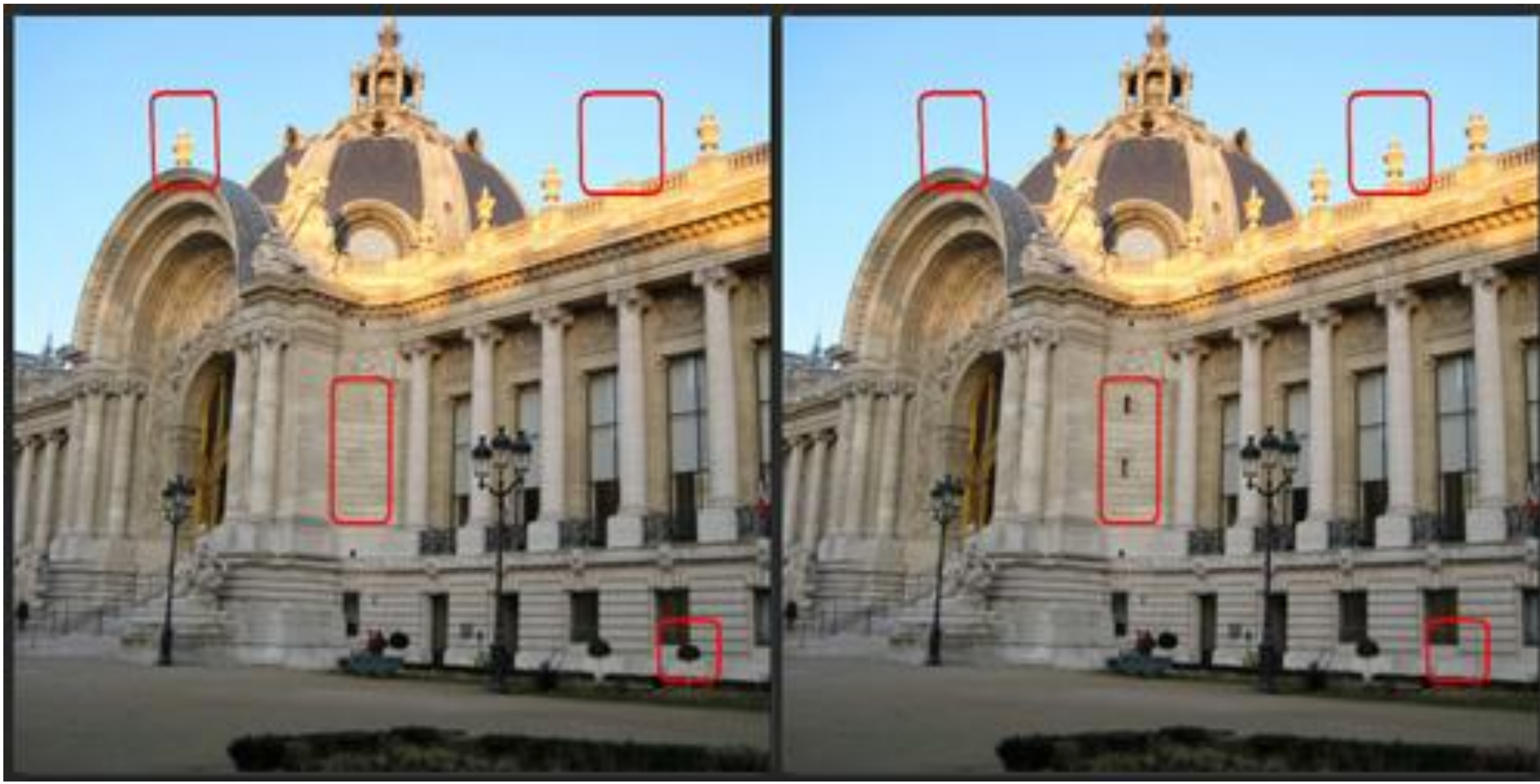
Physical Algorithms

Spot the Differences



Spot the Differences

Too Many!

Spot the Differences

Still Many!

Spot the Differences

Better Screen

Bigger Disk

More RAM

Cooler Design

…

Better Screen

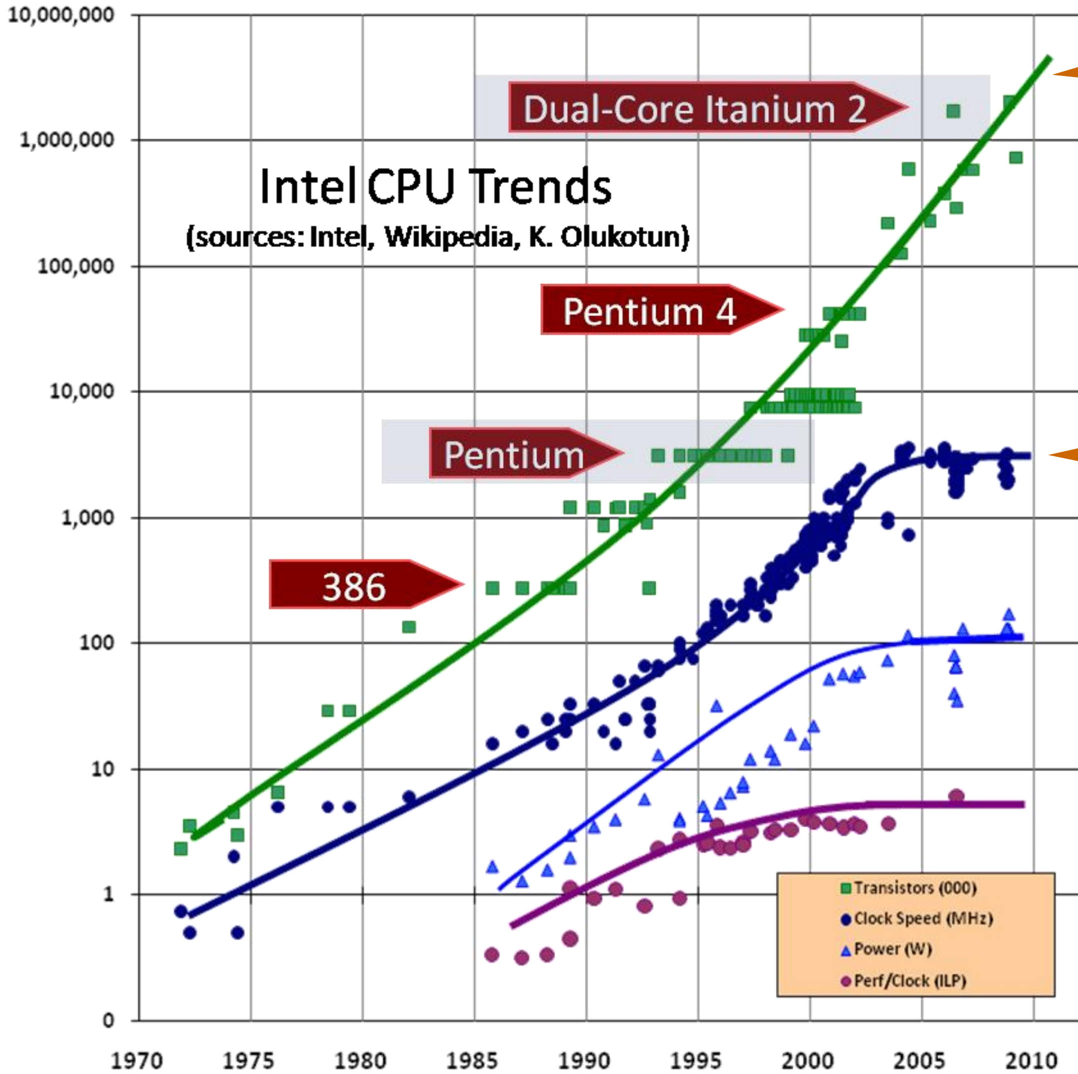
Bigger Disk

More RAM

Cooler Design

...

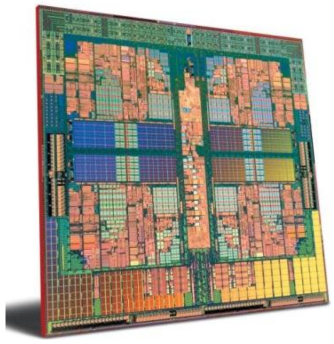
Same CPU Clock Speed



Transistor count still rising

Clock speed flattening sharply

Advent of multi-core processors!

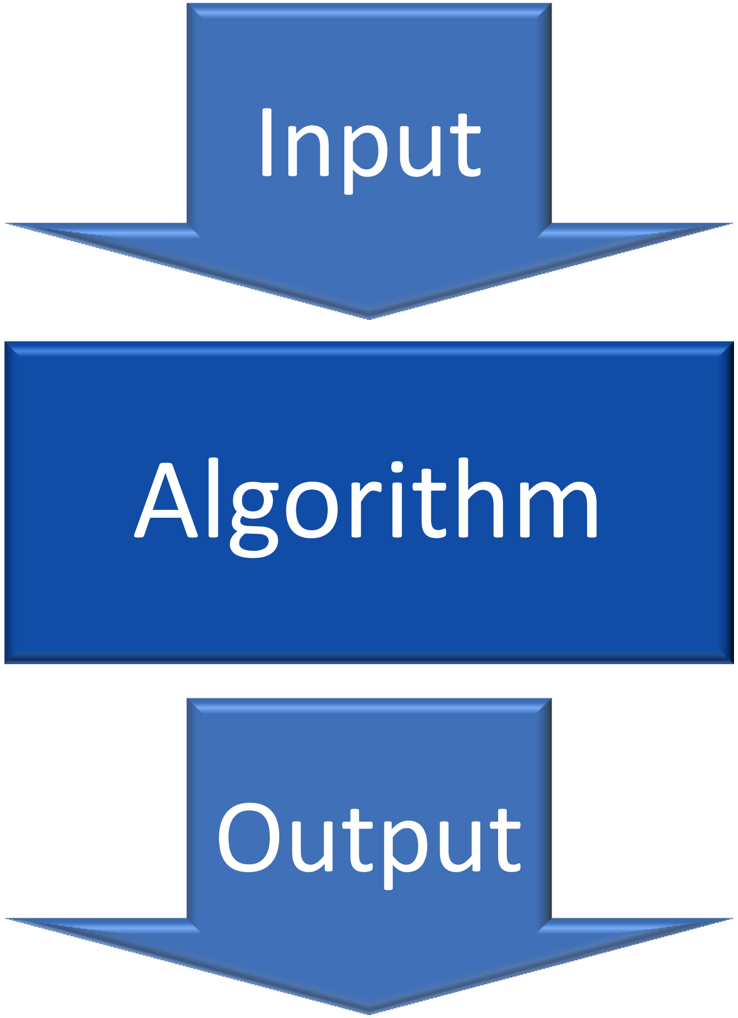


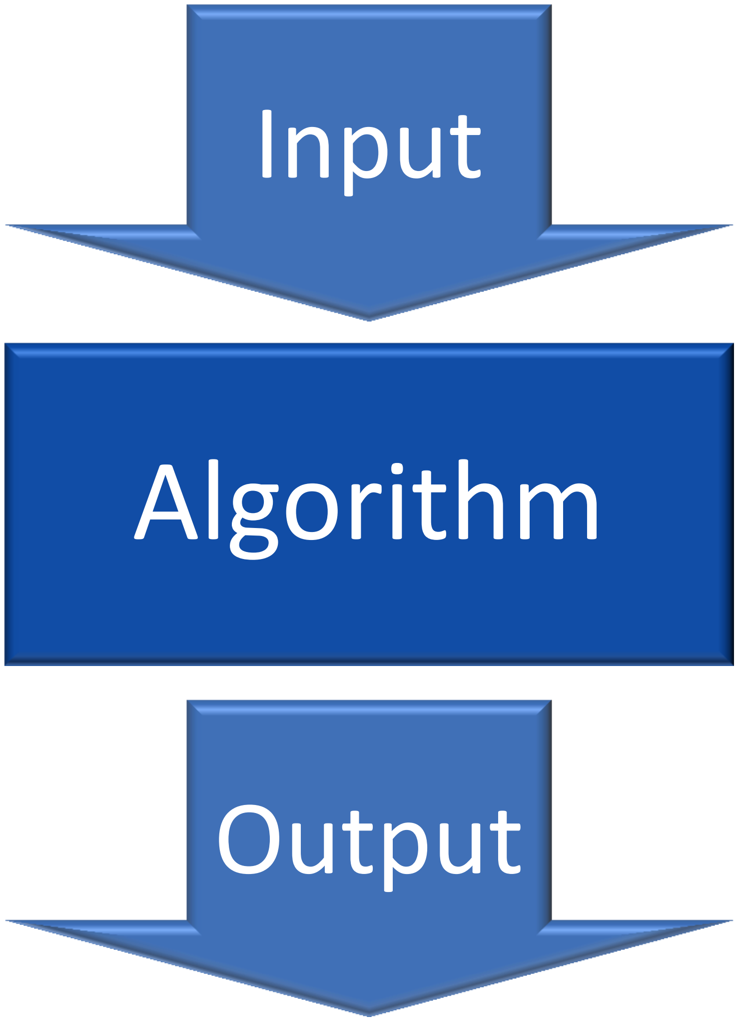
Why Should I Care?

Computer Science → Washing Machine Science

[Roger Boyle, Maurice Herlihy]

Algorithms

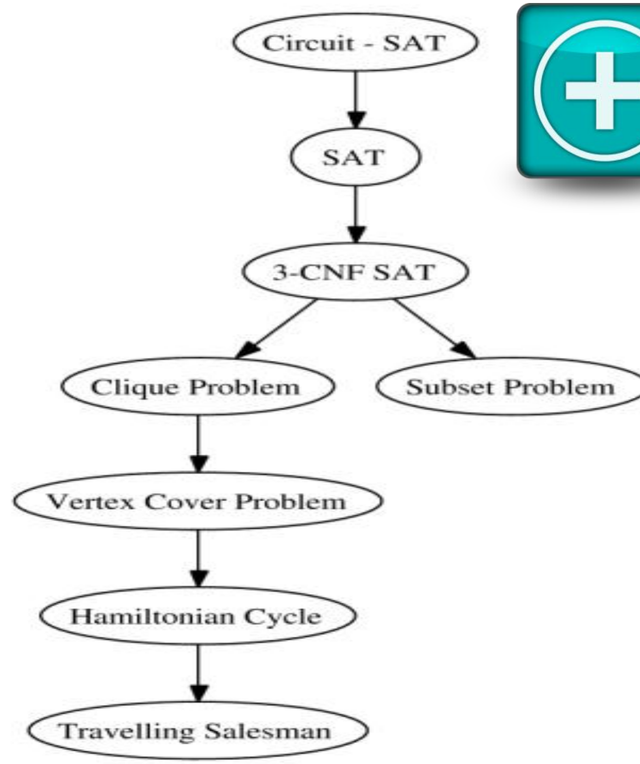

Input
Algorithm
Output



simple and robust model

comparable results

complexity theory

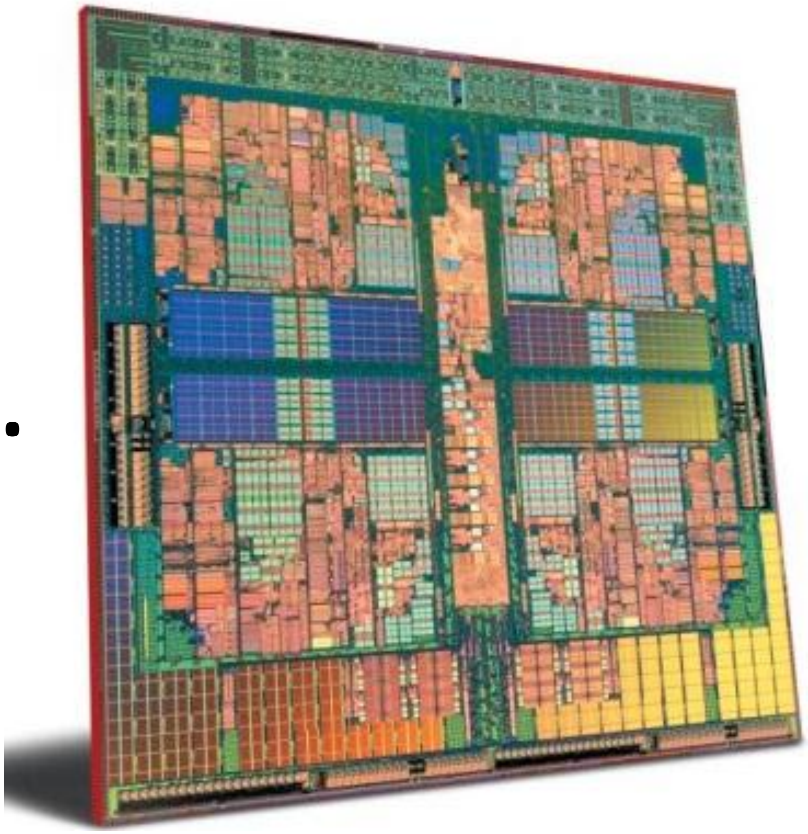


Input

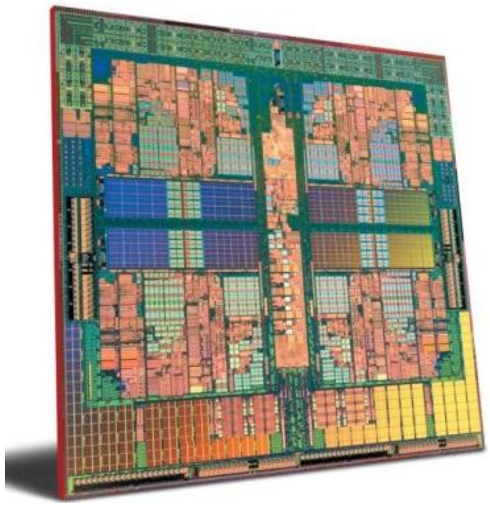
Algorithm

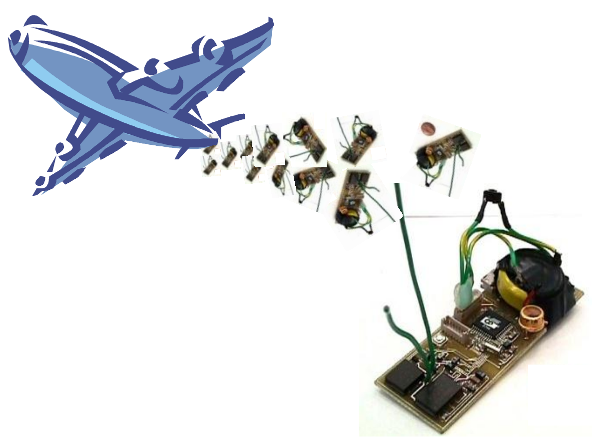
Output

vs.



The Future of Computing?





Talk Overview

Introduction & Motivation

Examples for Physical Algorithms
in the Context of Sensor Networks

What are Physical Algorithms?

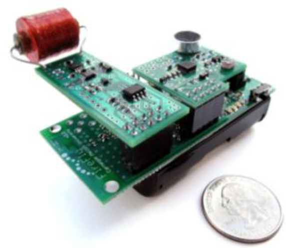
Clock Synchronization

Clock Synchronization in Networks

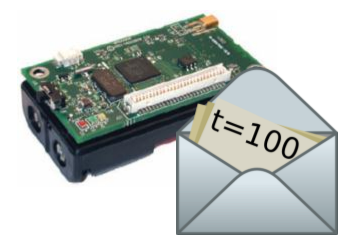
Global Positioning System (GPS)

Radio Clock Signal

AC-power line radiation



Synchronization messages

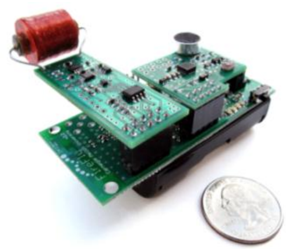


Clock Synchronization in Networks

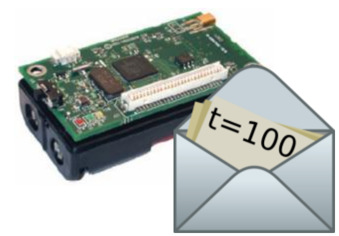
Global Positioning System (GPS)

Radio Clock Signal

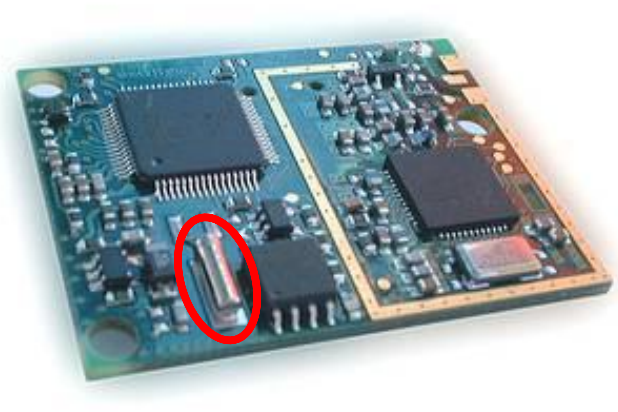
AC-power line radiation



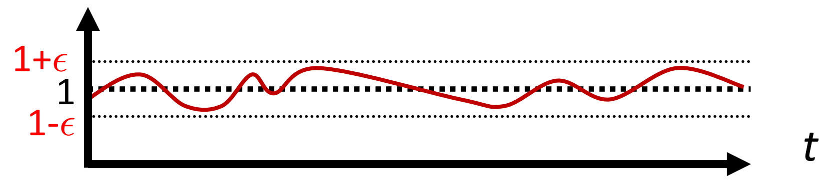
Synchronization messages

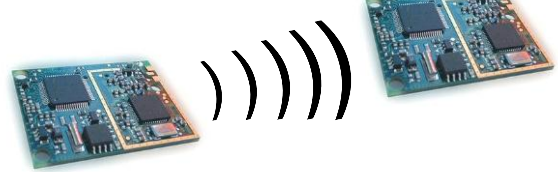


Problem: Physical Reality

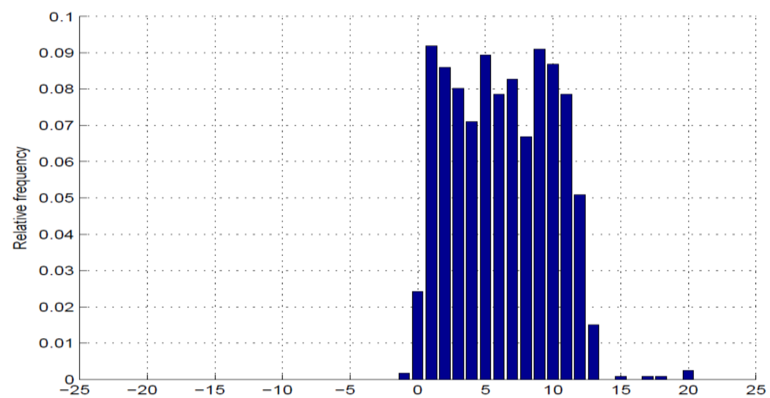


clock rate





message delay

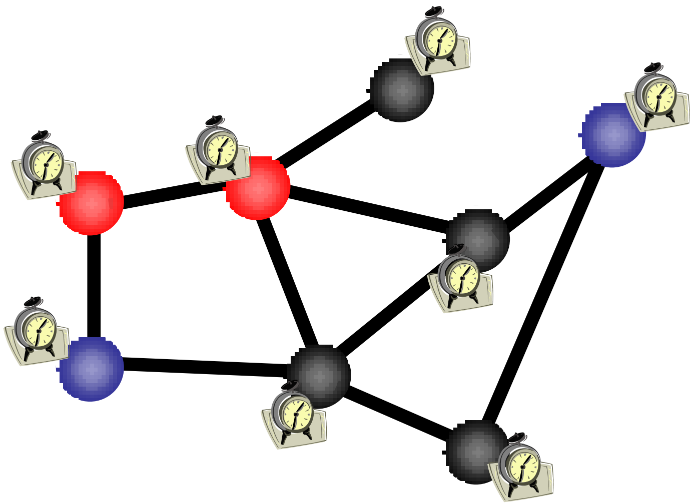


Clock Synchronization in Theory?

Given a communication network

1. Each node equipped with hardware clock with drift
2. Message delays with jitter

worst-case (but constant)



Goal: Synchronize Clocks ("Logical Clocks")

- Both global and local synchronization!

Time Must Behave!

- Time (logical clocks) should not be allowed to stand still or jump

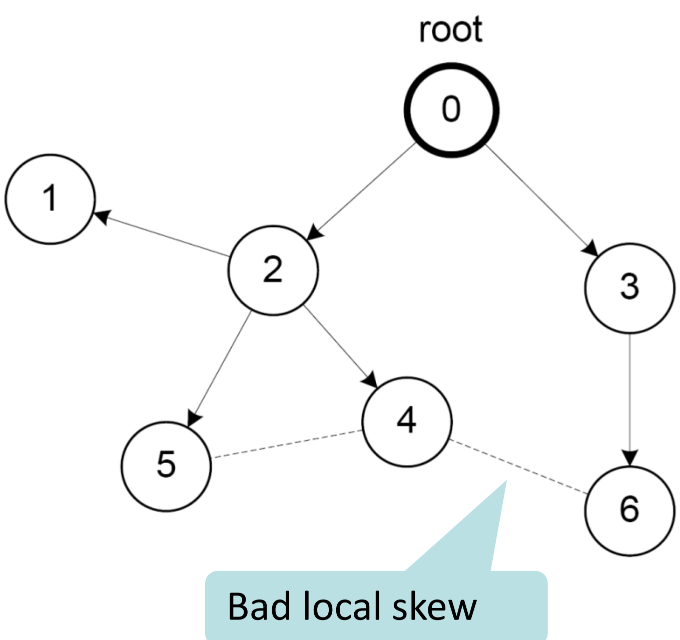
Time Must Behave!

- Time (logical clocks) should not be allowed to stand still or jump

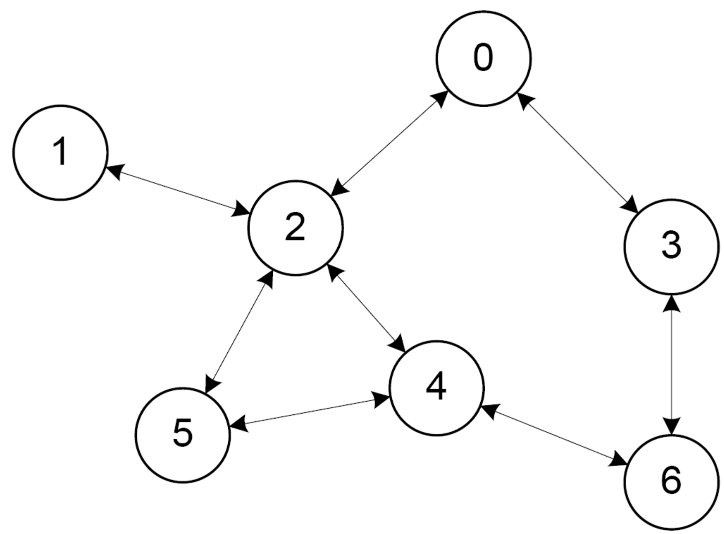
- Let’s be more careful (and ambitious):
- Logical clocks should always move forward
  - Sometimes faster, sometimes slower is OK.
  - But there should be a minimum and a maximum speed.
  - As close to correct time as possible!

Local Skew

Tree-based Algorithms
e.g. FTSP



Neighborhood Algorithms
e.g. GTSP

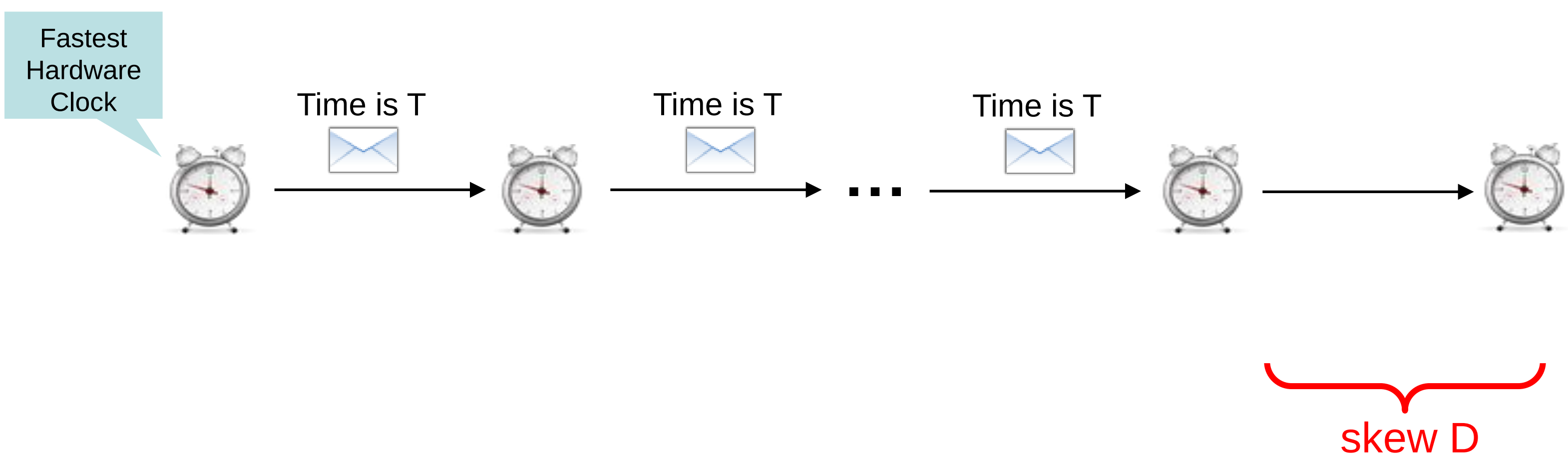


Synchronization Algorithms: An Example ("A^{max}")

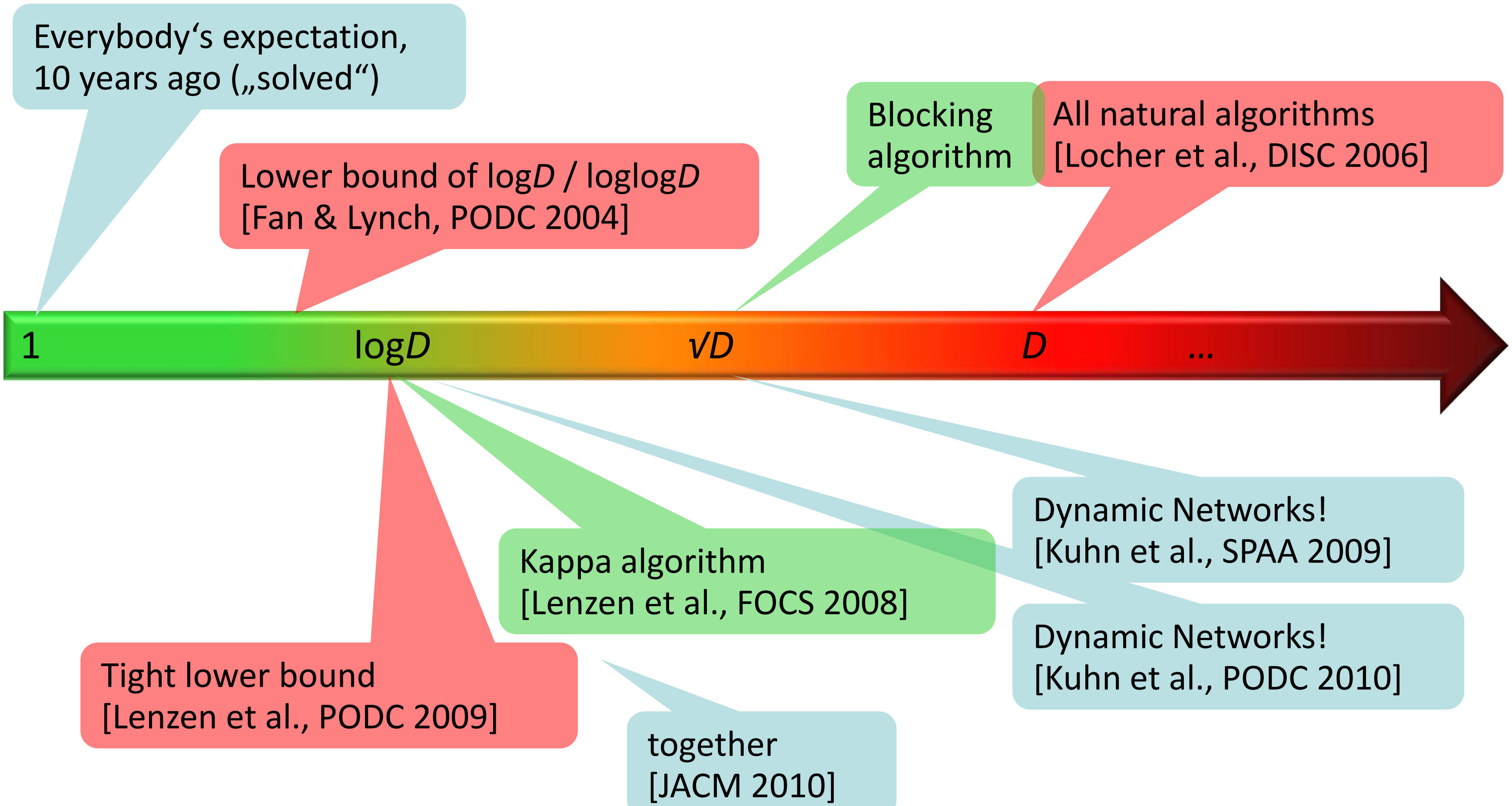
- Question: How to update the logical clock based on the messages from the neighbors?

- Idea: Minimizing the skew to the fastest neighbor
  - Set clock to maximum clock value you know, forward new values immediately

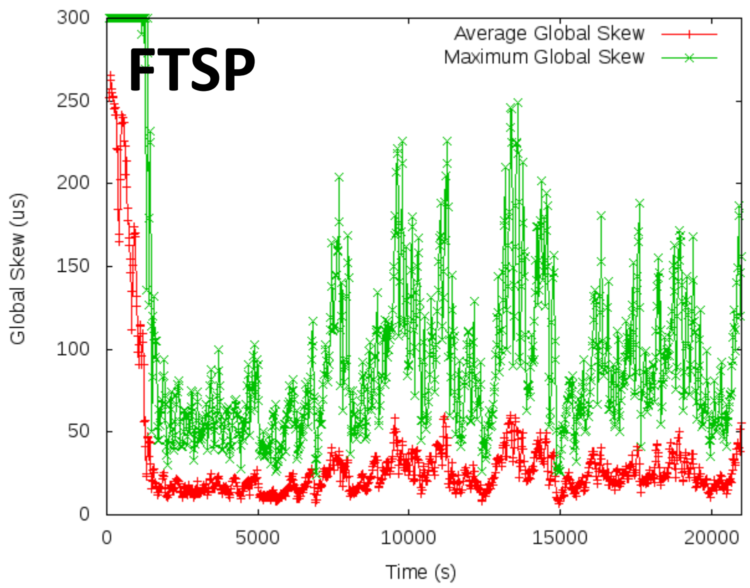
- First all messages are slow (1), then suddenly all messages are fast (0)!

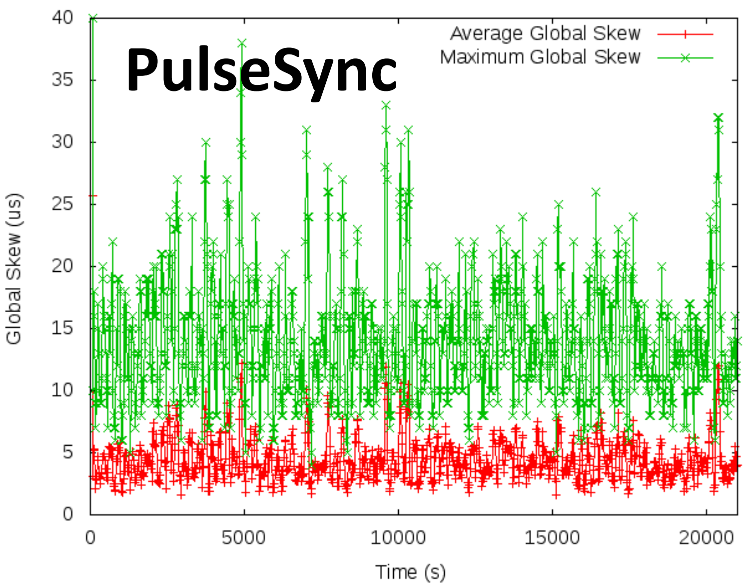


Local Skew: Overview of Results



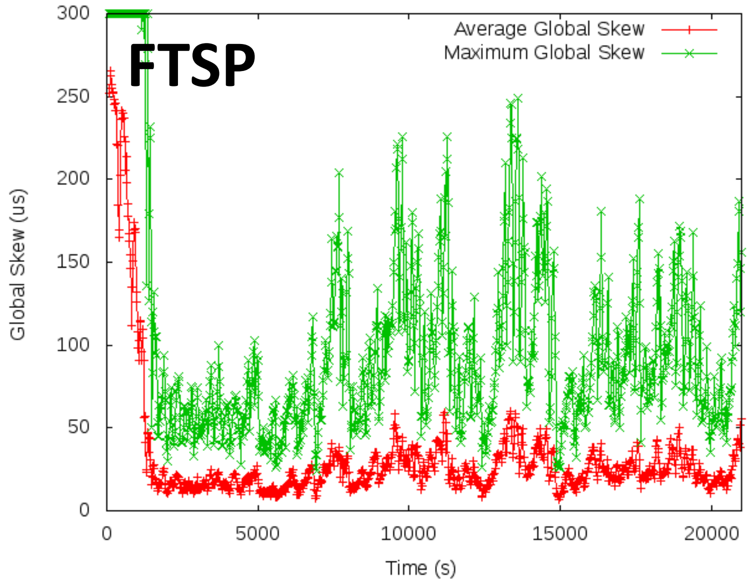
Experimental Results for Global Skew

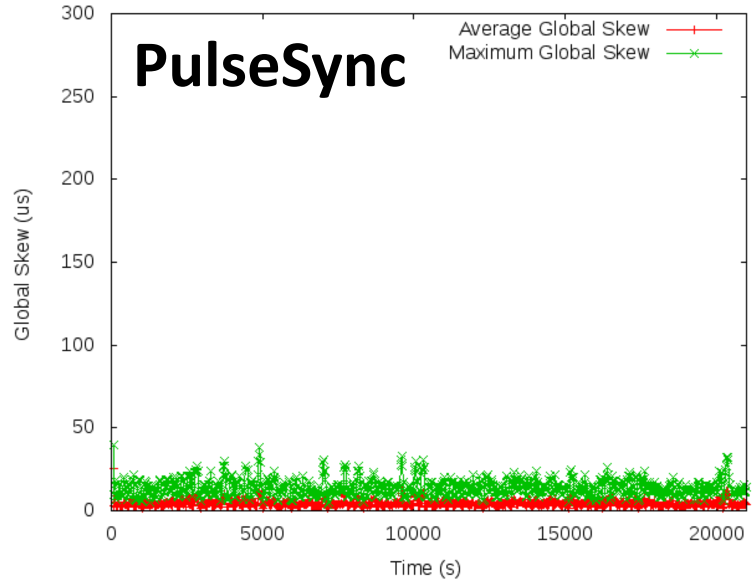




[Lenzen, Sommer, W, SenSys 2009]

Experimental Results for Global Skew





[Lenzen, Sommer, W, SenSys 2009]

Clock Synchronization vs. Car Coordination

- In the future cars may travel at high speed despite a tiny safety distance, thanks to advanced sensors and communication

Clock Synchronization vs. Car Coordination

- In the future cars may travel at high speed despite a tiny safety distance, thanks to advanced sensors and communication

- How fast & close can you drive?

- Answer possibly related to clock synchronization
  - clock drift ↔ cars cannot control speed perfectly
  - message jitter ↔ sensors or communication between cars not perfect

Wireless Communication

Wireless Communication

EE, Physics

Maxwell Equations

Simulation, Testing

‘Scaling Laws’

Network Algorithms

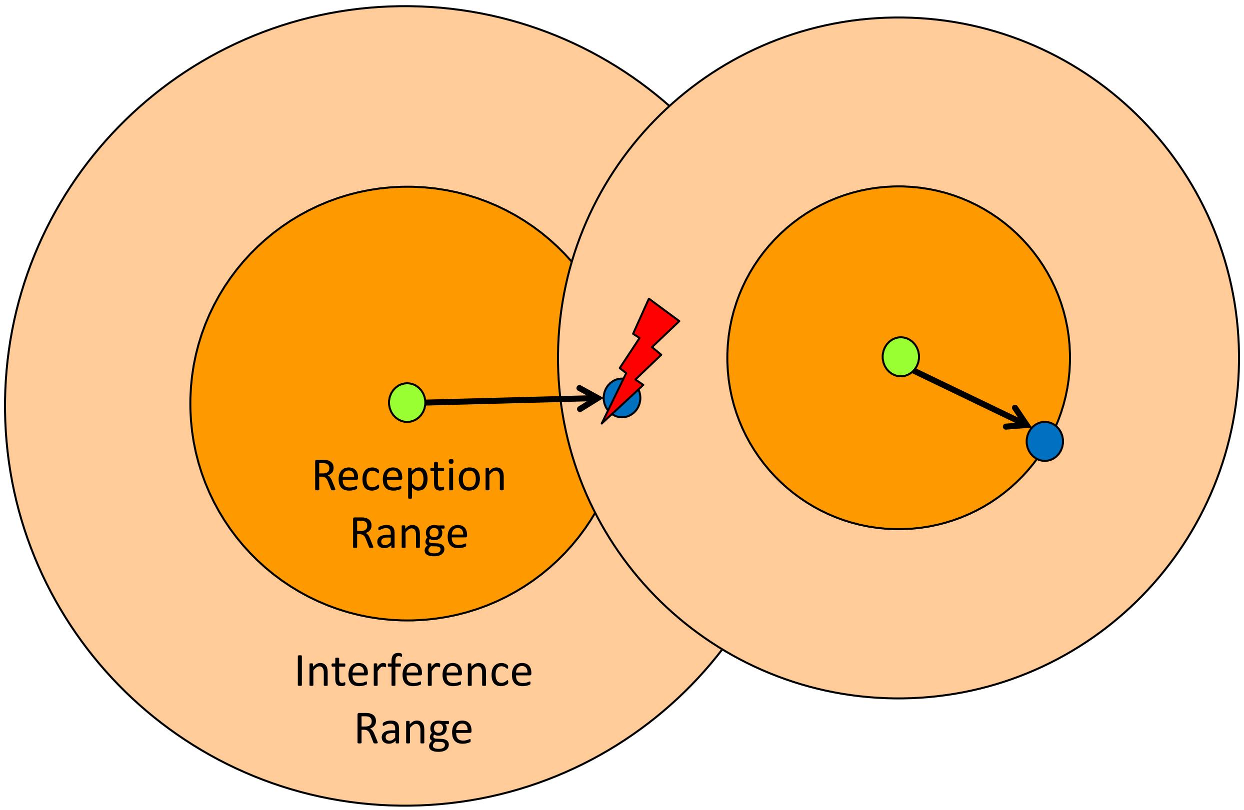
CS, Applied Math

[Geometric] Graphs

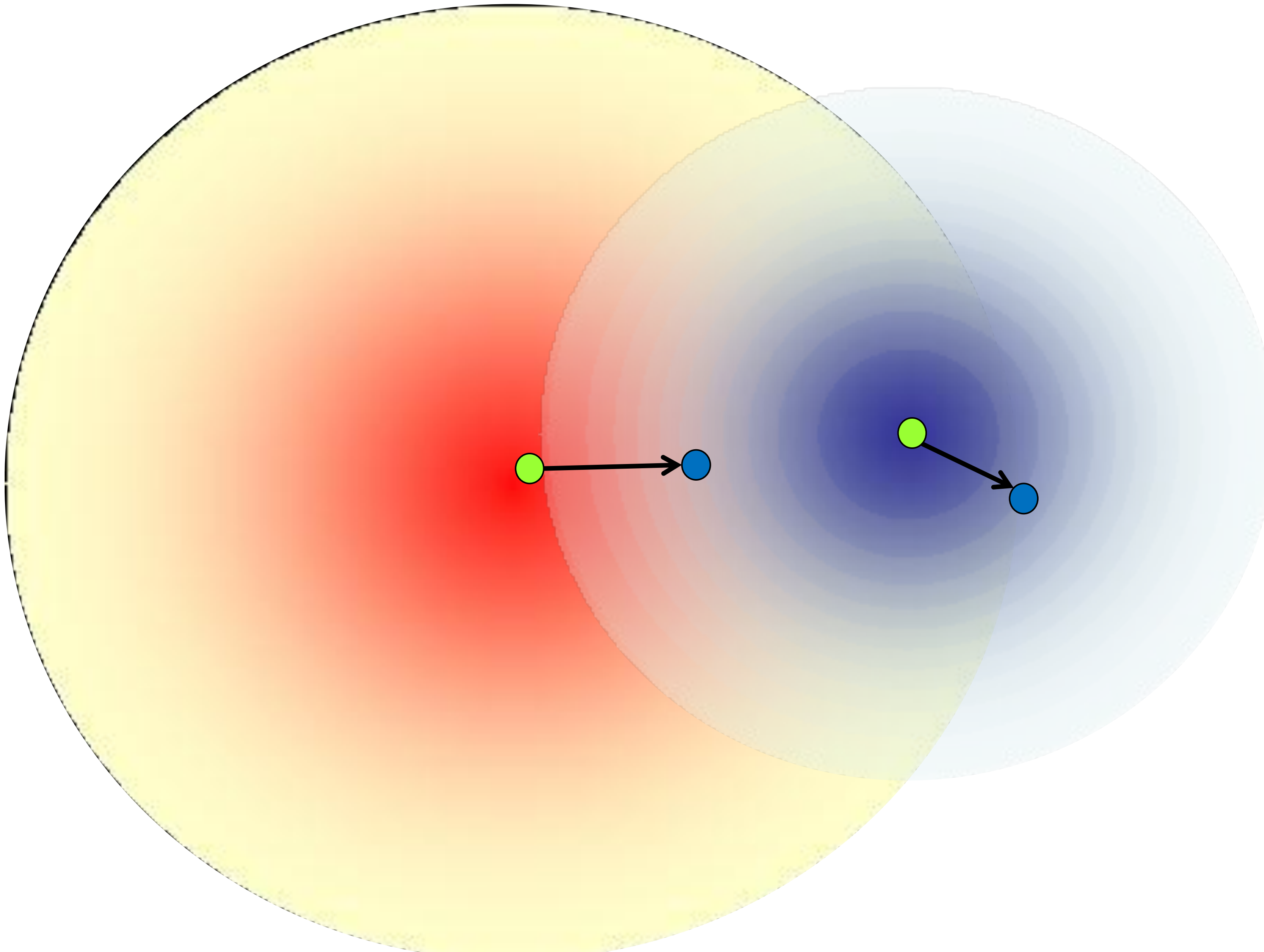
Worst-Case Analysis

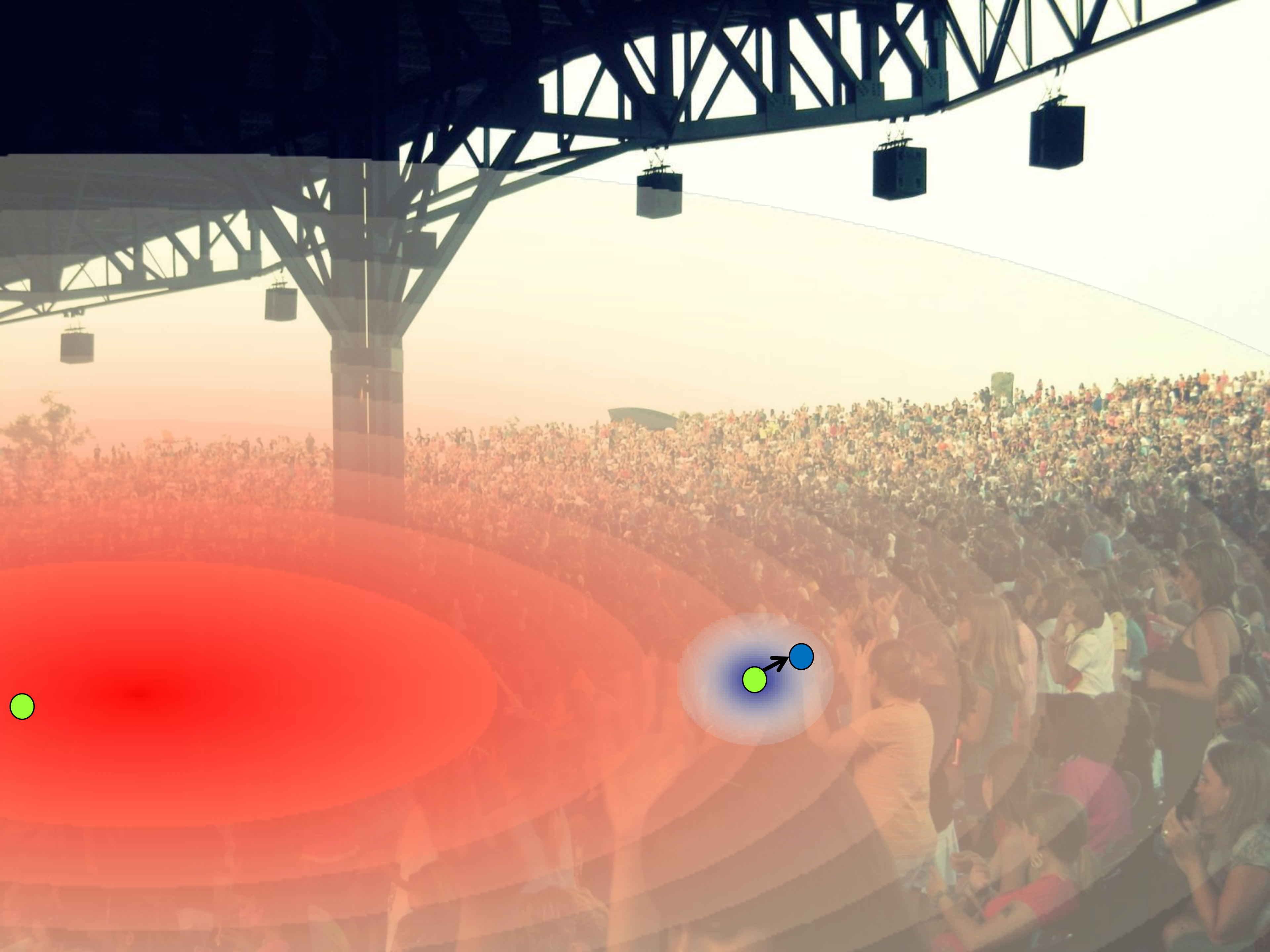
Any-Case Analysis

CS Models: e.g. Disk Model (Protocol Model)



EE Models: e.g. SINR Model (Physical Model)





Signal-To-Interference-Plus-Noise Ratio (SINR) Formula

$$\frac{\frac{P_u}{d(u,v)^{\alpha}}}{N + \sum_{w \in V \setminus \{u\}} \frac{P_w}{d(w,v)^{\alpha}}} \geq \beta$$

Received signal power from sender

Power level of sender *u*

Path-loss exponent

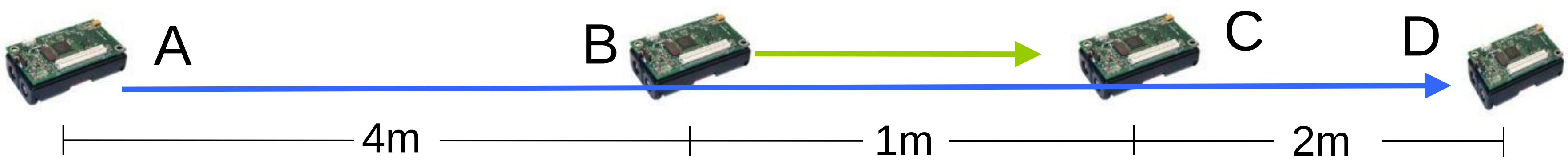
Noise

Minimum signal-to-interference ratio

Received signal power from all other nodes (=interference)

Distance between two nodes

Example: Protocol vs. Physical Model



Assume a single frequency (and no fancy decoding techniques!)

Is spatial reuse possible?

- NO — Protocol Model
- YES — With power control

Let α=3, β=3, and N=10nW

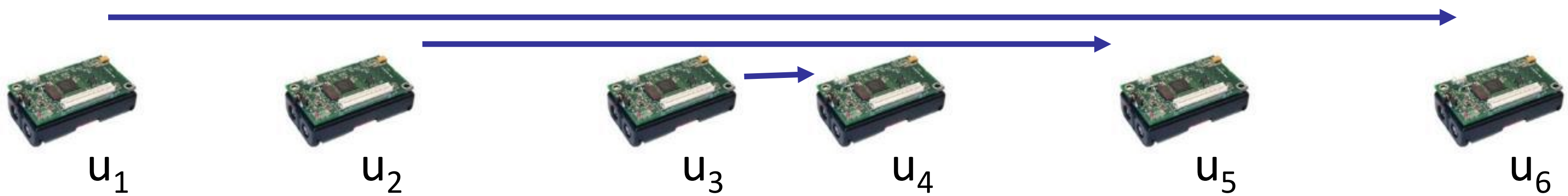
Transmission powers: P_B= -15 dBm and P_A= 1 dBm

SINR of A at D: $$\frac{1.26mW/(7m)^3}{0.01\mu W + 31.6\mu W/(3m)^3} \approx 3.11 \geq \beta$$

SINR of B at C: $$\frac{31.6\mu W/(1m)^3}{0.01\mu W + 1.26mW/(5m)^3} \approx 3.13 \geq \beta$$

This works in practice!

… even with very simple hardware (sensor nodes)



Time for transmitting 20‘000 packets:

	Time required	
	standard MAC	“SINR-MAC”
Node u_1	$721s$	$267s$
Node u_2	$778s$	$268s$
Node u_3	$780s$	$270s$

	Messages received	
	standard MAC	“SINR-MAC”
Node u_4	19999	19773
Node u_5	18784	18488
Node u_6	16519	19498

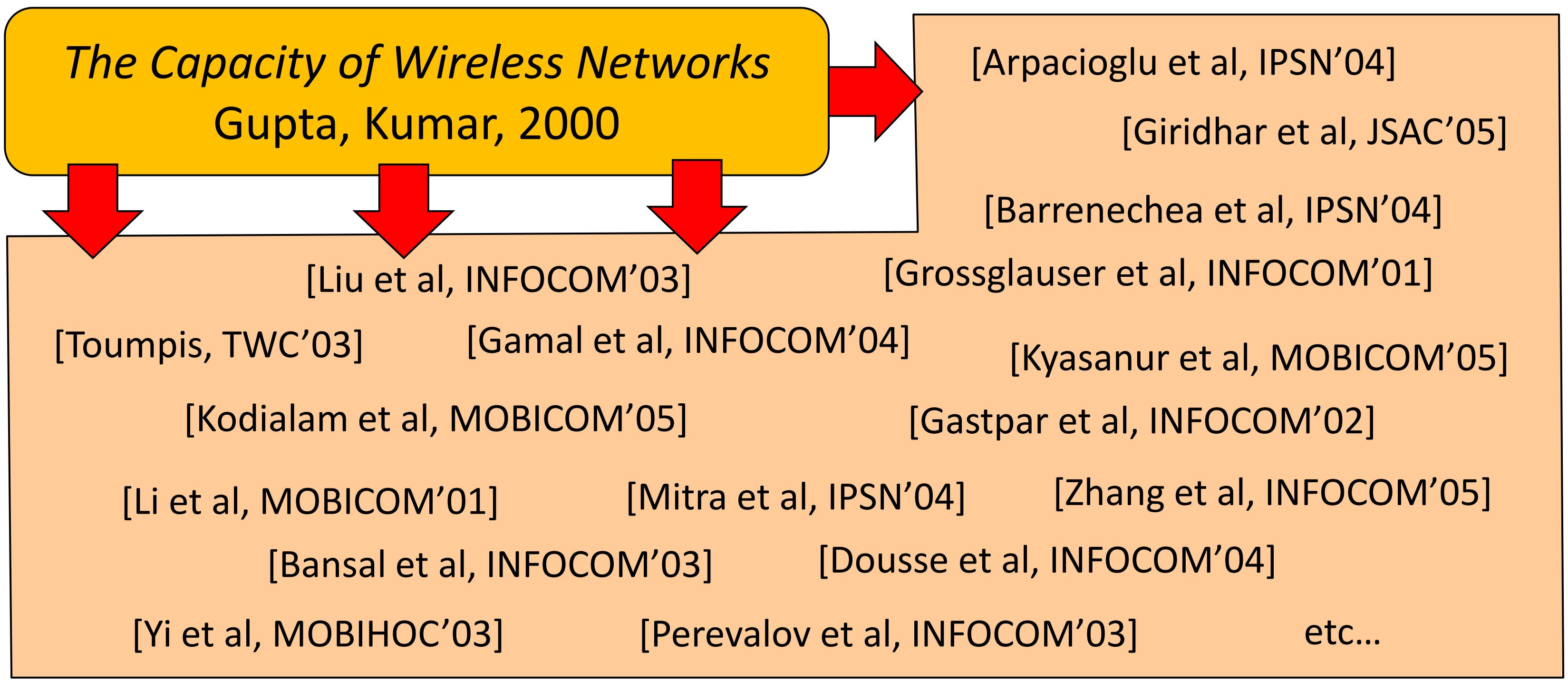
Speed-up is almost a factor 3

[Moscibroda, W, Weber, Hotnets 2006]

The Capacity of a Network

(How many concurrent wireless transmissions can you have)

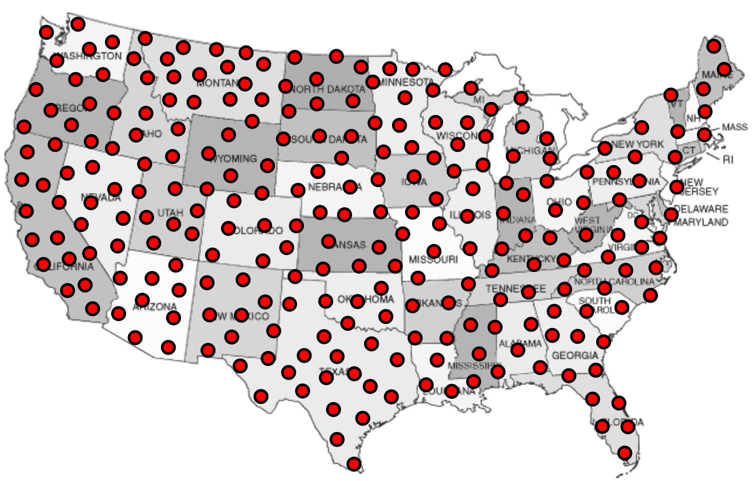
… is a well-studied problem in Wireless Communication

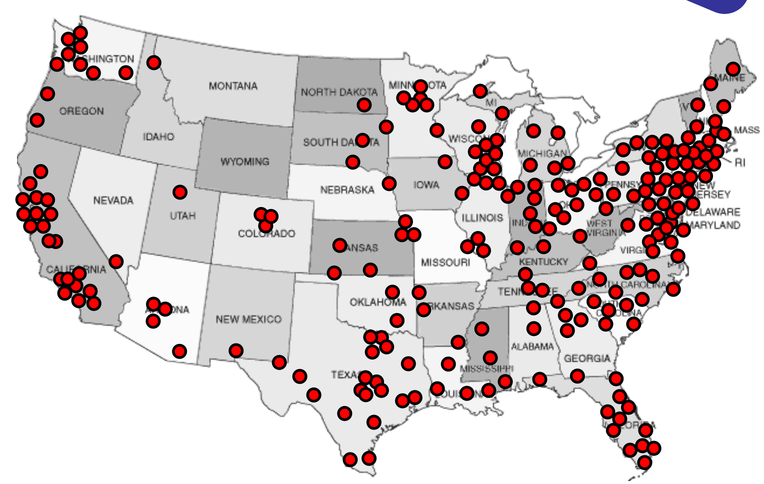


Network Topology?

- All these capacity studies make very strong assumptions on node deployment, topologies
  - randomly, uniformly distributed nodes
  - nodes placed on a grid
  - etc.

What if a network looks differently…?





"Convergecast Capacity" in Wireless Sensor Networks

[Moscibroda, W, 2006]
[Halldorsson, Mitra, 2012]

Worst-Case Capacity

[Giridhar, Kumar, 2005]

Best-Case Capacity

Networks / Model/Power	Max. rate in arbitrary, worst-case deployment	Max. rate in random, uniform deployment
Protocol Model	$\Theta(1/n)$	$\Theta(1/\log n)$
Physical Model (power control)	$\Omega(1/\log n)$	$\Omega(1/\log n)$

Exponential gap between protocol and physical model!

<u>The Price of Worst-Case Node Placement</u>

- Exponential in protocol model
- Polylogarithmic in physical model (almost no worst-case penalty!)

Physical Algorithms

Real Capacity

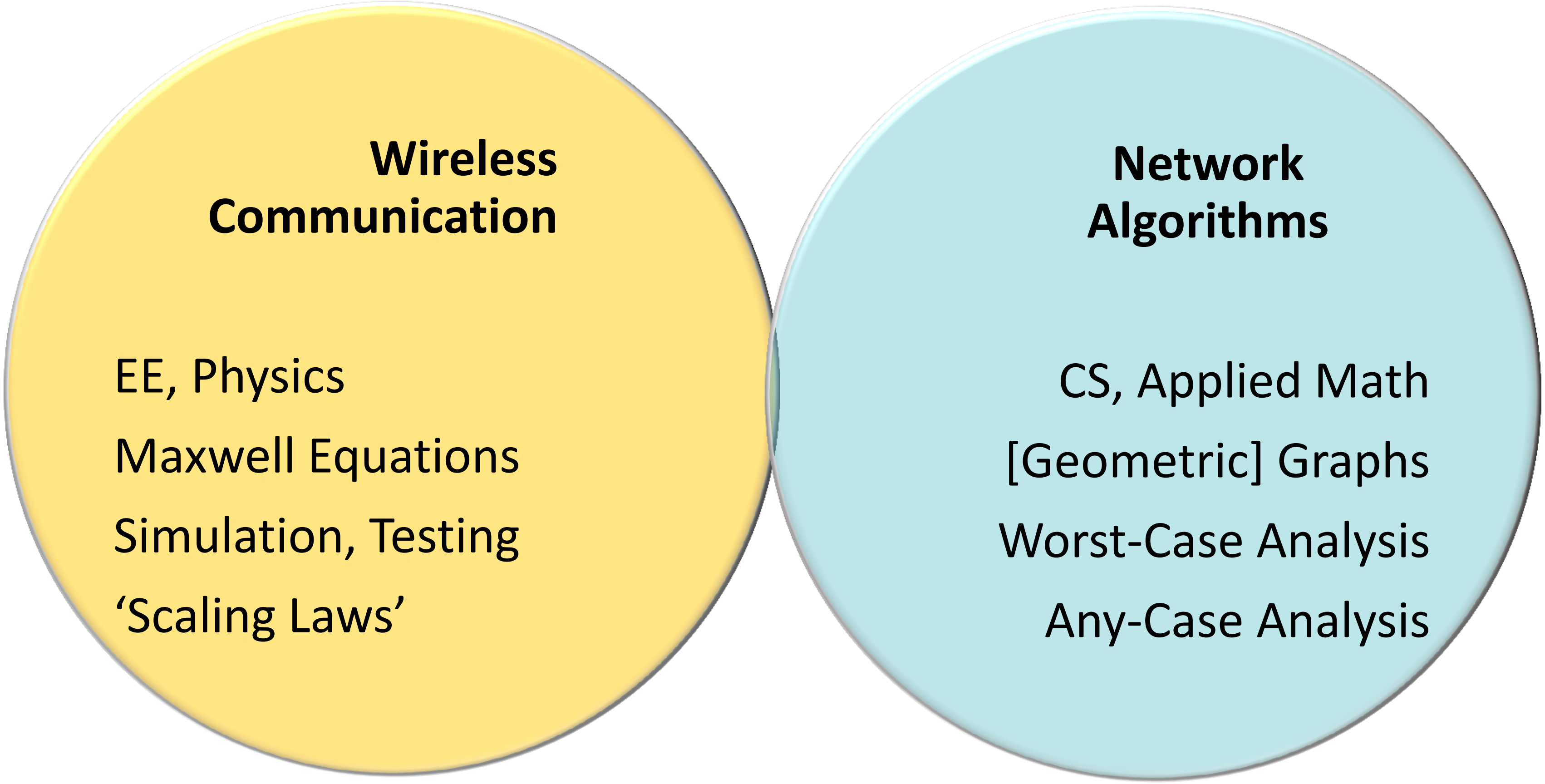
How much information can be transmitted in any network?

“Classic” Capacity

How much information can be transmitted in nice networks?

Worst-Case Capacity

How much information can be transmitted in nasty networks?


Wireless Communication
EE, Physics
Maxwell Equations
Simulation, Testing
'Scaling Laws'
Network Algorithms
CS, Applied Math
[Geometric] Graphs
Worst-Case Analysis
Any-Case Analysis

… is really taking off right now!

The Complexity of Connectivity …
Moscibroda et al., Infocom 2006

[Halldorsson et al, ICALP'11]

[Dams et al, ICALP'11]

[Hoefer et al, SPAA'10]

[Fanghaenel et al, PODC'09]

[Goussevskaia et al, INFOCOM'09]

[Lotker et al., INFOCOM'11]

[Kesselheim, SODA'11]

[Fanghaenel et al, ICALP'09]

[Kesselheim et al, DISC'10]

[Halldorsson et al, SODA'11]

[Avin et al, PODC'11]

[Kantor et al, STOC'11]

[Lebhar et al, IPDPS'09]

[Halldorsson, ESA'09]

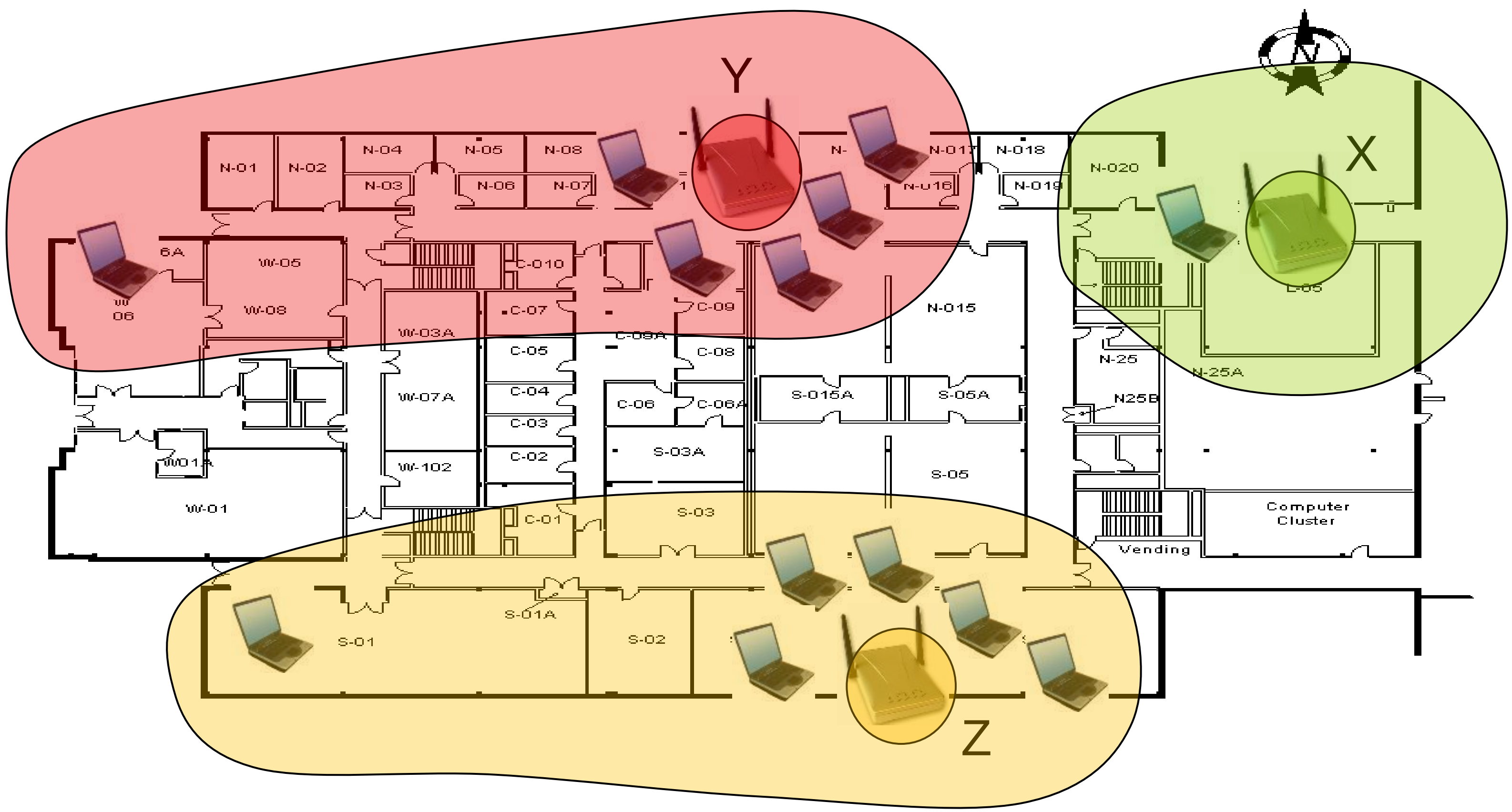
[Hua et al, TCS'09]

[Yu et al, DCOSS'11]

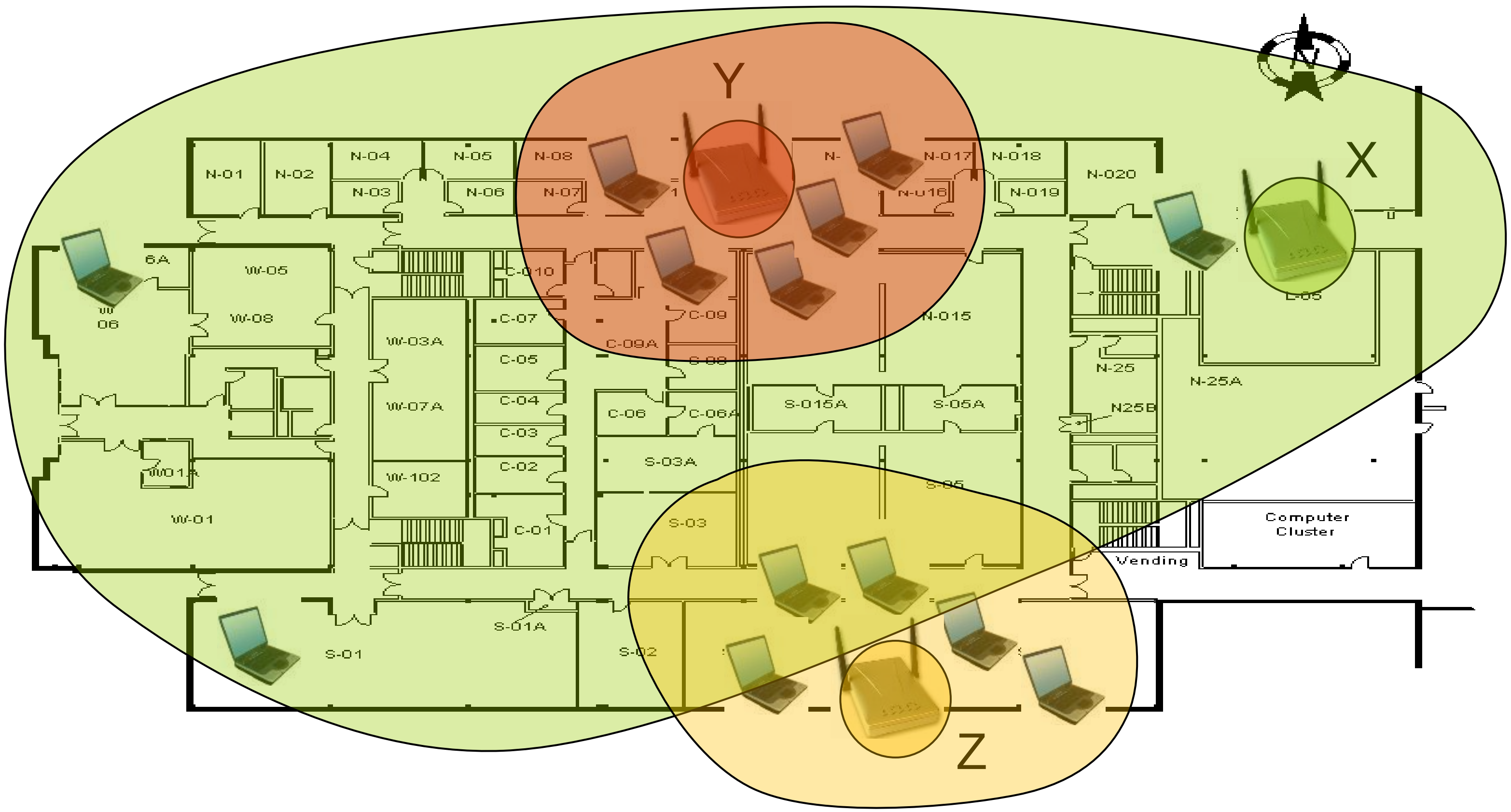
[Goussevskaia et al, MOBIHOC'07]

etc…

Possible Application – Hotspots in WLAN

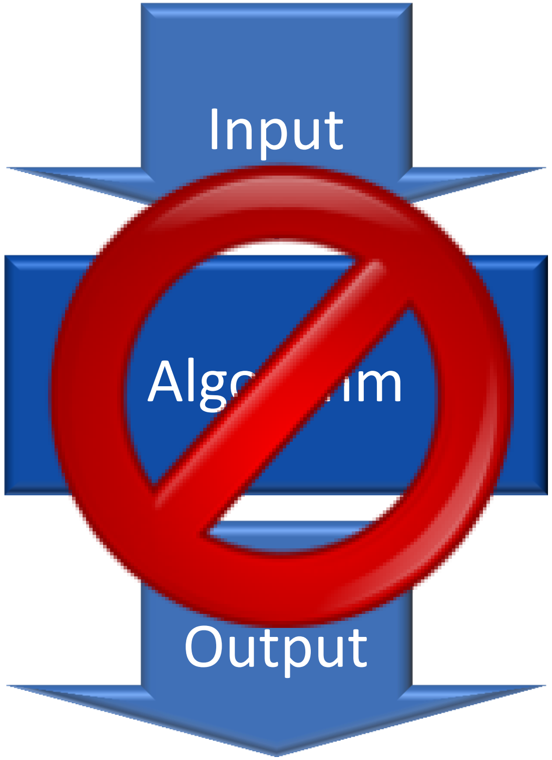


Possible Application – Hotspots in WLAN



Physical Algorithms?

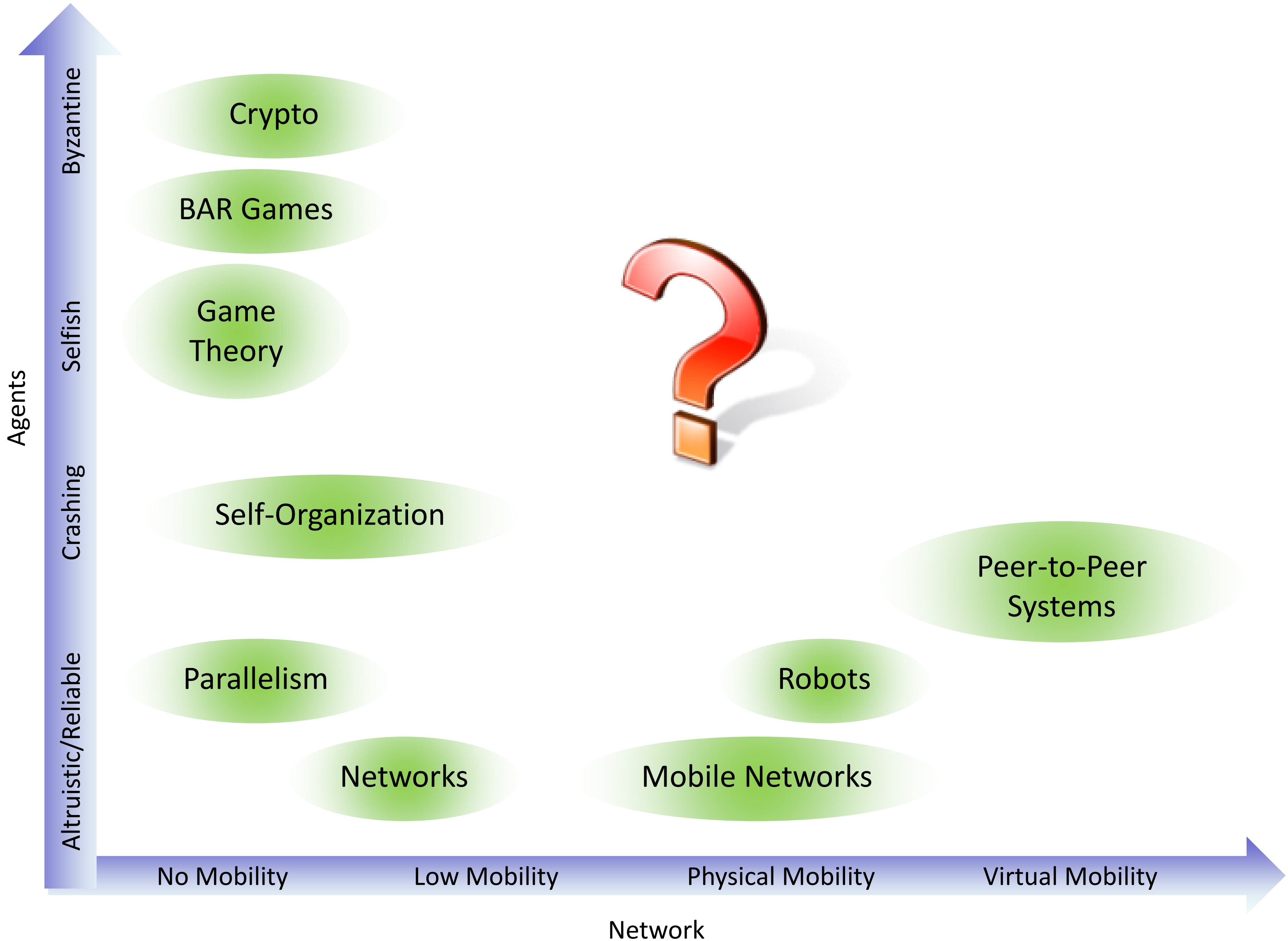
Physical Algorithms



no seq. input/output

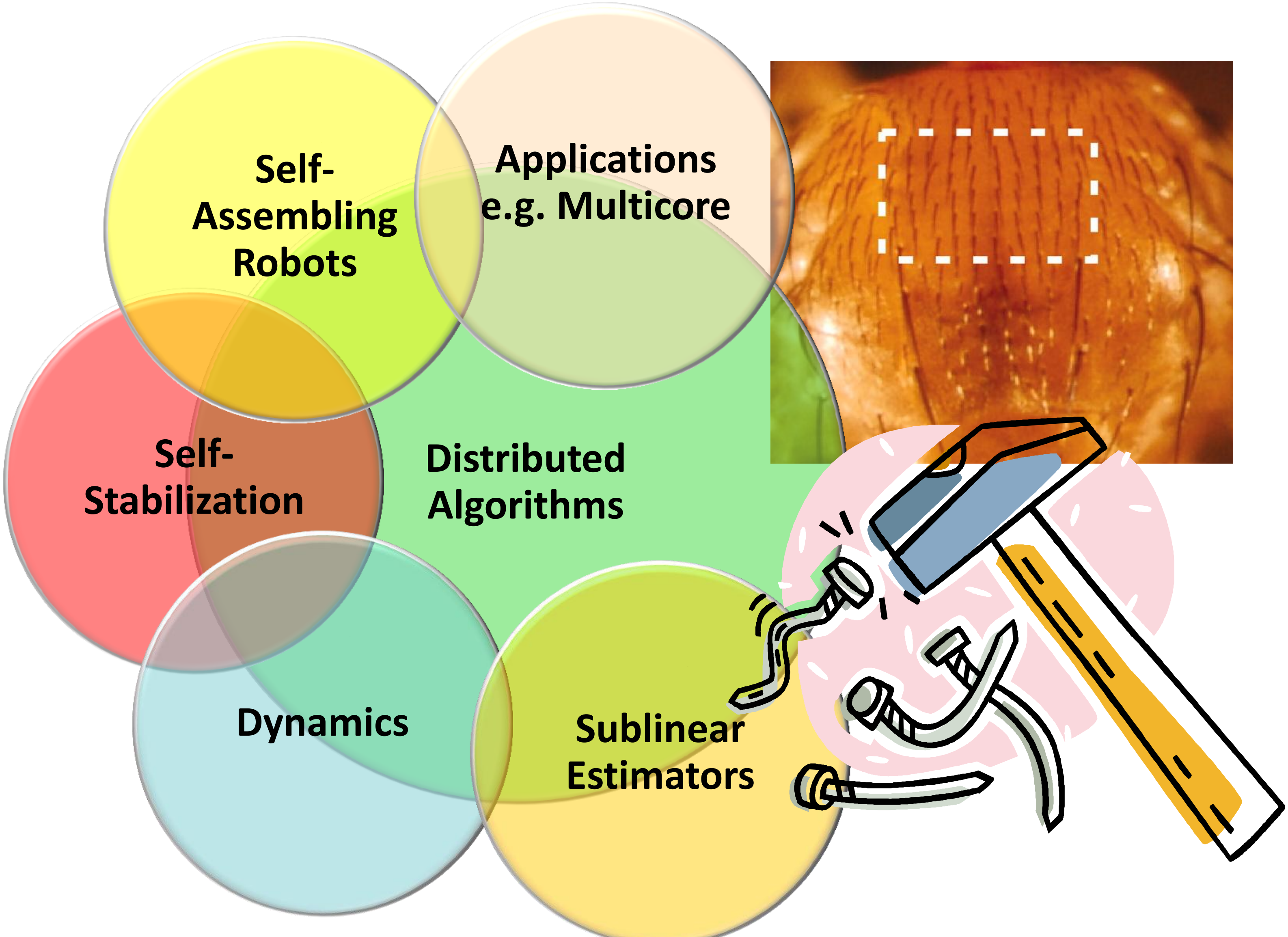
beyond laws of physics


Byzantine
Selfish
Crashing
Altruistic/Reliable
Agents
No Mobility
Low Mobility
Physical Mobility
Virtual Mobility
Network


Agents
Byzantine
Selfish
Crashing
Altruistic/Reliable
Crypto
BAR Games
Game Theory
Self-Organization
Parallelism
Networks
Mobile Networks
Robots
Peer-to-Peer Systems
No Mobility
Low Mobility
Physical Mobility
Virtual Mobility
Network

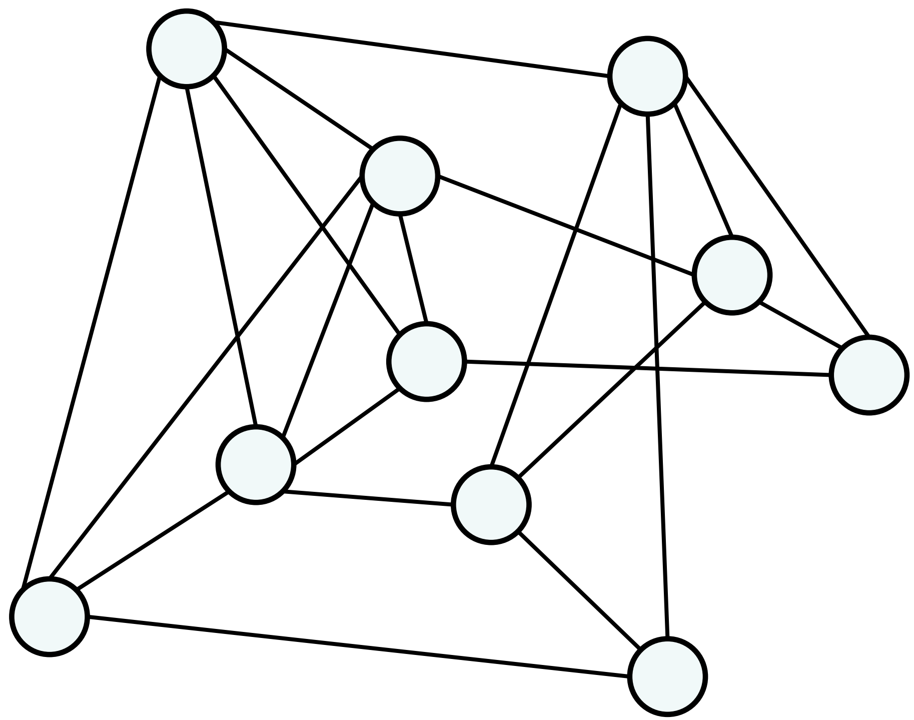
Some Unifying Theory?

Distributed Algorithms is Tool to Understand Physical Algorithms

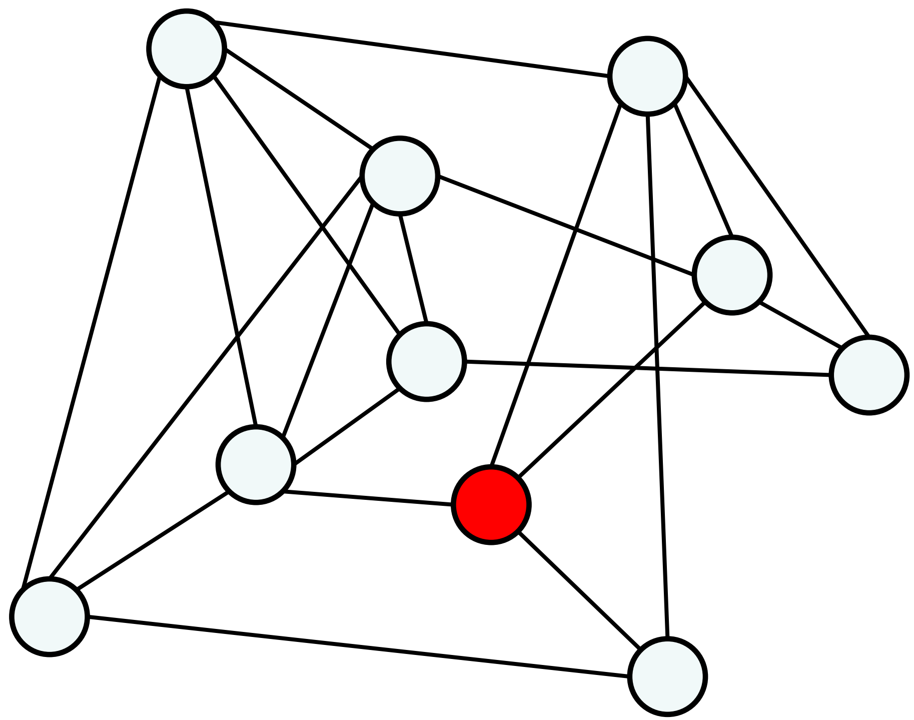


Distributed Algorithms: A Simple Example

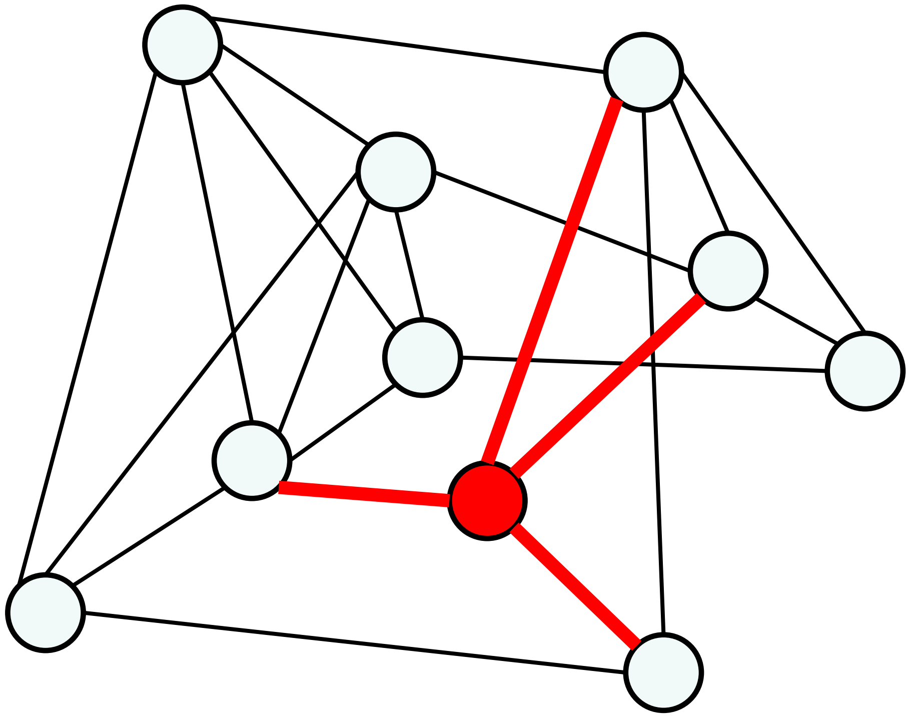
How Many Nodes in Network?



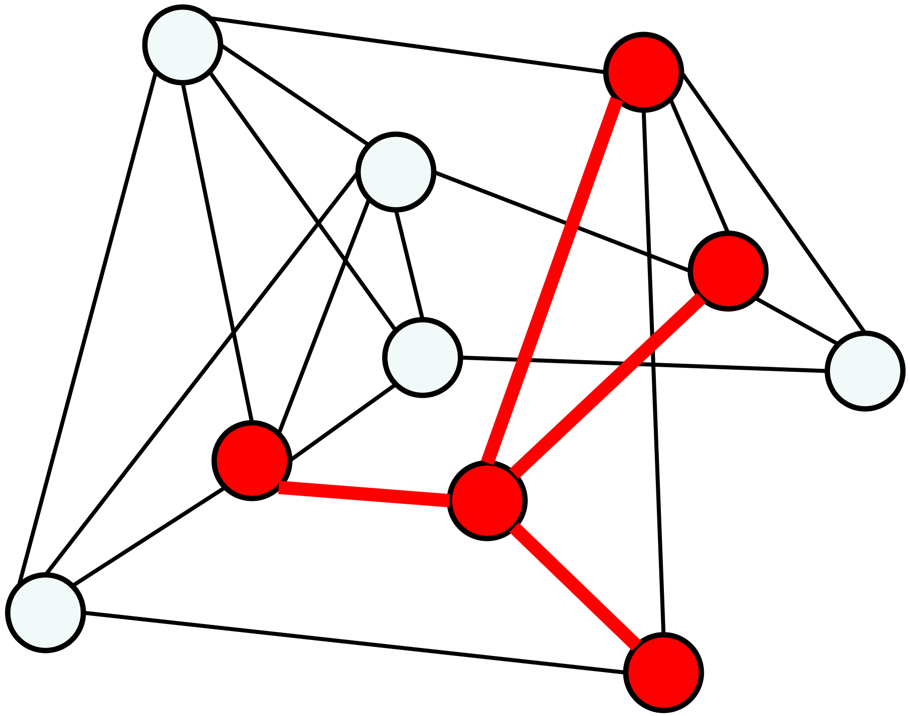
How Many Nodes in Network?



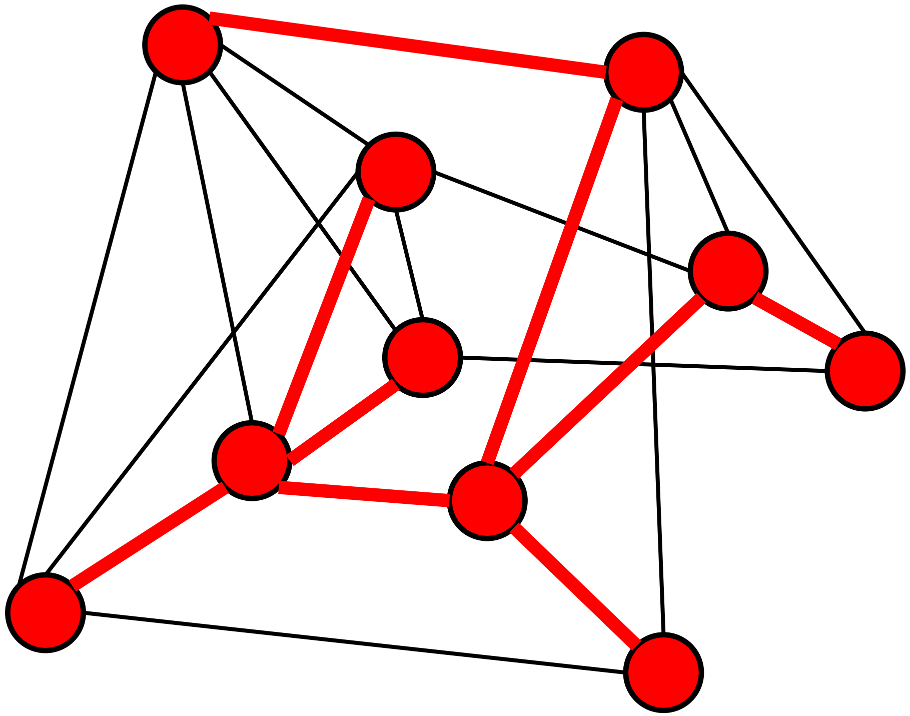
How Many Nodes in Network?



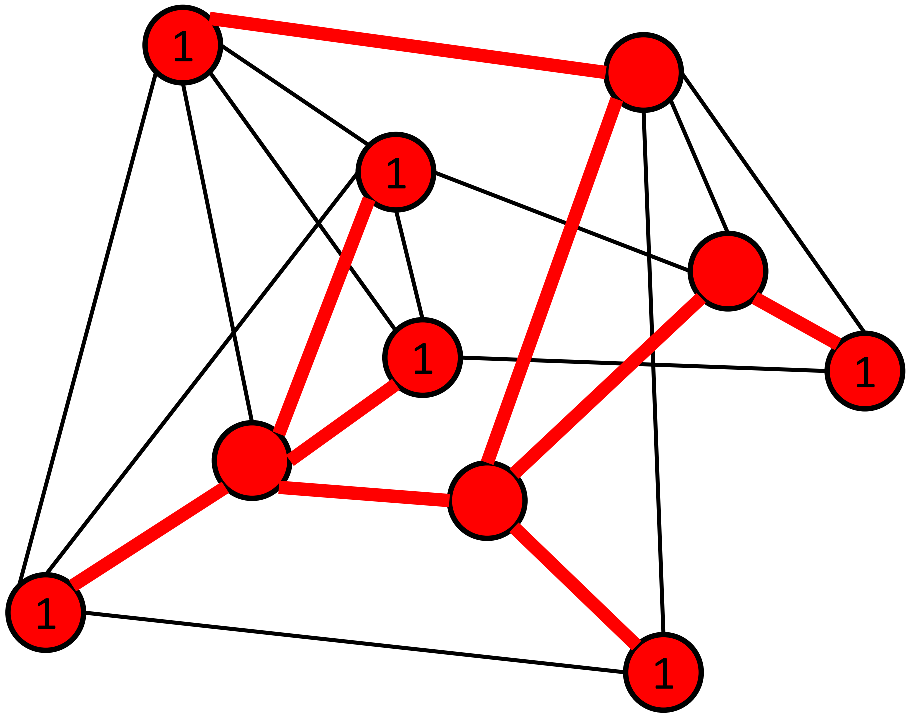
How Many Nodes in Network?



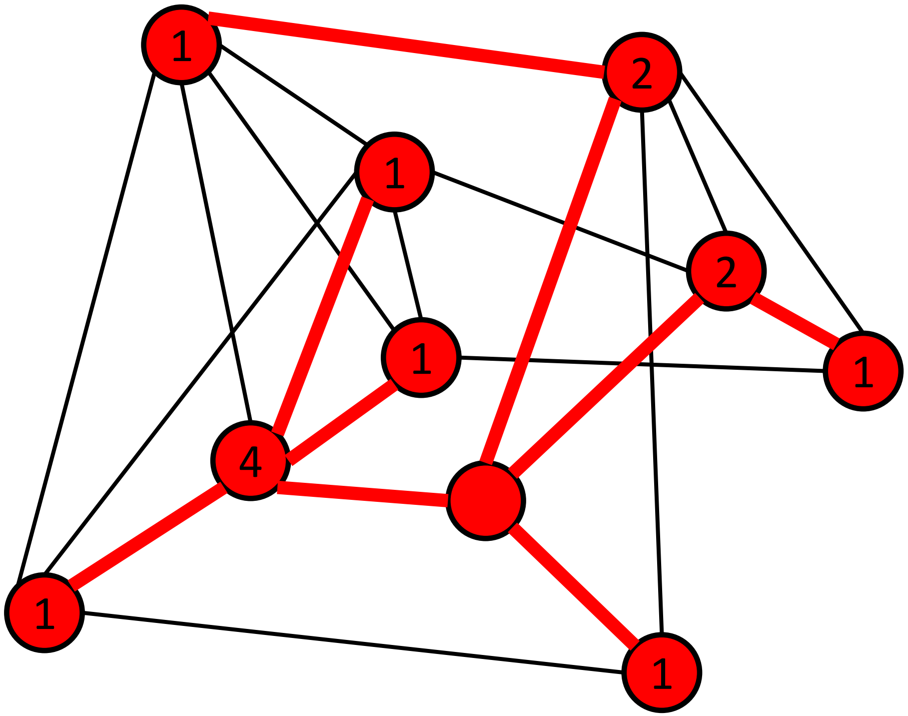
How Many Nodes in Network?



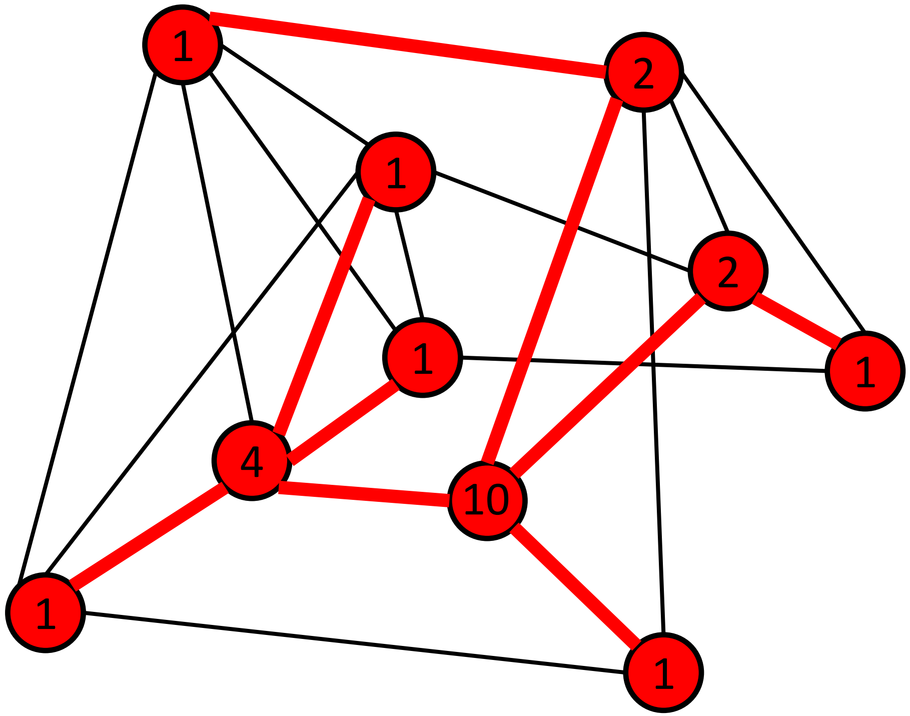
How Many Nodes in Network?



How Many Nodes in Network?

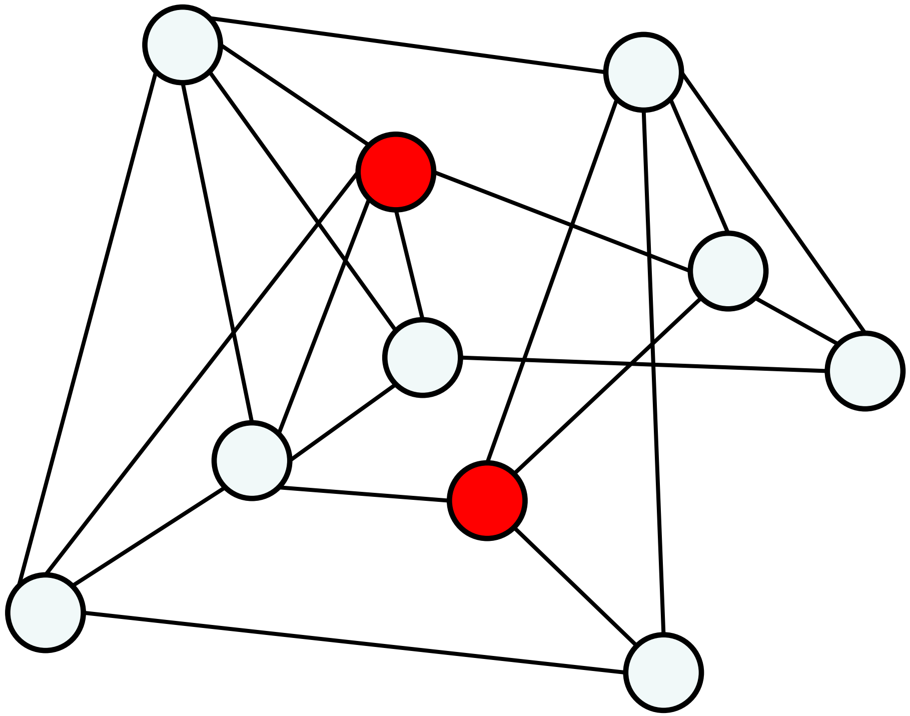


How Many Nodes in Network?



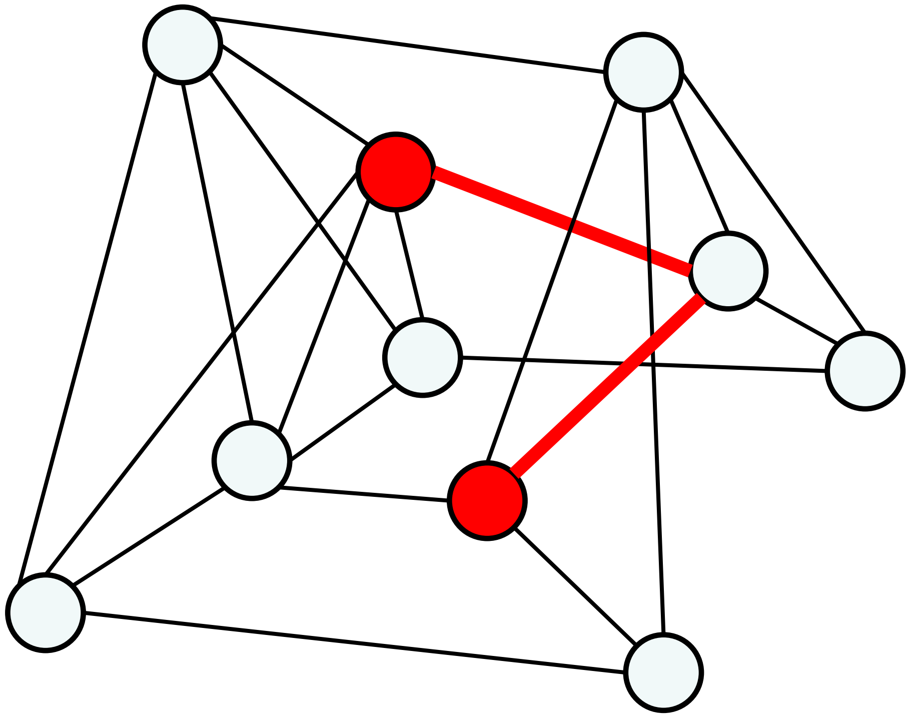
With a simple flooding/echo process, a network can find the number of nodes in time $O(D)$, where D is the diameter (size) of the network.

Diameter (Size) of Network?



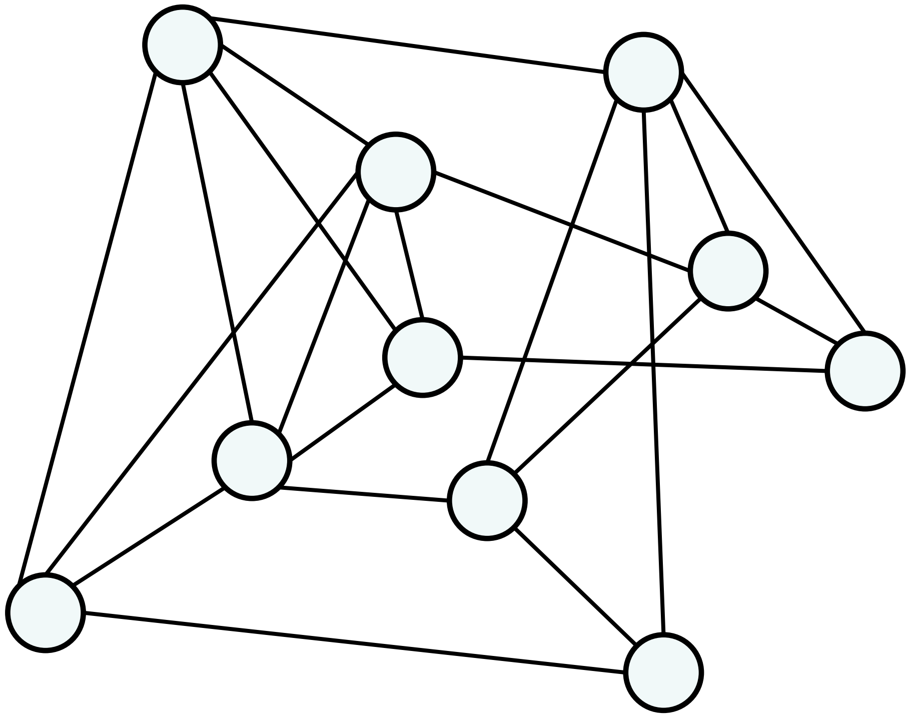
- Distance between two nodes = Number of hops of shortest path

Diameter (Size) of Network?



- Distance between two nodes = Number of hops of shortest path

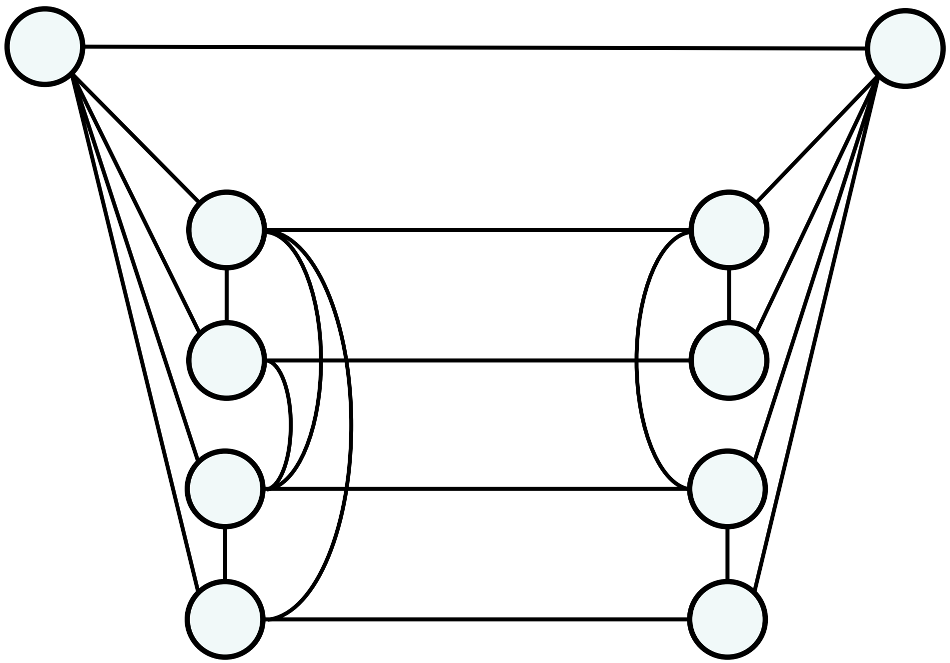
Diameter (Size) of Network?



- Distance between two nodes = Number of hops of shortest path
- Diameter of network = Maximum distance, between any two nodes

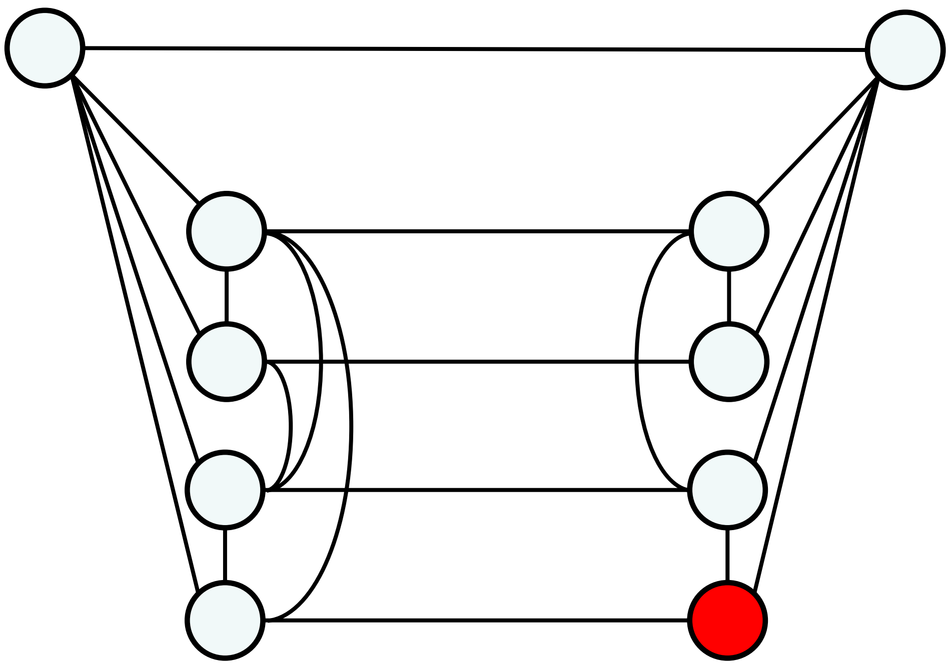
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



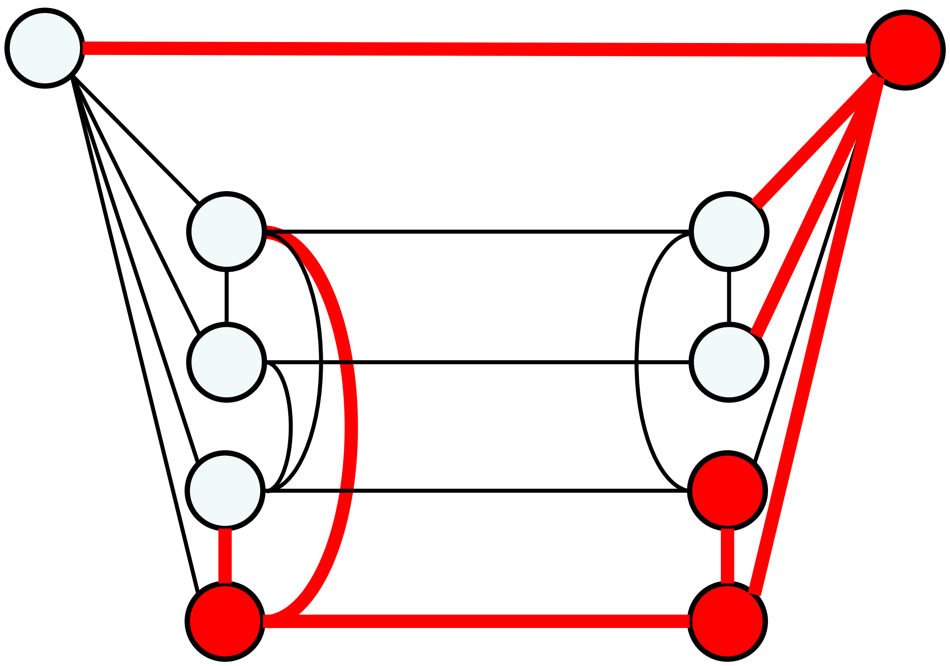
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



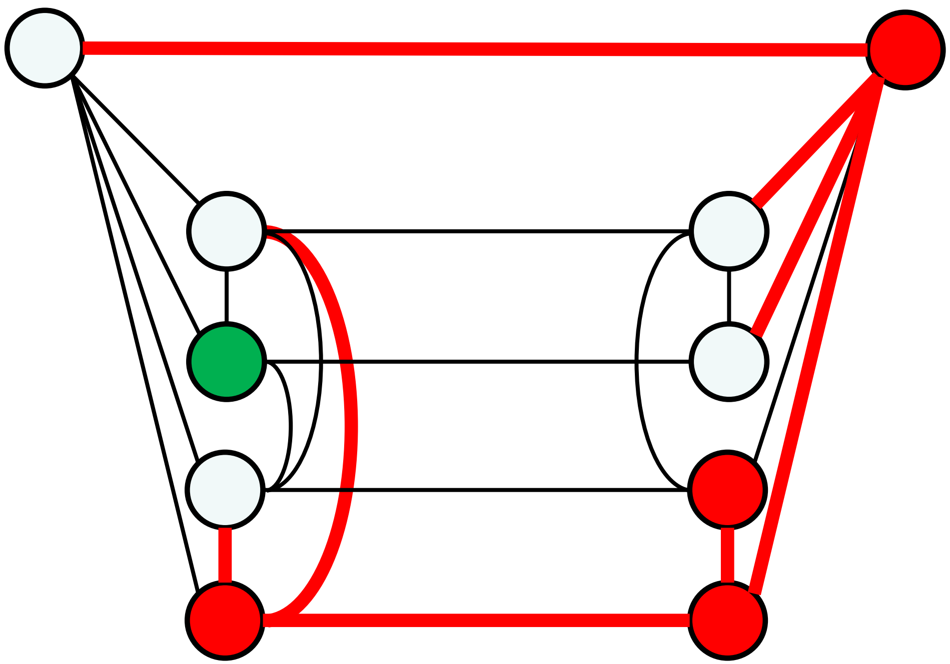
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



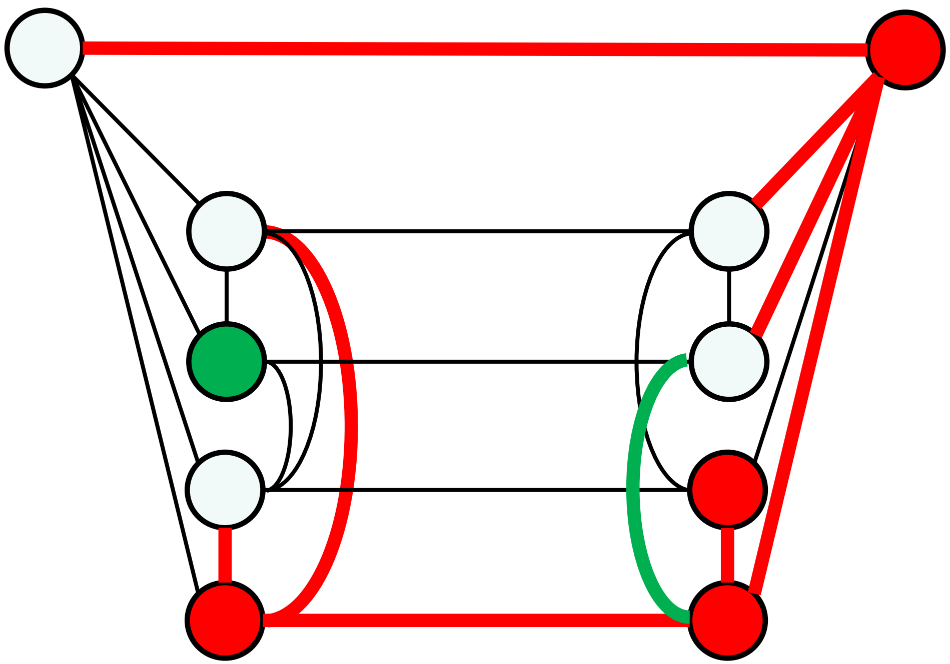
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



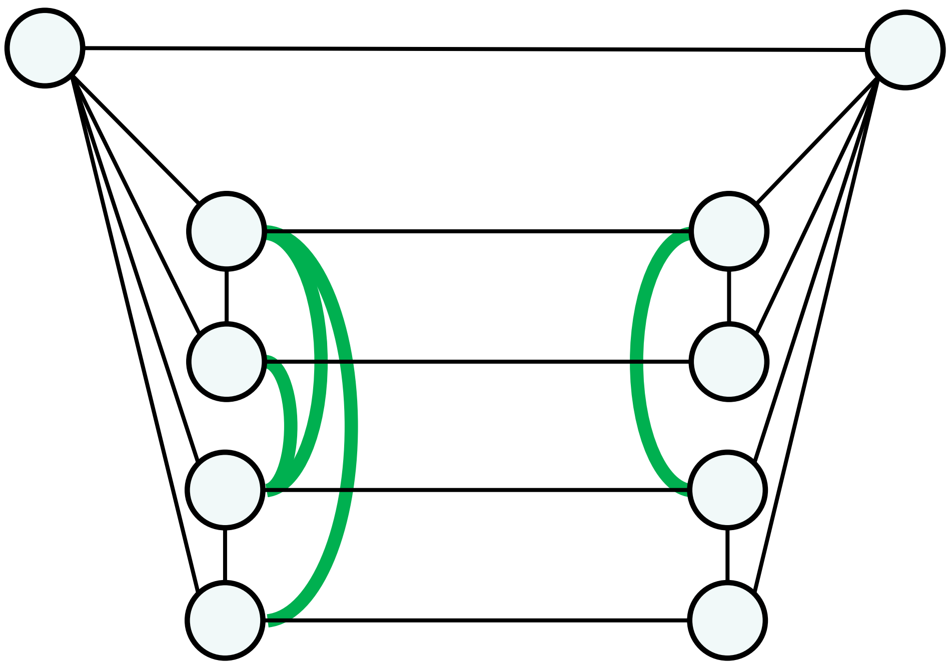
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



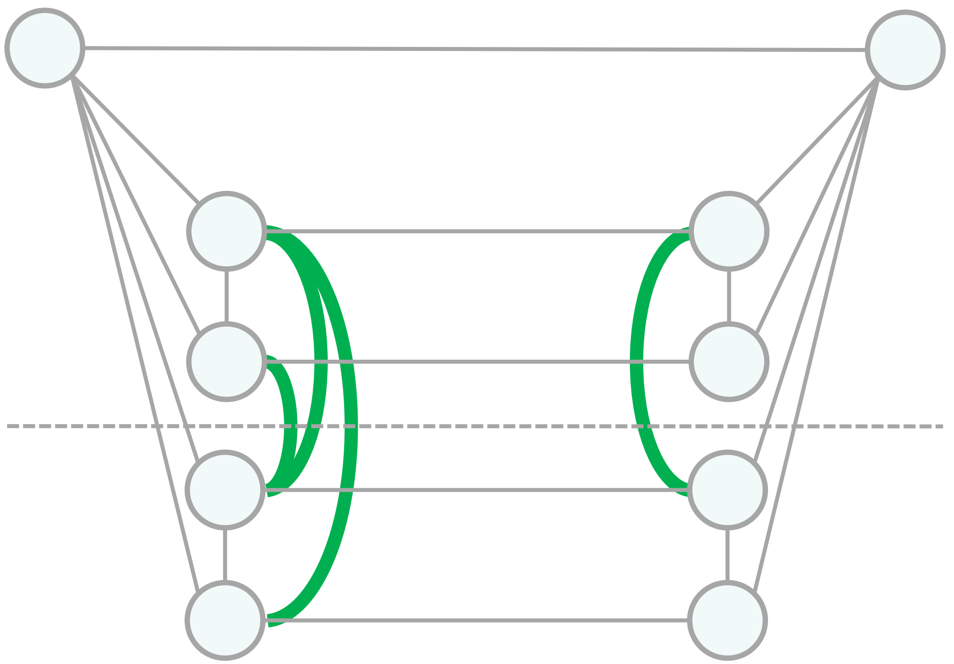
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



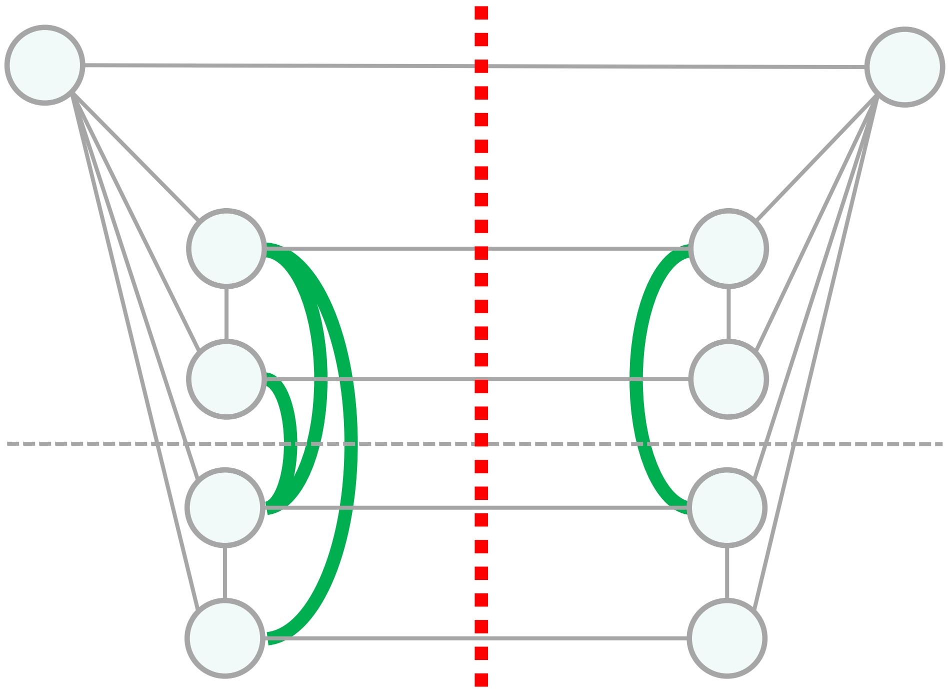
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



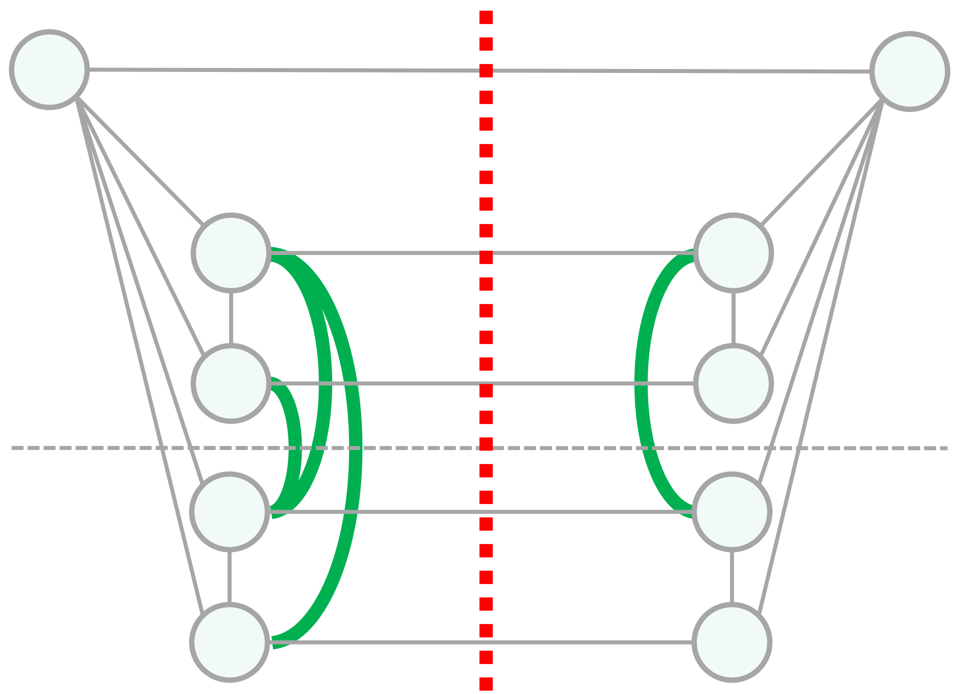
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)



Networks Cannot Compute Their Diameter in Sublinear Time!

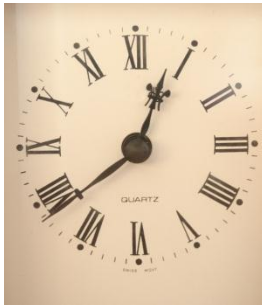
(even if diameter is just a small constant)

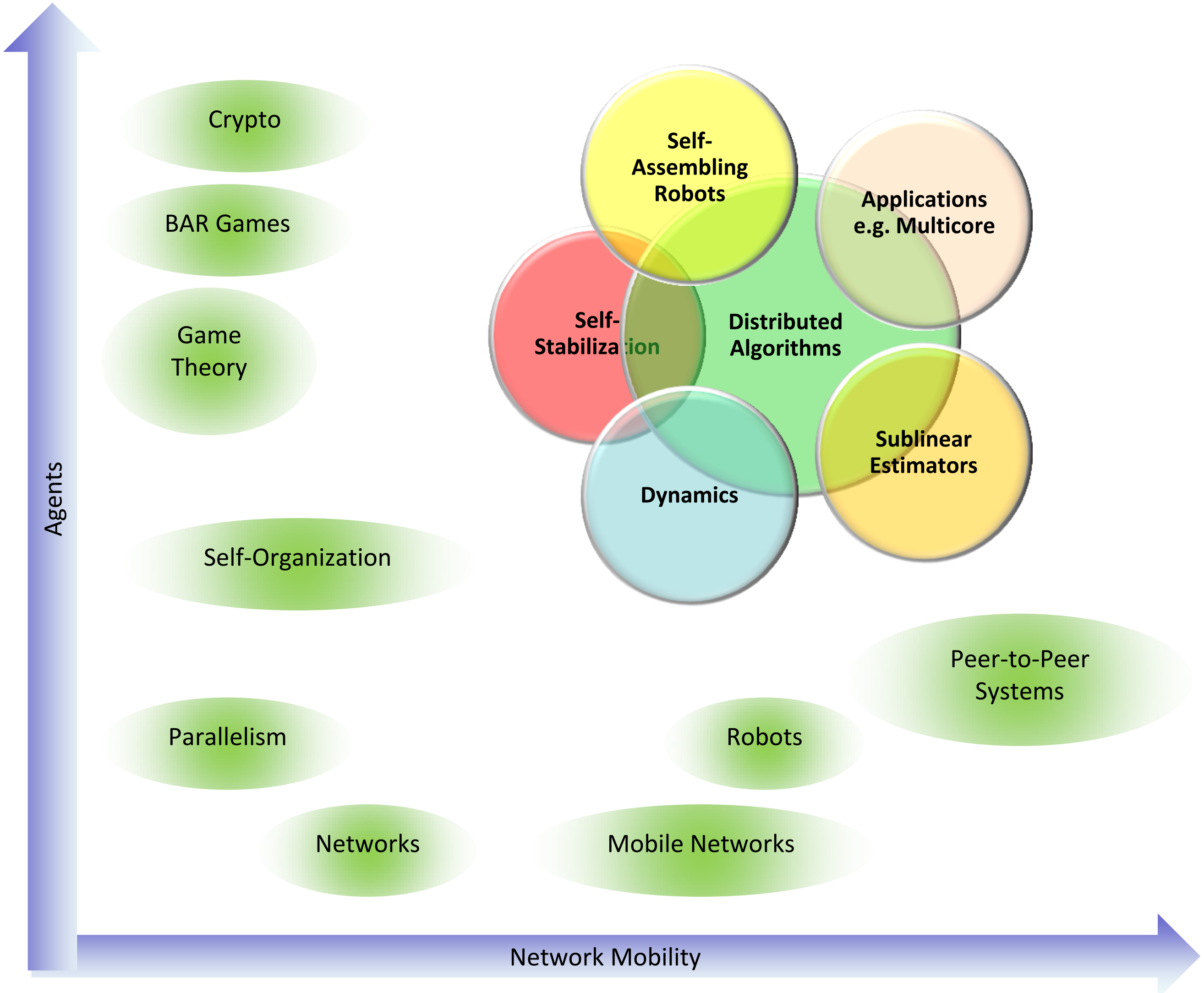


Pair of nodes not connected on both sides? We have $\Theta(n^2)$ information that has to be transmitted over $O(n)$ edges, which takes $\Omega(n)$ time!

[Frischknecht, Holzer, W, 2012]

Summary




Agents
Network Mobility
Crypto
BAR Games
Game Theory
Self-Organization
Parallelism
Networks
Mobile Networks
Robots
Peer-to-Peer Systems
Self-Assembling Robots
Applications e.g. Multicore
Self-Stabilization
Distributed Algorithms
Sublinear Estimators
Dynamics

Thank You!

Questions & Comments?

Thanks to my co-authors
Silvio Frischknecht
Stephan Holzer
Christoph Lenzen
Thomas Moscibroda
Thomas Locher
Philipp Sommer

www.disco.ethz.ch