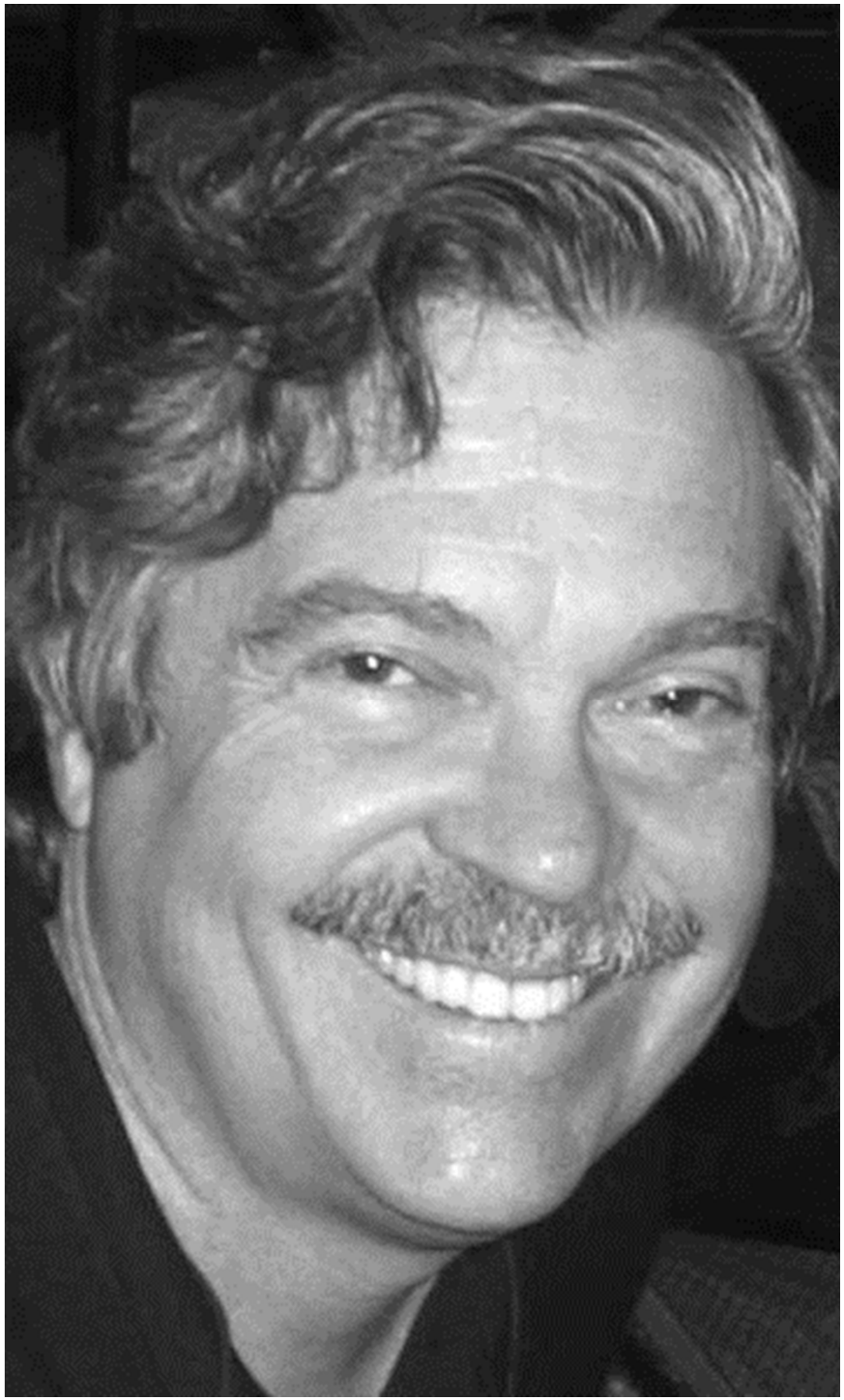

Theory Meets Practice
...it's about TIME!
Roger Wattenhofer
ETH Zurich – Distributed Computing – www.disco.ethz.ch

„People who are really serious about software should make their own hardware.”

Alan Kay



„People who are
really serious
about algorithms
should make their
own software.”

… or wait a long time for
algorithms to be discovered.

BitThief-Worded T-shirt from Zazzle.com - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.zazzle.com/bitthief_worded_shirt-235172699316920973

Stats eDoz ConfSearch Libra dvd

ZAZZLE
infinite one-of-a-kind-ness

Shopping Cart | Help | MyZazzle | Login

WHAT'S HOT | CATEGORIES | PARTICIPATE | CREATE

search Go

BitThief-Worded T-shirt by bit_thief

(0 votes)

Views: Model Product Design

front back

Ladies Basic T-Shirt

(starting at $21.00)

The classic t-shirt, made specifically for women. Pre-shrunk, 5.0 ounce 100% super-soft cotton, baby jersey knit. Coverstitched 3/4" bottom hem and sleeve opening. Custom contoured fit. Made by Bella.

Men Women Kid Baby

All Classic Fashion Sport Sustainable (info)

Color: Yellow (+$2.55) (info for dark colors)

Size: Adult 2X view size chart

S M L XL 2X

Model: Lindsay

Qty: 1 (save in bulk) $23.55 Add to cart

Customize: Change the design, add your own ideas! Customize it

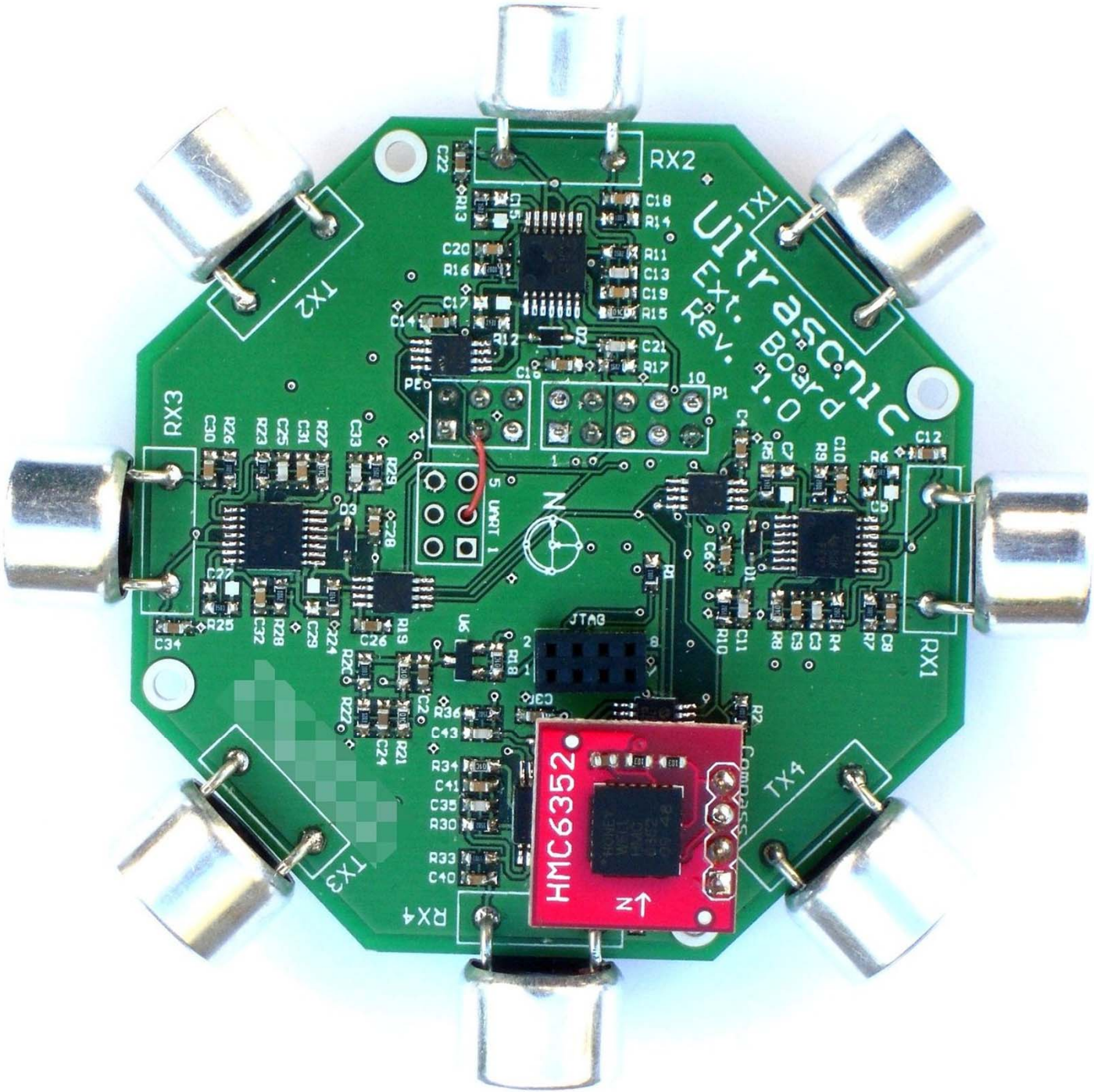
Email this | Link to this | Add to favorites | Bookmark this | Report violation

Marketplace Categories:

Arts, Design, Fashion > Graphic Design > Graffiti > Digital Graffiti

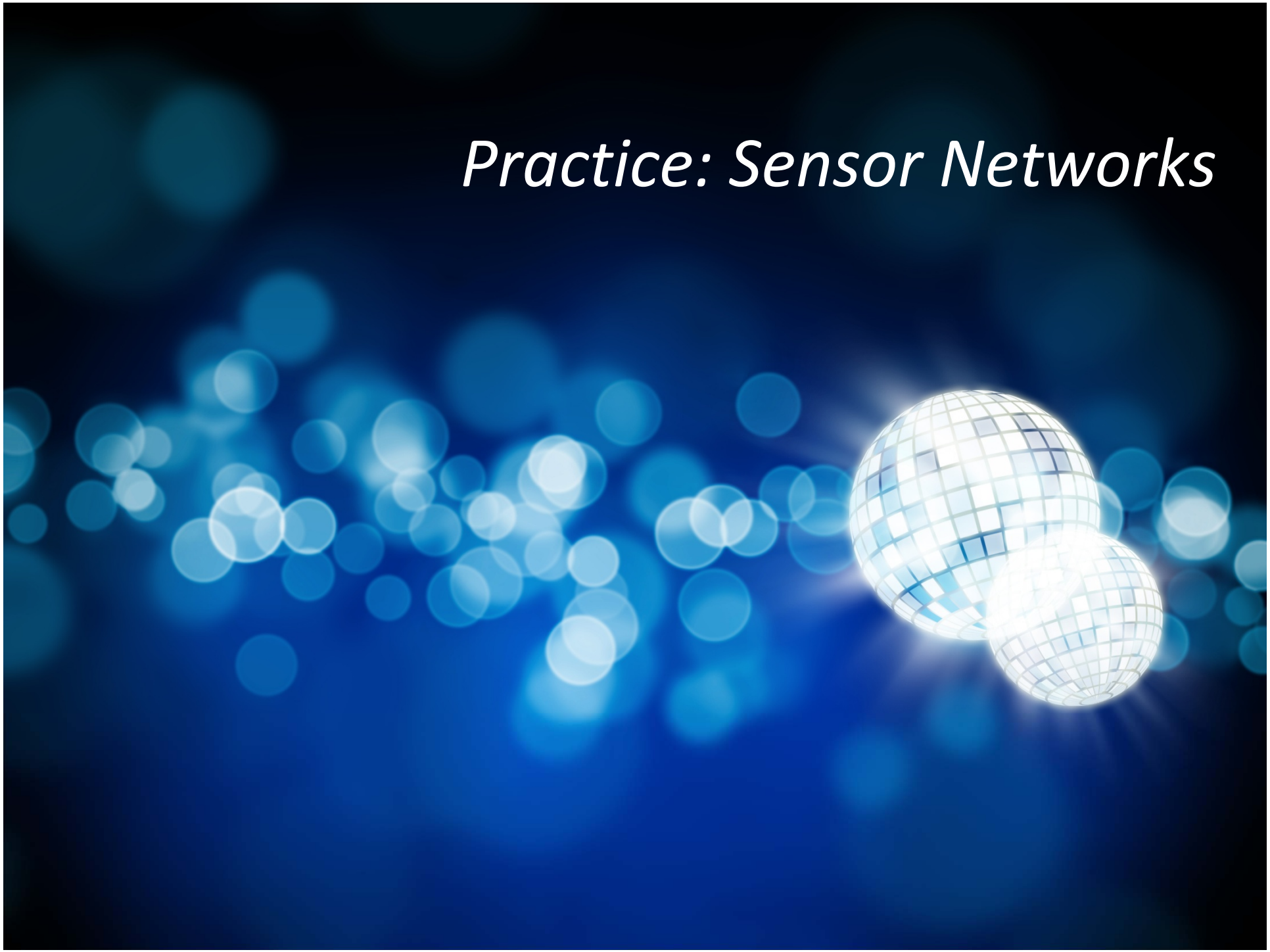
Find: henz Next Previous Highlight all Match case

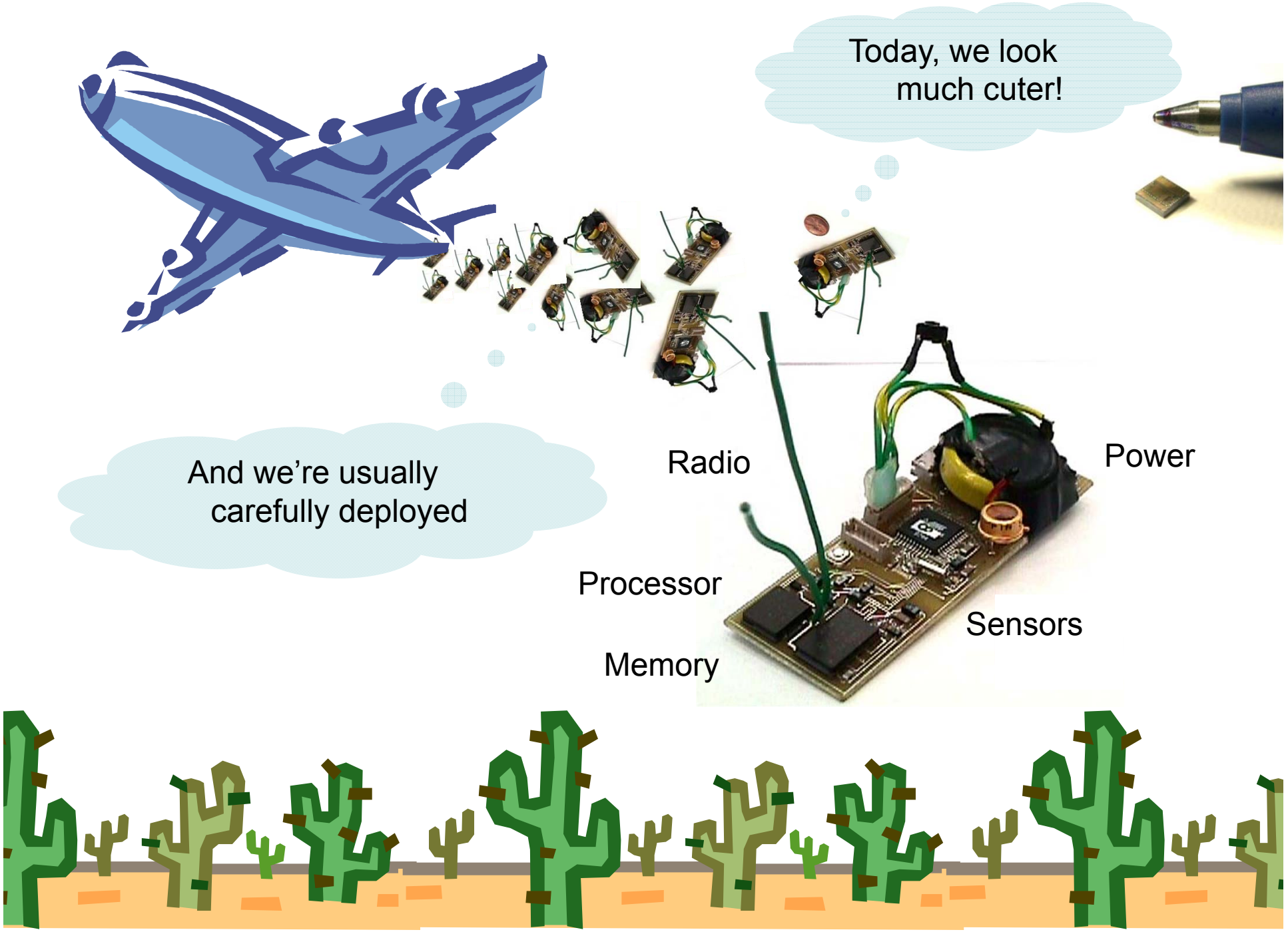
Done Adblock


RX2
TX1
TX2
RX3
RX1
TX4
TX3
RX4
Ultrasonic
Ext. Board
Rev. 1.0
JTAG
HMC6352

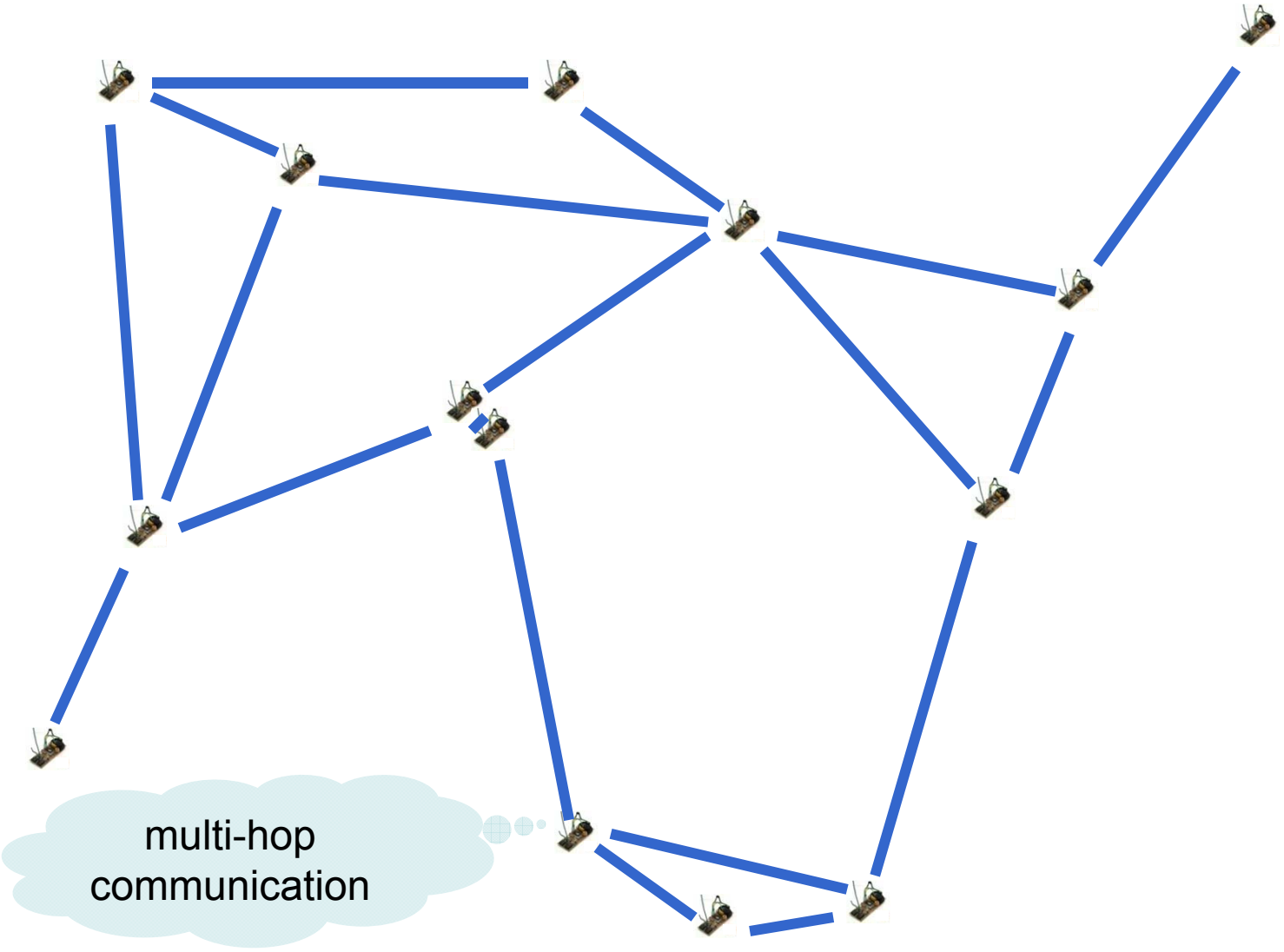
Theory Meets Practice?

Practice: Sensor Networks




Today, we look much cuter!
And we're usually carefully deployed
Radio
Power
Processor
Sensors
Memory

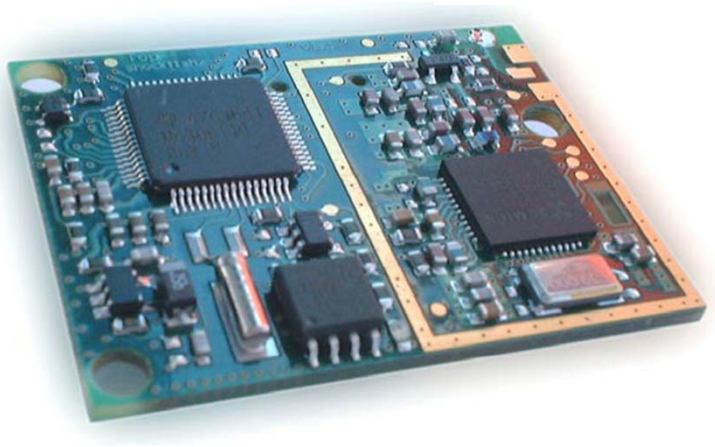
A Sensor Network After Deployment



A Typical Sensor Node: TinyNode 584

- TI MSP430F1611 microcontroller @ 8 MHz
- 10k SRAM, 48k flash (code), 512k serial storage
- 868 MHz Xemics XE1205 multi channel radio
- Up to 115 kbps data rate, 200m outdoor range

	Current Draw	Power Consumption
uC sleep with timer on	6.5 uA	0.0195 mW
uC active, radio off	2.1 mA	6.3 mW
uC active, radio idle listening	16 mA	48 mW
uC active, radio TX/RX at +12dBm	62 mA	186 mW
Max. Power (uC active, radio TX/RX at +12dBm + flash write)	76.9 mA	230.7mW



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

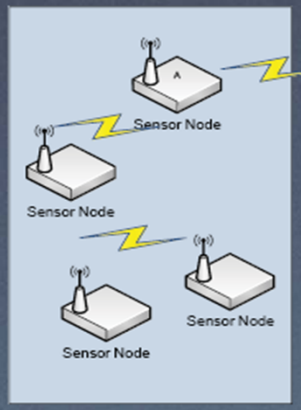
The PermaSense Project
Matterhorn Field Site Installations

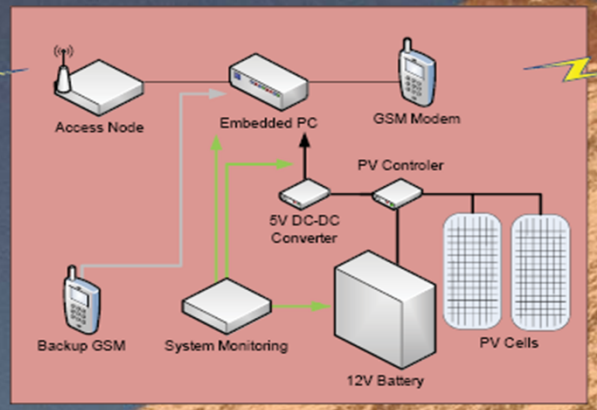
TIK
Computer Engineering and Networks Laboratory



Sensor node installations targeting 3 years unattended lifetime

Base station mounted under a combined sun/rain hood





Base station and solar panels on the field site at Matterhorn

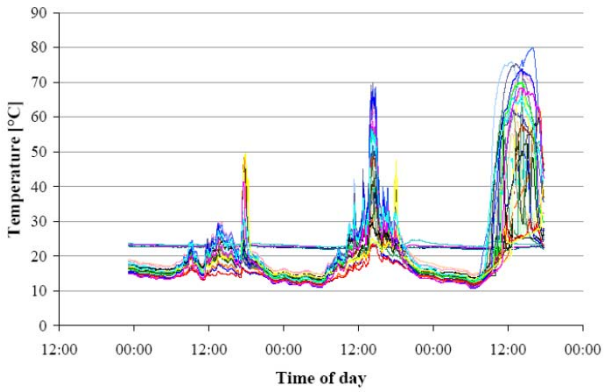

Base station power supply, system monitoring and a backup GSM modem are housed separately

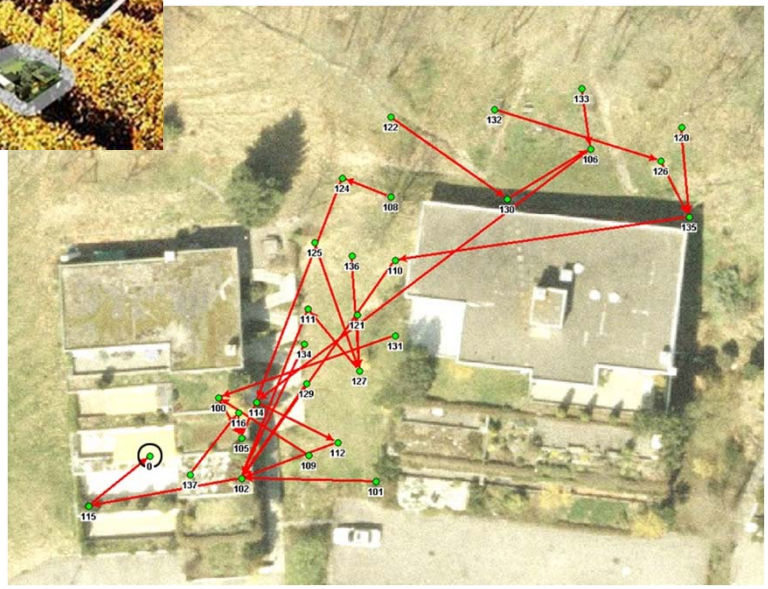
Go

Example: Dozer

- Up to 10 years of network life-time
- Mean energy consumption: 0.066 mW
- Operational network in use > 3 years
- High availability, reliability (99.999%)

[Burri et al., IPSN 2007]





Is Dozer a theory-meets-practice success story?

- Good news
  - Theory people can develop good systems!
  - Sensor network (systems) people write that Dozer is one of the “best sensor network systems papers”, or: “In some sense this is the first paper I'd give someone working on communication in sensor nets, since it nails down how to do it right.”

- Bad news: Dozer does not have an awful lot of theory inside
- Ugly news: Dozer v2 has even less theory than Dozer v1
- Hope: Still subliminal theory ideas in Dozer?

Energy-Efficient Protocol Design

- Communication subsystem is the main energy consumer
  - Power down radio as much as possible

TinyNode	Power Consumption
uC sleep, radio off	0.015 mW
Radio idle, RX, TX	30 – 40 mW

- Issue is tackled at various layers
  - MAC
  - Topology control / clustering
  - Routing

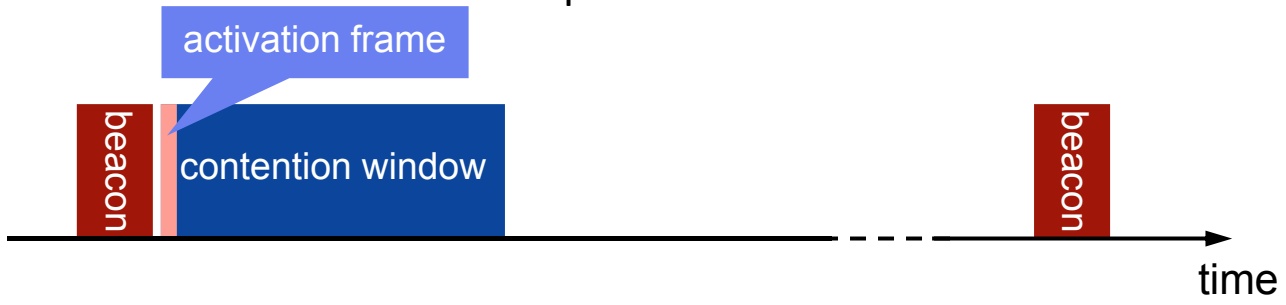
➡ Orchestration of the whole network stack to achieve duty cycles of ~ 0.1%

Dozer System

- Tree based routing towards data sink
  - No energy wastage due to multiple paths
  - Current strategy: SPT

- TDMA based link scheduling
  - Each node has two independent schedules
  - No global time synchronization

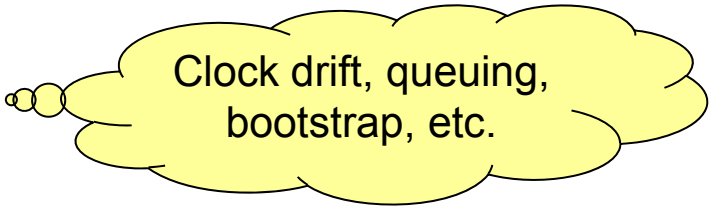
- The parent initiates each TDMA round with a beacon
  - Enables integration of disconnected nodes
  - Children tune in to their parent's schedule

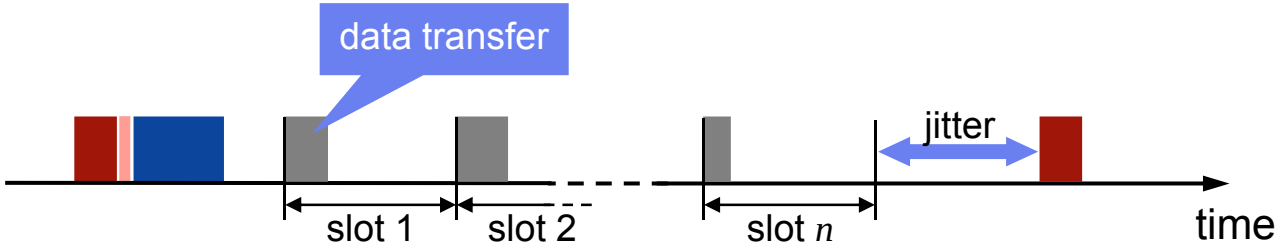


Dozer System

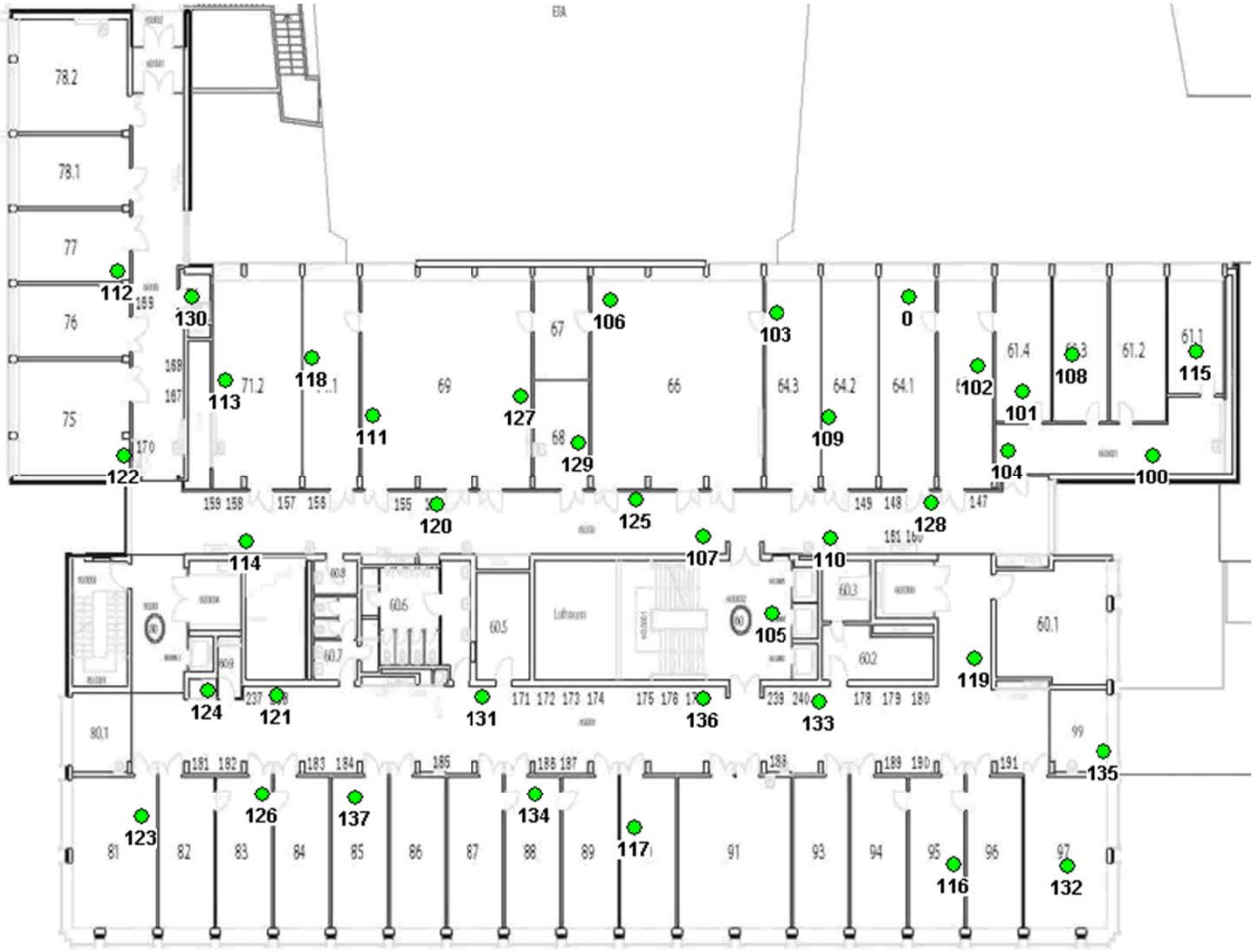
- Parent decides on its children data upload times
  - Each interval is divided into upload slots of equal length
  - Upon connecting each child gets its own slot
  - Data transmissions are always ack'ed

- No traditional MAC layer
  - Transmissions happen at exactly predetermined point in time
  - Collisions are explicitly accepted
  - Random jitter resolves schedule collisions

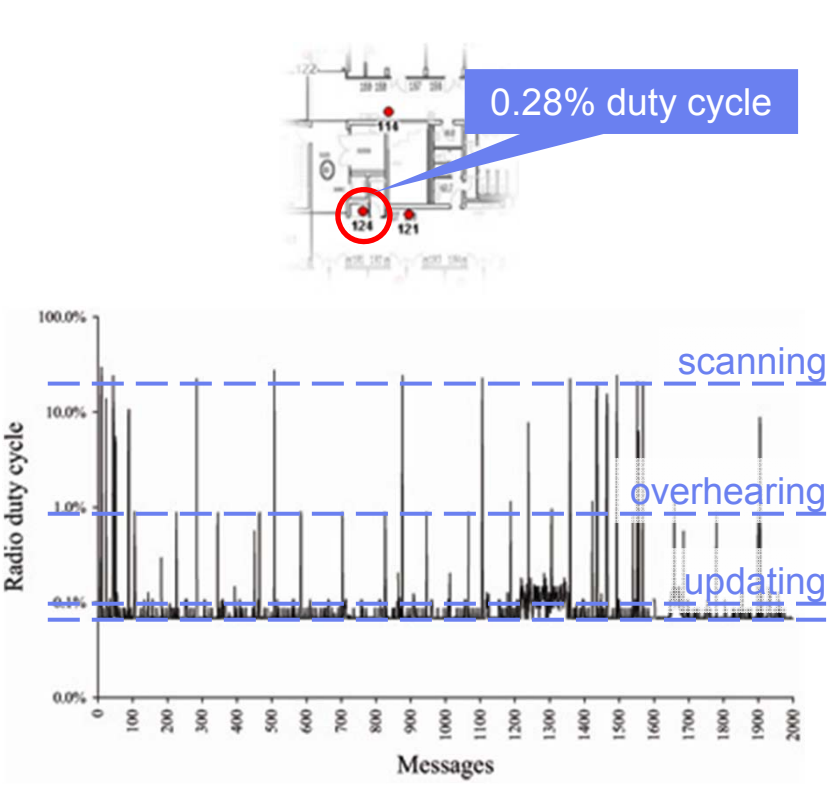


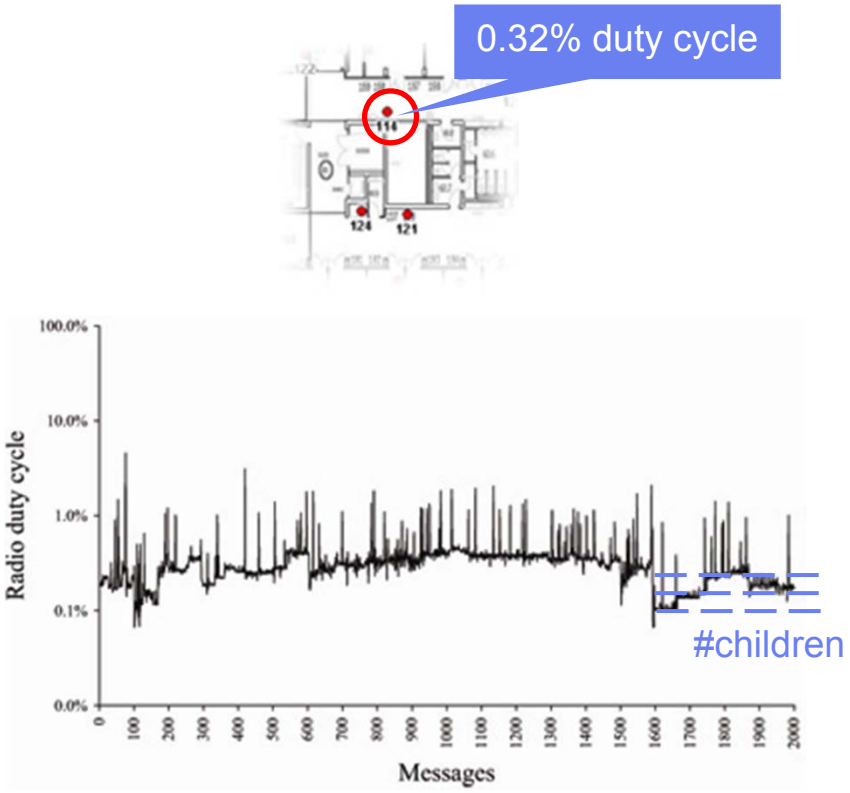


Dozer in Action



Energy Consumption





- Leaf node
- Few neighbors
- Short disruptions

- Relay node
- No scanning

no theory ☹

Theory for sensor networks, what is it good for?!

How many lines of pseudo code //
Can you implement on a sensor node?

The best algorithm is often complex //
And will not do what one expects.

Theory models made lots of progress //
Reality, however, they still don't address.

My advice: invest your research £££s //
in ... impossibility results and lower bounds!


Example: Theory of Constraints
...it's about TIME!

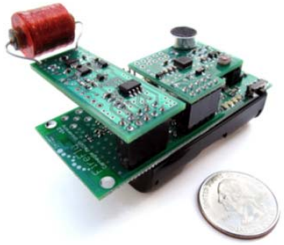
Clock Synchronization in Practice

- Many different approaches for clock synchronization

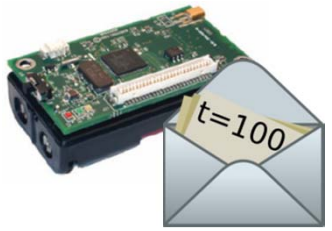
Global Positioning System (GPS)

Radio Clock Signal

AC-power line radiation

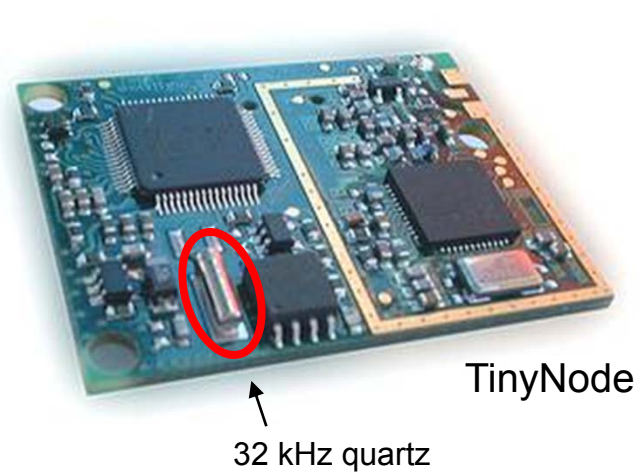


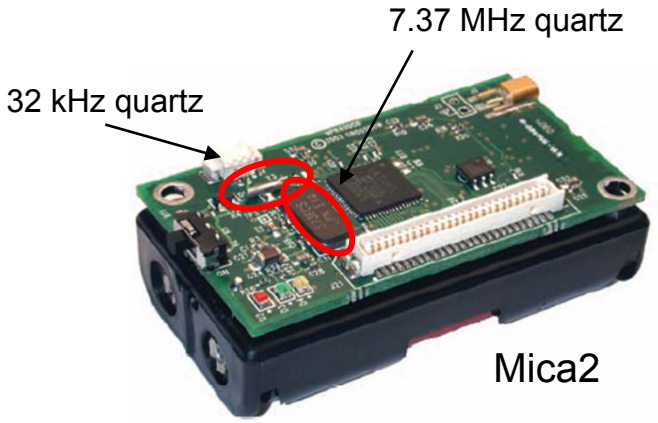
Synchronization messages



Oscillators

- Structure
  - External oscillator with a nominal frequency (e.g. 32 kHz or 7.37 MHz)
  - Counter register which is incremented with oscillator pulses
  - Works also when CPU is in sleep state

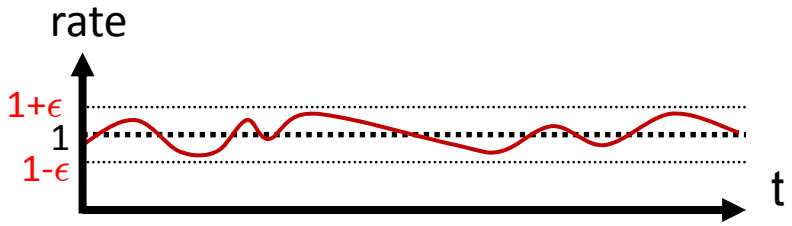




Platform	System clock	Crystal oscillator
Mica2	7.37 MHz	32 kHz, 7.37 MHz
TinyNode 584	8 MHz	32 kHz
Tmote Sky	8 MHz	32 kHz

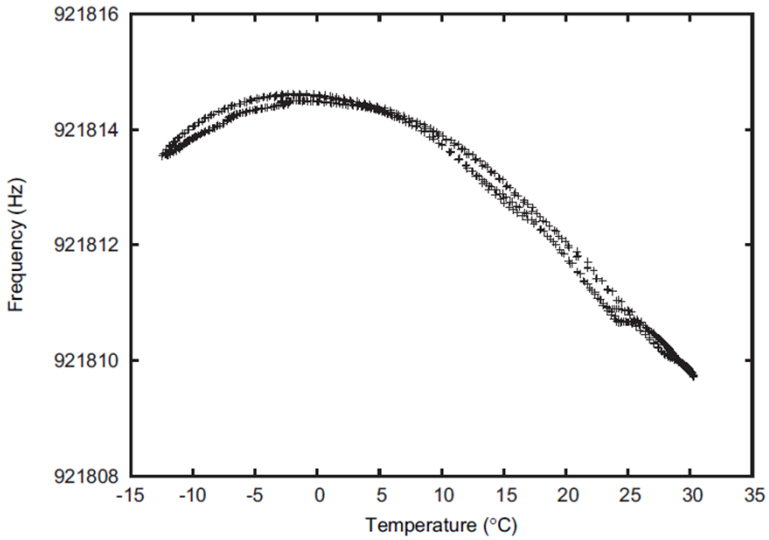
Clocks Experience Drift

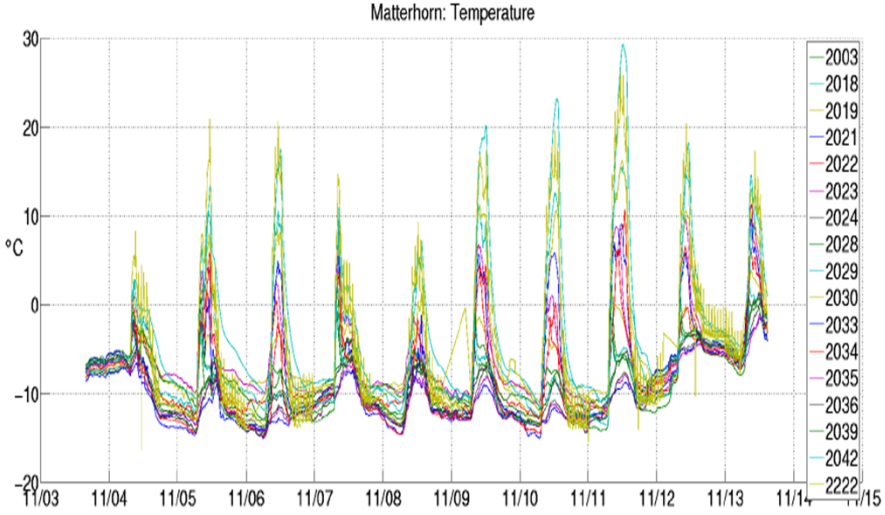
- Accuracy
  - Clock drift: random deviation from the nominal rate dependent on power supply, temperature, etc.



This is a drift of up to 50 µs per second or 0.18s per hour

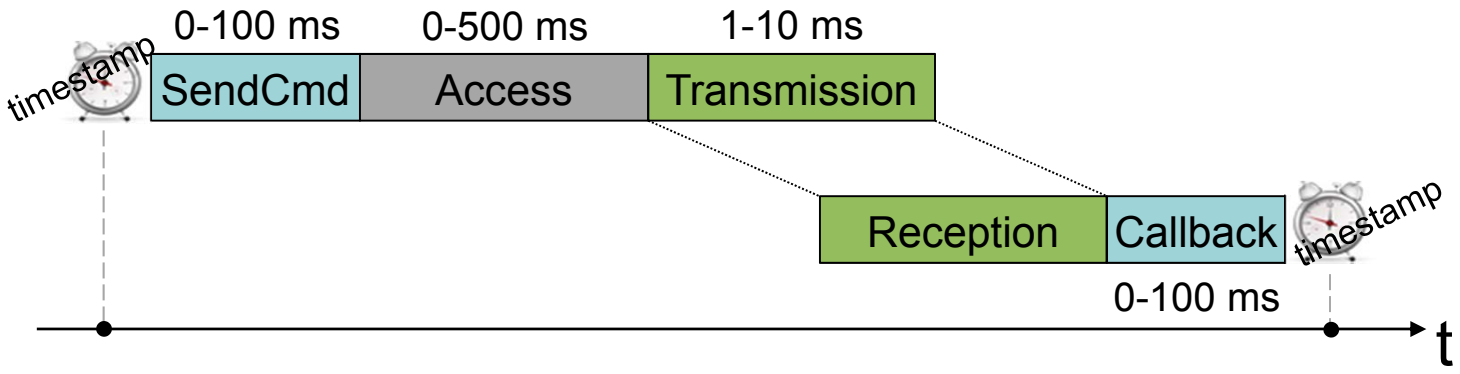
  - E.g. TinyNodes have a maximum drift of 30-50 ppm at room temperature





Messages Experience Jitter in the Delay

- Problem: Jitter in the message delay
  - Various sources of errors (deterministic and non-deterministic)



- Solution: Timestamping packets at the MAC layer [Maróti et al.]
  - → Jitter in the message delay is reduced to a few clock ticks

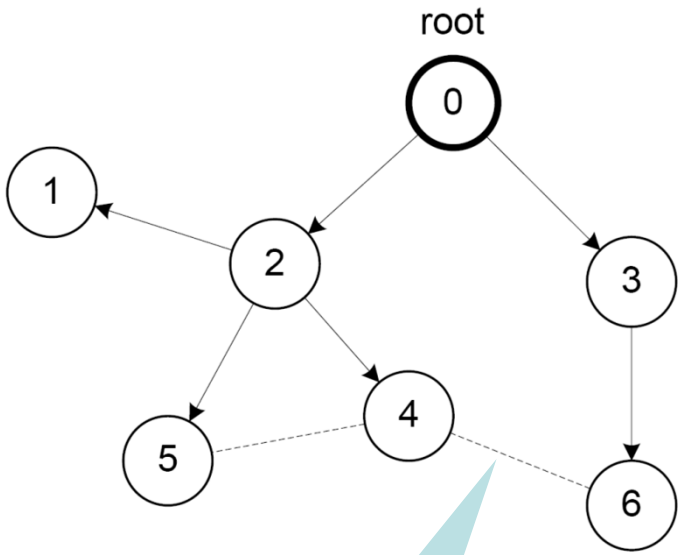
Clock Synchronization in Networks?

- *Time, Clocks, and the Ordering of Events in a Distributed System*
  L. Lamport, Communications of the ACM, 1978.
- *Internet Time Synchronization: The Network Time Protocol (NTP)*
  D. Mills, IEEE Transactions on Communications, 1991
- *Reference Broadcast Synchronization (RBS)*
  J. Elson, L. Girod and D. Estrin, OSDI 2002
- *Timing-sync Protocol for Sensor Networks (TPSN)*
  S. Ganeriwal, R. Kumar and M. Srivastava, SenSys 2003
- *Flooding Time Synchronization Protocol (FTSP)*
  M. Maróti, B. Kusy, G. Simon and Á. Lédeczi, SenSys 2004
- and many more ...

FTSP: State of the art clock sync protocol for networks.

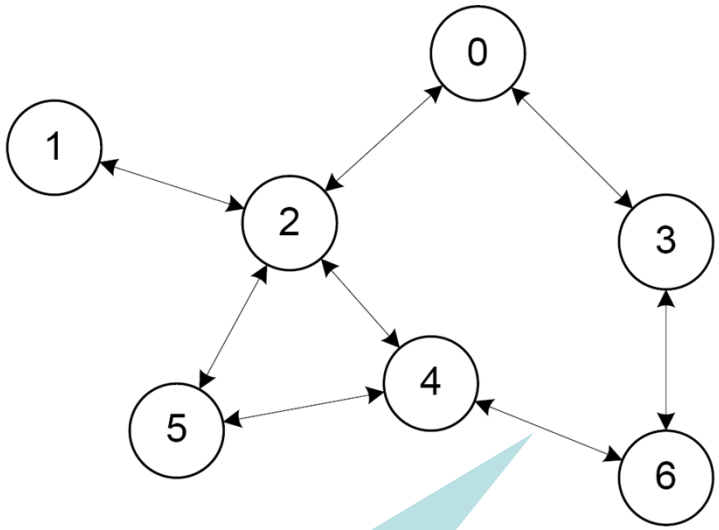
Variants of Clock Synchronization Algorithms

Tree-like Algorithms
e.g. FTSP



Bad local skew

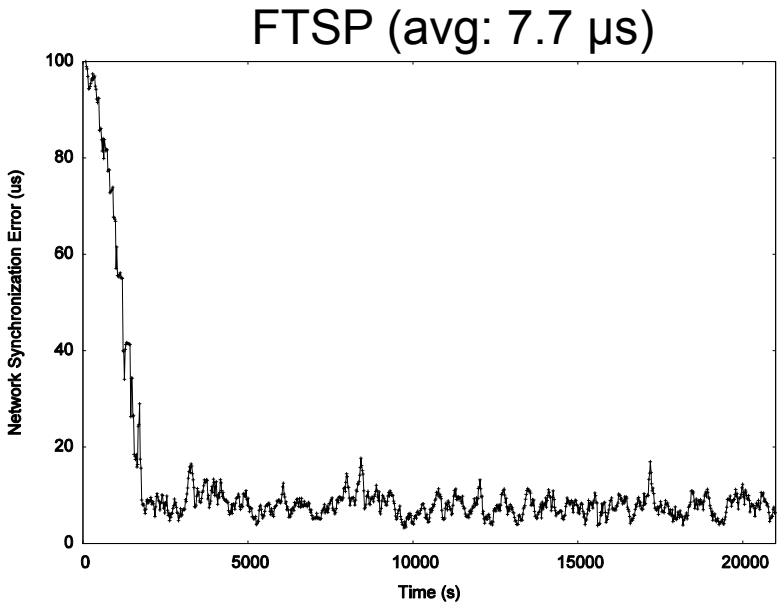
Distributed Algorithms
e.g. GTSP
[Sommer et al., IPSN 2009]

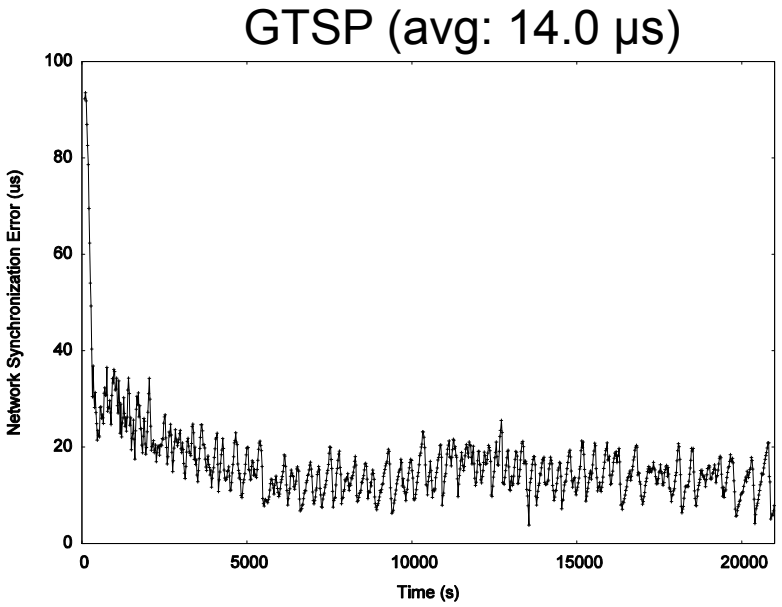


All nodes consistently average errors to *all* neighbors

FTSP vs. GTSP: Global Skew

- Network synchronization error (global skew)
  - Pair-wise synchronization error between any two nodes in the network



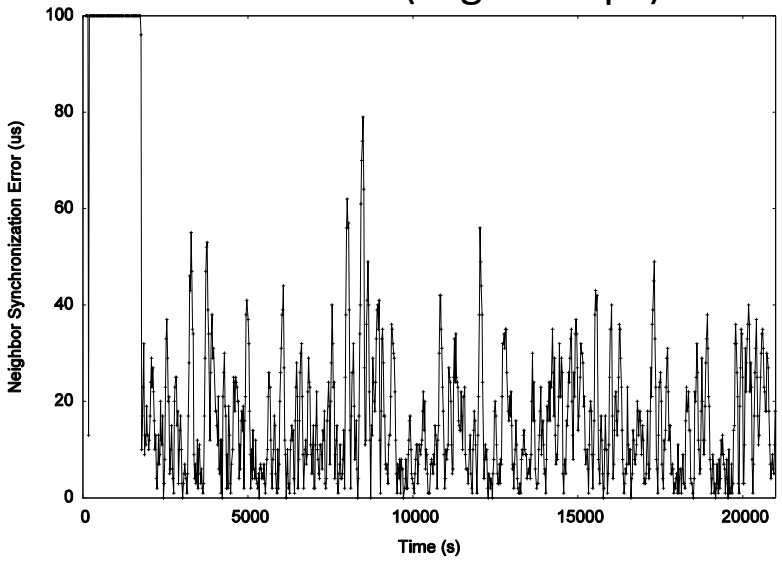


FTSP vs. GTSP: Local Skew

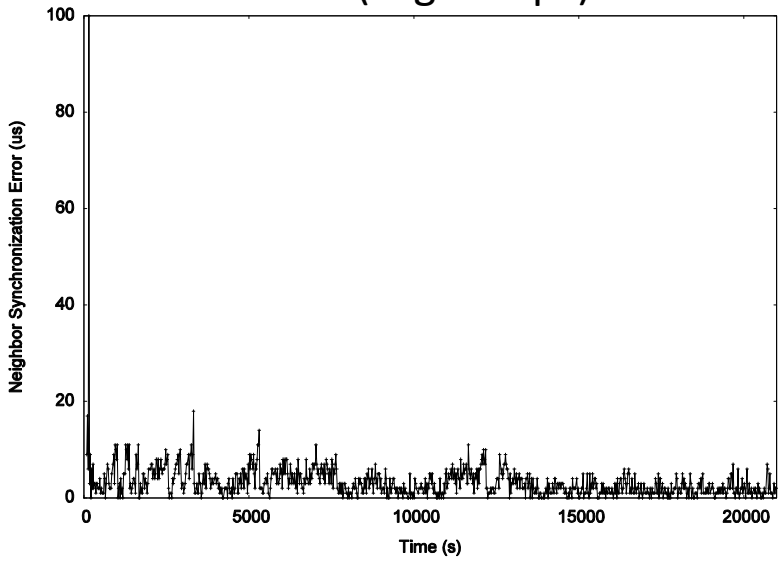
- Neighbor Synchronization error (local skew)
  - Pair-wise synchronization error between neighboring nodes

- Synchronization error between two direct neighbors:

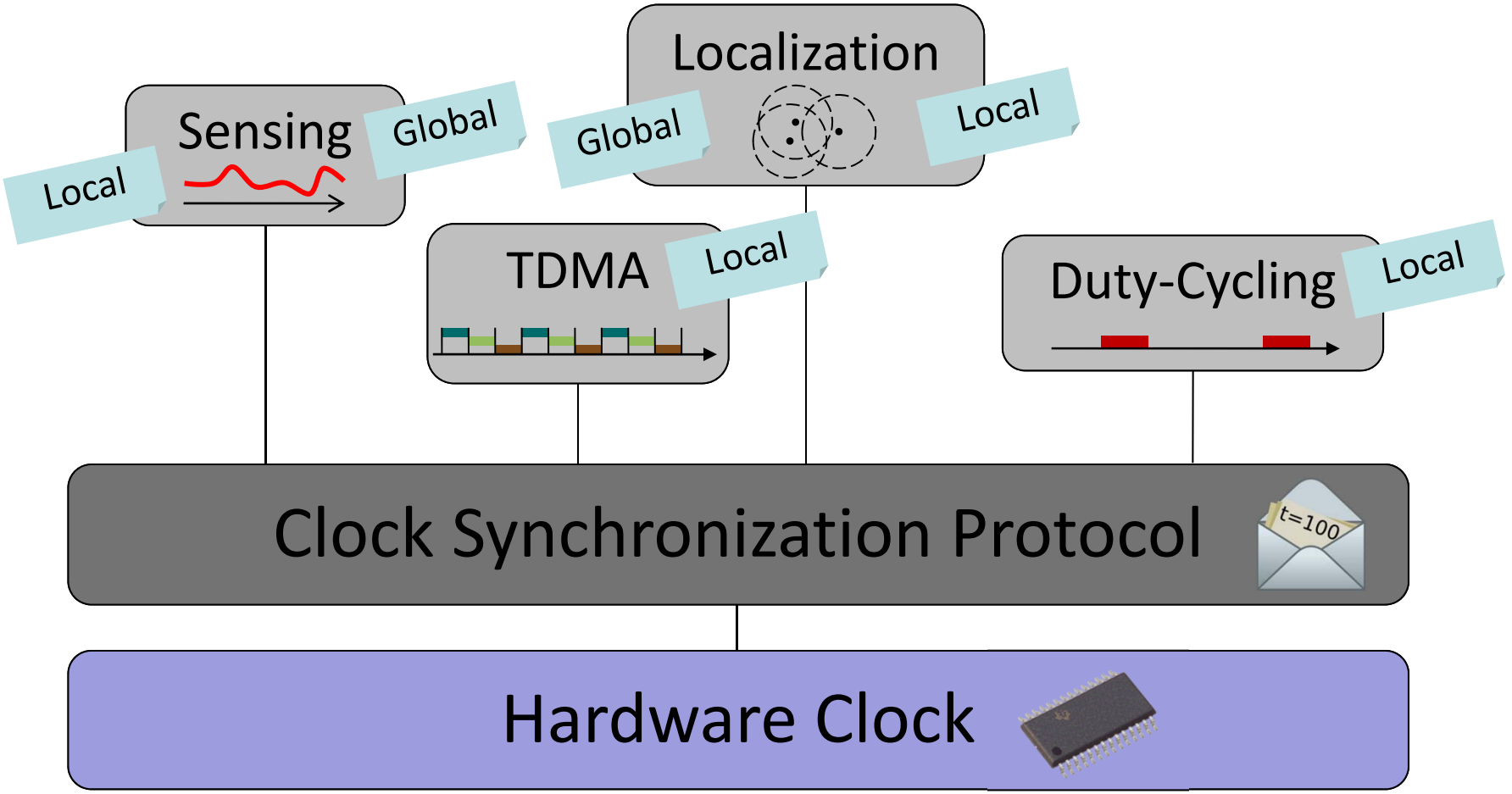
FTSP (avg: 15.0 µs)



GTSP (avg: 2.8 µs)



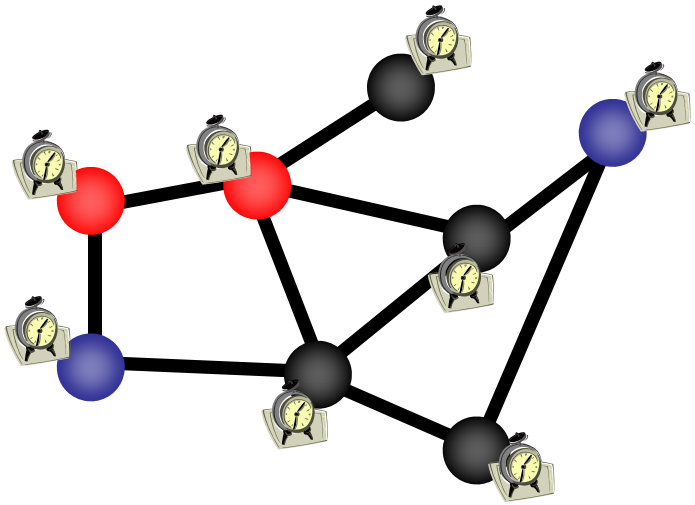
Time in (Sensor) Networks



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

- 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!

Formal Model

- Hardware clock $H_v(t) = \int_{[0,t]} h_v(\tau)\, d\tau$ with clock rate $h_v(t) \in [1-\epsilon, 1+\epsilon]$

Clock drift ϵ is typically small, e.g. $\epsilon \approx 10^{-4}$ for a cheap quartz oscillator

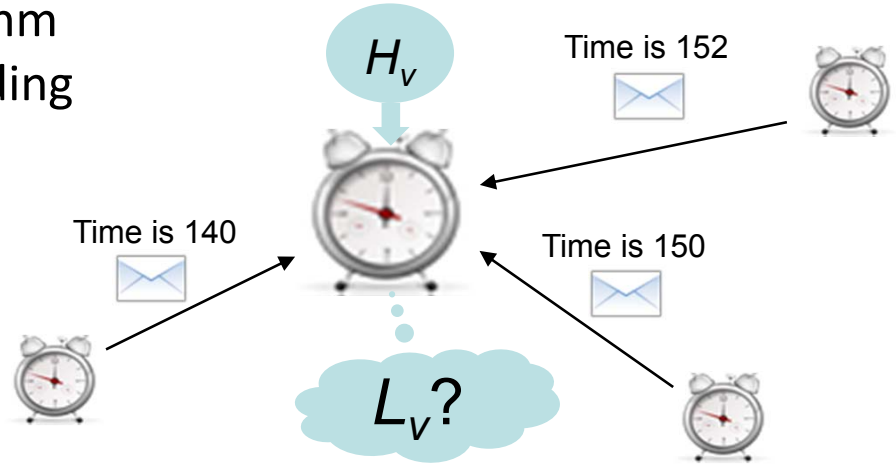
- Logical clock $L_v(\cdot)$ which increases at rate at least 1 and at most β

Logical clocks with rate much less than 1 behave differently...

- Message delays $\in [0,1]$

Neglect fixed share of delay, normalize jitter

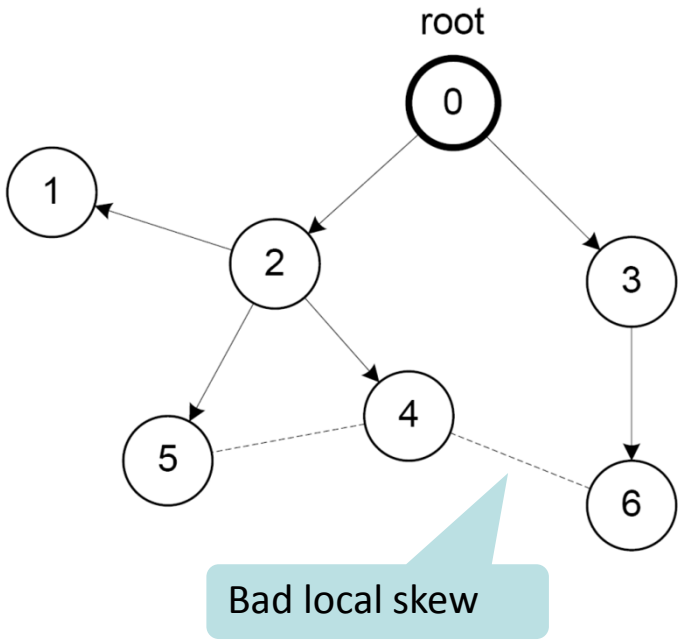
- Employ a synchronization algorithm to update the logical clock according to hardware clock and messages from neighbors



Local Skew

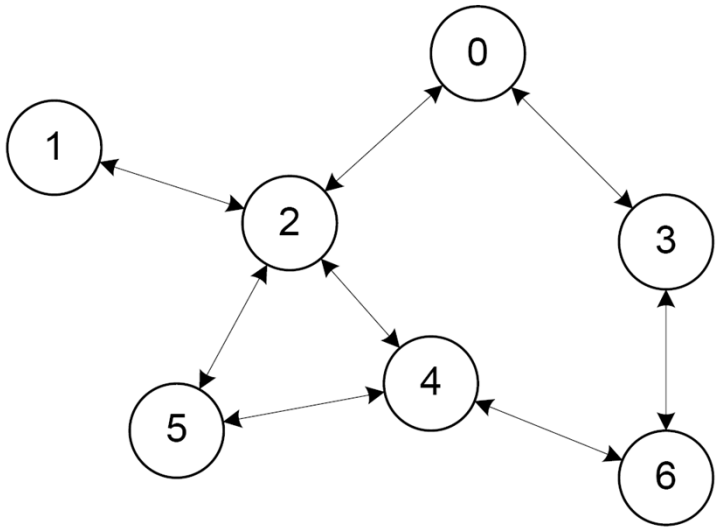
Tree-like Algorithms

e.g. FTSP



Distributed Algorithms

e.g. GTSP

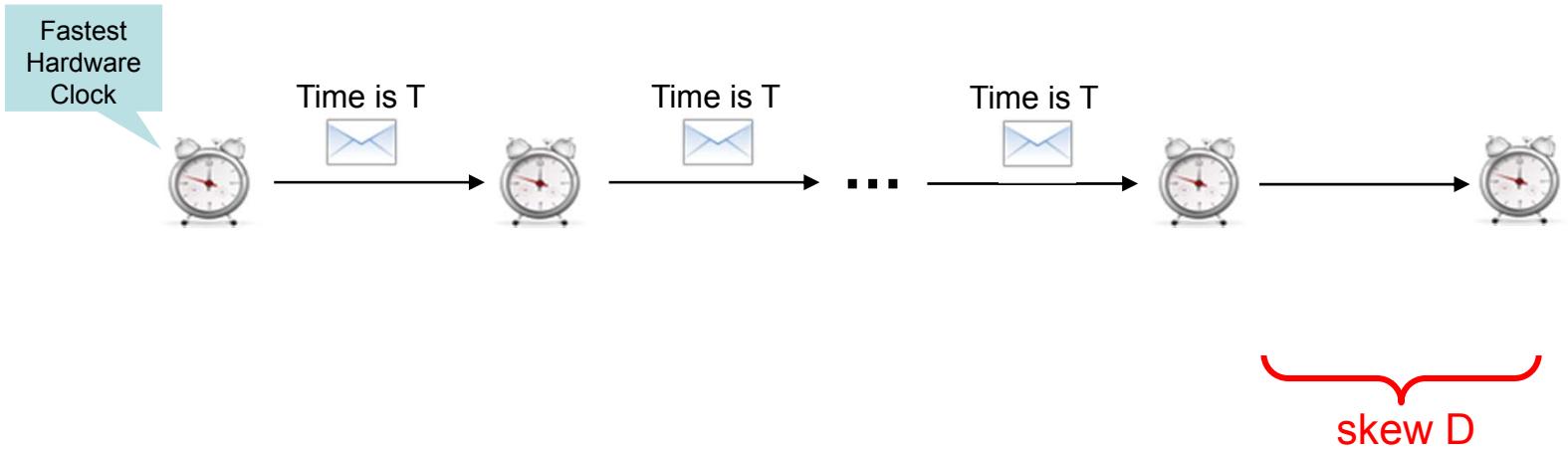


Synchronization Algorithms: An Example ("A^{max}")

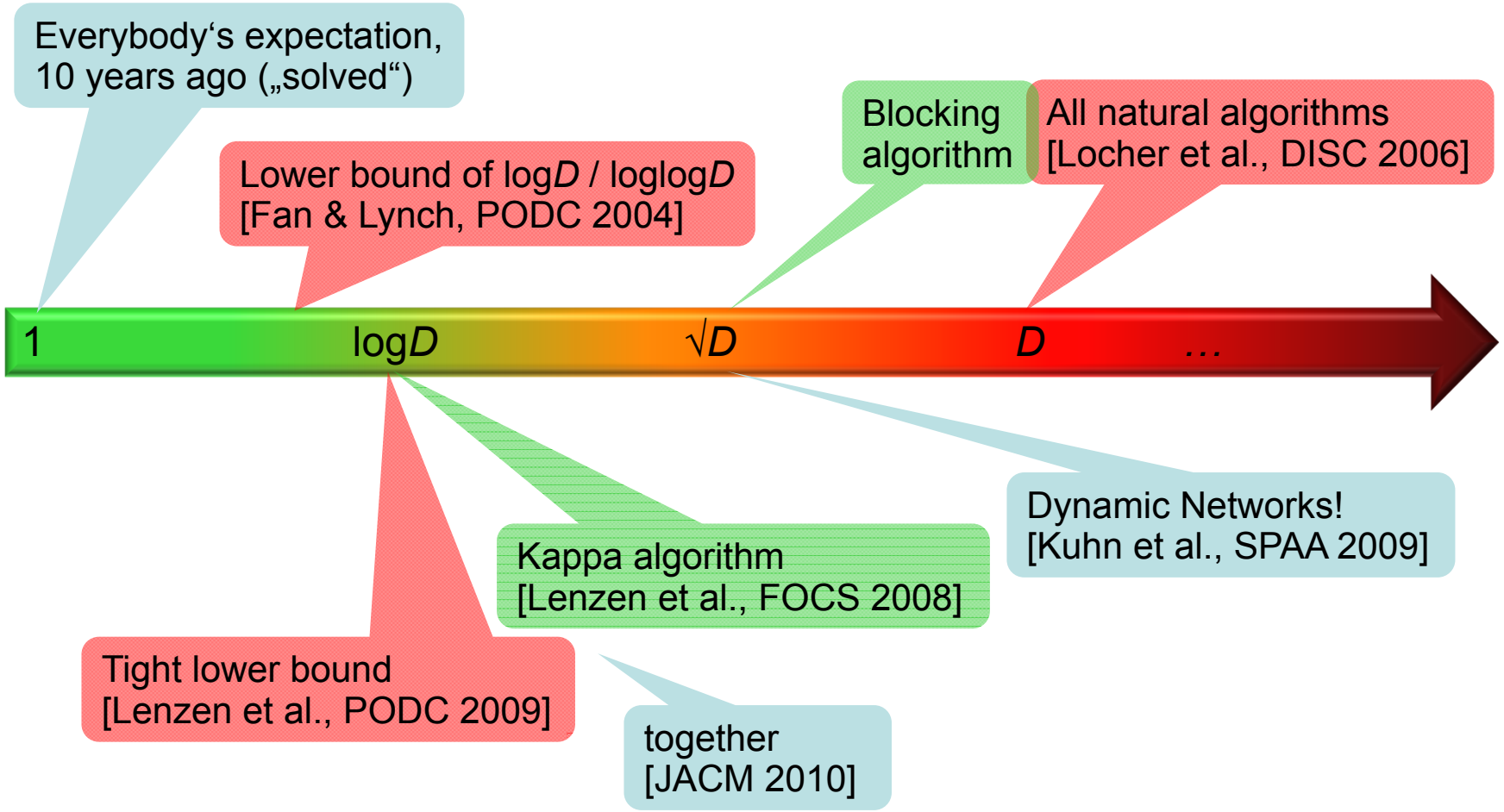
- Question: How to update the logical clock based on the messages from the neighbors?

- 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



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

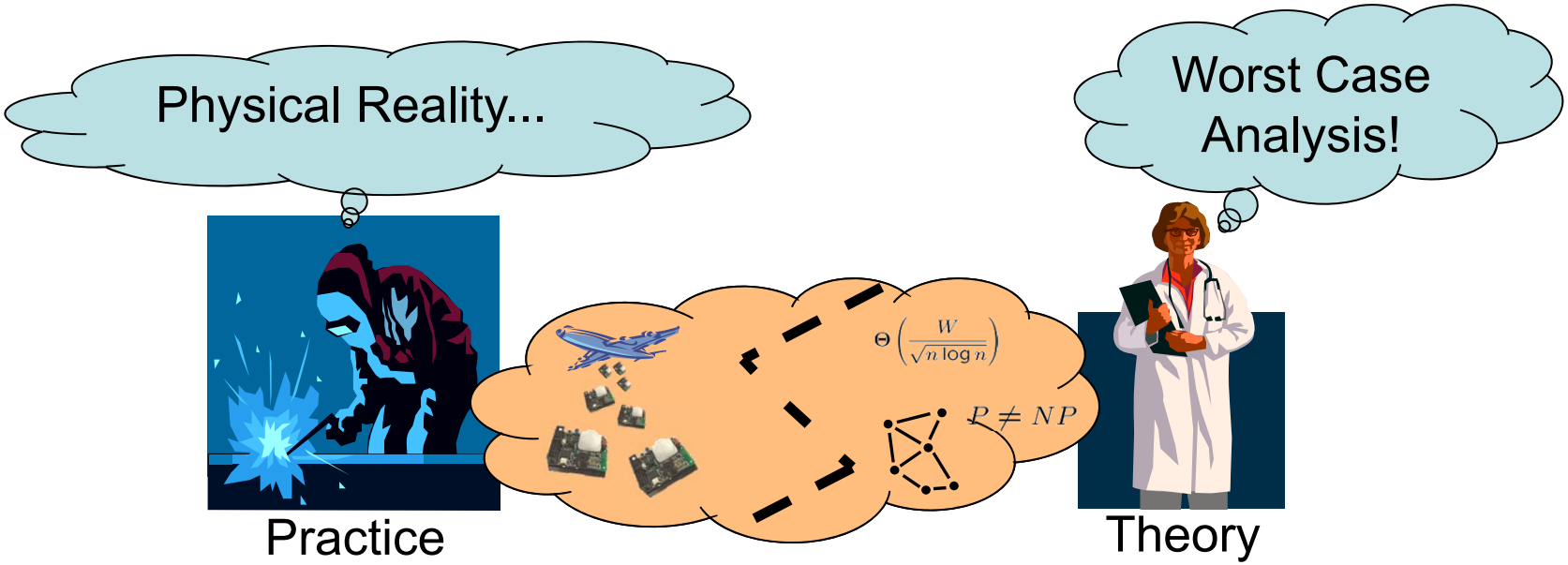
Example: Clock Synchronization?!?

...it's about TIME!

Roger Wattenhofer

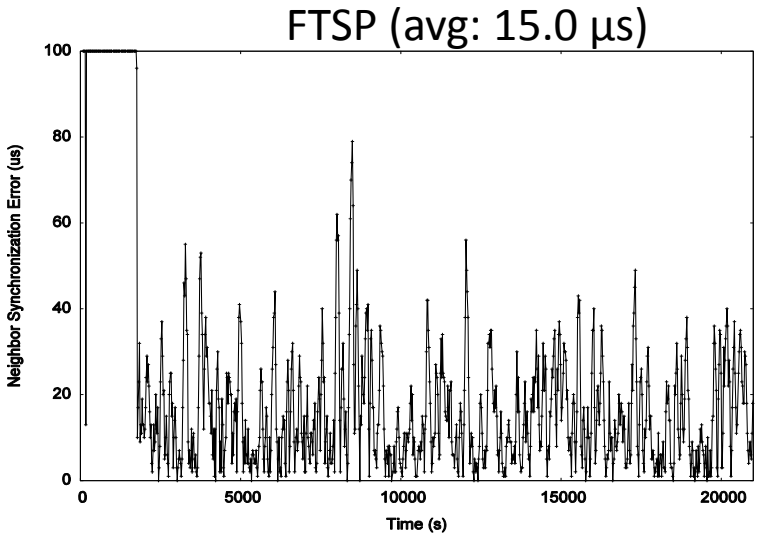
ETH Zurich – Distributed Computing Group

One Big Difference Between Theory and Practice, Usually!

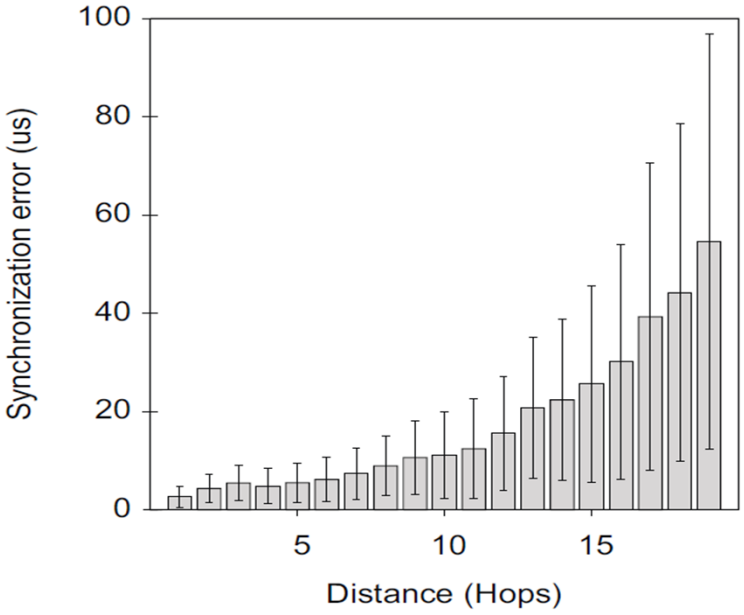


„Industry Standard“ FTSP in Practice

- As we have seen FTSP does have a local skew problem
- But it’s not all that bad…



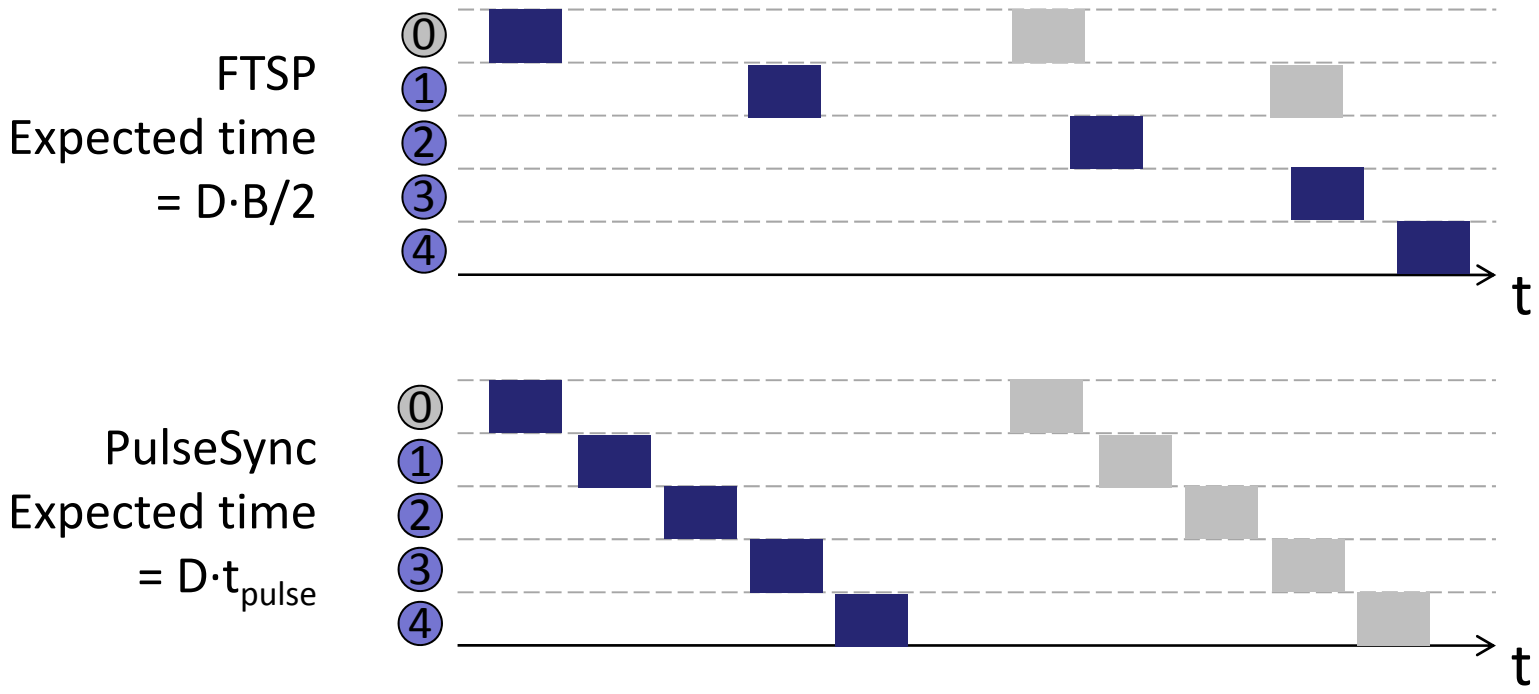
- However, tests revealed another (severe!) problem:
- FTSP does not scale: Global skew grows exponentially with network size…



The PulseSync Protocol

[Lenzen et al., S...]

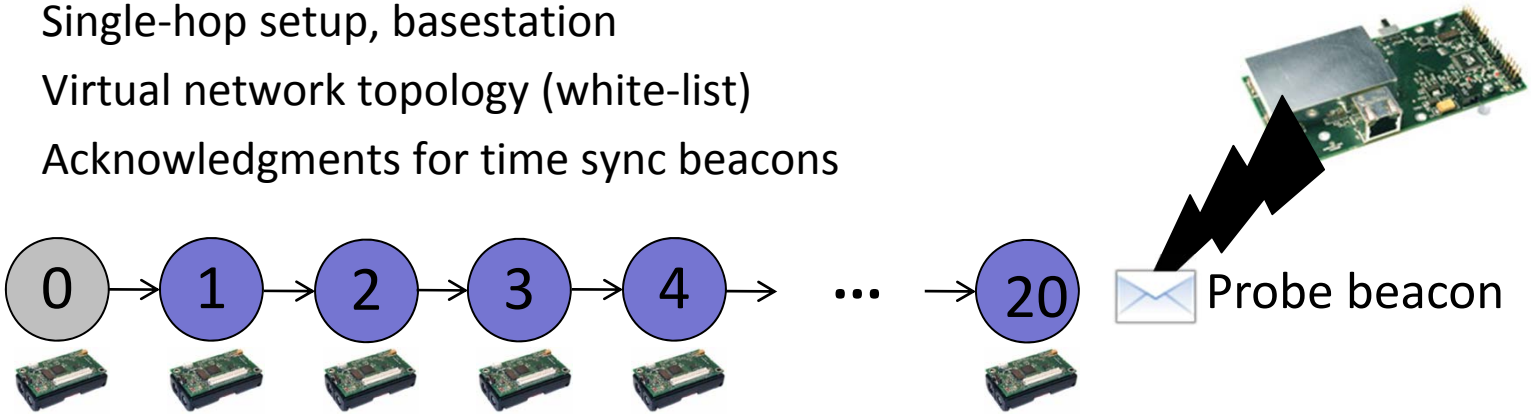
1) Remove self-amplifying of synchronization error
2) Send fast synchronization pulses through the network
   - Speed-up the initialization phase
   - Faster adaptation to changes in temperature or network topology



Evaluation

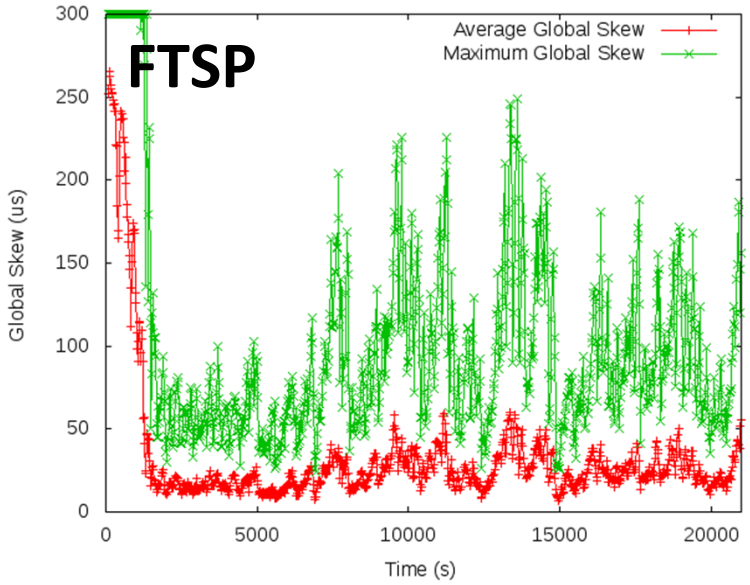
- Testbed setup
  - 20 Crossbow Mica2 sensor nodes
  - PulseSync implemented in TinyOS 2.1
  - FTSP from TinyOS 2.1

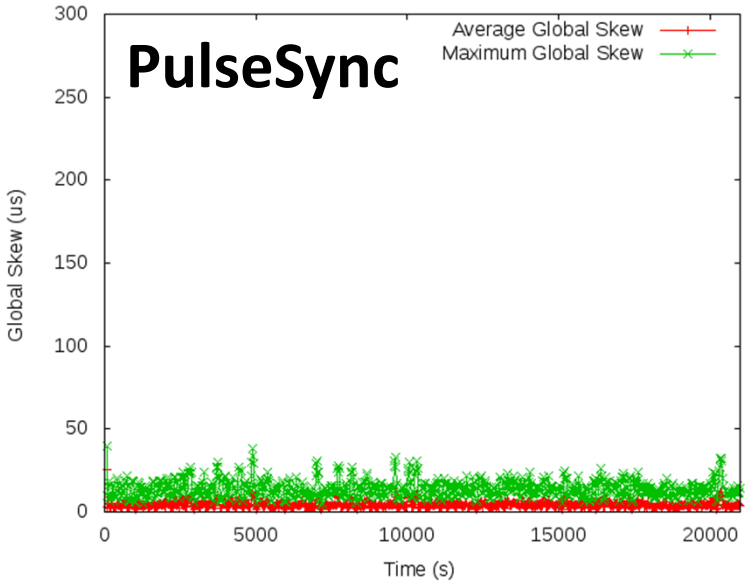
- Network topology
  - Single-hop setup, basestation
  - Virtual network topology (white-list)
  - Acknowledgments for time sync beacons



Experimental Results

- Global Clock Skew
  - Maximum synchronization error between any two nodes

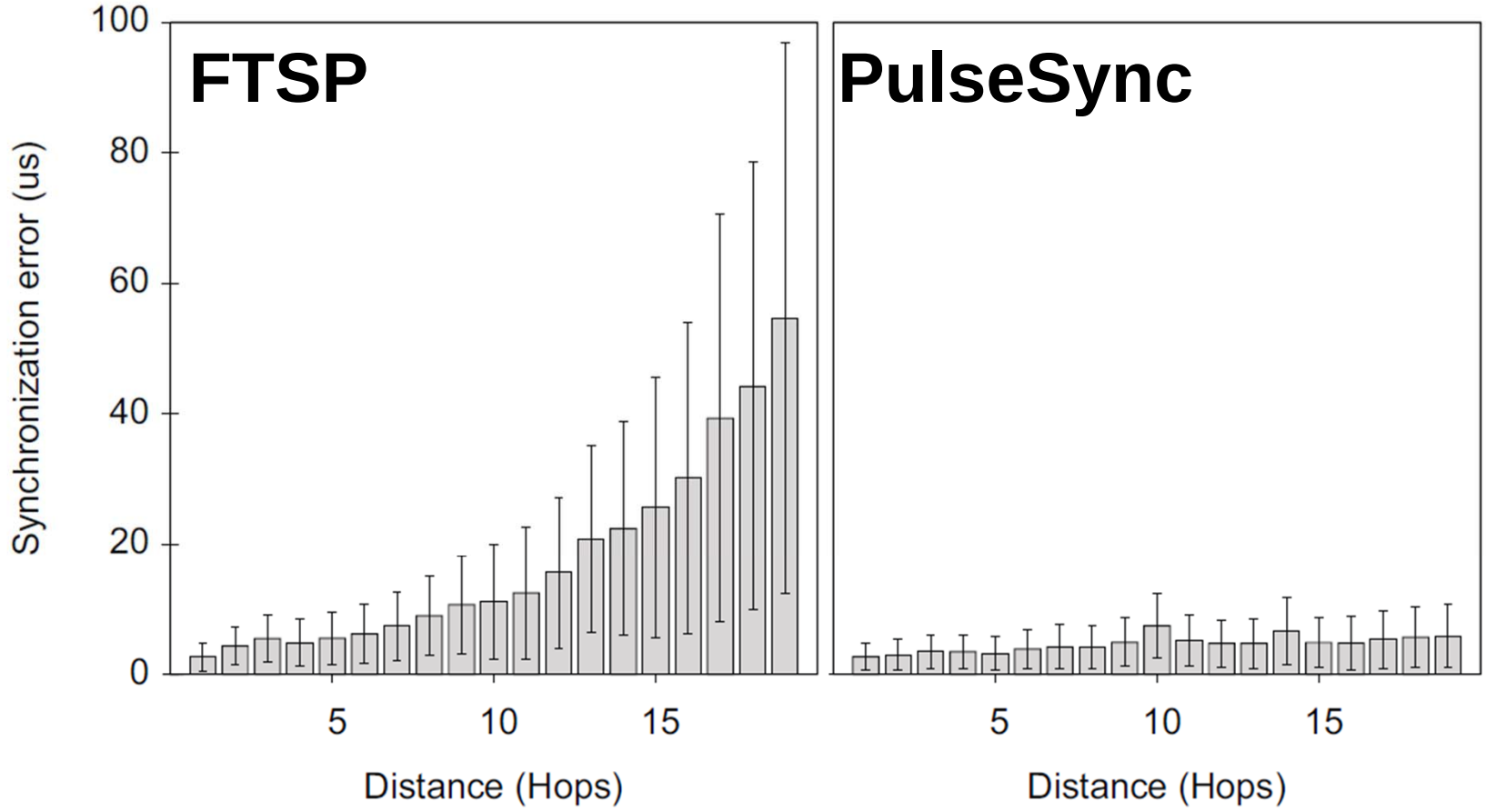




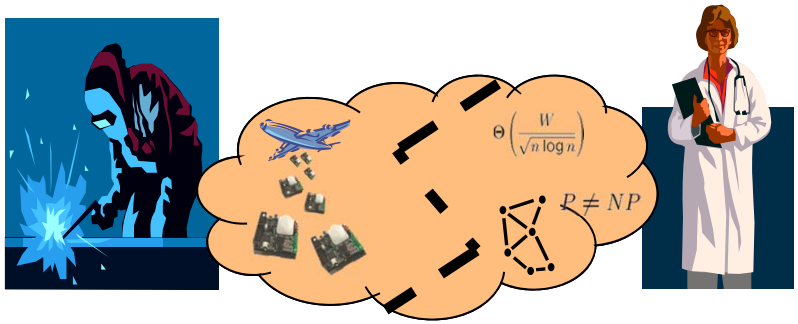
Synchronization Error	FTSP	PulseSync
Average (t>2000s)	23.96 µs	4.44 µs
Maximum (t>2000s)	249 µs	38 µs

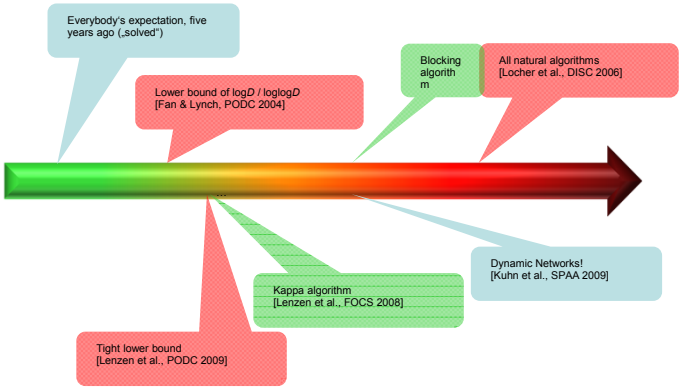
Experimental Results

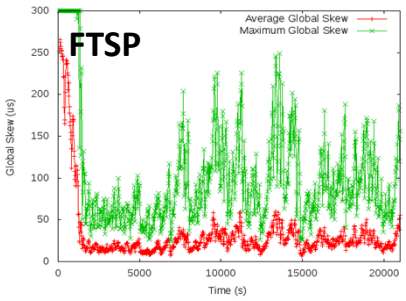
- Synchronization error vs. hop distance

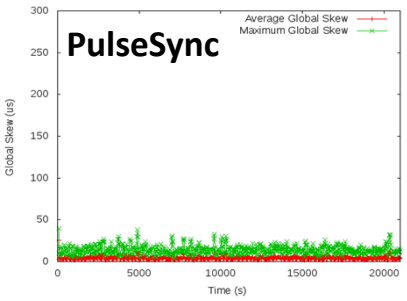


Summary


$\Theta\left(\frac{W}{\sqrt{n}\log n}\right)$
$\mathcal{P} \neq NP$

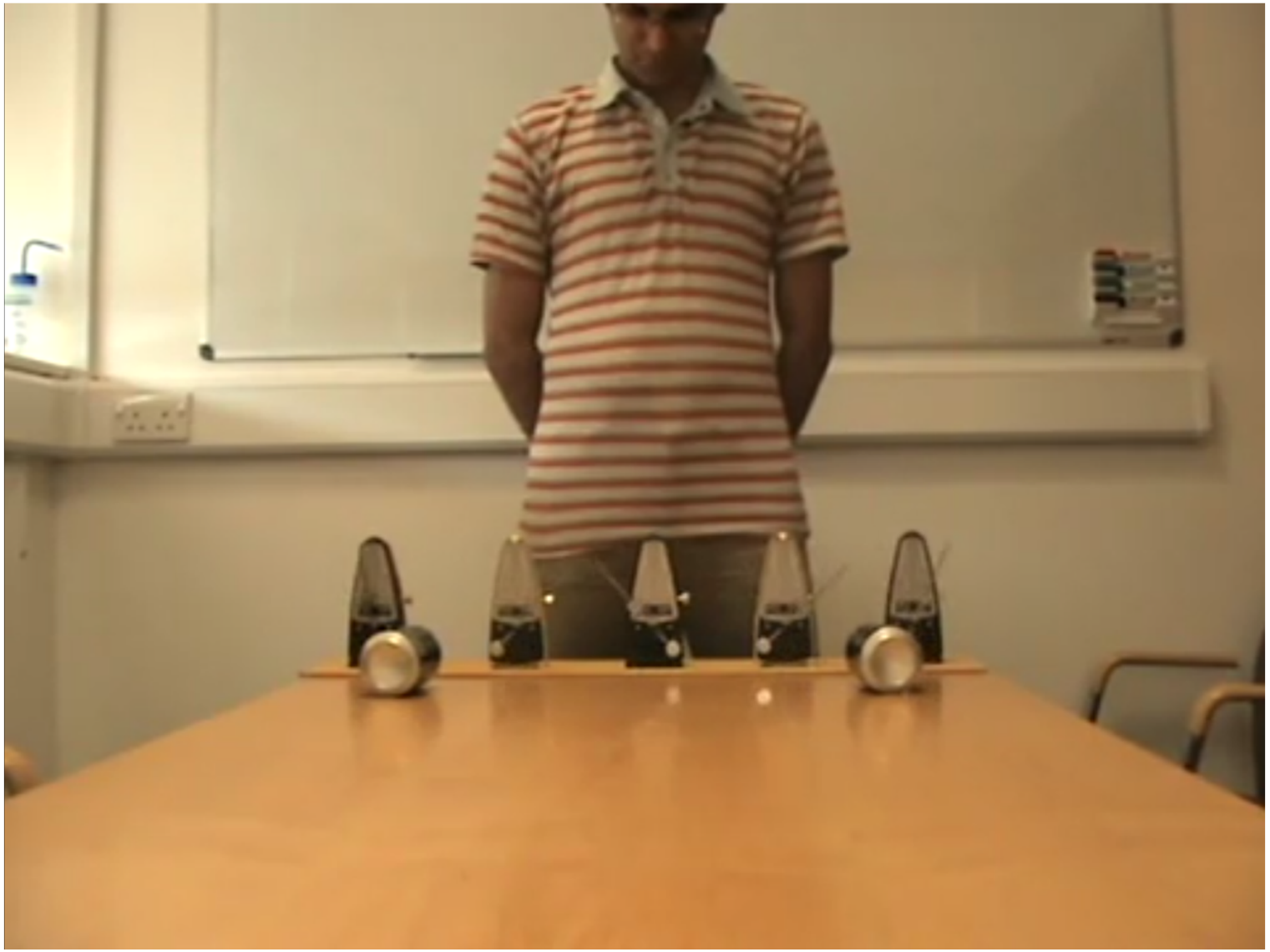

Everybody's expectation, five years ago („solved")
Lower bound of log*D* / loglog*D* [Fan & Lynch, PODC 2004]
Blocking algorithm
All natural algorithms [Locher et al., DISC 2006]
Dynamic Networks! [Kuhn et al., SPAA 2009]
Kappa algorithm [Lenzen et al., FOCS 2008]
Tight lower bound [Lenzen et al., PODC 2009]


FTSP
Average Global Skew
Maximum Global Skew
Global Skew (us)
Time (s)


PulseSync
Average Global Skew
Maximum Global Skew
Global Skew (us)
Time (s)


Merci!
Questions & Comments?
Thanks to my co-authors
Nicolas Burri
Christoph Lenzen
Thomas Locher
Philipp Sommer
Pascal von Rickenbach

YouTube Clock Synchronization



Open Problems

- global vs. local skew
- worst-case vs. reality (Gaussian?)
- accuracy vs. convergence
- accuracy vs. energy efficiency
- dynamic networks
- fault-tolerance (Byzantine clocks)
- applications, e.g. coordinating physical objects (example with cars)