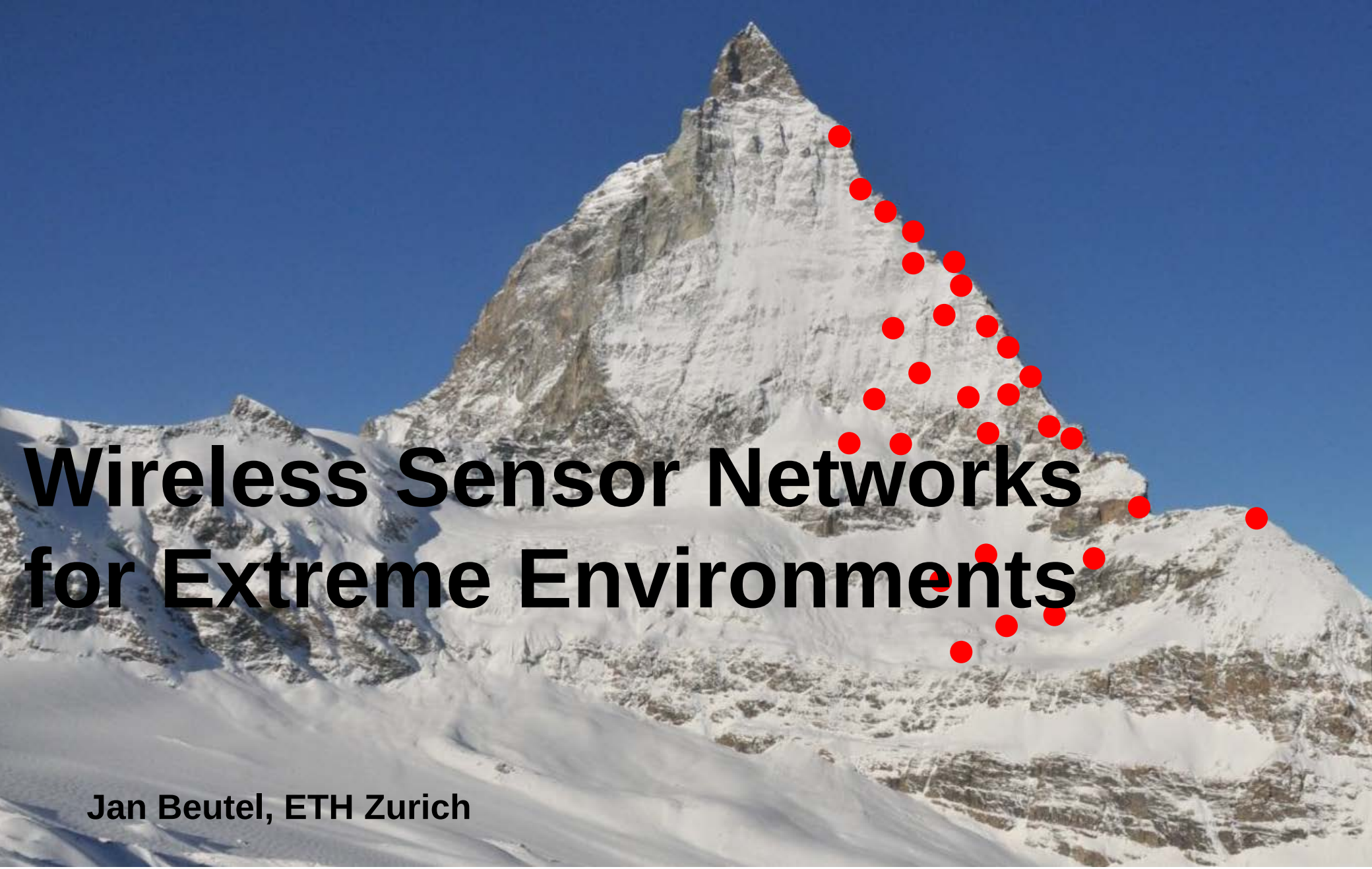


Wireless Sensor Networks for Extreme Environments

Jan Beutel, ETH Zurich

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TIK
Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks

Schedule

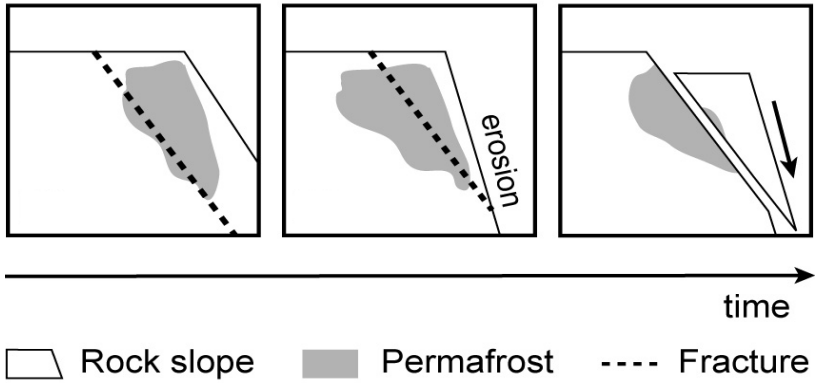
- Wireless Sensor Networks – Applications in Environmental Monitoring – PermaSense
- System Design – Low-Power Architectures – Wireless Protocols
- Test + Debugging Methodology – Examples
- End-to-end Data Management – Data Quality Analysis/Control
- PermaSense Demo – Q/A

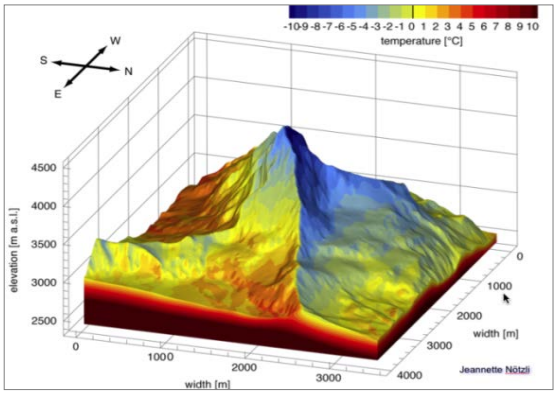
Wireless Sensor Networks Applications in Environmental Monitoring – PermaSense

PermaSense – Aims and Vision

Geo-science and engineering collaboration aiming to:

- provide **long-term high-quality** sensing in **harsh environments**
- facilitate near-complete **data recovery** and **near real-time delivery**
- obtain **better quality** data, more effectively
- obtain measurements that have **previously been impossible**
- provide **relevant information** for research or decision making, **natural hazard early-warning systems**





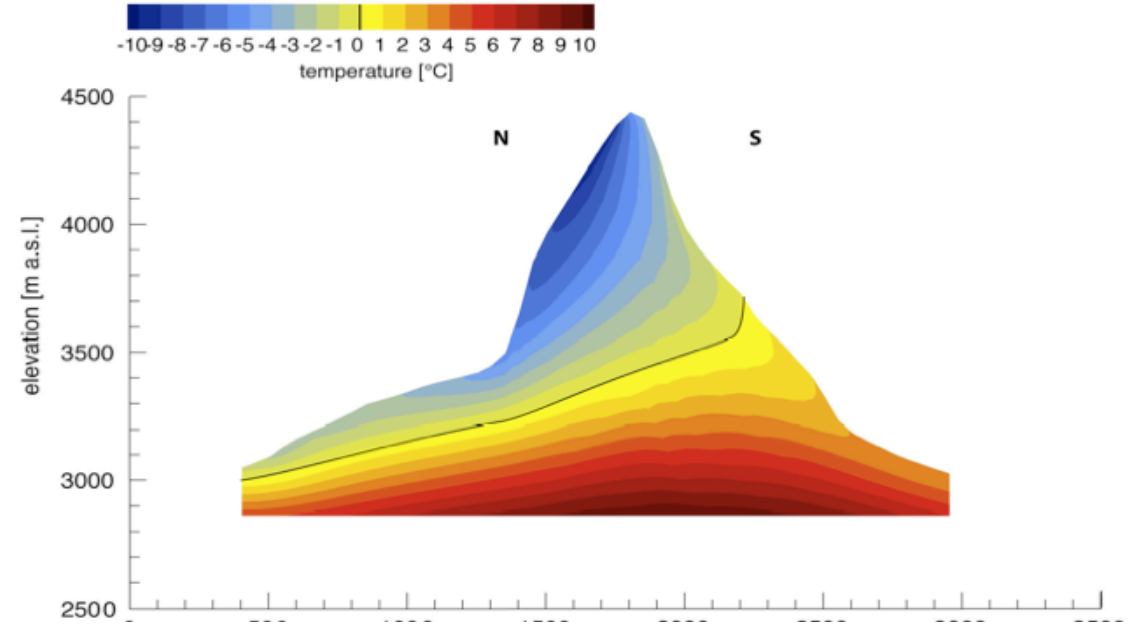
Understanding Root Causes of Catastrophes

Eiger east-face rockfall, July 2006, images courtesy of Arte Television

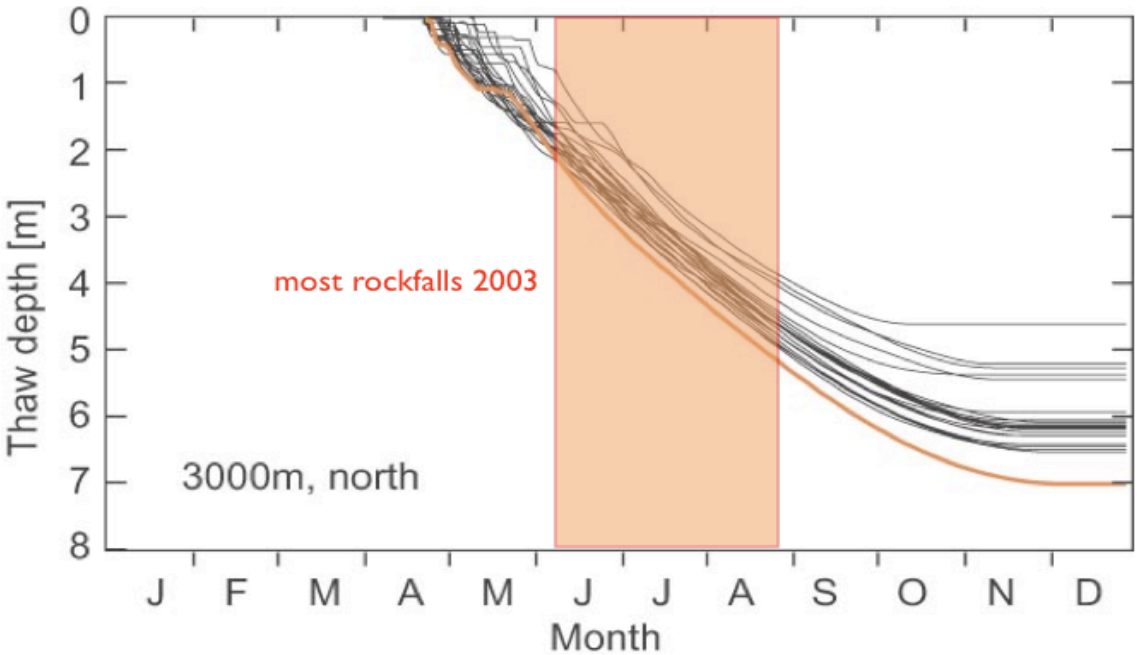

19.11.2008


16.7.2003
Photo: Bruno Jelk

- Rockfall release mechanisms and their connection to time and climate (change) are not understood



thawing depth evolution



S. Gruber et al. 2004


Our patient does not fit
into a laboratory


So the laboratory has to
go on the mountain

PermaSense

- Consortium of several projects, start in 2006
- Multiple disciplines (geo-science, engineering)
- Fundamental as well as applied research
- More than 20 people, 9 PhD students

- Nano-tera.ch X-Sense is a PermaSense project

nano-tera.ch

NCCR MICS
National Competence Center In Research
Mobile Information and Communication Systems

Bundesamt für Umwelt BAFU
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

FNSNF
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

cces

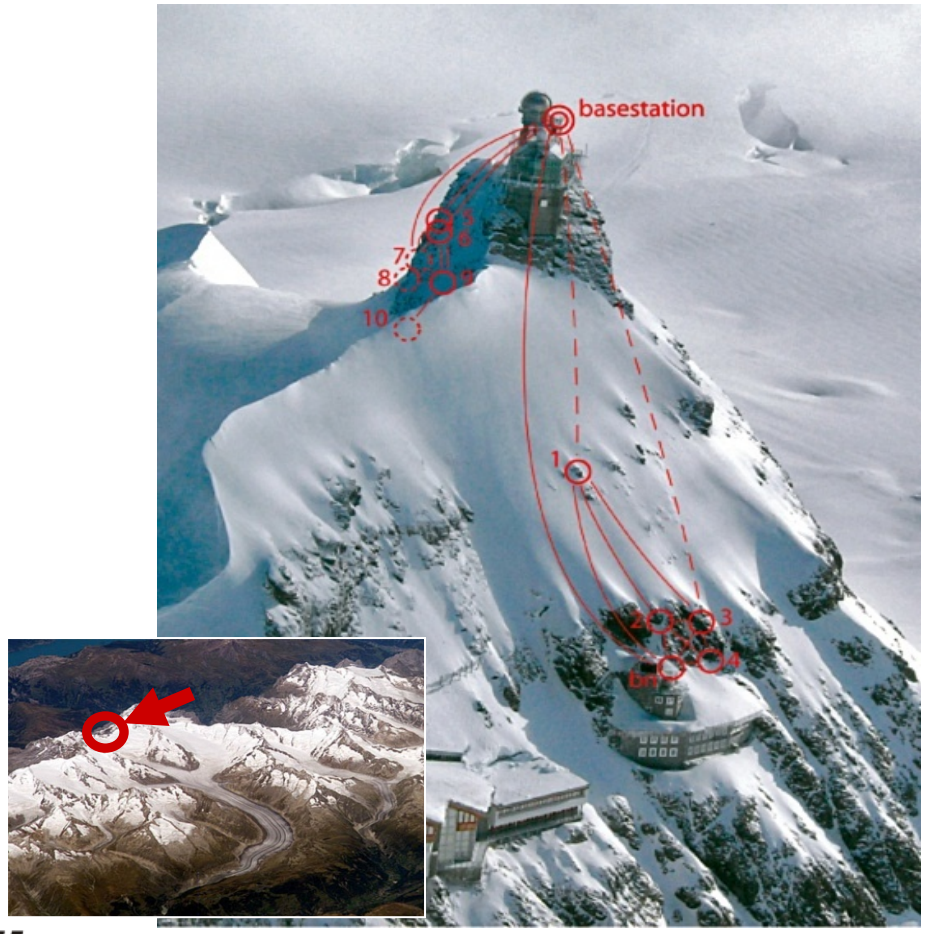
http://www.permasense.ch

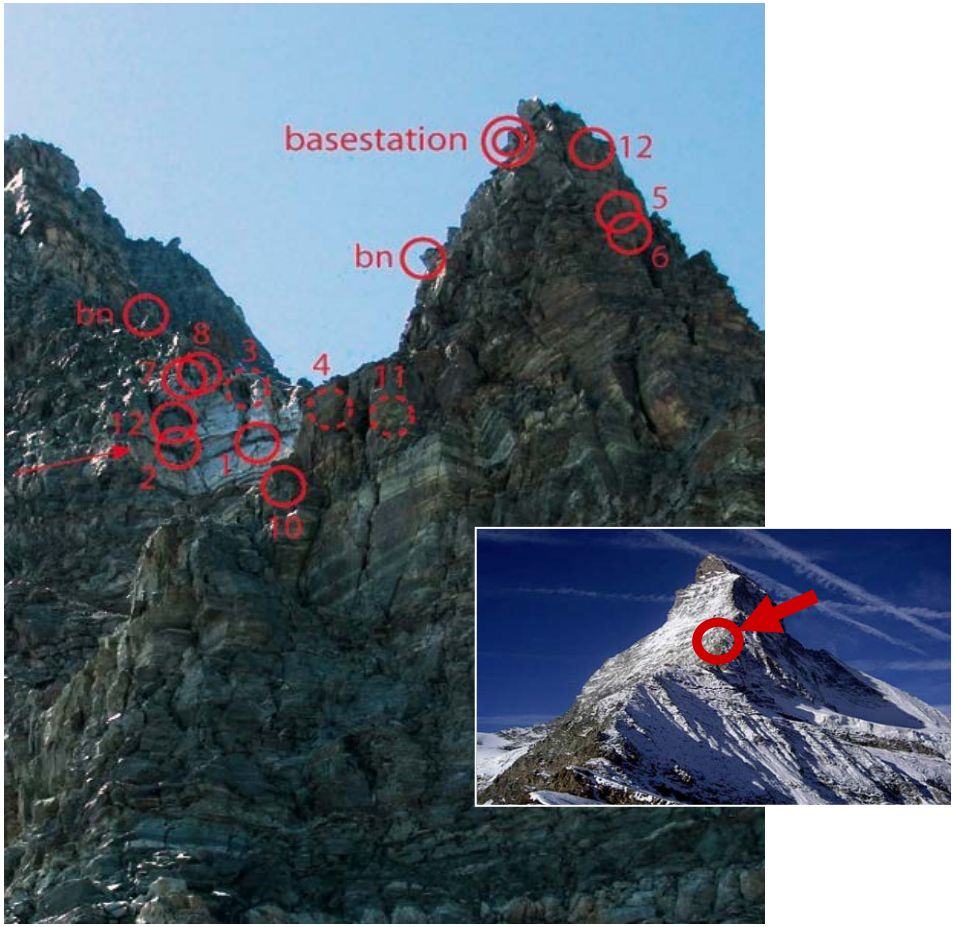
PermaSense – Competence in Outdoor Sensing

- Wireless systems, low-latency data transmission
- Customized sensors
- Ruggedized equipment
- Data management
- Planning, installing, operating (years) large deployments

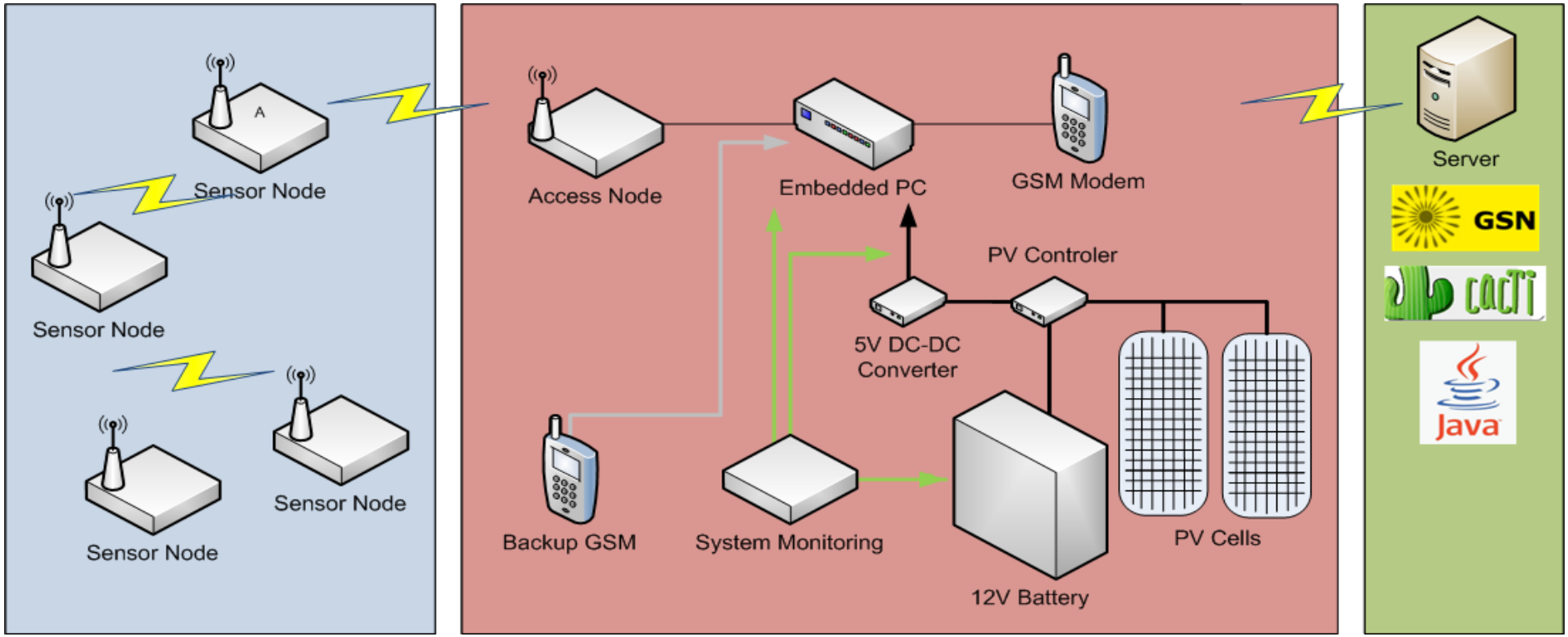
PermaSense Deployment Sites 3500 m a.s.l.

A **scientific instrument** for **precision sensing** and data recovery in environmental extremes



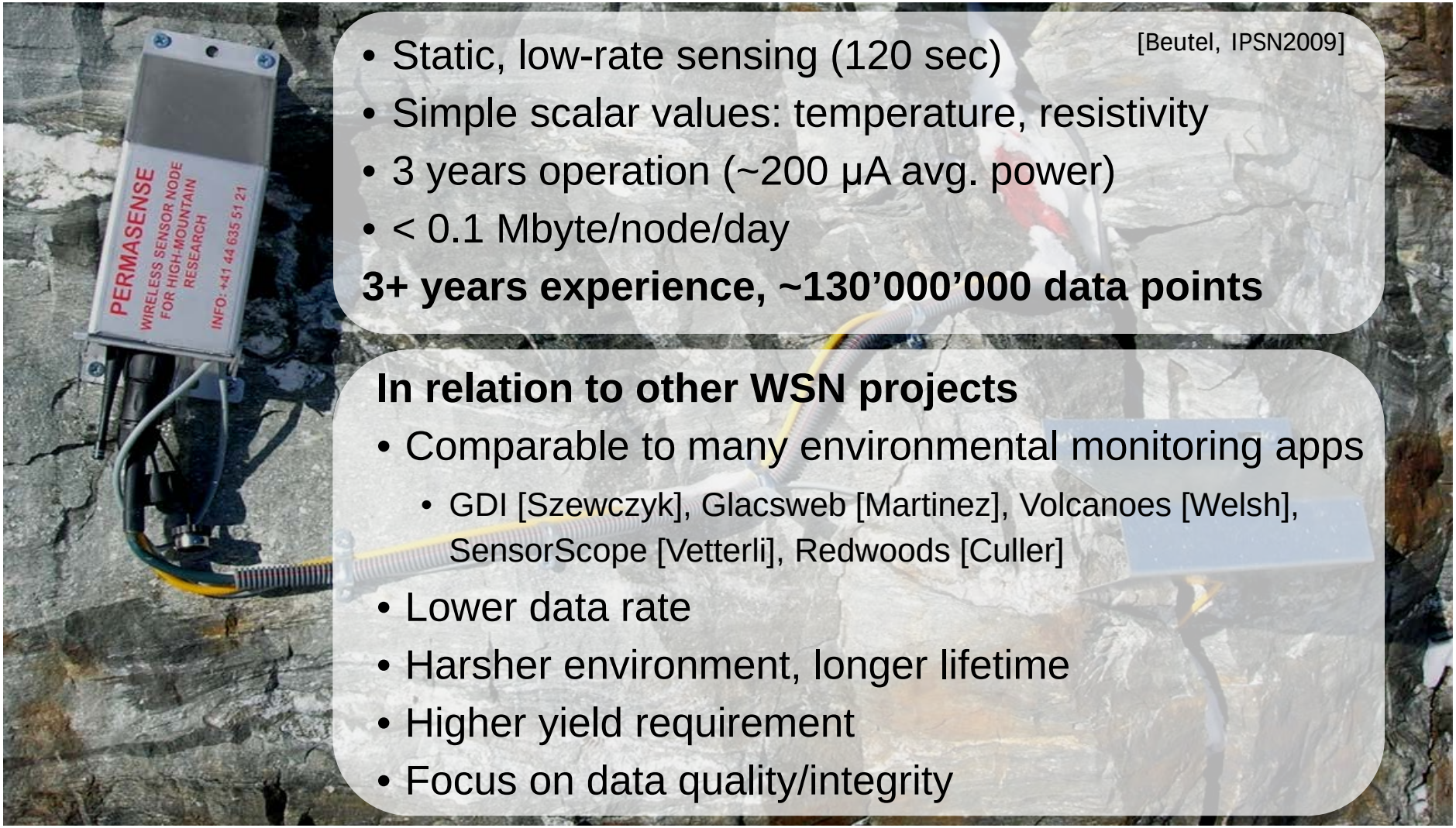


PermaSense – System Architecture



Sensor network **Base station** **Backend**

Simple Low-Power Wireless Sensors



[Beutel, IPSN2009]

- Static, low-rate sensing (120 sec)
- Simple scalar values: temperature, resistivity
- 3 years operation (~200 µA avg. power)
- < 0.1 Mbyte/node/day

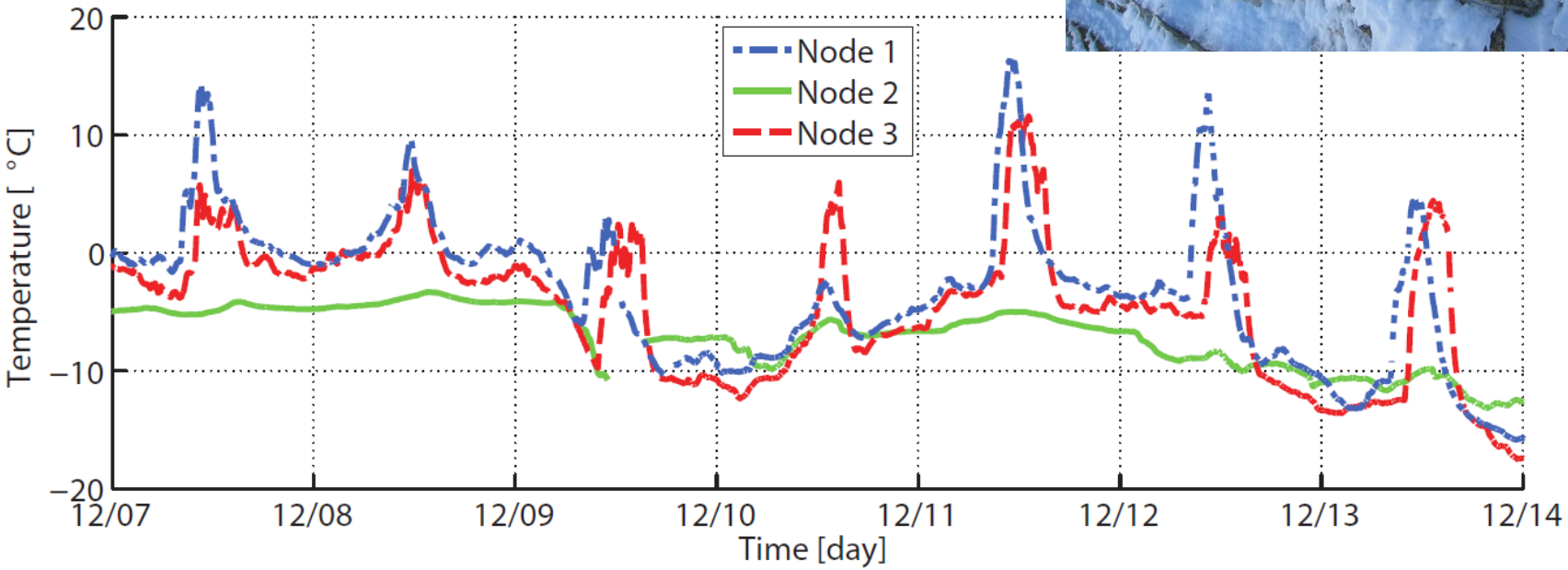
3+ years experience, ~130’000’000 data points

In relation to other WSN projects

- Comparable to many environmental monitoring apps
  - GDI [Szewczyk], Glacsweb [Martinez], Volcanoes [Welsh], SensorScope [Vetterli], Redwoods [Culler]
- Lower data rate
- Harsher environment, longer lifetime
- Higher yield requirement
- Focus on data quality/integrity

Challenge: The Physical Environment

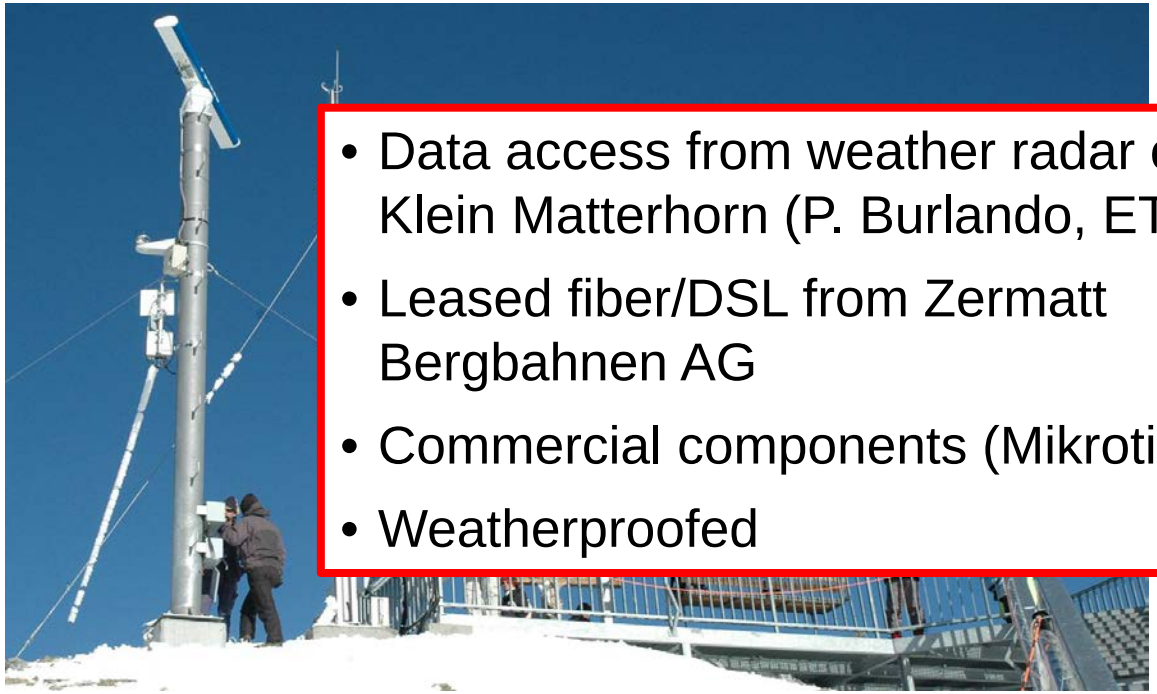
- Lightning, avalanches, rime, prolonged snow/ice cover, rockfall
- Strong daily variation of temperature
  - −30 to +40°C
  - ΔT ≦ 20°C/hour



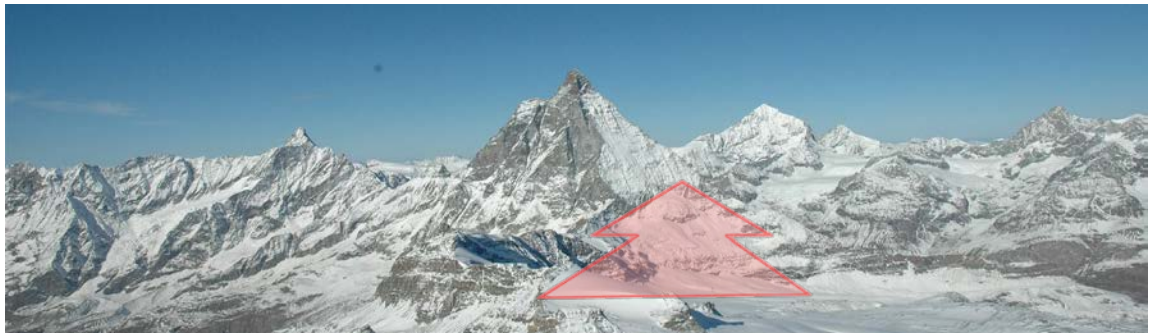
Field Site Support

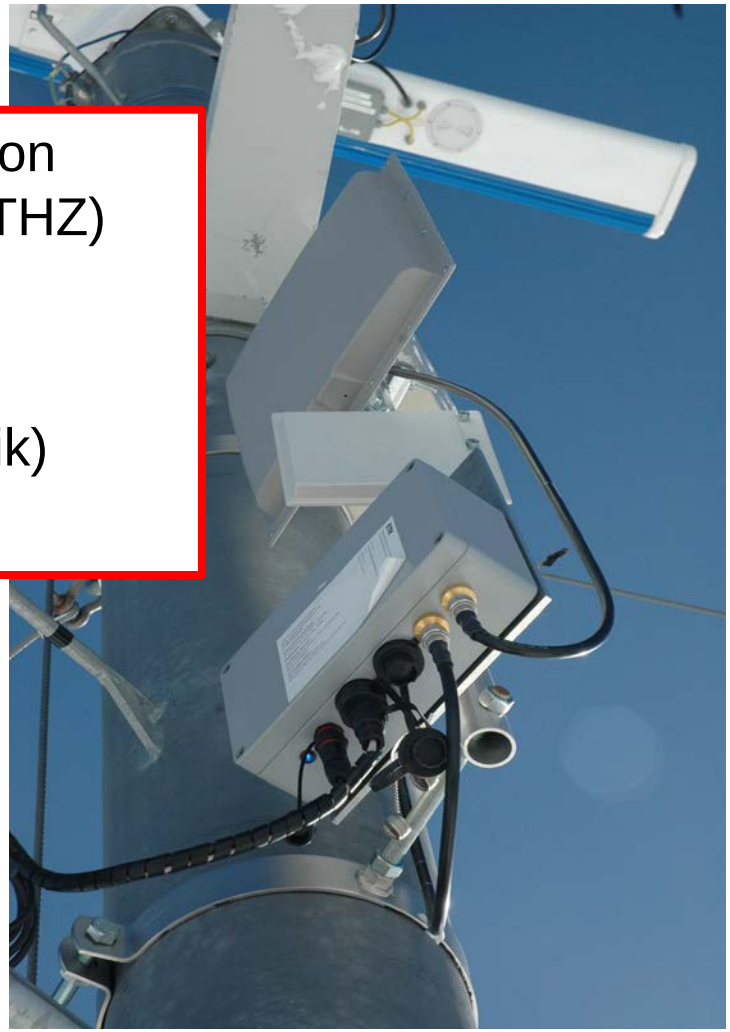
- Base station
  - On-site data aggregation
  - Embedded Linux
  - Solar power system
  - Redundant connectivity
  - Local data buffer
  - Database synchronization
- Cameras
  - PTZ webcam
  - High resolution imaging (D-SLR)
- Weather station
- Remote monitoring and control

WLAN Long-haul Communication



- Data access from weather radar on Klein Matterhorn (P. Burlando, ETHZ)
- Leased fiber/DSL from Zermatt Bergbahnen AG
- Commercial components (Mikrotik)
- Weatherproofed

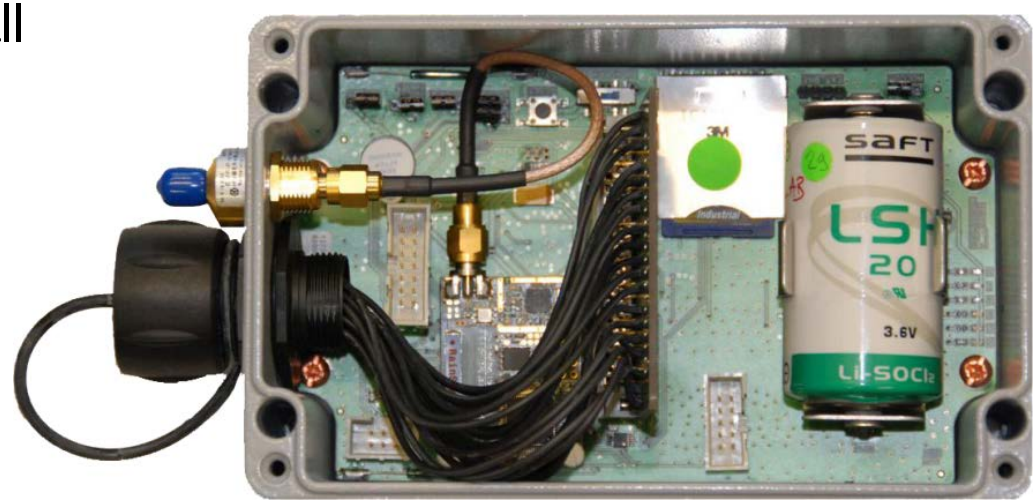


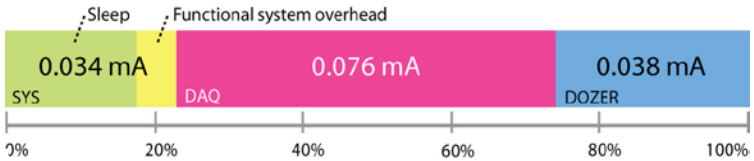


PermaSense – Sensor Node Hardware

[Beutel, IPSN2009]

- Shockfish TinyNode584
  - MSP430, 16-bit, 8MHz, 10k SRAM, 48k Flash
  - LP Radio: XE1205 @ 868 MHz
- Waterproof housing and connectors
- Protective shoe, easy install
- Sensor interface board
  - Interfaces, power control
  - Temp/humidity monitor
  - 1 GB memory
- 3-year life-time
  - Single Li-$SOCl_2$ battery, 13 Ah
  - ~300 µA power budget

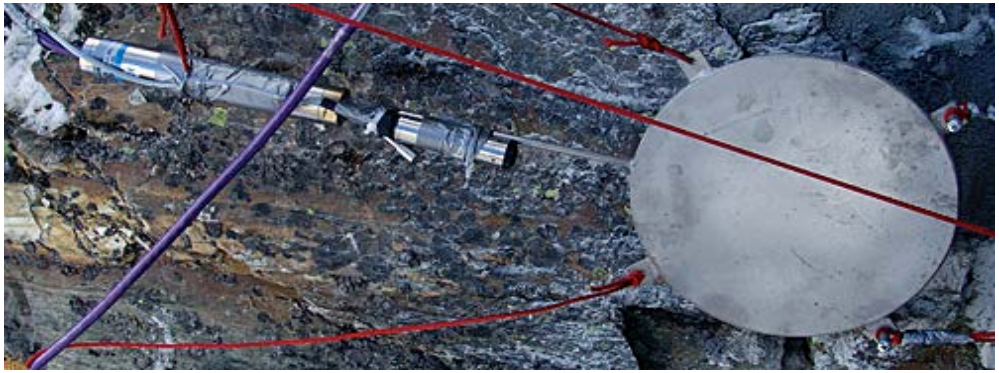




measured avg. current consumption ~148 µA

PermaSense – Sensor Types

- Sensor rods (profiles of temperature and electric conductivity)
- Thermistor chains
- Crack meters
- Water pressure
- Ice stress
- Self potential

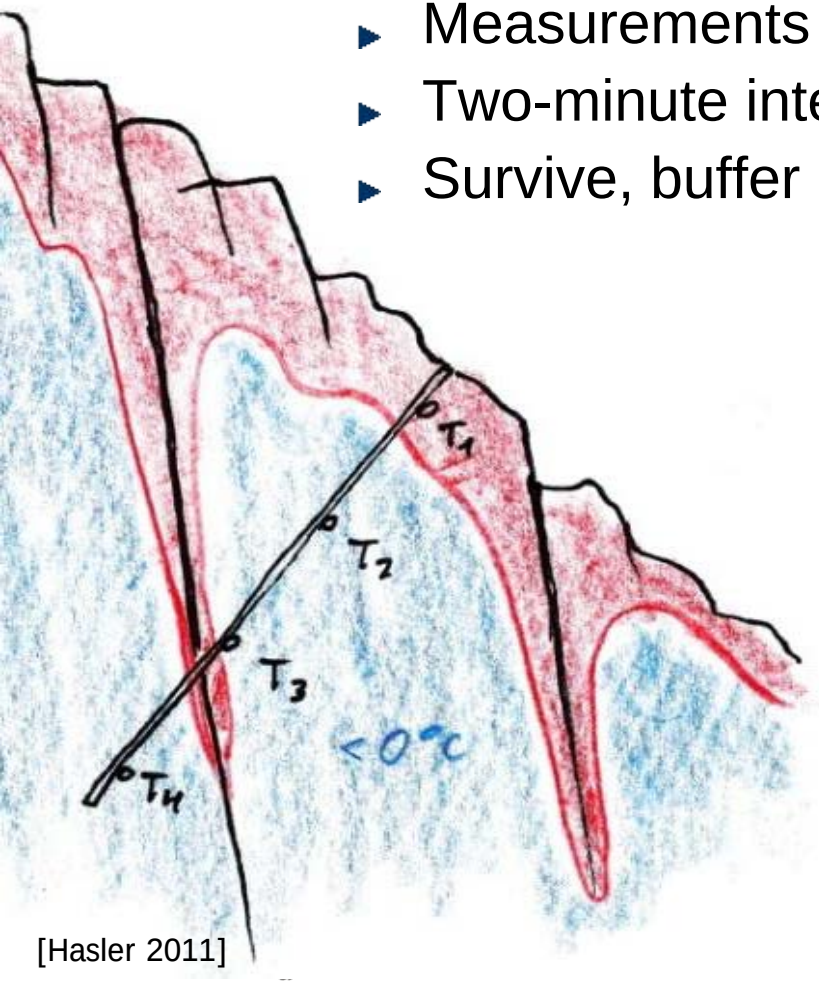


- Data: Simple sensors, constant rate sampling, scalar values

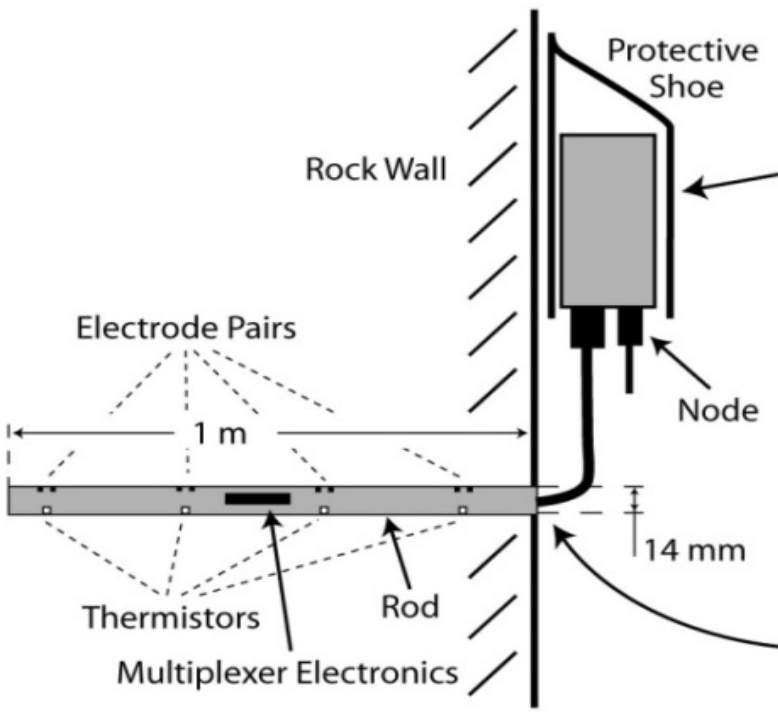
Established: Rock/ice Temperature

Aim: Understand temperatures in heterogeneous rock and ice

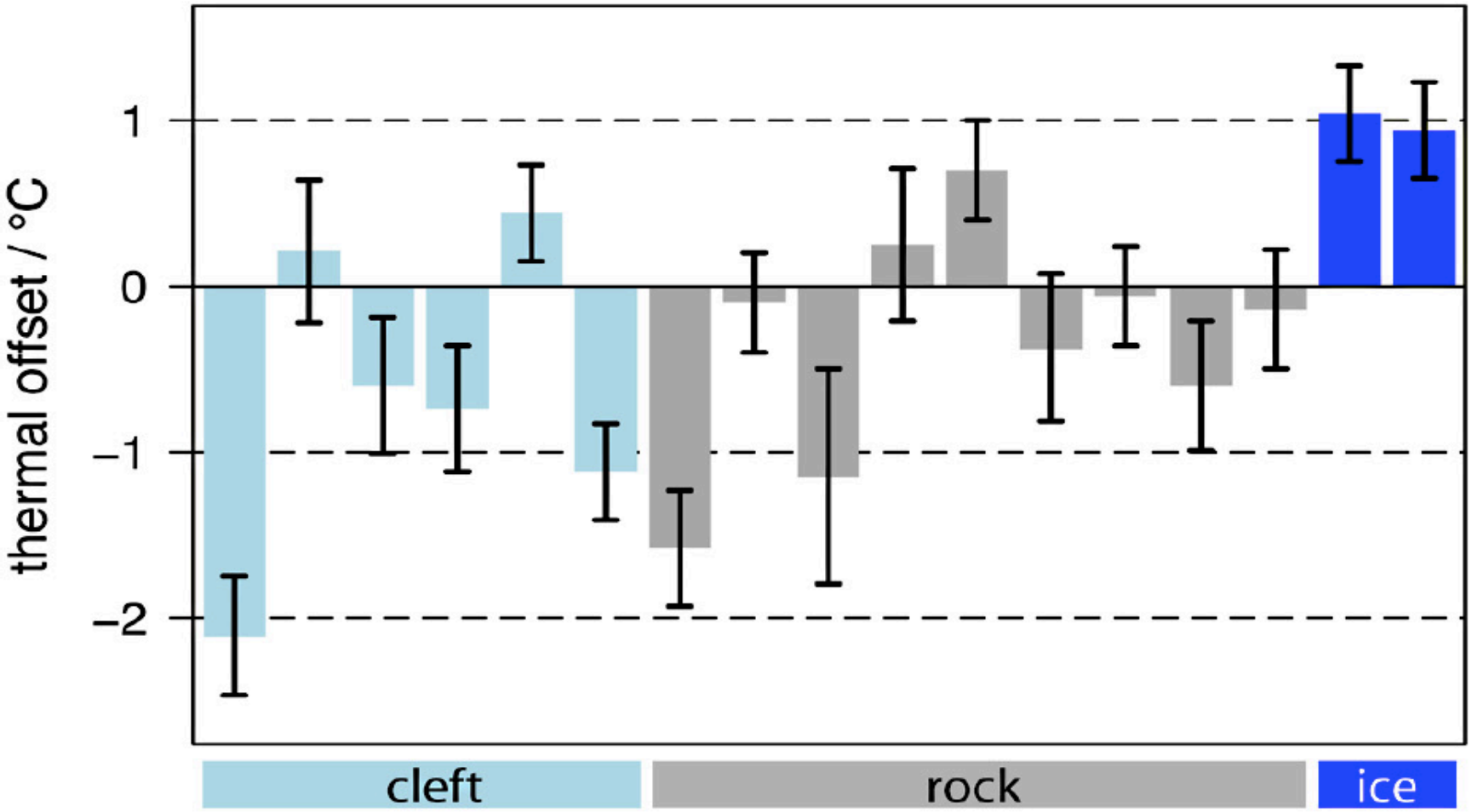
- Measurements at several depths
- Two-minute interval, autonomous for several years
- Survive, buffer and flush periods without connectivity



[Hasler 2011]



Results: Rock/ice Temperatures



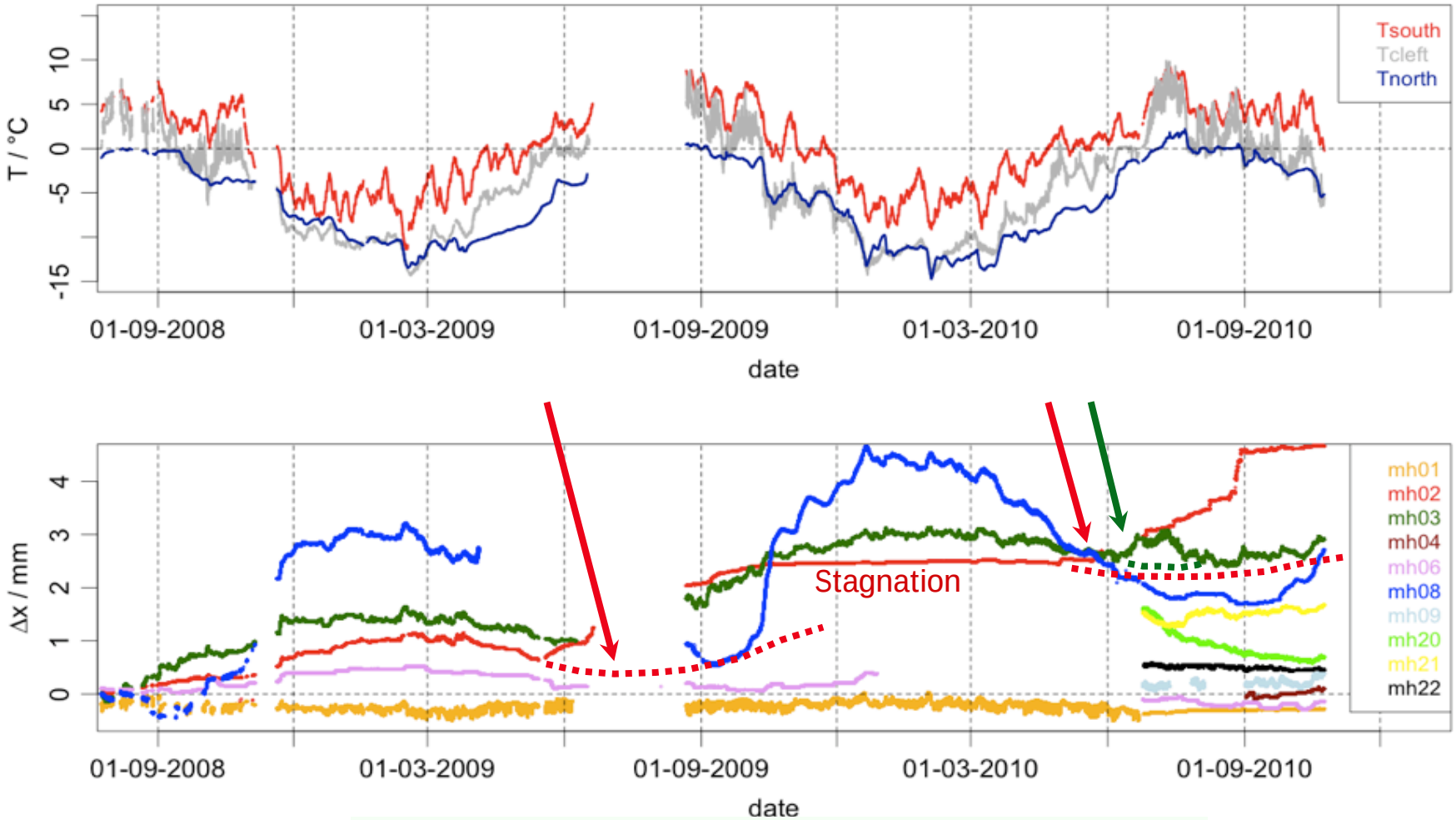
[Hasler, A., Gruber, S. & Haeberli, W. (in review) Temperature variability and thermal offset in steep alpine rock and ice faces, *The Cryosphere]*

Established: Crack Dilatation

Aim: To understand temperature/ice-conditioned rock kinematics

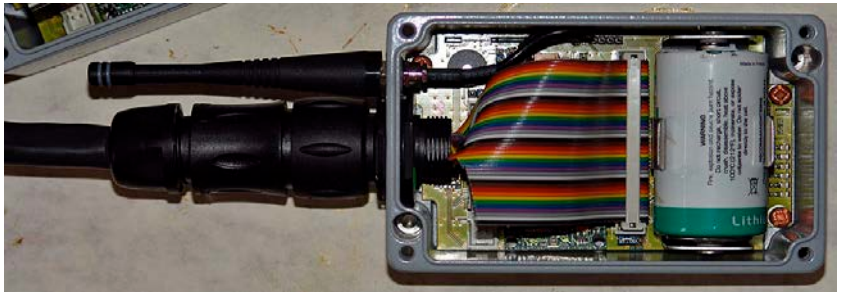
- Temperature-compensated, commercial instrument
- Auxiliary measurements (temperature, additional axes,…)
- Two-minute interval, autonomous for several years
- Protection against snow-load and rock fall

Results: Rock Kinematics

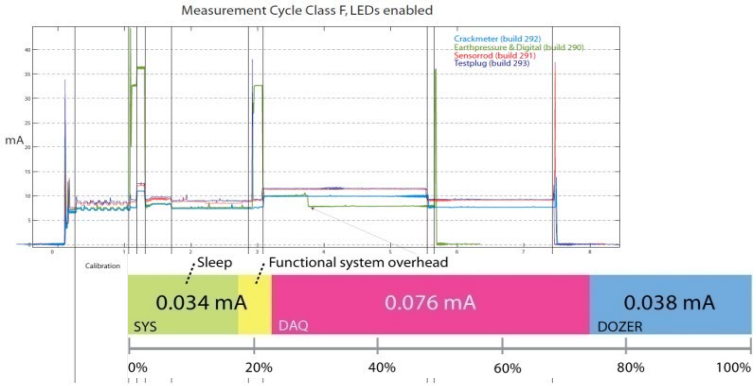


[Hasler, A., Gruber, S. & Beutel, J. (in revision) Kinematics of steep bedrock permafrost, *Journal of Geophysical Research*]

Key PermaSense Challenges

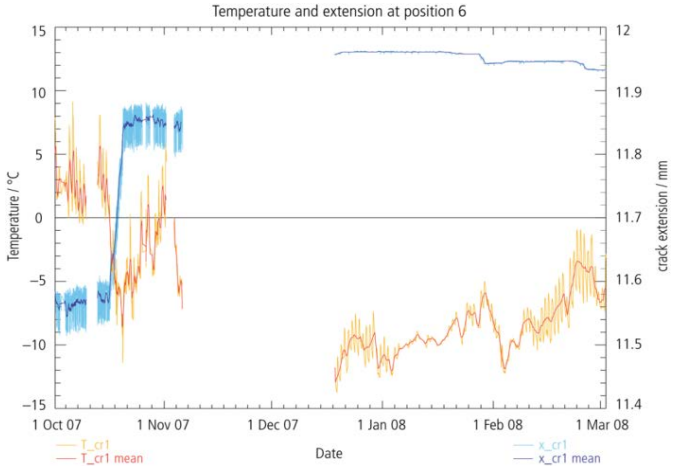


System Integration



Correct Test and Validation

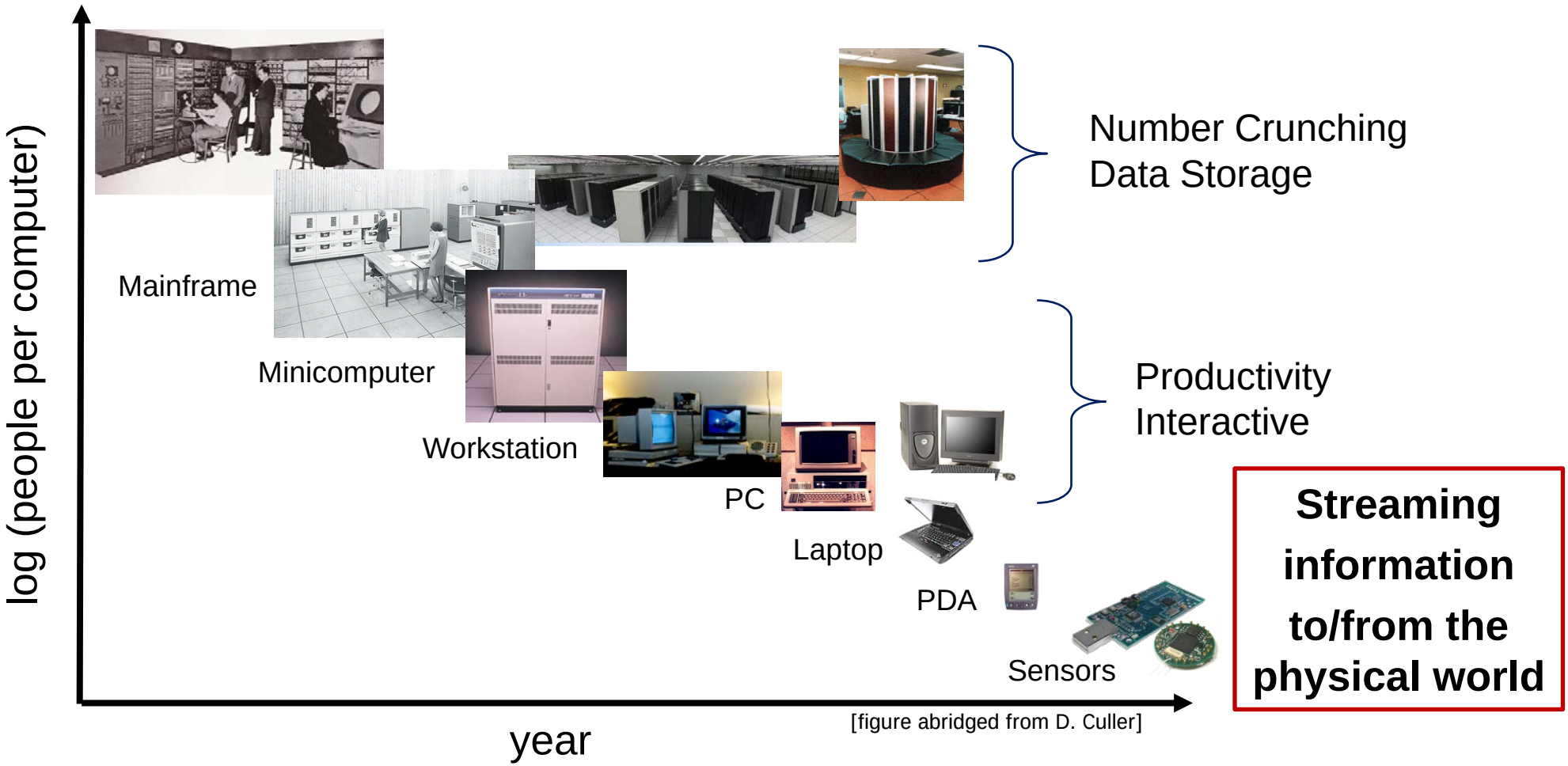
Actual Data



Interdisciplinary Team

System Design – Low-Power Architectures – Wireless Protocols

Bell's Law: New Class of Computers Every 10Y



[figure abridged from D. Culler]

Highly Resource Constrained

Distributed State

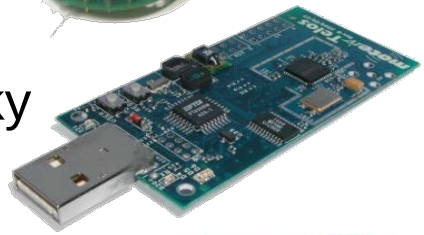
Wireless Networked Embedded Systems

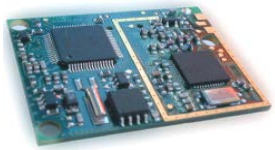
Unreliable Communication

Interaction and Tight Embedding in Environment

Basic Concepts of WSN Platforms

- “Mote class” devices
  - Microcontroller + low-power radio
  - Battery powered
  - Many custom applications
  - Large design space, many variants
  - Most prominent examples: Mica2, Mica2Dot, Tmote Sky



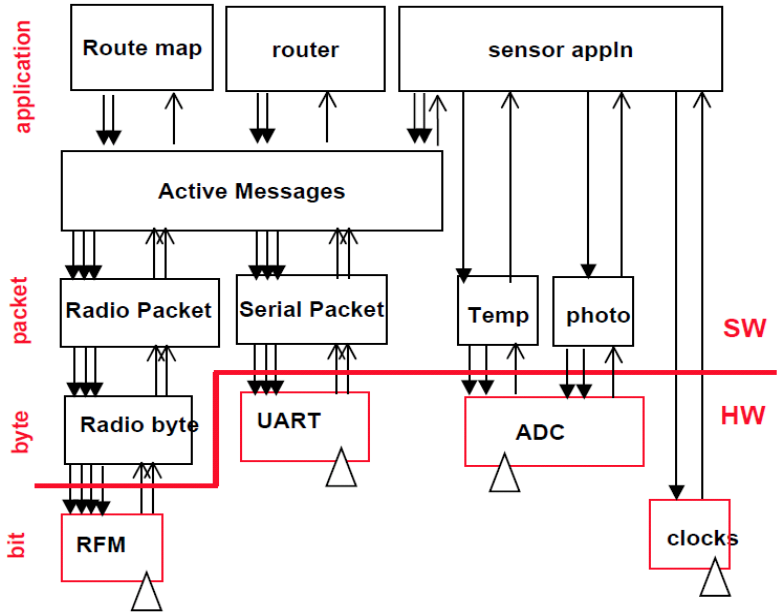


- Hardware is packaged with
  - System software and apps
  - Base stations, network access
  - Server-side solutions (backends)
  - Tools (e.g. simulators, virtual machines …)

- First anticipation: Small = cheap = low complexity

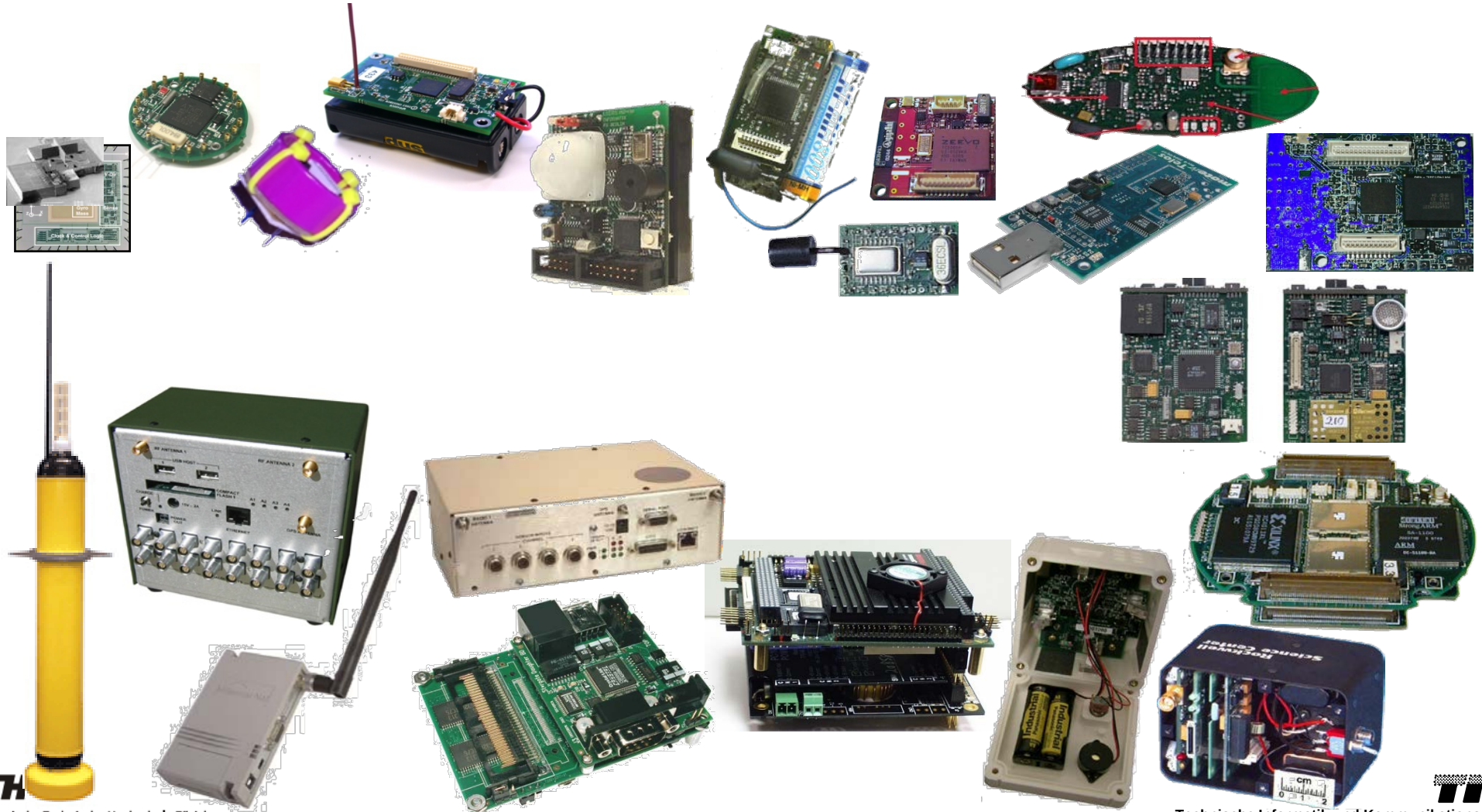
A Popular Software System – TinyOS

- Event driven "Operating System"
  - Written in nesC, a C dialect
  - Geared towards simple applications (max. 10k RAM, 40 k ROM)
  - Basically just a collection of drivers and an event queuing system
  - Event handlers must not block…
- Compositional nature
  - Hard- and software components
  - Interfaces
  - Modularity
- De-facto standard in WSN applications
  - Popular in academia and industry



- Many (comparable) systems exist: Contiki, SOS, Mantis, BTnut…

WSN Platform Variants



WSN Design and Development Peculiarities

- Interaction with the environment
  - Cannot stop time and use breakpoints
- Visibility of what is going on
  - Getting information in and out of the network
- Low-power modes
- Disconnected medium
  - Wireless is unreliable
  - Split phase operation, interaction across the ensemble
- Lack of comprehensive models/analysis
  - Often there is lot's of data but no clue what to do with it

Low-power wireless: Where is the bar?
110 uA is baseline, 20 uA is research goal
99.5-9% reliability over multiple 70% links
[D. Culler, SenSys 2007]

PermaSense Wireless technology for a laboratory on the mountain

System Design – Architectural Considerations

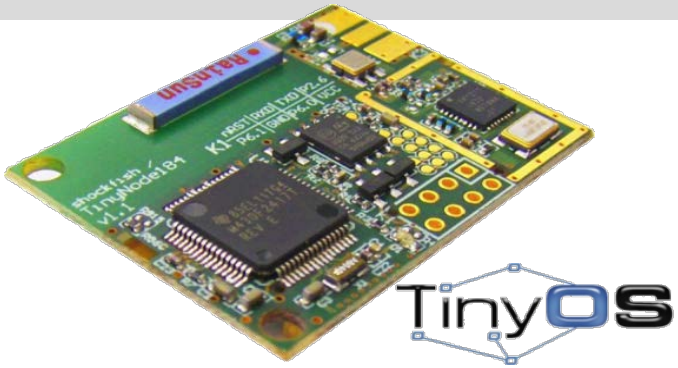
- Requirements
  - Capabilities for multiple sensors
  - Extreme low-power
  - High quality data acquisition (ADC resolution on MSP430 not sufficient)
  - Reliability in extreme environment

- One platform vs. family of devices? Make vs. buy? Existing components?

- Decisions
  - Modular, accommodating different sensors on one platform
  - Single, switchable serial bus architecture
  - Strict separation of operating phases (TDMA)
  - Enough storage for disconnected operation over months
  - Temp/Humidity/System voltage supervision in every box

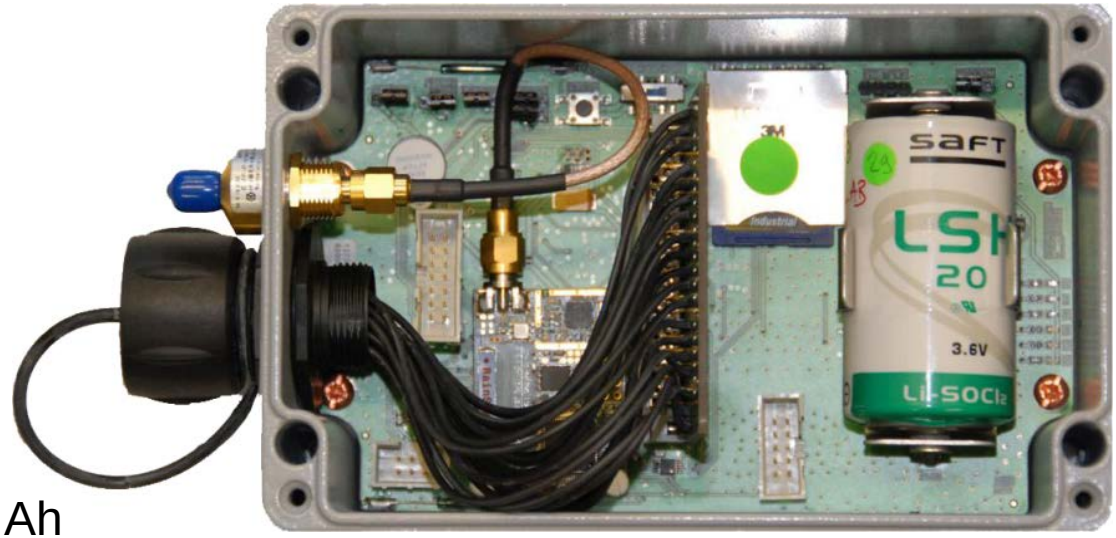
Sensor Node Hardware

- Shockfish TinyNode584
  - MSP430, 16-bit, 8MHz, 10k SRAM, 48k Flash
  - LP Radio: XE1205 @ 868 MHz



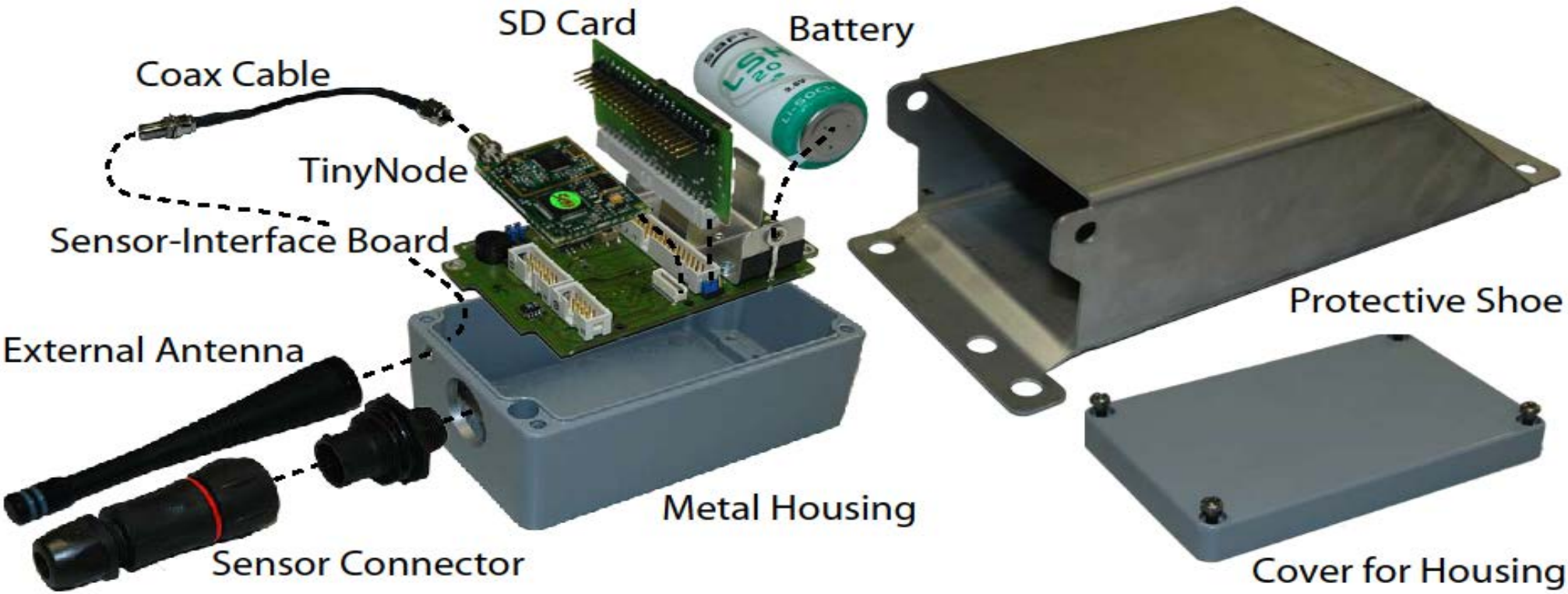
- Waterproof housing and connectors

- Sensor interface board
  - Interfaces, power control
  - Temp/humidity monitor
  - 1 GB Flash memory

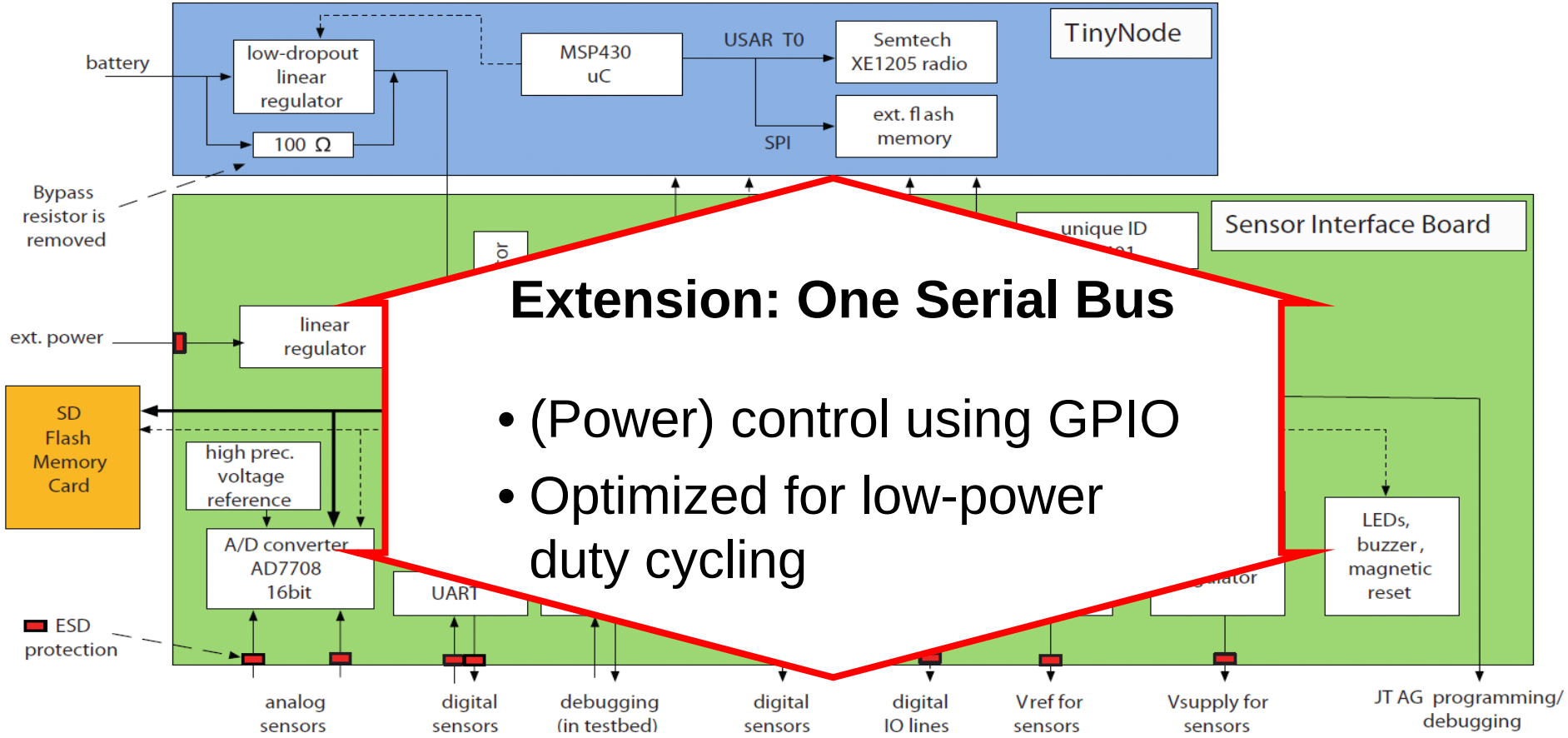


- 3-year life-time
  - Single Li-$SOCl_2$ battery, 13 Ah
  - ~300 μA power budget

Ruggedized for Alpine Extremes



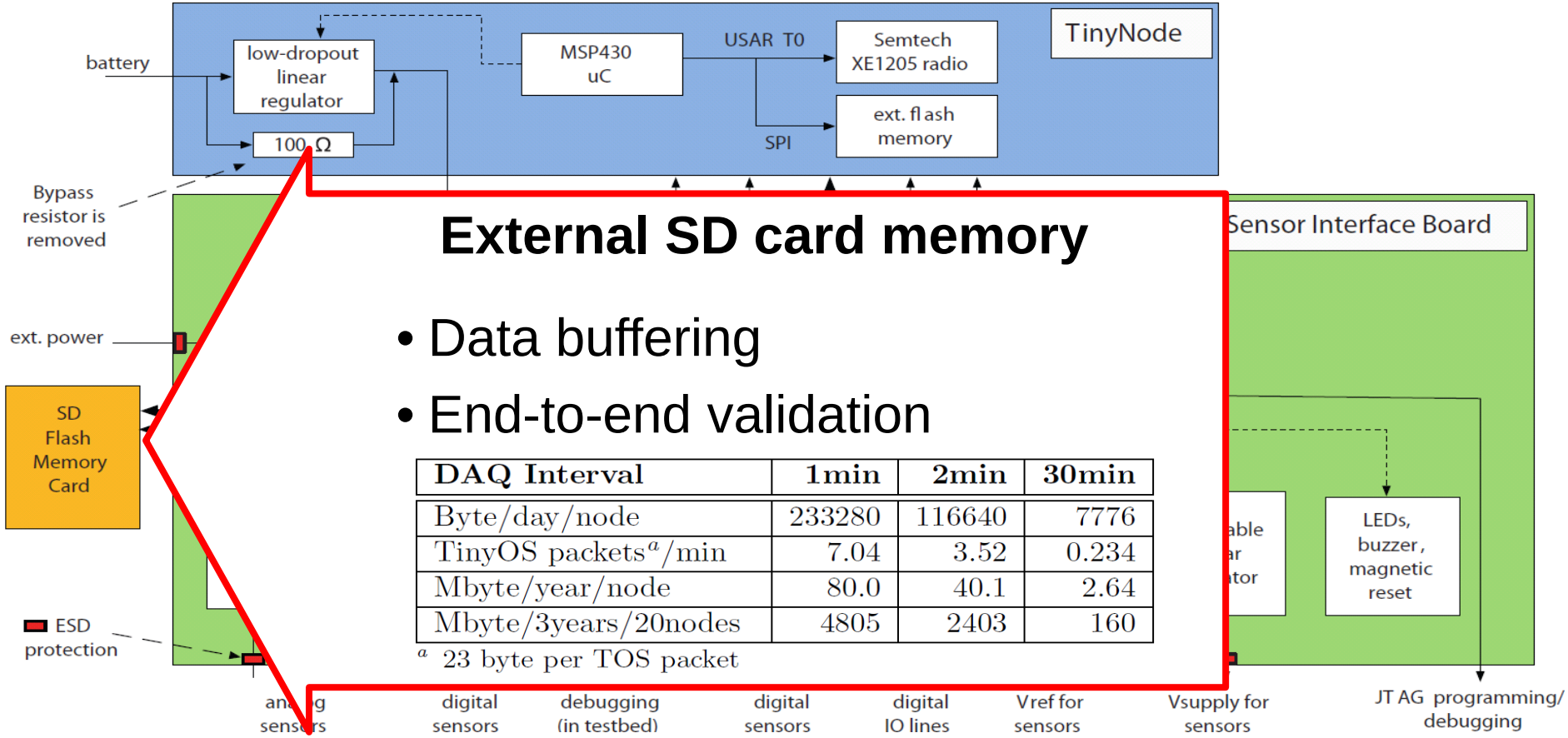
TinyNode + Sensor Interface Board



Extension: One Serial Bus

- (Power) control using GPIO
- Optimized for low-power duty cycling

External Storage Extension



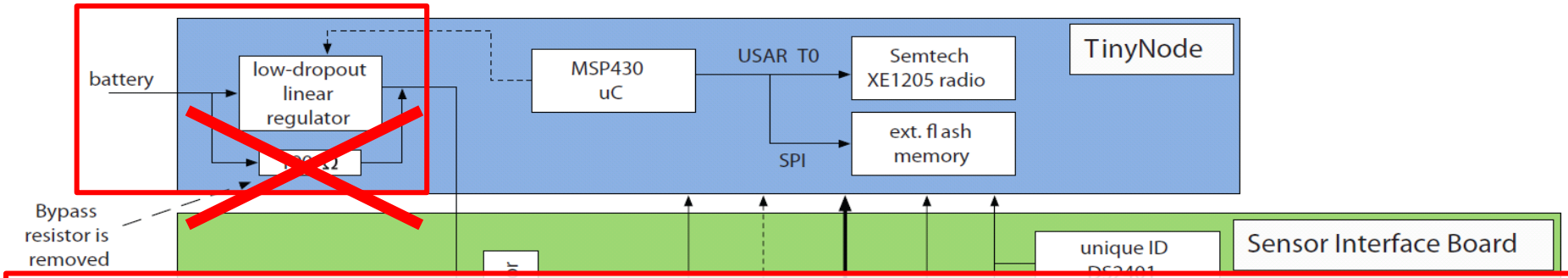
External SD card memory

- Data buffering
- End-to-end validation

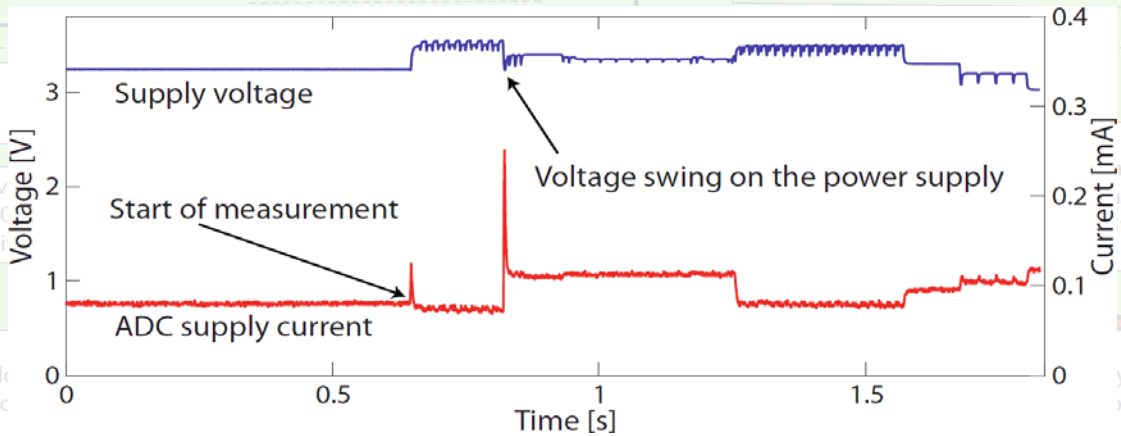
DAQ Interval	1min	2min	30min
Byte/day/node	233280	116640	7776
TinyOS packets[a]/min	7.04	3.52	0.234
Mbyte/year/node	80.0	40.1	2.64
Mbyte/3years/20nodes	4805	2403	160

[a] 23 byte per TOS packet

Power Optimization – Squeeze with Implications

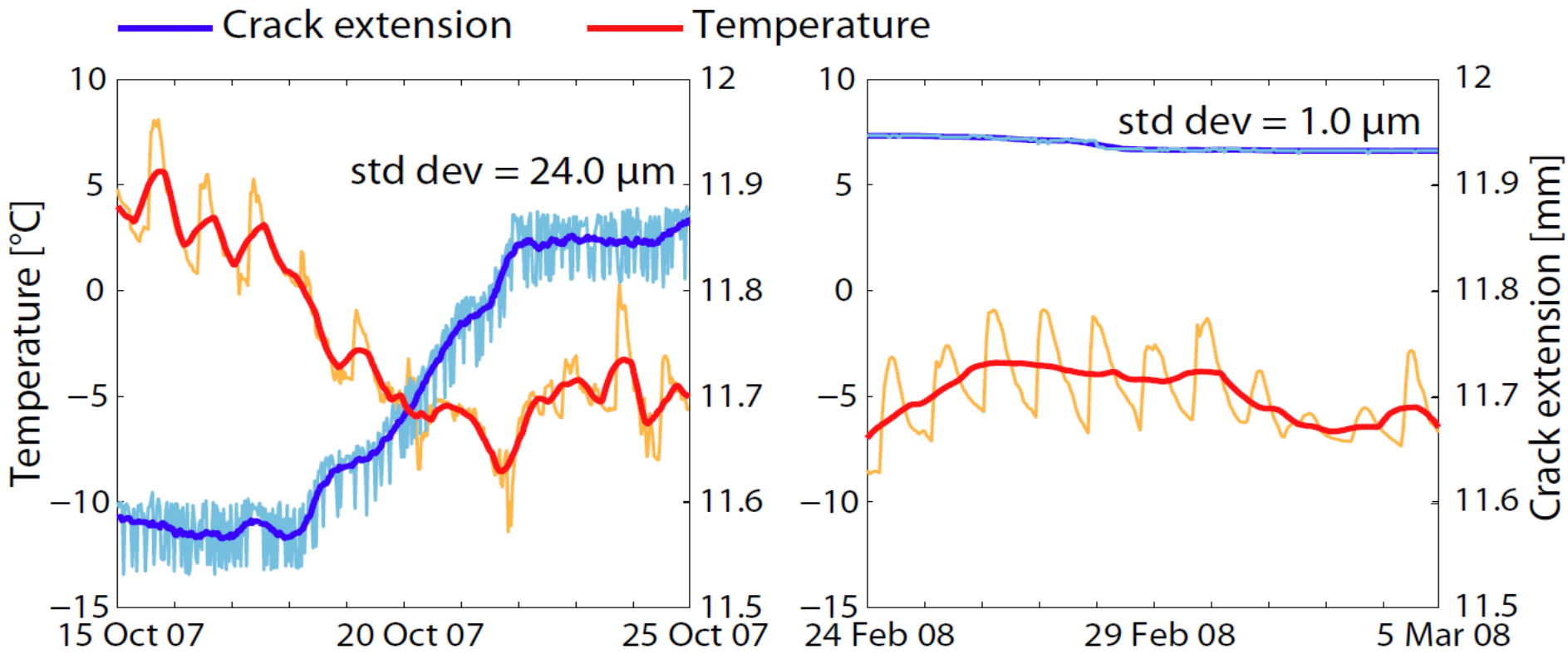


- Regulator uses 17uA quiescent current
- Bypass used to shutdown regulator -> ~1uA in standby



- No Bypass increases ADC accuracy: stddev 0.8844 -> 0.0706

Power Quality Increases Data Accuracy

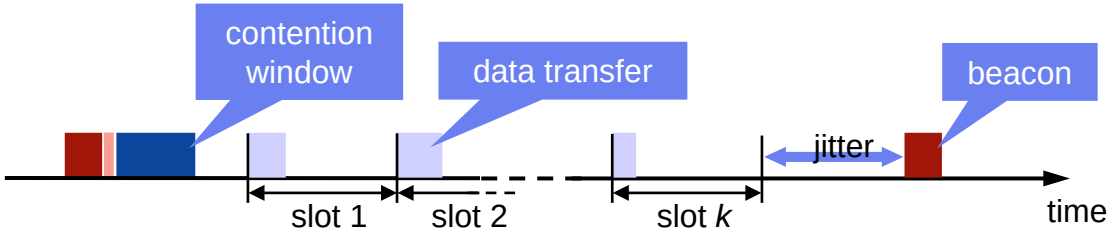


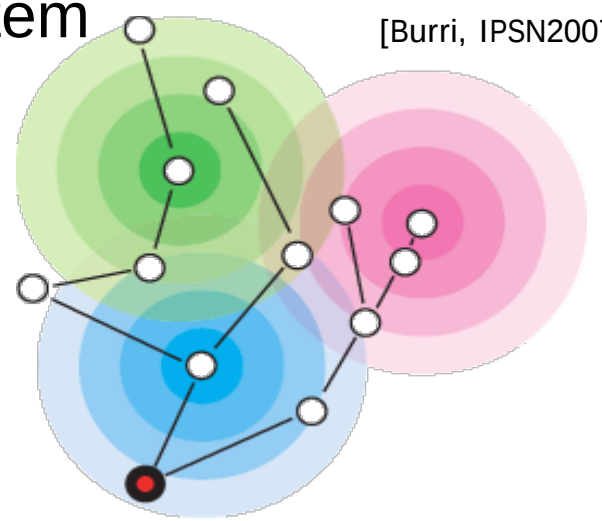
Before **After**

Ultra Low-Power Multi-hop Networking

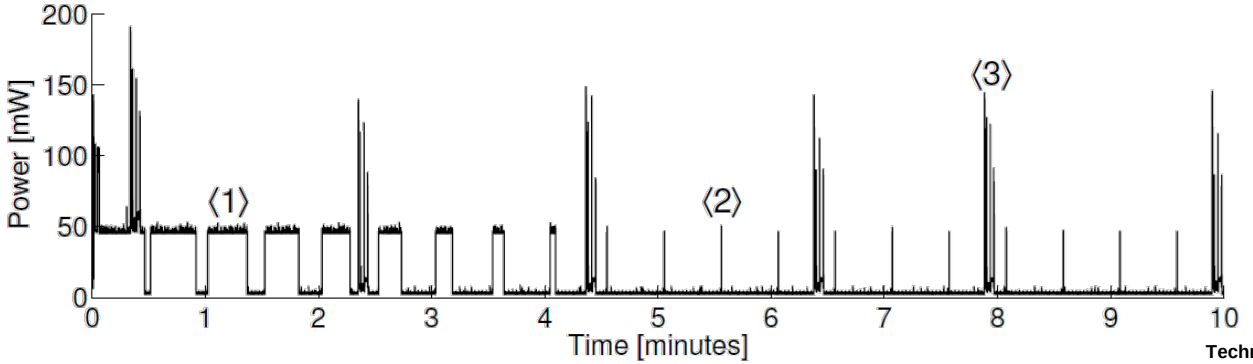
- Dozer ultra low-power data gathering system
  - Beacon based, 1-hop synchronized TDMA
  - Optimized for ultra-low duty cycles
  - **0.167%** duty-cycle, **0.032mA** (@ 30sec beacons)

[Burri, IPSN2007]



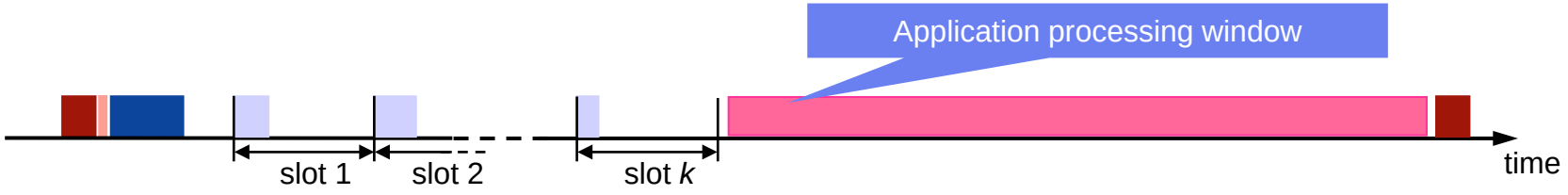


- But in reality: Connectivity can not be guaranteed…
  - Situation dependent transient links (scans/re-connects use energy)
  - Account for long-term loss of connectivity (snow!)

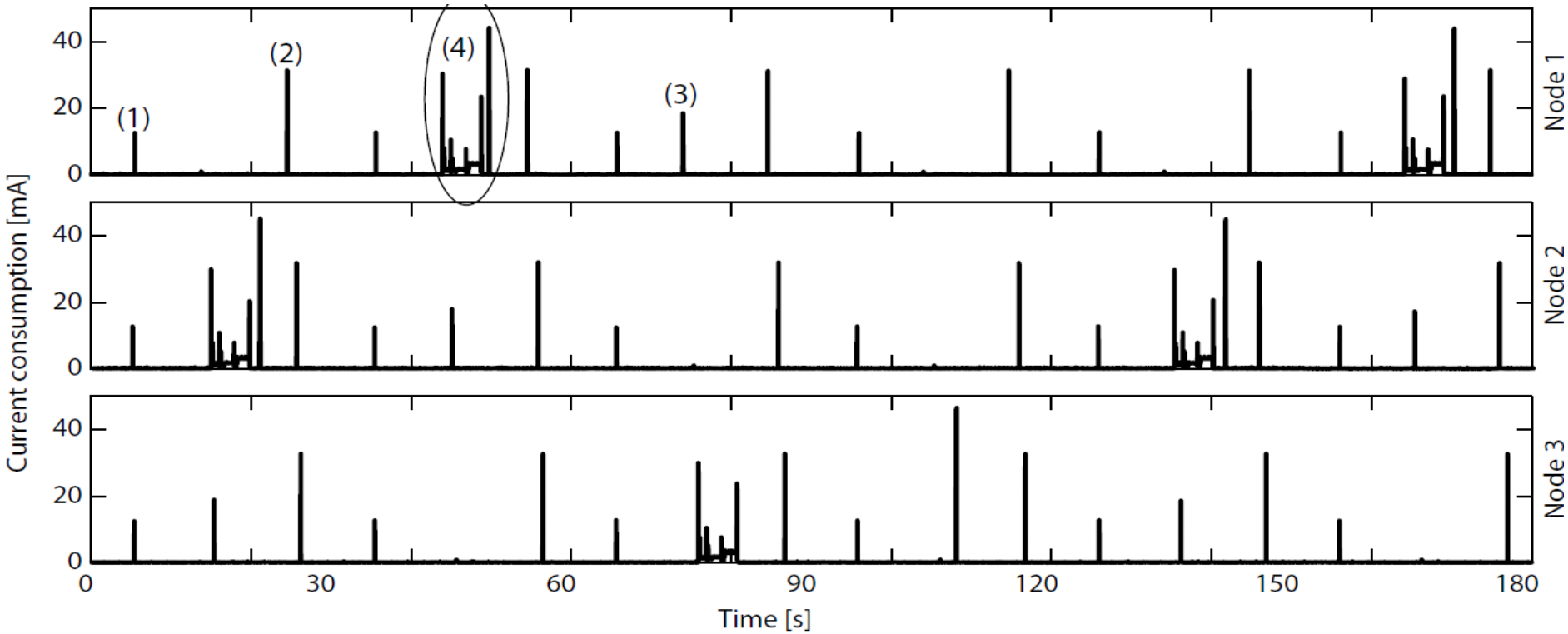


TDMA System Integration

- System-level, round-robin scheduling
  - “Application processing window” between data transfers and beacons
- Local temperature clock drift compensation
- Separation of communication and application processing
  - Custom DAQ/storage routine per application/sensor
  - Less interference between software components
  - Strict periodicity
  - Timing guarantees



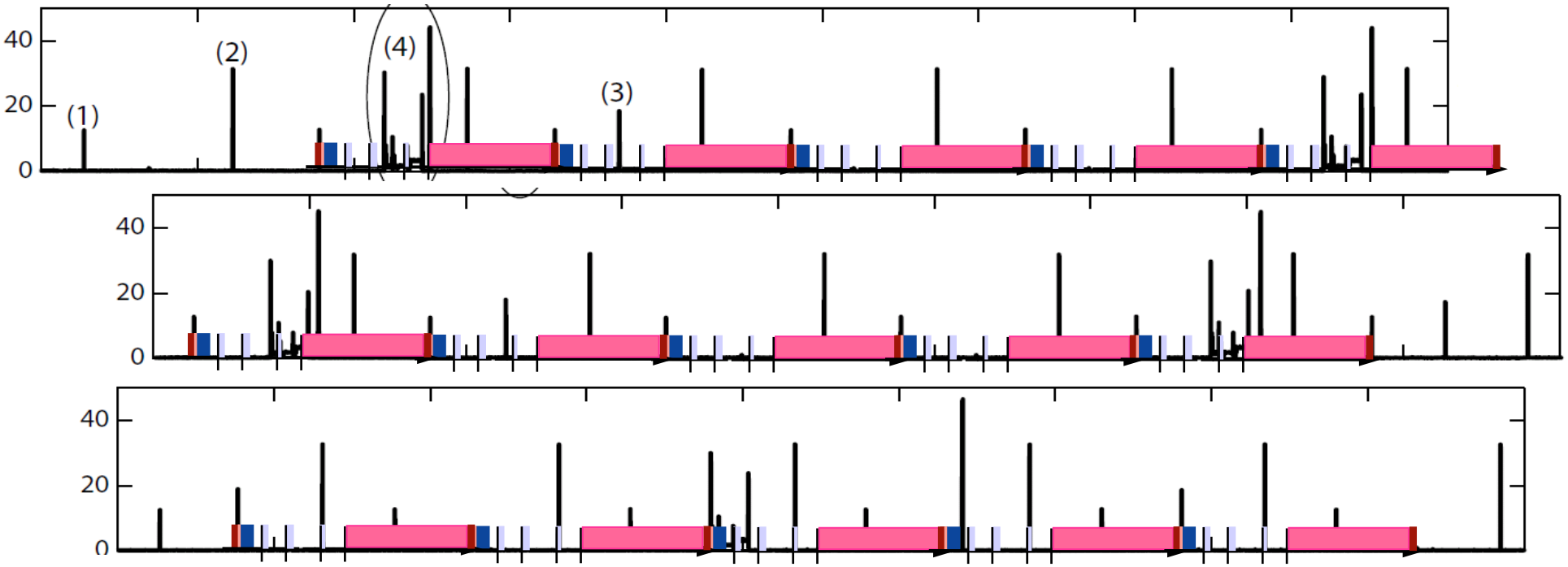
1-hop Validation – Simultaneous Power Traces



(1) Receive parent beacon (2) Send beacon to children (3) Send data to parent (4) Sampling sensor (DAQ)

Multihop Concurrency Causes Timing Violations

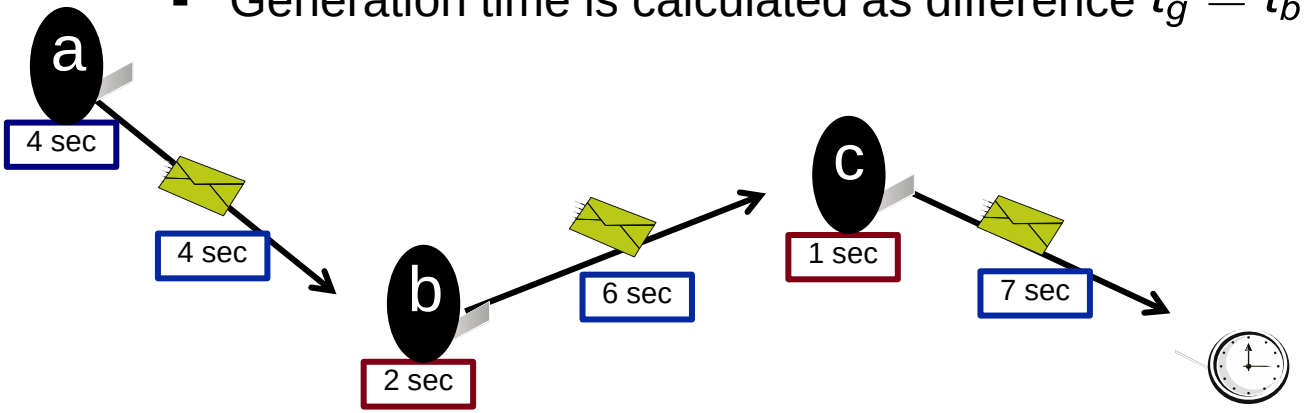
- In reality due to phase shift timing violations and interference occur
  - Communication arbitration over multiple hops
  - Priority is given to communication
  - Slight variations in periodicity



Global Time Stamping

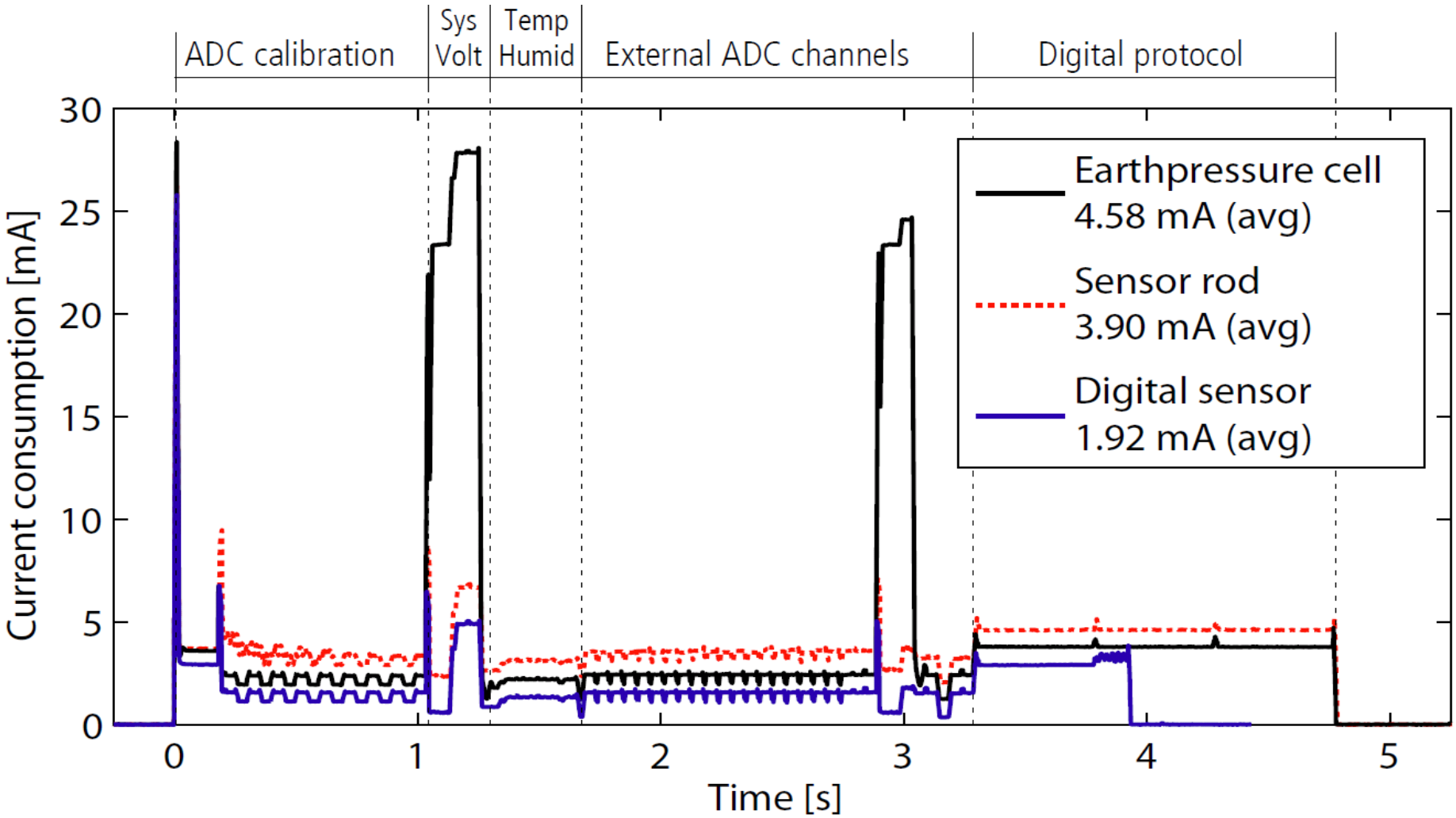
- No network-wide time synchronization available
  - Implications on data usage

- Elapsed time on arrival
  - Sensor nodes measure/accumulate packet sojourn time
  - Base station annotates packets with UTC timestamps
  - Generation time is calculated as difference $\tilde{t}_g = t_b - \tilde{t}_s$



2011/04/14 10:03:31 – 7 sec
= 2011/04/14 10:03:24

Sensors Contribute to Power Consumption

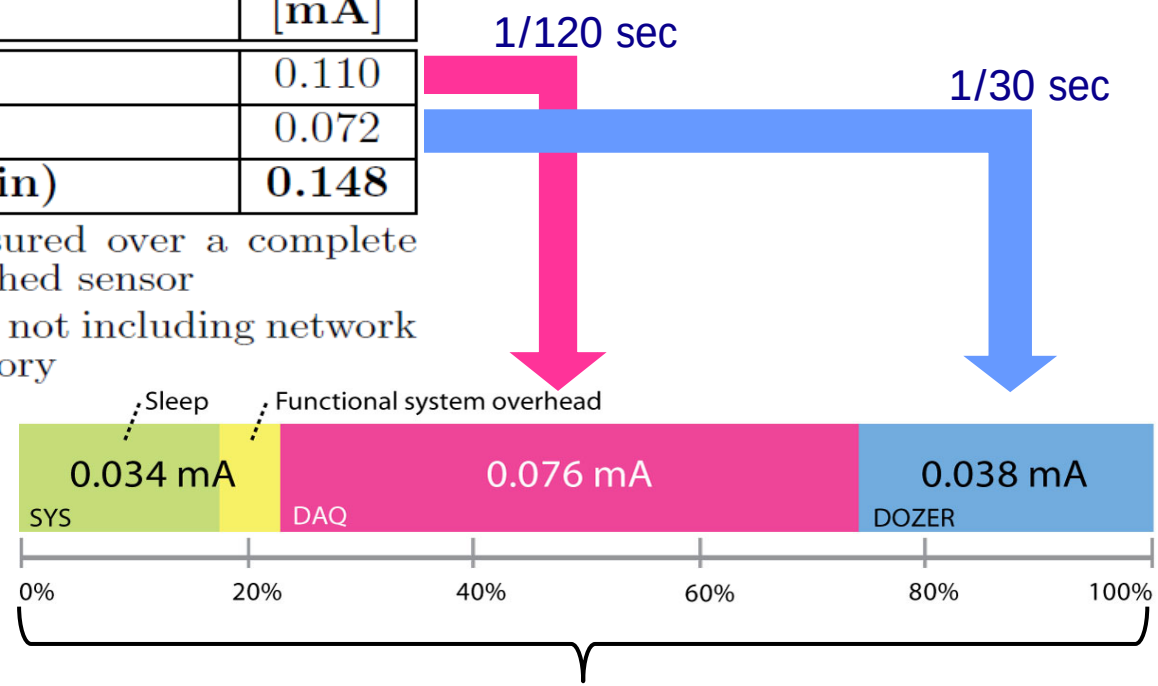


Total Power Performance Analysis

Operating Mode Characterization	[mA]
Sleep	0.026
DAQ active[a]	2.086
Dozer RX idle	13.64
Dozer RX	14.2
Dozer TX	54.6
Measured Average Values	**[mA]**
DAQ only (2min)	0.110
Dozer only (30sec/2min)[b]	0.072
PermaDozer total (30sec/2min)	**0.148**

[a] Averages power consumption measured over a complete DAQ routine execution without attached sensor

[b] Dozer only includes communication, not including network initialization and access to flash memory



148 uA average power

Test + Debugging Methodology – Examples

WSN Design and Development

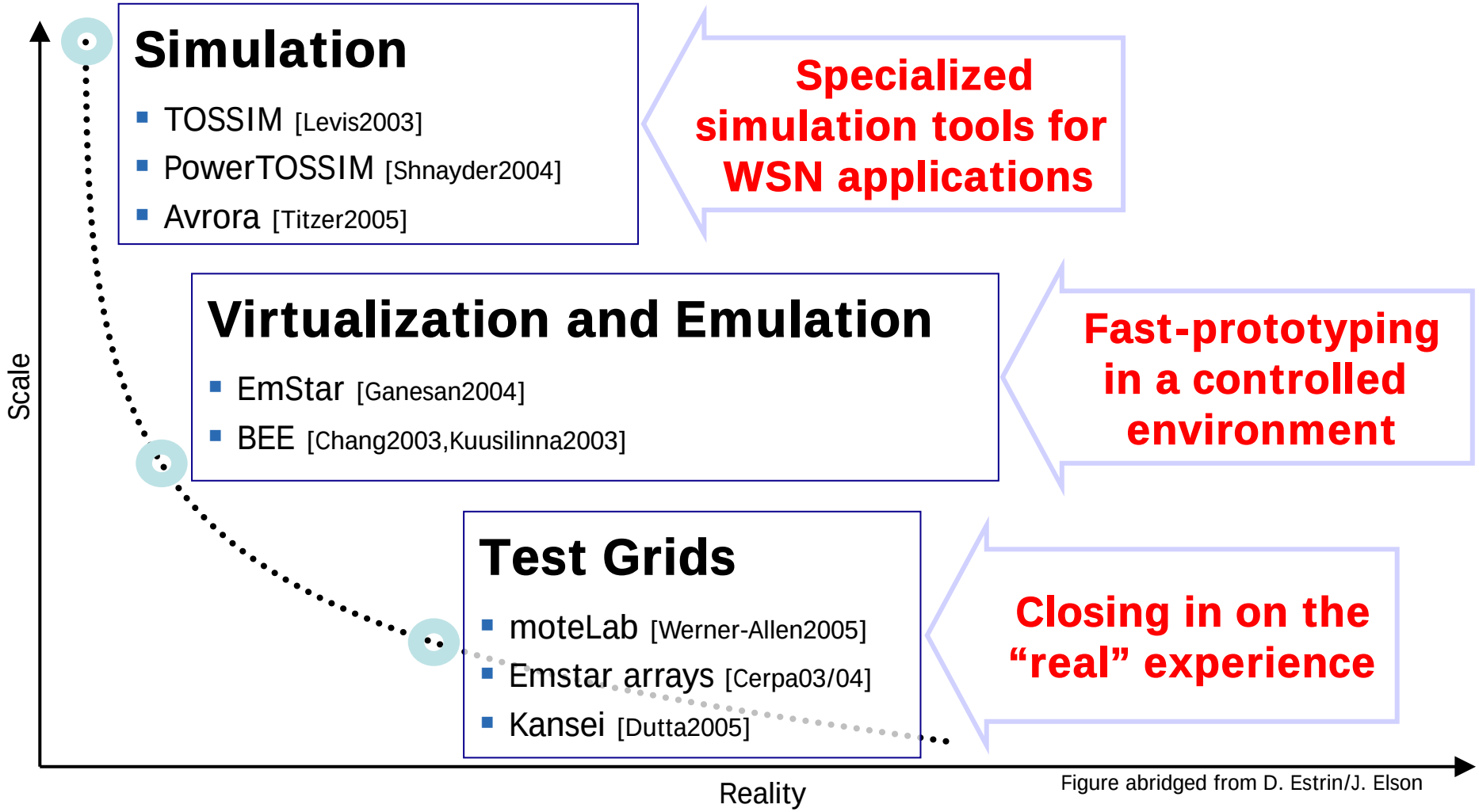
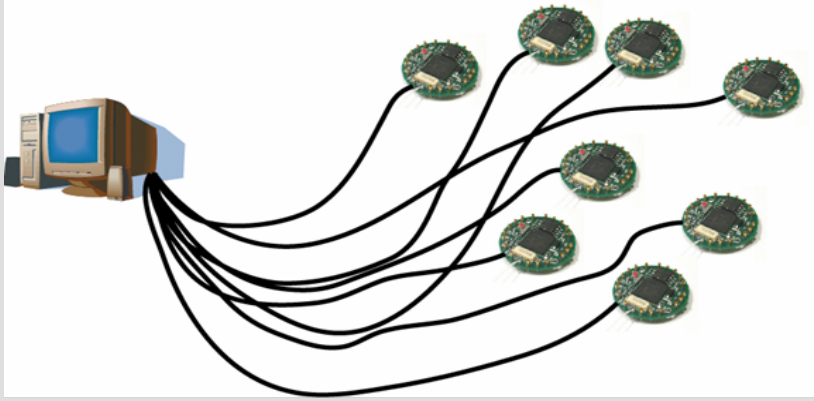


Figure abridged from D. Estrin/J. Elson

Code Distribution Using A Fixed Testbed

- Traditional test grid
  - Wired backchannel
  - Programming and logging from a remote station
  - Simple centralized control and data collection



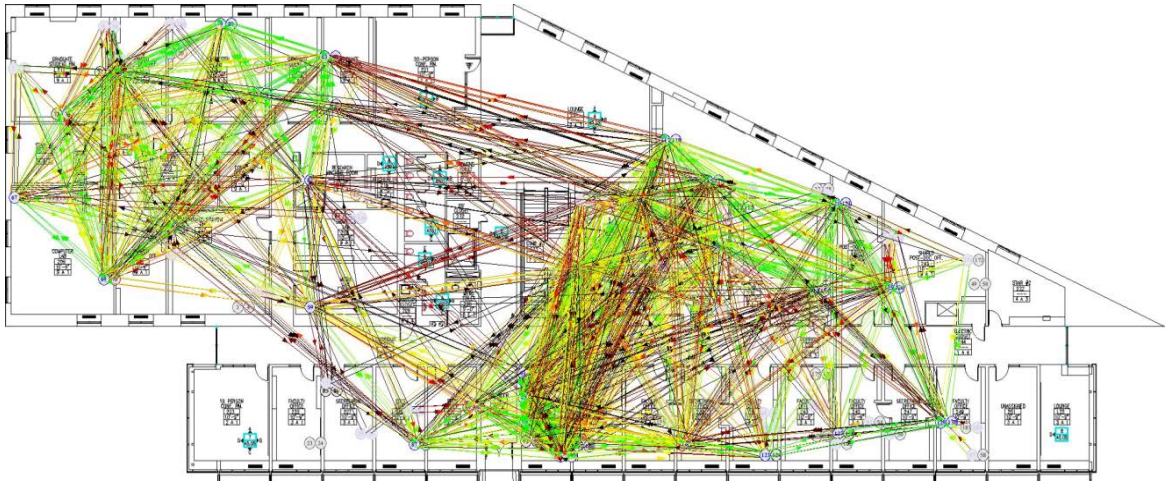
MoteLab @ Harvard

Ethernet-based

Multi-user arbitration

~200 nodes

Indoor environment



[Werner-Allen – IPSN 2005]

WSN Design and Development

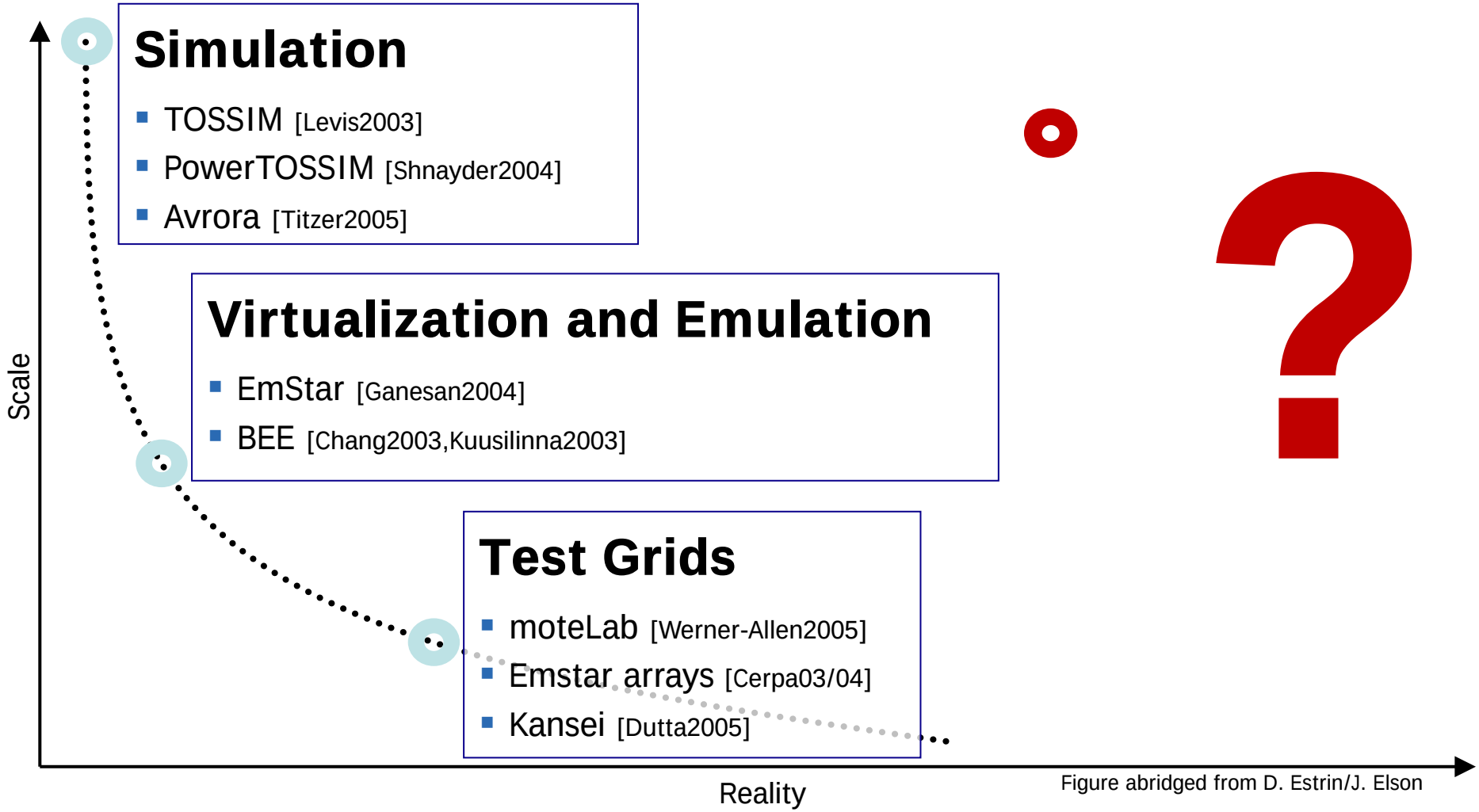


Figure abridged from D. Estrin/J. Elson

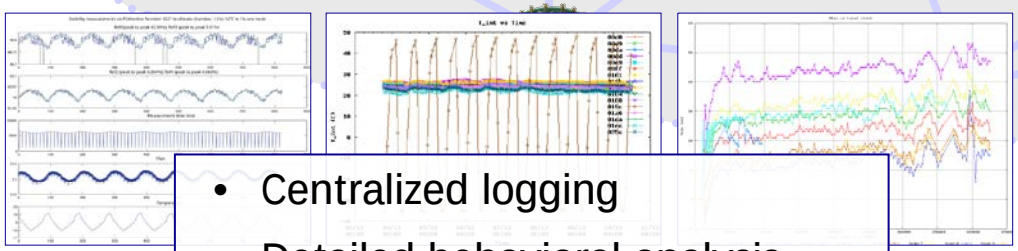
WSN Testbed – Deployment-Support Network

DSN Testbed Key Differentiators

- Distributed observers, local intelligence
- Mobility: Wireless, battery powered

DSN Testbed Functionality

- Remote reprogramming
- Extraction of log data
- Stimuli, e.g. fault injection
- Synchronization of traces and actions

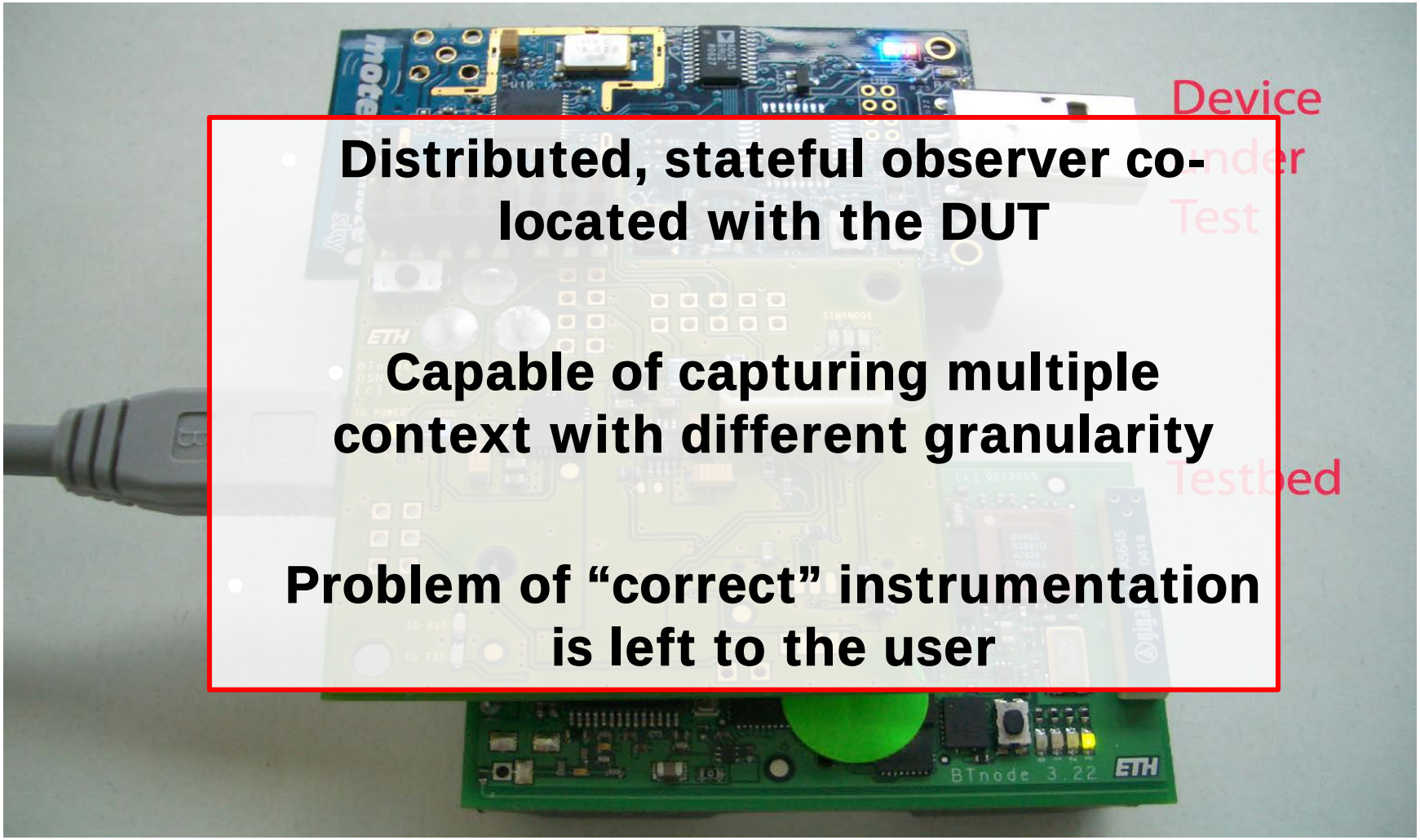


- Centralized logging
- Detailed behavioral analysis

Target Sensor Network

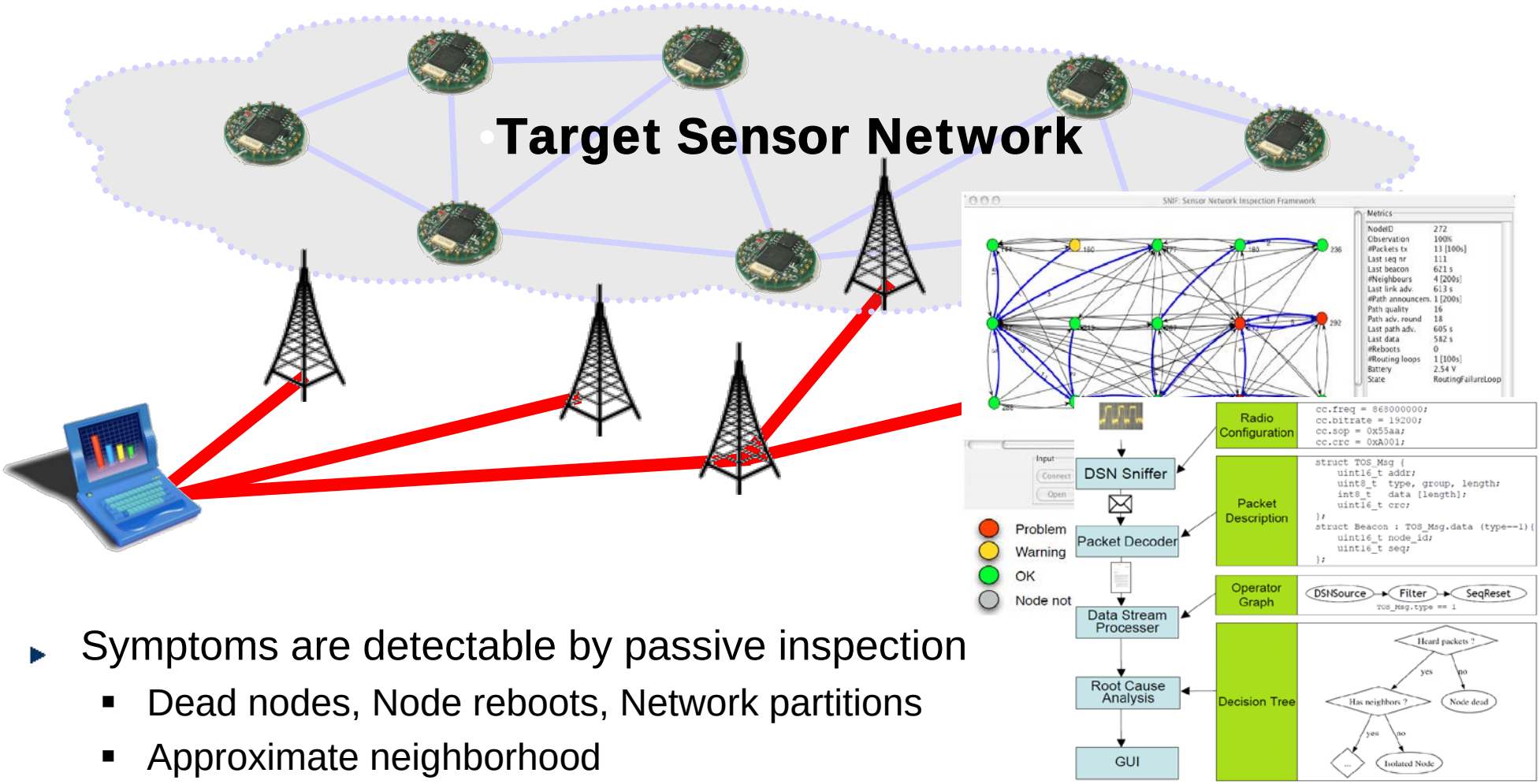
[SenSys2004, IPSN2005, EWSN2007]

DSN – A Distributed Testbed with Local State



- **Distributed, stateful observer co-located with the DUT**
- **Capable of capturing multiple context with different granularity**
- **Problem of “correct” instrumentation is left to the user**

Other Approaches – Packet Level Overhearing



- Symptoms are detectable by passive inspection
  - Dead nodes, Node reboots, Network partitions
  - Approximate neighborhood
- Failure root causes remain unclear

[Ringwald DCOSS 2007, Roemer IPSN 2009]

Evaluation With Passive Distributed Assertions

- Assertions: express belief that a condition holds

```
int i;
...
assert(i > 50);
```

- Assertions over distributed program variables
  - Value of variable i should equal value of variable k at node 100

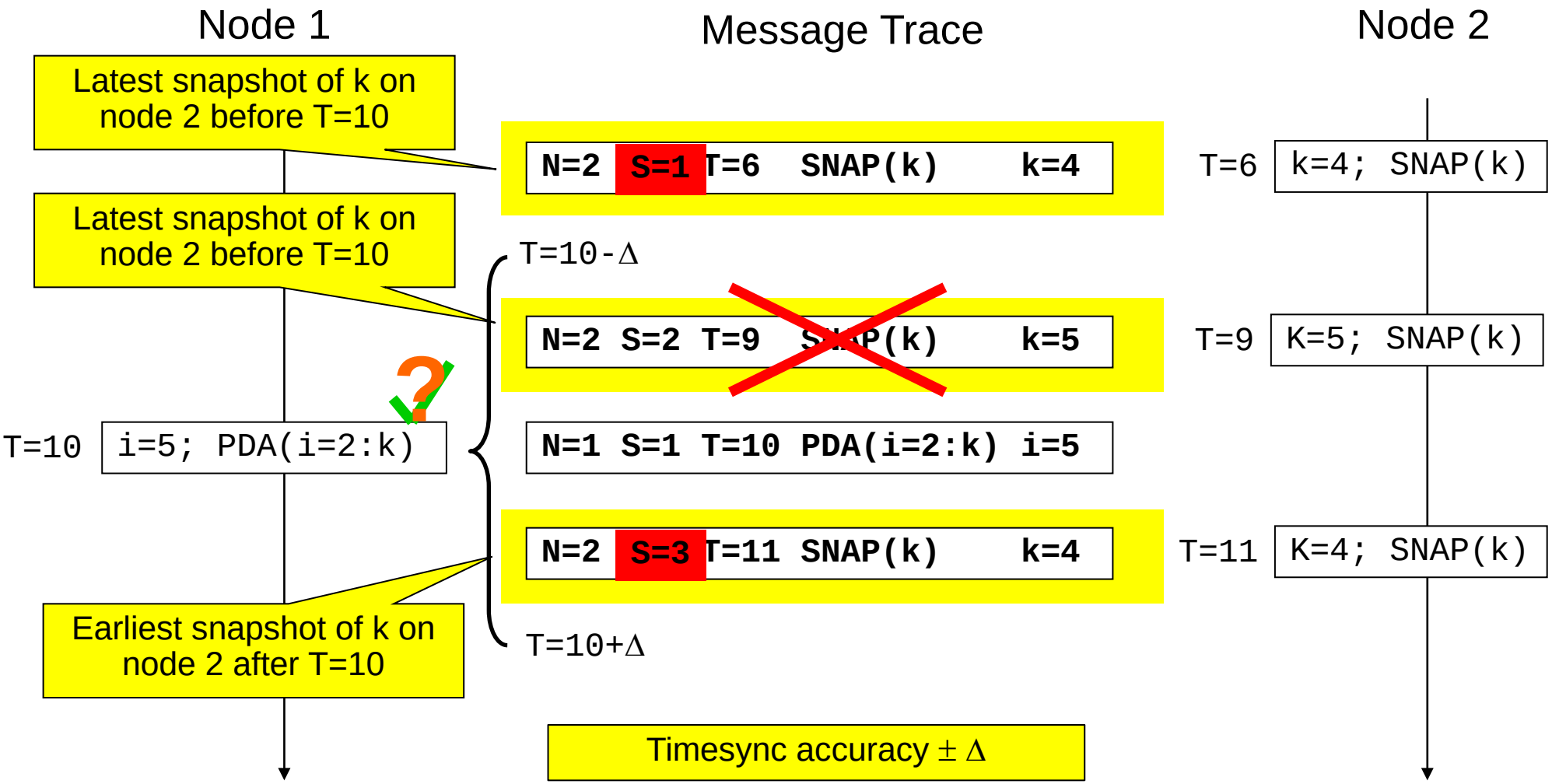
```
int i;
...
PDA(i = 100:k);
```

Variable k on node 100

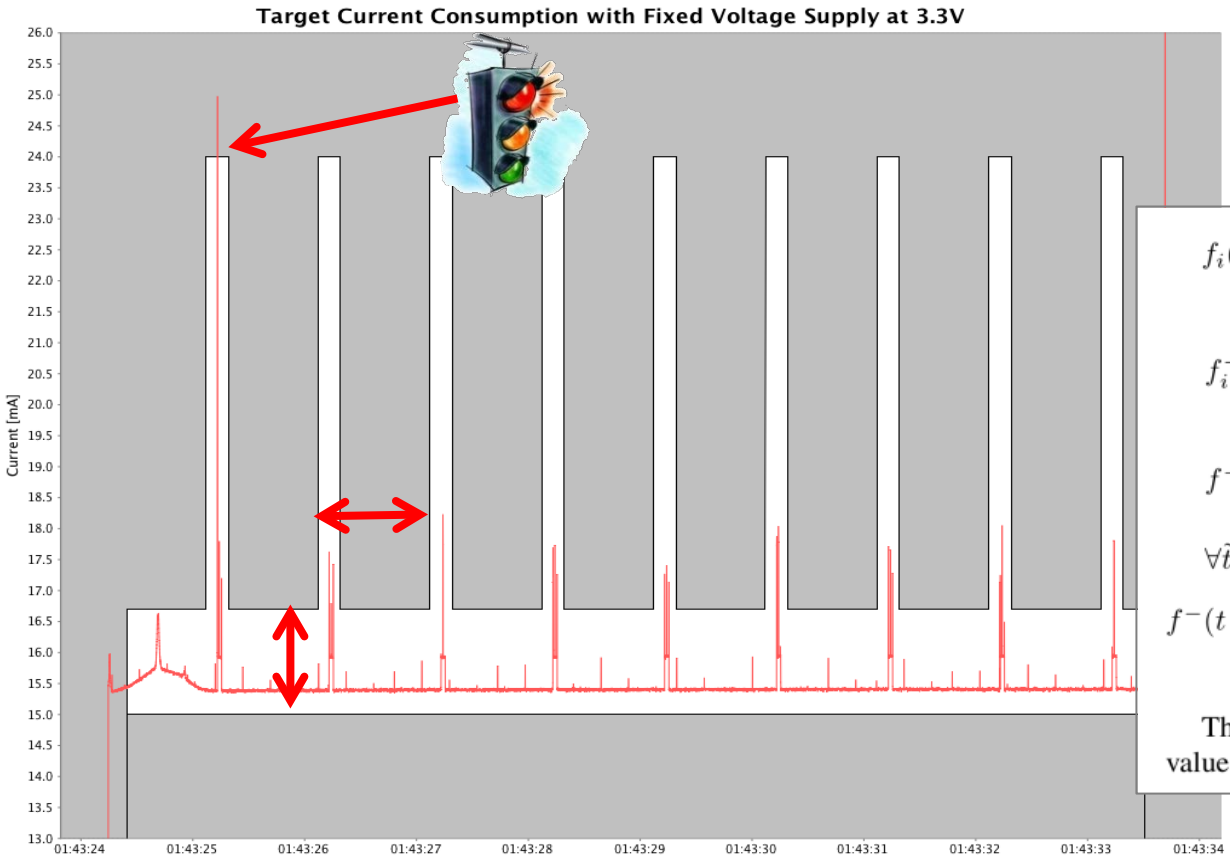
- Checked by passive inspection
  - Nodes only broadcast assertions (**PDA msg**), changes of relevant variables (**SNAP msg**)
  - Overheard by sniffer network
  - Minimize interference

```
k = ...;
SNAP(k);
```

Assertion Evaluation Example



Validation of Traces using Formal Bounds



$$f_i(t) = \begin{cases} a_0 + a_1 \cdot x + \dots & \text{if } t \in [t_{i-1}, t_i) \\ 0 & \text{if } t \notin [t_{i-1}, t_i) \end{cases}$$

$$f_i^-(t) = \begin{cases} f_i(t) - \Delta y^- & \text{if } t \in [t_{i-1}, t_i) \\ 0 & \text{if } t \notin [t_{i-1}, t_i) \end{cases}$$

$$f^-(t) = \sum_{i=1}^{n} f_i^-(t)$$

$$\forall \tilde{t} \in [-\Delta t, \Delta t], \forall i \in \mathbb{N}:$$

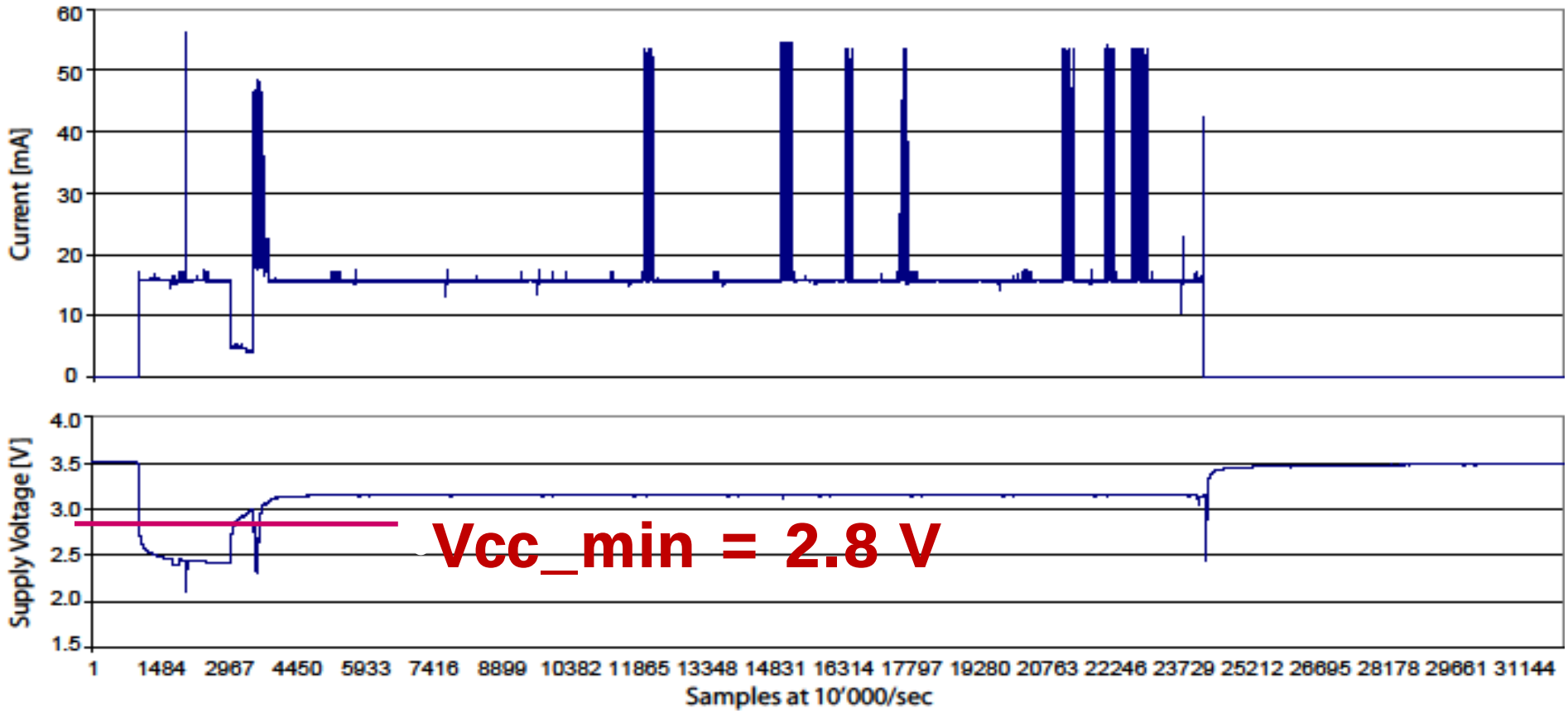
$$f^-(t + \tilde{(t_k)}) = \begin{cases} f^-(t_i^-) & \text{if } -f(ti^-) + f(t_{i^+})) \leq 0 \\ f^-(t_i^+) & \text{if } -f(ti^-) + f(t_{i^+})) > 0 \end{cases}$$

The upper bound f^+ follows accordingly with a bound value Δy^+.

- Assertions based on reference traces/specification
- Integrated with each build (regression testing)

[WEWSN2008, SUTC2008]

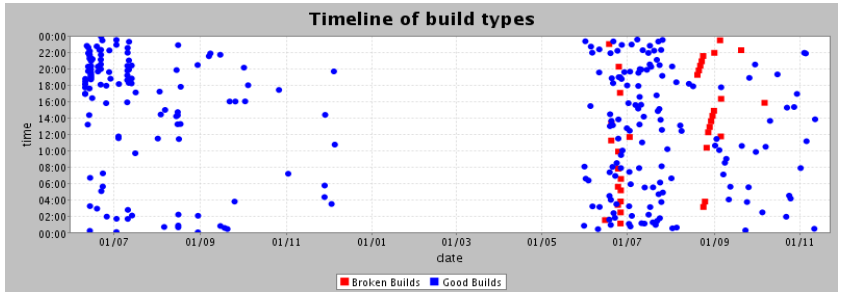
Pitfalls – Dangerous Voltage Drops

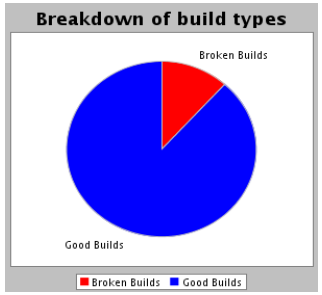


Regression Tests Using Continuous Integration

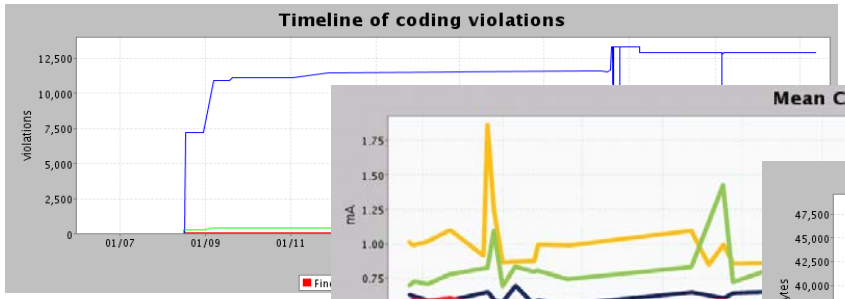
On code change applications are built from scratch and analyzed

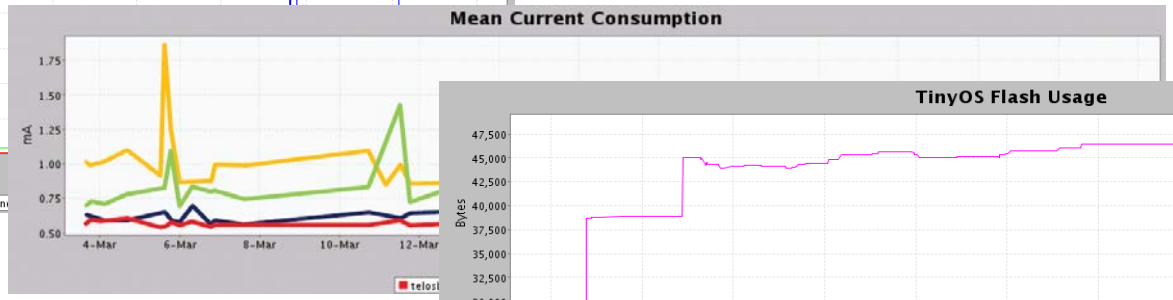
- Standard practice in enterprise level software development
- Deeper understanding of long term development trends
- Service to the TinyOS community, increasing software quality

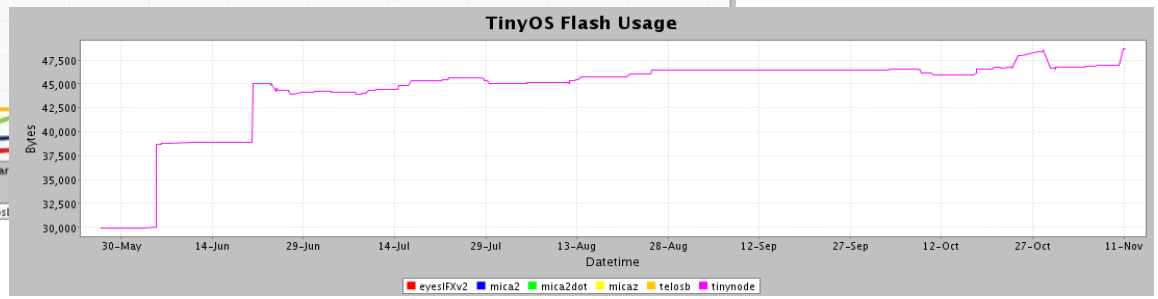




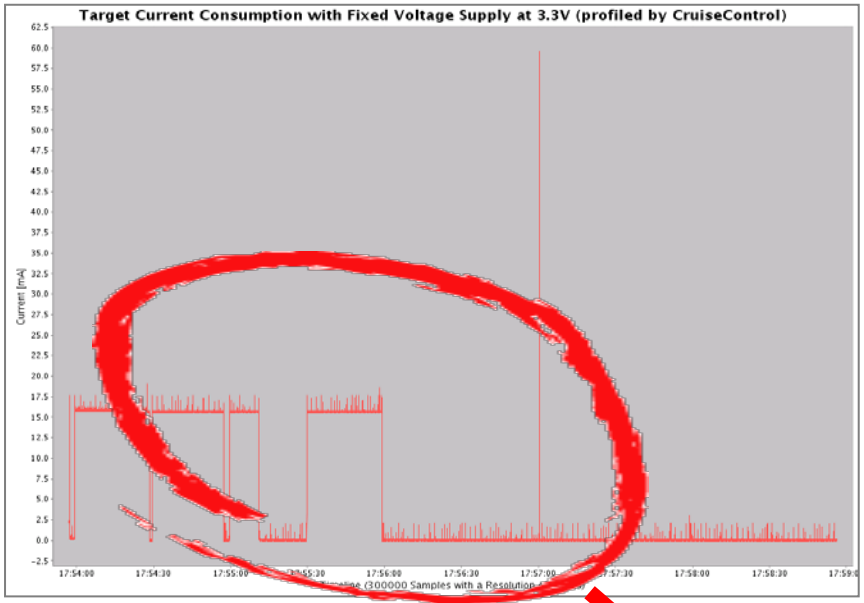
+4500 TinyOS-2.x regression builds over the last 2 years at ETHZ

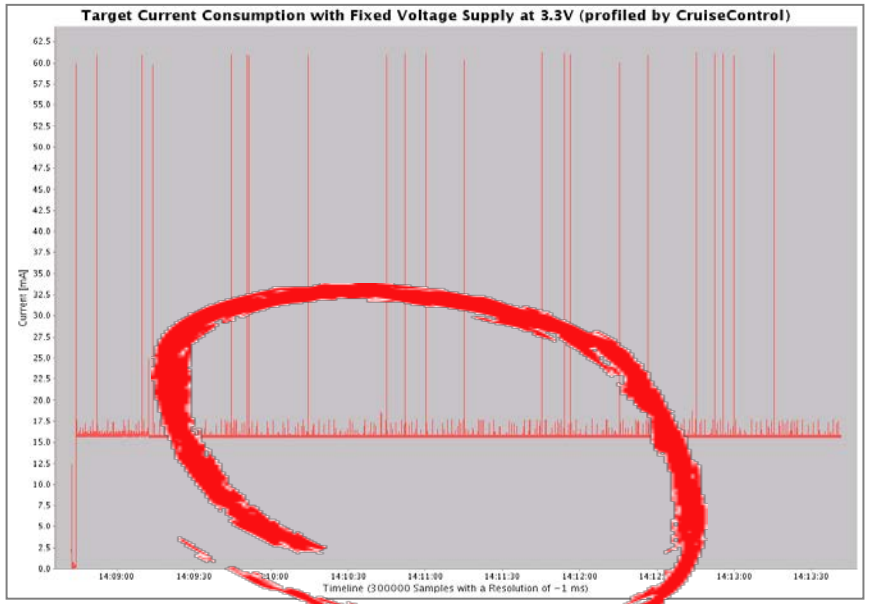


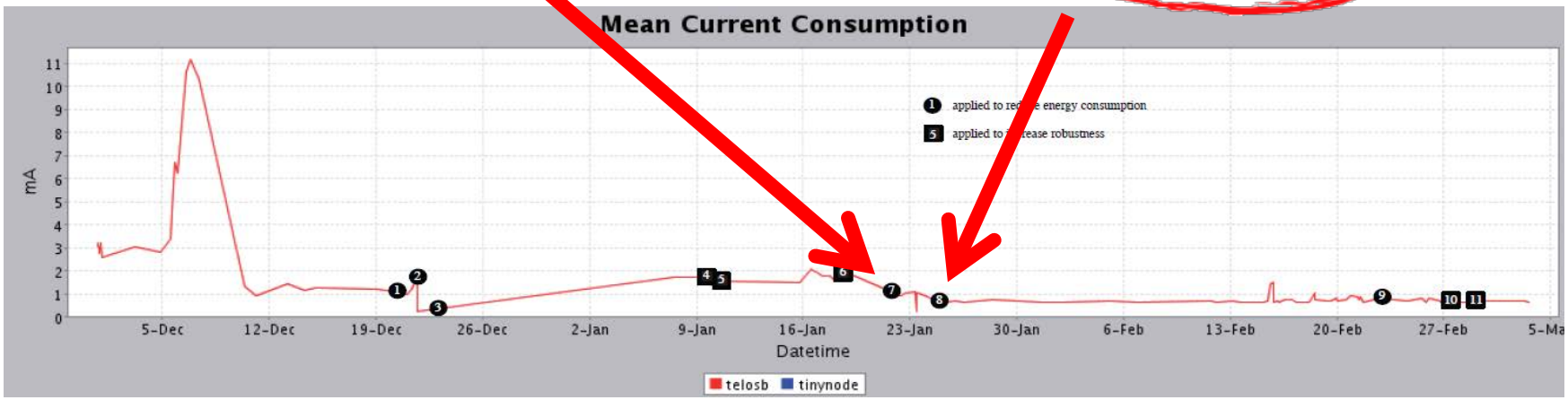




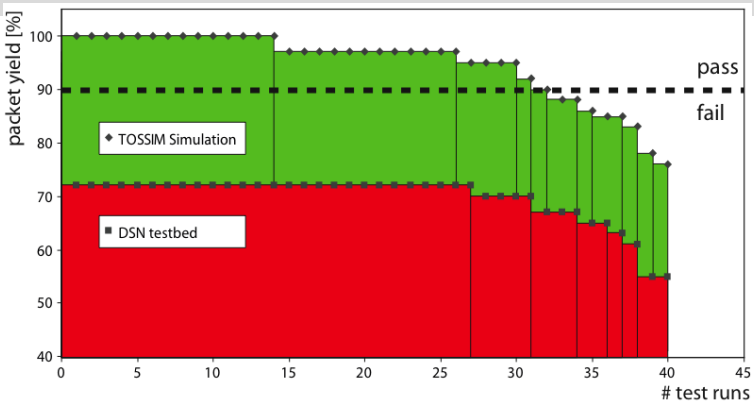
Quantification of Change with Visualizations



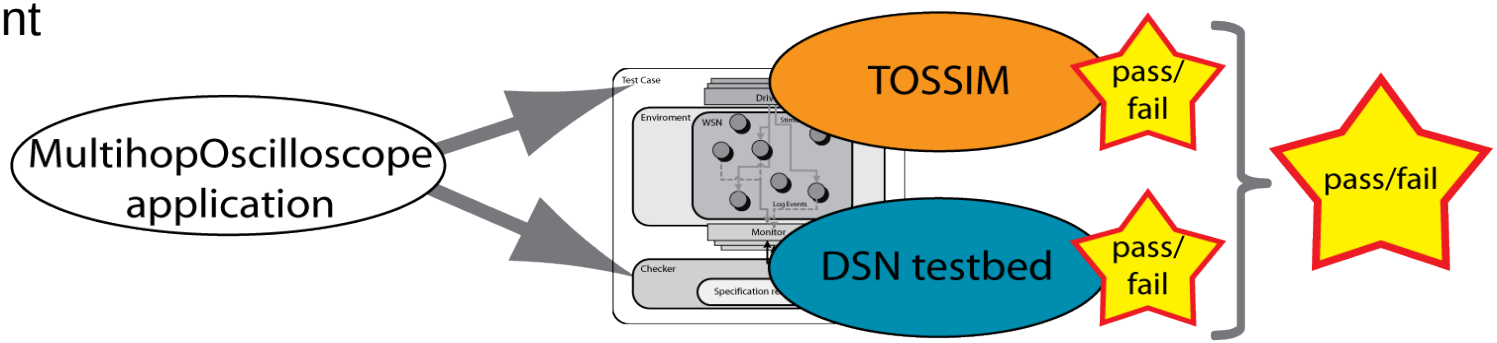




Multilevel Regression Testing Framework



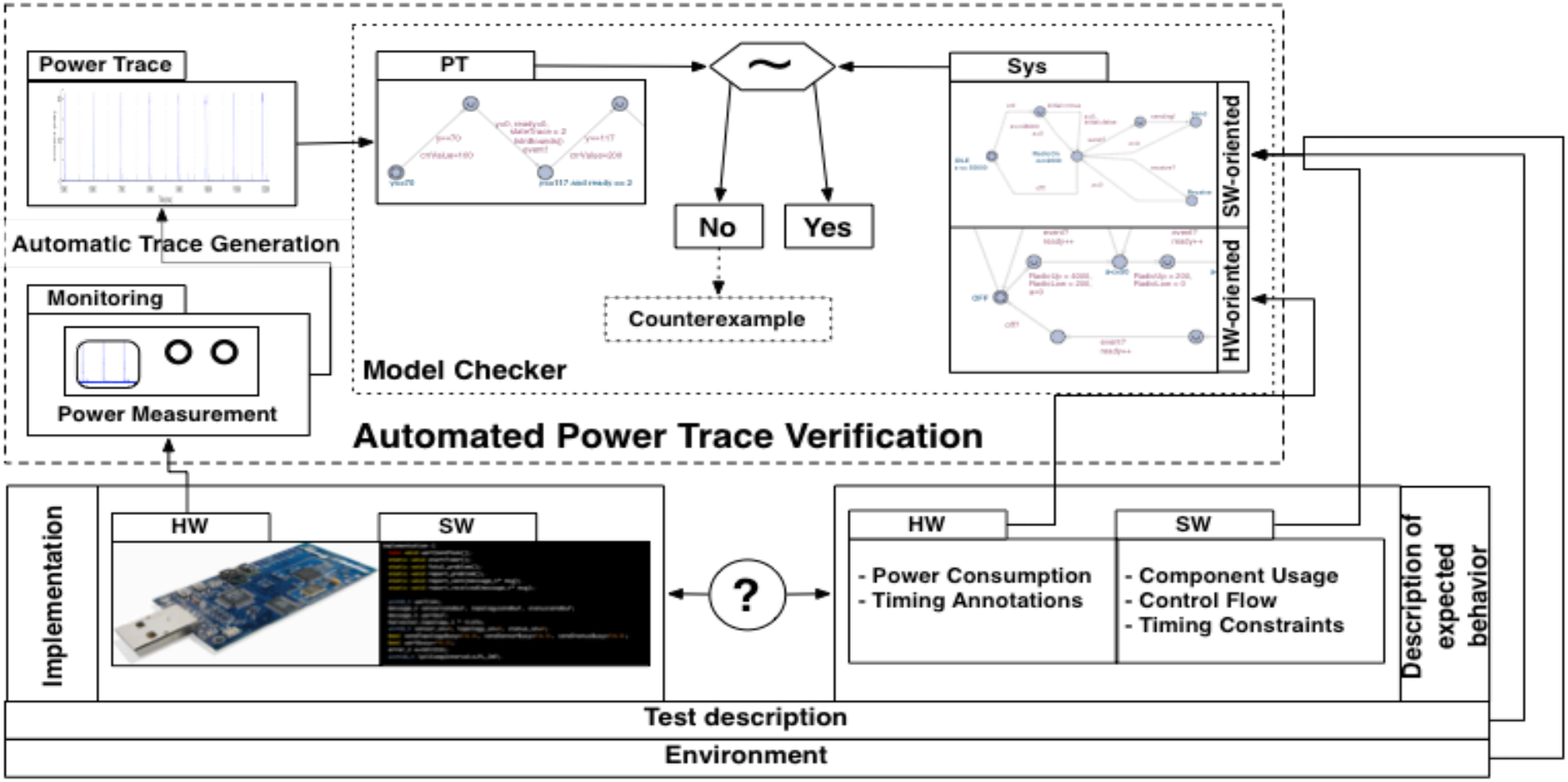
- Multiple levels of abstraction
  - Simulation (e.g. TOSSIM, AVRORA)
  - Emulation (e.g. EmStar)
  - Testbeds (e.g. DSN, moteLab)
- "Test Case" wraps around application code
  - Environment
  - Monitors
  - Stimuli



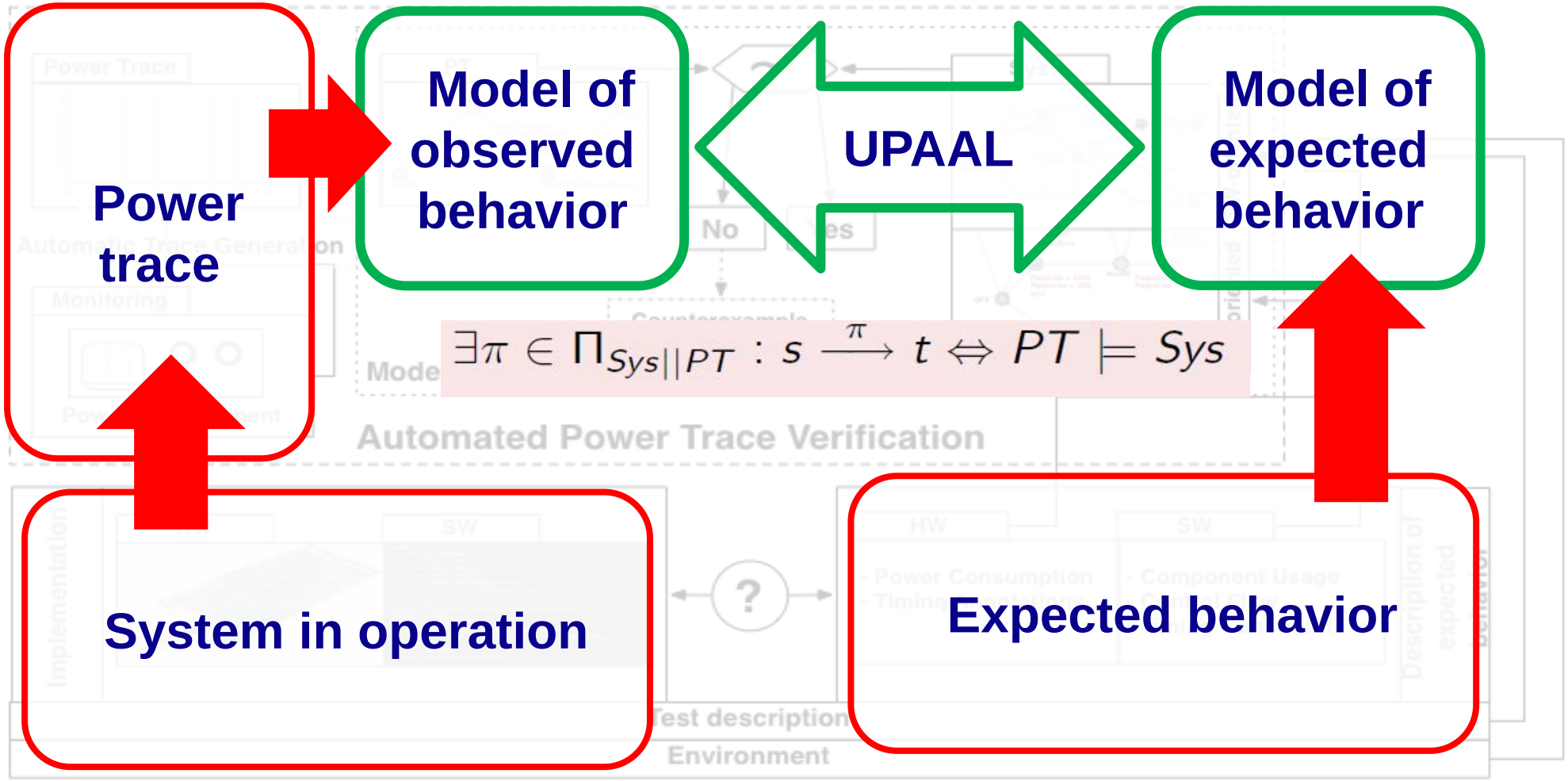
- Global pass/fail conditions
  - Unit testing
  - Parameter extraction, reporting

[EmNets 2007]

Conformance Testing using Timed Automata

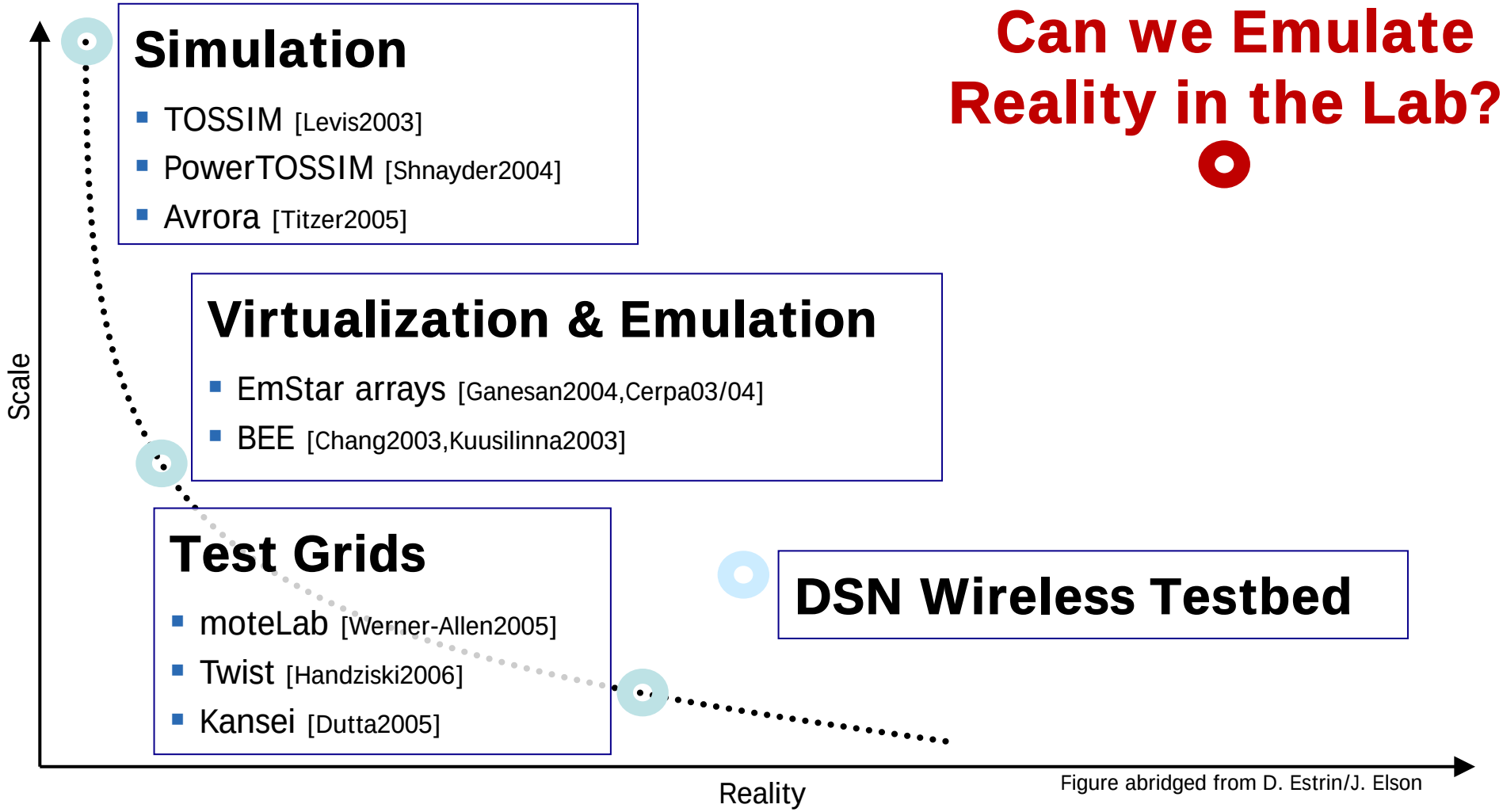


Conformance Testing using Timed Automata



[Woehrle FORMATS 2009]

WSN Design and Development Tools

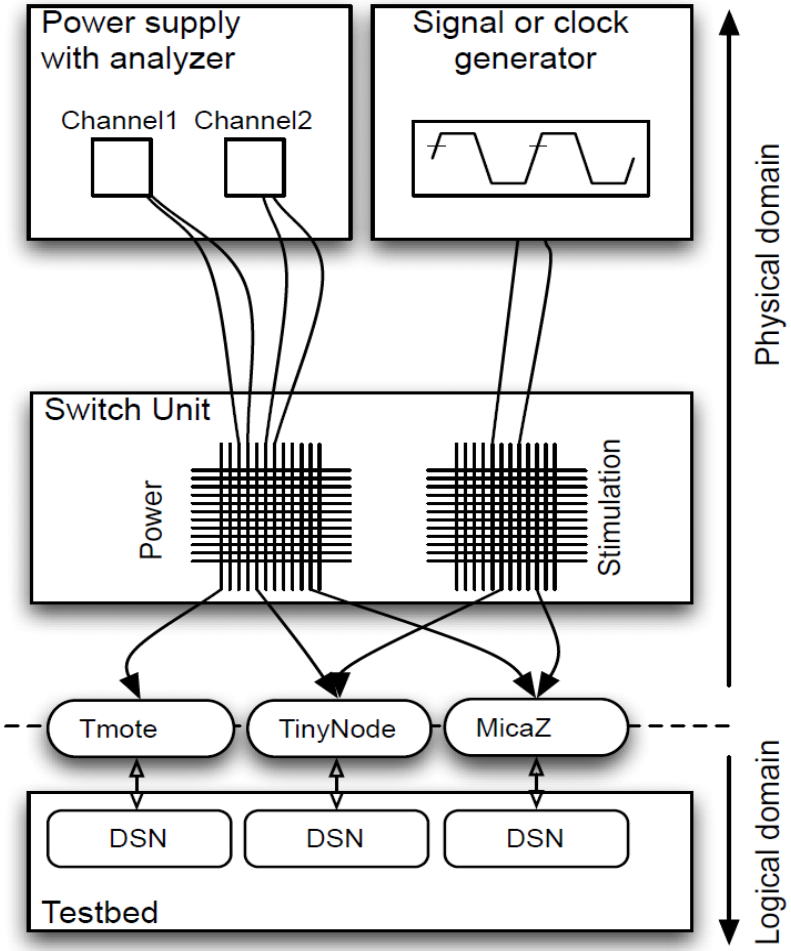


Can we Emulate Reality in the Lab?

Figure abridged from D. Estrin/J. Elson

Physical Emulation Architecture

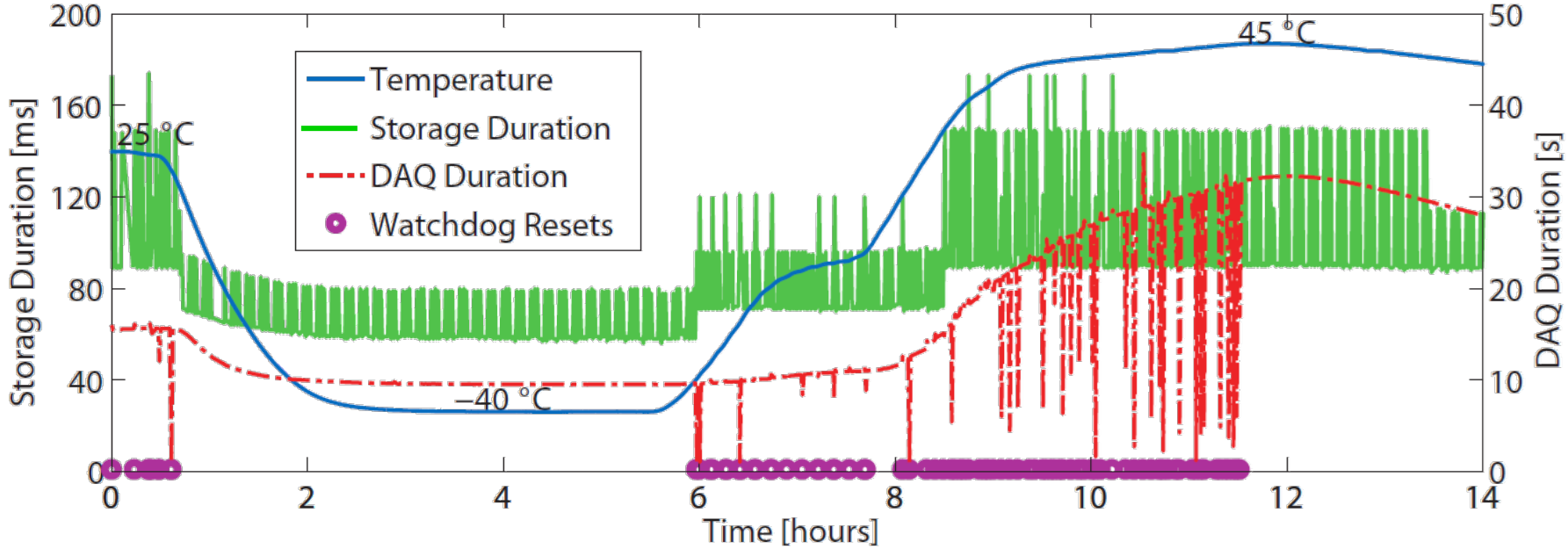
- Influence of power sources/quality
- Detailed physical characterization
- Emulation of environment and resources
  - Temperature Cycle Testing (TCT)
  - Controlled RF attenuation
  - Sensor stimuli and references



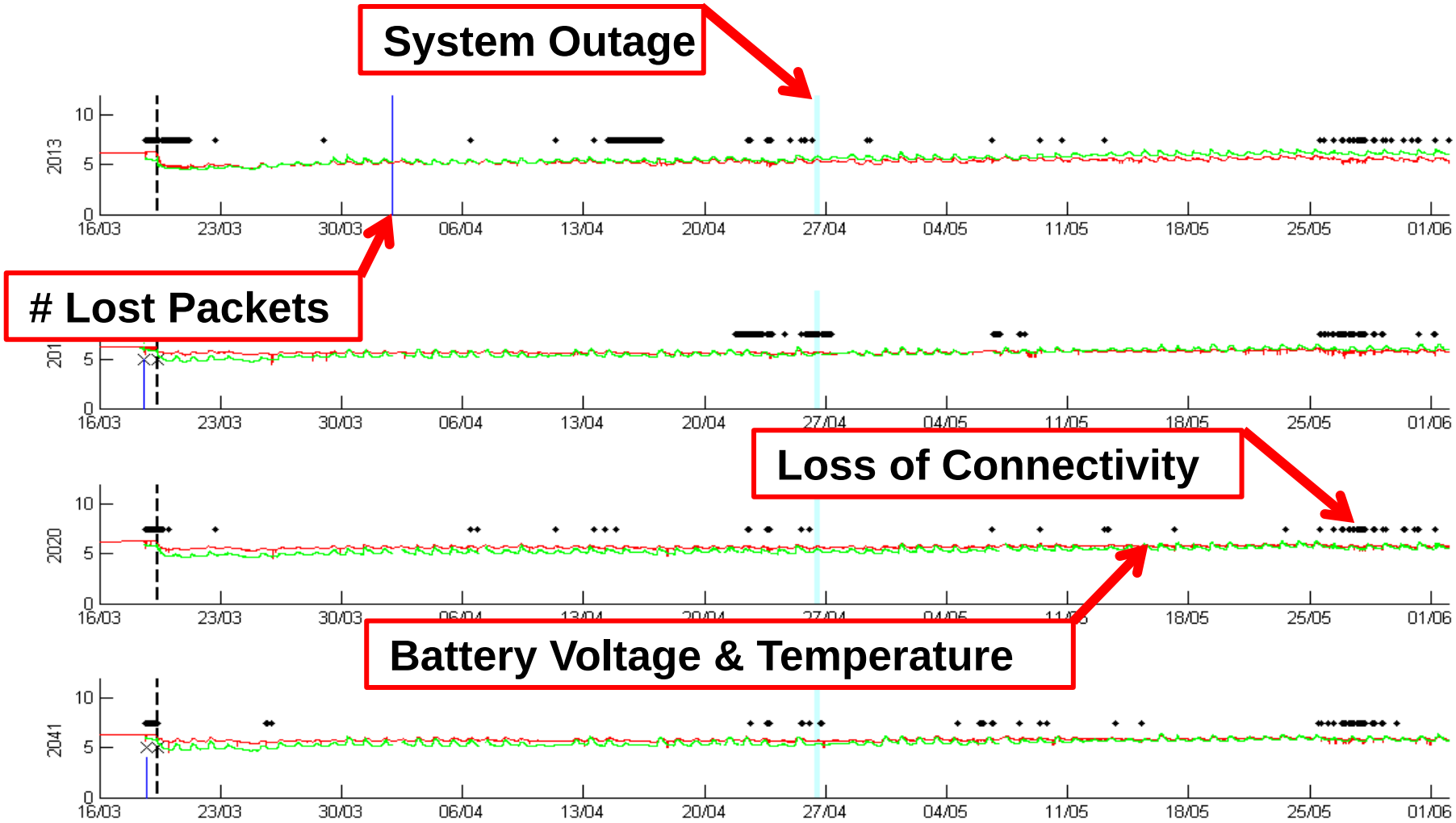
[EmNets2007]

Impact of Environmental Extremes

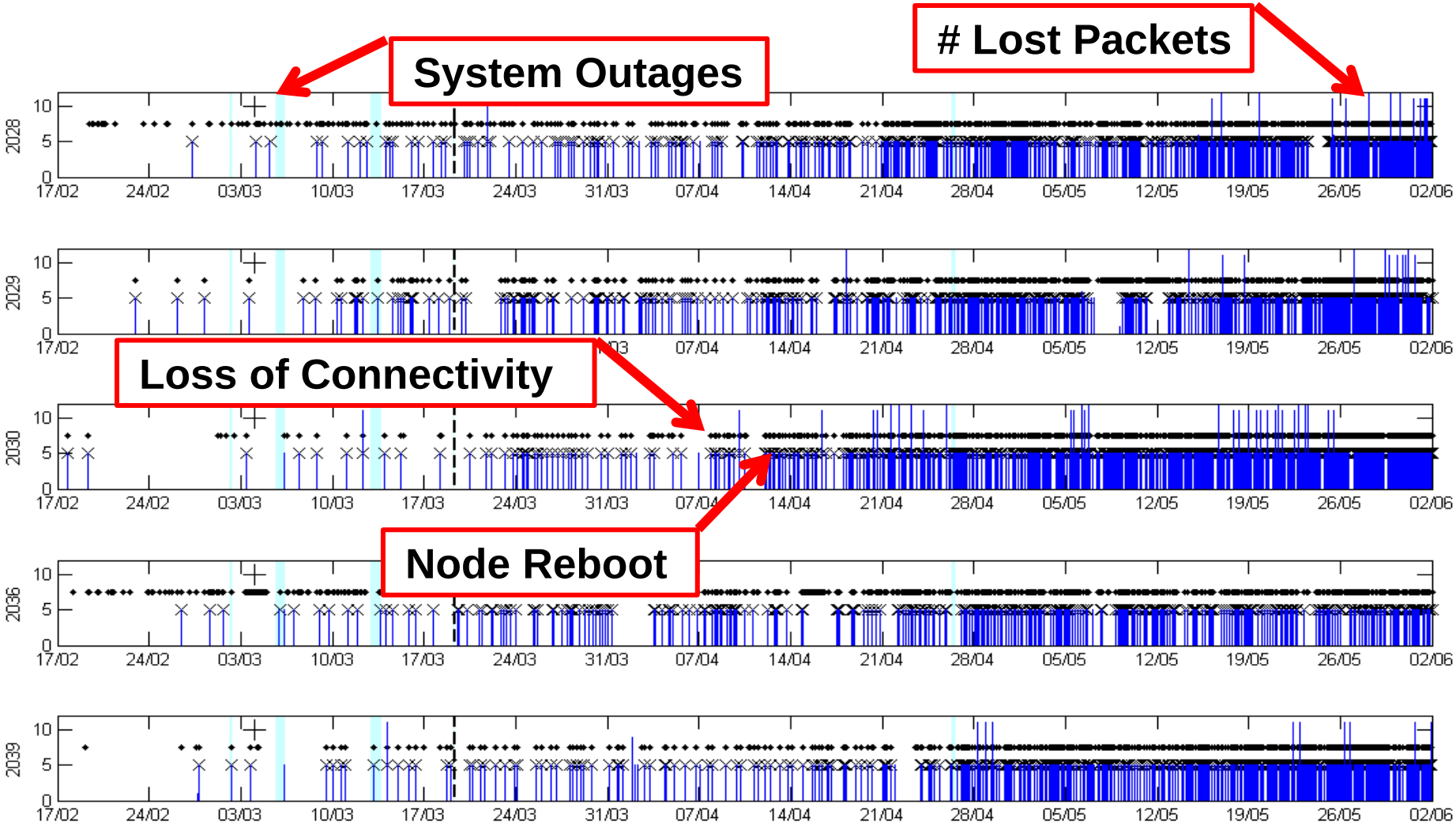
- Software testing in a climate chamber
  - Clock drift compensation yields ± 5ppm
- Validation of correct function
- Tighter guard times increase energy efficiency



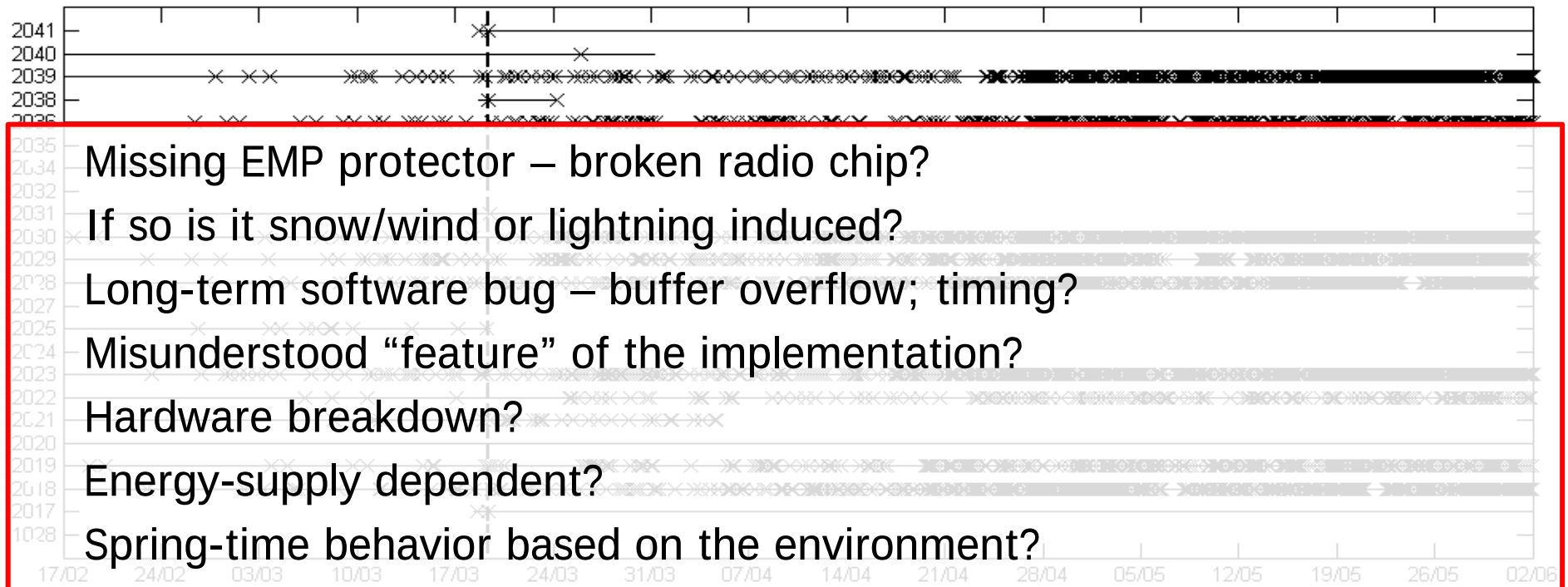
Experiences – Behavioral Data from the Field



3 Months Later – Increasing Runtime Errors



Understanding the Exact Causes is Hard...



Missing EMP protector – broken radio chip?

If so is it snow/wind or lightning induced?

Long-term software bug – buffer overflow; timing?

Misunderstood “feature” of the implementation?

Hardware breakdown?

Energy-supply dependent?

Spring-time behavior based on the environment?

Cosmic rays at 3500 m a.s.l. ?

Can external effects trigger the reed contact on the reset?

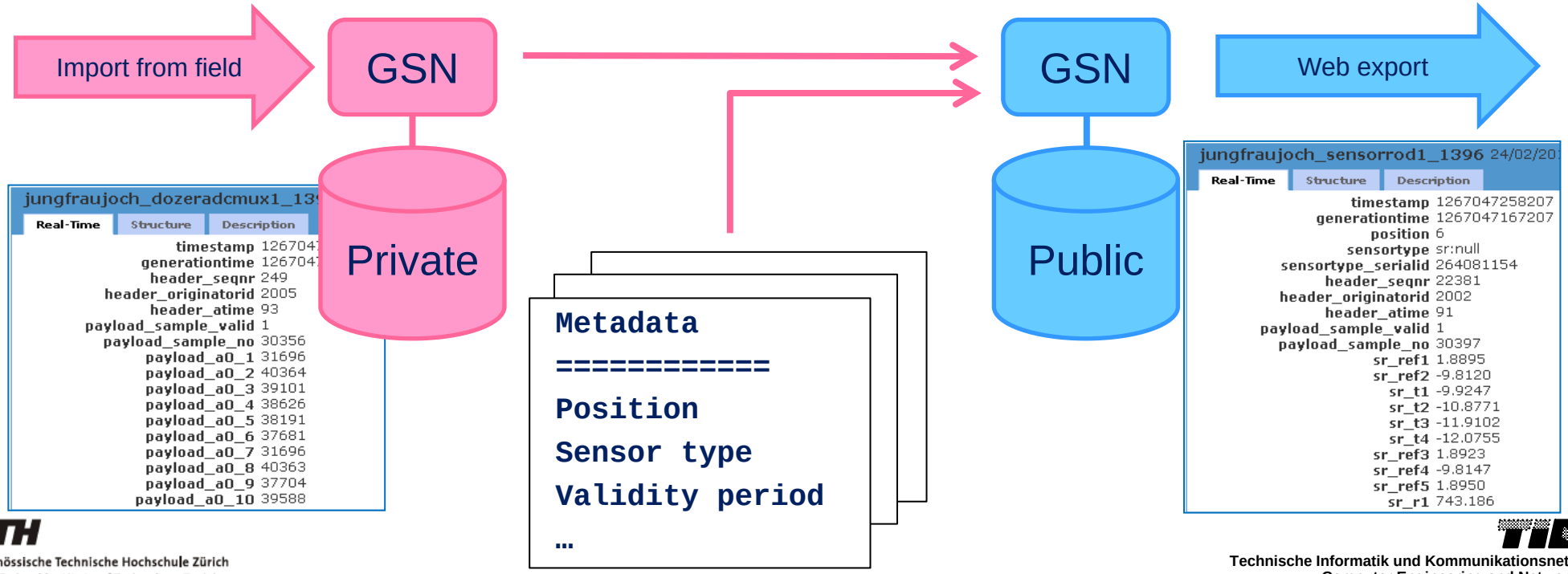
...

Probably not feasible to test / reproduce this in the lab!

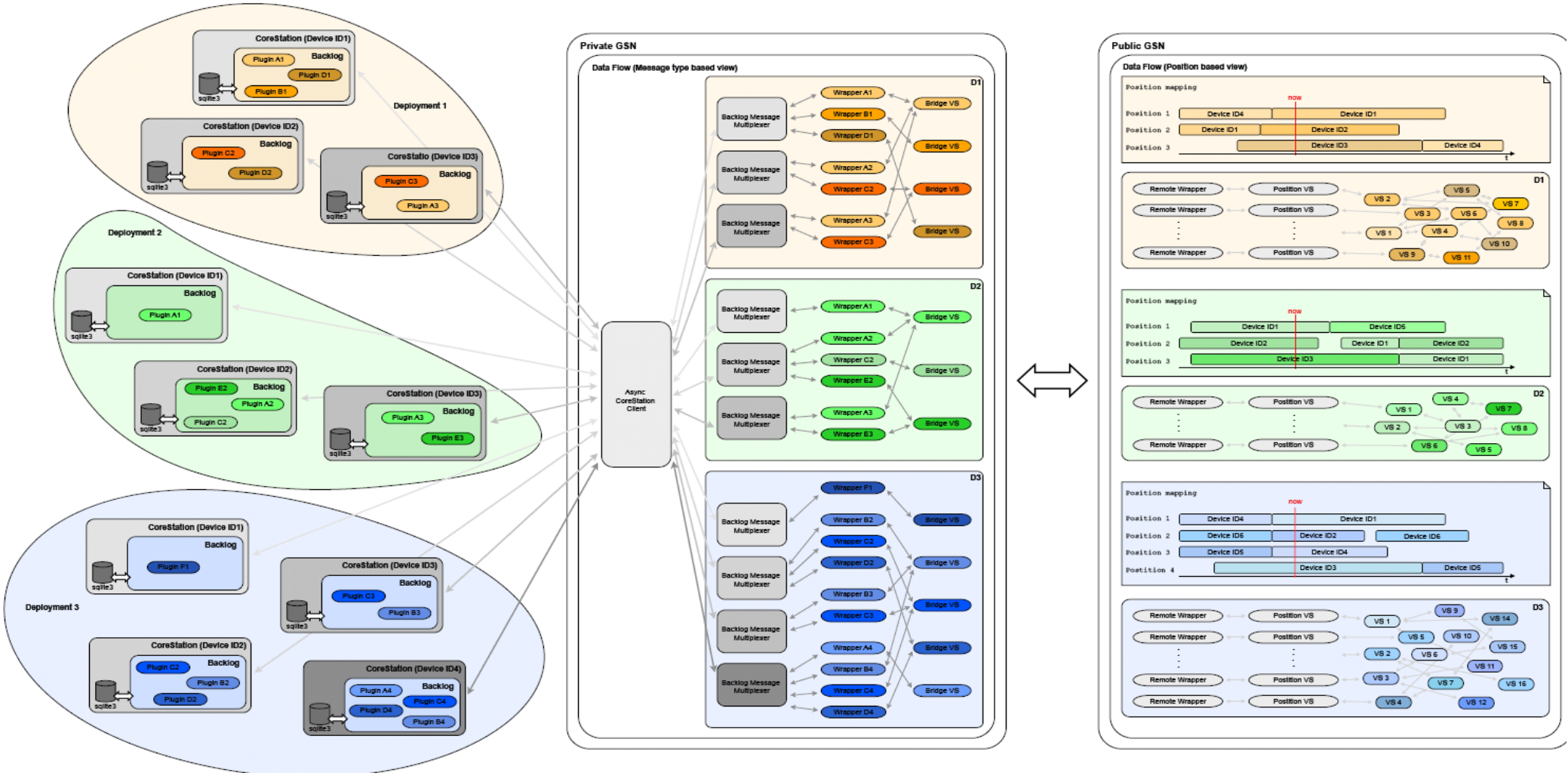
End-to-end Data Management – Data Quality Analysis/Control

Data Management – Online Semantic Data

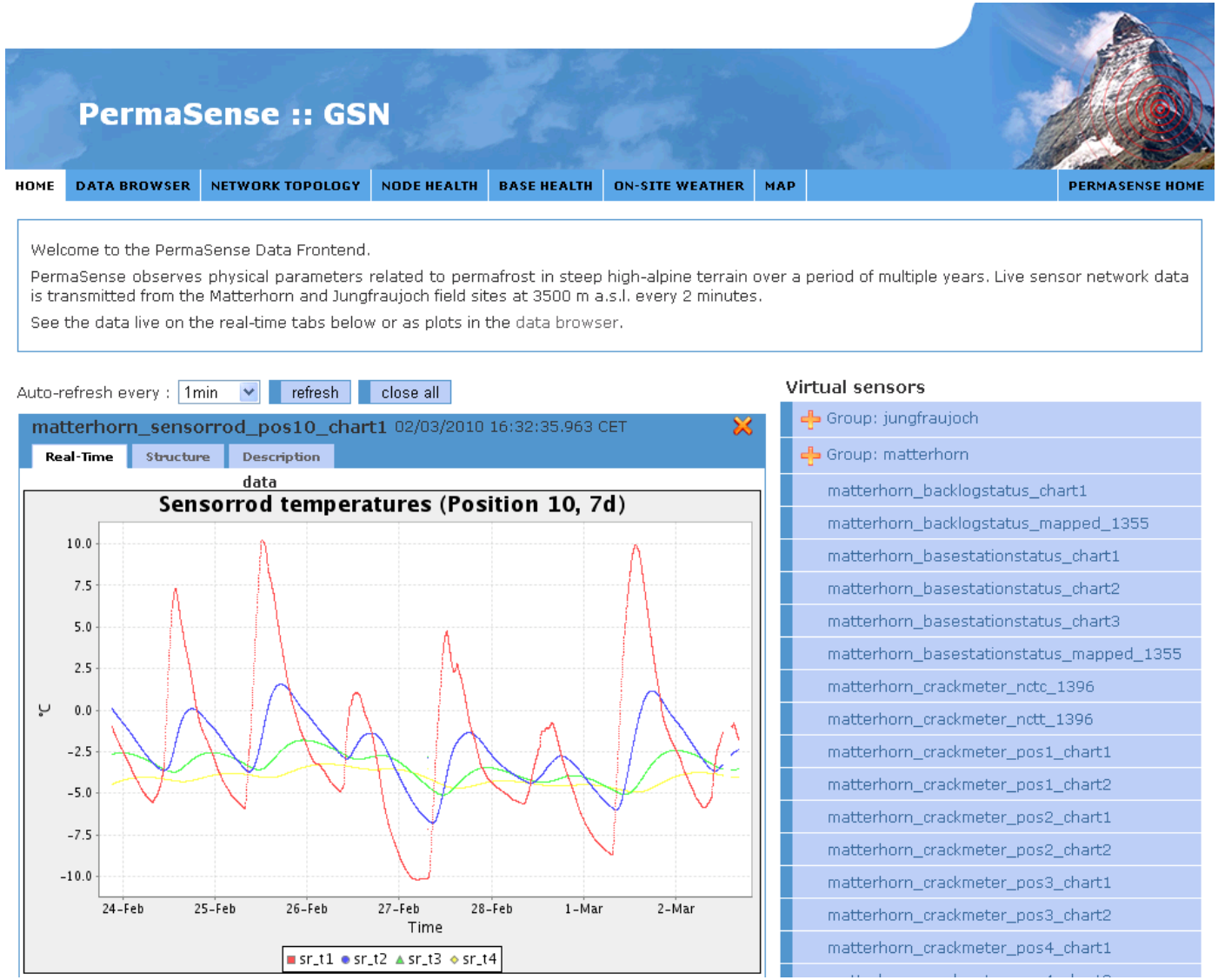
- Global Sensor Network (GSN)
  - Data streaming framework from EPFL (K. Aberer)
  - Organized in “virtual sensors”, i.e. data types/semantics
  - Hierarchies and concatenation of virtual sensors enable on-line processing
  - Dual architecture translates data from machine representation to SI values, adds metadata



Multi-Site, Multi-Station, Multi-Revision Data…

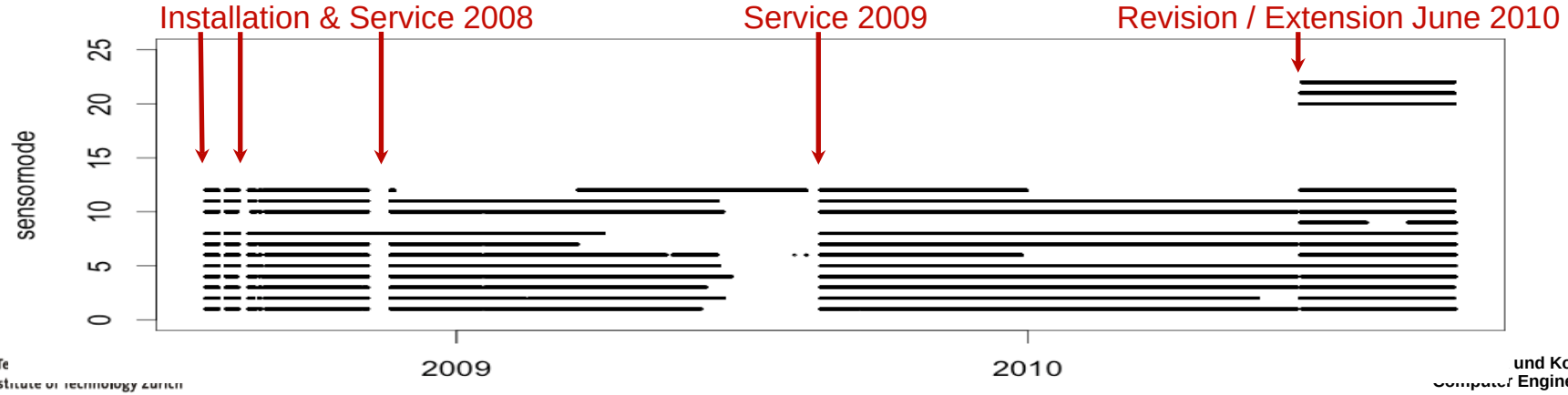


Central Web-based Data Access



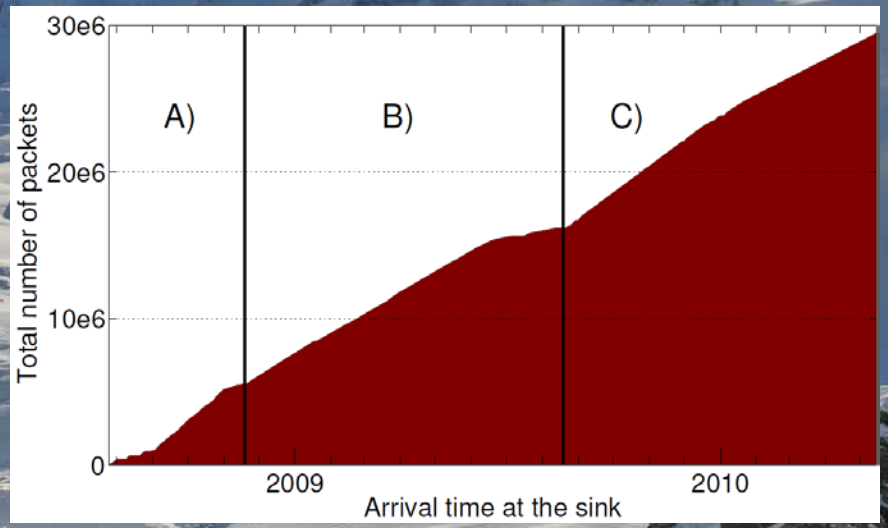
Value of Online Data

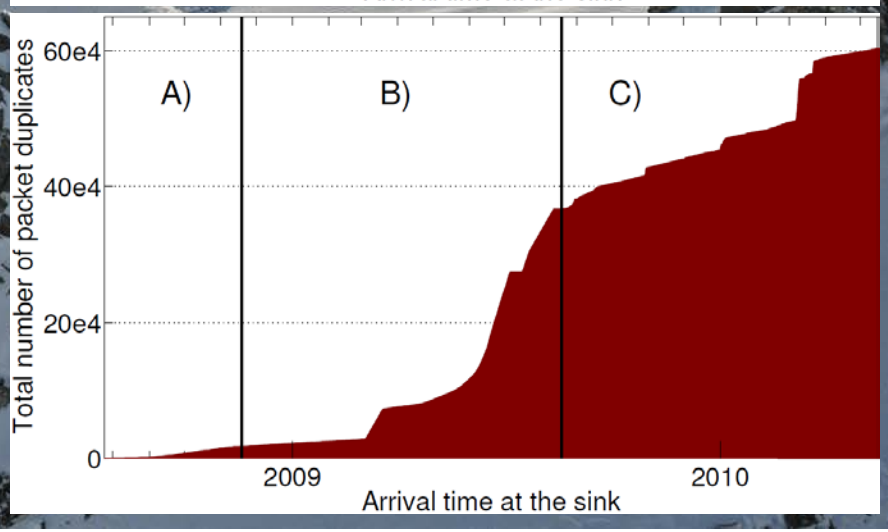
- **Data quality/integrity**
  - Inconsistencies
  - Duplicates
  - Gaps
  - Sporadic/systematic
  - Long-term stability
- **Equipment health**
- Future: Early warning…



Challenge: Data Integrity/Validation

- August 2008 – June 2010
- Single sink
- Up to 19 sensor nodes
- TinyOS/Dozer [Burri, IPSN2007]
- Constant rate sampling
- < 0.1 MByte/node/day





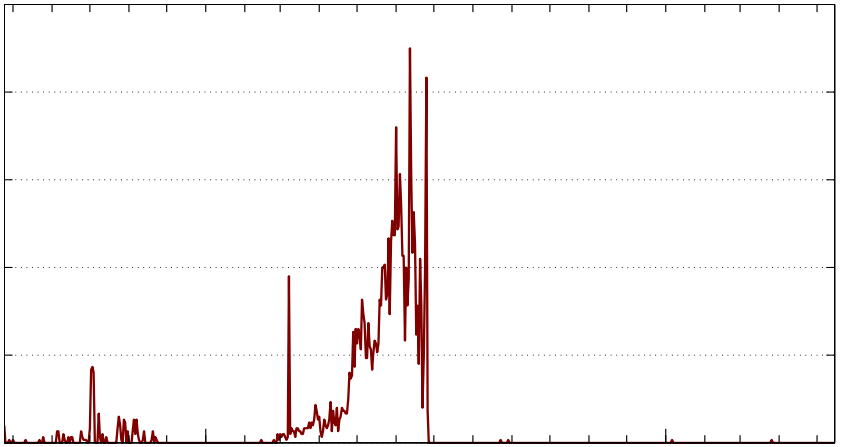
Sensor Data Outlier Filtering

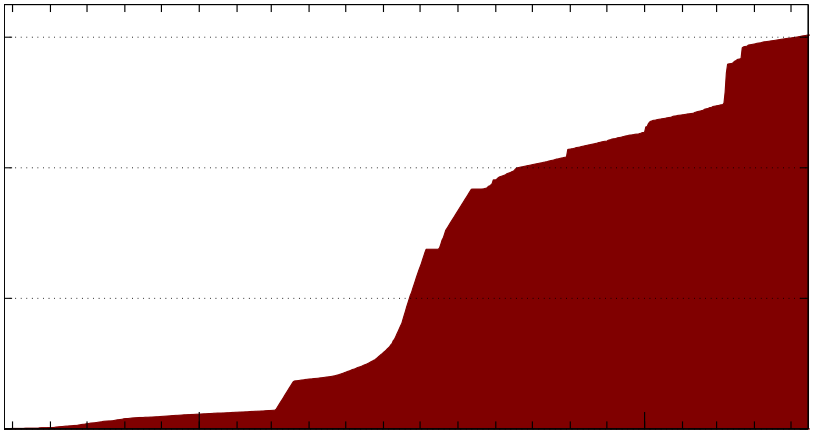
- A. Hasler: Threshold-based removal of bogus data, down sampling from 2 to 10 minutes sampling interval
- Tolle *et al.*: Temperature measurements, outlier rejection based on battery voltage level
- E. Elnahrawy *et al.:* Bayesian approach for cleaning noisy sensor data
- H. Jeung *et al.*: Data cleaning with model-based anomaly detector

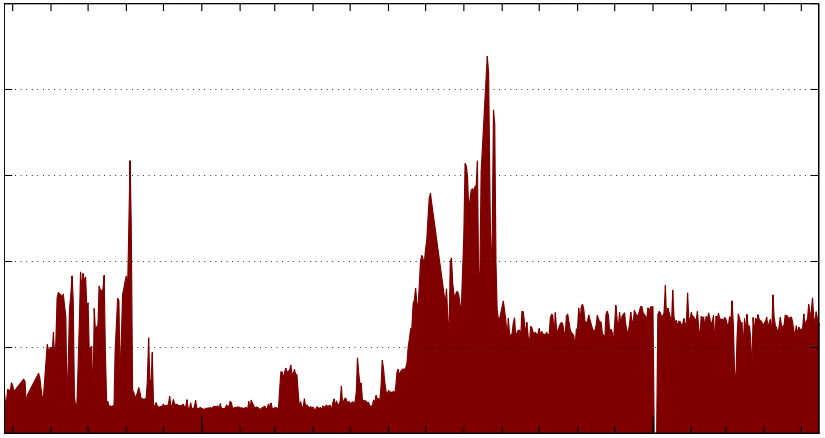
- Necessary step to mitigate artifacts of faulty sensors
- Usually done by scientific data user/domain expert
- Must assume a certain input data quality

Currently Untouched Artifacts

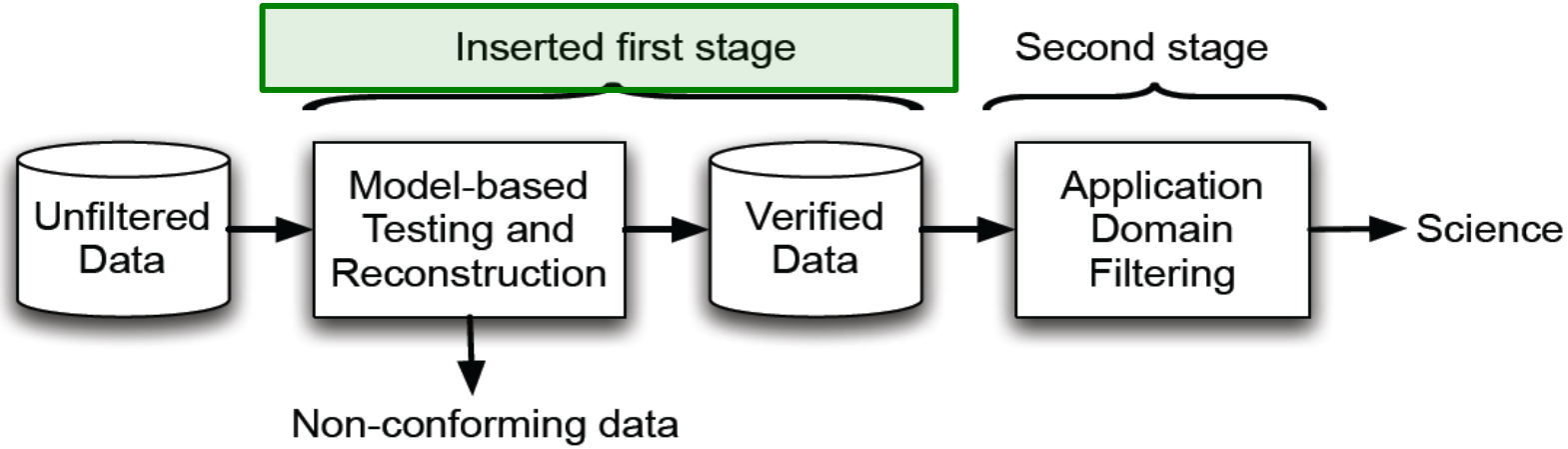
- We can observe
  - Packet duplicates
  - Node restarts
  - Order inconsistencies
    - Temporal vs. logical







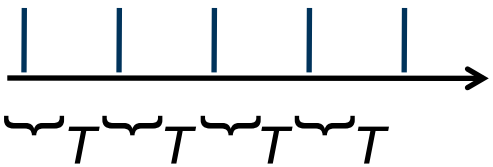
Goals of the Data Analysis



- Validate packets based on a model of the real system
  - For valid packets
    1. Add extra packet ordering information
    2. Provide guarantees on time information
  - Mark other packets as non-conforming

Model of Multi-hop Data Collection

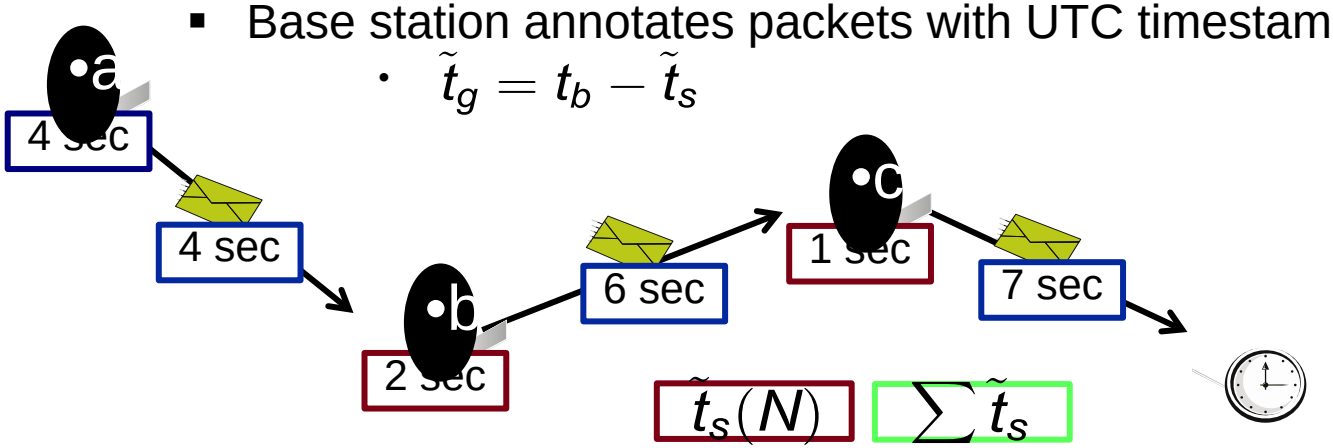
- Periodic sampling
  - Sampling period T



- Sequencing
  - Increasing sequence number
  - Resets on arithmetic overflow

$$s(i) = i \bmod s_{max}$$

- Elapsed time on arrival
  - Sensor nodes measure packet sojourn time
  - Base station annotates packets with UTC timestamps
    - $\tilde{t}_g = t_b - \tilde{t}_s$

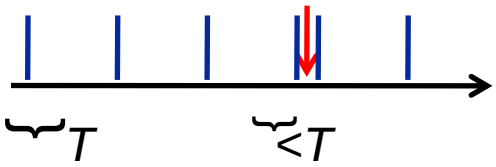


2011/04/14 10:03:31 – 7 sec
= 2011/04/14 10:03:24

Error Model

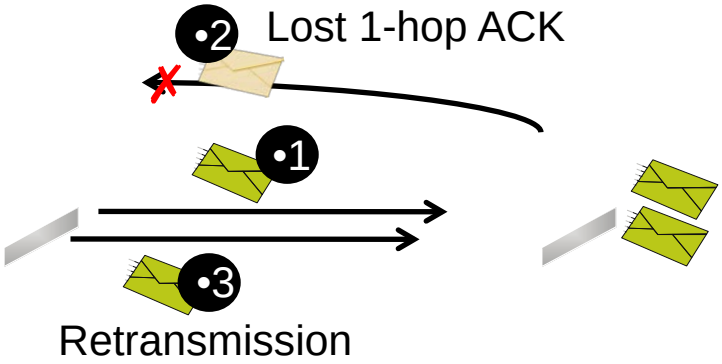
- Clock drift $\rho \in [-\hat{\rho}; \hat{\rho}]$
  - Affects measurement of
    - Sampling period T
    - Packet sojourn time t_s
  - Indirectly leading to ordering inconsistencies
    - Temporal vs. logical

- Node restarts
  - Cold restart: Power cycle
  - Soft restart: Watchdog reset

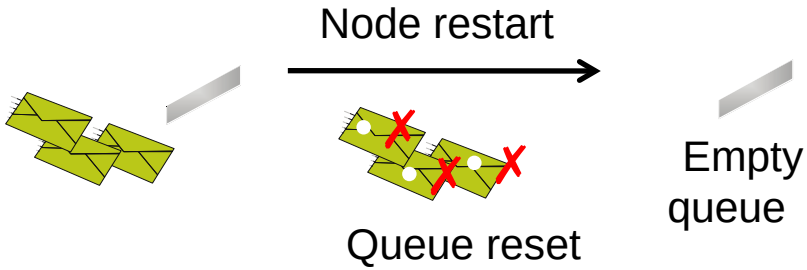


Shortens sampling period

- Packet duplicates



- Packet loss



Formal System Model (1/2)

Considering a single sensor node with source address o:

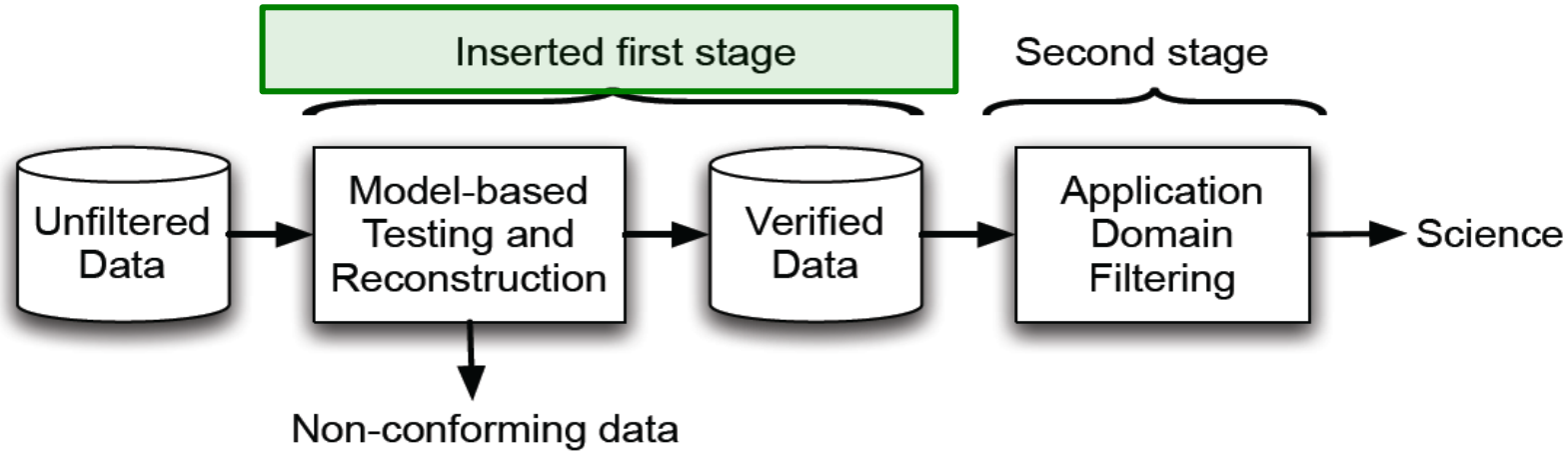
- Abstract sequence counter: i
- – at last cold restart: i_{offset}
- Packet sequence number: $s(i) = (i - i_{\text{offset}}) \bmod s_{\max}$
- Sampling period: T
- Clock drift and resolution: $\rho \in [-\hat{\rho}; \hat{\rho}],\ \hat{t}_u$
- Packet generation time: $t_g(i) = t_g(i-1) + \dfrac{T}{1+\rho(i)}$

Formal System Model (2/2)

- Estimated sojourn time on node N: $\tilde{t}_s(N, i)$
- Estimated total sojourn time: $\tilde{t}_s(i) = \sum_{N \in \mathcal{N}_{(o,i)}} \tilde{t}_s(N, i)$
- Arrival time at base station: $t_b(i)$
- Estimated generation time: $\tilde{t}_g(i) = t_b(i) - \tilde{t}_s(i)$
- Maximum network diameter: $\hat{h}$
- Error bounds on generation time calculation:

$$t_g(i) - \tilde{t}_g(i) \in \left[-\frac{\tilde{t}_s(i) \cdot \hat{\rho} + \hat{h} \cdot \hat{t}_u}{1 - \hat{\rho}}, \frac{\tilde{t}_s(i) \cdot \hat{\rho}}{1 + \hat{\rho}} \right]$$

Data Processing



- Input format: $(o, s, p, \tilde{t}_s, t_b)$
  - Origin o, Sequence number s, total sojourn time $\tilde{t}_s$, payload p, arrival time t_b
- Output format: $(o, s, p, \tilde{t}_s, t_b, id, [t_g^l, t_g^u])$
  - Unique packet identifier *id* reflects temporal order of generation
  - Bounds on packet generation time $[t_g^l, t_g^u]$

Analysis Concepts

- Remove uncertainty caused by sequence number
  $s(i) = (i - i_{\text{offset}}) \bmod s_{\text{max}}$
  - Assign packets to epochs
  - Determine unique packet *id*

  problems:
  - arithmetic overflow
  - node restarts

- Determine upper and lower bounds on generation time
  - Use forward and backward reasoning

- Remove non-compliant packets
  - Duplicated packets
  - Incorrect time information
  - Behavior not covered by formal model

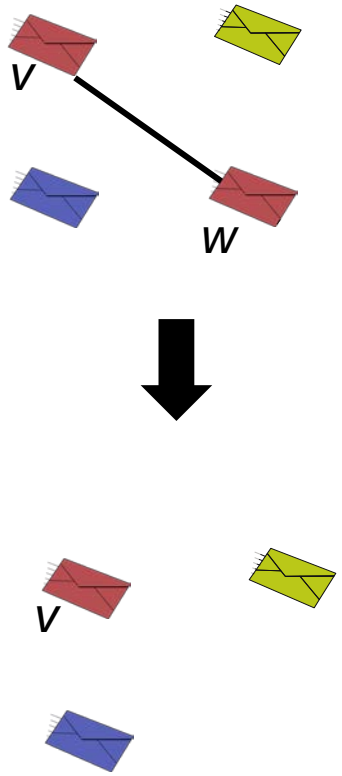
  problems:
  - clock drift
  - node restarts

Bounds on Packet Generation Time

- Worst-case bounds for a single packet
  - $t_g(i) \in [t_g^l(i), t_g^u(i)]$
  - $t_g^l(i) := t_b(i) - \frac{\tilde{t}_s(i) + \hat{h} \cdot \hat{t}_u}{1 - \hat{\rho}}$
  - $t_g^u(i) := t_b(i) - \frac{\tilde{t}_s(i)}{1 + \hat{\rho}}$
- Forward and backward reasoning is applied to tighten these bounds
  - Requirement: Exact ordering information

Duplicate Filtering

- We consider packets with
  - the same source address o
  - the same sequence number s
  - an equal payload p

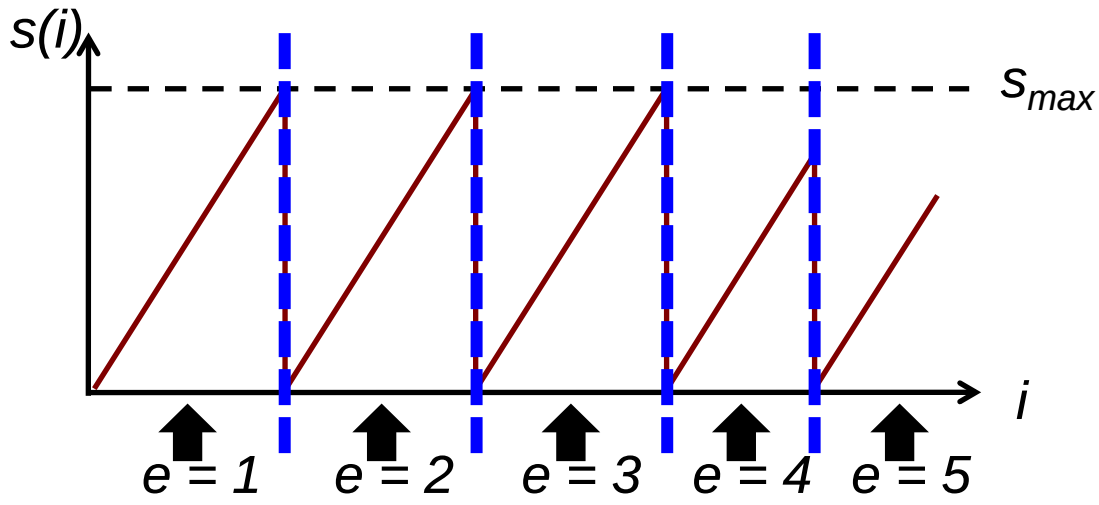


- We construct a graph *G* = *(V, E)*
  - $(v, w) \in E \Leftrightarrow (t_g^u(v) \geq t_g^l(w)) \wedge (t_g^u(w) \geq t_g^l(v))$

- Duplicate-free data set is achieved by only considering packets that are within the maximum independent set of *G*

Separate Data into Epochs

- Observation: Sequence number *s(i)* resets to zero
  - Every s_{max} packets due to arithmetical overflow
  - After a cold restart due to loss of state



- After epoch assignment: $id(i) = e(i) \cdot s_{\max} + s(i)$
  - *k* "generated before" *l* ⇔ *id(k) < id(l)*

Epoch Assignment (1/3)

- For each packet, calculate a reference point
  - $T_C(i) = \tilde{t}_g(i) - s(i) \cdot T$
- Ideal case: Perfect clocks, absence of node restarts
  - Packets belonging to the same epoch have an equal reference point T_C
- Real case: Imperfect clocks, node restarts
  - Epoch assignment based on bound ΔT_C

Epoch Assignment (2/3)

Theorem 1

All packets *k, l* that belong to the same epoch, *i.e.*, *e(k)* = *e(l)*, satisfy

$$|T_c(k) - T_c(l)| \leq \Delta T_c$$

where

$$\Delta T_c = (s_{\max} - 1)(\hat{\rho} T + T - T') + T' + 2\hat{\rho} \cdot t_s^{\max}$$

where $t_s^{\max}$ is an upper bound on the network sojourn time, *i.e.*, $t_s(k) \leq t_s^{\max}$ and

$$T' = \frac{1}{\frac{1+\hat{\rho}}{T} + \frac{1}{t_{\text{reset}}}}$$

Epoch Assignment (3/3)

Theorem 2

Suppose that the generation period T satisfies

$$T > 2(1+\hat{\rho})\frac{\Delta T_c}{s_{\max}}$$

Then all packets *k, l* that belong to different epochs, *e.g.*, *e(k)* < *e(l)*, satisfy

$$T_c(l) - T_c(k) > \Delta T_c$$

Where ΔT_c s defined in Theorem 1.

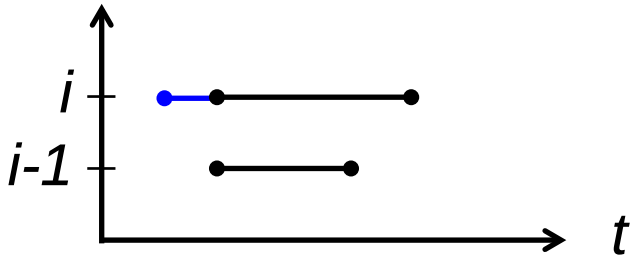
Forward and Backward Reasoning

- Initially set worst-case bounds $t_g(i) \in [t_g^l(i), t_g^u(i)]$ are often too pessimistic

- Given the correct order of packet generation, initially set bounds can be improved by using information from temporarily adjacent packets

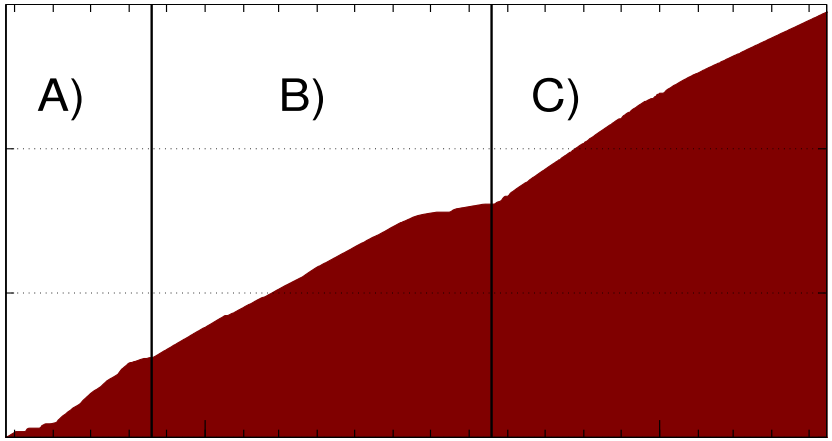
- Example: A packet cannot be generated earlier than its predecessor

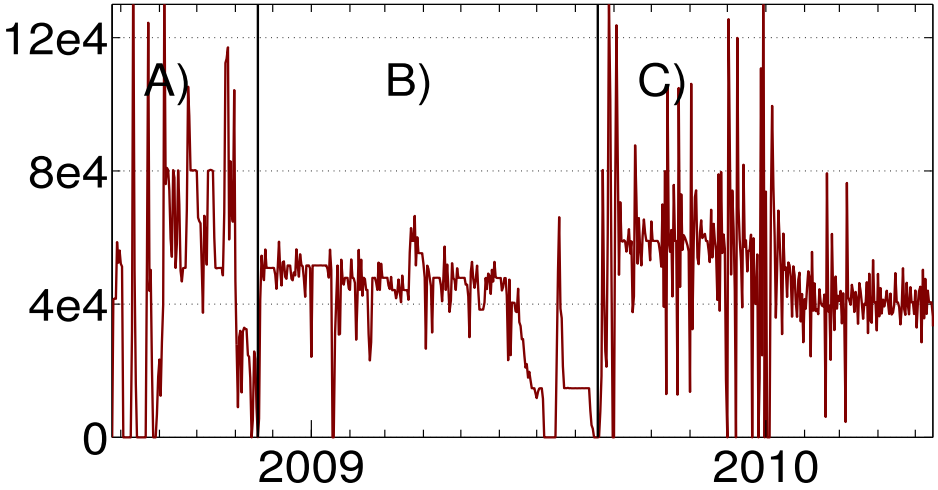
$$t_g^l(i) := \max\left(t_g^l(i), t_g^l(i-1)\right)$$



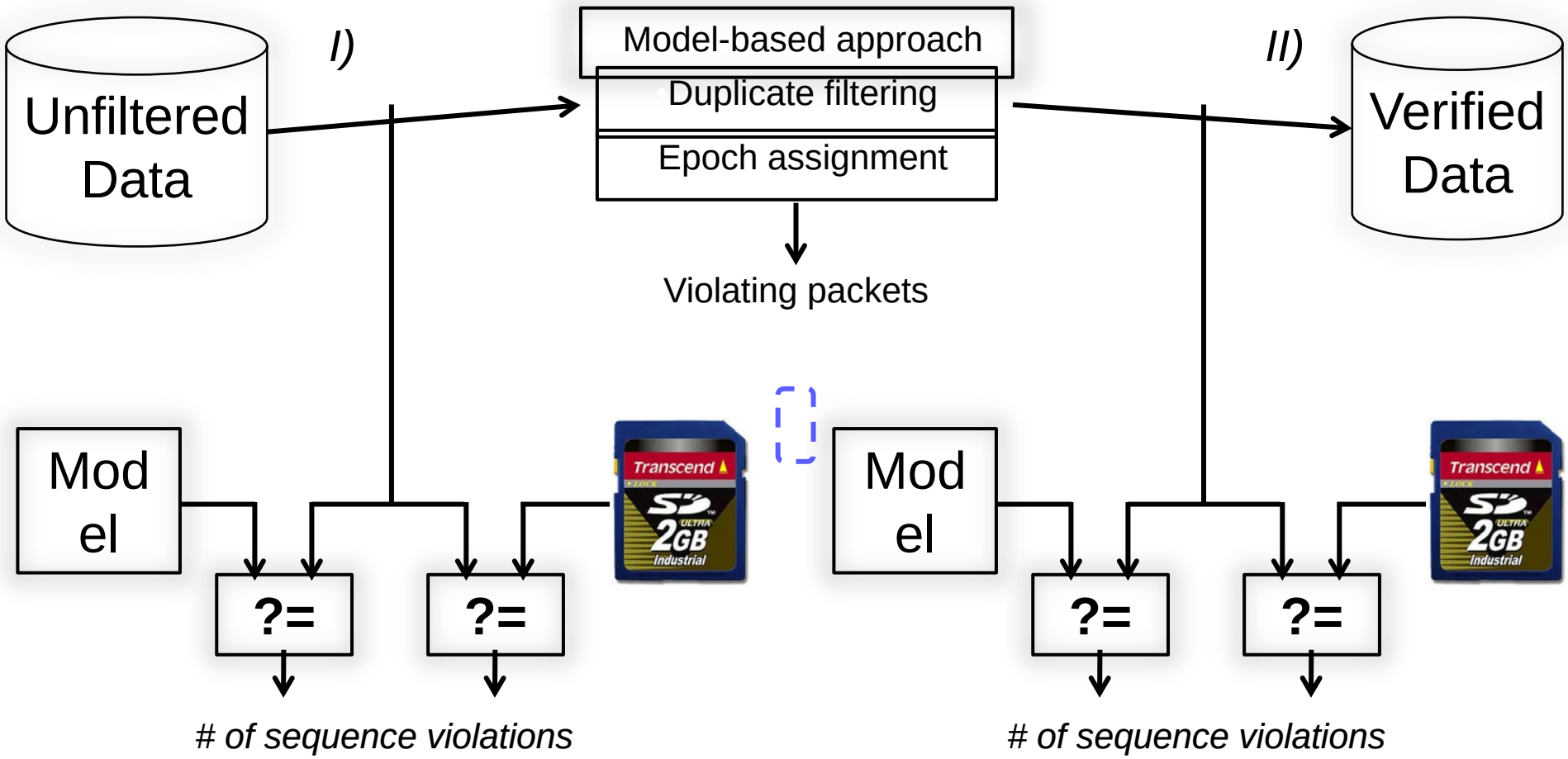
Matterhorn Deployment Data

- Three phases of system operation
  A. Initial difficulties with hardware and software
     - Non-conforming system operation
  B. Sensor nodes subject to a high number of restarts
  C. Daily shut down of base station due to insufficient energy





Model Validation (1/2)



Model Validation (2/2)

Counter	A) Jul 08-Nov 08	B) Nov 08-Aug 09	C) Sep 09-May 10
Total packets	1,064,884 (100.0%)	2,180,684 (100.0%)	2,703,998 (100.0%)
I) Unfiltered data set			
Sequence violations (Signal model)	189,645 (17.8%)	65,839 (3.0%)	46,987 (1.7%)
Sequence violations (Ext. storage)	*n/a*	69,004 (3.2%)	47,253 (1.7%)
II) Model-based approach			
Discarded packets	432,826 (40.6%)	69,829 (3.2%)	124,554 (4.6%)
Packet duplicates	4,020 (0.4%)	69,422 (3.2%)	44,601 (1.7%)
$t_s(i) > t_s^{\max}$	0 (0.0%)	0 (0.0%)	0 (0.0%)
Failed epoch assignment	235,927 (22.2%)	277 (0.0%)	2,466 (0.1%)
Invalid interval $t_g^{\mu,l}(i)$	192,879 (18.1%)	130 (0.0%)	77,487 (2.9%)
Accepted packets	632,058 (59.4%)	2,110,855 (96.8%)	2,579,444 (95.4%)
Sequence violations (Signal model)	7 (0.0%)	11 (0.0%)	0 (0.0%)
Sequence violations (Ext. storage)	*n/a*	1 (0.0%)	0 (0.0%)

- Previously “dirty” data set has been restored for use
- Appropriate method for continuous system validation

Conclusions

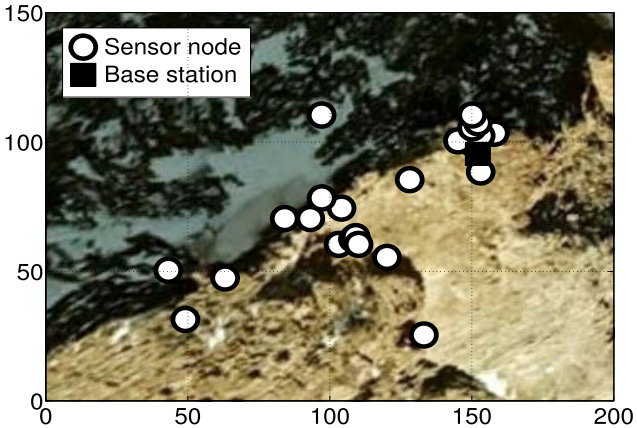
- Data integrity testing and order reconstruction based on a system model of a real system

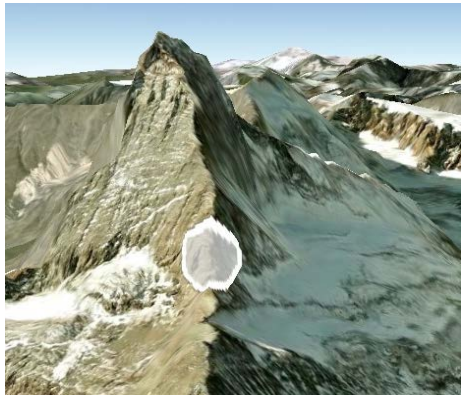
- Give guarantees on data quality
  - Duplicate-free data
  - Correct temporal order of generation
  - Correct logical ordering

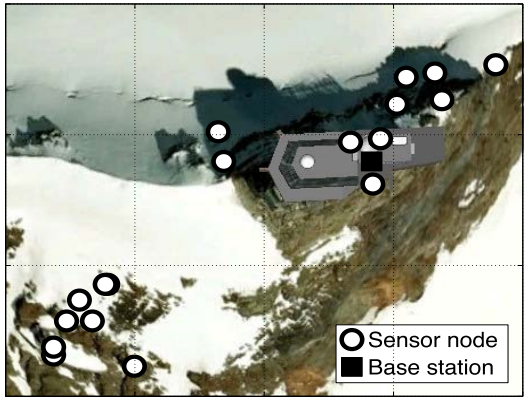
- Proposed intermediate packet filtering step facilitates the usage of wireless sensor networks for applications that require highest data quality

[Matthias Keller, Lothar Thiele, Jan Beutel: Reconstruction of the Correct Temporal Order of Sensor Network Data, IPSN 2011, April 2011, pp. 282-293]

Case Study Deployments







Long-term Data Quality

	Matterhorn	Jungfraujoch	Thur
Period	Jun 2010 – May 2011	Jan 2010 – May 2011	Mar 2010 – May 2011
Total packets	29,275,461 (100.0%)	27,723,571 (100.0%)	7,189,877 (100.0%)
Accepted packets	27,786,712 (94.9%)	26,026,120 (93.9%)	7,105,879 (98.8%)
Packet duplicates	1,462,213 (5.0%)	1,472,664 (5.3%)	83,992 (1.2%)
Violating packets	26,536 (0.1%)	224,787 (0.8%)	6 (0.0%)

PermaDozer Performance Analysis

Deployment	Matterhorn			Jungfraujoch			Thur		
# of samples	10,347,562			4,805,958			2,785,221		
Link class	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$	**# of links**	$\bar{\sigma}_R$	$\max \sigma_R$
Bidirectional communication -- downlink									
Good links (RSSI > -65 dBm)	143 (33.7%)	2.1	7.6	14 (20.6%)	6.3	9.4	20 (31.8%)	3.4	5.0
Med. links (-65 > RSSI > -85 dBm)	208 (49.1%)	2.9	16.6	40 (58.8%)	3.0	7.0	41 (65.1%)	3.2	9.9
Bad links (RSSI ≤ -85 dBm)	73 (17.2%)	1.4	4.3	14 (20.6%)	1.5	2.4	2 (3.2%)	2.2	2.8
Bidirectional communication -- uplink									
Good links (RSSI > -65 dBm)	149 (35.1%)	2.1	7.3	14 (20.6%)	5.8	8.6	17 (27.0%)	3.5	5.6
Med. links (-65 > RSSI > -85 dBm)	201 (47.4%)	3.1	15.5	38 (55.9%)	3.0	6.7	44 (69.8%)	3.3	12.6
Bad links (RSSI ≤ -85 dBm)	74 (17.5%)	1.7	5.7	16 (23.5%)	1.9	2.5	2 (3.2%)	1.7	2.2

Deployment	Matterhorn	Jungfraujoch	Thur
Packet counters, mean per node			
Unique	1,117,659	1,117,338	915,903
Missing	667	5,368	29
Duplicates	59,333 (5.0%)	85,901 (7.1%)	11,293 (1.2%)
Data yield			
∅	99.94%	99.53%	100.00%
min	99.88%	96.28%	99.99%
max	100.00%	100.00%	100.00%
Radio duty cycle			
max	0.26%	1.31%	0.12%
Median	0.14%	0.34%	0.07%

- The received signal strength indicator (RSSI) is measured for every successful reception of a packet
- Ratio between signal strength and noise floor
- Higher ratio of duplicates at more challenging environments at Matterhorn and Jungfraujoch

[Matthias Keller, Matthias Woehrle, Roman Lim, Jan Beutel, Lothar Thiele: Comparative Performance Analysis of the PermaDozer Protocol in Diverse Deployments, SenseApp 2011, October 2011, accepted for publication]

PermaSense Demo, Q/A

- ETH Zurich
  - Computer Engineering and Networks Lab
  - Geodesy and Geodynamics Lab
- University of Zurich
  - Department of Geography
- EPFL
  - Distributed Information Systems Laboratory
- University of Basel
  - Department Computer Science

Interested in more?

http://www.permasense.ch