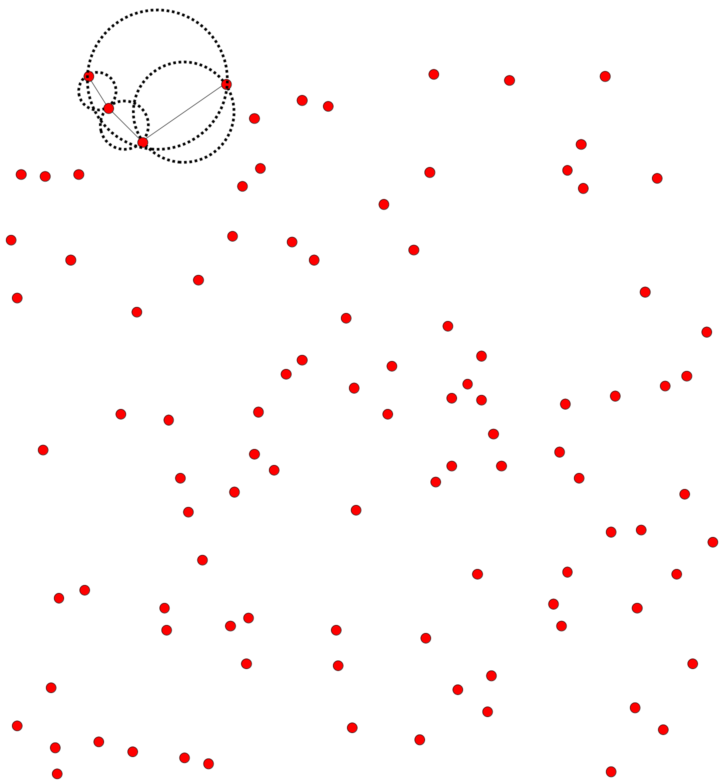


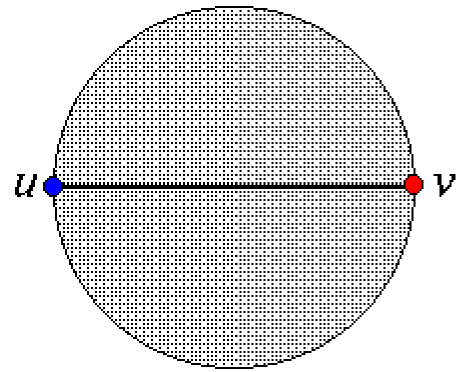


Source: Mirela Damian

Gabriel Graph (GG)

- Edge (u,v) exists iff disk(u,v) is free of other nodes





Source: Mirela Damian

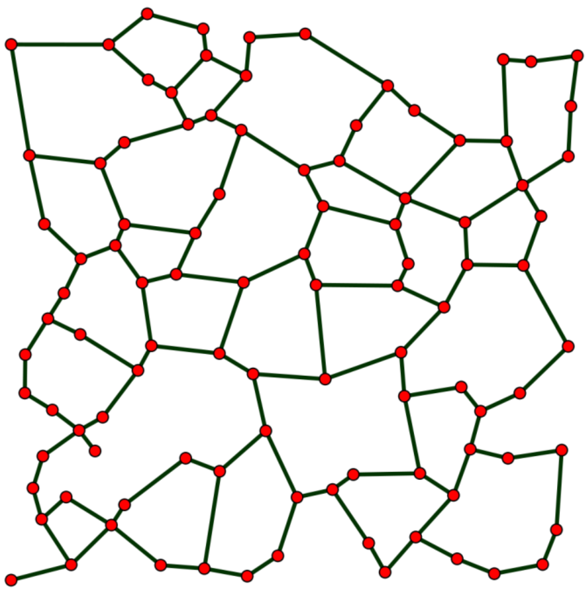
Topology Control

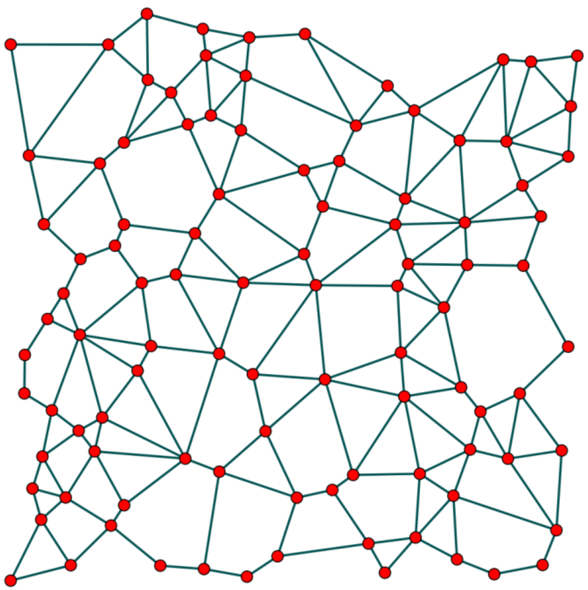
- Easy to calculate locally
  - Only 1-hop information about position needed

- The resulting graph is planar and sparse
  - $|E| \leq 3\ |V|$

- RNG $\subset$ GG

- Also: connected





Relation of These Algorithms to the LOCAL Model?

- Intuitively, these algorithms are local
  - GPSR selects the next hop to the target coordinates based on the coordinates of the 1-hop neighbors
  - For GG and RNG, the edge removal decision depends on node coordinates in the 1-hop neighborhood

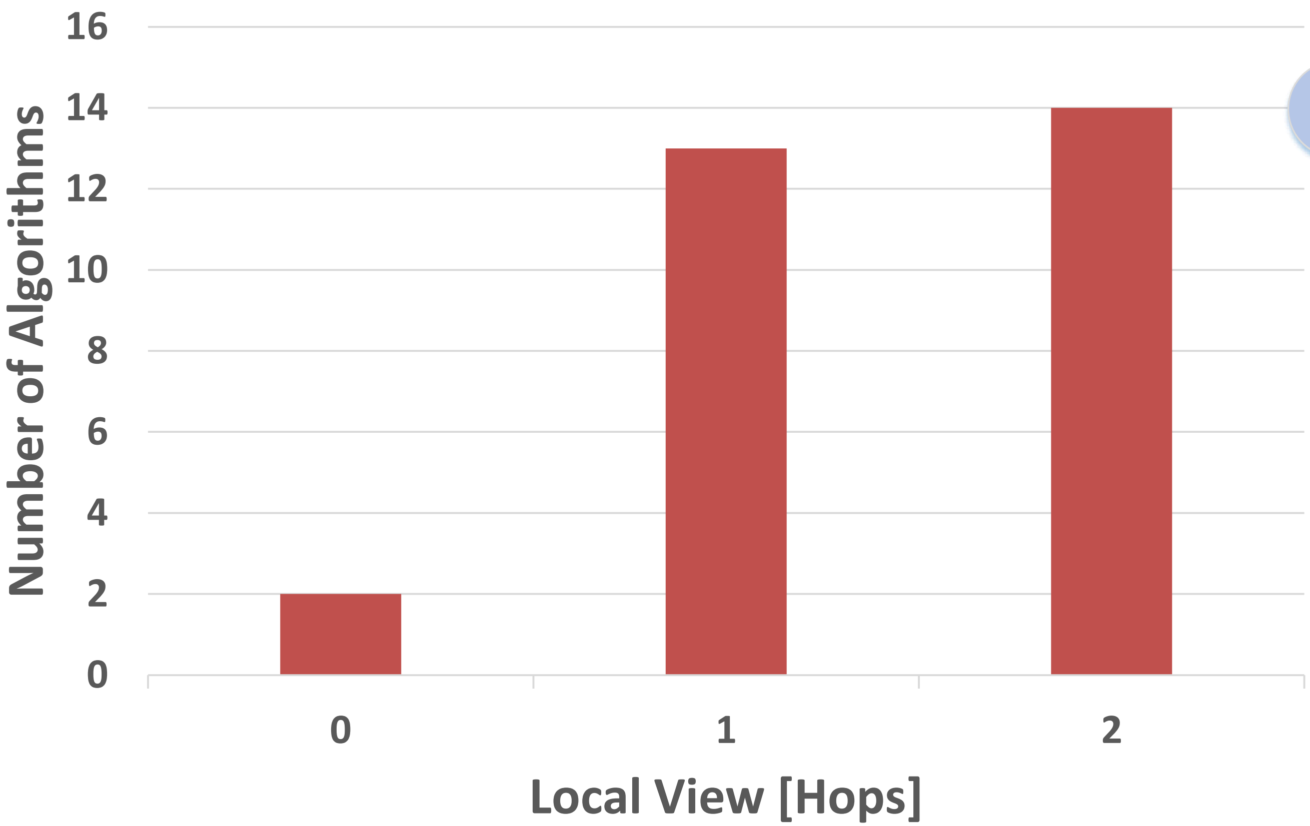
- What about the LOCAL model
  - **GPSR is absolutely non-local according to the LOCAL model** because data may be forwarded an arbitrary number of hops through the network
  - GG and RNG may indeed be implemented in the LOCAL model

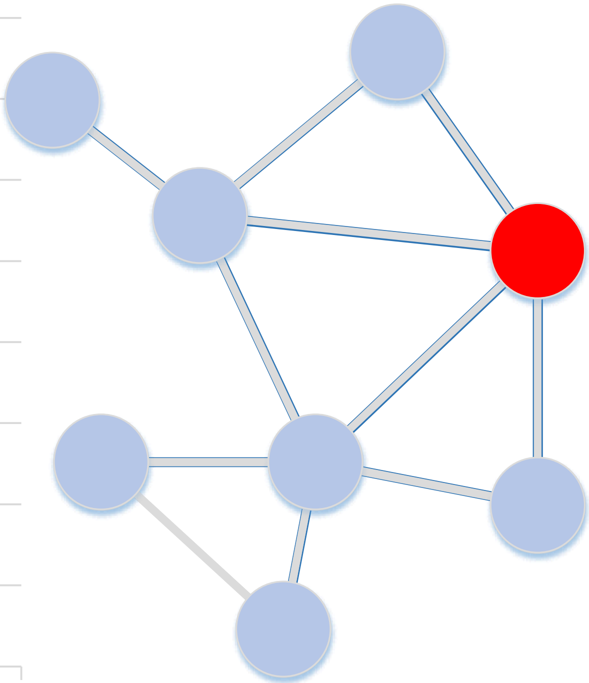
Key Observations

- Many problems that are not solvable in the theoretical model can be **heuristically** tackled in practice by local algorithms!

- **Most importantly: Most of the investigated algorithms (e.g., GPSR) are not covered by the $\mathcal{LOCAL}$ model!**

Another Key Observation: Size of the „Local View“



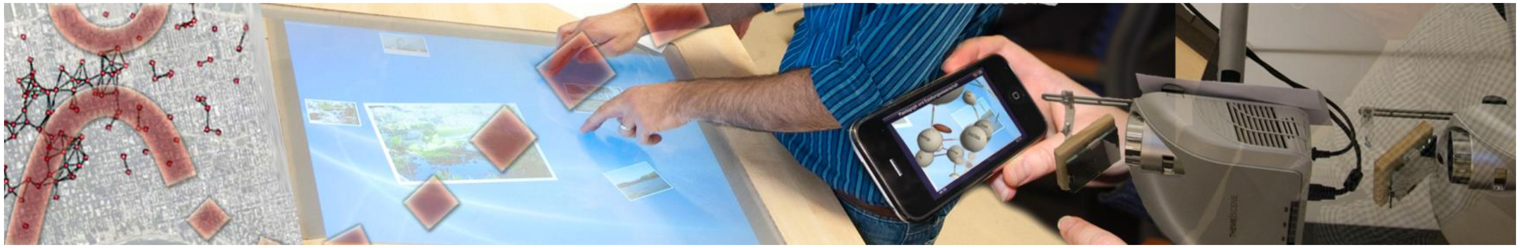


A local view of only two hops is sufficient to tackle many problems!

How to Define Locality in Practice?

- The algorithms claiming to be local differ from each other with respect to their degree of locality
- Hence, it is not meaningful to give a broad definition that covers all of them
- Open research question: Is it possible to define classes of locality?

Strictly local algorithms (hard requirements)	Distributed algorithms (low requirements)



OUTLINE

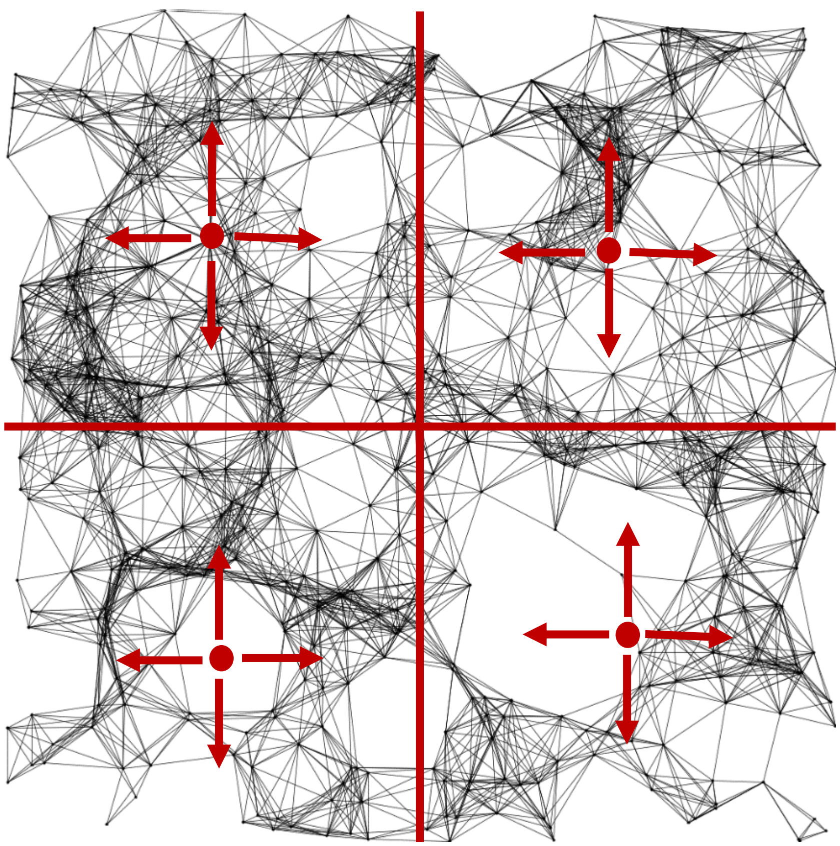
- Motivation
- Local Algorithms in Theoretical Context
- Theory Meets Practice…
- **Summary**

Summary and Outlook

- What is a local algorithm?
- Well, that depends on the chosen perspective…
- Local algorithms in theory…
- … and in practice

Outlook: Local/Global Trade-off

- Obviously, there is a trade-off between the performance of an algorithm and its degree of locality
- How can this trade-off be controlled?
  - Increasing k
  - Divide the network into regions and apply centralized algorithms?



References

- Pierre Fraigniaud, Mika Gs, Amos Korman, and Jukka Suomela. 2013. What Can Be Decided Locally Without Identifiers?. In Proceedings of the Symposium on Principles of Distributed Computing (PODC). 157–165.
- Lachezar Krumov, Adriana Andreeva, and Thorsten Strufe. 2010a. Resilient Peer-to-Peer Live-Streaming using Motifs. In Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM). 1–8.
- Lachezar Krumov, Immanuel Schweizer, Dirk Bradler, and Thorsten Strufe. 2010b. Leveraging Network Motifs for the Adaptation of Structured Peer-to-Peer-Networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM). 1–5.
- G. Kulcsár, M. Stein, I. Schweizer, G. Varró, M. Mühlhäuser, A. Schürr. 2014. *Rapid Prototyping of Topology Control Algorithms by Graph Transformation*. In Proceedings of the International Workshop on Graph-Based Tools (GraBaTs).
- N. Linial. Locality in Distributed Graph Algorithms. SIAM Journal on Computing, 21(1)(1):193{201, February 1992.
- David Peleg. 2000. Distributed Computing: A Locality-Sensitive Approach. Monographs on Discrete Mathematics and Applications, Vol. 5. Society for Industrial and Applied Mathematics.
- Immanuel Schweizer, Michael Wagner, Dirk Bradler, Max Mühlhäuser, and Thorsten Strufe. 2012. kTC - Robust and Adaptive Wireless Ad-hoc Topology Control. In Proceedings of the International Conference on Computer Communication Networks (ICCCN). 1–9.
- Jukka Suomela. 2013. Survey of Local Algorithms. Comput. Surveys 45, 2 (2013), 24:1–24:40.
- Brad Karp and H. T. Kung. 2000. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom). 243–254.