Let’s get Physical!
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
Clock speed flattening sharply

Transistor count still rising

Advent of multi-core processors!
The Future of Computing
Why Should I Care?
Computer Science → Washing Machine Science

[Roger Boyle, Maurice Herlihy]
Algorithms
simple and robust model
comparable results
complexity theory

Algorithm

Input

Output

Circuit - SAT
SAT
3-CNF SAT
Clique Problem  Subset Problem
Vertex Cover Problem
Hamiltonian Cycle
Travelling Salesman
Input → Algorithm → Output

vs.

[Image of a complex circuit board]
The Future of Computing?
Talk Overview

Introduction & Motivation

Some Examples for Physical Algorithms

What are Physical Algorithms?
Well-Known Examples
Statistical Physics

Properties of random graphs
"Static" view
Statistical Physics

Properties of random graphs
“Static” view

Physical Algorithms

People will make decisions
[Kleinberg 2000]
Natural Algorithms

[ Bernard Chazelle, 2009 ]
BitThief-Worded T-shirt by bit_thief

Ladies Basic T-Shirt

Color: Yellow (+ $2.55)
Size: Adult 2X
Model: Lindsay

Price: $23.55

Add to cart

Customize: Change the design, add your own ideas!
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

Goal: Synchronize Clocks (“Logical Clocks”)

• Both global and local synchronization!

worst-case (but constant)
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

Bad local skew
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)!

![Diagram showing clock updates and skew](image)
Local Skew: Overview of Results

- Everybody’s expectation, 10 years ago („solved“)
- Lower bound of $\log D / \log \log D$ [Fan & Lynch, PODC 2004]
- Kappa algorithm [Lenzen et al., FOCS 2008]
- Tight lower bound [Lenzen et al., PODC 2009]
- Dynamic Networks! [Kuhn et al., SPAA 2009]
- Dynamic Networks! [Kuhn et al., PODC 2010]
- All natural algorithms [Locher et al., DISC 2006]
- Blocking algorithm
- 1
- $\log D$
- $\sqrt{D}$
- $D$
- ...
Experimental Results for Global Skew

FTSP

PulseSync

[Lenzen, Sommer, W, SenSys 2009]
Experimental Results for Global Skew

FTSP

PulseSync

[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

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

- Power level of sender \( u \)
- Path-loss exponent \( \alpha \)
- Noise
- Received signal power from sender
- Received signal power from all other nodes (=interference)
- Minimum signal-to-interference ratio
- Distance between two nodes
Example: Protocol vs. Physical Model

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

Let $\alpha=3$, $\beta=3$, and $N=10\text{nW}$

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

Is spatial reuse possible?

- NO [Protocol Model]
- YES [With power control]

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

SINR of B at C:
$$\frac{31.6\mu\text{W}/(1\text{m})^3}{0.01\mu\text{W} + 1.26\text{mW}/(5\text{m})^3} \approx 3.13 \geq \beta \quad \text{👍}$$
This works in practice!

... even with very simple hardware (sensor nodes)

Time for transmitting 20’000 packets:

<table>
<thead>
<tr>
<th>Node</th>
<th>Standard MAC</th>
<th>“SINR-MAC”</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>721s</td>
<td>267s</td>
</tr>
<tr>
<td>$u_2$</td>
<td>778s</td>
<td>268s</td>
</tr>
<tr>
<td>$u_3$</td>
<td>780s</td>
<td>270s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>Messages received</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_4$</td>
<td>19999</td>
</tr>
<tr>
<td>$u_5$</td>
<td>18784</td>
</tr>
<tr>
<td>$u_6$</td>
<td>16519</td>
</tr>
</tbody>
</table>

Speed-up is almost a factor 3

The Capacity of a Network
(How many concurrent wireless transmissions can you have)
... is a well-studied problem in Wireless Communication

The Capacity of Wireless Networks
Gupta, Kumar, 2000

[Liu et al, INFOCOM’03] [Grossglauser et al, INFOCOM’01]
[Toumpis, TWC’03] [Gamal et al, INFOCOM’04] [Kodialam et al, MOBICOM’05]
[Kodialam et al, MOBICOM’05] [Kyasanur et al, MOBICOM’05] [Gastpar et al, INFOCOM’02]
[Li et al, MOBICOM’01] [Mitra et al, IPSN’04] [Zhang et al, INFOCOM’05]
[Bansal et al, INFOCOM’03] [Dousse et al, INFOCOM’04]
[Yi et al, MOBIHOC’03] [Perevalov et al, INFOCOM’03] etc…
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...?
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?
"Convergecast Capacity" in Wireless Networks

<table>
<thead>
<tr>
<th>Networks</th>
<th>Max. rate in arbitrary, worst-case deployment</th>
<th>Max. rate in random, uniform deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/Power</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol Model</td>
<td>( \Theta(1/n) )</td>
<td>( \Theta(1/\log n) )</td>
</tr>
<tr>
<td>Physical Model</td>
<td>( \Omega(1/\log^3 n) )</td>
<td>( \Omega(1/\log n) )</td>
</tr>
</tbody>
</table>

The Price of Worst-Case Node Placement
- Exponential in protocol model
- Polylogarithmic in physical model (almost no worst-case penalty!)
Wireless Communication

EE, Physics
Maxwell Equations
Simulation, Testing
‘Scaling Laws’

Network Algorithms

CS, Applied Math
[Geometric] Graphs
Worst-Case Analysis
Any-Case Analysis
Possible Application – Hotspots in WLAN
Possible Application – Hotspots in WLAN
Physical Algorithms?
Physical Algorithms

- no seq. input/output
- beyond laws of physics
Agents

Network Theory

Self-Organization

BAR Games

Crypto

Game Theory

Selfish

Crashing

Altruistic/Reliable

Networks

Mobile Networks

Parallelism

Robots

Peer-to-Peer Systems

Byzantine

Network

No Mobility

Low Mobility

Physical Mobility

Virtual Mobility
Some Unifying Theory?
Example: Maximal Independent Set (MIS)

- Given a mobile network, nodes with unique IDs.
- Maintain a Maximal Independent Set (MIS)
  - a non-extendable set of pair-wise non-adjacent nodes

A simple algorithm:

IF no higher ID neighbor is in MIS → join MIS
IF higher ID neighbor is in MIS → do not join MIS

Can be implemented by constantly sending (ID, in MIS or not in MIS)
Algorithm is simple, and it will eventually stabilize!
Example

**IF no higher ID neighbor is in MIS → join MIS**
**IF higher ID neighbor is in MIS → do not join MIS**
Example

IF no higher ID neighbor is in MIS $\rightarrow$ join MIS

IF higher ID neighbor is in MIS $\rightarrow$ do not join MIS

69 17 11 10 7 4 3 1
Example

IF no higher ID neighbor is in MIS $\Rightarrow$ join MIS
IF higher ID neighbor is in MIS $\Rightarrow$ do not join MIS
Example

IF no higher ID neighbor is in MIS $\rightarrow$ join MIS
IF higher ID neighbor is in MIS $\rightarrow$ do not join MIS

• What if we have minor changes?
Example

IF no higher ID neighbor is in MIS $\Rightarrow$ join MIS
IF higher ID neighbor is in MIS $\Rightarrow$ do not join MIS

What if we have minor changes?
Example

IF no higher ID neighbor is in MIS $\Rightarrow$ join MIS
IF higher ID neighbor is in MIS $\Rightarrow$ do not join MIS

- What if we have minor changes?
Example

IF no higher ID neighbor is in MIS → join MIS
IF higher ID neighbor is in MIS → do not join MIS

> 69 17 11 10 7 4 3 1

• What if we have minor changes?

> 69 17 11 10 7 4 3 1
Example

IF no higher ID neighbor is in MIS $\Rightarrow$ join MIS
IF higher ID neighbor is in MIS $\Rightarrow$ do not join MIS

- What if we have minor changes?

```
69  17  11  10  7  4  3  1
```
Example

IF no higher ID neighbor is in MIS → join MIS
IF higher ID neighbor is in MIS → do not join MIS

• What if we have minor changes?

• Proof by animation: Stabilization time is linear in the diameter of the network
  – We need an algorithm that does not have linear causality chain („butterfly effect“)
Local Algorithms

• Given a graph, each node must determine its decision as a function of the information available within radius $t$ of the node.

• Or: Each node can exchange a message with all neighbors, for $t$ communication rounds, and must then decide.

• Or: Change can only affect nodes in distance $t$.

• Or: ...
Locality is Way to Understand Physical Algorithms

- Self-Assembling Robots
- Applications e.g. Multicore
- Local Algorithms
- Sublinear Estimators
- Dynamics
- Self-Stabilization
Results: MIS

Upper Bounds

1
log* n
log n
n

General Graphs, Randomized
[different groups, 1986]

[Schneider et al., 2008]

Growth-Bounded Graphs
[Linial, 1992]

Join MIS with prob
1/degree, repeat

Lower Bounds

Growth-Bounded Graphs
[Linial, 1992]

General Graphs
[Kuhn et al., 2004, 2006, 2010]

...similarly connected dominating sets, coloring, matching, covering, packing, max-min LPs, etc.
Lower Bound Example: Minimum Dominating Set (MDS)

- **Input:** Given a graph (network), nodes with *unique IDs*.
- **Output:** Find a Minimum Dominating Set (MDS)
  - Set of nodes, each node is either in the set itself, or has neighbor in set

**Differences between MIS and MDS**
- Central (non-local) algorithms: MIS is trivial, whereas MDS is *NP-hard*
- Instead: Find an MDS that is “close” to minimum (*approximation*)
  - *Trade-off* between time complexity and approximation ratio
Lower Bound for MDS: Intuition

- Two graphs ($m << n$). Optimal dominating sets are marked red.

| $|DS_{OPT}| = 2.$ | $|DS_{OPT}| = m+1.$ |
Lower Bound for MDS: Intuition (2)

- In local algorithms, nodes must decide only using local knowledge.
- In the example **green** nodes see exactly the same neighborhood.

- So these **green** nodes must decide the same way!
Lower Bound for MDS: Intuition (3)

- But however they decide, one way will be devastating (with $n = m^2$)!

$|\text{DS}_{\text{OPT}}| = 2.$

$|\text{DS}_{\text{OPT without green}}| \geq m.$

$|\text{DS}_{\text{OPT with green}}| > n$
Graph Used in the Lower Bound

- The example is for $t = 3$.
- All edges are in fact special bipartite graphs with large enough girth.
Lower Bounds

- Results: Many “local looking” problems need non-trivial $t$.
- E.g., a polylogarithmic dominating set approximation (or a maximal independent set, etc.) needs at least $\Omega(\log \Delta)$ and $\Omega(\log^{\frac{1}{2}} n)$ time.

Local Algorithms ("Tight" Lower & Upper Bounds)

- \( \log^* n \)
- About \( \log n \)

**Diameter**

- Growth-Bounded Graphs (different problems)
  - E.g., dominating set approximation in planar graphs

- Approximations of dominating set, vertex cover, etc.

- MIS, maximal matching, etc.

- Covering and packing LPs

- MST, Sum, etc.
Summary & Open Problems

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

Network Mobility

Agents
Thank You!

Questions & Comments?

Thanks to my co-authors
Fabian Kuhn
Christoph Lenzen
Thomas Moscibroda
Thomas Locher
Johannes Schneider
Philipp Sommer

www.disco.ethz.ch
Let’s get Physical!
Let’s get Physical!

Roger Wattenhofer