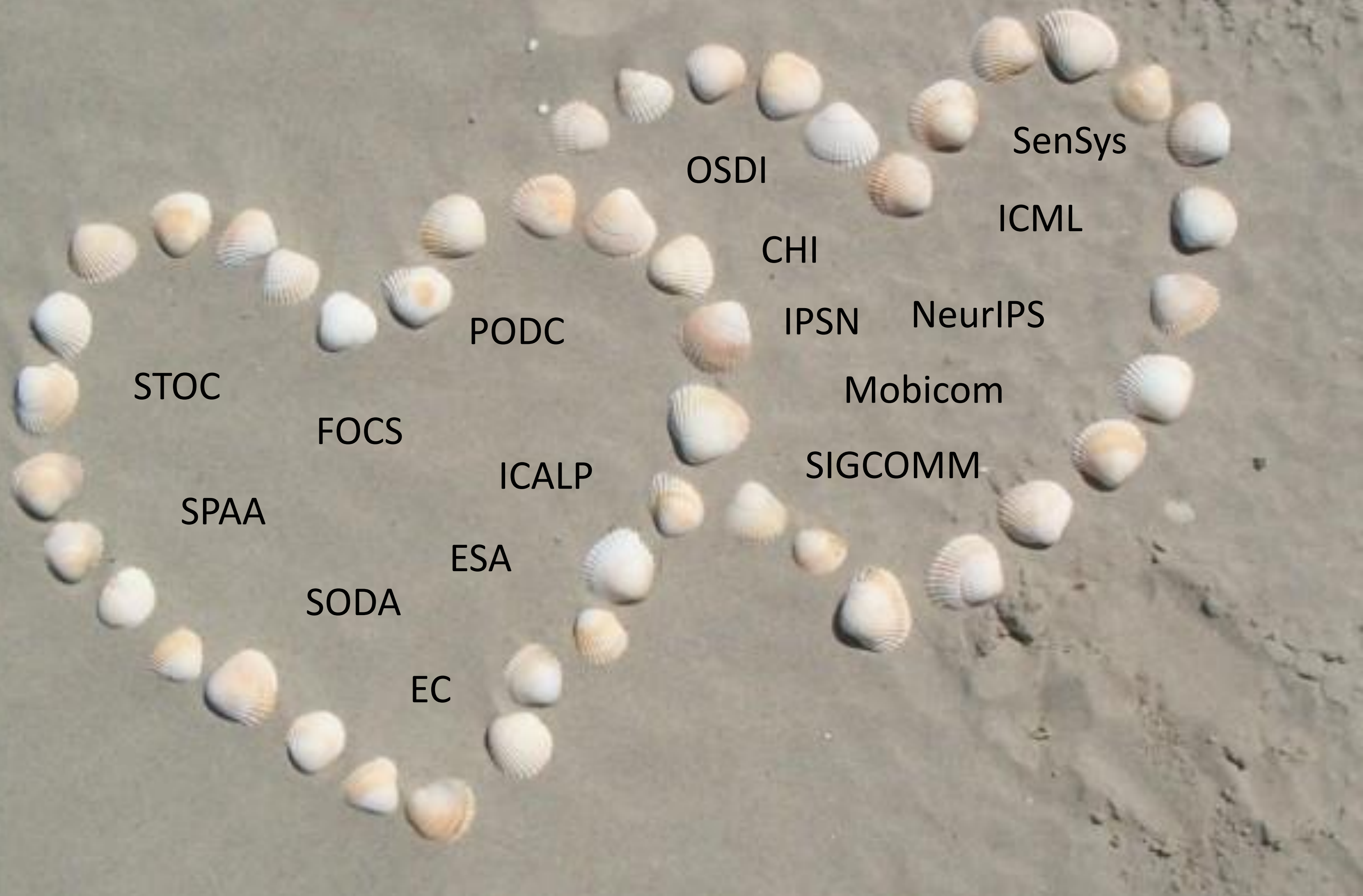
Wireless Network Algorithms
Looking Back & Moving Forward

PARENTAL ADVISORY EXPLICIT CONTENT

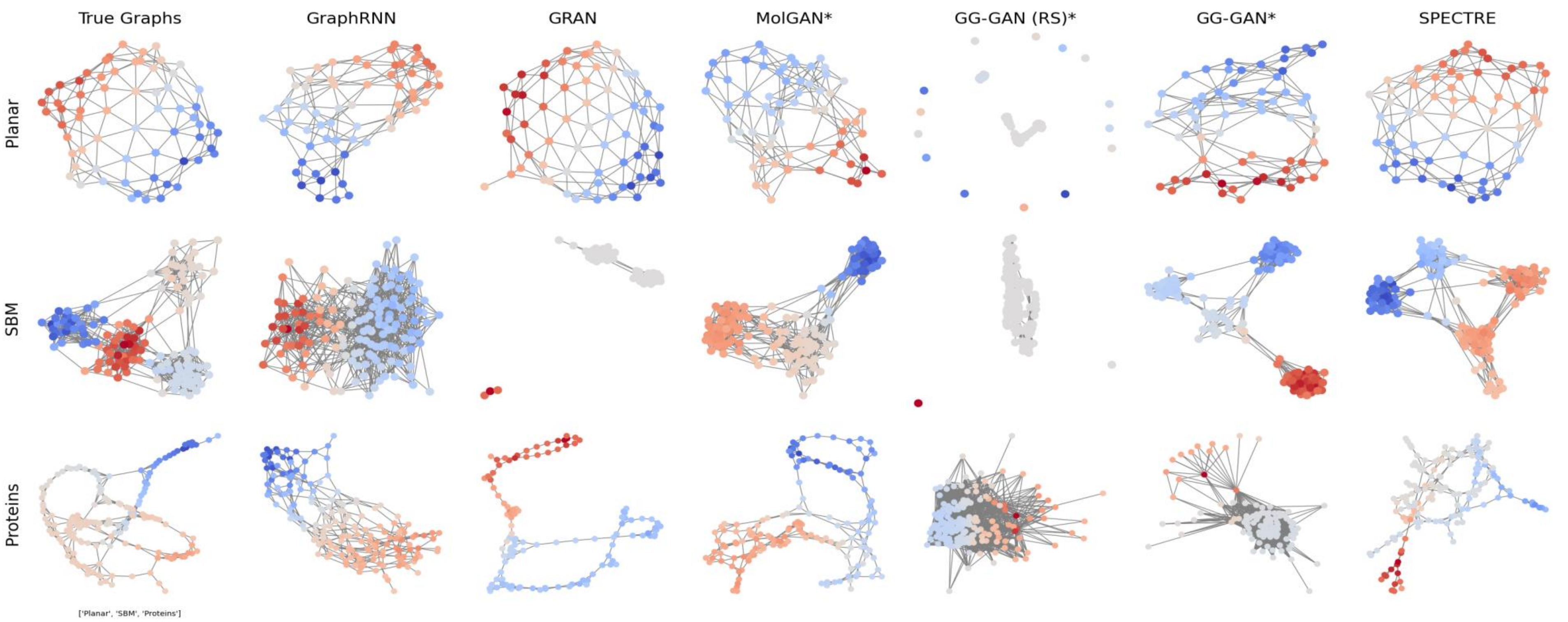
Roger Wattenhofer

Theory & Practice



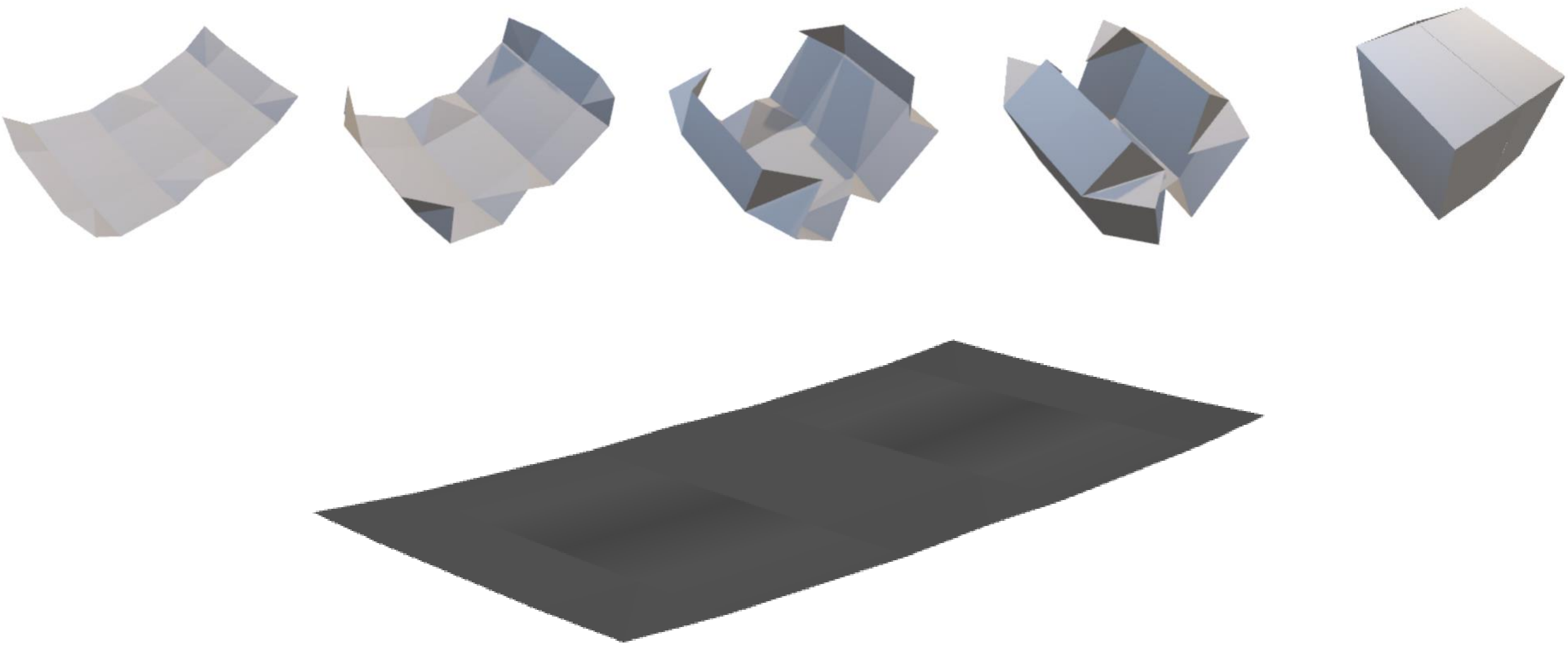
SPECTRE : Spectral Conditioning Helps to Overcome the Expressivity Limits of One-shot Graph Generators

Karolis Martinkus [1] **Andreas Loukas** [* 2] **Nathanaël Perraudin** [* 3] **Roger Wattenhofer** [1]



Automating Rigid Origami Design

Jeremia Geiger, Karolis Martinkus, Oliver Richter, Roger Wattenhofer



International Symposium on Algorithmics of Wireless Networks

Ad Hoc Networks
Autonomous Mobile Robots
Communication Protocols
Complexity and Computability
Computational Models
Data Aggregation and Fusion
Dynamic Networks, Temporal Graphs
Energy Management, Power Saving
Fault Tolerance and Dependability
Game Theoretic Aspects
Infrastructure Discovery
Internet of Things
Localization
Medium Access Control

Mobility and Dynamics
Obstacle Avoidance
Pattern Formation, Experimental Analysis
Population Protocols, Swarm Computing
Resource Efficiency
RFID Algorithms
Routing and Data Propagation
Self-stabilization, Self-* Properties
Sensor Networks
Systems and Testbeds
Time Synchronization
Topology Control
Tracking
Virtual Infrastructures

International Symposium on Algorithmics of Wireless Networks

Ad Hoc Networks
Autonomous Mobile Robots
Communication Protocols
Complexity and Computability
Computational Models
Data Aggregation and Fusion
Dynamic Networks, Temporal Graphs
Energy Management, Power Saving
Fault Tolerance and Dependability
Game Theoretic Aspects
Infrastructure Discovery
Internet of Things
Localization
Medium Access Control

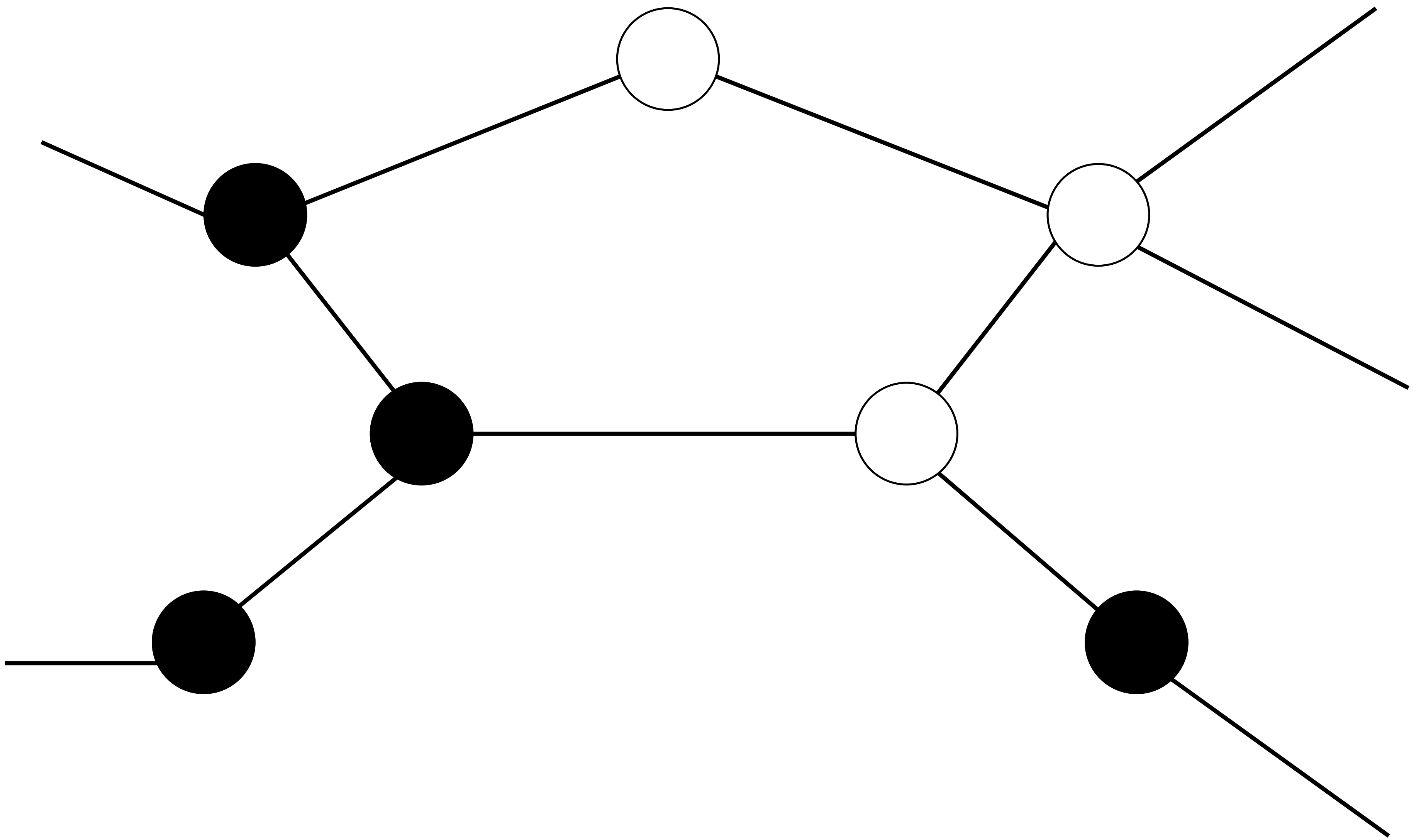
Mobility and Dynamics
~~Obstacle Avoidance~~
Pattern Formation, Experimental Analysis
Population Protocols, Swarm Computing
Resource Efficiency
RFID Algorithms
Routing and Data Propagation
Self-stabilization, Self-* Properties
Sensor Networks
Systems and Testbeds
Time Synchronization
Topology Control
Tracking
Virtual Infrastructures

Minority Process

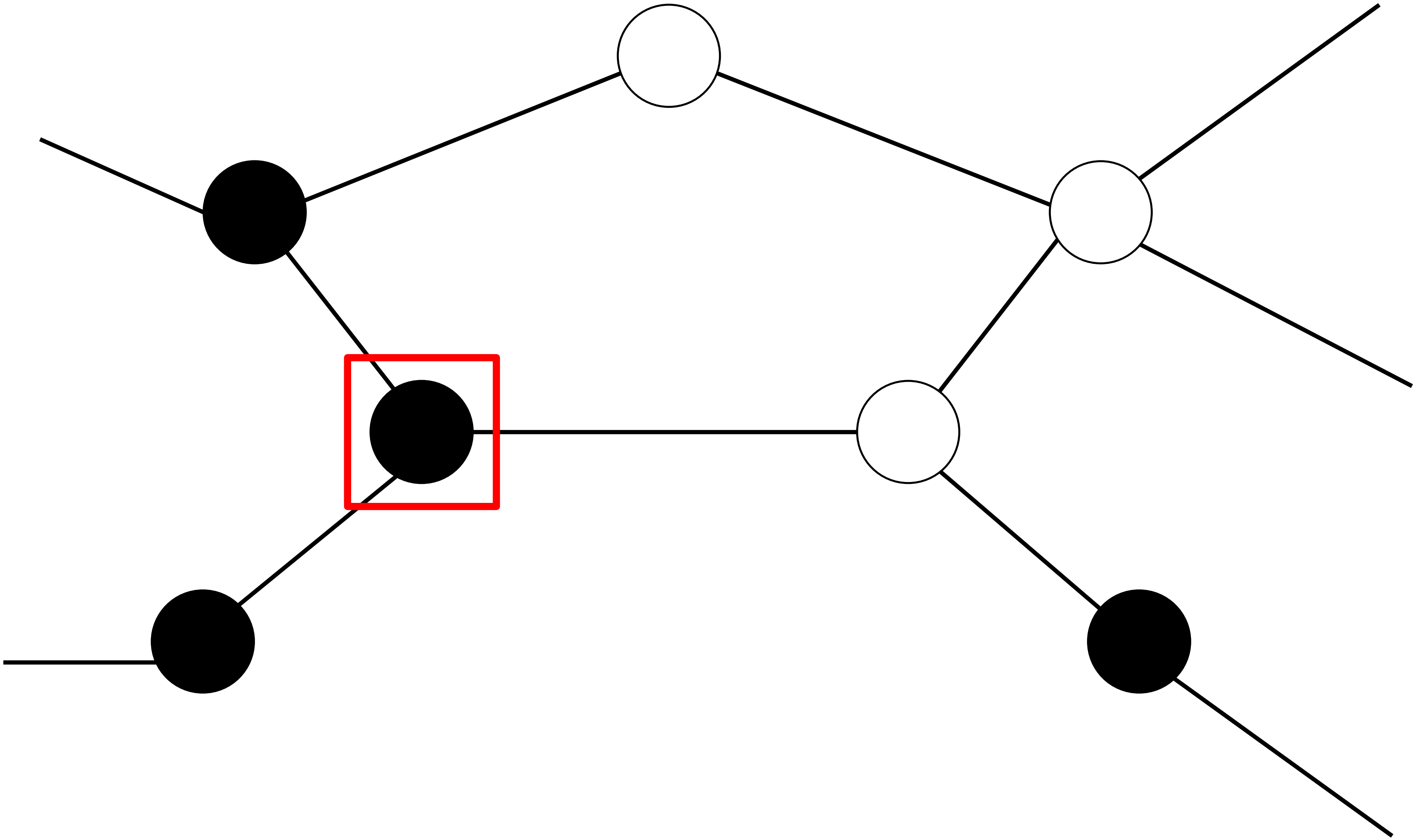
Minority Process

Minority Process

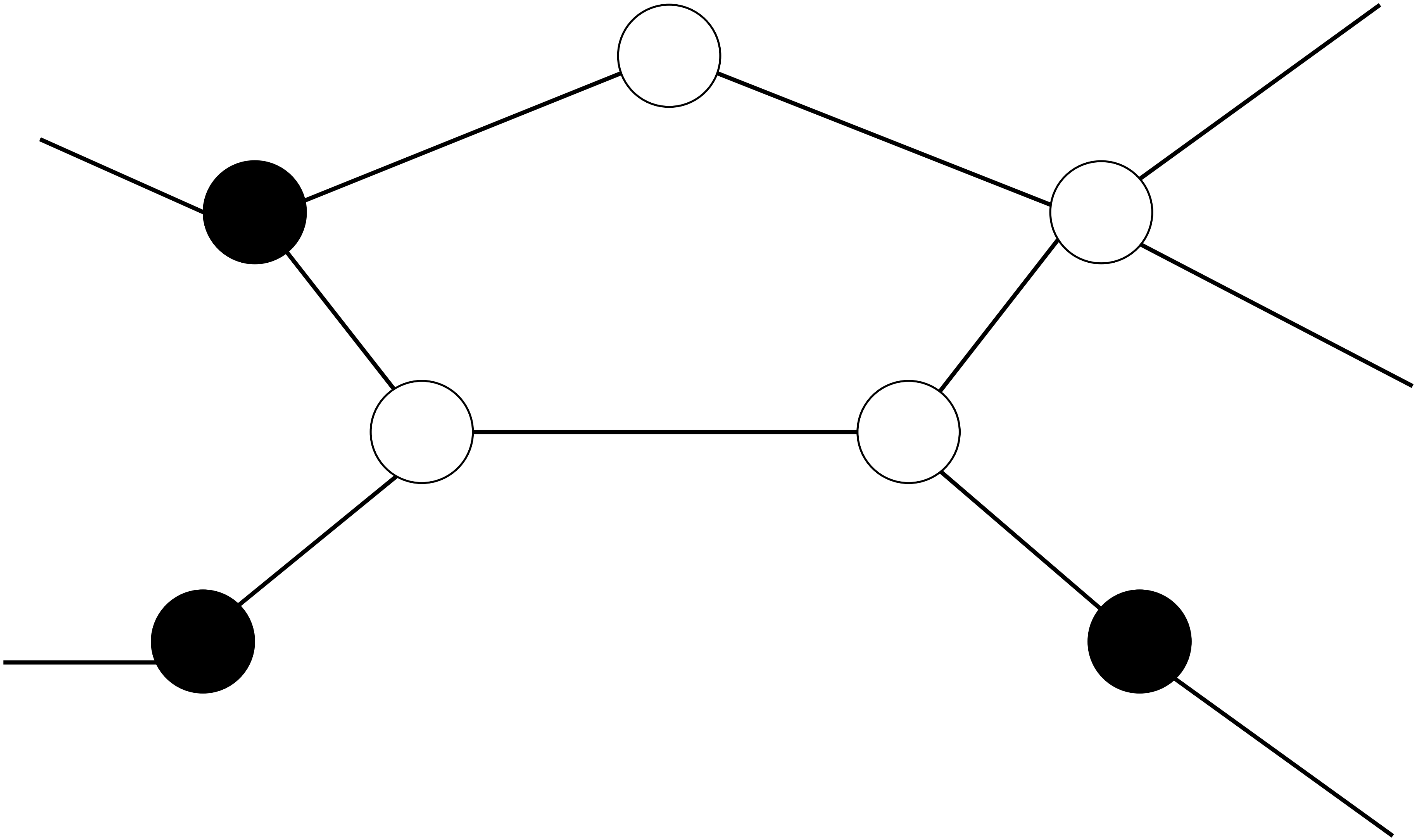
Minority Process



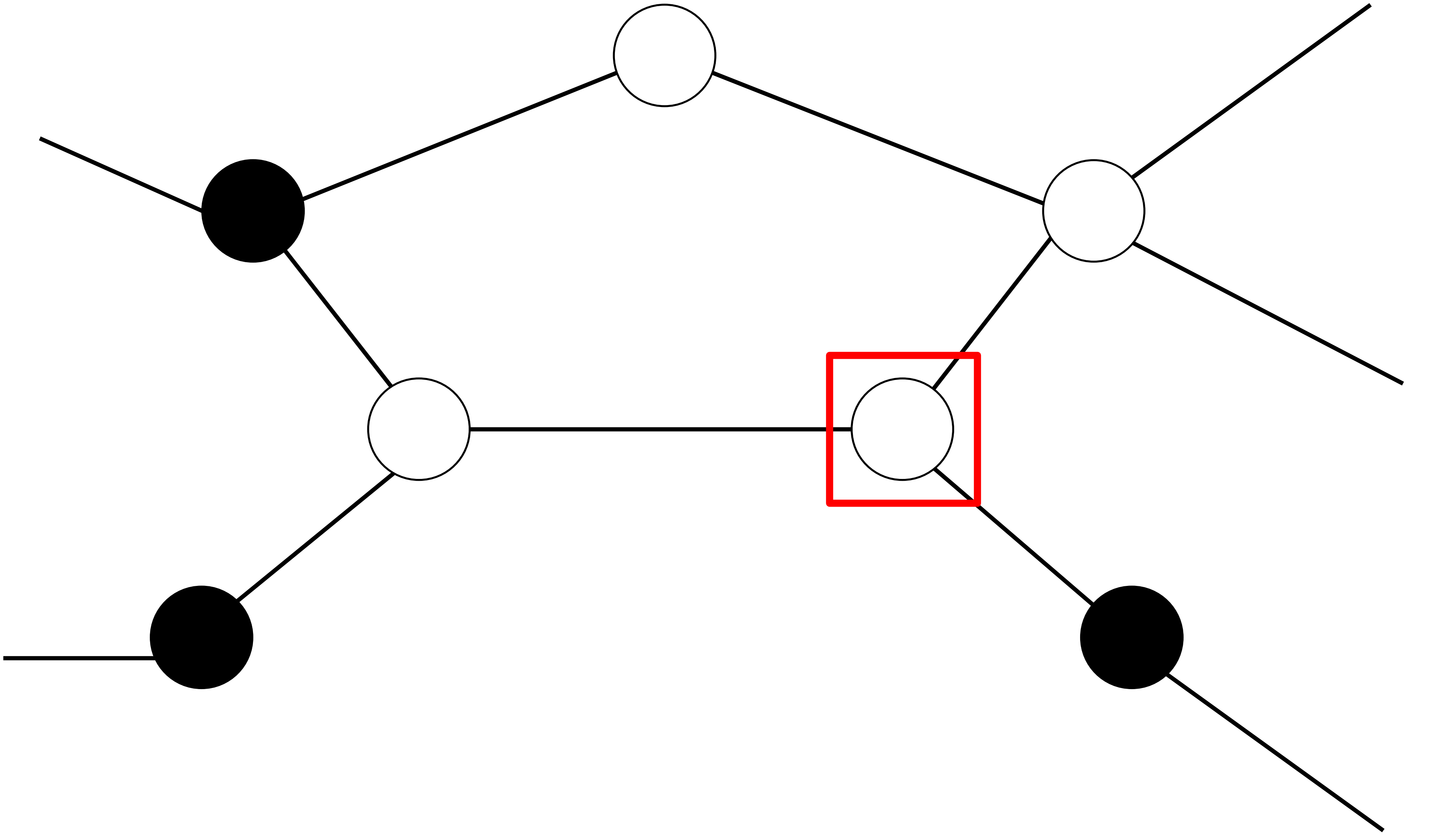
Minority Process



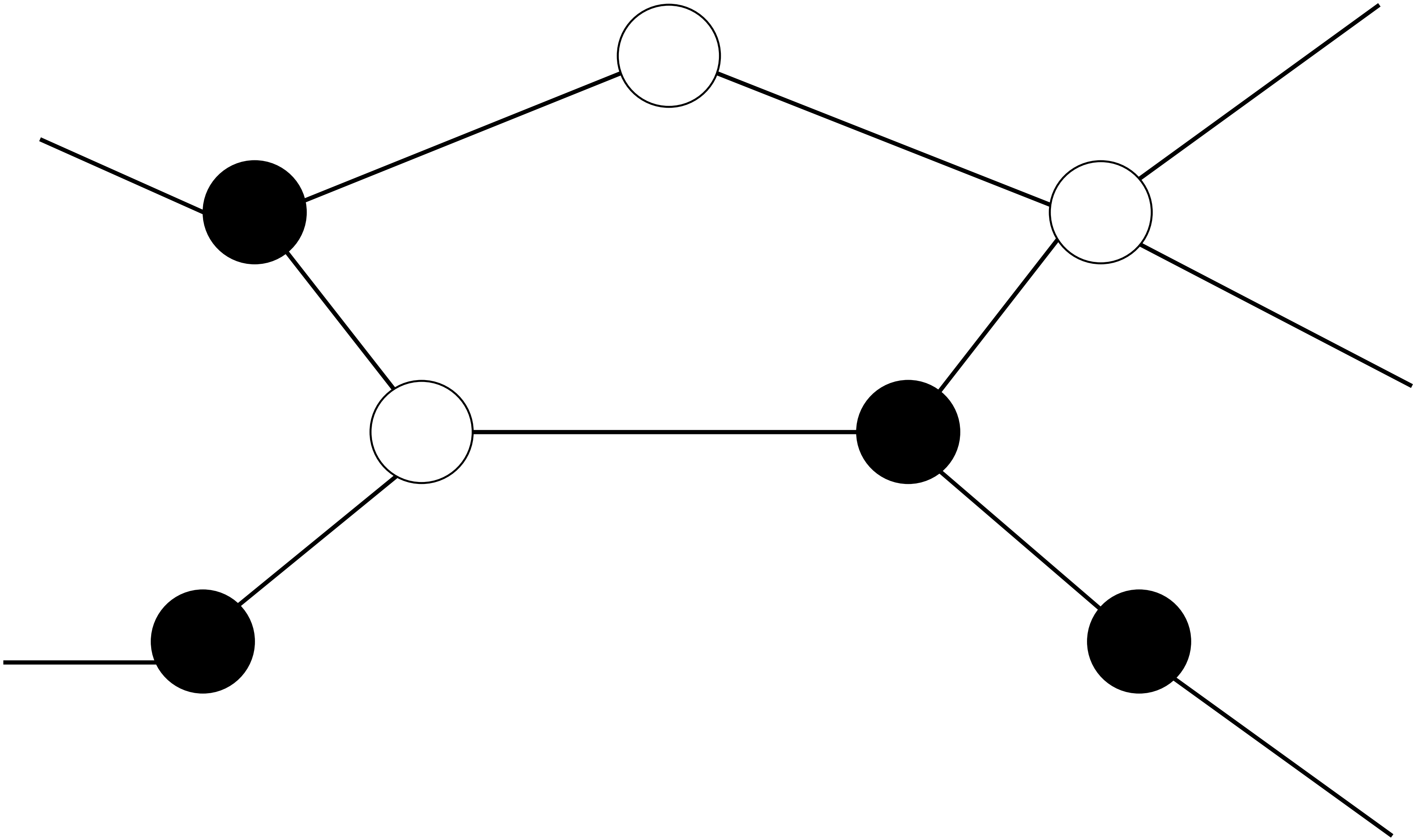
Minority Process



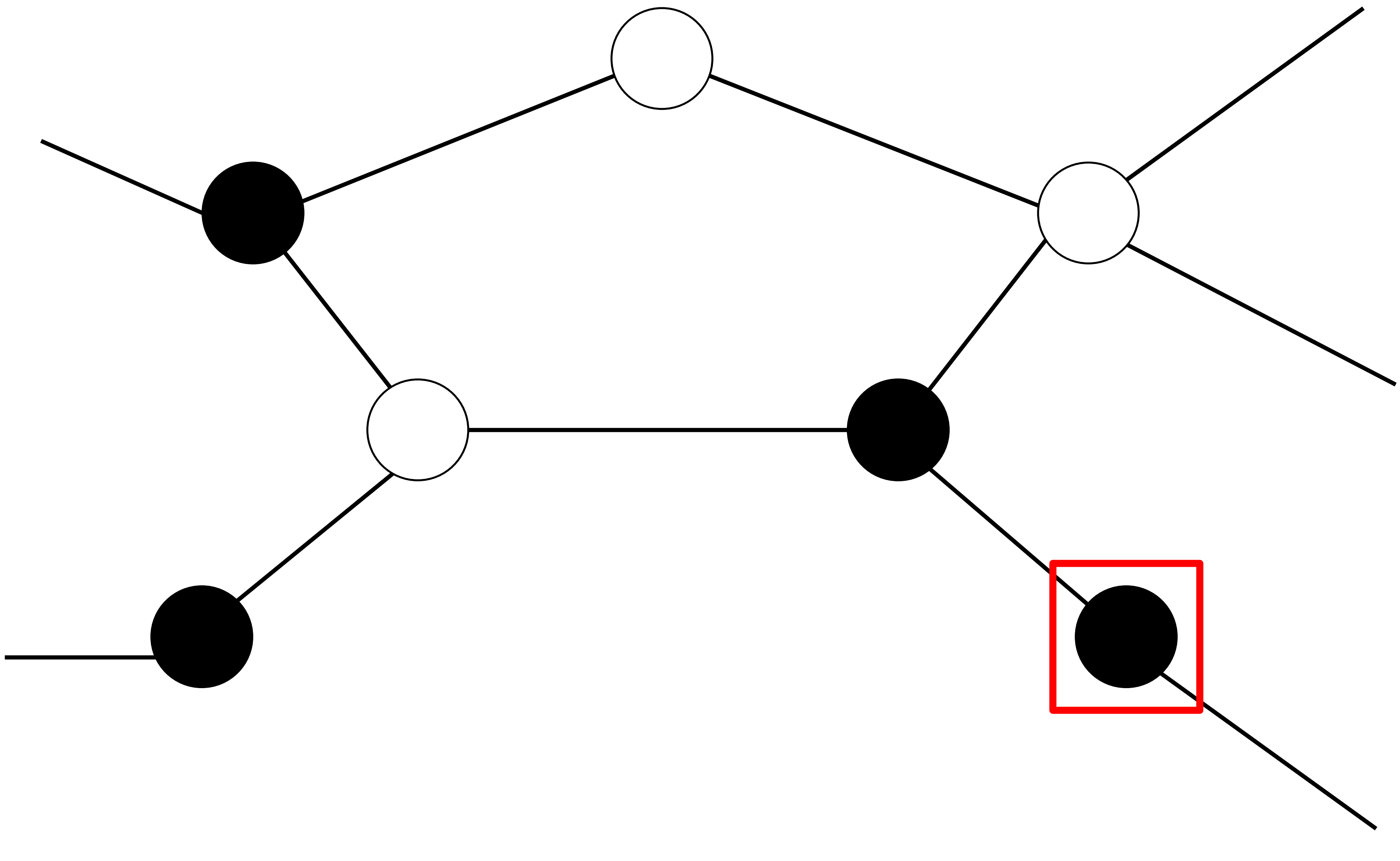
Minority Process



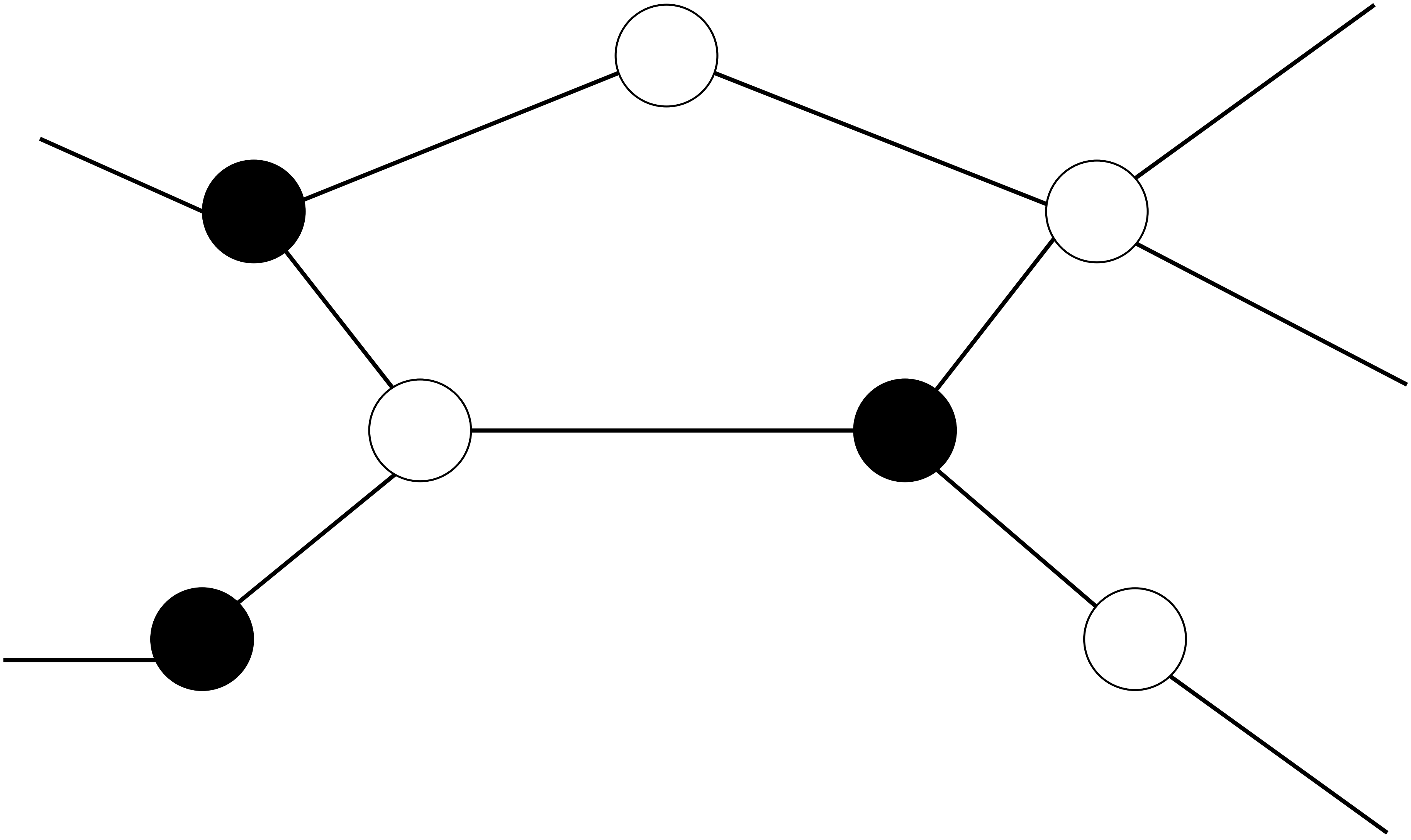
Minority Process



Minority Process



Minority Process



Stabilization time

- Synchronous – all switchable nodes

Stabilization time

- Synchronous – all switchable nodes

Stabilization time

- Synchronous – all switchable nodes

Stabilization time

- Synchronous – all switchable nodes

Stabilization time

- Synchronous – all switchable nodes

Stabilization time

- Synchronous – all switchable nodes

  → *possibly infinite*

Stabilization time

- Synchronous – all switchable nodes

  → *possibly infinite*

- Sequential – one node at a time

Stabilization time

- Synchronous – all switchable nodes

  → *possibly infinite*

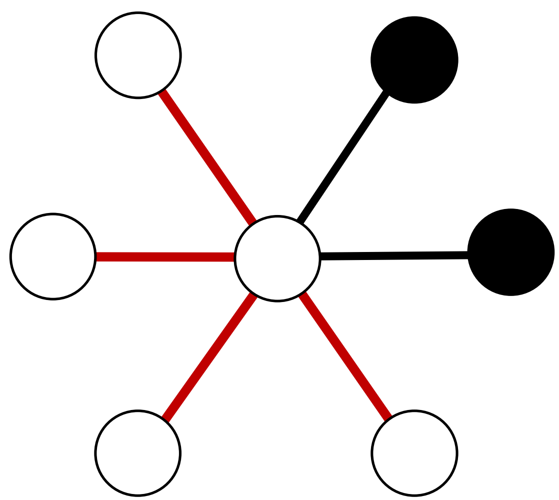
- **Sequential – one node at a time**

Stabilization time

- Synchronous – all switchable nodes

  → *possibly infinite*

- **Sequential – one node at a time**

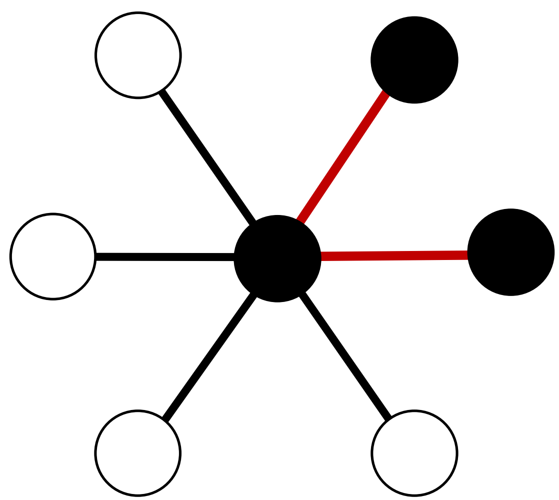


Stabilization time

- Synchronous – all switchable nodes

  → *possibly infinite*

- **Sequential – one node at a time**



Stabilization time

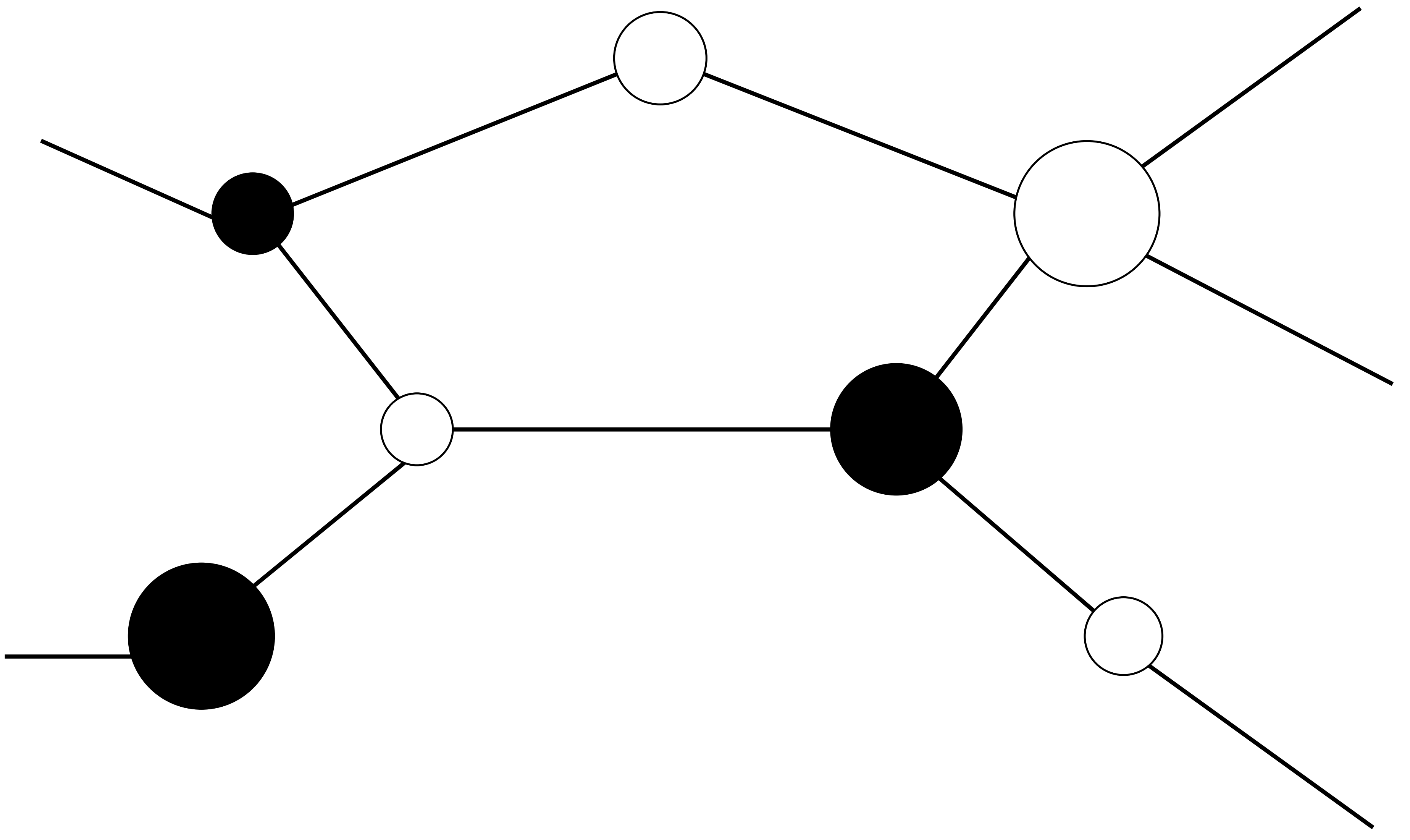
- Synchronous – all switchable nodes

  → *possibly infinite*

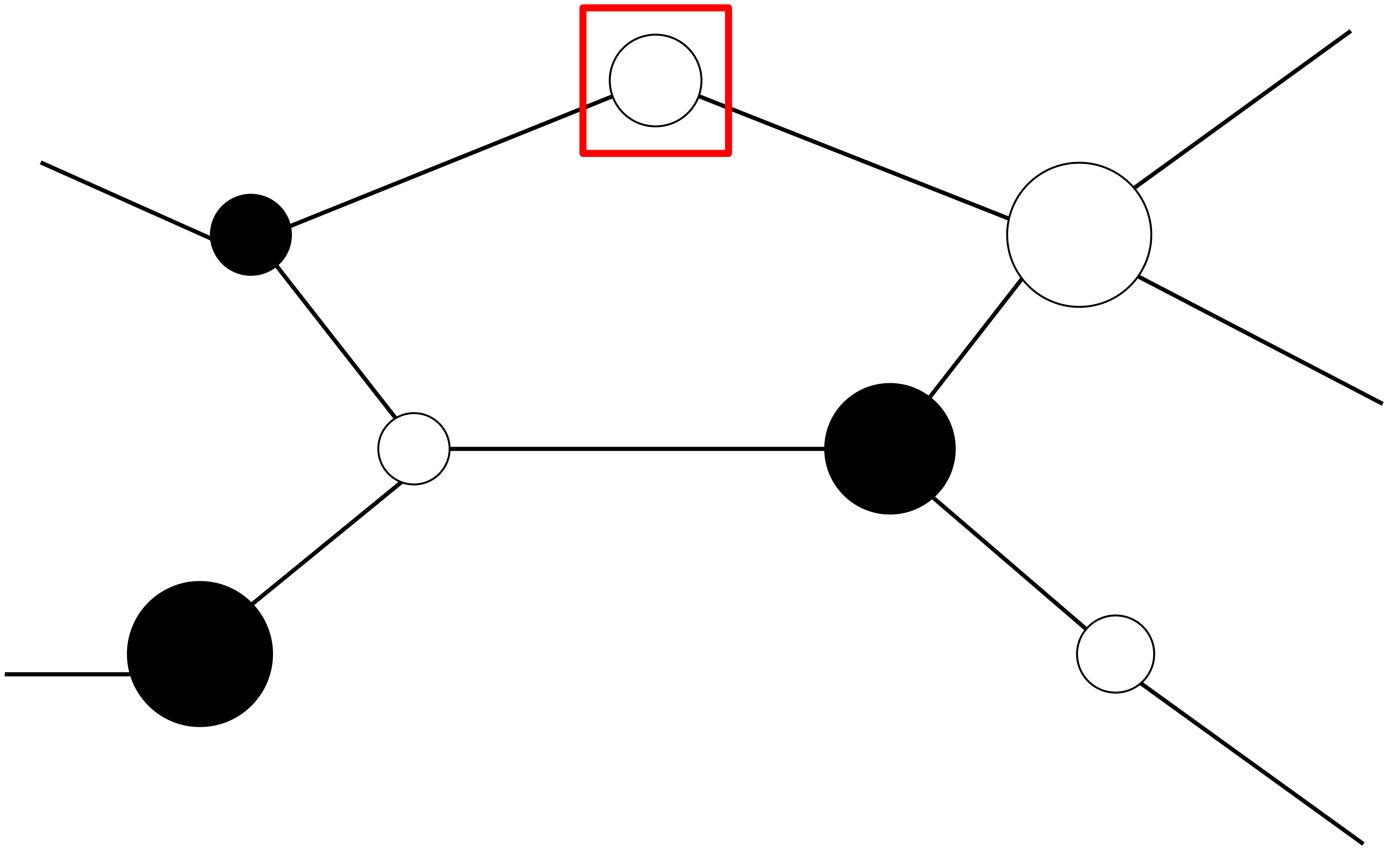
- **Sequential – one node at a time**

  → *always stabilizes!*

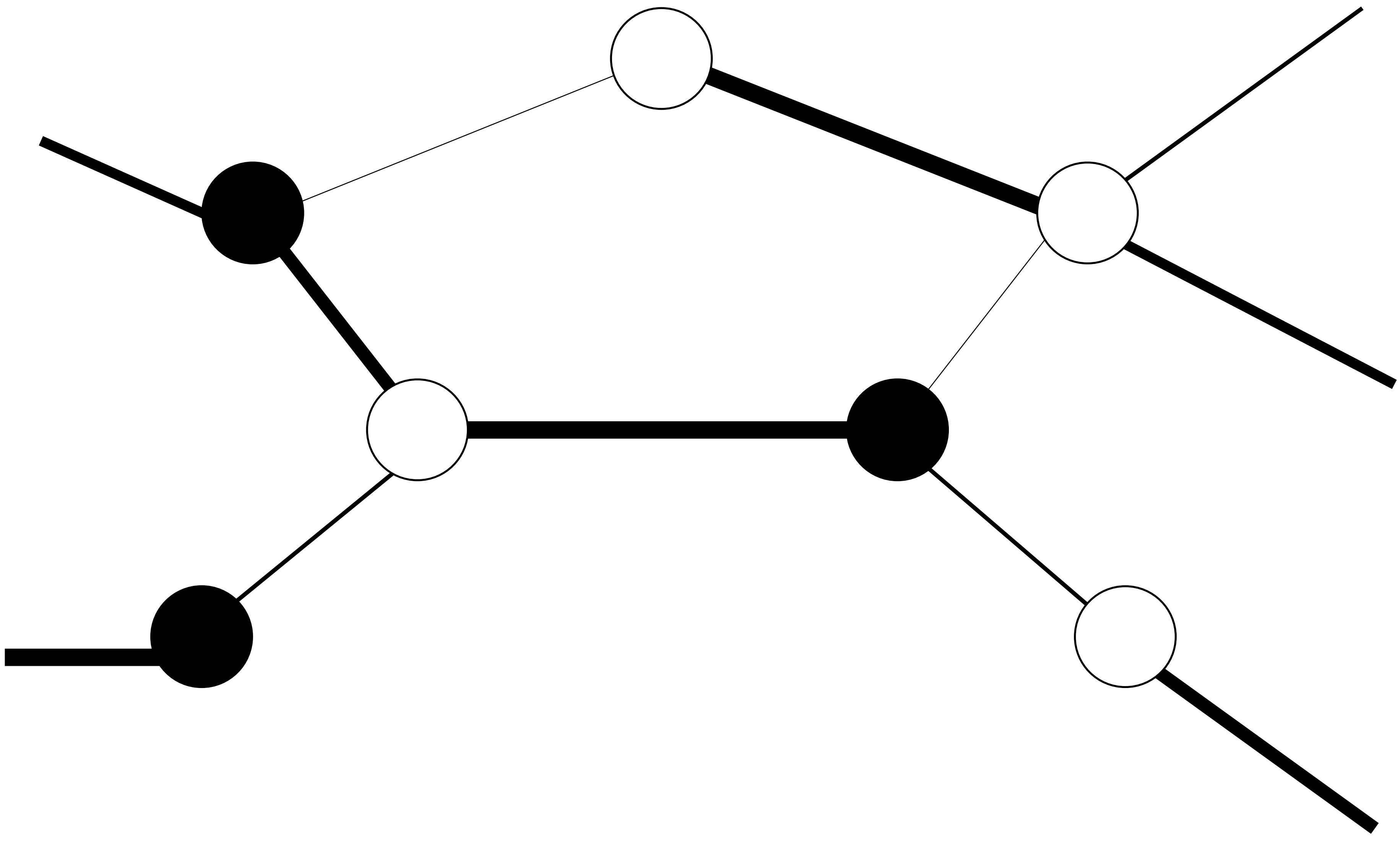
Weighted Minority Process



Weighted Minority Process

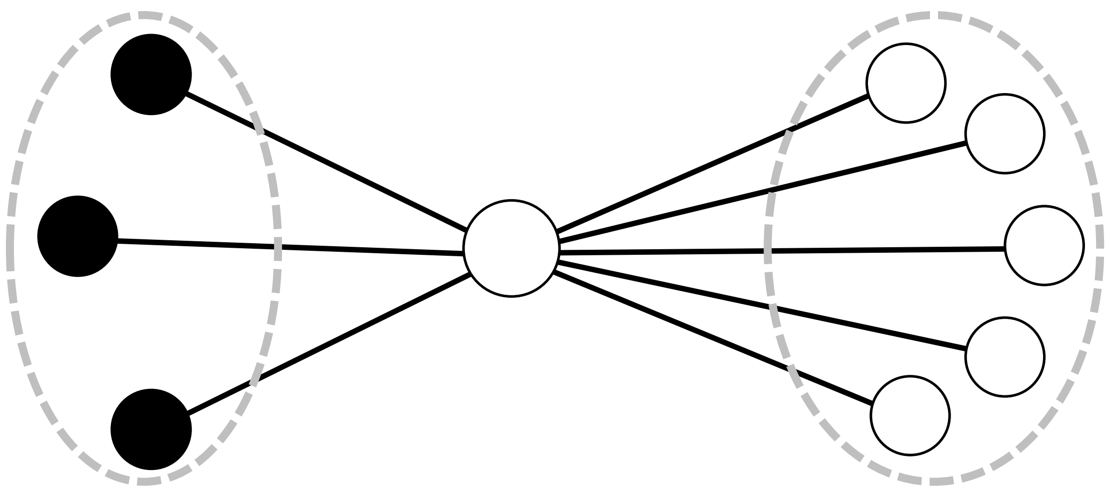


Weighted Minority Process



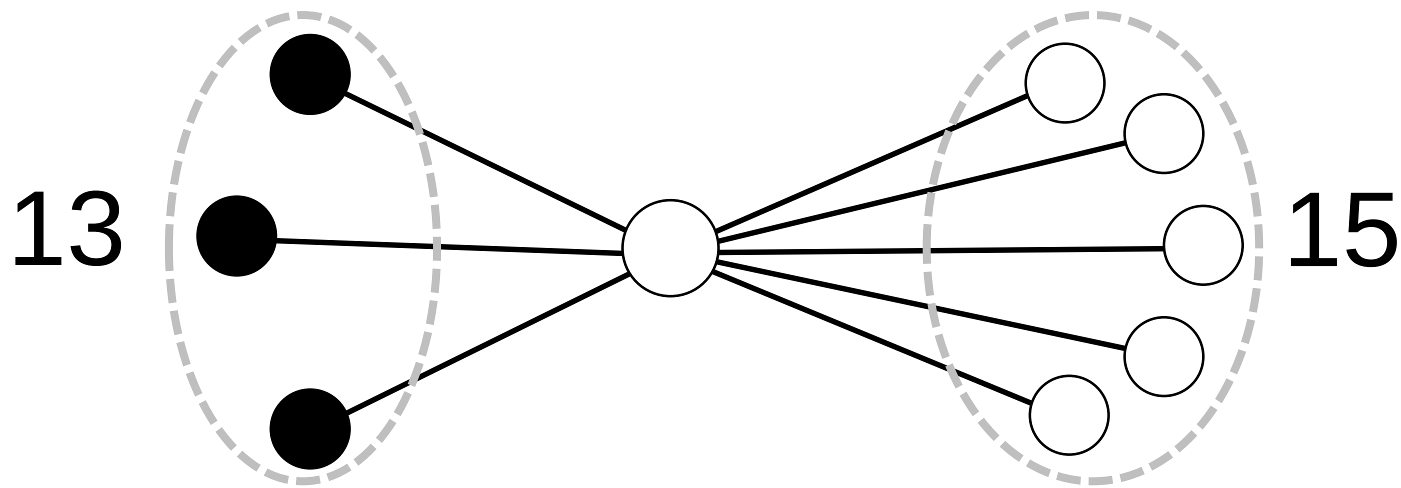
Switching Condition

- Basic – *for any improvement*



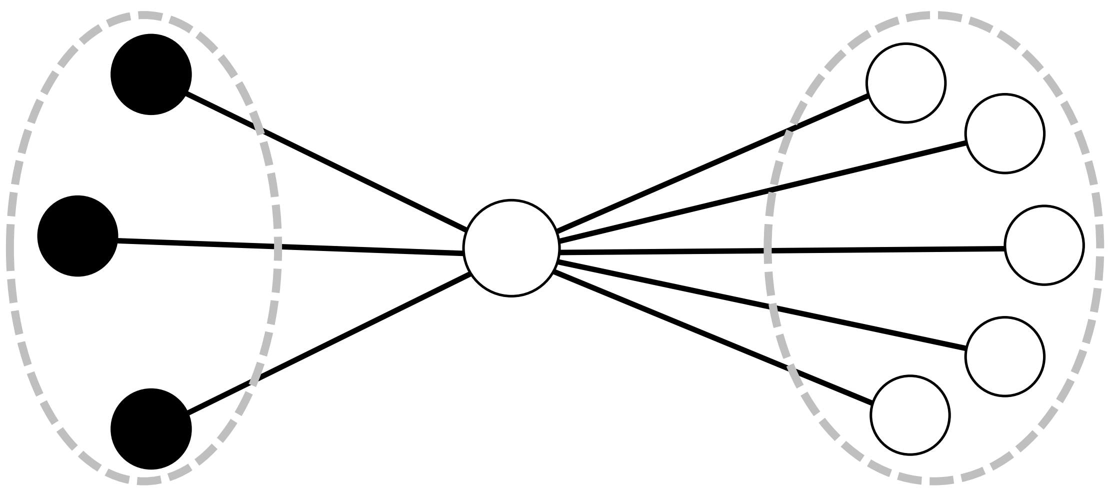
Switching Condition

- Basic – *for any improvement*



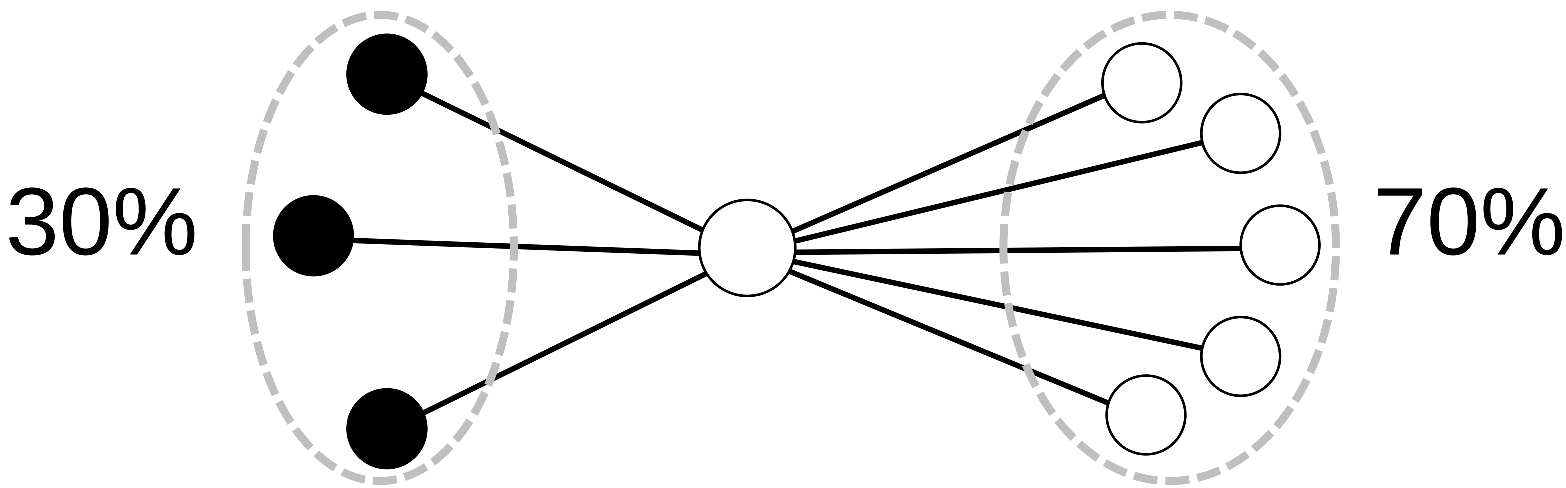
Switching Condition

- Basic – *for any improvement*
- Proportional – *conflict with ϱ portion of (weighted) neighborhood*



Switching Condition

- Basic – *for any improvement*
- Proportional – *conflict with* **ϱ** *portion of (weighted) neighborhood*



Stabilization time

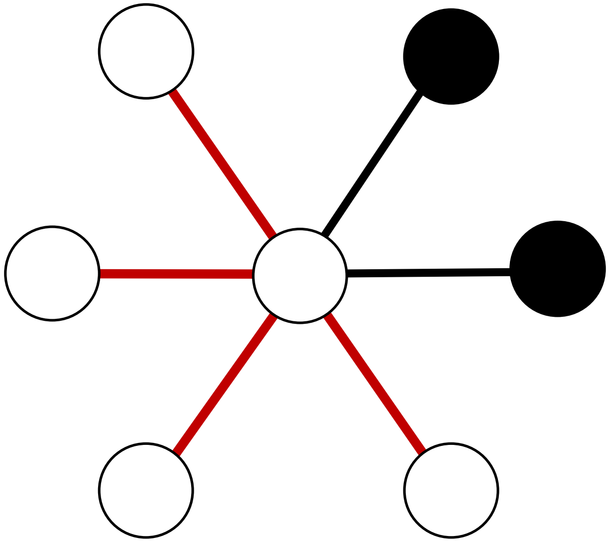
Upper bound:

- Sequential model: $O(2^n)$

Stabilization time

Upper bound:

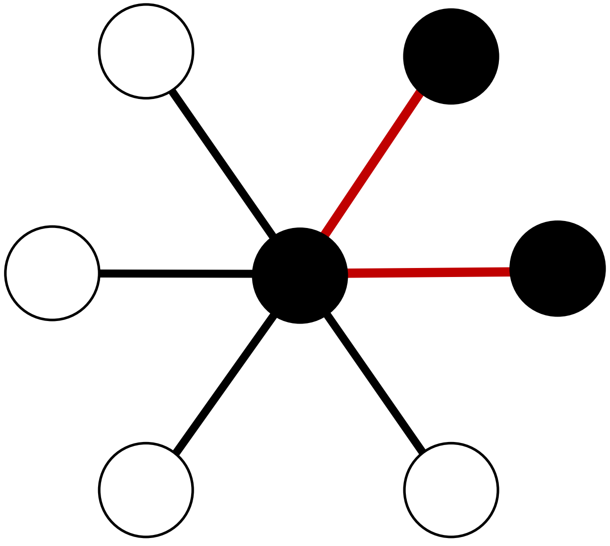
- Sequential model: O(2^n)



Stabilization time

Upper bound:

- Sequential model: $O(2^n)$



Stabilization time

Upper bound:

- Sequential model: $O(2^n)$

Lower bound?

Majority vs. Minority

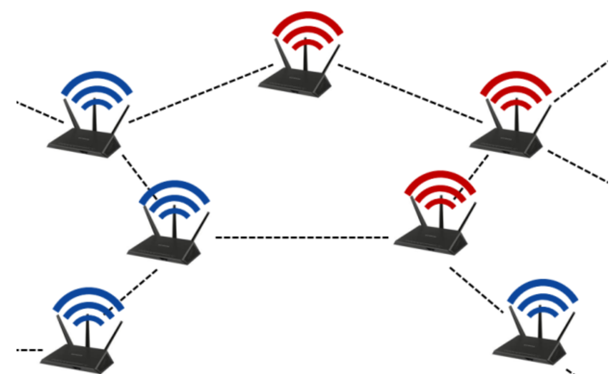
Majority process

(*take **most** frequent color*)

steps: $2^{\Theta(n)}$

Minority process

(*take **least** frequent color*)



steps: ?

Majority vs. Minority

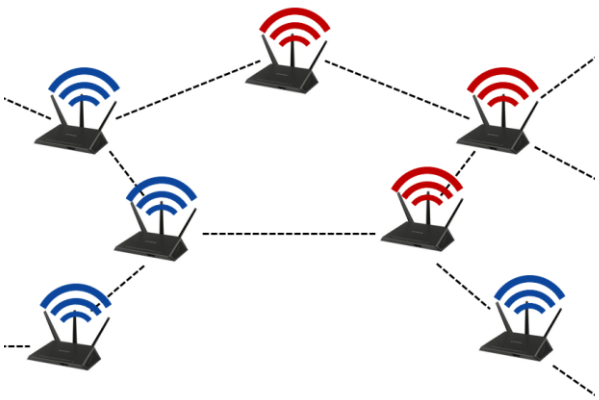
Majority process

(*take **most** frequent color*)

steps: $2^{\Theta(n)}$

Minority process

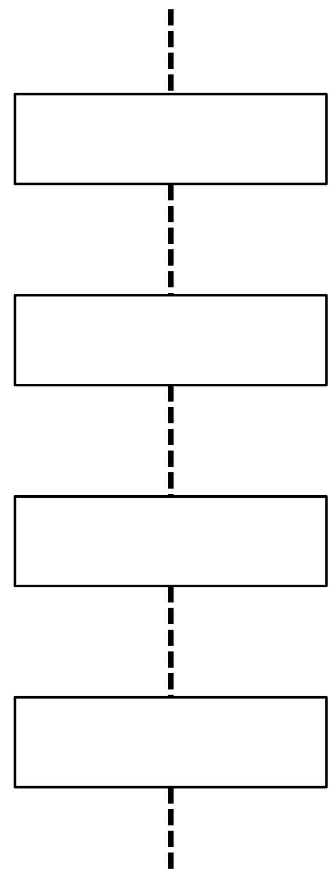
(*take **least** frequent color*)



steps: $\mathbf{2^{\Theta(n)}}$

Lower bound construction

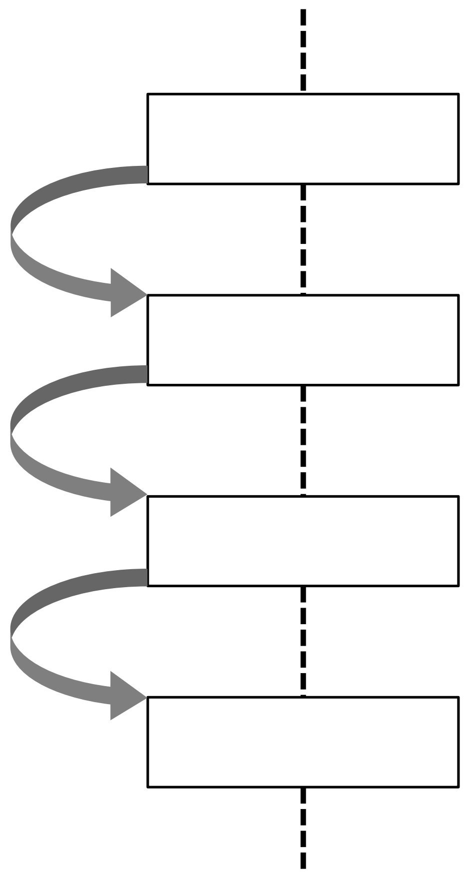
Higher weights
Less switches



Lower weights
More switches

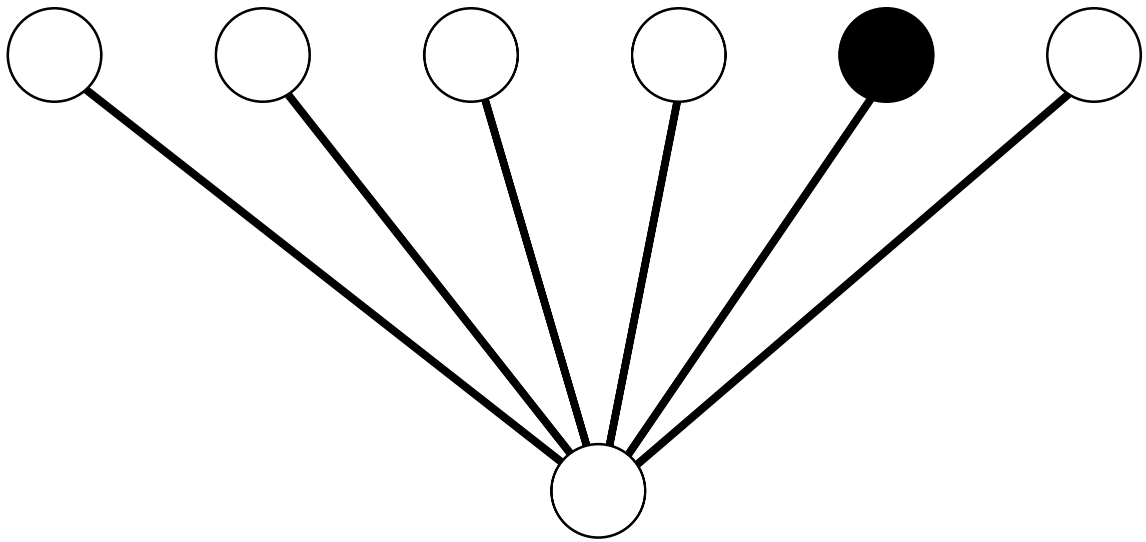
Lower bound construction

Higher weights
Less switches



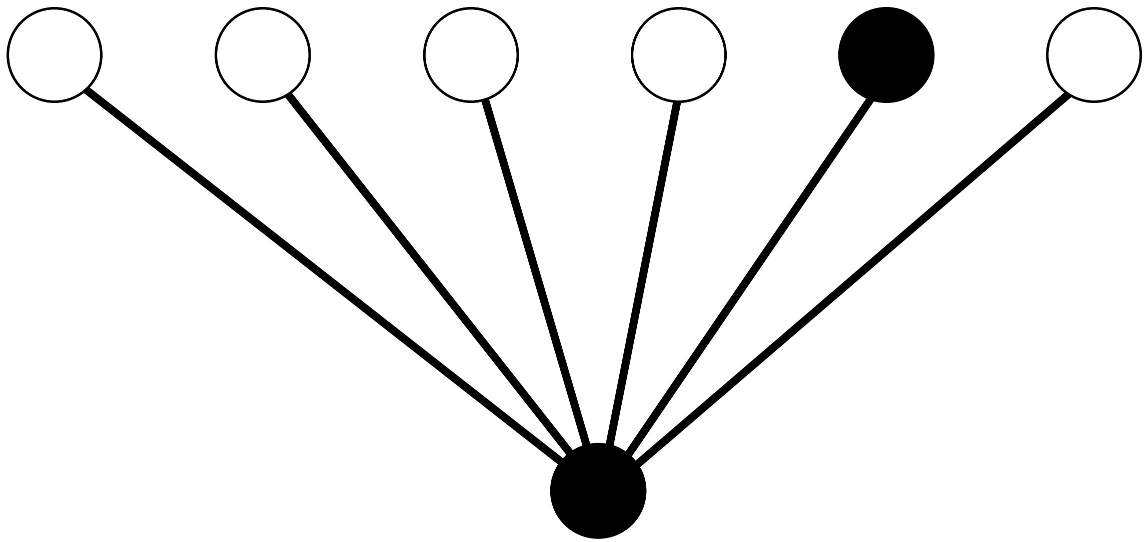
Lower weights
More switches

Lower bound construction



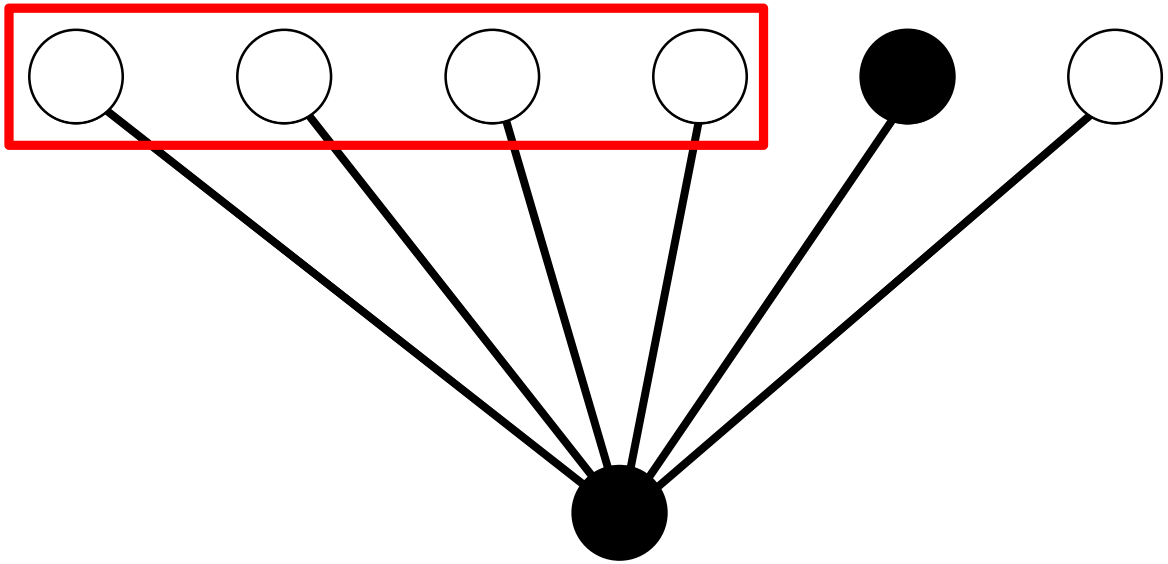
$$\varrho = \frac{5}{6}$$

Lower bound construction



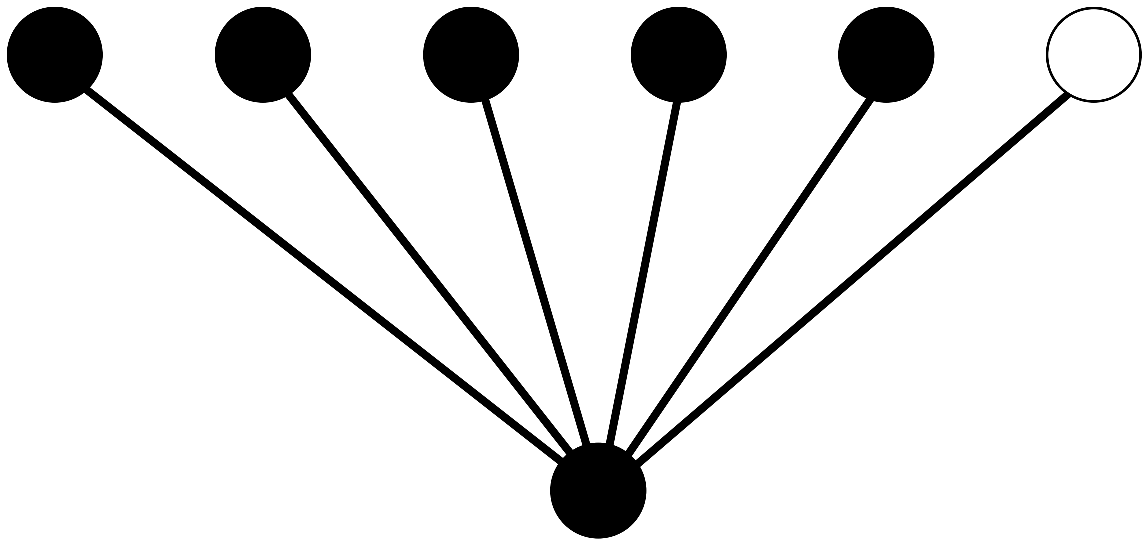
$\varrho = \frac{5}{6}$

Lower bound construction



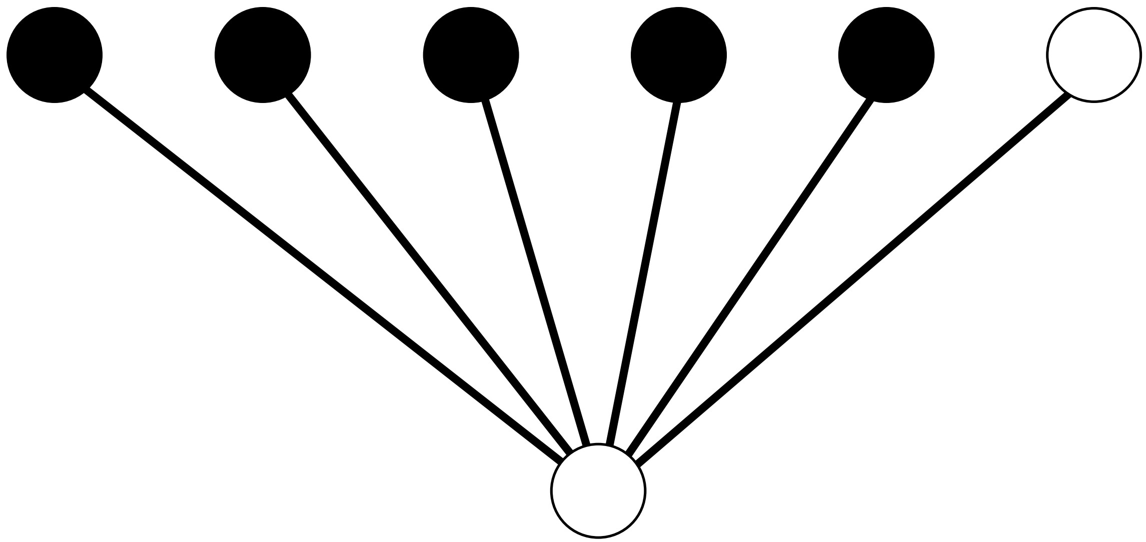
$$\varrho = \frac{5}{6}$$

Lower bound construction



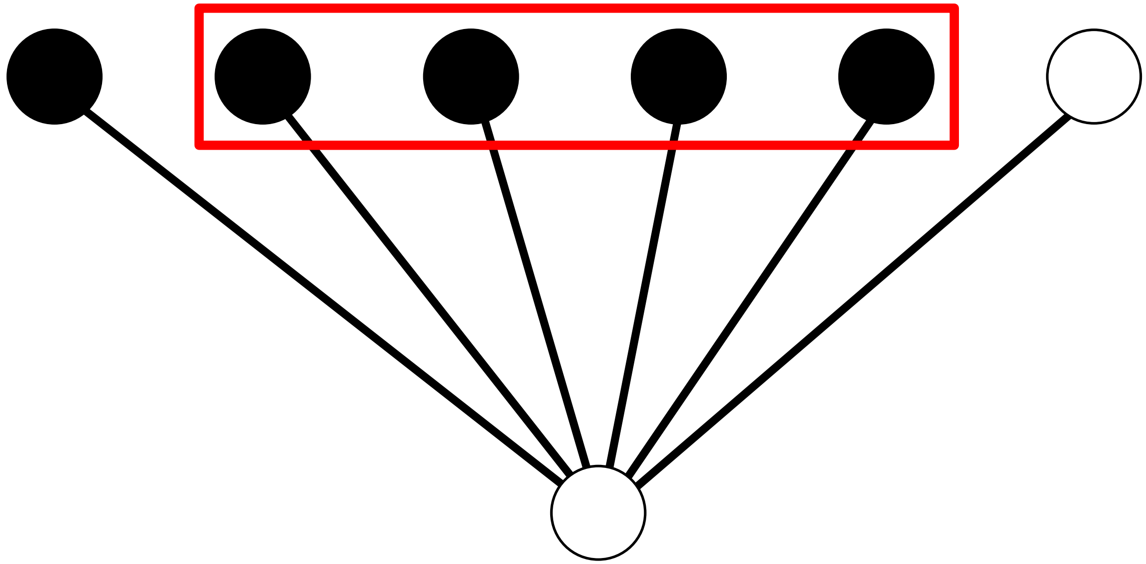
$$\varrho = \frac{5}{6}$$

Lower bound construction



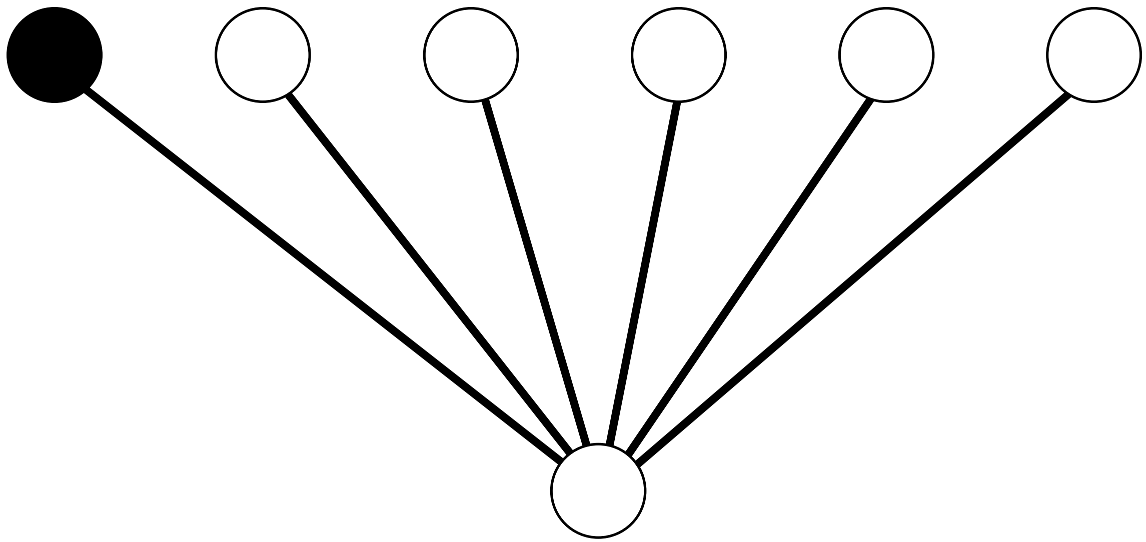
$$\varrho = \frac{5}{6}$$

Lower bound construction



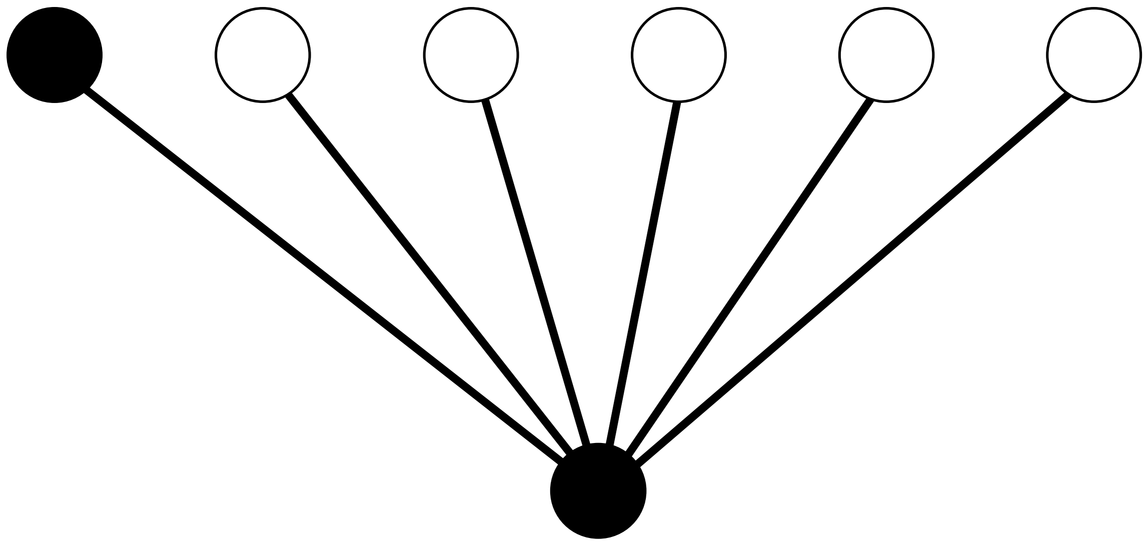
$$\varrho = \frac{5}{6}$$

Lower bound construction



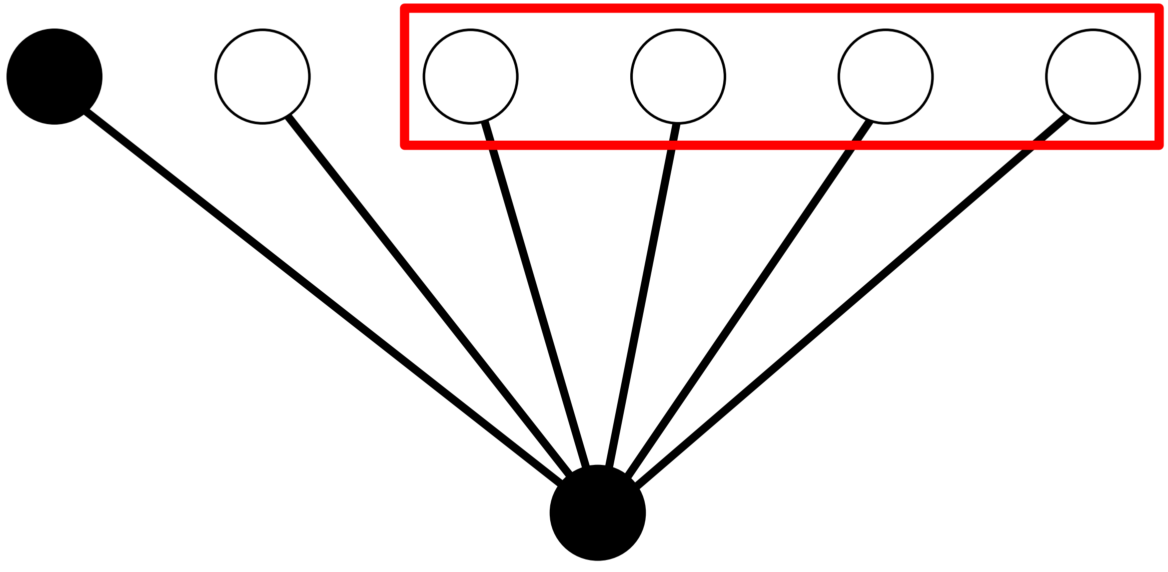
$$\varrho = \frac{5}{6}$$

Lower bound construction



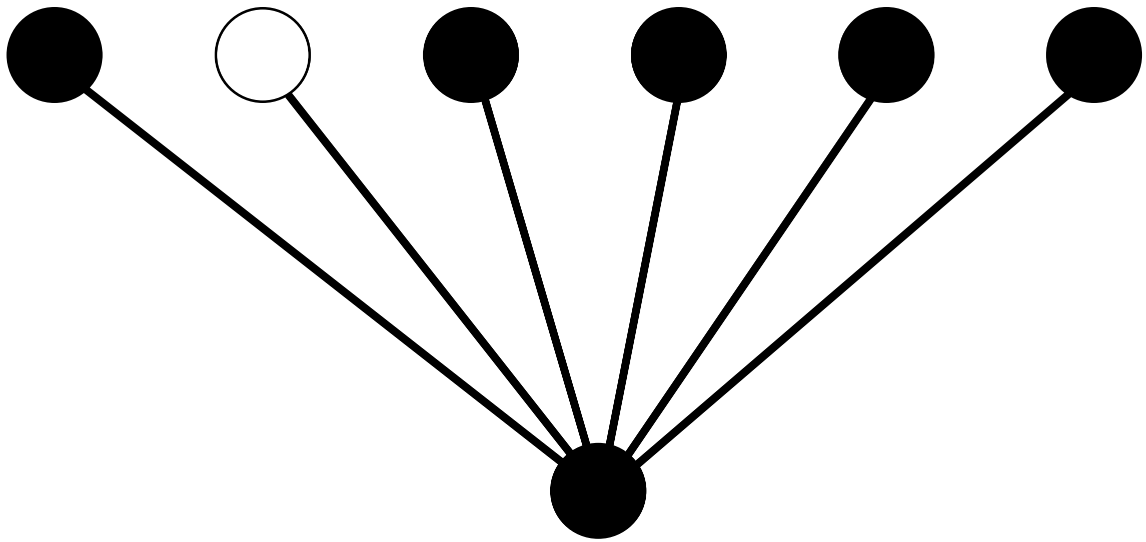
$$\varrho = \frac{5}{6}$$

Lower bound construction



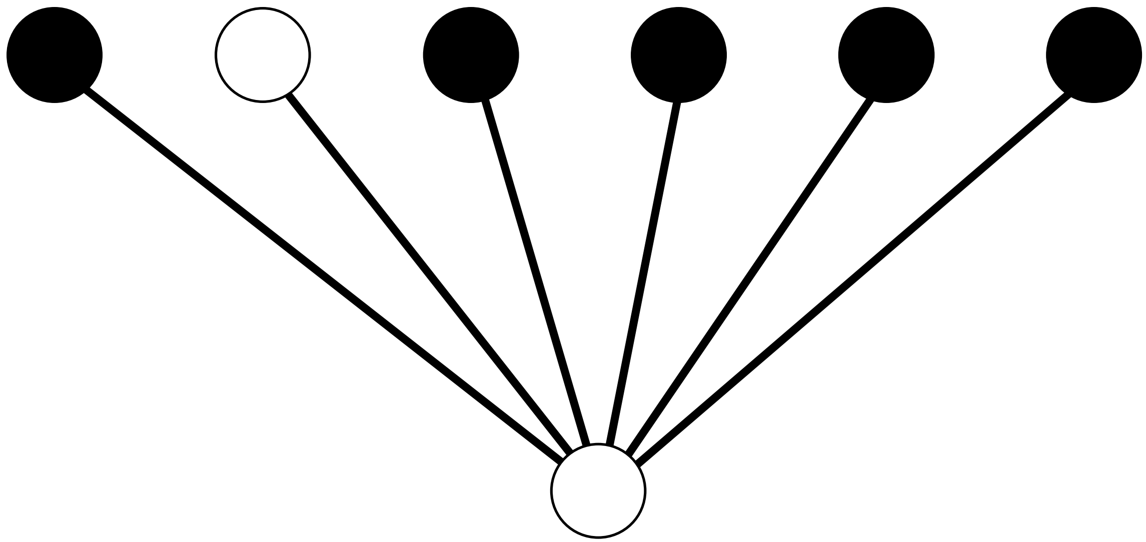
$$\varrho = \frac{5}{6}$$

Lower bound construction



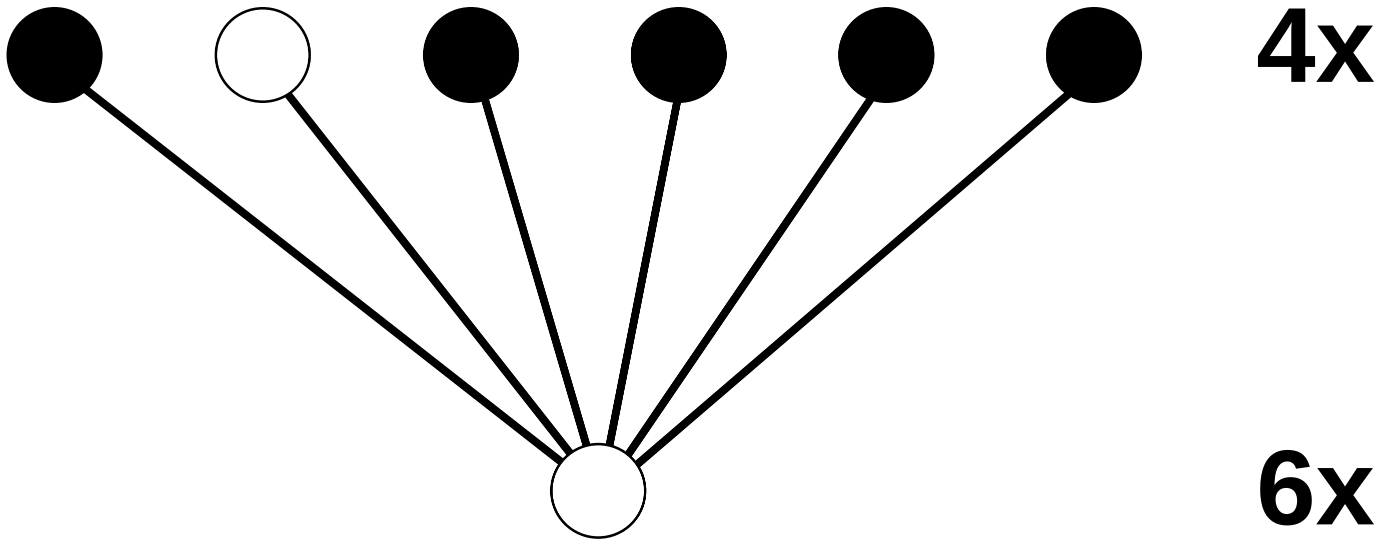
$$\varrho = \frac{5}{6}$$

Lower bound construction



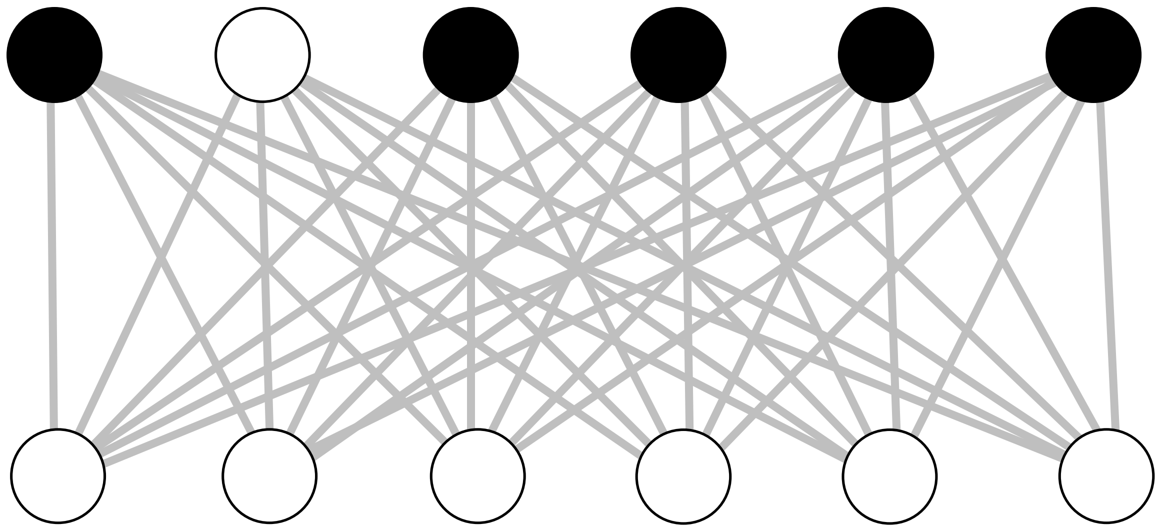
$$\varrho = \frac{5}{6}$$

Lower bound construction



$$\varrho = \frac{5}{6}$$

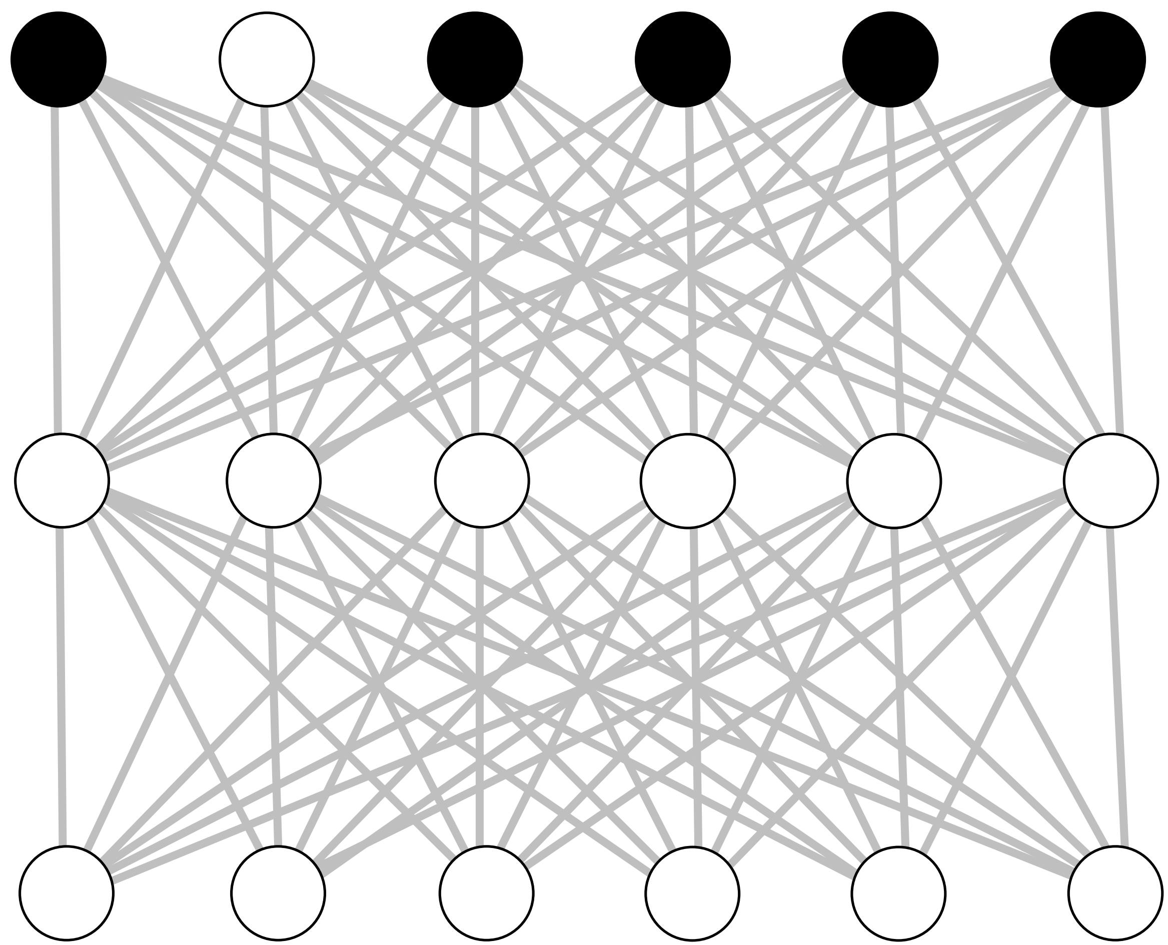
Lower bound construction



$$\varrho = \frac{5}{6}$$

Lower bound construction

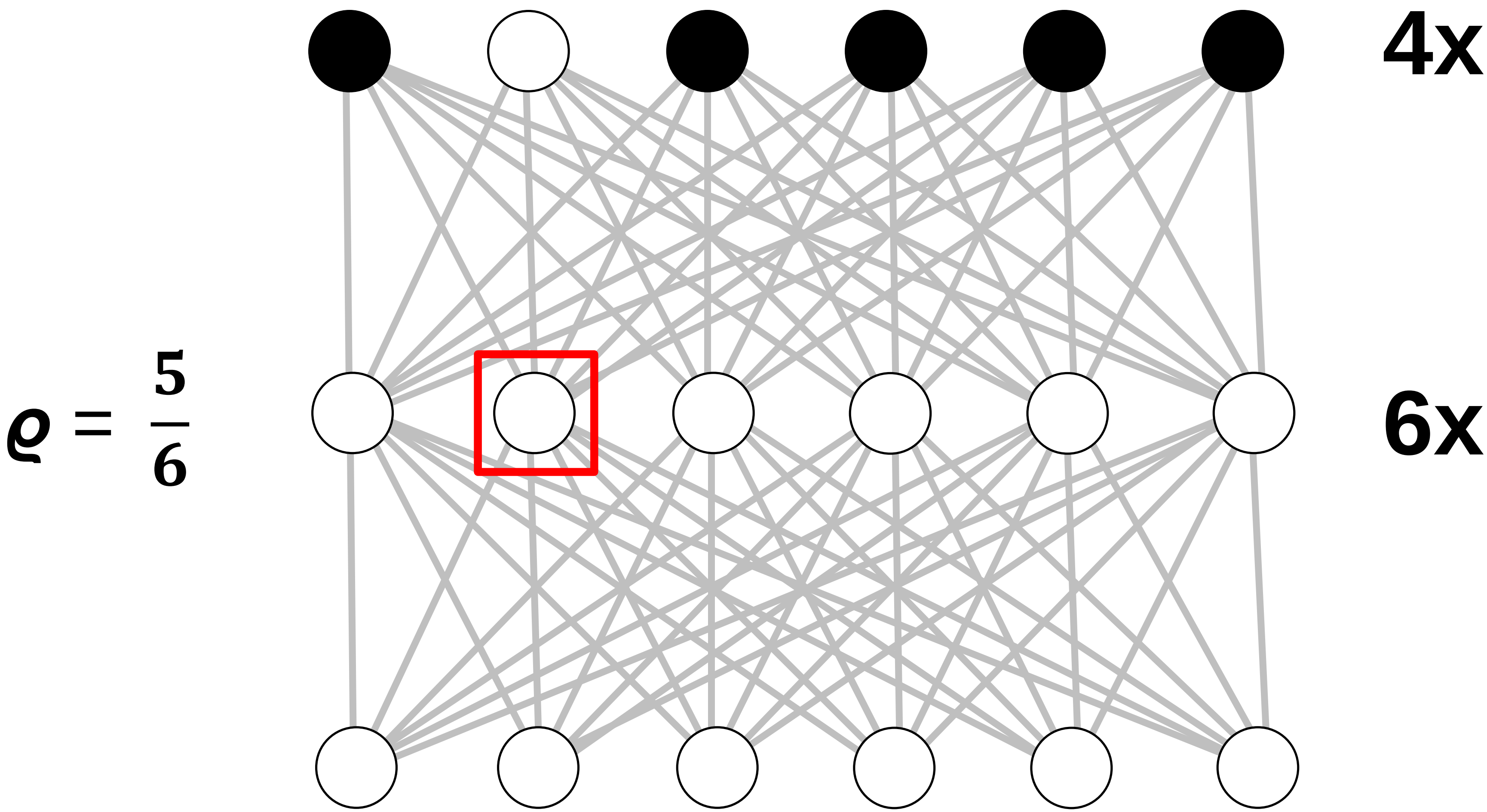
$$\varrho = \frac{5}{6}$$



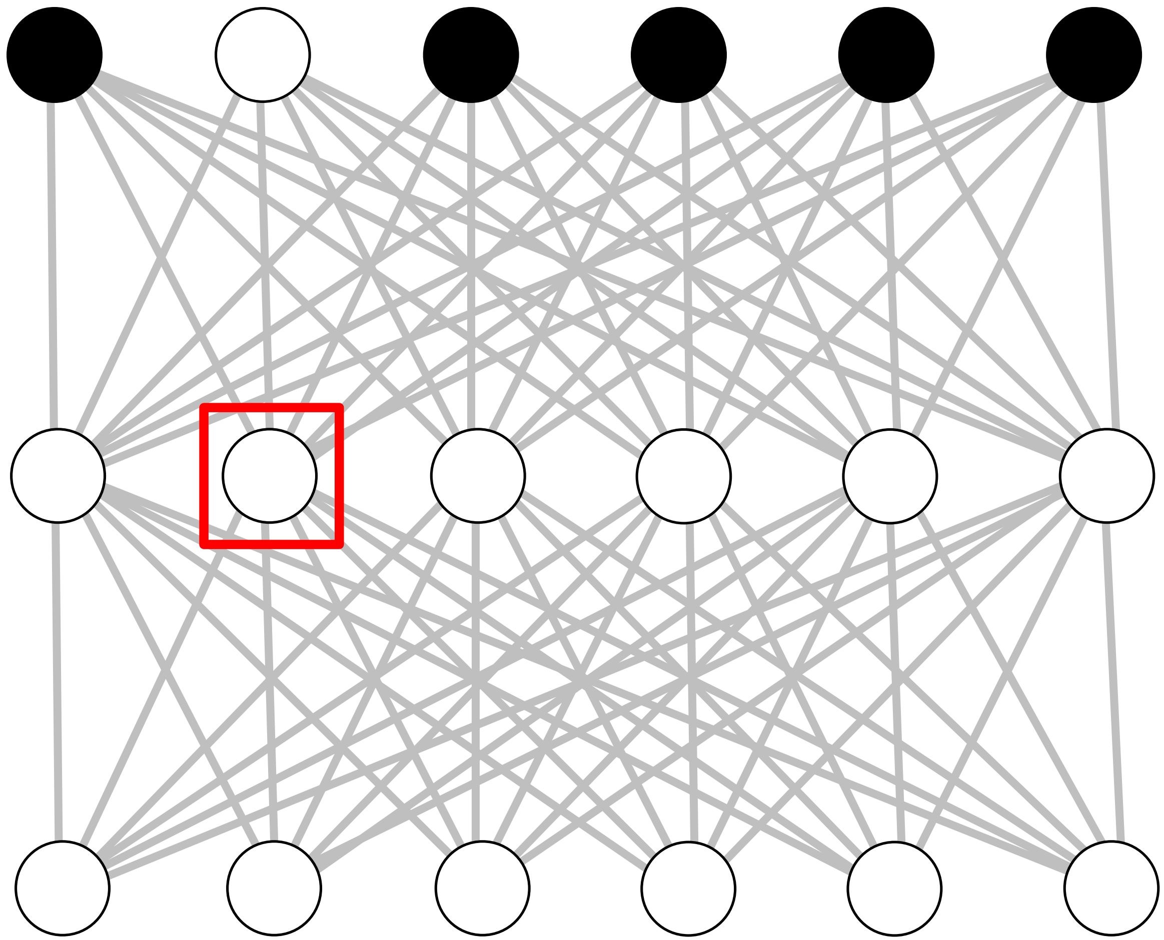
4x

6x

Lower bound construction



Lower bound construction



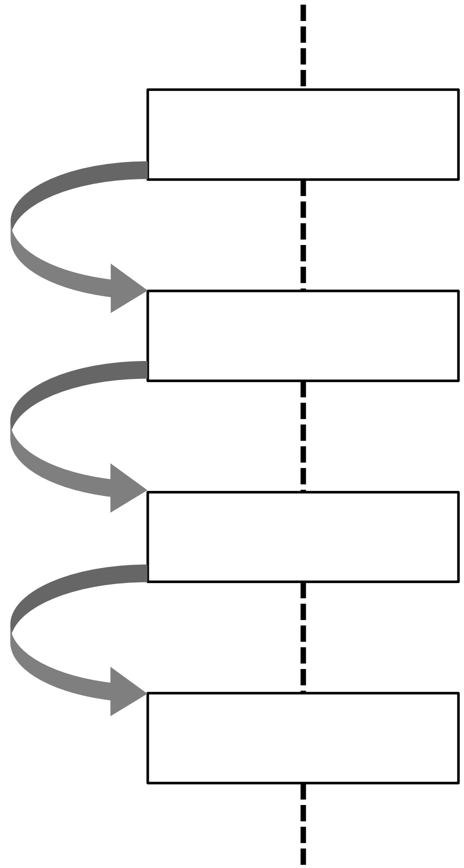
$\varrho = \frac{5}{6} - \varepsilon$

4x

6x

Lower bound construction

↑ Higher weights
Less switches



s

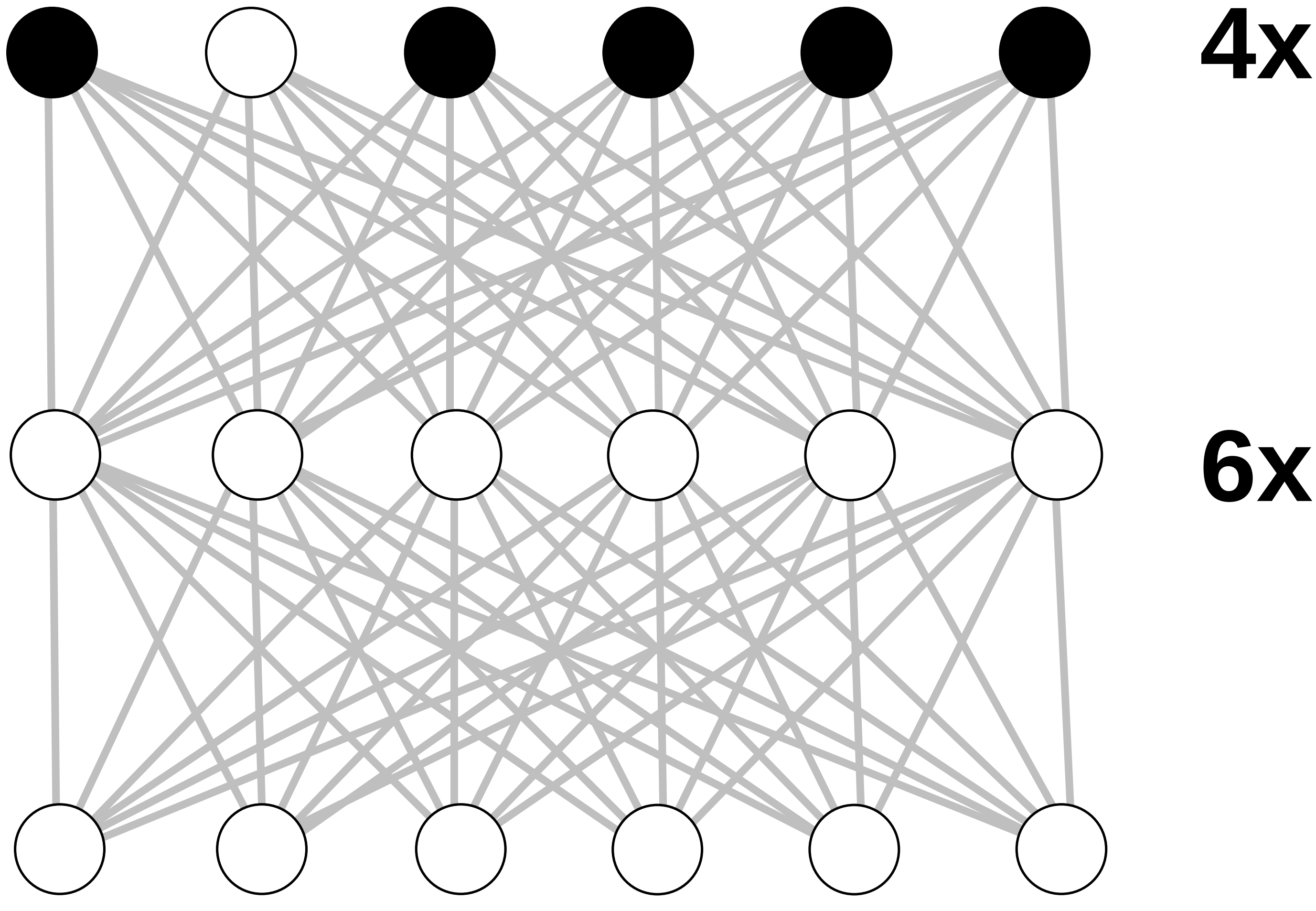
$s \cdot \frac{6}{4}$

$s \cdot \left(\frac{6}{4}\right)^2$

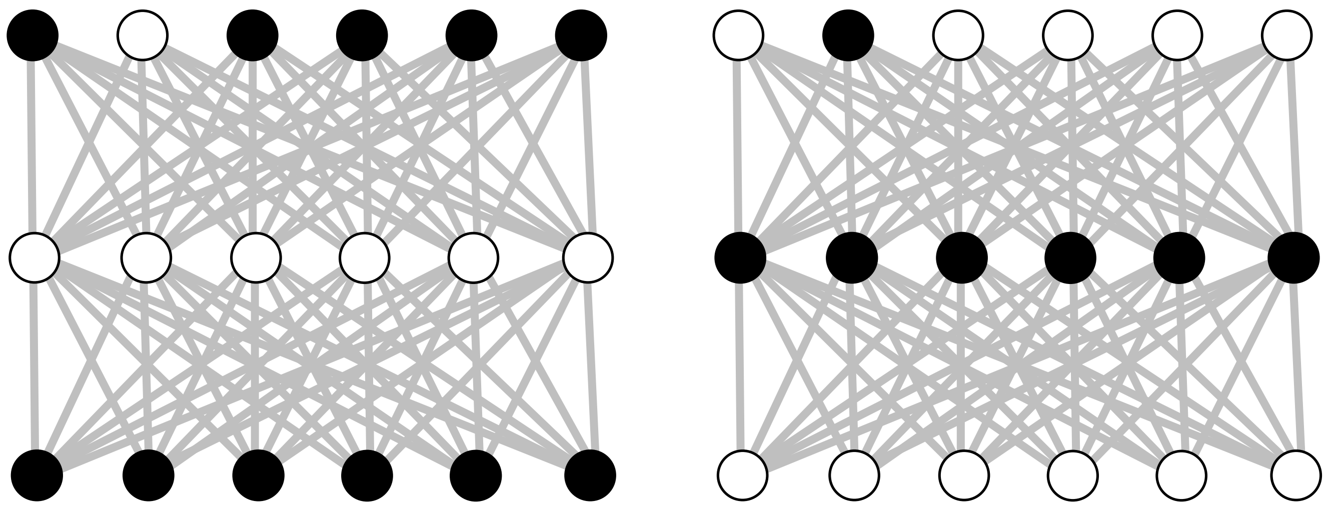
$s \cdot \left(\frac{6}{4}\right)^3$

↓ Lower weights
More switches

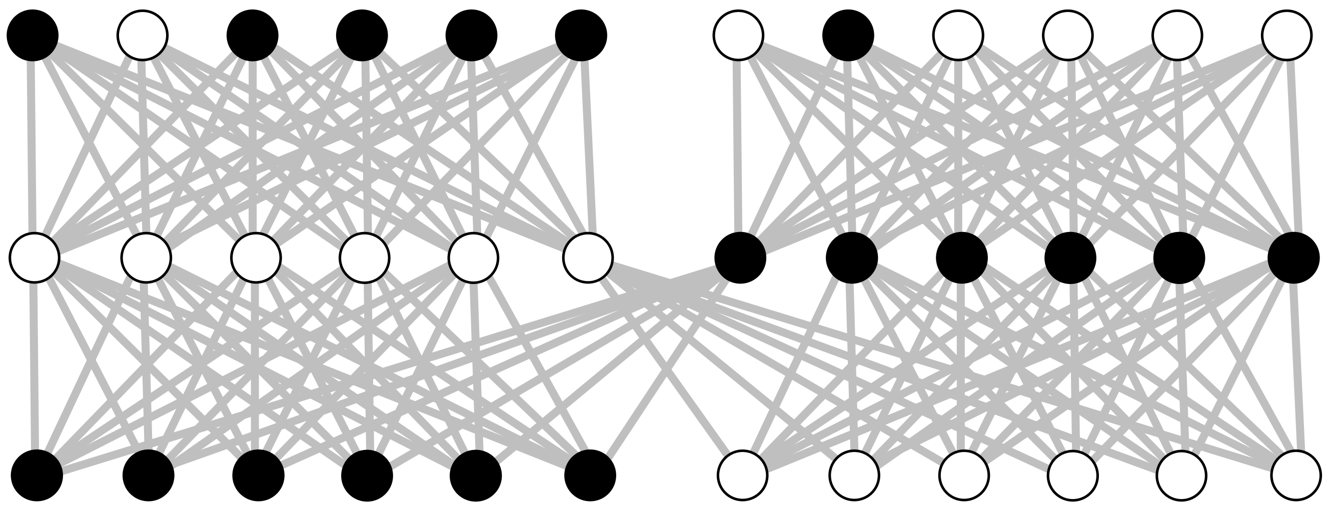
Lower bound construction



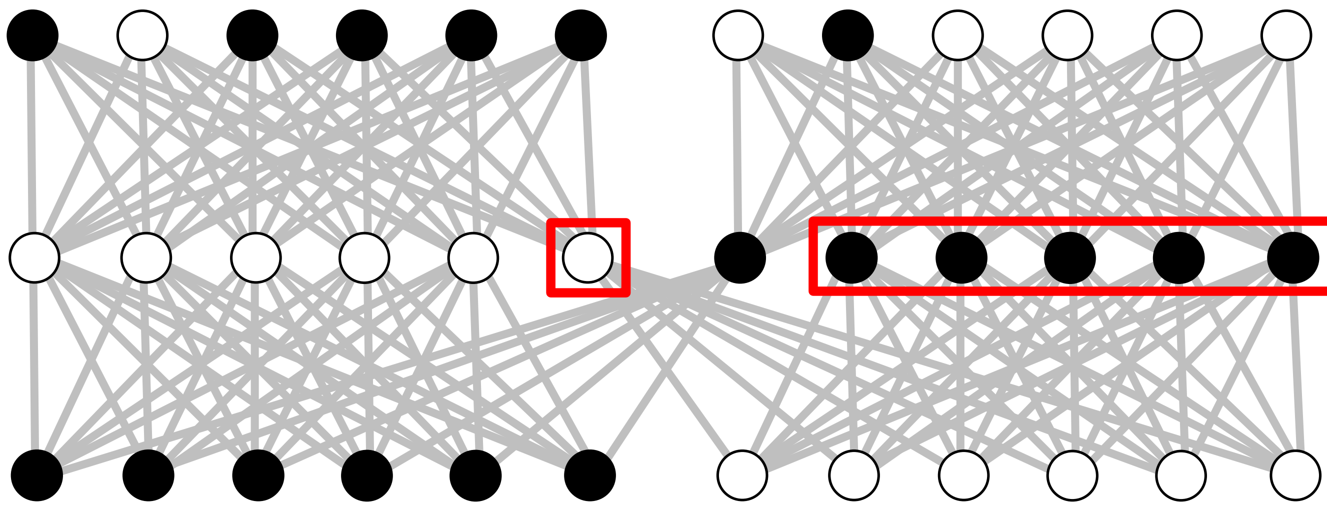
Lower bound construction



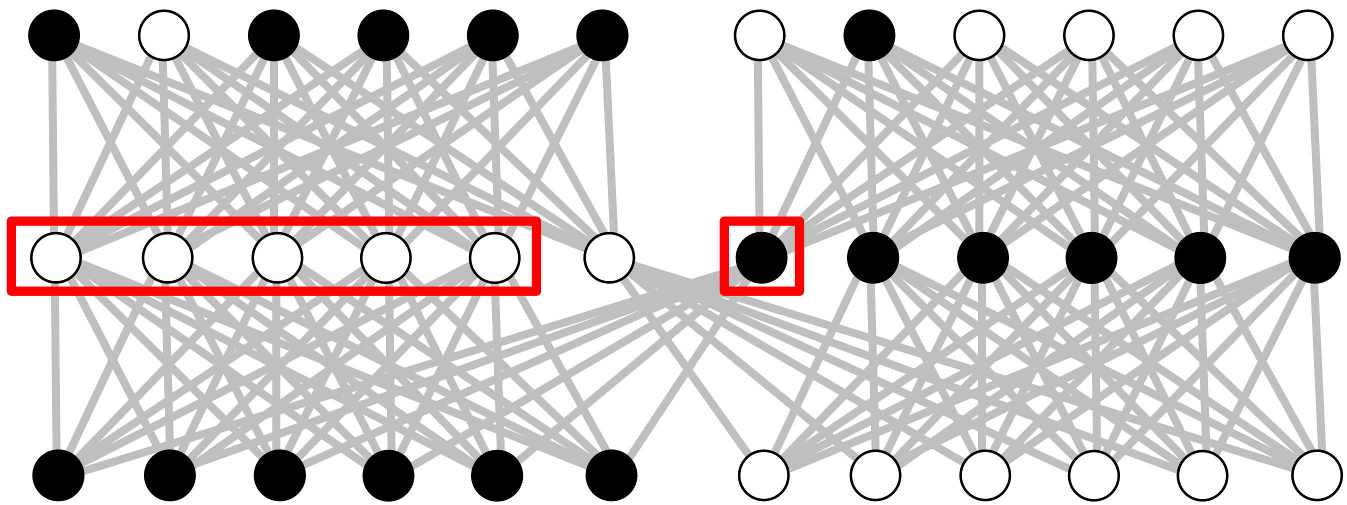
Lower bound construction



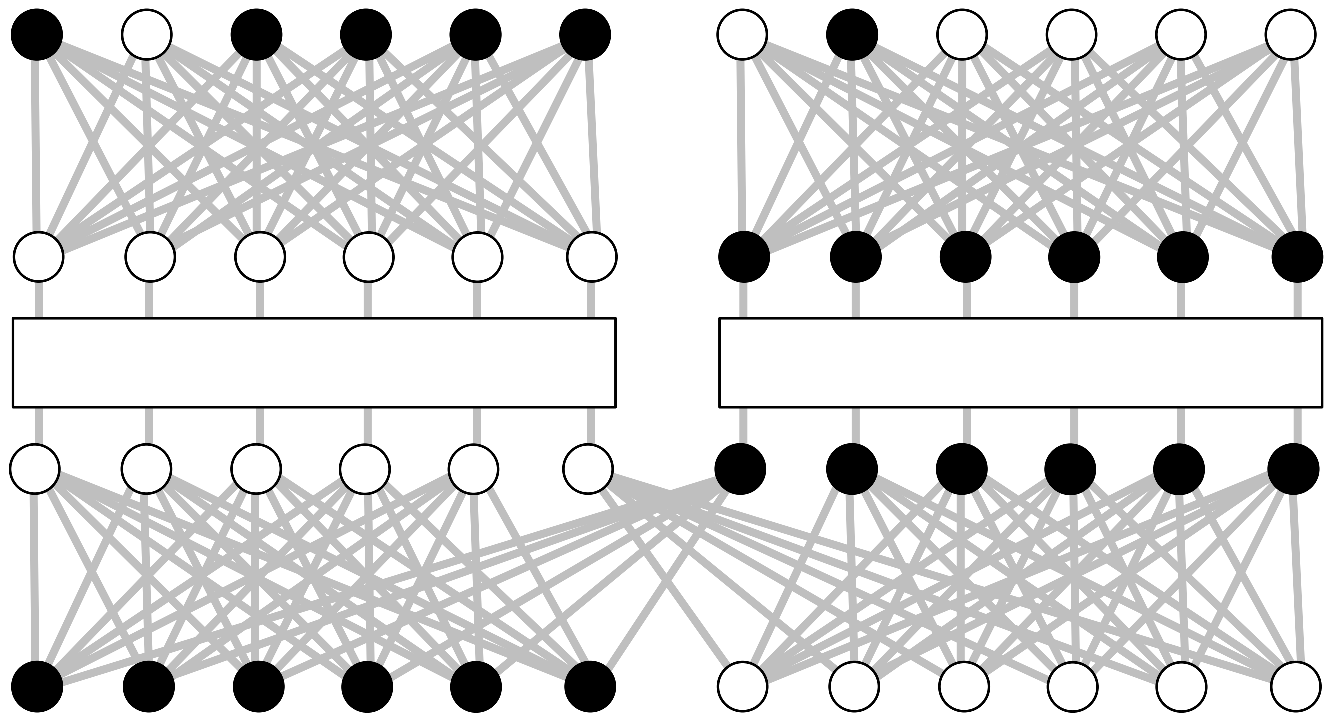
Lower bound construction



Lower bound construction



Lower bound construction

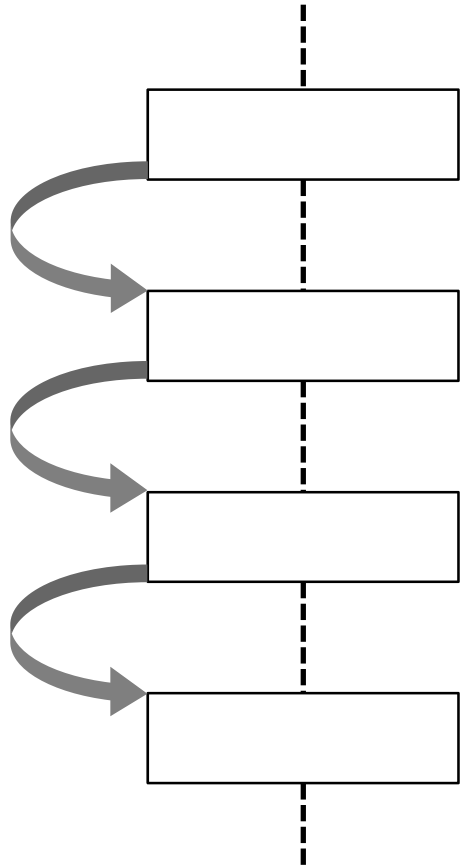


Lower bound construction

$O(1)$ level size

$\Theta(n)$ levels

$\left(\frac{6}{4}\right)^{\Theta(n)}$ switches



s

$s \cdot \frac{6}{4}$

$s \cdot \left(\frac{6}{4}\right)^2$

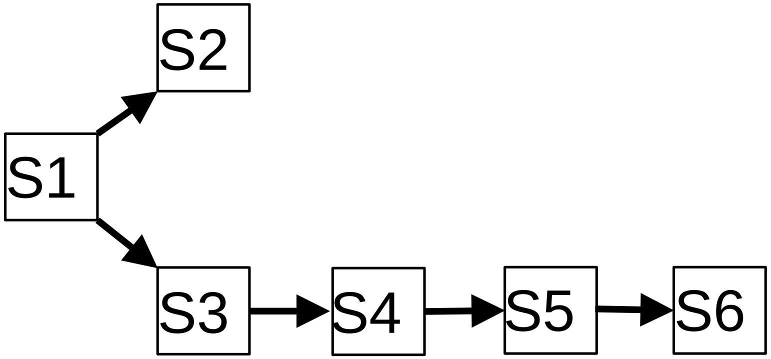
$s \cdot \left(\frac{6}{4}\right)^3$

Sequential model – order of nodes?

- **Adversarial:** a specific sequence

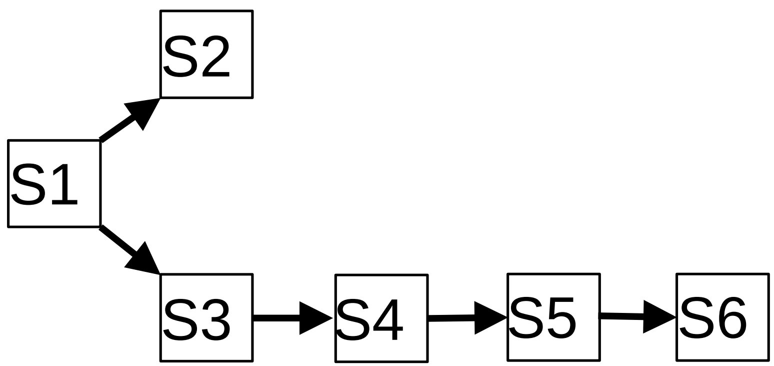
Sequential model – order of nodes?

- **Adversarial:** a specific sequence



Sequential model – order of nodes?

- **Adversarial:** a specific sequence
- **Benevolent**



Sequential model – order of nodes?

- **Adversarial:** a specific sequence

- **Benevolent:** every sequence

Sequential model – order of nodes?

- **Adversarial:** a specific sequence
- **Benevolent:** every sequence

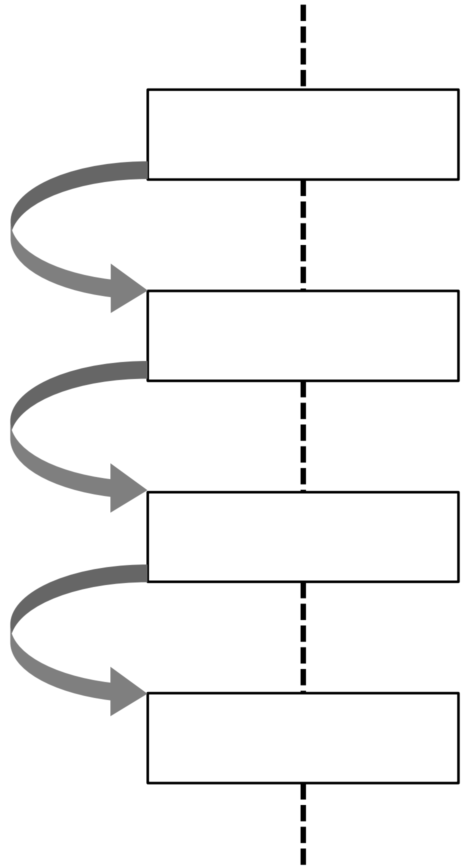
  → *allow only one possible sequence!*

Benevolent case

$O(1)$ level size

$\Theta(n)$ levels

$\left(\frac{6}{4}\right)^{\Theta(n)}$ switches



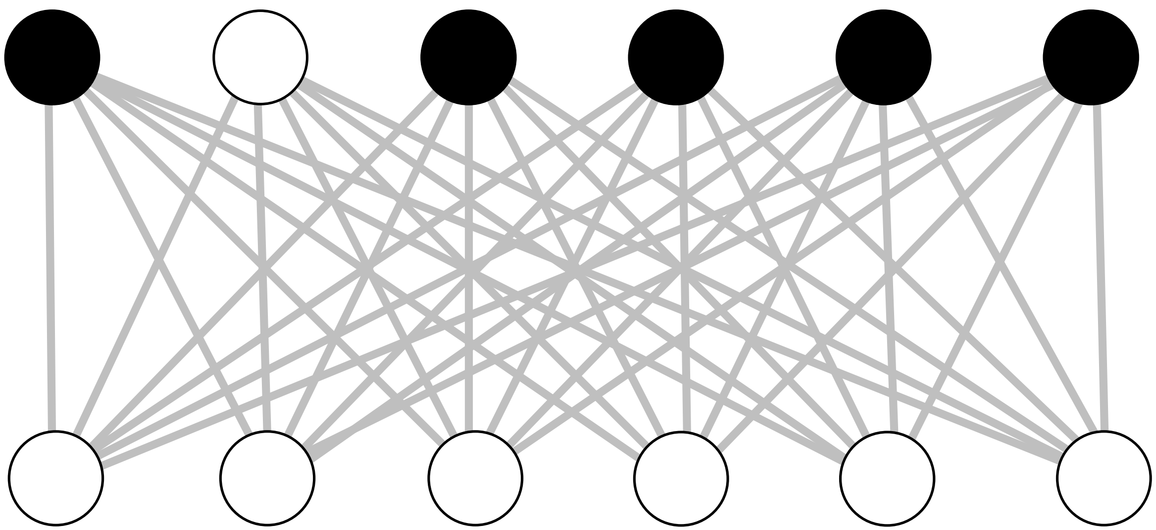
s

$s \cdot \frac{6}{4}$

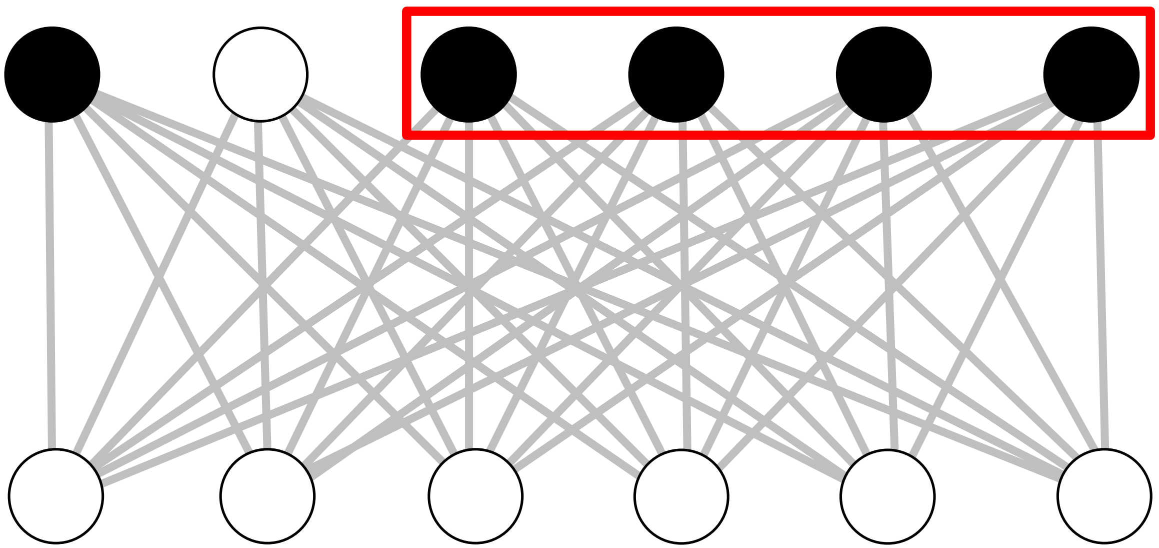
$s \cdot \left(\frac{6}{4}\right)^2$

$s \cdot \left(\frac{6}{4}\right)^3$

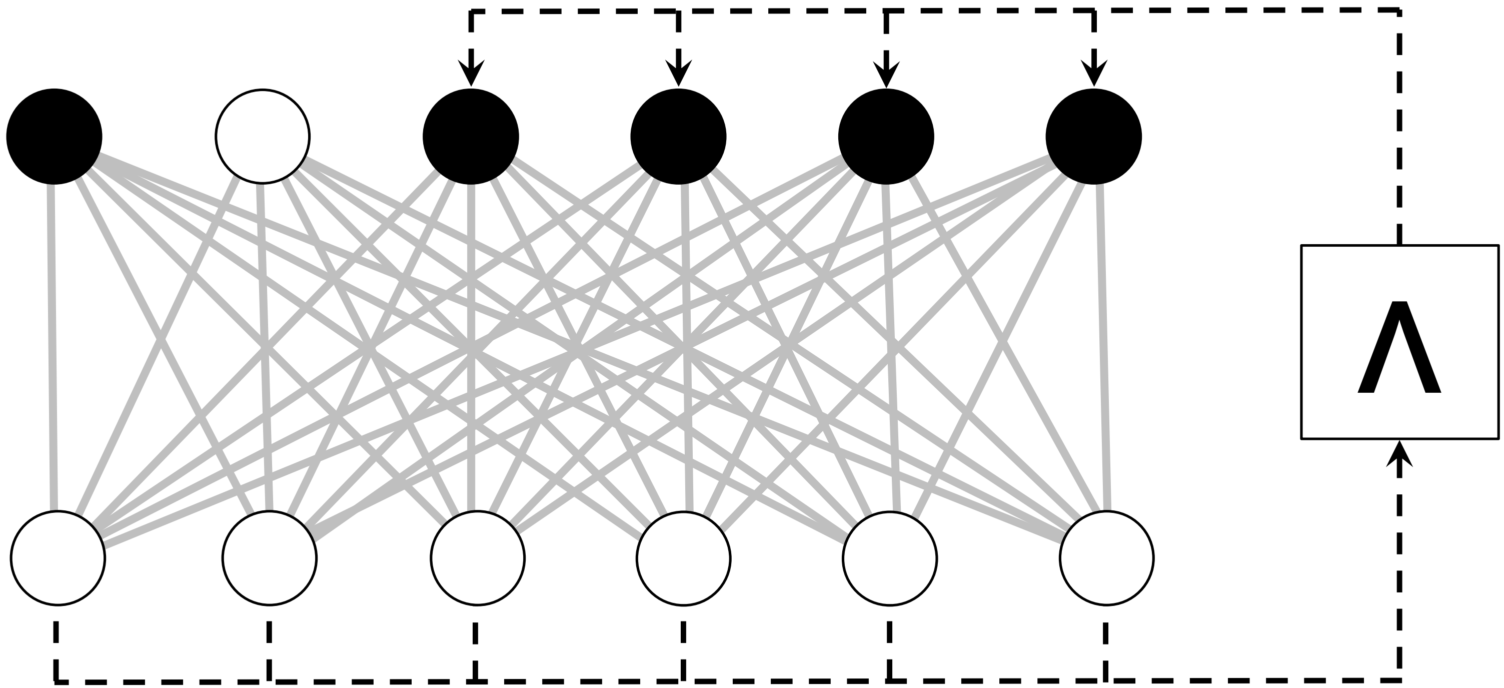
Benevolent case – logical gates



Benevolent case – logical gates



Benevolent case – logical gates



More Variants by [Papp, W]

Unweighted → $\Theta(n^2)$

Random initialization → $\Theta(n^2)$

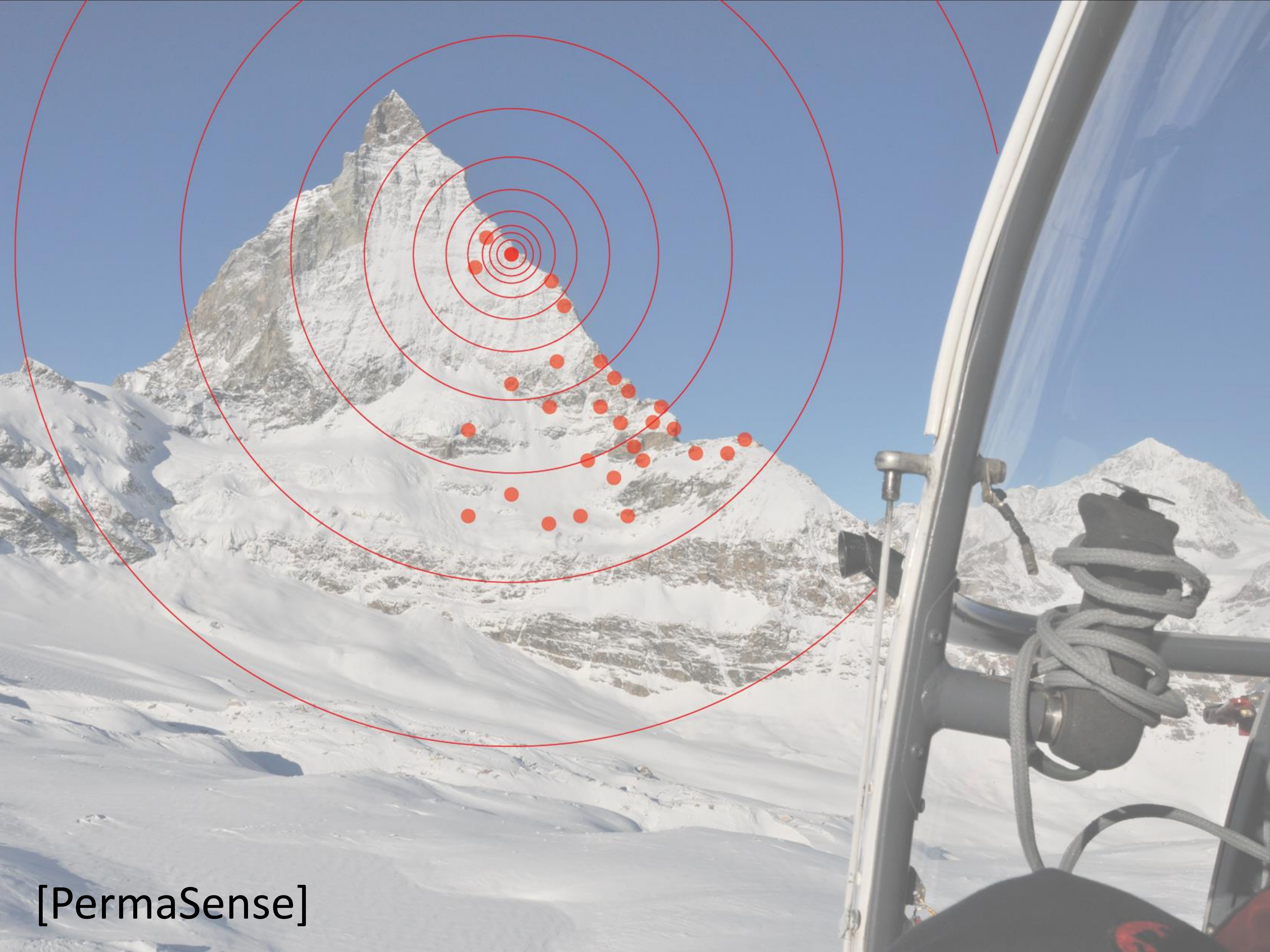
Random init & proportional → depends on ϱ

More general model → $f(\lambda) := \max_{\varphi \in (0, \frac{1-\lambda}{2}]} \dfrac{\log\left(\frac{1-\varphi}{\lambda+\varphi}\right)}{\log\left(\frac{1-\varphi}{\varphi}\right)}$

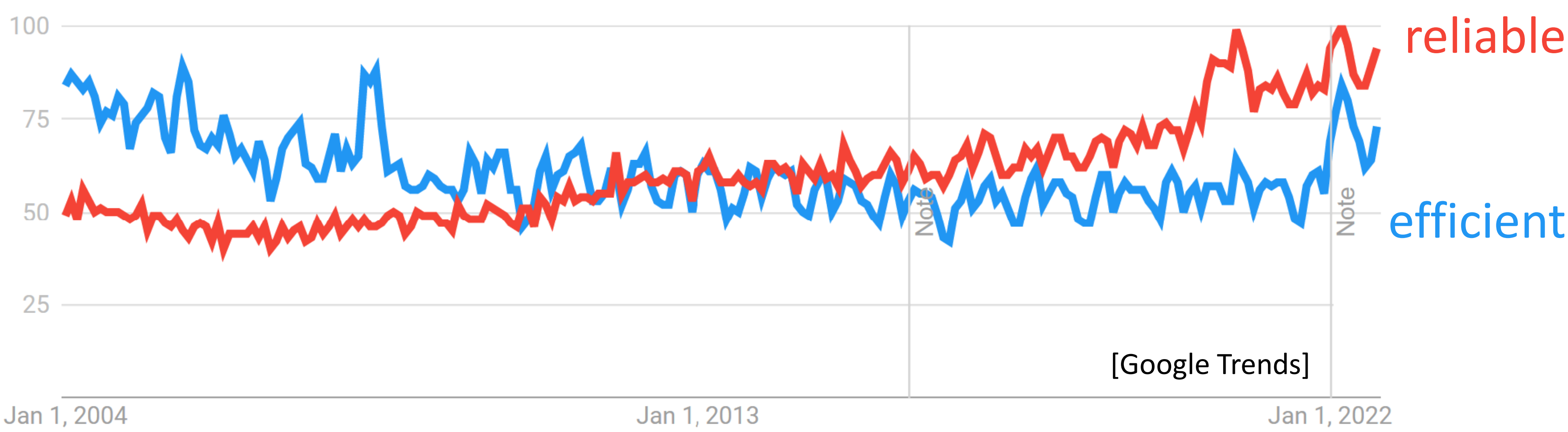
Sensor Networks

Algorithms for Sensor Networks
…What Is It Good For?!

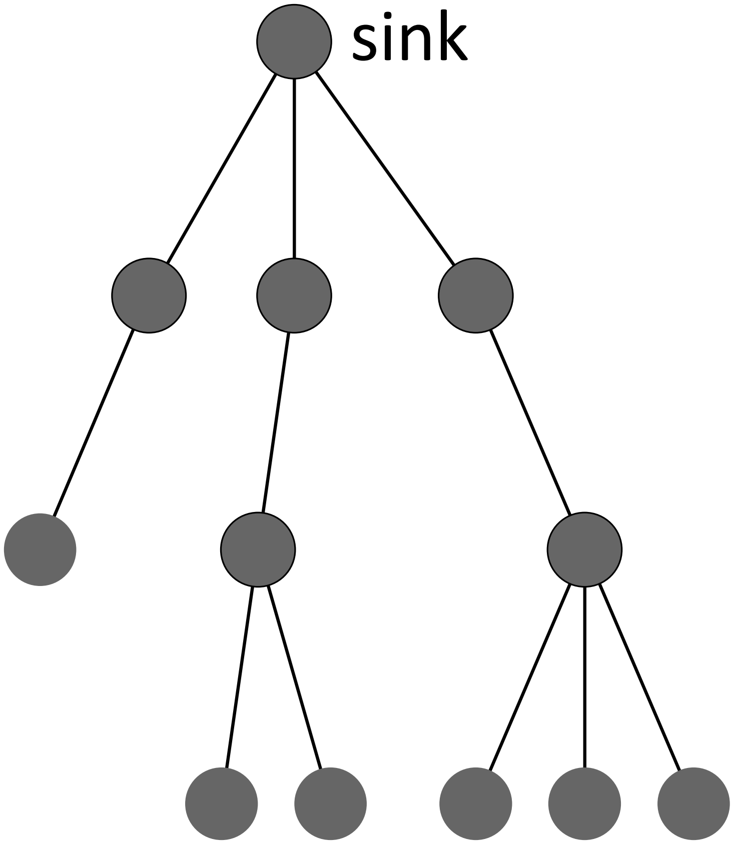
[W, Algosensors 2008!]


[PermaSense]

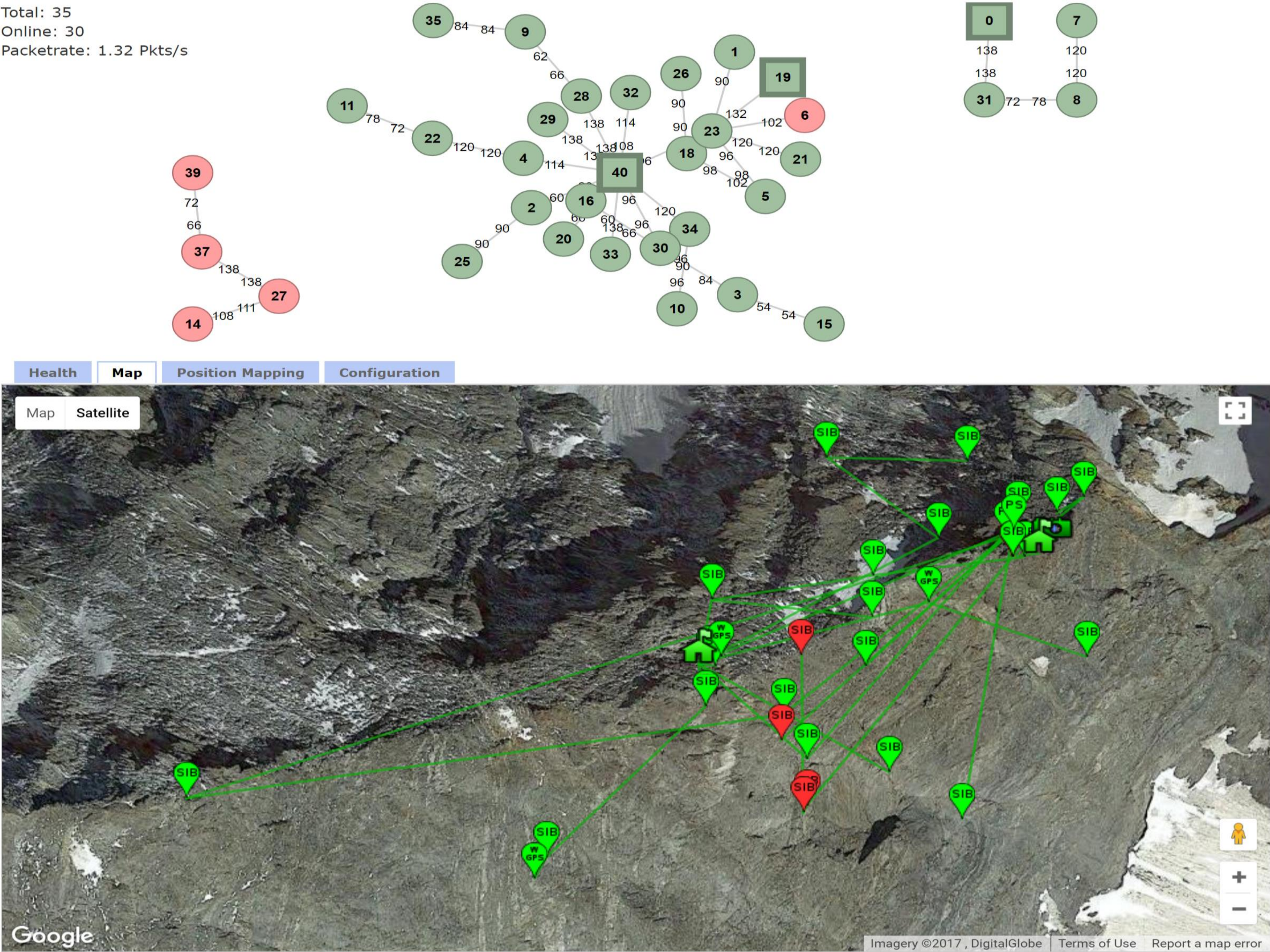
Efficiency and Reliability


100
75
50
25
reliable
efficient
Note
Note
[Google Trends]
Jan 1, 2004
Jan 1, 2013
Jan 1, 2022

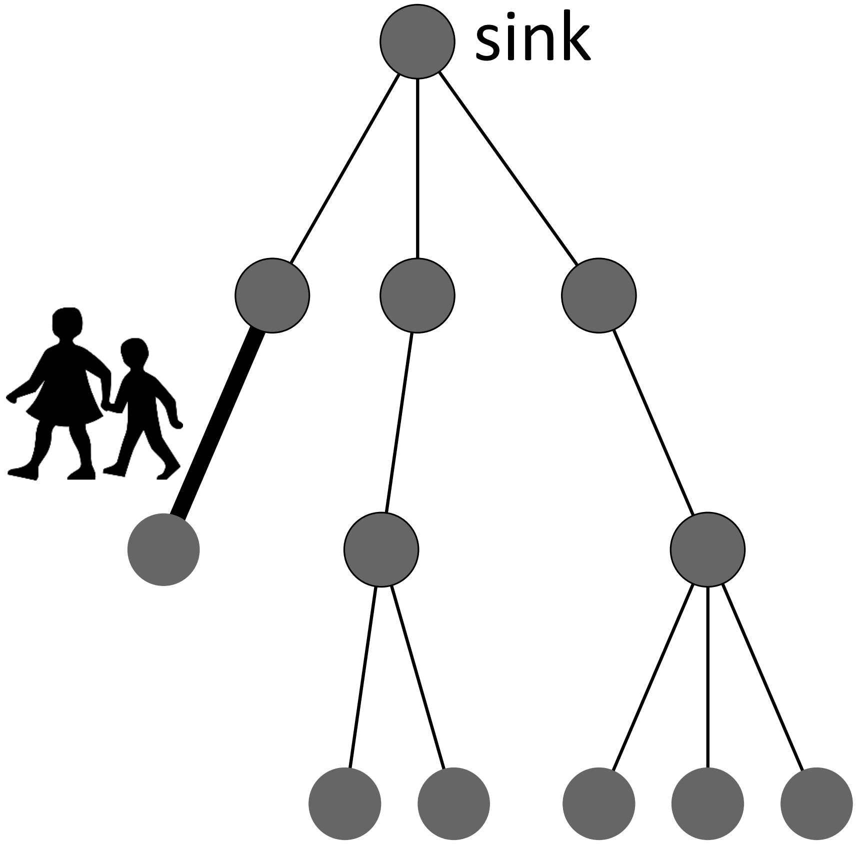
Energy Efficiency



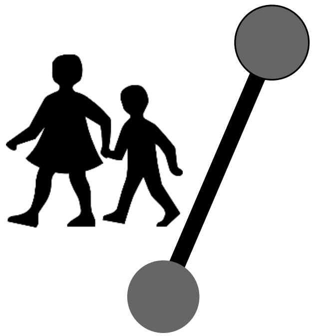
[Burri, von Rickenbach, W]


Total: 35
Online: 30
Packetrate: 1.32 Pkts/s
Health
Map
Position Mapping
Configuration
Map
Satellite
SIB
PS
W GPS
Google
Imagery ©2017 , DigitalGlobe
Terms of Use
Report a map error

Energy Efficiency

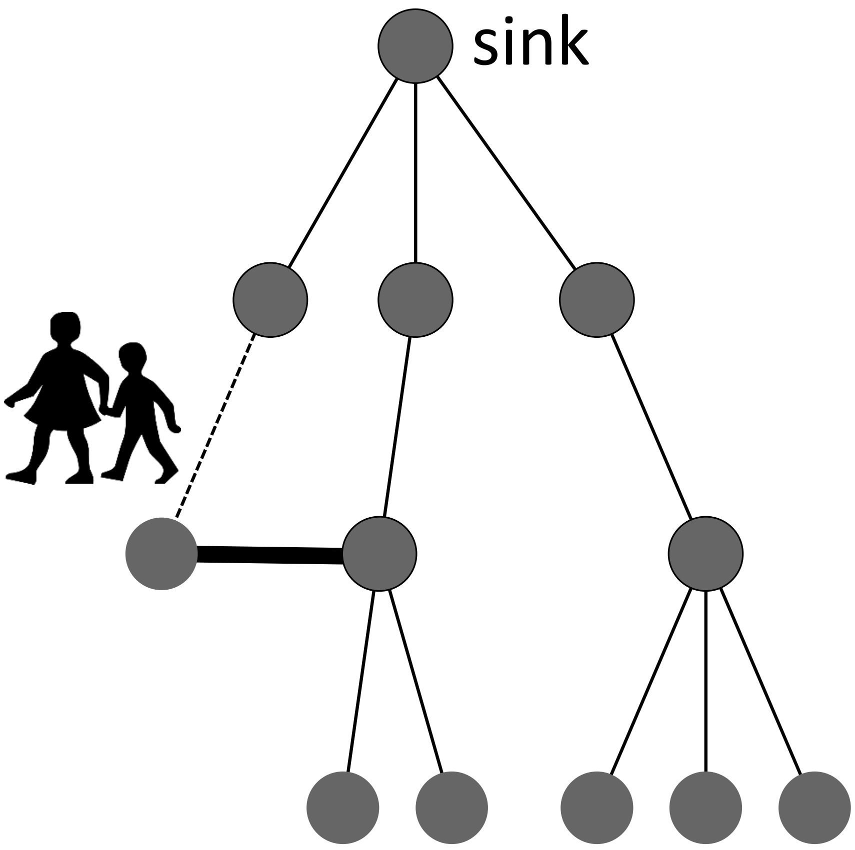


Energy Efficiency

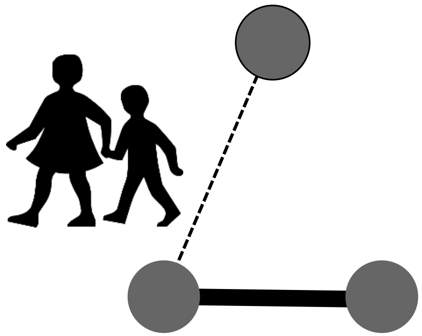


duty cycling, wake up e.g. every 10 seconds
parent synchronizes children
no network wide synchronization
mean energy consumption: 0.066mW, 10y battery

Reliability



Reliability



nodes send beacons to reconnect orphans
collisions are explicitly accepted
availability & reliability: 99% to 99.999%


RECHARGEMENT


Wireless vehicle detection systems
for outdoor parking lots
[tinynode]

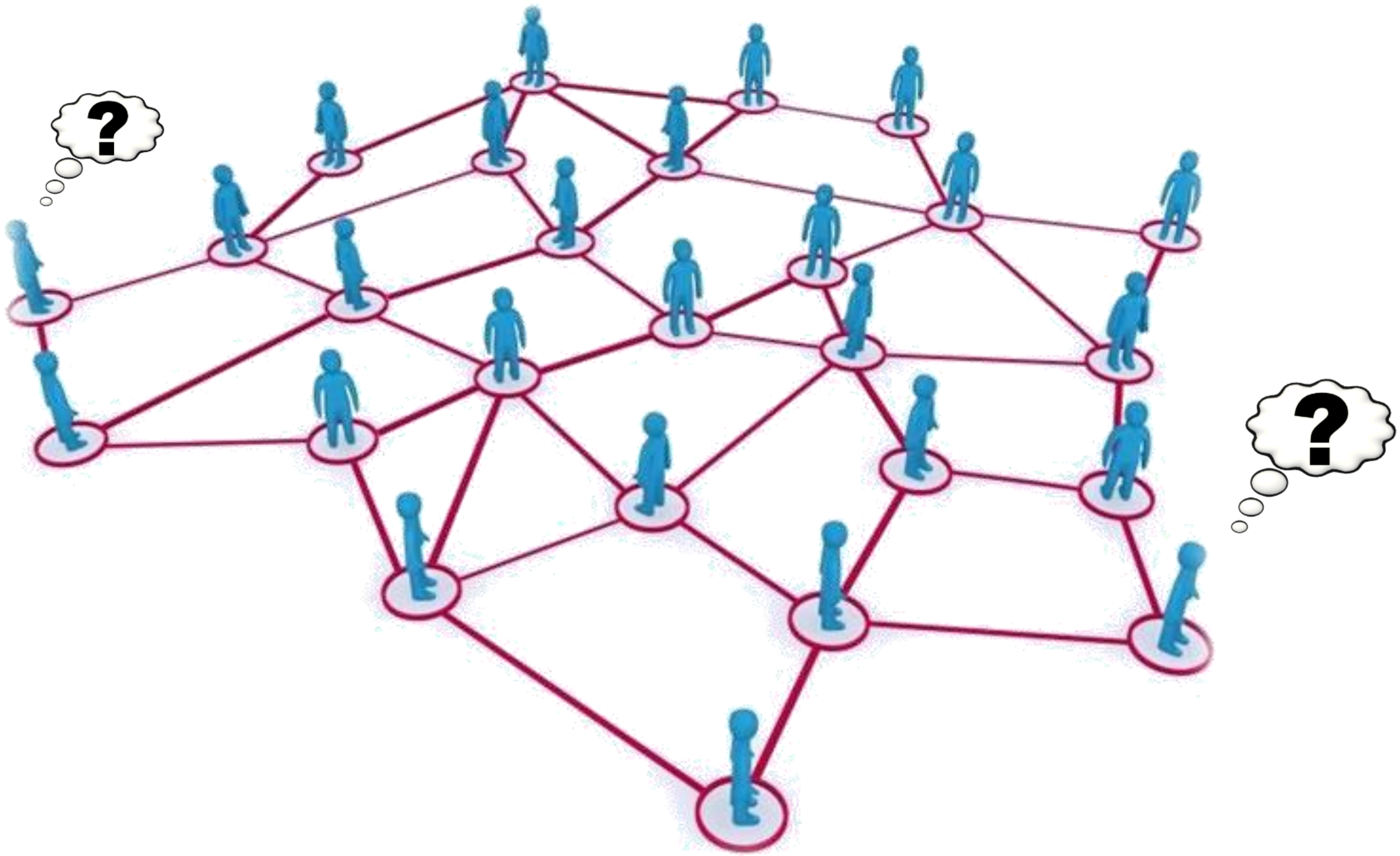
Where’s the Algorithmic Theory?

“no network wide synchronization”

Time Synchronization

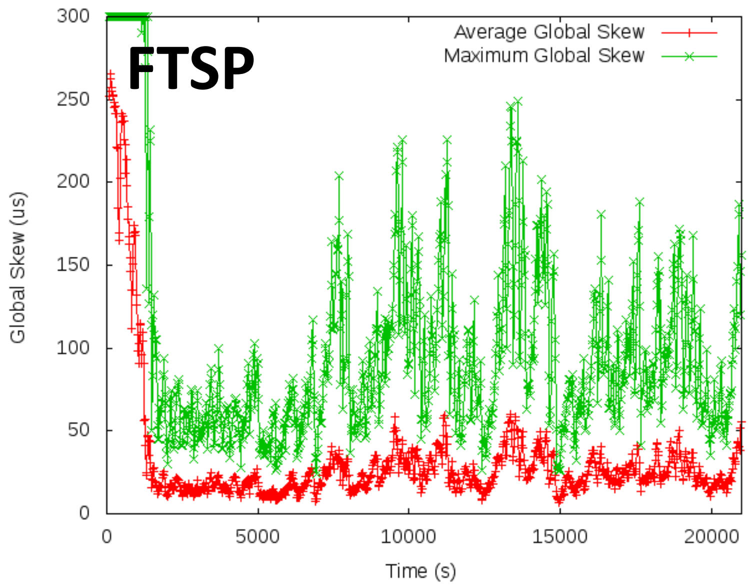
Network Synchronization is Tough

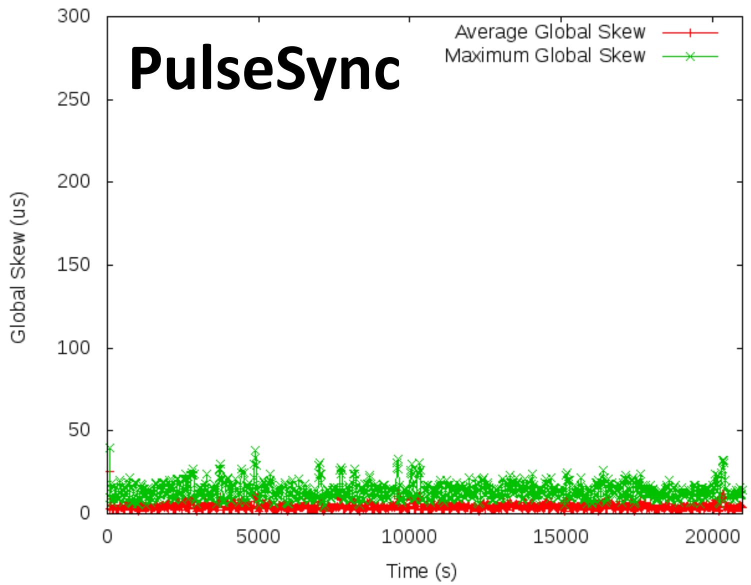
Network Synchronization



Tree Based Protocols

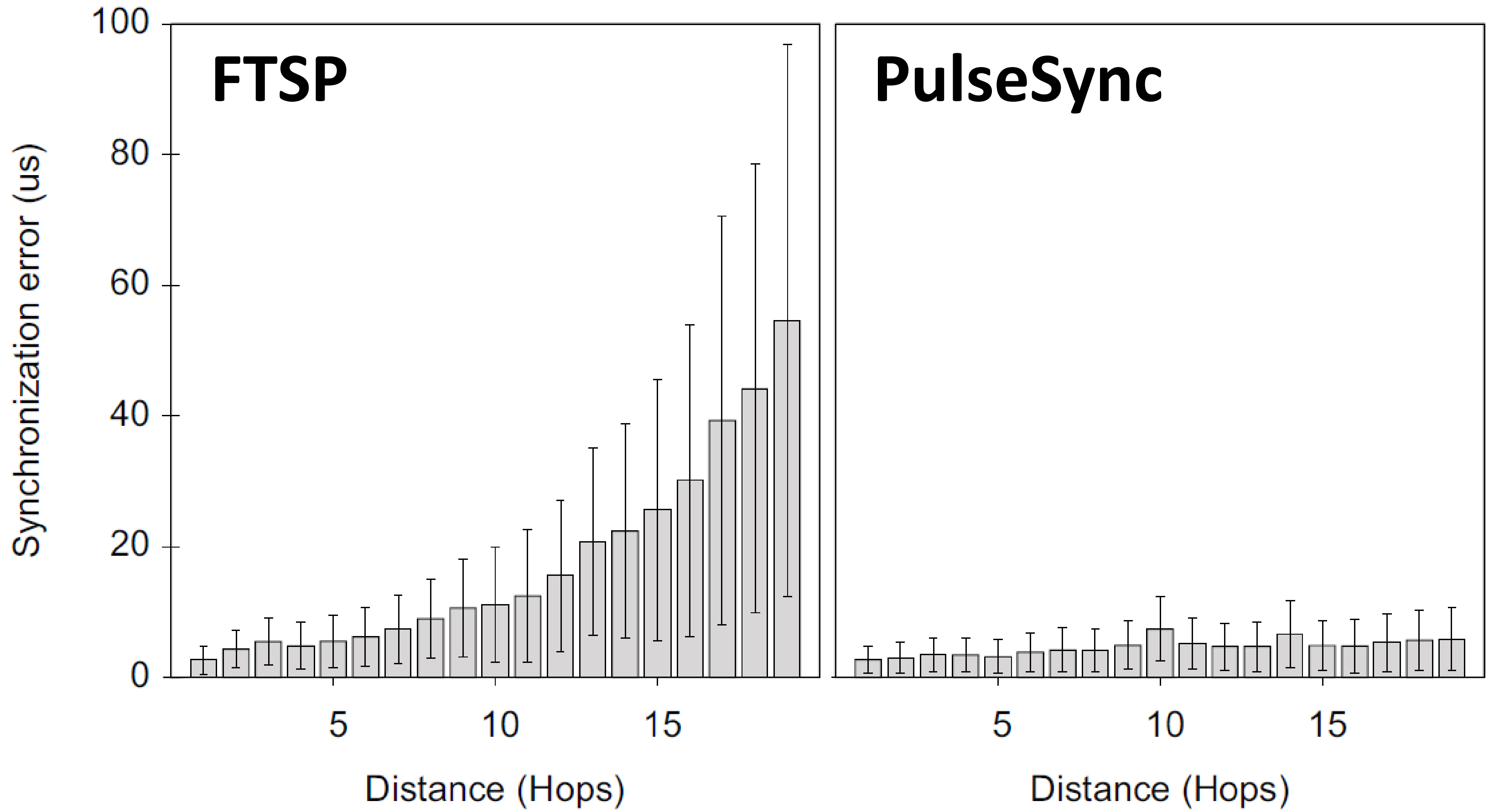
[Lenzen, Sommer, W, TON]



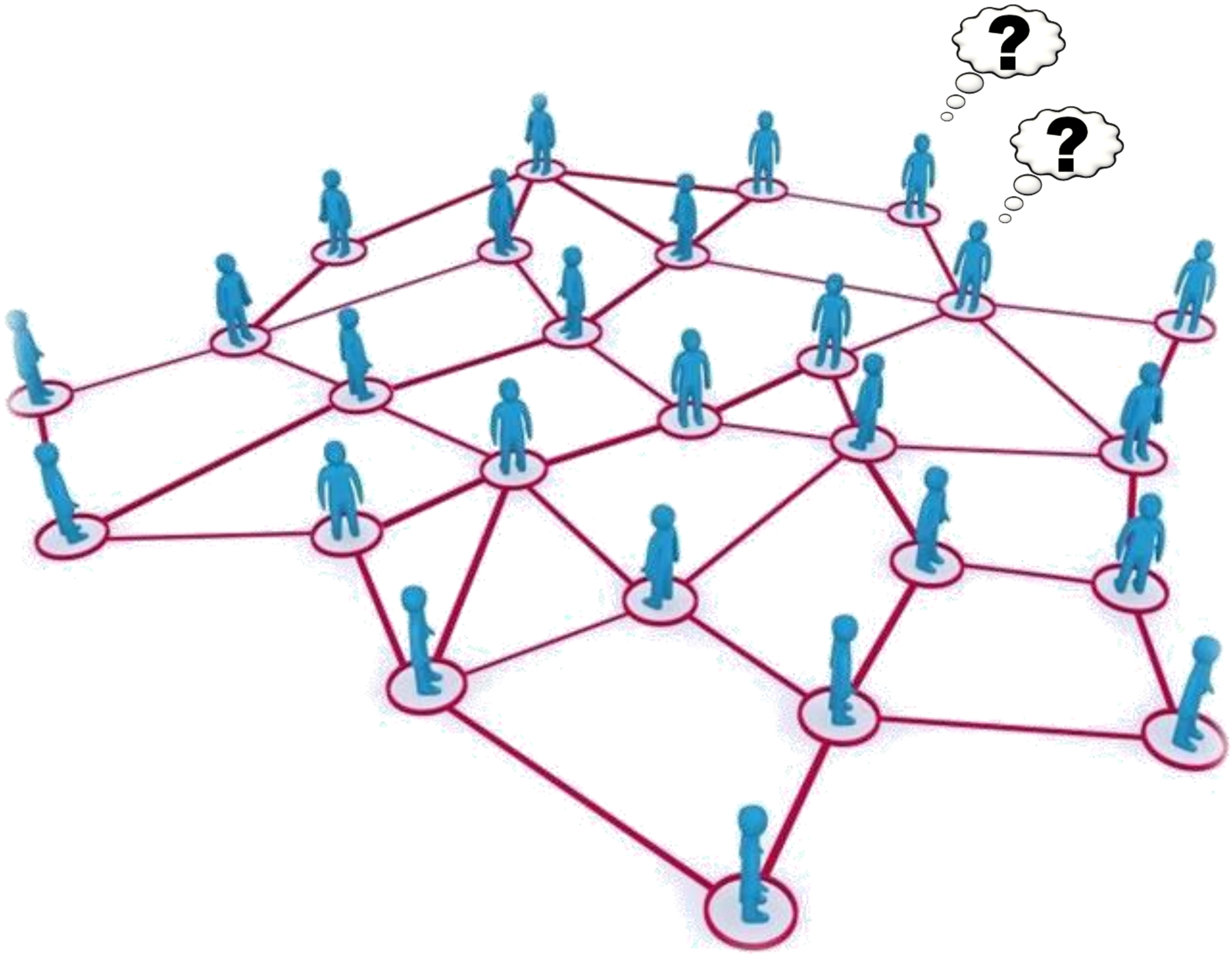


Synchronization Error	FTSP	PulseSync
Average (t > 2000s)	23.96 µs	4.44 µs
Maximum (t > 2000s)	249 µs	38 µs

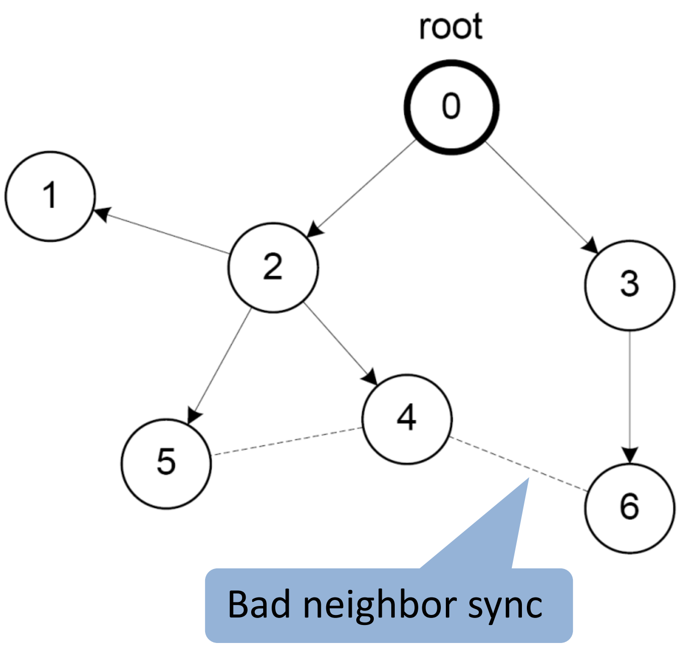
Error with Distance



Neighbor Synchronization

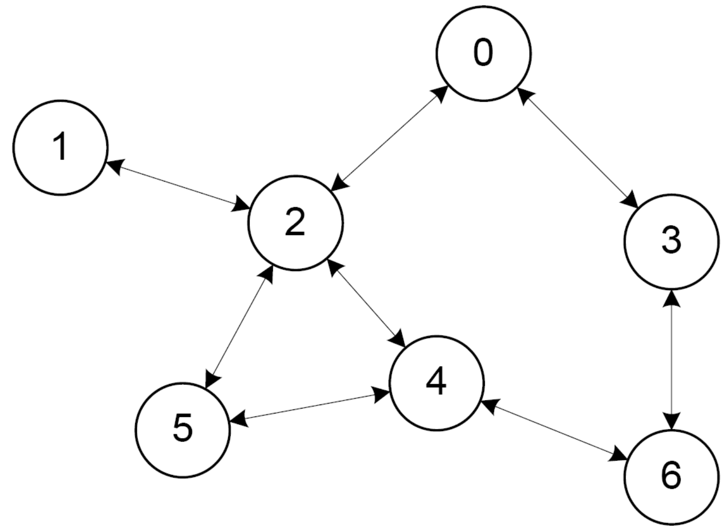


Neighbor Synchronization?



Tree-based Algorithms

e.g. FTSP

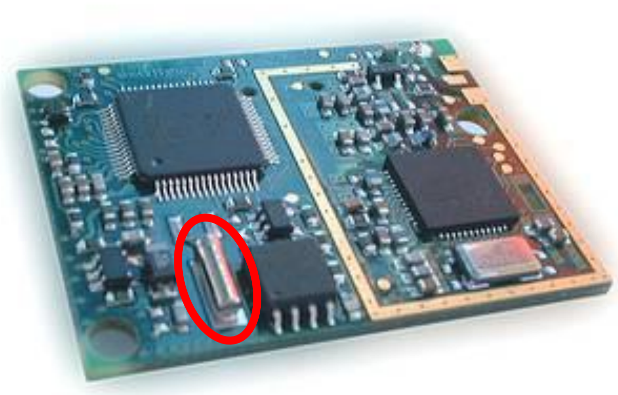


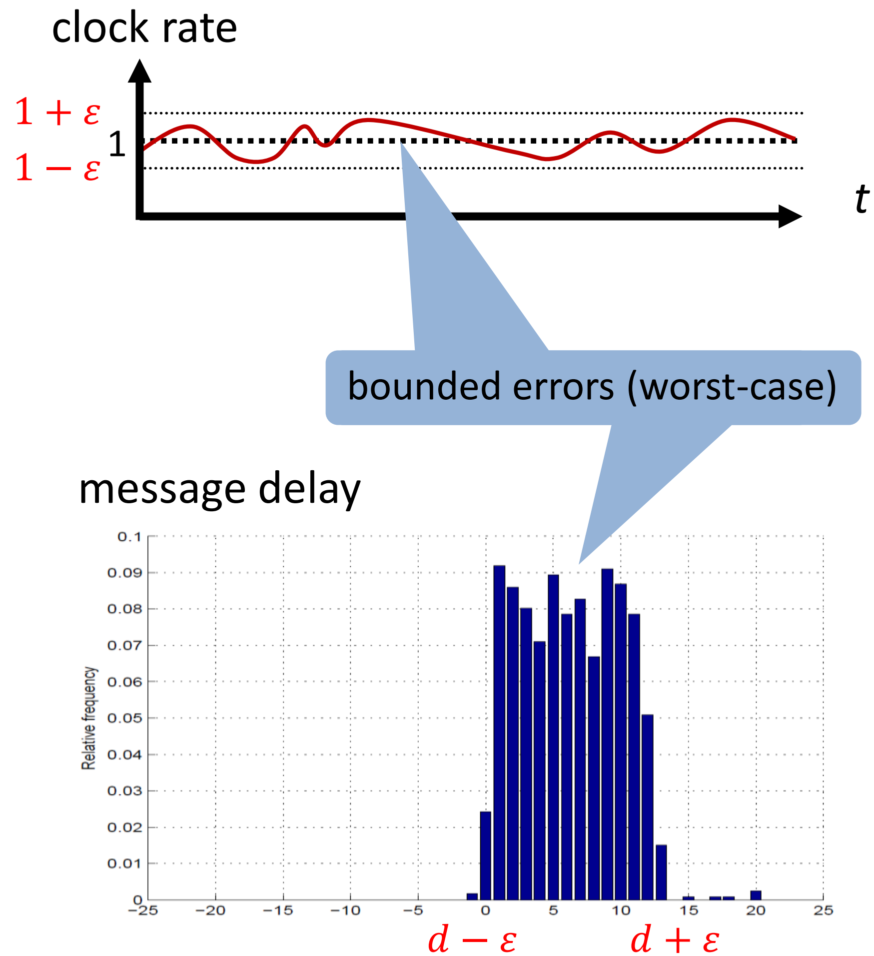
Neighborhood Algorithms

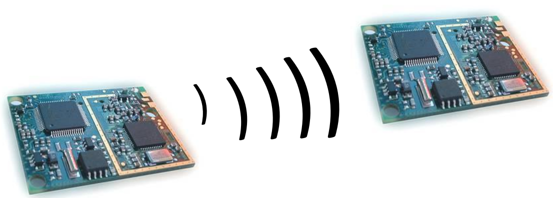
e.g. GTSP

Theorem:
Neighbor Sync is Somewhat Tough

Model: Drift & Jitter







Reasonable Time Must Behave!

no stopping

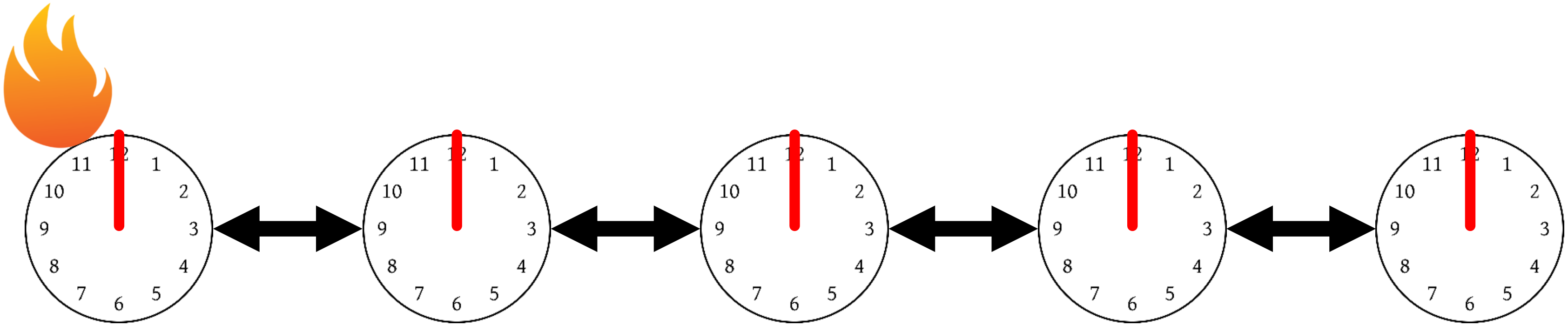
no jumping

Example: Neighbor Sync is Hard

sync to fastest neighbor

message delay = 1

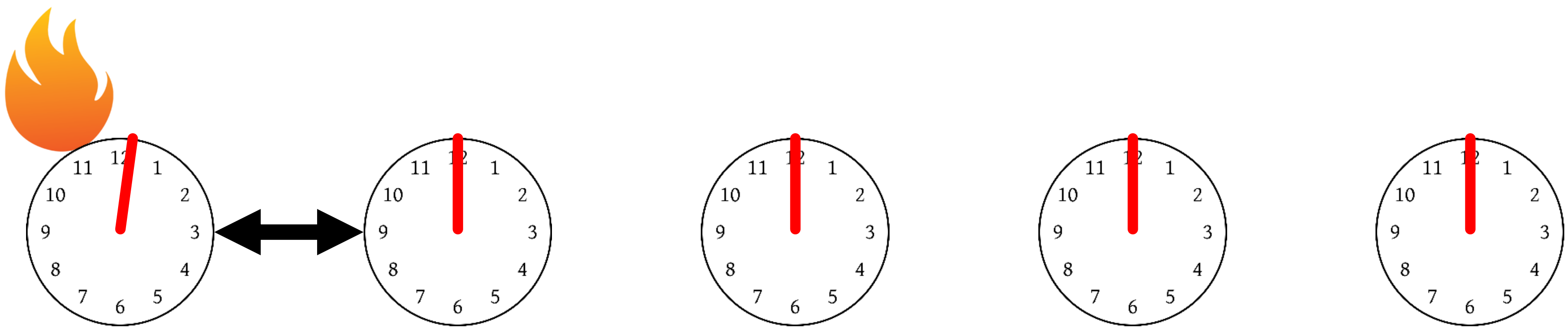
$d + \varepsilon$



Example: Neighbor Sync is Hard

sync to fastest neighbor

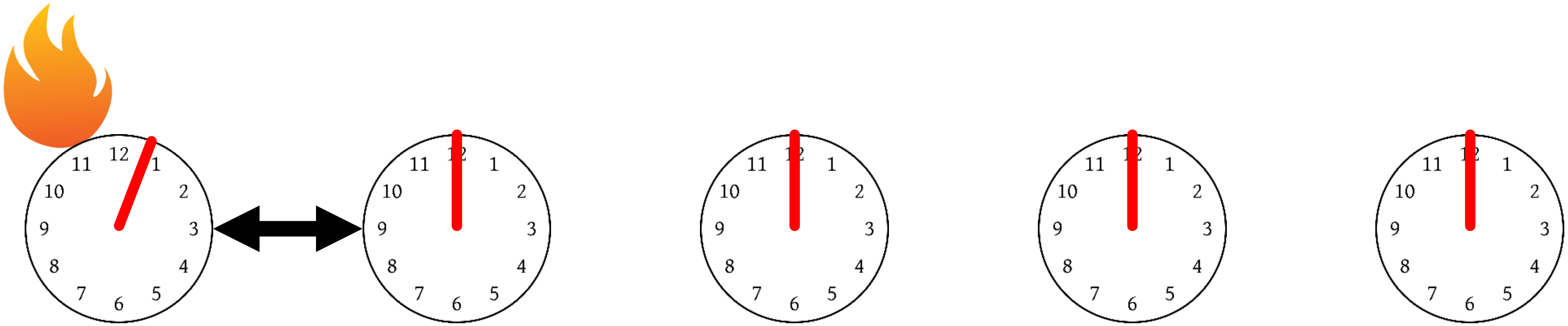
message delay = 1



Example: Neighbor Sync is Hard

sync to fastest neighbor

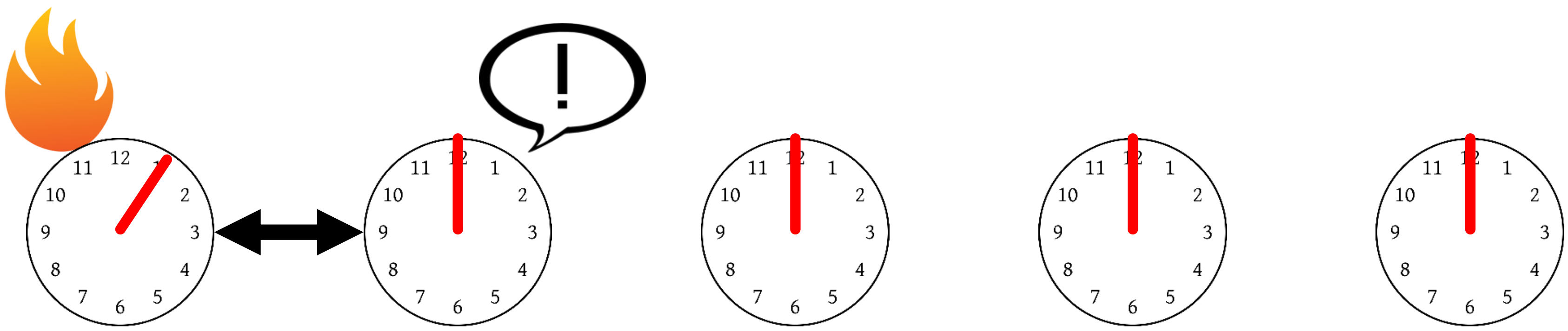
message delay = 1



Example: Neighbor Sync is Hard

sync to fastest neighbor

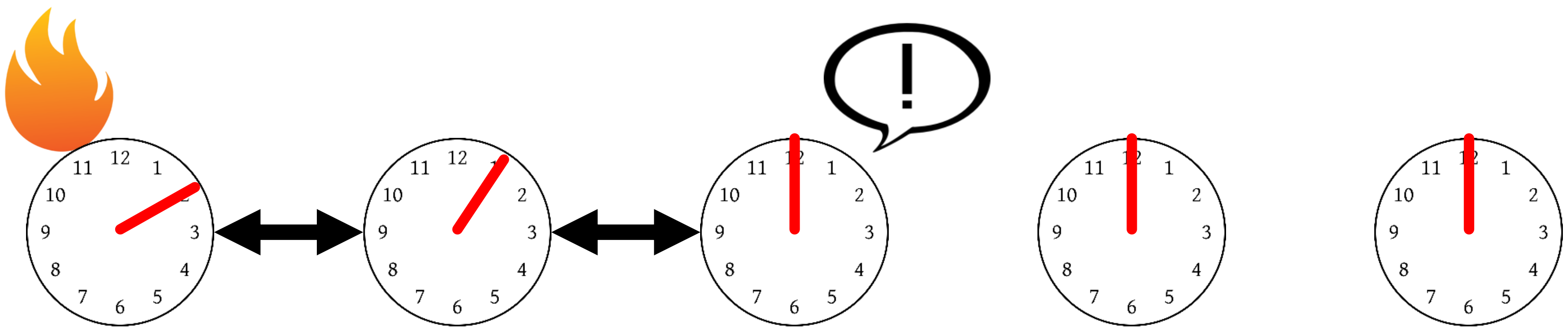
message delay = 1



Example: Neighbor Sync is Hard

sync to fastest neighbor

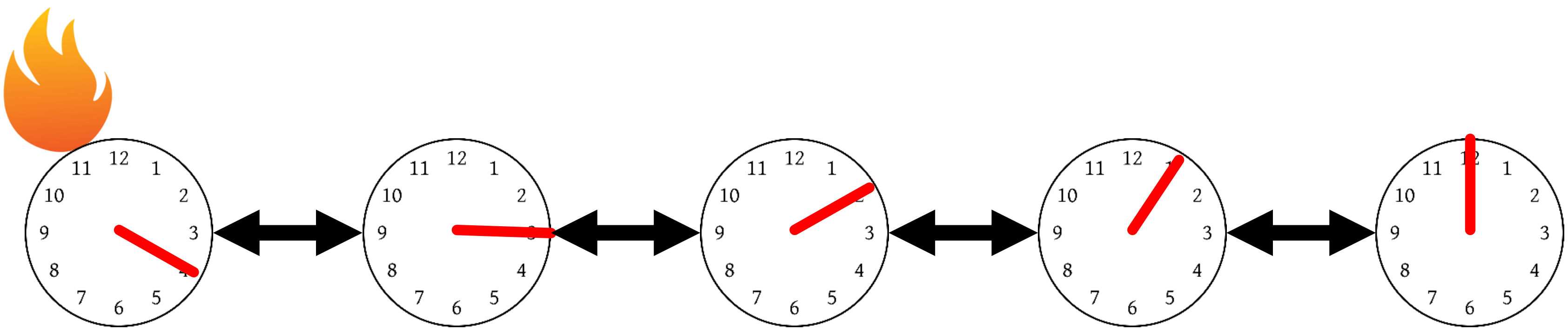
message delay = 1



Example: Neighbor Sync is Hard

sync to fastest neighbor

message delay = 1

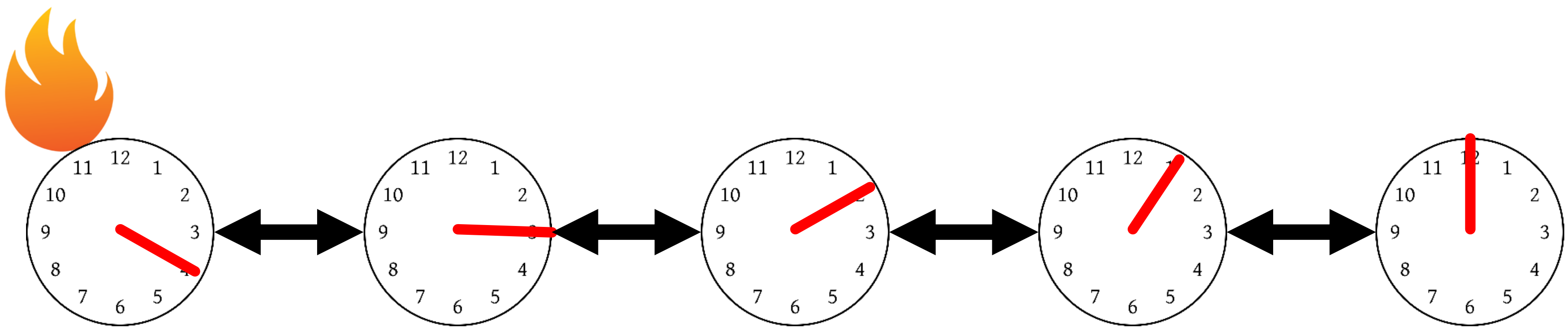


Example: Neighbor Sync is Hard

sync to fastest neighbor

message delay = $\cancel{1}$ 0

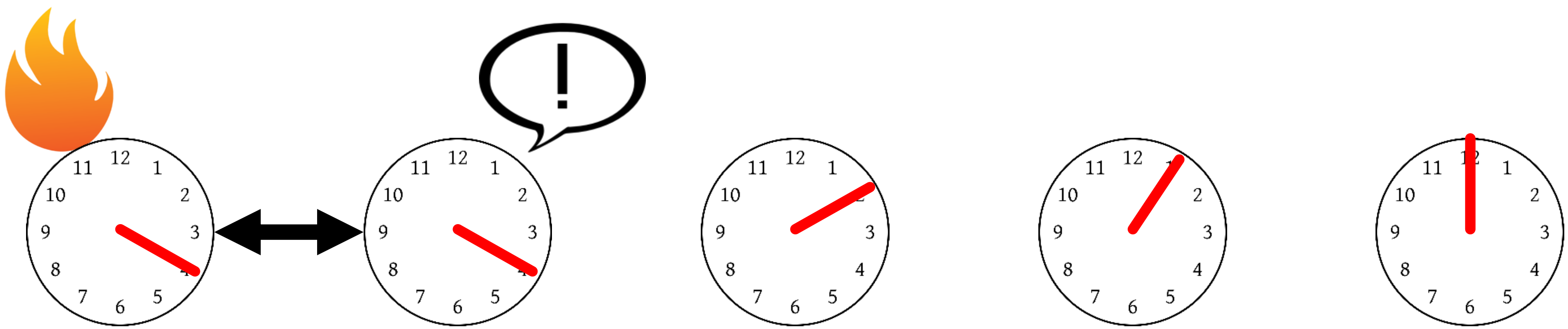
$d - \varepsilon$



Example: Neighbor Sync is Hard

sync to fastest neighbor

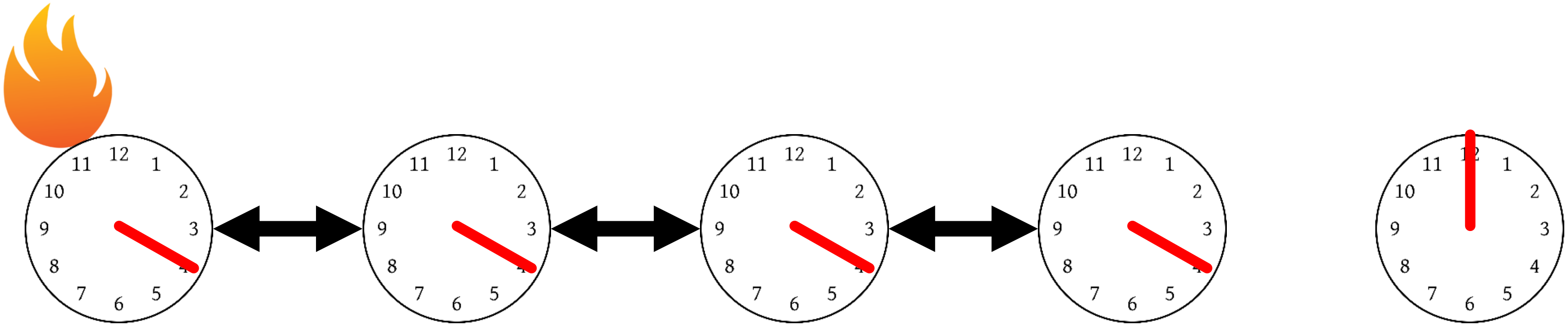
message delay = ~~1~~ 0



Example: Neighbor Sync is Hard

sync to fastest neighbor

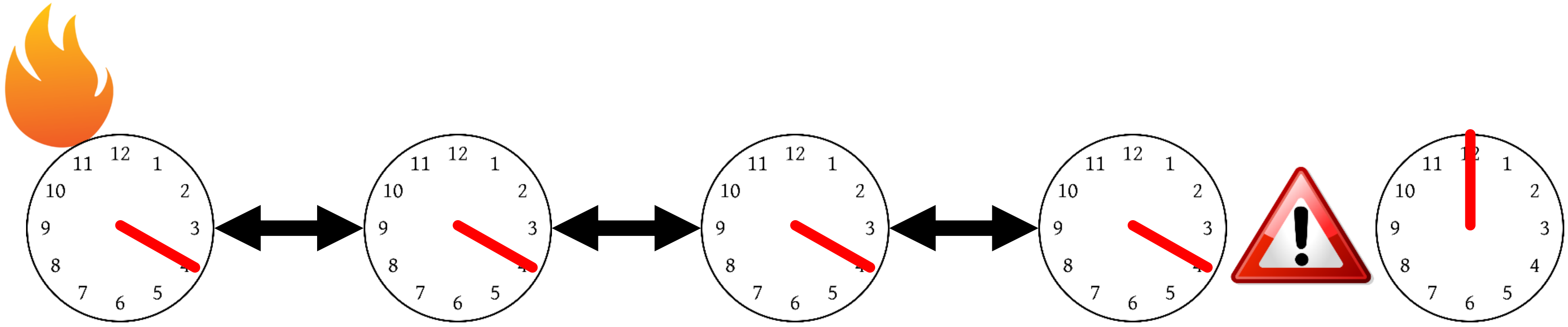
message delay = ~~1~~ 0



Example: Neighbor Sync is Hard

sync to fastest neighbor

message delay = ~~1~~ 0

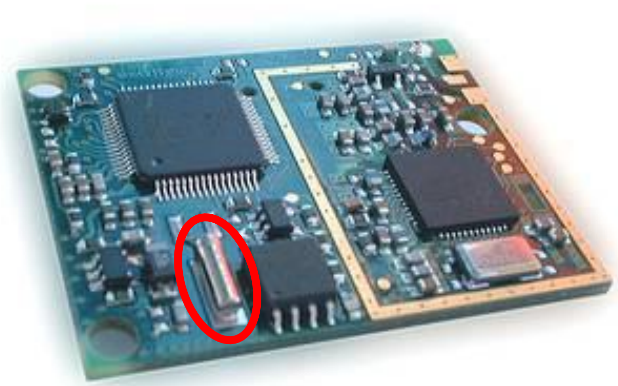


Sync To Fastest Neighbor: Local Skew Can Be Diameter

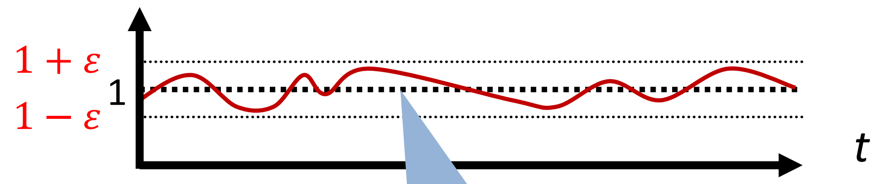
Average of Neighbors:
Local Skew Can Be Diameter Squared

Better Protocol?

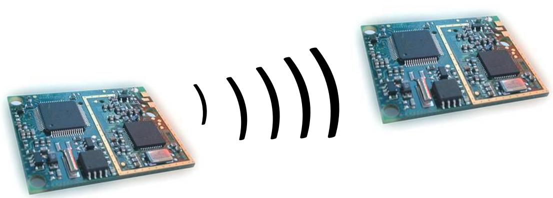
Reminder: Drift & Jitter



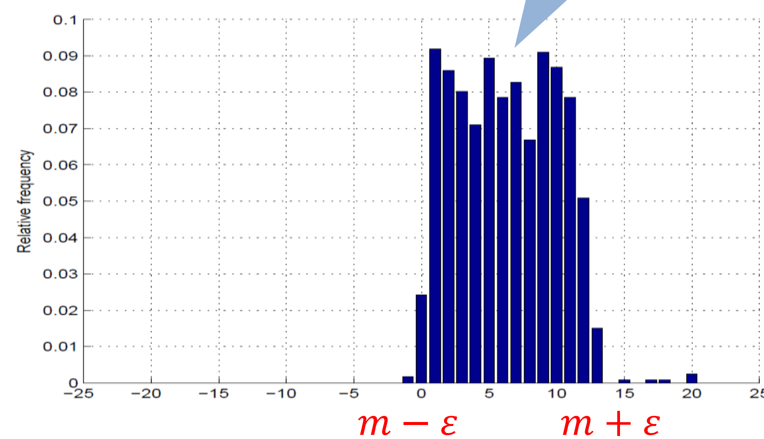
clock rate



bounded errors (worst-case)



message delay



Theorem:
Neighbor Sync is Somewhat Tough

neighbor sync error = log diameter
lower bound: difficult proof
matching upper bound: not trivial as well

[Lenzen, Locher, W, JACM]

“The Future”


56


Starlink

Global Positioning System

GPS

[ESA]

31 GPS

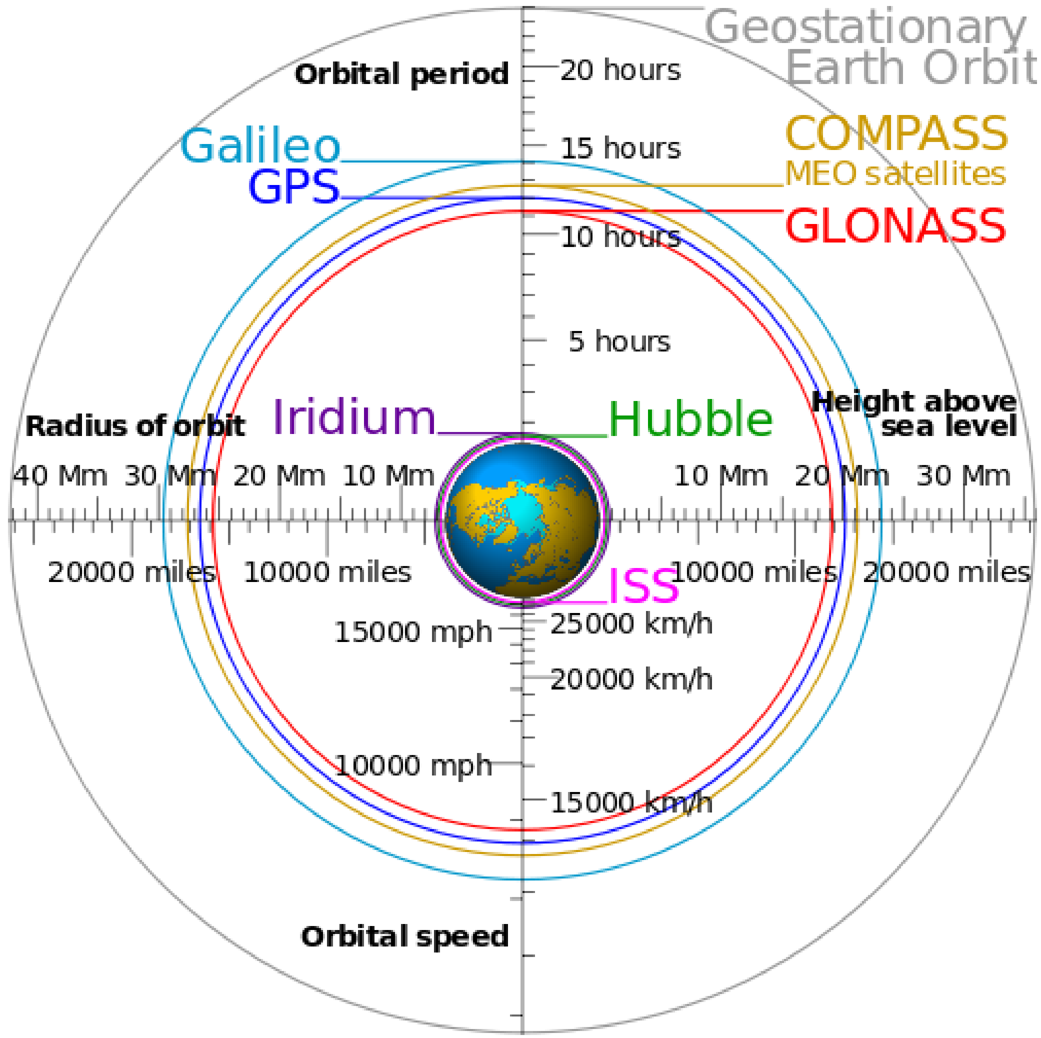
24 GLONASS

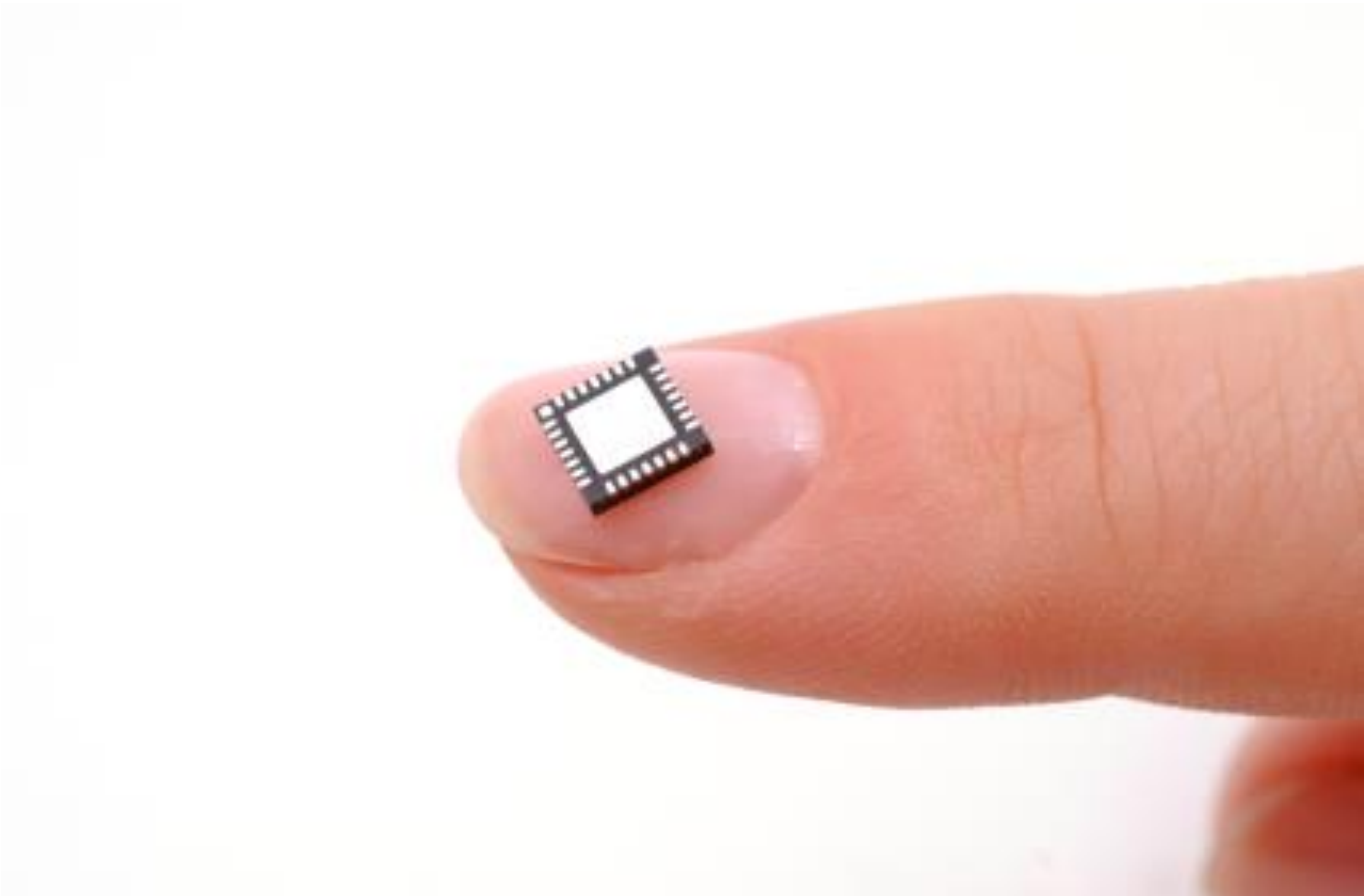
35 BeiDou (2020)

23 Galileo (2020)

7 NAVIC (local)

4 QZSS (local)



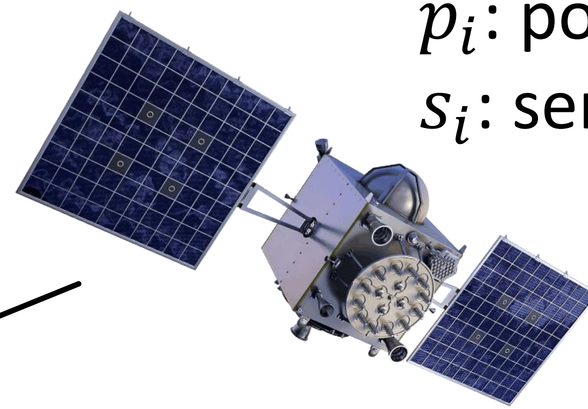



digi traq
A Product of HidnSeek
www.hidnseek.com
OLDENDORFF
LEOPOLD OLDENDORFF
Ninth Ave
16th St
New York
Exxon
23 min
BRIO
1.3 MILES
279 CALS
3900 STEPS
TAG HEUER CONNECTED
SWISS ENGINEERED
GPS
14x OPTICAL ZOOM
CANON ZOOM LENS 14x IS
5.0-70.0mm 1:3.1-5.9
PowerShot SX230 HS

Atomic Clock Inside

Transmit Time + “Position” + …

$$||p_i - h||/c = d_i - \theta$$



p_i: position

s_i: sending time

$d_i = r_i - s_i$

c: speed of light

h: 3D position

r_i: receiving time

θ: time offset

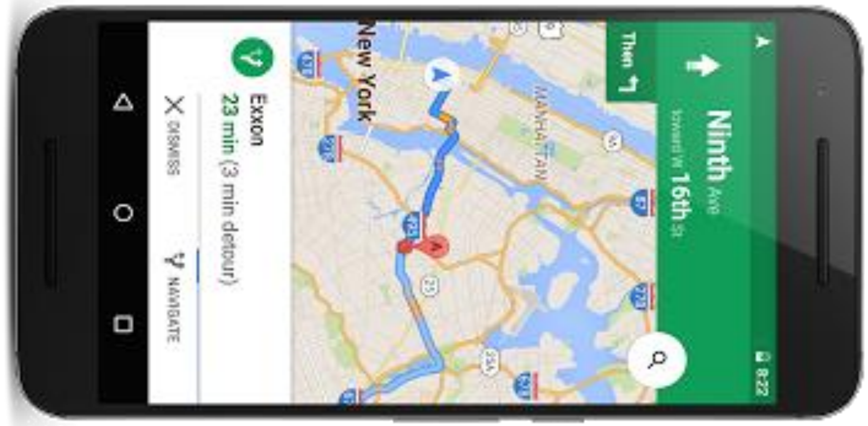
Location, Location, Location

… and Time

Tracking

Roger Wattenhofer

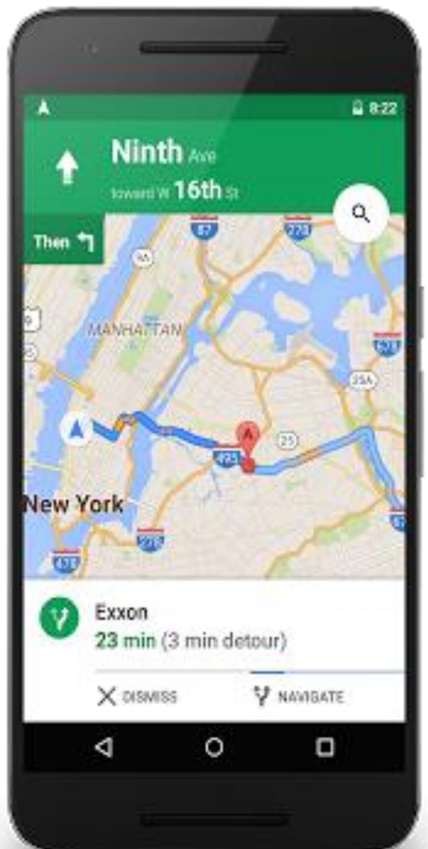

digi traq
A Product of HidnSeek
www.hidnseek.com


New York
Exxon
23 min (3 min detour)
DISMISS
NAVIGATE
MANHATTAN
Ninth Ave
toward W 16th St
Then
8:22


1.3 MILES
279 CALS
3900 STEPS
TAG HEUER
TAG HEUER CONNECTED
SWISS ENGINEERED


GPS
14x OPTICAL ZOOM
Canon
CANON ZOOM LENS 14x IS
5.0-70.0mm 1:3.1-5.9
PowerShot
SX230 HS

Energy Problem



~~1 day~~

with GPS: 6 hours

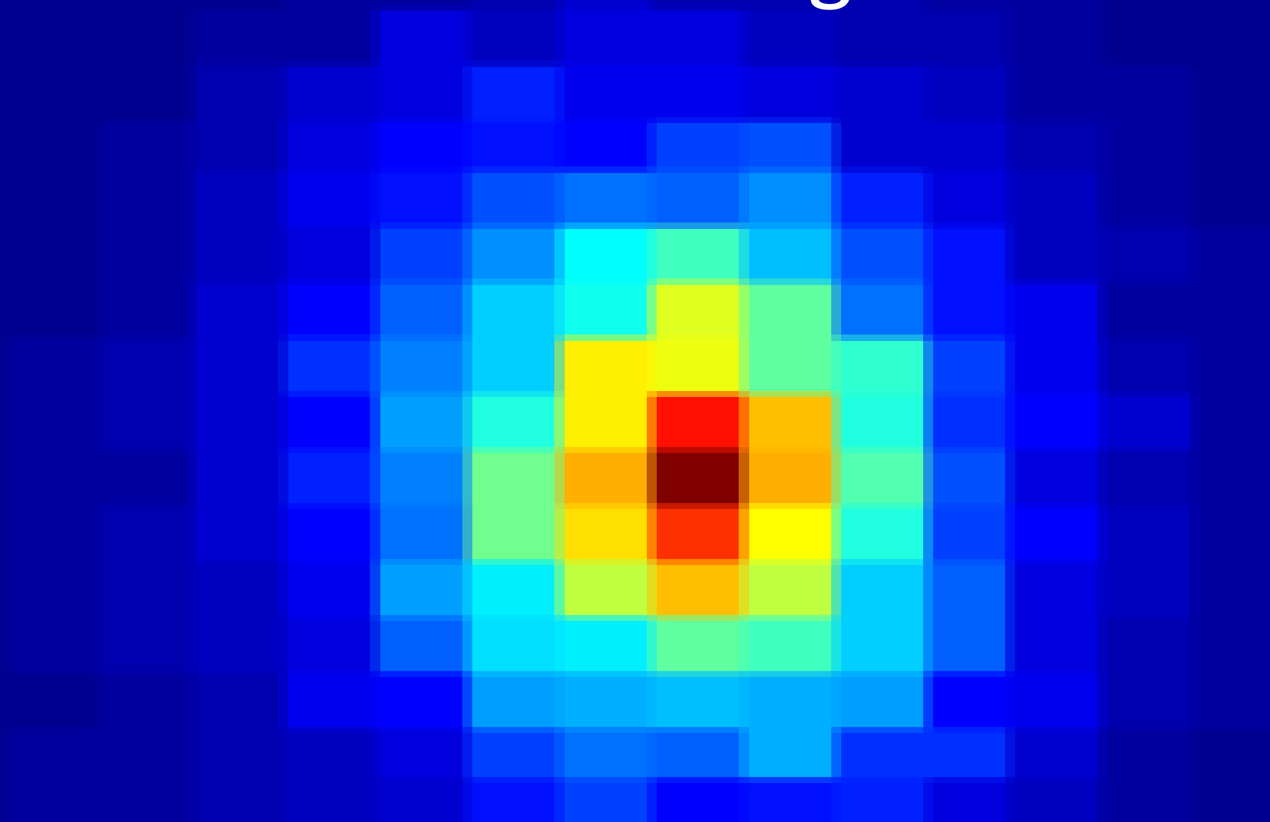
Delay Problem

Position and time only 30s after turning camera on

Delay Problem

Better: Record 1ms of raw GPS data when taking picture

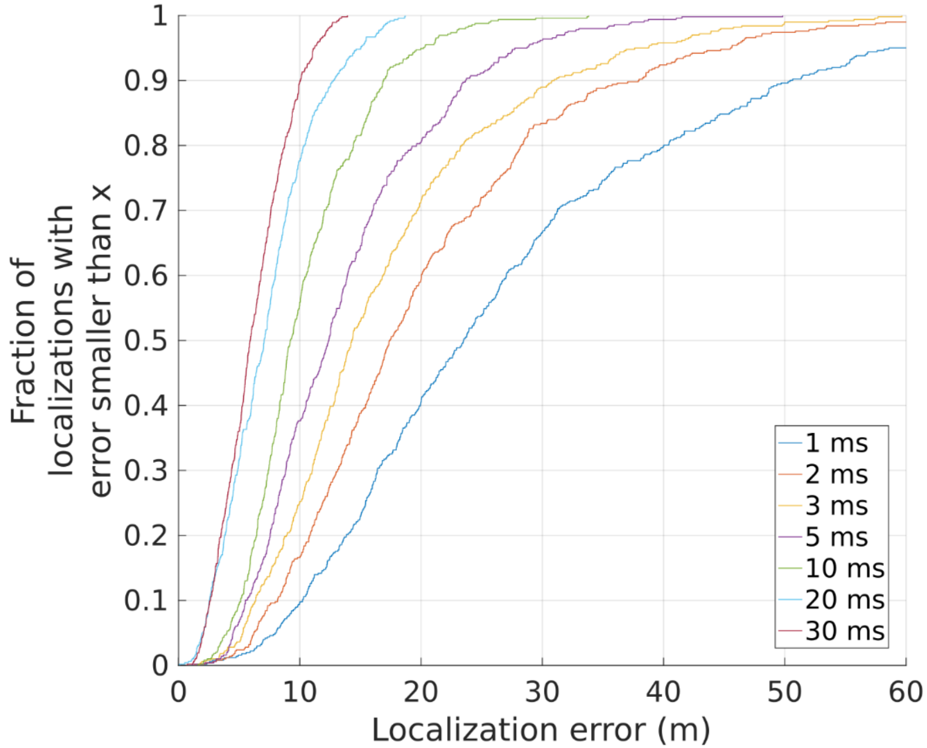
Coarse Time Navigation



Branch and Bound Search

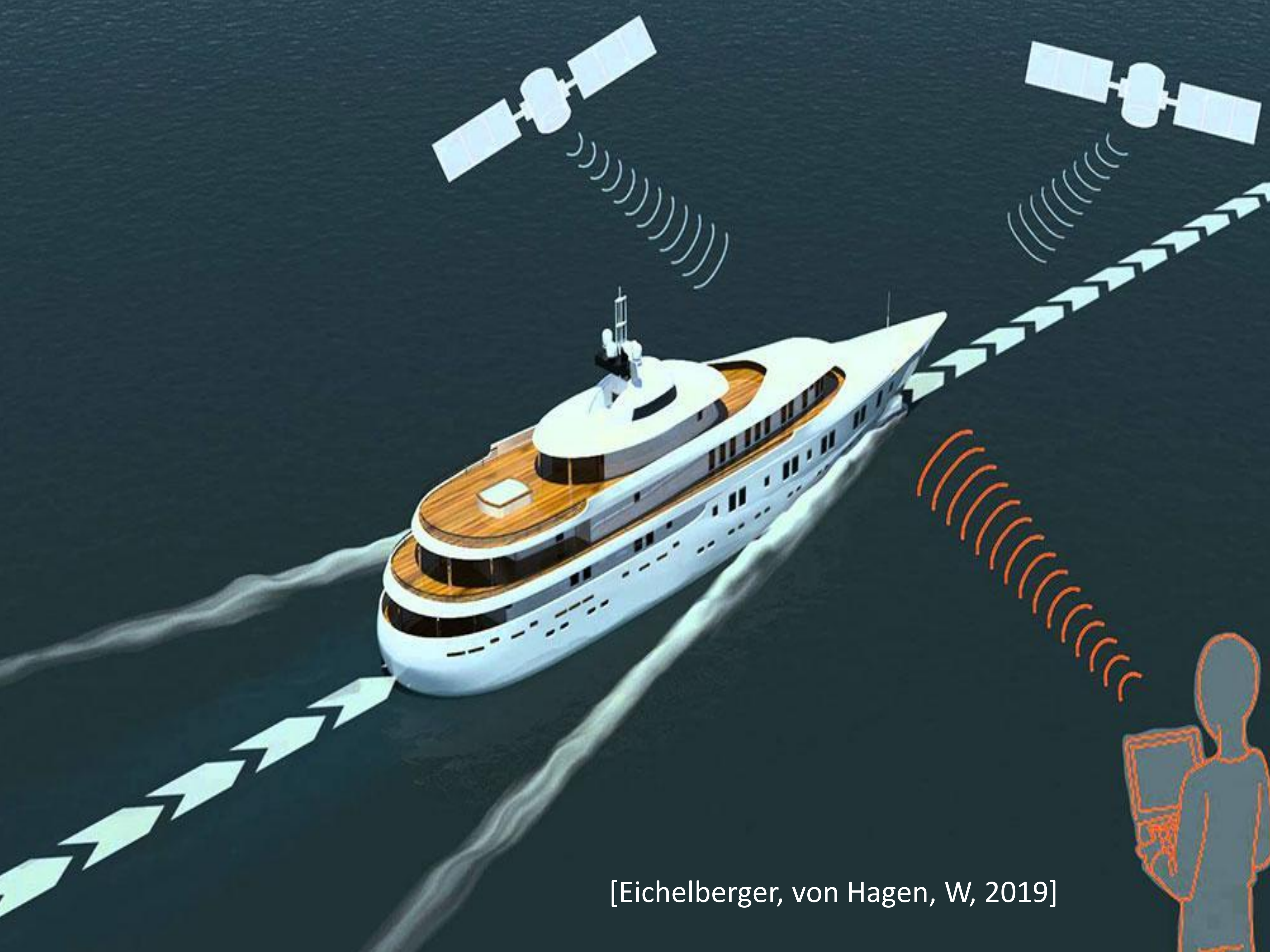
[Bissig, Eichelberger, W]

Coarse Time Results

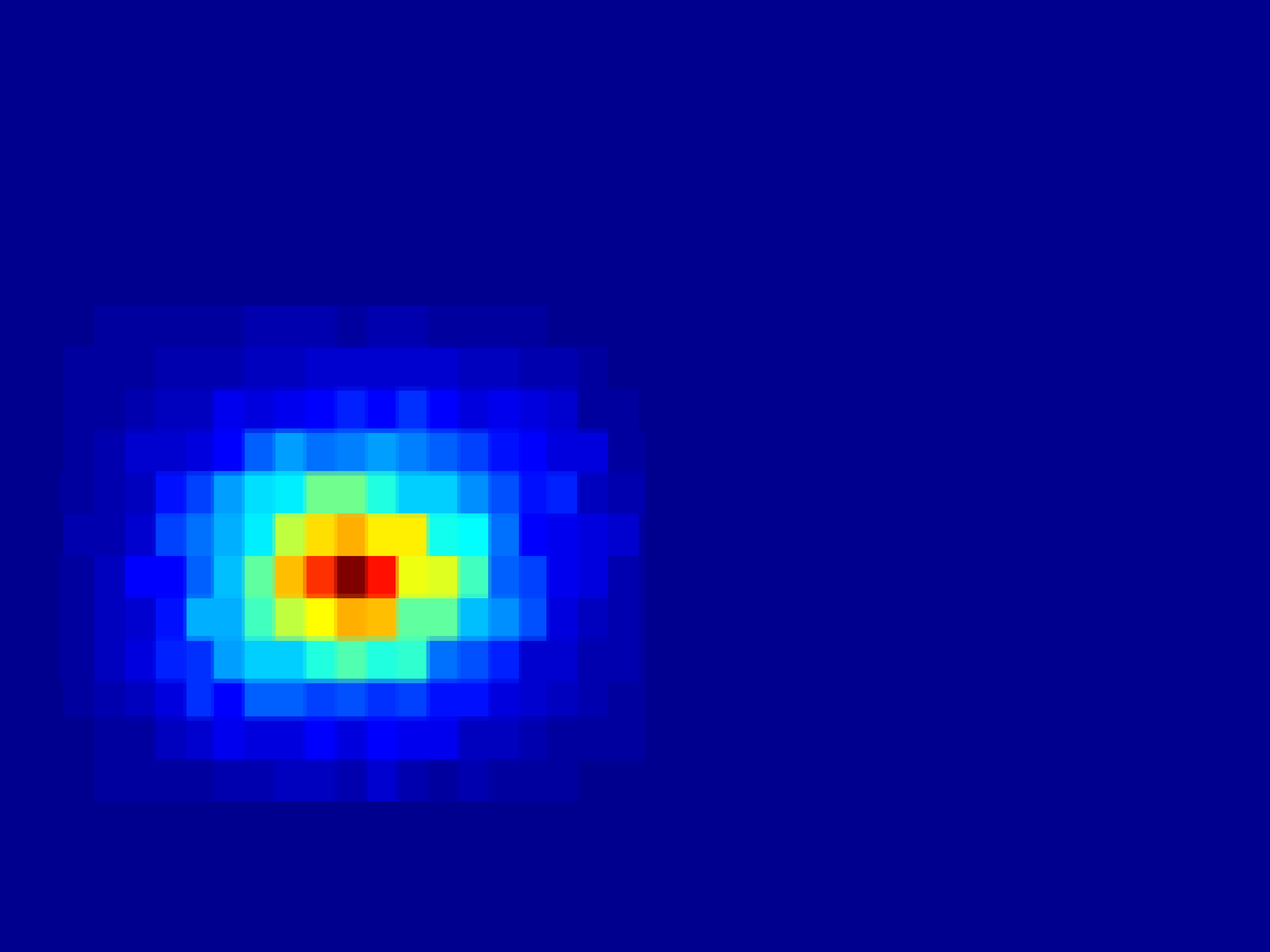


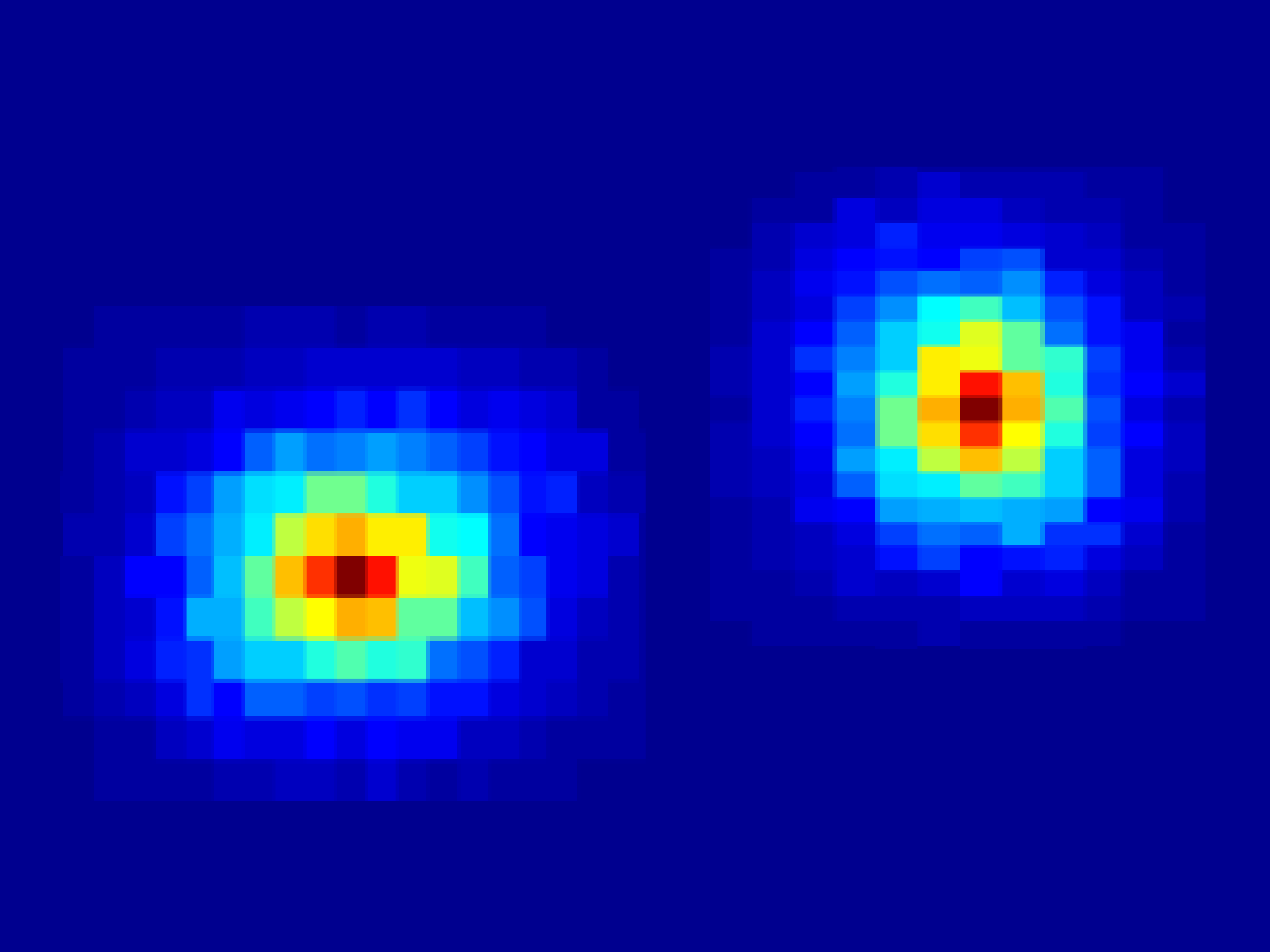

15 minute duty cycle
18 µW mean power
2 years with coin cell
[Eichelberger, von Hagen, W]

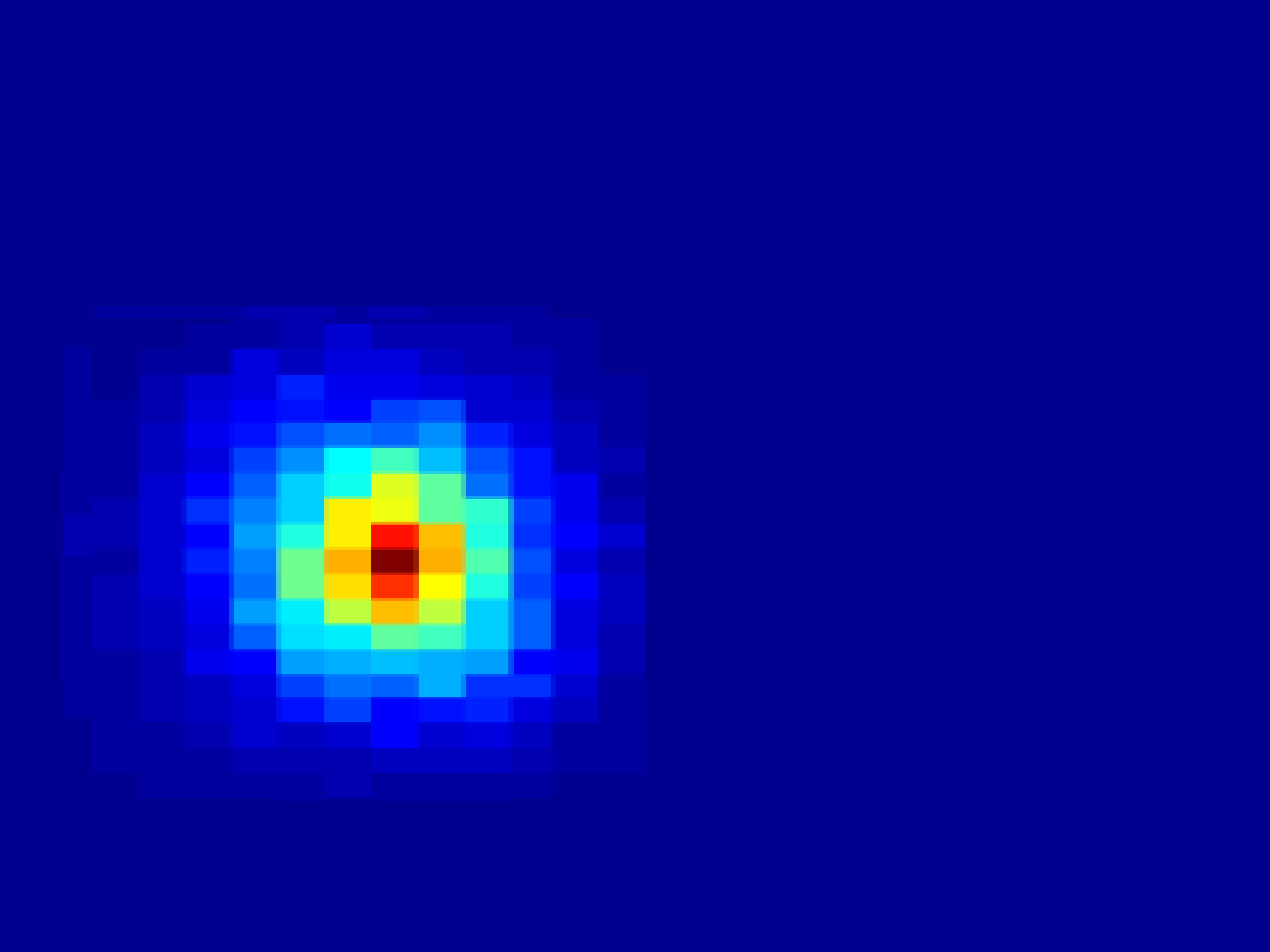
GPS Spoofing



[Eichelberger, von Hagen, W, 2019]





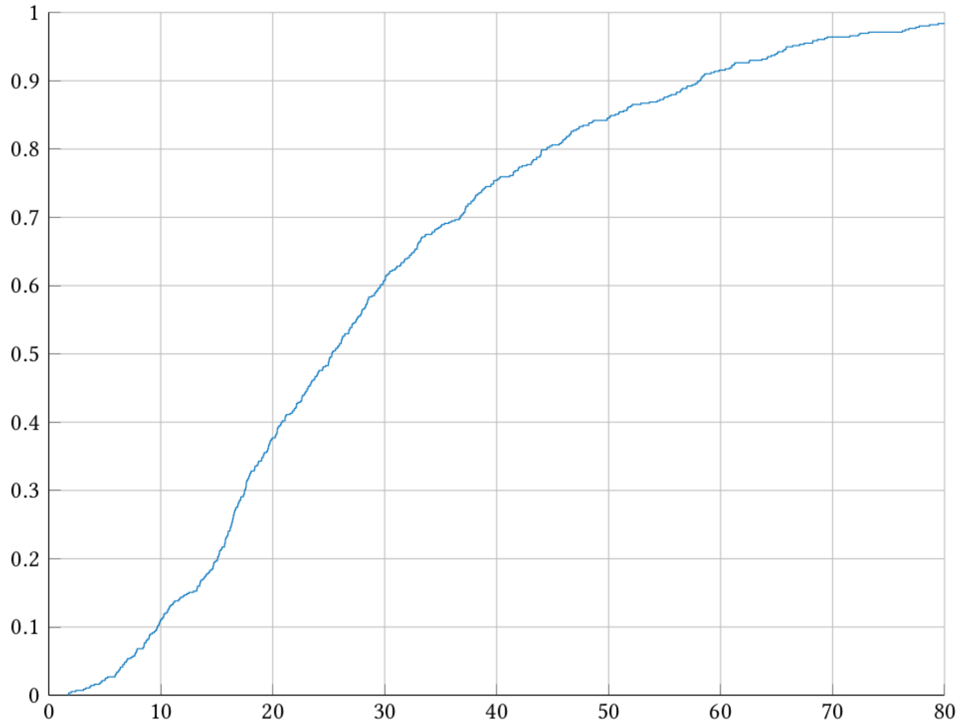


Indoor «GPS»

ADS-B

[Eichelberger, Luchsinger, Tanner, W]

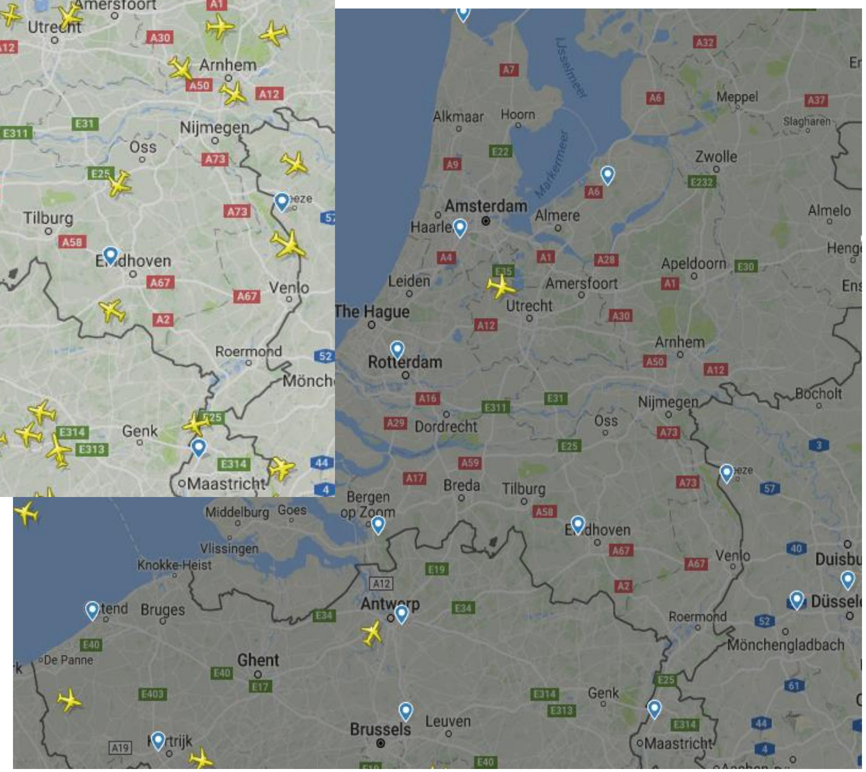
Does it work?



Localization error [m]

Still many challenges…

Alkmaar
Hoorn
Amsterdam
Almere
Zwolle
Apeldoorn
Amersfoort
Leiden
The Hague
Utrecht
Rotterdam
Arnhem
Nijmegen
Dordrecht
Oss
Breda
Tilburg
Bergen op Zoom
Middelburg
Goes
Eindhoven
Vlissingen
Knokke-Heist
Venlo
Bruges
Antwerp
Roermond
De Panne
Ghent
Genk
Brussels
Maastricht
IJsselmeer
Markermeer



Alkmaar
Hoorn
Meppel
Zwolle
Amsterdam
Almere
Almelo
Apeldoorn
Leiden
Amersfoort
The Hague
Utrecht
Rotterdam
Arnhem
Nijmegen
Bocholt
Oss
Dordrecht
Breda
Tilburg
Bergen op Zoom
Middelburg
Goes
Eindhoven
Vlissingen
Venlo
Knokke-Heist
Antwerp
Bruges
Roermond
Mönchengladbach
Ghent
De Panne
Genk
Brussels
Leuven
Maastricht
IJsselmeer
Markermeer

Summary

Questions? Comments?

Roger Wattenhofer

International Symposium on Algorithmics of Wireless Networks

Ad Hoc Networks
Autonomous Mobile Robots
Communication Protocols
Complexity and Computability
Computational Models
Data Aggregation and Fusion
Dynamic Networks, Temporal Graphs
Energy Management, Power Saving
Fault Tolerance and Dependability
Game Theoretic Aspects
Infrastructure Discovery
Internet of Things
Localization
Medium Access Control

Mobility and Dynamics
Obstacle Avoidance
Pattern Formation, Experimental Analysis
Population Protocols, Swarm Computing
Resource Efficiency
RFID Algorithms
Routing and Data Propagation
Self-stabilization, Self-* Properties
Sensor Networks
Systems and Testbeds
Time Synchronization
Topology Control
Tracking
Virtual Infrastructures