Networked Embedded Systems

Part 1: Designing Predictable and Efficient Networked Embedded Systems

Part 2: Process Chain for End-to-End Sensing in Disruptive Environments

Lothar Thiele

Jan Beutel

ETH Zurich

Networked Embedded Systems

Part 1: Designing Predictable and Efficient Networked Embedded Systems

Part 2: Process Chain for End-to-End Sensing in Disruptive Environments

Lothar Thiele

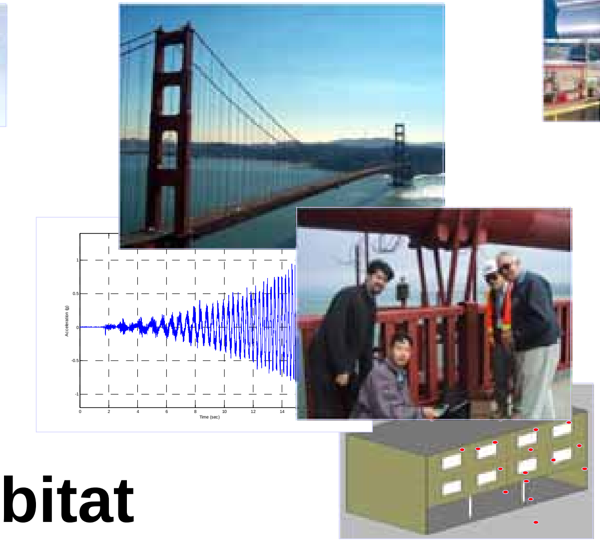
Jan Beutel

ETH Zurich

Networked Embedded Systems - Application Visions

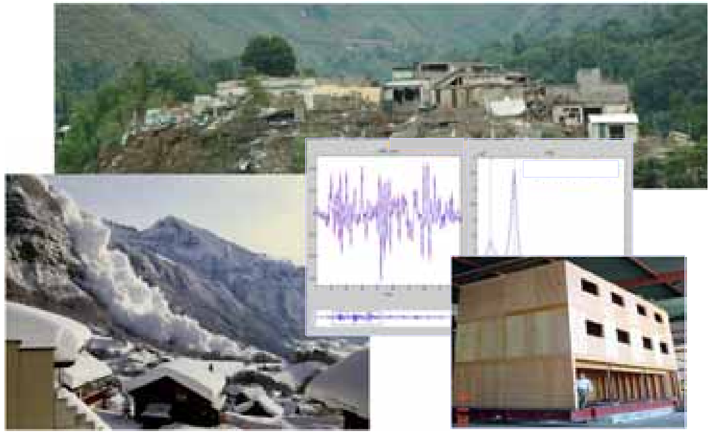
Forest

Maintenance



Factory Automation

Natural Hazards

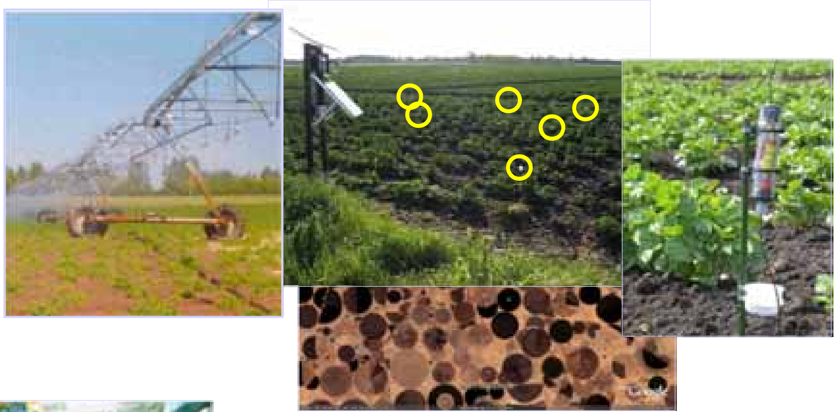


Animal Habitat

Networked Embedded Systems - Application Visions

Building Automation

Precision Agriculture



Health Care

Logistics

Security Systems

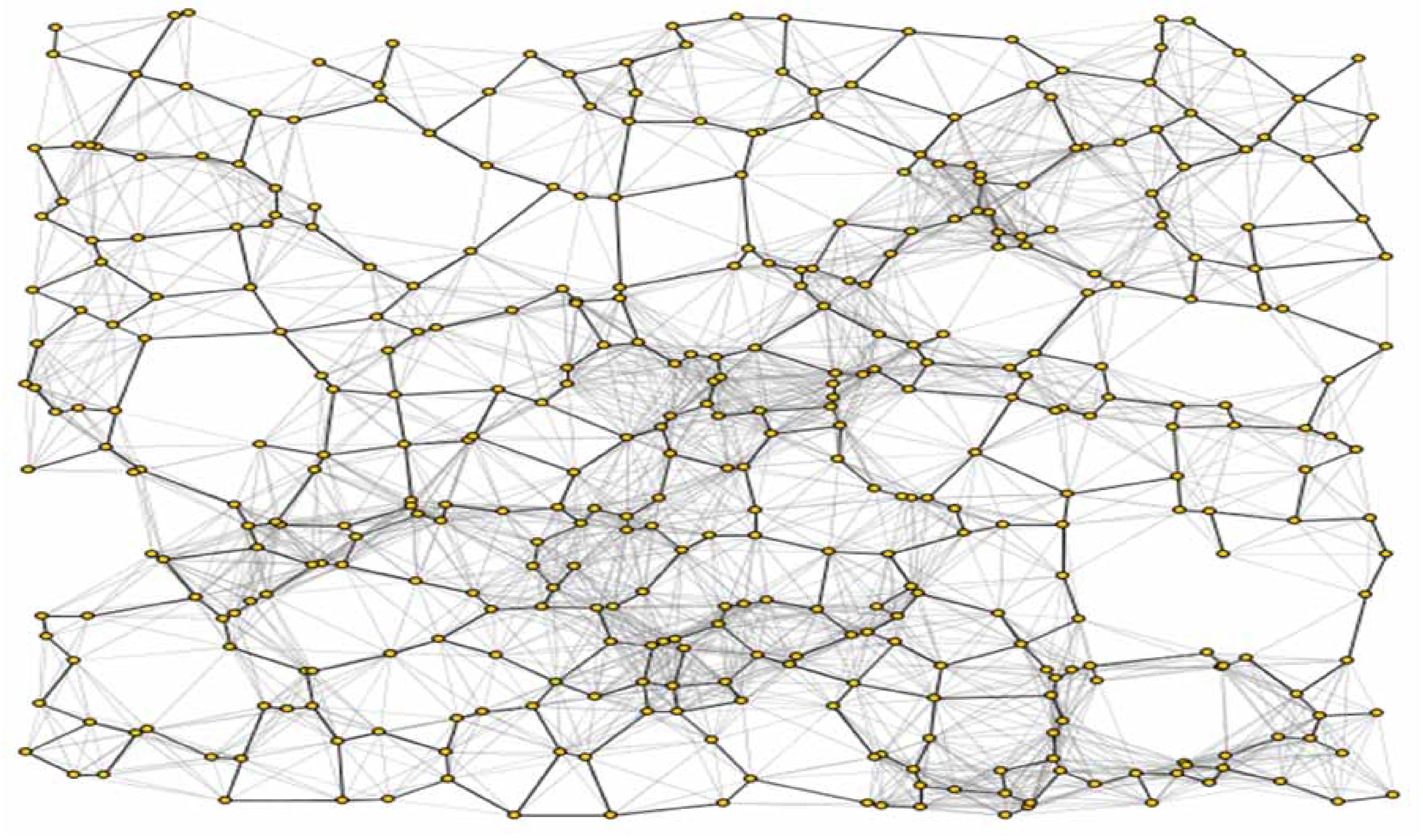
Social Interactions

Some of our Deployments

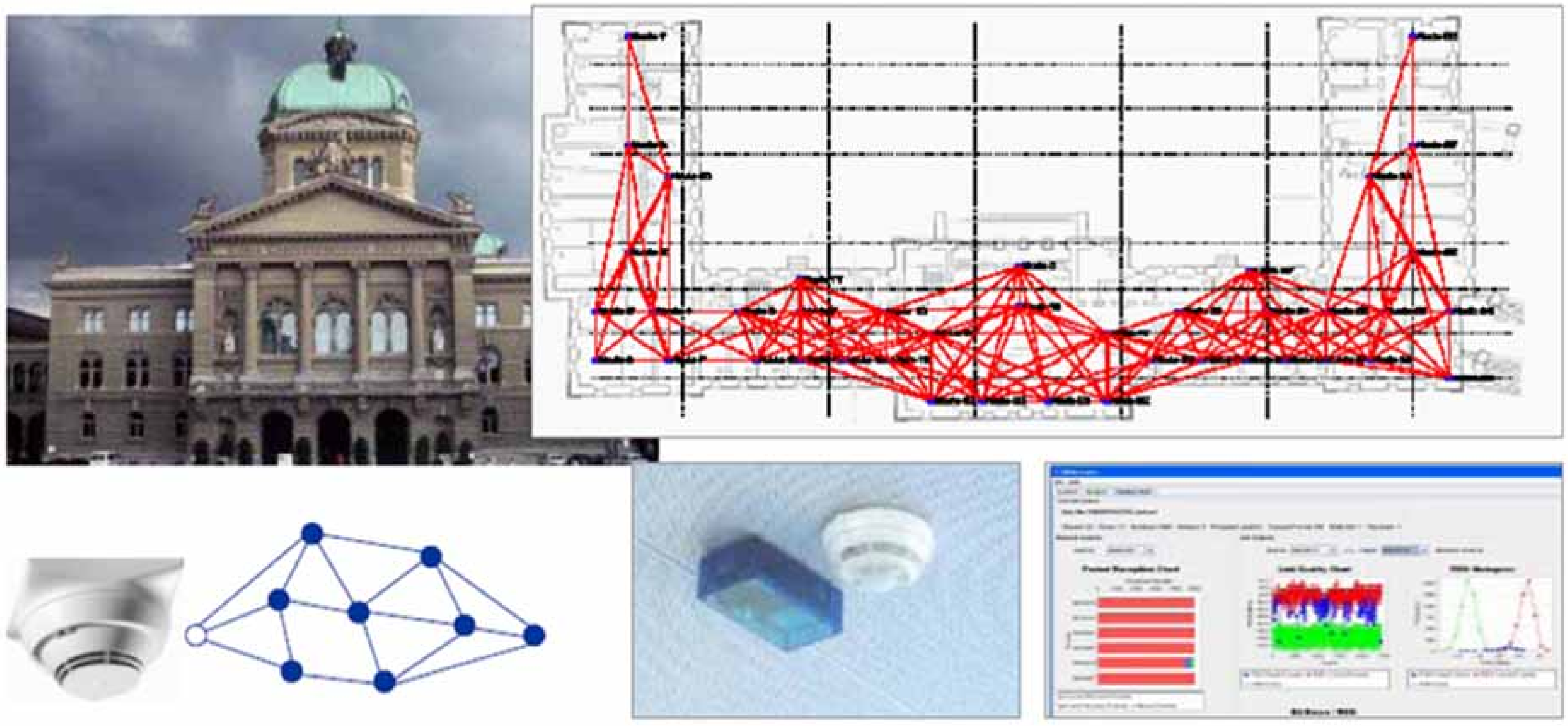
- ***Sensor Networks for Environmental Monitoring***
  - Sensor networks in extreme environments
  - Permafrost in alpine regions
  - Early warning scenarios
  - Sensing environmental quality in cities

- ***Safety, Security and Comfort***
  - Safety and Security Networks (Siemens Building Technologies)
  - Sensor networks for building diagnosis

The Networked Embedded System Vision



Safety/Security Monitoring I



Siemens Building Technologies

Safety/Security Monitoring II

First certified product on the market, based on multi-hop networking.

Hard real-time and dependability constraints.

Battery operated.

Based on innovative low power MAC and routing protocols.

PermaSense I

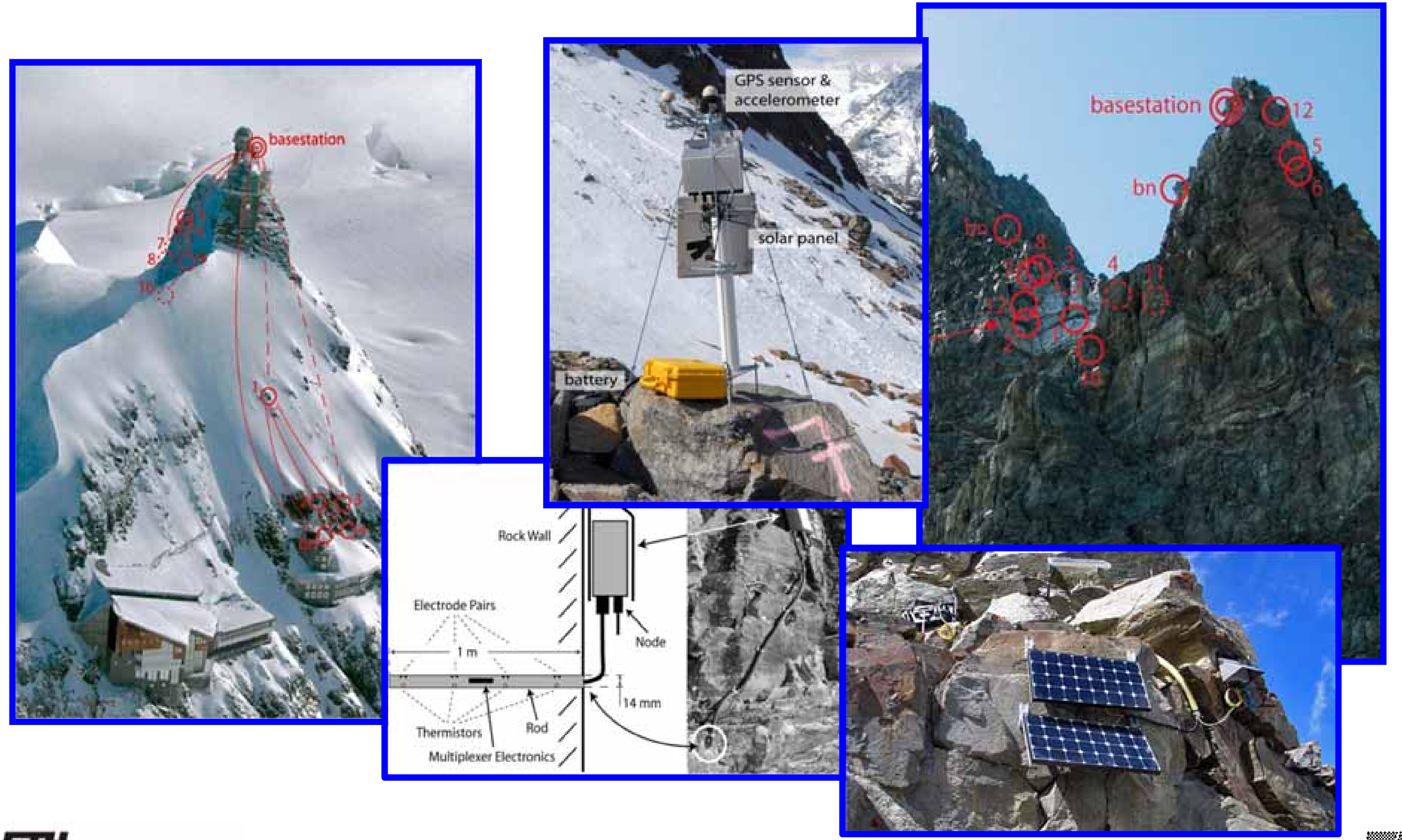
We aim to:

- provide ***long-term high-quality*** sensing in ***harsh environments***
- obtain measurements that have ***previously been impossible*** (high resolution in time and space)

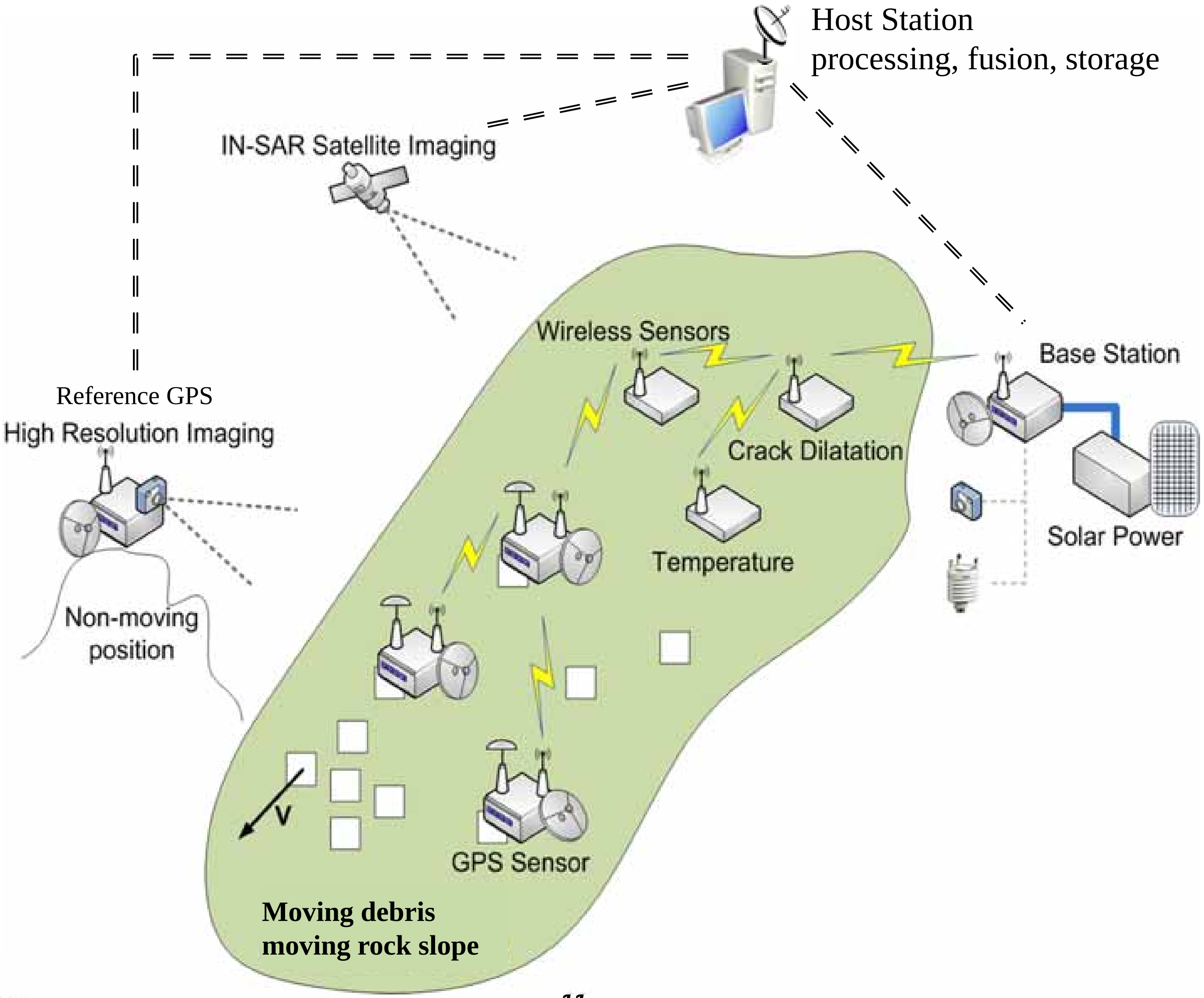
Reliability, delivery of information in ***near real-time***, and integration of ***diverse sensors*** are ingredients

- for ***quantifing environmental phenomena*** and
- for the next generation of ***early-warning systems***.

PermaSense II



PermaSense III



OpenSense I

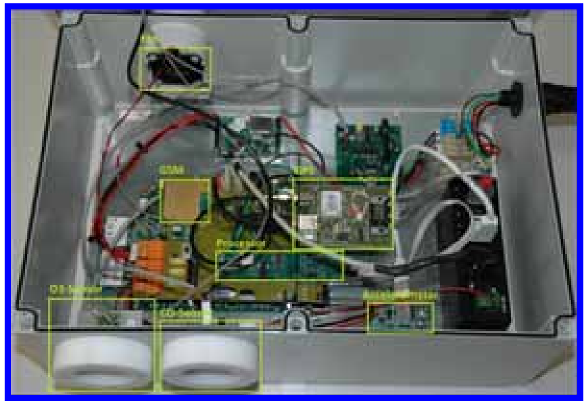
- *Environmental quality* in urban areas is a global concern.

- *Officials*
  - environmental engineers: location of pollution sources
  - municipalities: creating incentives to reduce environmental footprint
  - public health studies
- *Citizens*
  - advice for outside activities
  - assessment of long-term exposure
  - pollution maps

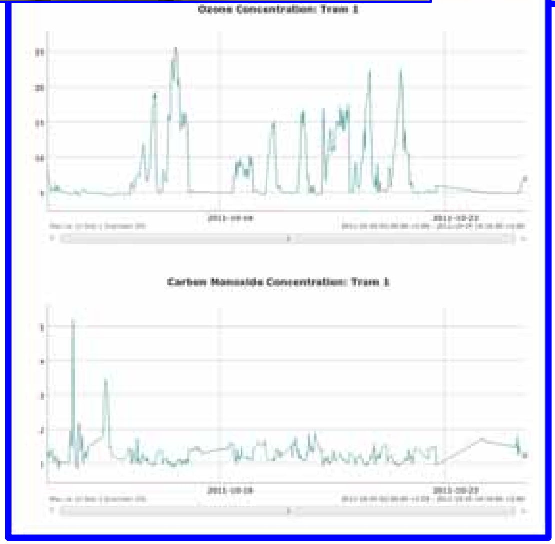
OpenSense II

route selection

end-to-end processing



system design



modeling
classification
calibration

Understand the System

Data Pipeline
Cyber-Physical Interactions
Application
Users

Promises

Sensor nodes are cheap, so we can have plenty of them.

Nodes may be cheap, but deployment and maintenance is expensive.

Additional redundant nodes make the system fault tolerant automatically.

More nodes make the system more fragile.

Why is it so difficult to design efficient and predictable networked embedded systems?

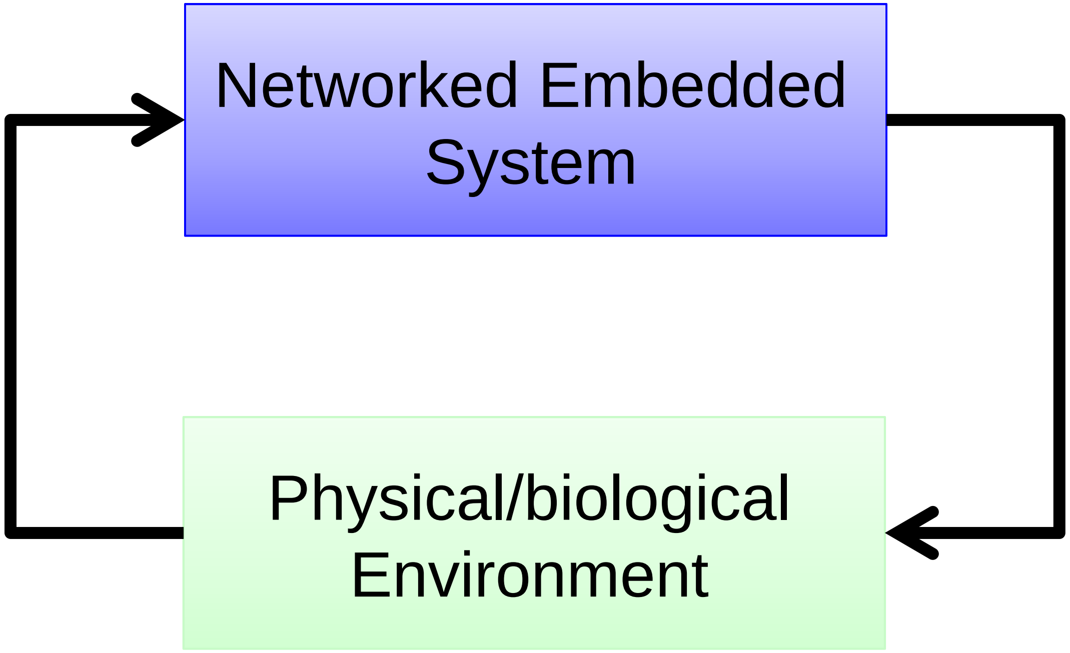
In Computer Science, the physical world has been (successfully) abstracted away from 'computation'.

In Computer Science, the physical world has been (successfully) abstracted away from ‘computation’.

But embedded systems are closely integrated into their environment.

Interacting with the Environment

Functional Interaction



Networked Embedded System

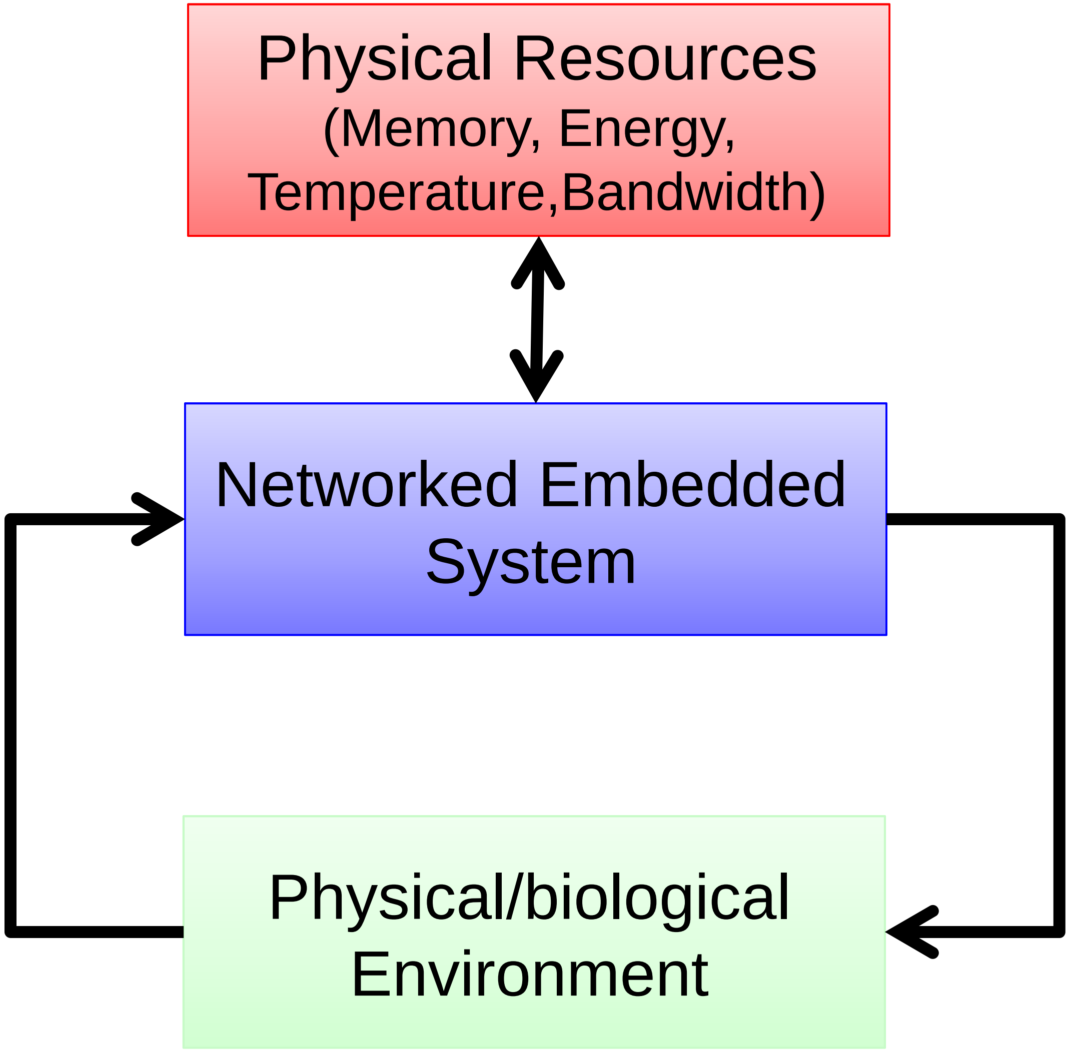
Physical/biological Environment

In Computer Science, the physical world has been (successfully) abstracted away from ‘computation’.

But embedded systems are closely integrated into their environment.

Working close to resource limits (energy, memory, bandwidth) makes systems extremely fragile.

Interacting with the Environment



Resource Interaction

Physical Resources
(Memory, Energy, Temperature,Bandwidth)

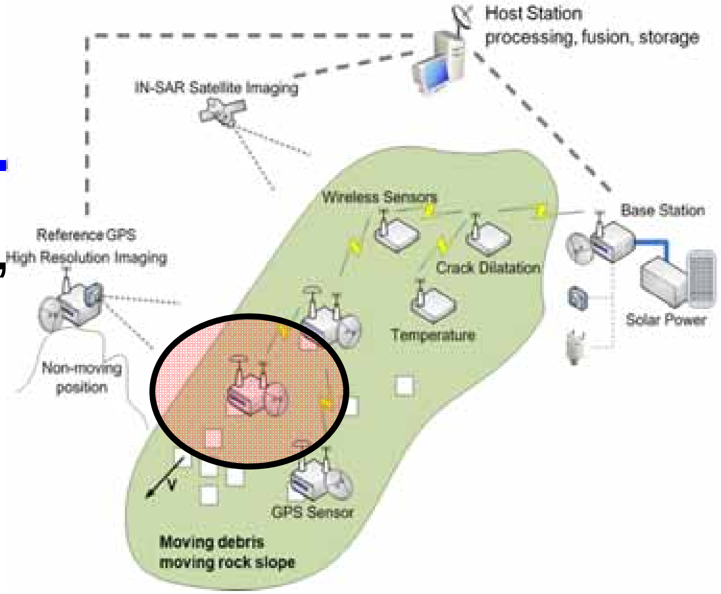
Networked Embedded System

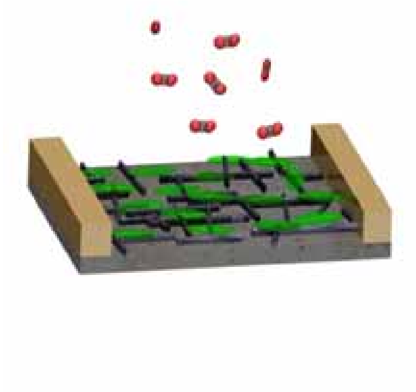
Functional Interaction

Physical/biological Environment

Challenges

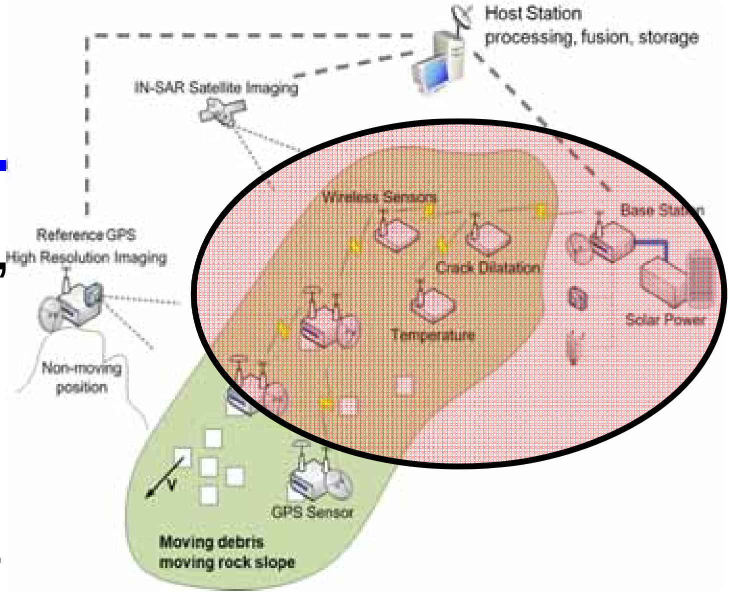
- ***Sensors and interfaces*** (air pollution, sensitivity, calibration, low energy, low cost, integration, …)

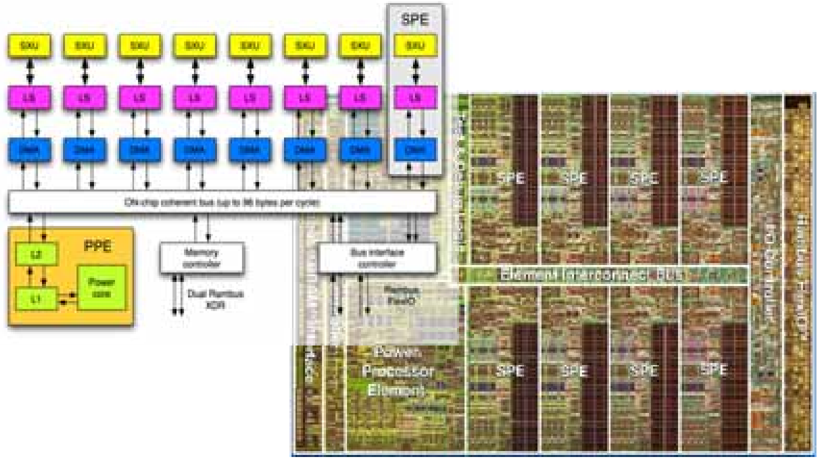




Challenges

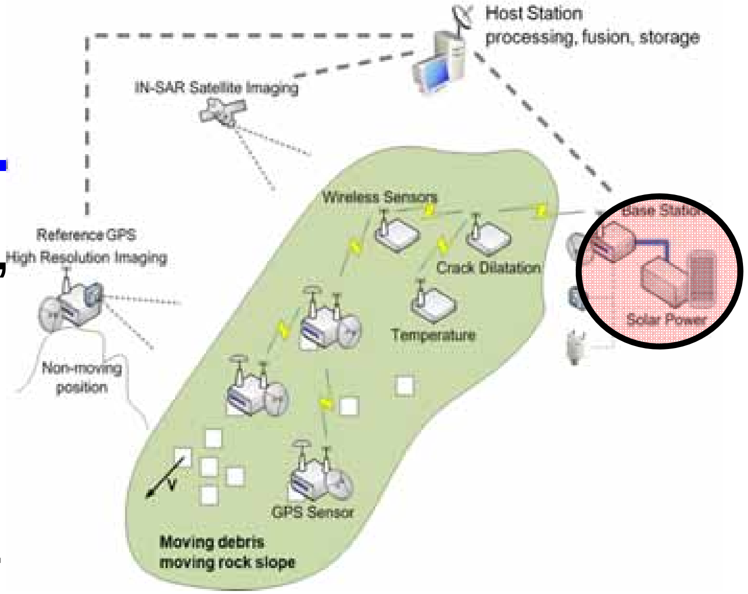
- ***Sensors and interfaces*** (air pollution, sensitivity, calibration, low energy, low cost, integration, …)
- **Low energy operation** (technology integration, parallel processing, power management, …)

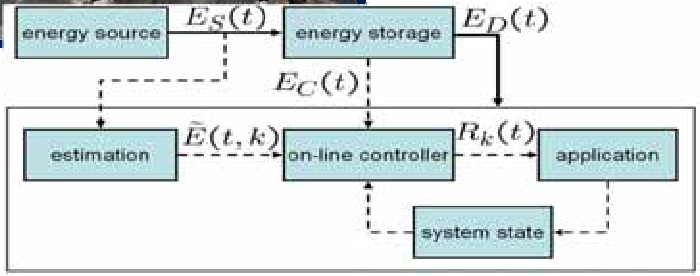




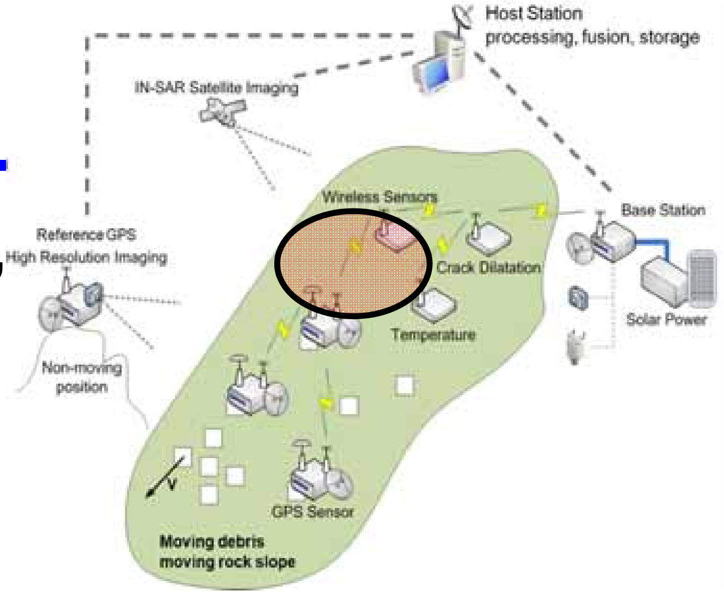
Challenges

- ***Sensors and interfaces*** (air pollution, sensitivity, calibration, low energy, low cost, integration, …)
- **Low energy operation** (technology integration, parallel processing, power management, …)
- ***Energy harvesting*** (application control, …)

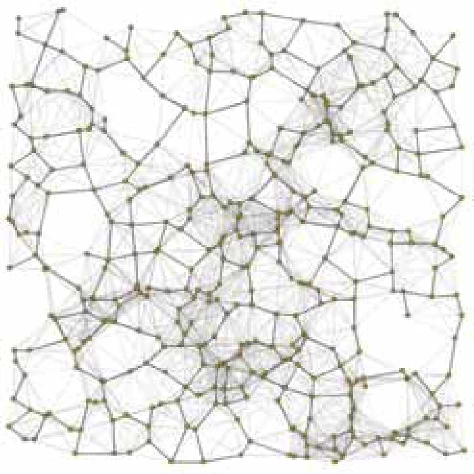




Challenges

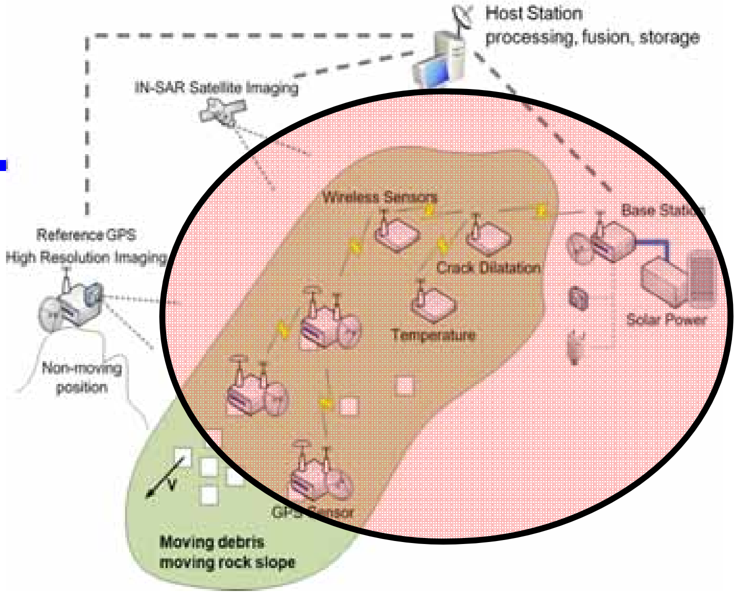


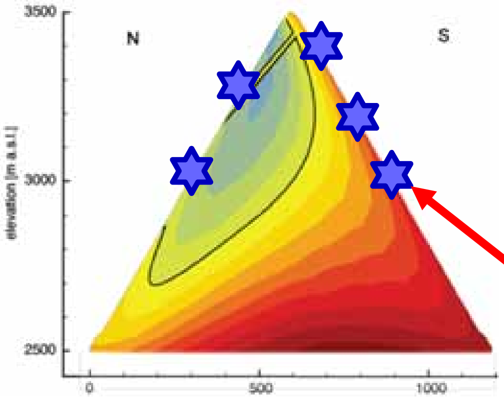
- ***Sensors and interfaces*** (air pollution, sensitivity, calibration, low energy, low cost, integration, …)
- **Low energy operation** (technology integration, parallel processing, power management, …)
- ***Energy harvesting*** (application control, …)
- ***(Wireless) networking*** (capacity, ad-hoc networking, low power, new communication principles, mobility, …)



Challenges

- ***Distributed control*** (sensor-actuator coupling, energy balancing, …)

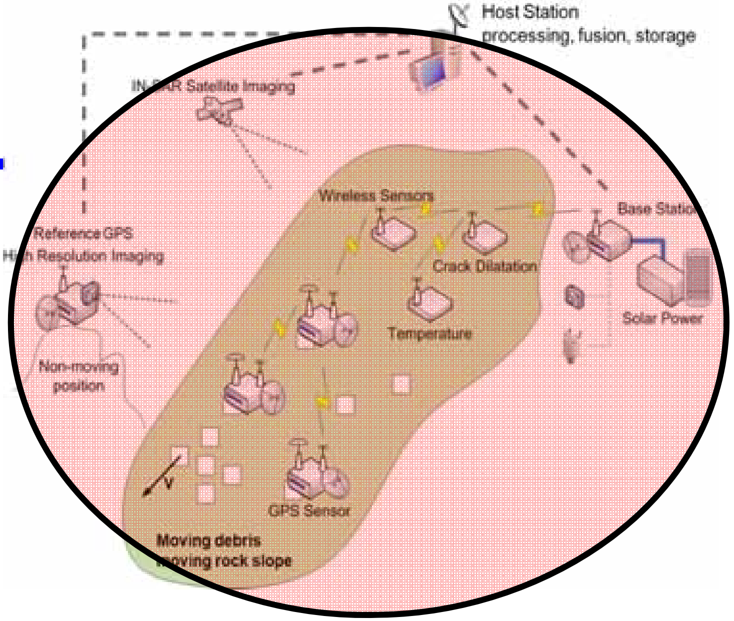


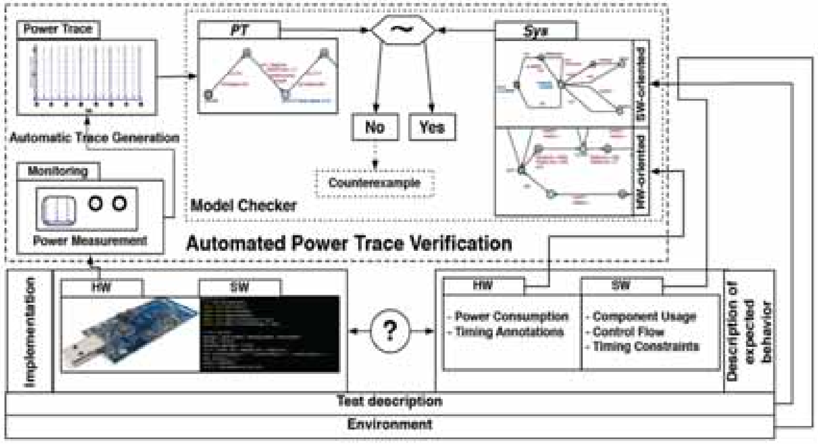


actuation and sensing

Challenges

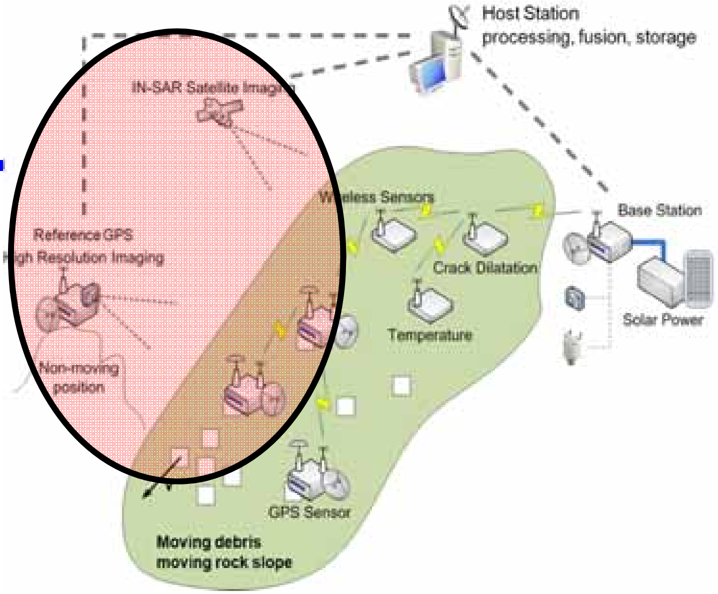
- ***Distributed control*** (sensor-actuator coupling, energy balancing, …)
- ***Predictability and reliability*** (formal verification, (energy) testing, observability…)

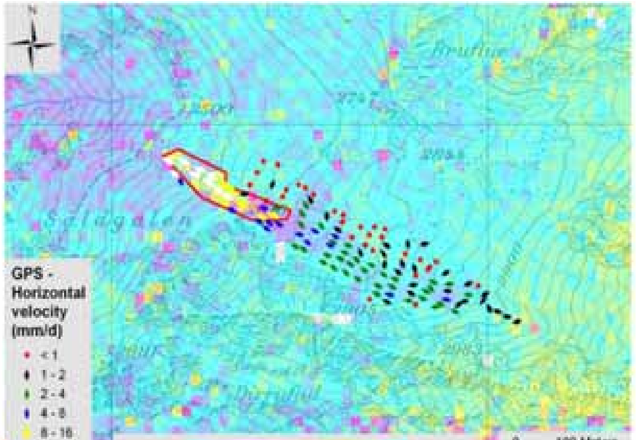


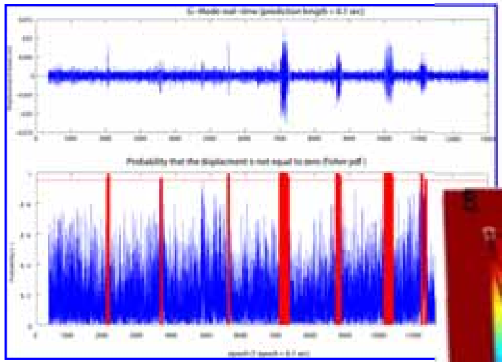


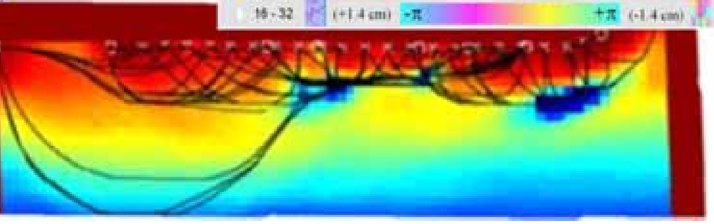
Challenges

- ***Distributed control*** (sensor-actuator coupling, energy balancing, …)
- ***Predictability and reliability*** (formal verification, (energy) testing, observability…)
- ***Modeling*** (new models and methods for data analysis, sensing at diverse spatial and temporal scales, data fusion)









Predictability and Efficiency

⇓

Trustworthy Systems

Contents

- ***Part 1***
  - *Energy Harvesting*
  - Power Testing
  - Predictable Communication Protocols

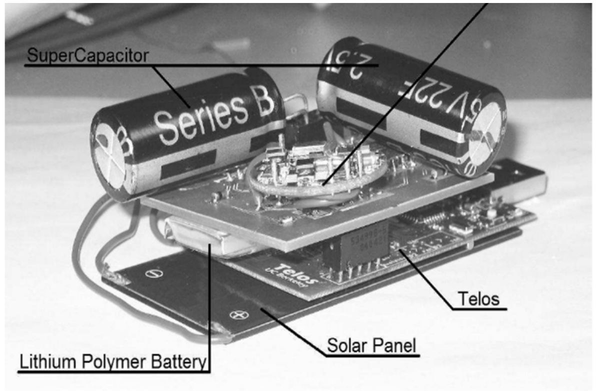
- ***Part 2***
  - PermSense System Design
  - Test & Validation Infrastructure
  - Data Cleaning & Validation
  - Calibration of Mobile Sensors
  - Complex Sensing

Energy Harvesting

- how automatic control can help -

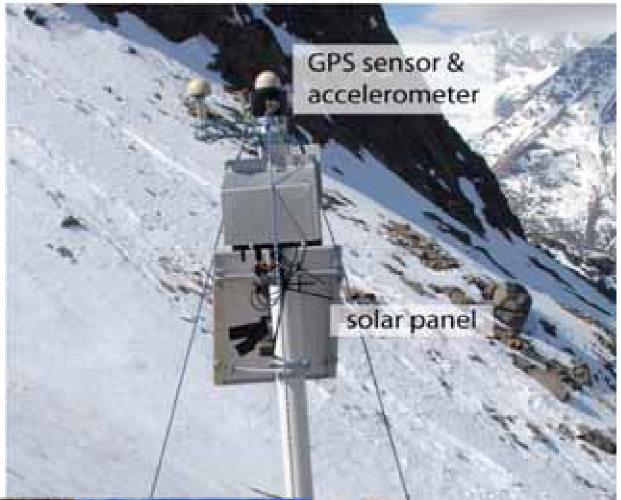
Energy Harvesting

[Prometheus: Culler]

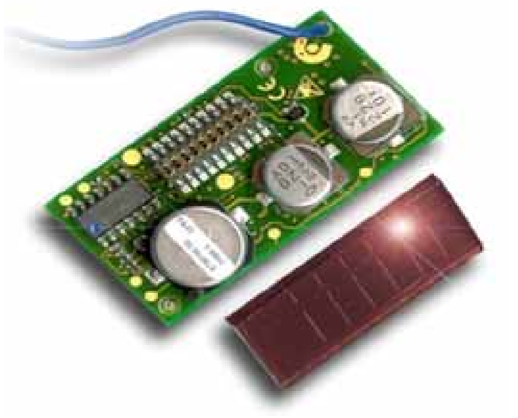


[Heliomote: Srivastava]

[PermaSense]



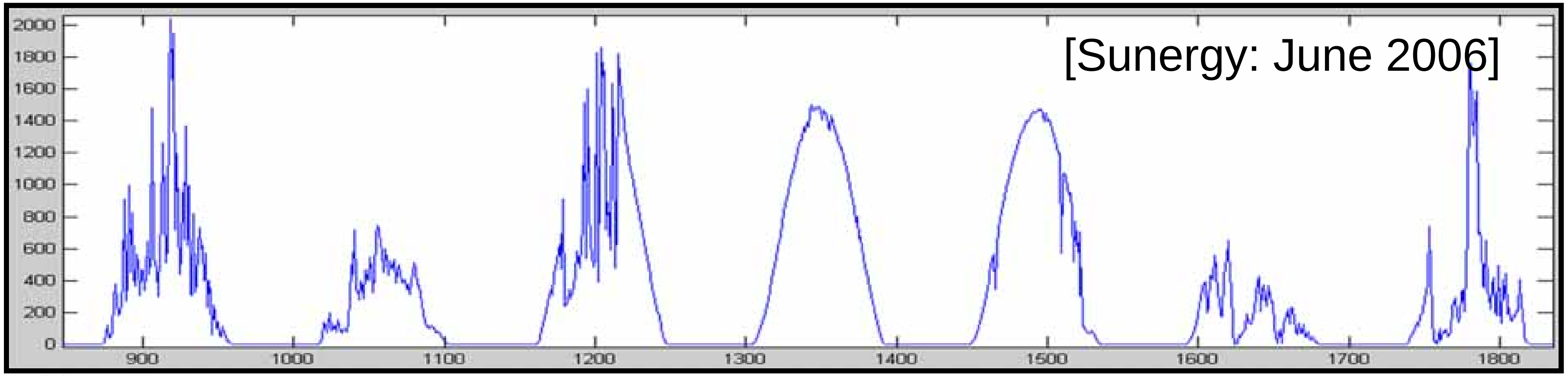
[STM100: enocean.com]



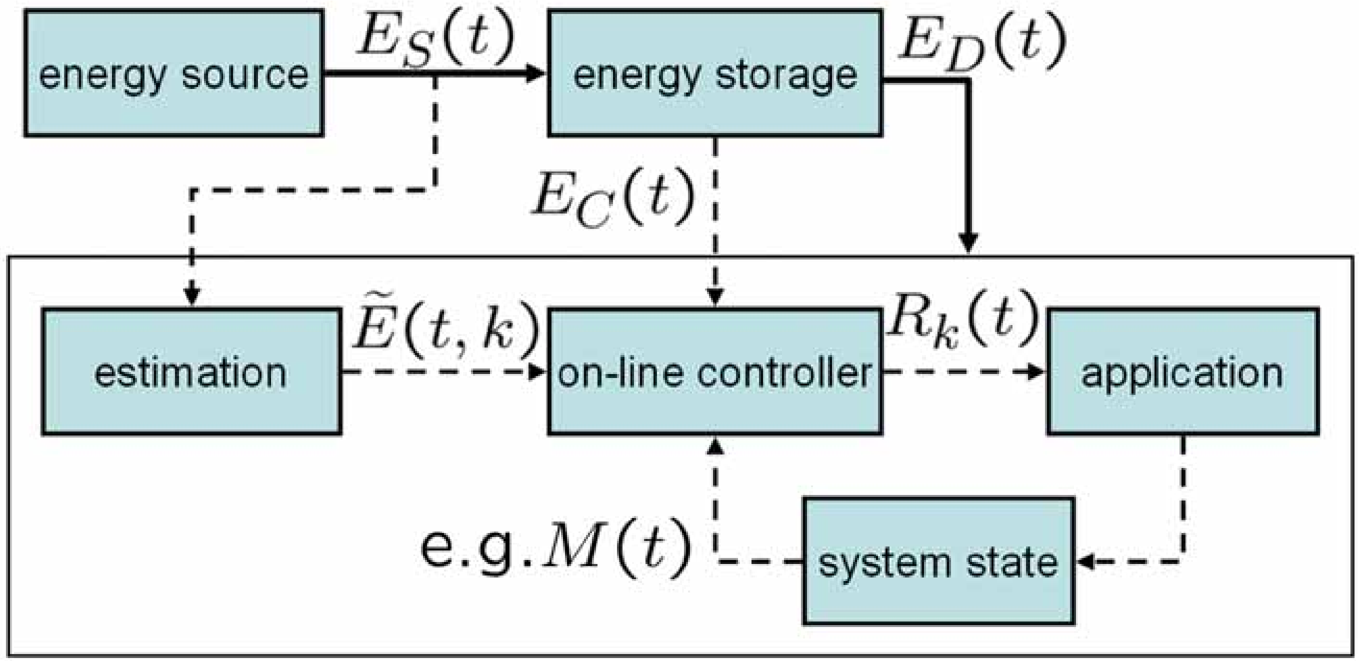
Problem Statement

When should we use energy ?

- Given a single embedded system. How do we
  - … avoid energy underflows?
  - … avoid energy overflows?
  - … dimension the battery?
  - ... deal with real-time constraints?
  - ... cope with resource-constraint systems ?

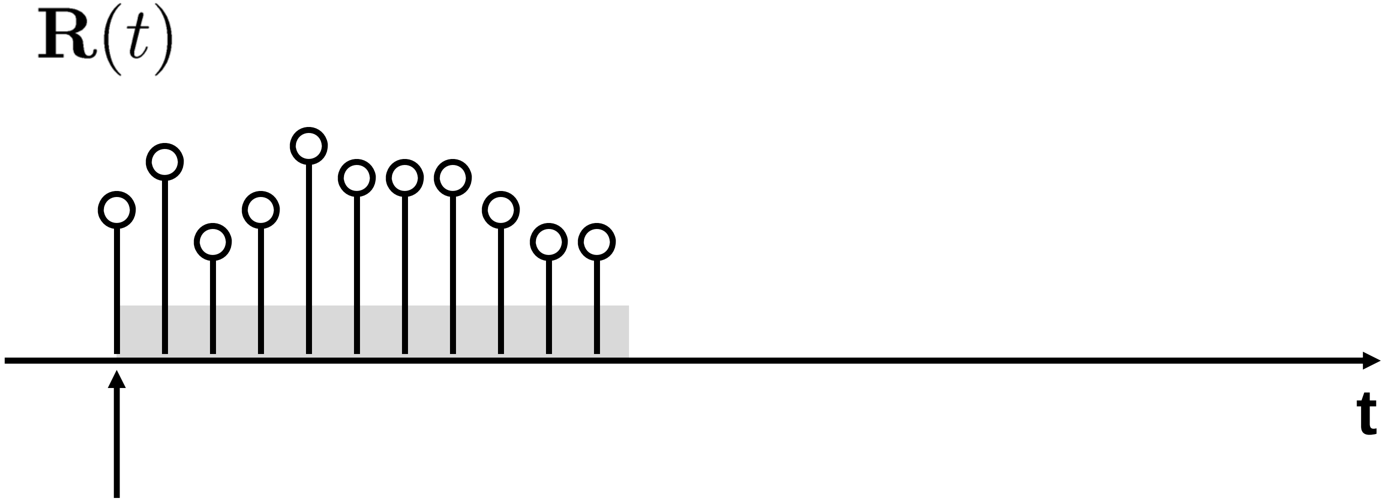


System Model



System Model

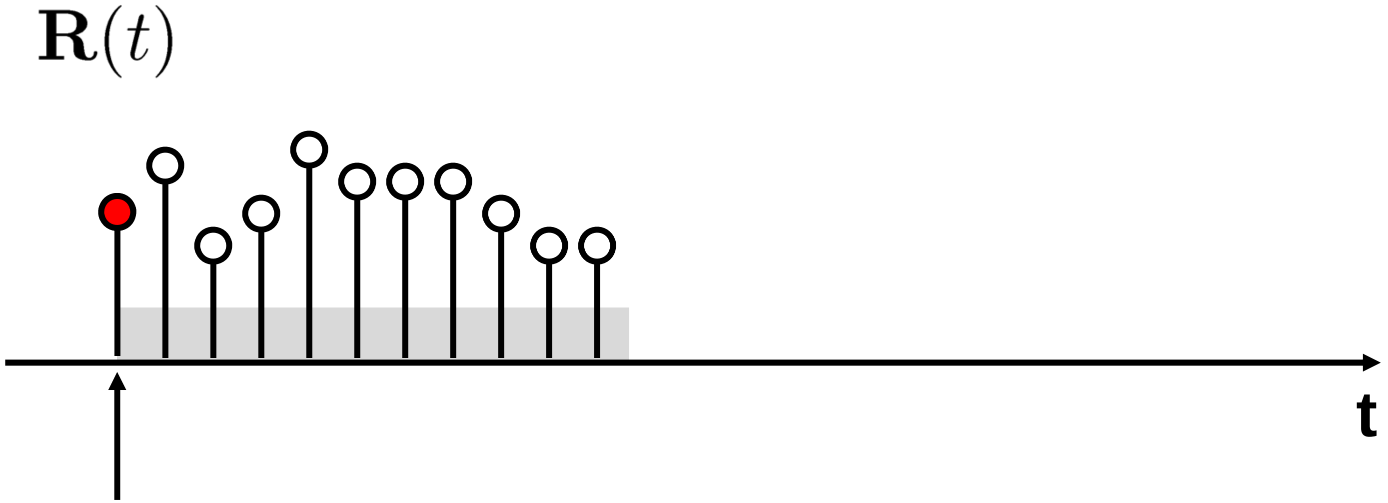
- Optimization problem: finite horizon control



current time *t*
current state (memory, battery, …)
current environment (input power)

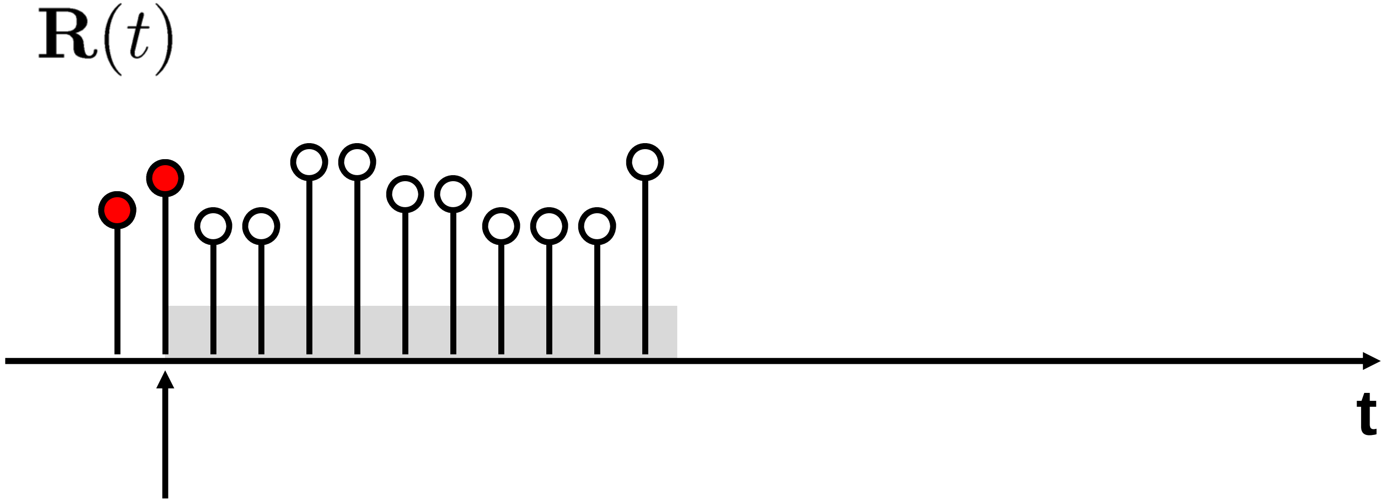
System Model

- Optimization problem: finite horizon control



System Model

- Optimization problem: finite horizon control



System Model

- Linear program (LP) specification (example)

$$\text{maximize } \lambda$$

Maximize long-term utilization

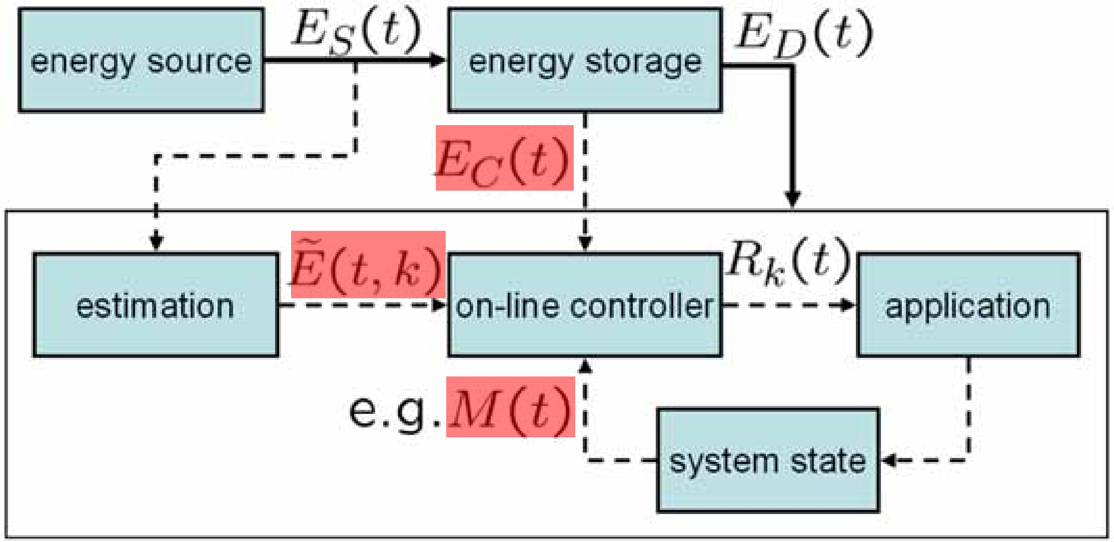
$$\mathbf{R}\,(t+k\cdot L)\geq\lambda \quad \forall 0\leq k<N$$

$$E_C(t+k\cdot L)=E_C(t)+\sum_{j=0}^{k-1}(\tilde{E}(t,j)-\mathbf{E}^T\cdot\mathbf{R}(t+j\cdot L)) \quad \forall 0\leq k\leq N$$

$$M(t+k\cdot L)=M(t)+\sum_{j=0}^{k-1}\mathbf{M}^T\cdot\mathbf{R}(t+j\cdot L) \quad \forall 0\leq k\leq N$$

$$0\leq E_C(t+k\cdot L)\leq E_{max}$$

$$0\leq M(t+k\cdot L)\leq M_{max}$$

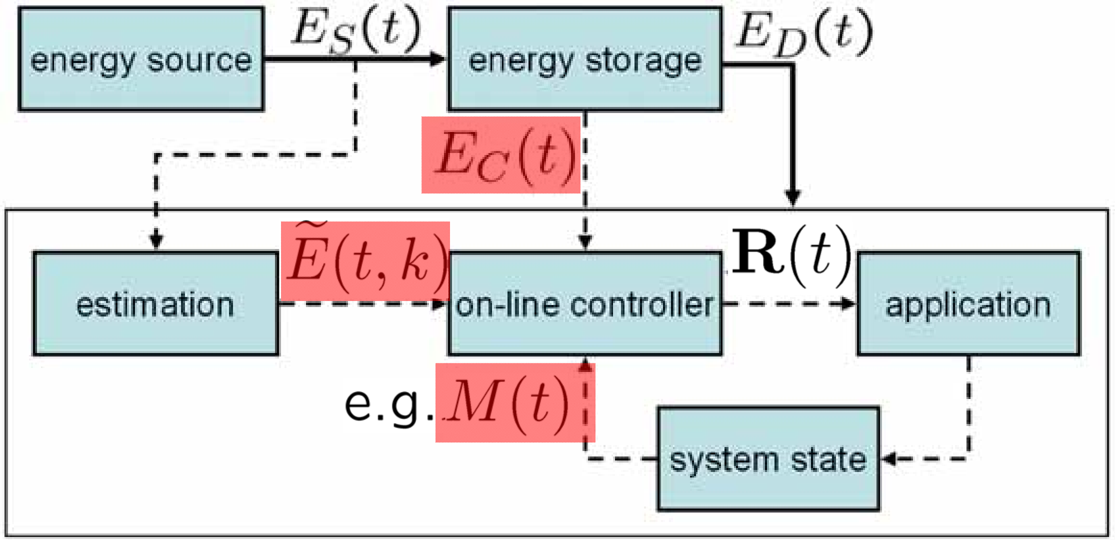


Solving a linear program (LP) in a resource-constraint sensor node at each time step ?

Multiparametric Linear Programming

- Efficient run-time implementation by applying a multiparametric programming approach
- Calculate optimal solution of the mp-LP as an explicit function of the state vector

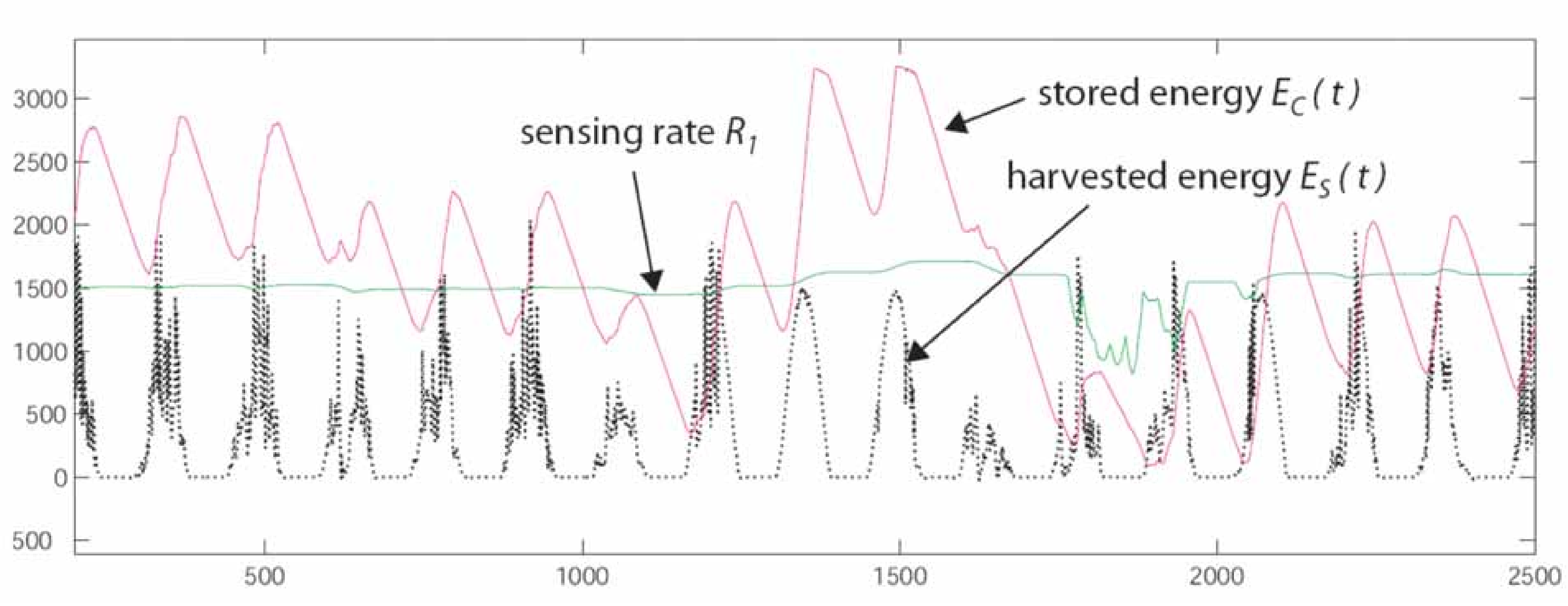
$$\mathbf{X}(t) = \Big(\qquad , \qquad , \ \ldots \ , \qquad , \ \ldots \qquad \Big)^T$$



Online complexity of mp-LP

- sometimes acceptable …

$$\mathbf{R}_{opt}(t) = \mathbf{B}_i \mathbf{X}(t) + \mathbf{C}_i \text{ if } \mathbf{H}_i \mathbf{X}(t) \leq \mathbf{K}_i, i = 1, \ldots, N_{CR} = 7$$



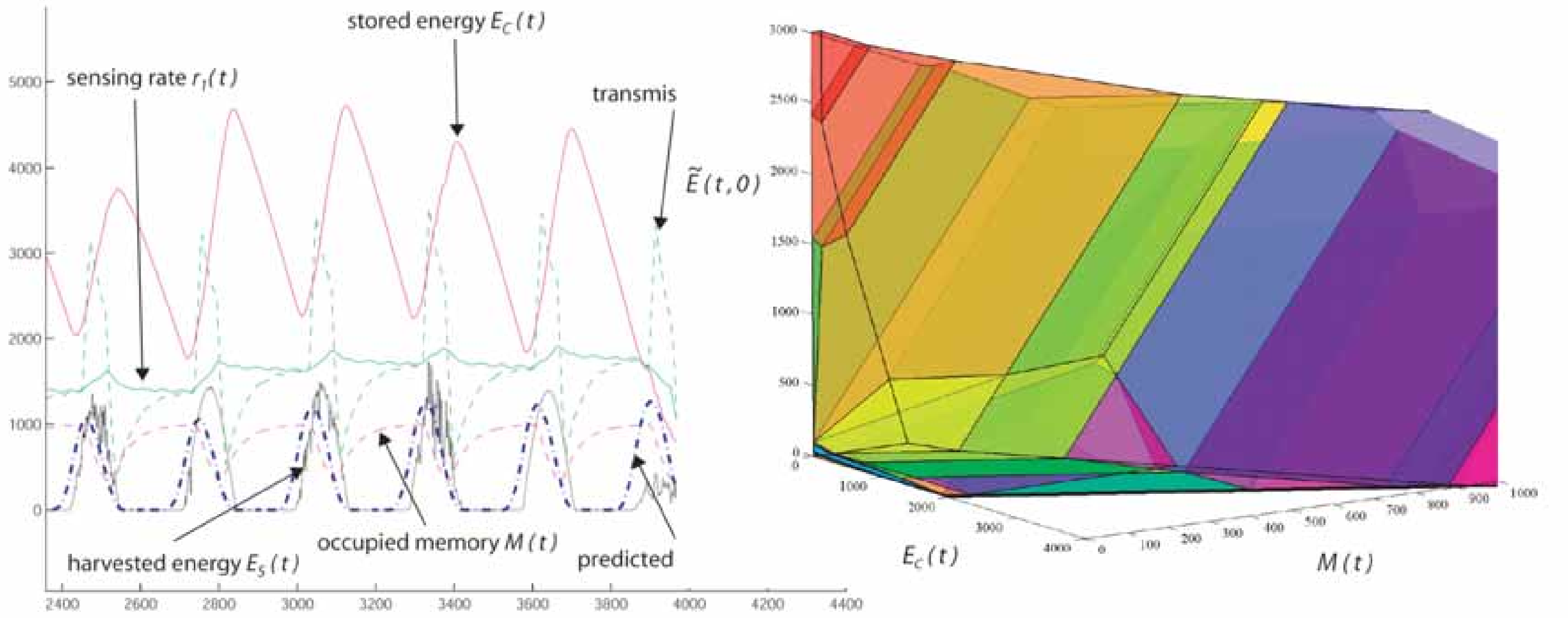
Online complexity of mp-LP

- sometimes not.

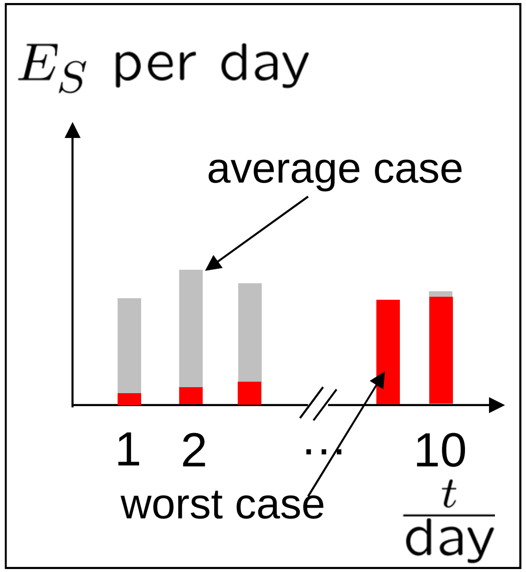
Bottleneck !!!

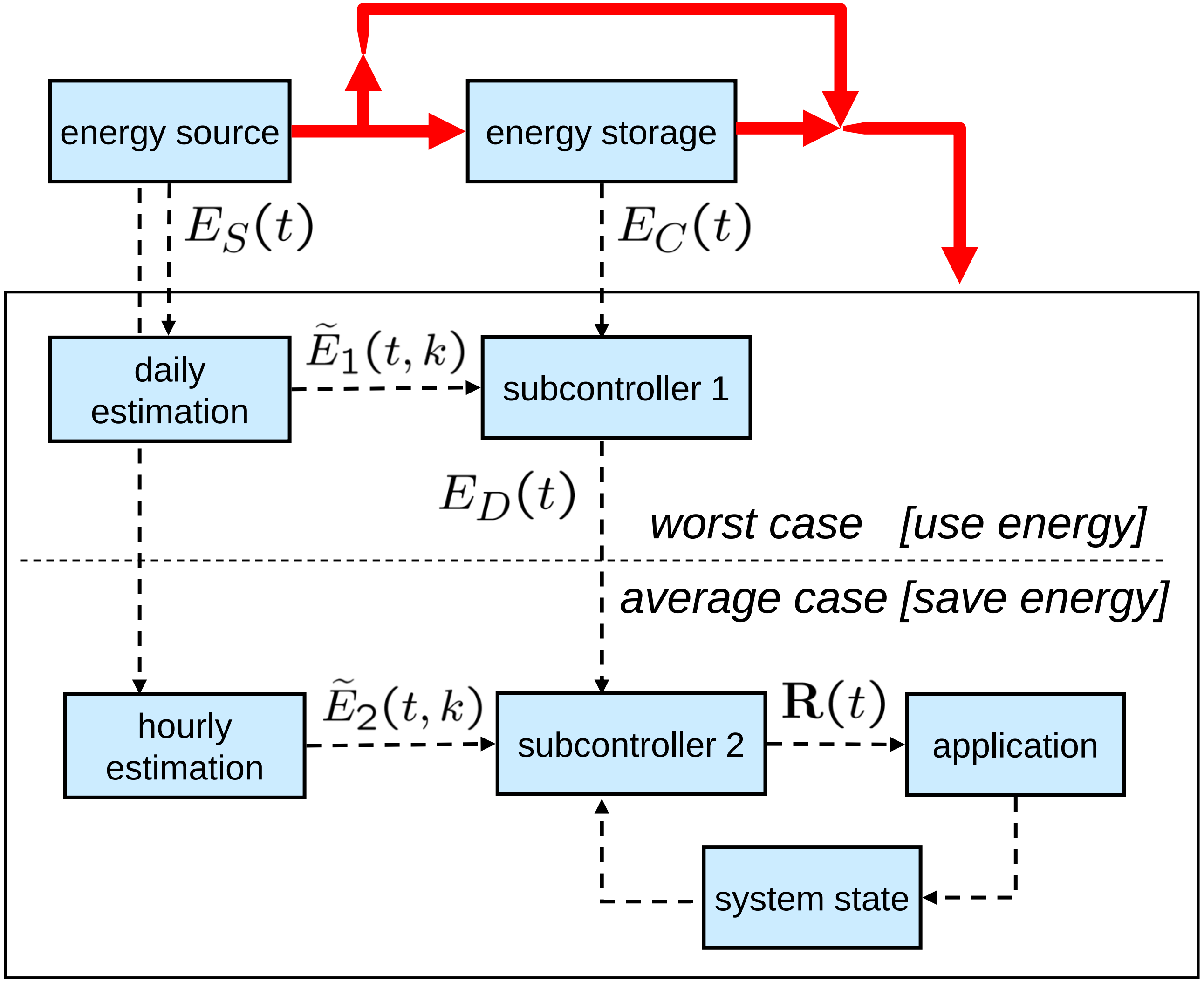
9x7 matrices

$$\mathbf{R}_{opt}(t) = \mathbf{B}_i\mathbf{X}(t) + \mathbf{C}_i \quad \text{if } \mathbf{H}_i\mathbf{X}(t) \le \mathbf{K}_i, i = 1, \ldots, 1049$$



Hierarchical Control Design

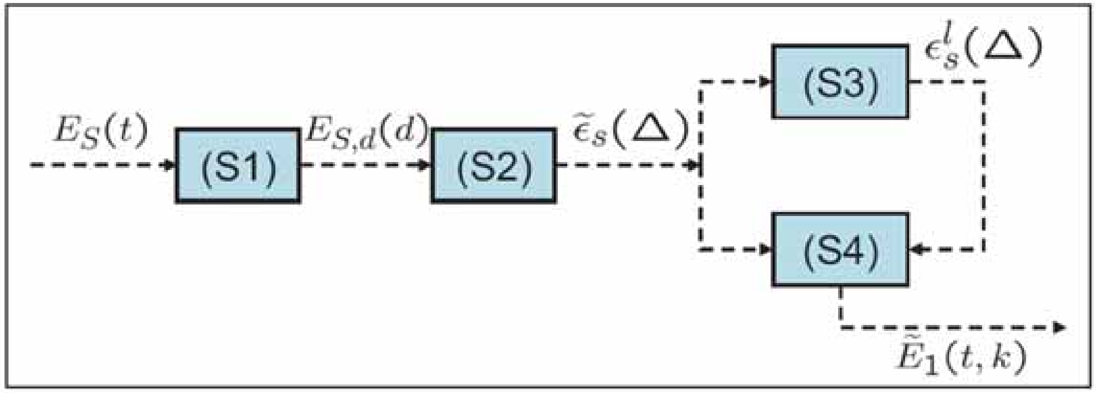




Daily worst-case energy prediction

Algorithm 1 Daily worst-case energ

```
Input: $(t, E_S(t))$
Output: $\widetilde{E}_1(t,k)\ \forall\, 1 \le k \le N$
  if $t == 0$ then
    $\epsilon_s^l(\Delta) = \infty \quad \forall\, 1 \le \Delta \le N$
  if $t \bmod L_1 == 0$ then
    $E_{S,d}(d) = \sum_{i=(d-1)\cdot L}^{d\cdot(L-1)} E_S(t-i) \qquad \forall 1 \le d \le N$   (S1)
    $\widetilde{\epsilon}_s(\Delta) = \sum_{i=1}^{\Delta} E_{S,d}(i) \qquad \forall 0 \le \Delta \le N$   (S2)
    $\epsilon_s^l(\Delta) = \epsilon_s^l(\Delta) + \Delta \cdot \gamma \qquad \forall 1 \le \Delta \le N$   (S3)
    $\epsilon_s^l(\Delta) = \min\left[\epsilon_s^l(\Delta), \widetilde{\epsilon}_s(\Delta)\right] \qquad \forall 1 \le \Delta \le N$   (S3)
    $\widetilde{E}_1(t,k) = \max_{0 \le l \le N-k} \left\{\epsilon_s^l(k+l) - \widetilde{\epsilon}_s(l)\right\}\ \forall k \ge 1$   (S4)
```



Approximate mp-LP

- *Principle*
  - Generate a large number of samples X_i
  - For many states X_i, solve the linear program and calculate the optimal rates $R_{opt}(X_i)$
  - Perform piecewise-linear fitting[1] of the data $(R_{opt}(X_i), X_i)$ to obtain the approximation $\hat{R}(X)$

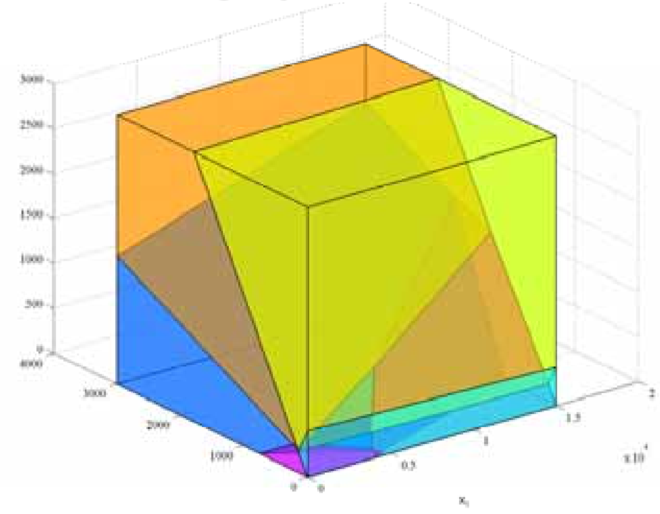
[1] A. Magnani and S. Boyd, "Convex piecewise-linear fitting," in *Optimization and Engineering (submitted)*, http://www.stanford.edu/~boyd/reports/cvx_pwl_fit.pdf, April 2006.

Approximate mp-LP

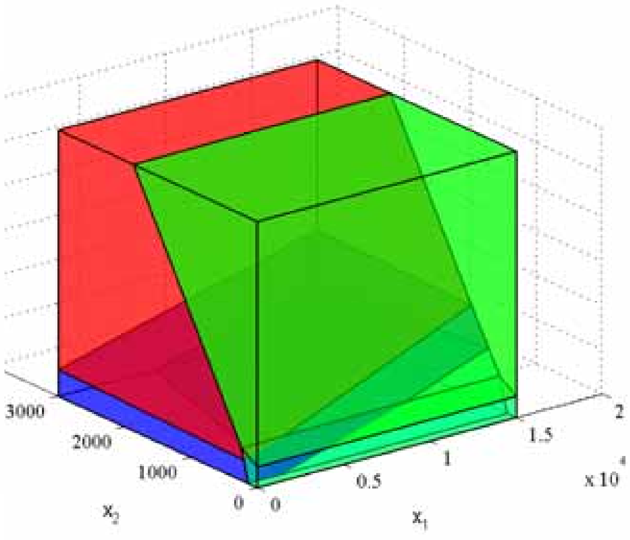
$$\hat{\mathrm{R}}(t) = \hat{\mathrm{B}}_i \mathbf{X}(t) + \hat{\mathrm{C}}_i$$

$$\text{if } \hat{\mathrm{H}}_i \mathbf{X}(t) \le \hat{\mathrm{K}}_i, \quad i = 1, \ldots, \hat{N}_{CR} \le N_{CR} = 161$$

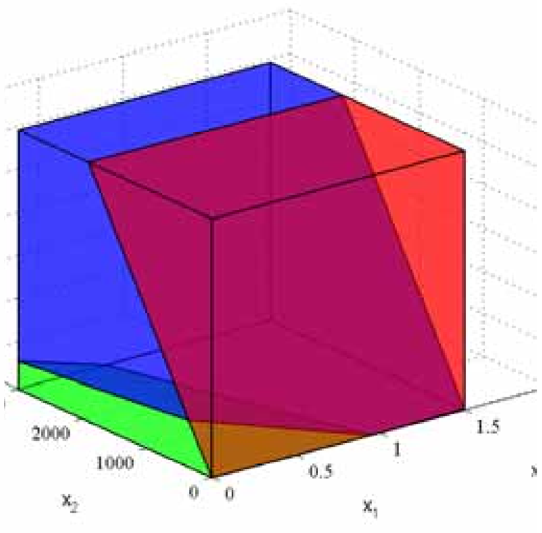
$\hat{N}_{CR} = 10$



$\hat{N}_{CR} = 5$

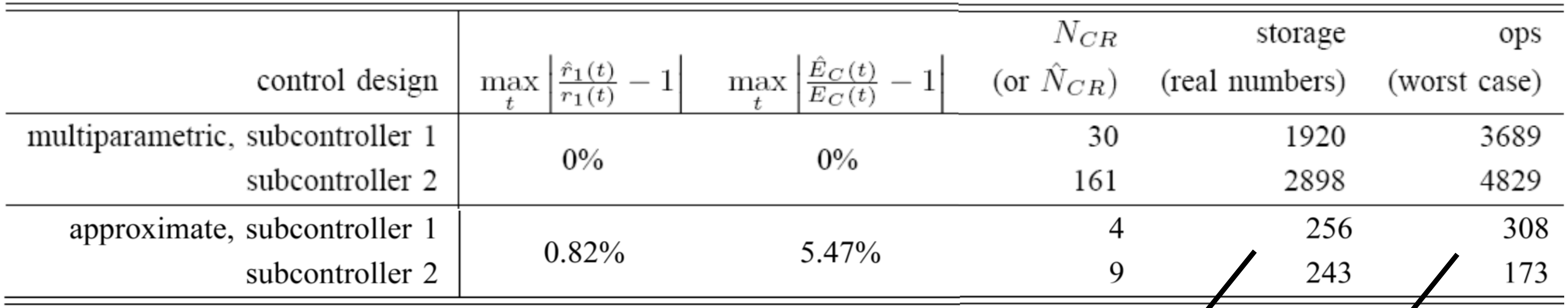


$\hat{N}_{CR} = 3$



Approximate mp-LP

- Comparison with optimal solution



control design	$\max_t \left\lvert \frac{\hat{r}_1(t)}{r_1(t)} - 1 \right\rvert$	$\max_t \left\lvert \frac{\hat{E}_C(t)}{E_C(t)} - 1 \right\rvert$	N_{CR} (or $\hat{N}_{CR}$)	storage (real numbers)	ops (worst case)
multiparametric, subcontroller 1	0%	0%	30	1920	3689
subcontroller 2			161	2898	4829
approximate, subcontroller 1	0.82%	5.47%	4	256	308
subcontroller 2			9	243	173

Reduction: 89.6% 96.4%

Contents

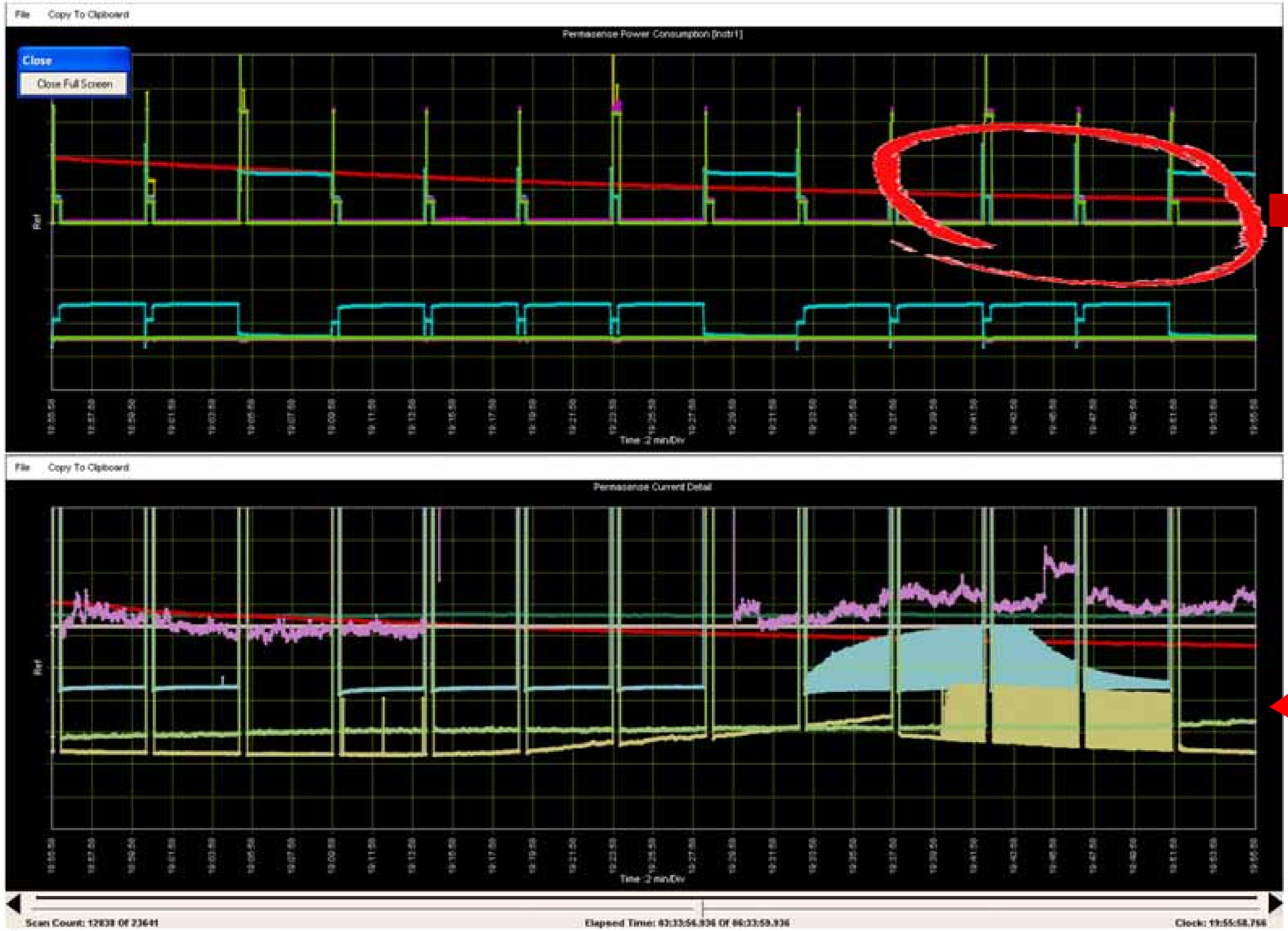
- ***Part 1***
  - Energy Harvesting
  - *Power Testing*
  - Predictable Communication Protocols

- ***Part 2***
  - PermSense System Design
  - Test & Validation Infrastructure
  - Data Cleaning & Validation
  - Calibration of Mobile Sensors
  - Complex Sensing

Power Testing

- how formal methods can help -

Example –Power Profiling

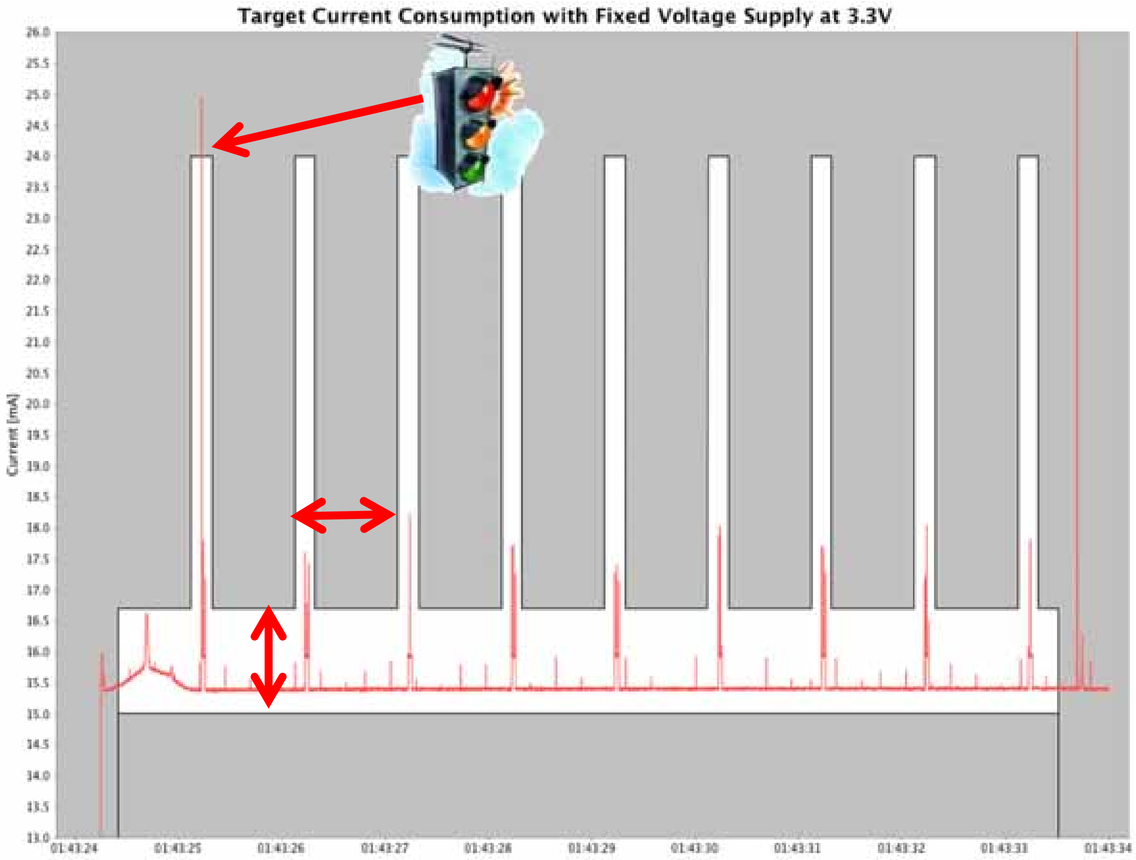


Scan Count: 12838 Of 23641

Elapsed Time: 03:33:56.936 Of 06:33:59.936

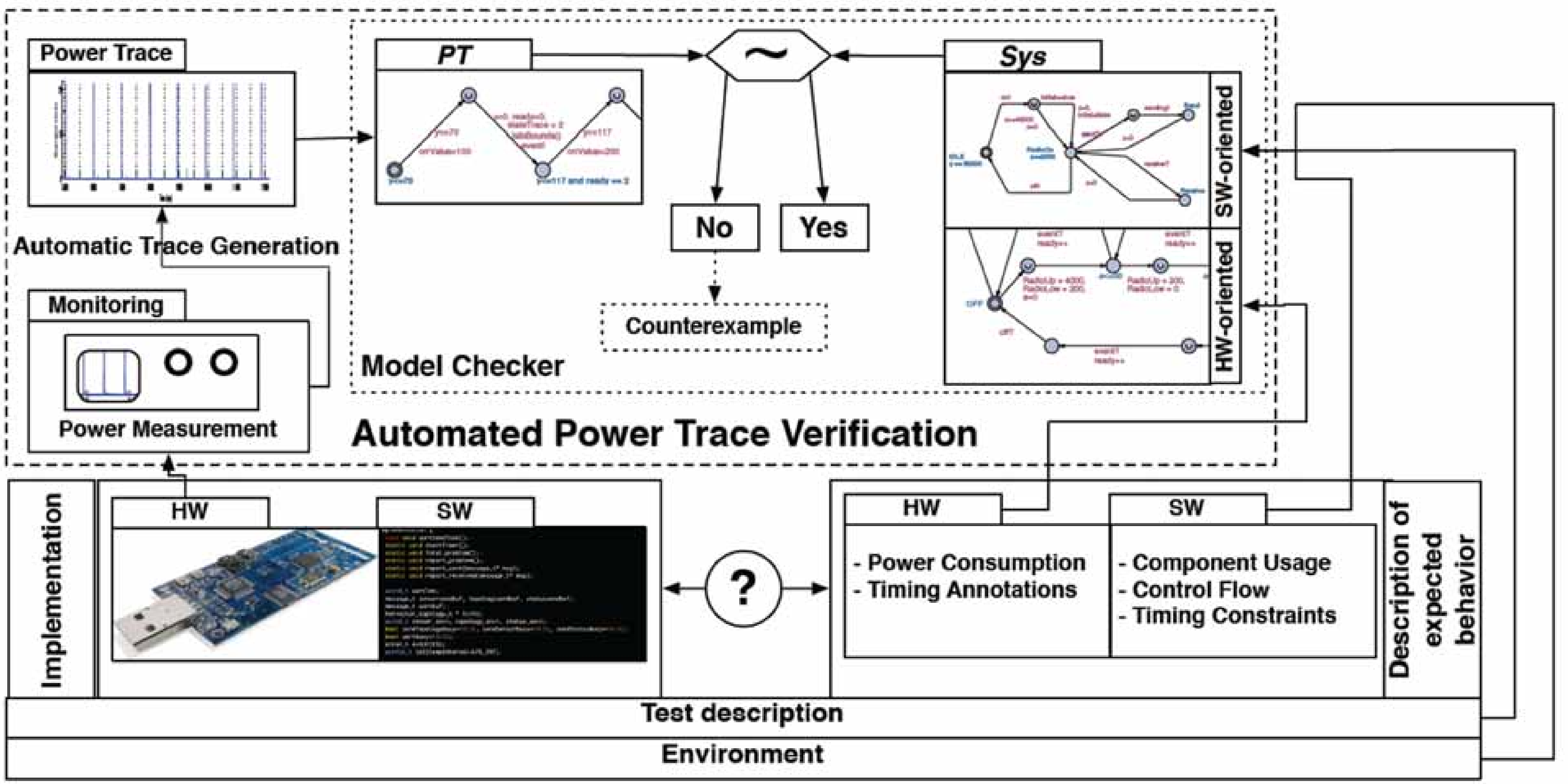
Clock: 19:55:58.766

Solution 1: Power Trace Unit Testing

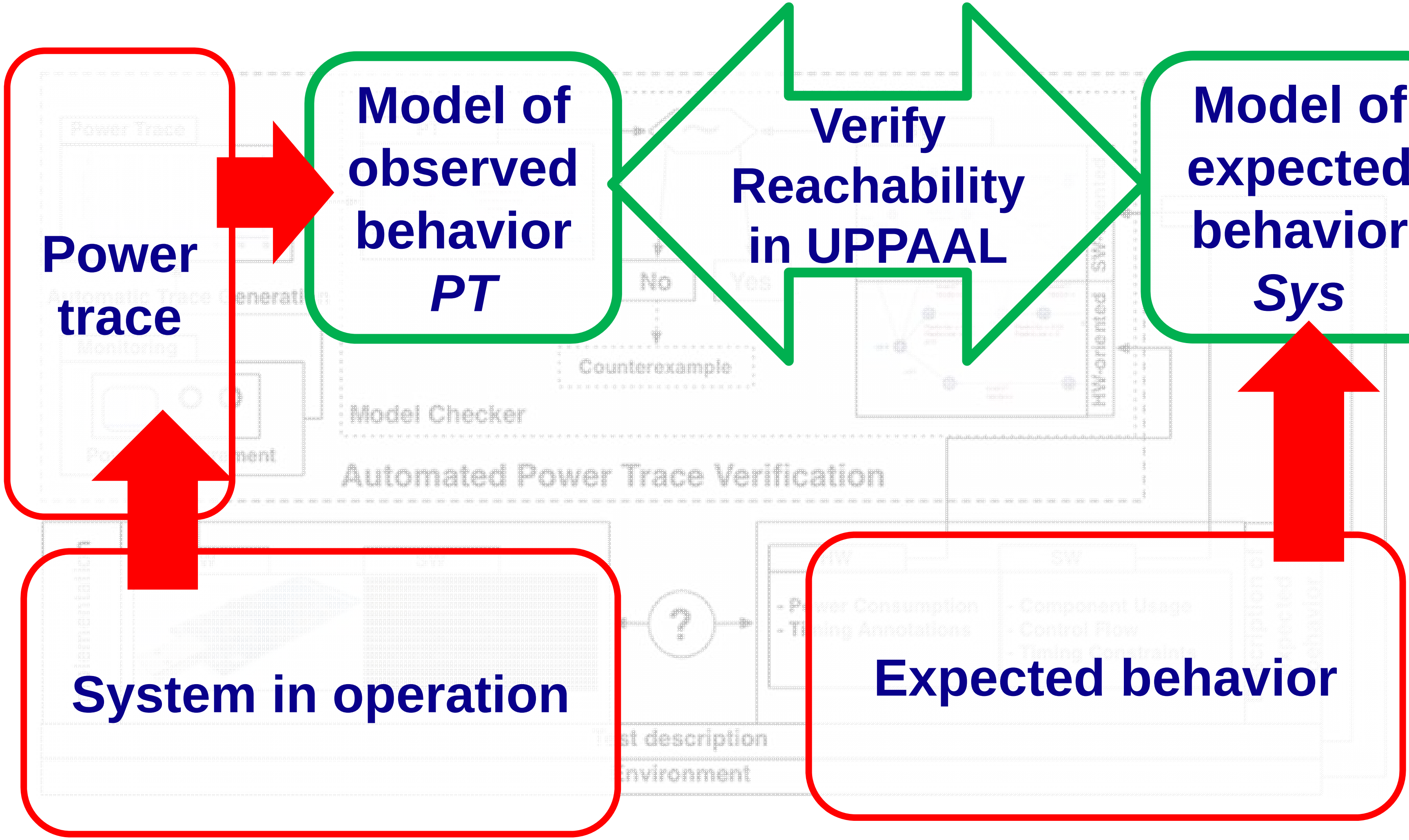


- Assertions based on reference traces/specification
- Power regression testing

Solution 2: Formal Conformance Test



Solution 2: Formal Conformance Test



Power trace

Model of observed behavior *PT*

Verify Reachability in UPPAAL

Model of expected behavior *Sys*

System in operation

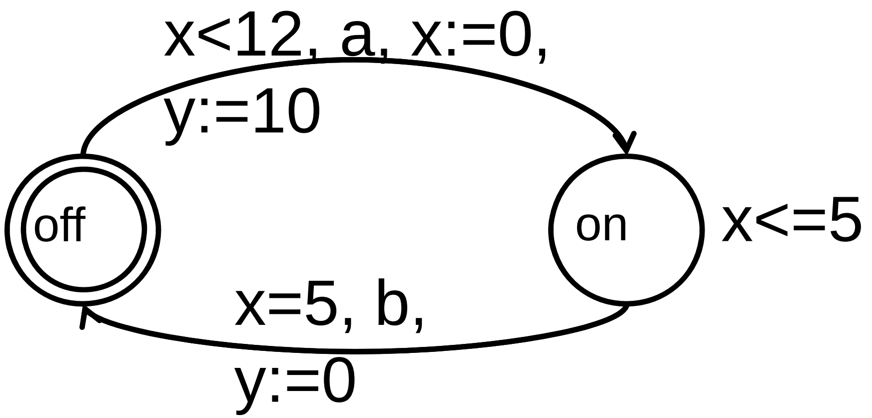
Expected behavior

[FORMATS 2009]

Timed Automata (TA extended with data variables)

- *Why timed automata ?*
  - Infinite state system
  - But, reachability is decidable
  - Mature tools are available (Uppaal, Kronos)
  - Power consumption as data variable extension

clock x;
label a,b;
int y:=0; // [in mA]

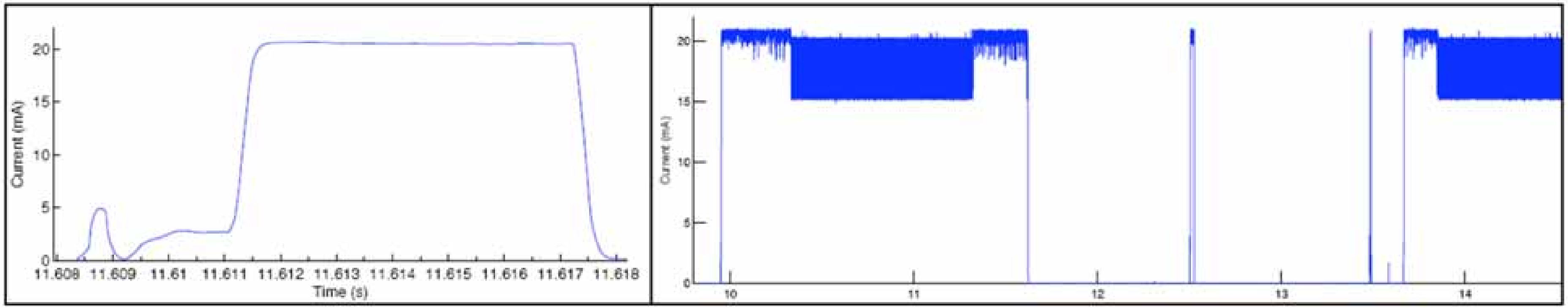


Formal Conformance Test using TA

- Use of timed automata for conformance testing
- ***Modular modeling*** of
  - Hardware
  - Software
  - Testing Environment
  - Trace
- ***Complexity reduction*** by
  - noise removal
  - abstraction and interval arithmetic
- Testing by ***reachability analysis*** of the composed model (UPPAAL)

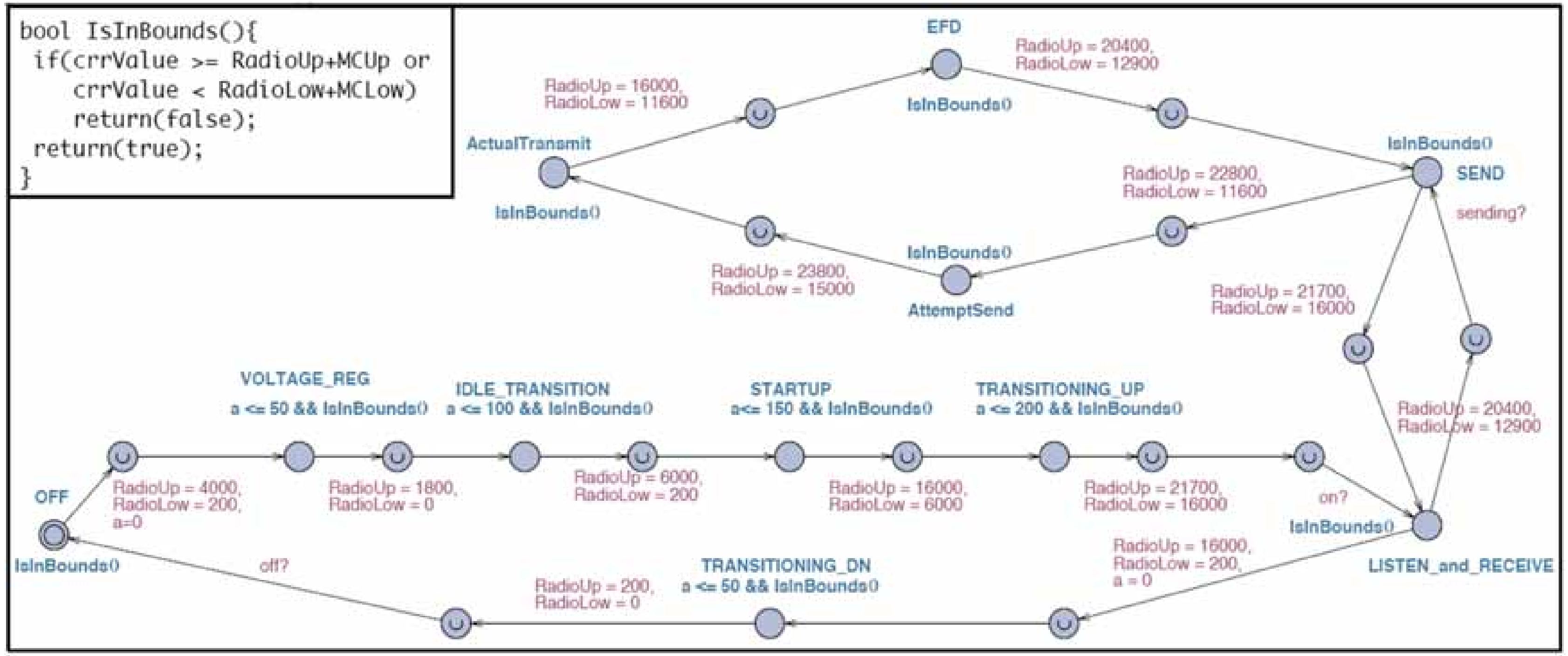
Formal Conformance Test using TA

- ***Experiment***
  - TMOTE Sky (MSP 430, TI CC2420) running TinyOS2
  - Harvester application, LPL MAC protocol
  - Besides testing a correct run (complex trace), several errors have been introduced (missing wake-up, inject error in low power scheduler, wrong low power state of MC, specification error).



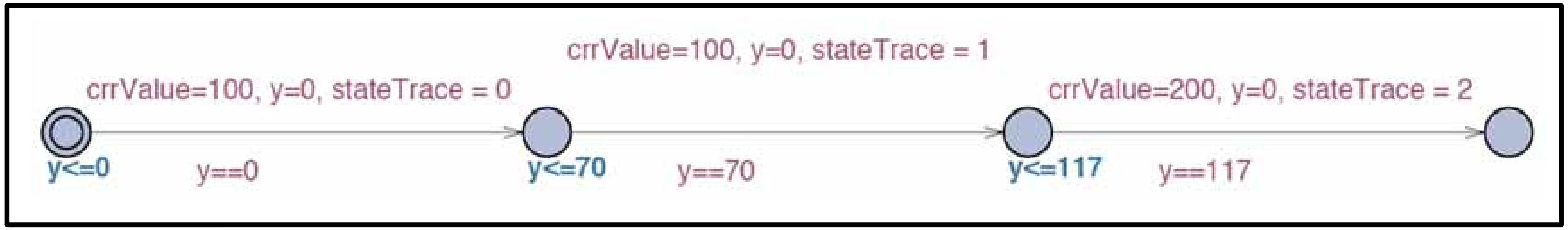
Formal Conformance Test using TA

Hardware Model



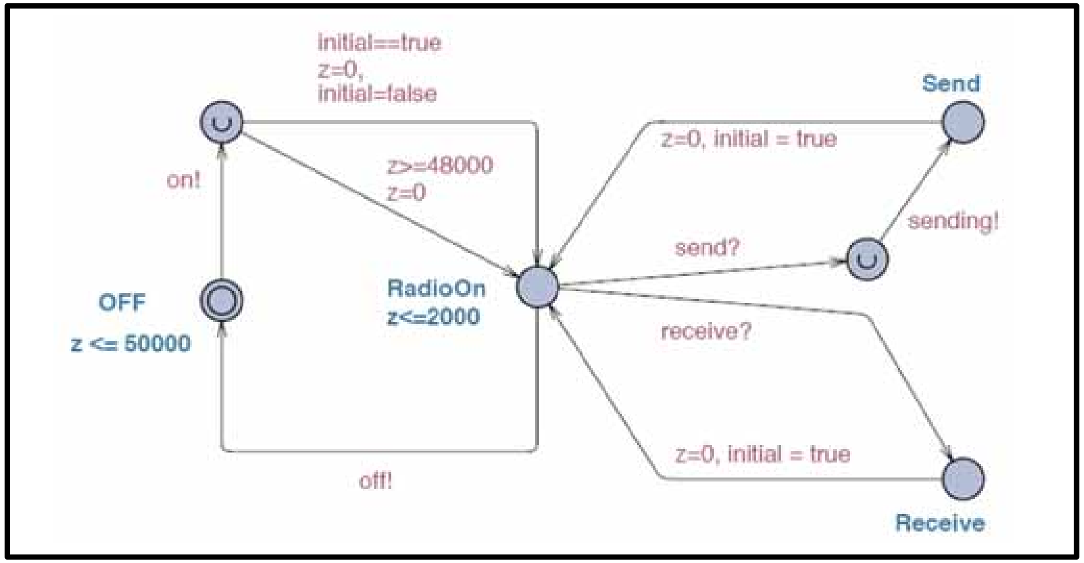
```
bool IsInBounds(){
 if(crrValue >= RadioUp+MCUp or
    crrValue < RadioLow+MCLow)
    return(false);
 return(true);
}
```

Trace Automaton



Formal Conformance Test using TA

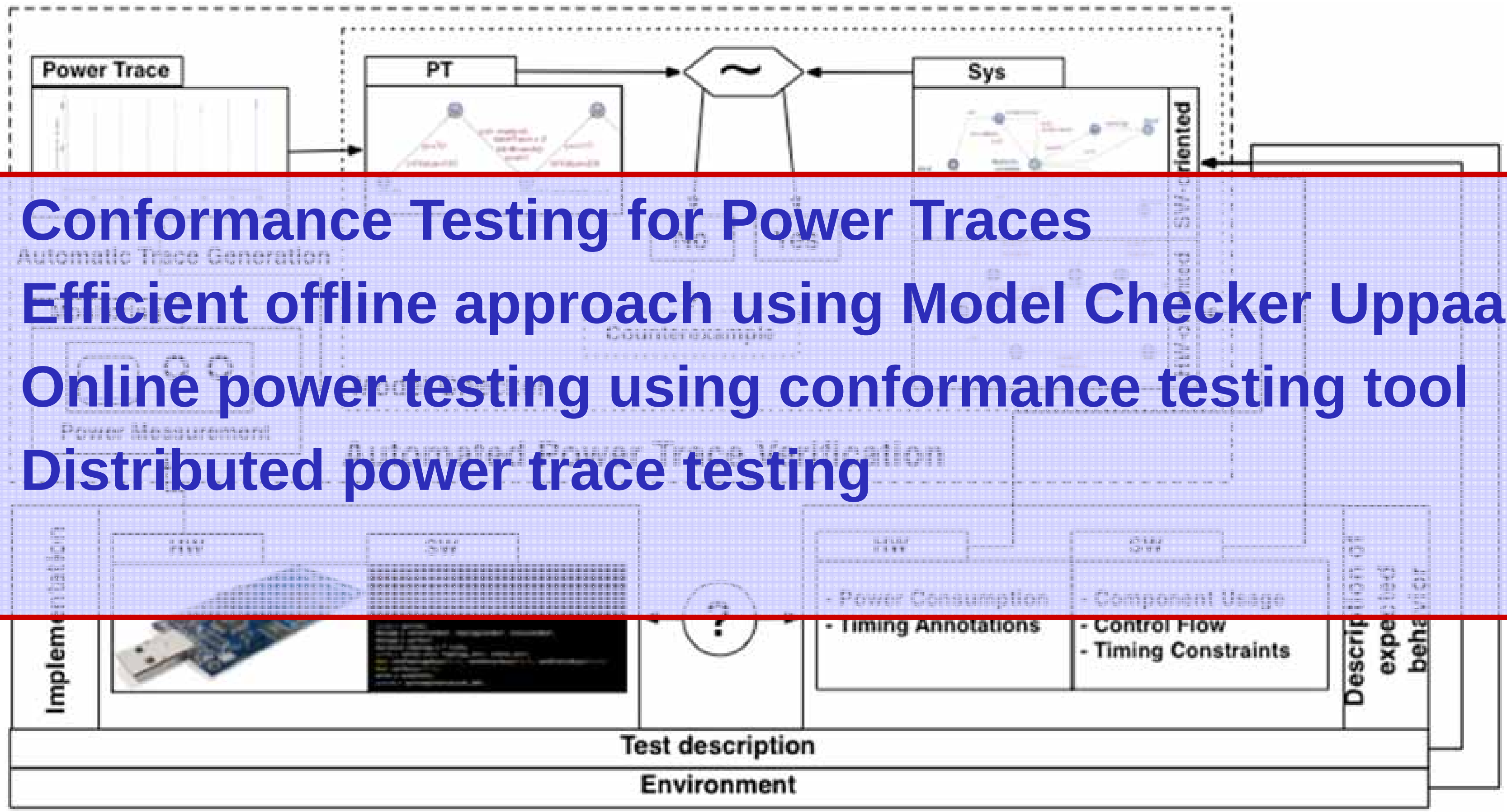
Software Model



Verification Results

Model	Samples	Compression		Runtime	
		Intervals	100uA steps	Intervals	100uA steps
Wake-up	1000000	2089	13812	8s	182s
Inject	990000	2040	13666	5s	167s
Complex Trace	310000	21811	89678	15min	Fails
MC state	1000000	1086	5539	4s	22s
Specification	1000000	1478	4578	7s	22s

Summary (PTT)



- Conformance Testing for Power Traces
- Efficient offline approach using Model Checker Uppaal
- Online power testing using conformance testing tool
- Distributed power trace testing

Contents

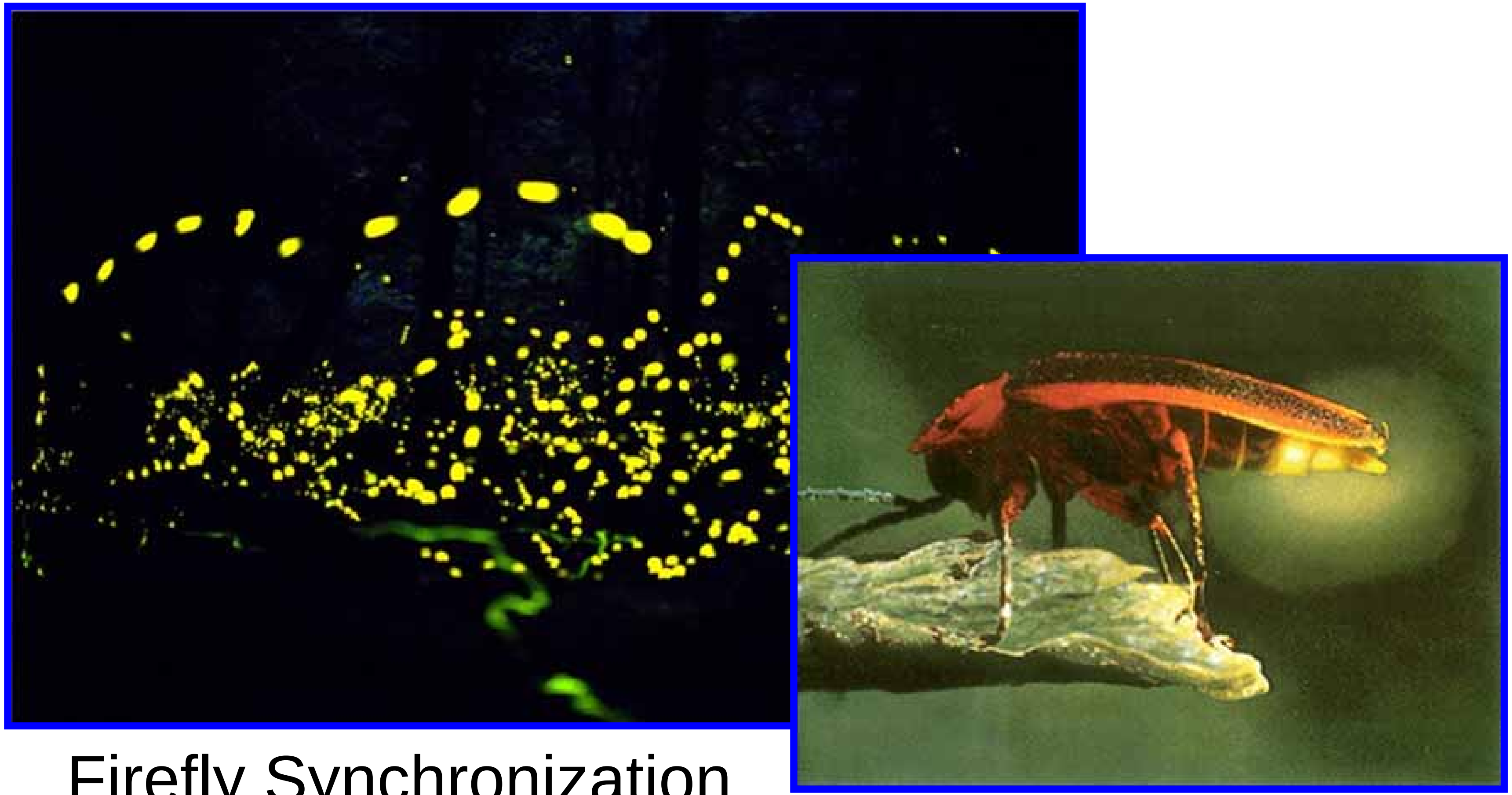
- ***Part 1***
  - Energy Harvesting
  - Power Testing
  - *Predictable Communication Protocols*

- ***Part 2***
  - PermSense System Design
  - Test & Validation Infrastructure
  - Data Cleaning & Validation
  - Calibration of Mobile Sensors
  - Complex Sensing

Communication

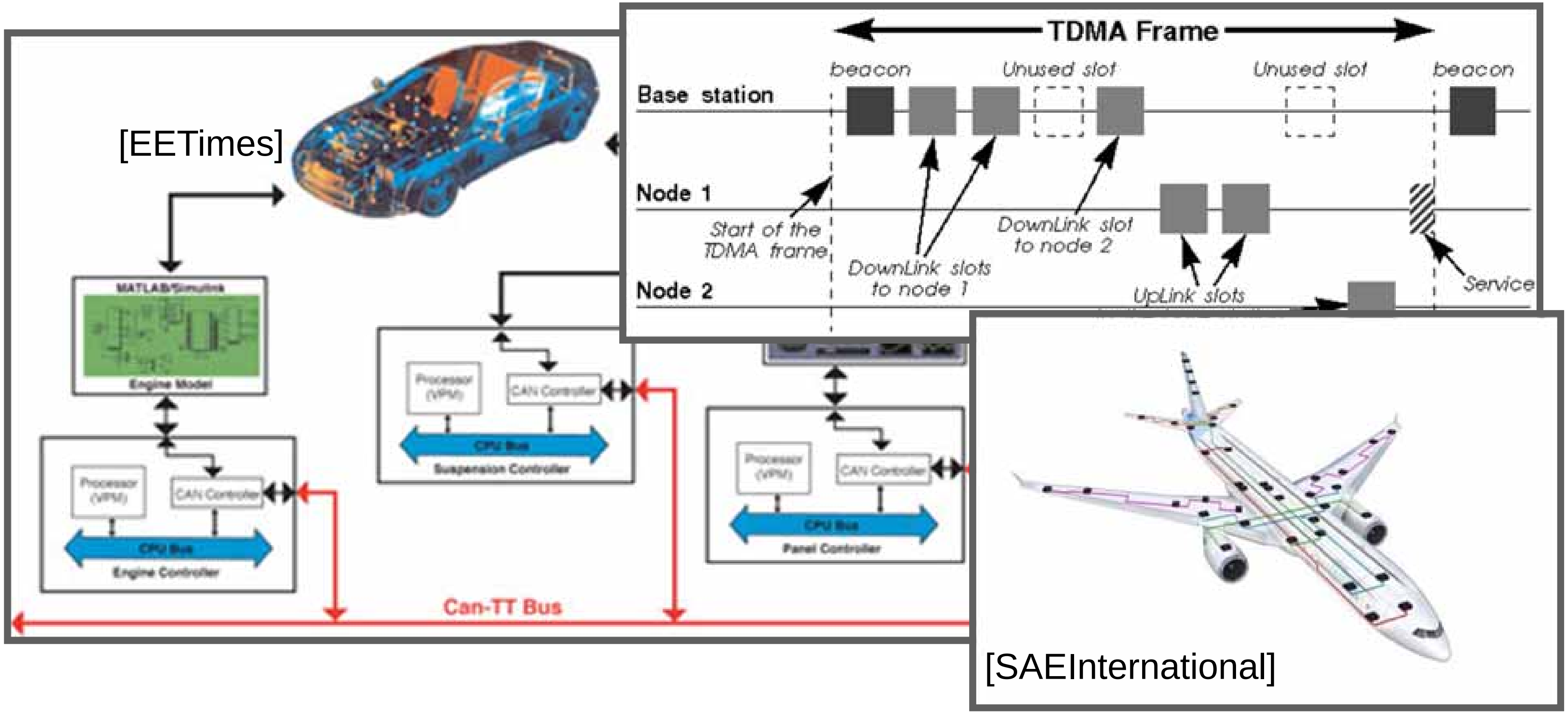
- how biology and history can help -

Predictable and Efficient Communication



Firefly Synchronization

Predictable and Efficient Communication



[EETimes]

[SAEInternational]

Bus-based communication

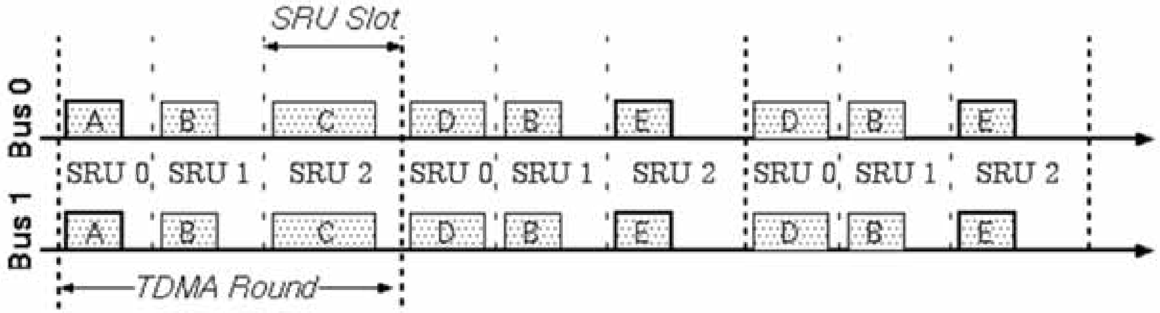
Time-triggered operation

[Kopetz et. al.]

Designing Predictable (Wired) Systems

Example: Time-triggered architecture

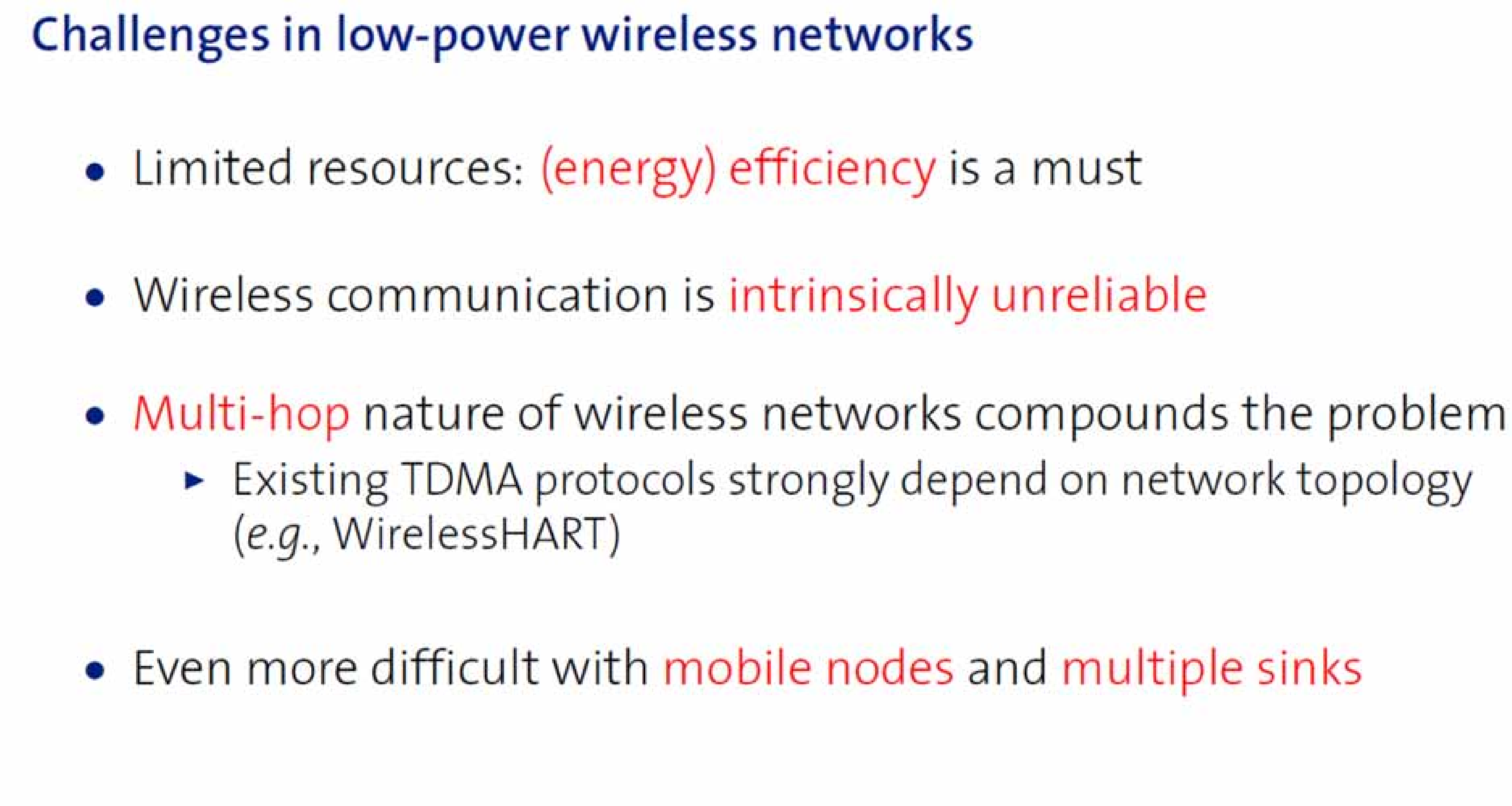
- Communication occurs over shared buses
- Nodes are time-synchronized
- Communication is organized in TDMA rounds
    - Rounds are divided into slots
    - Slots are assigned to nodes based on a fixed schedule



[Scheidler *et al.*, Time-Triggered Architecture (TTA)]

Is it feasible to design time-triggered low-power wireless networks?

Challenges

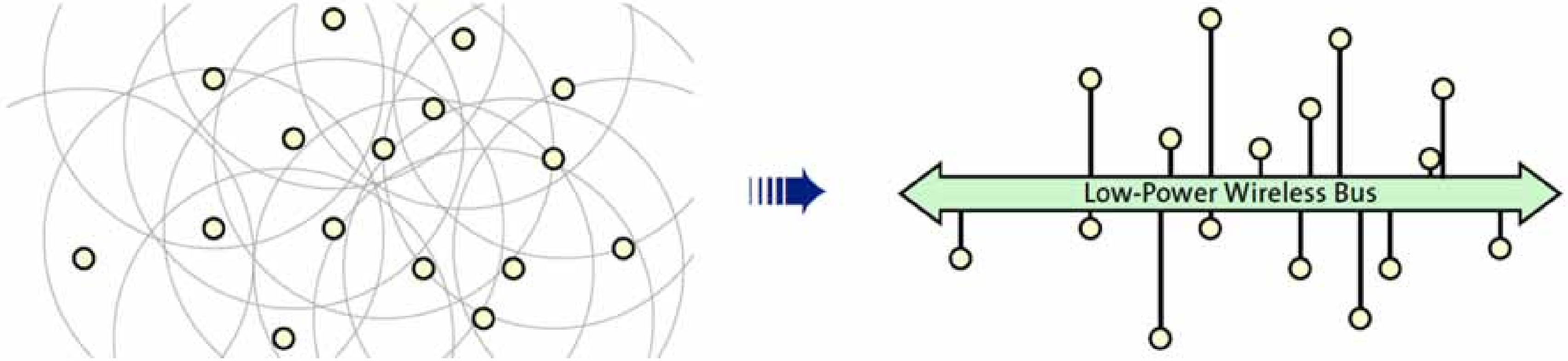


Challenges in low-power wireless networks

- Limited resources: (energy) efficiency is a must
- Wireless communication is intrinsically unreliable
- Multi-hop nature of wireless networks compounds the problem
  - Existing TDMA protocols strongly depend on network topology (*e.g.*, WirelessHART)
- Even more difficult with mobile nodes and multiple sinks

The Low-Power Wireless Bus

Shared bus for low-power wireless networks



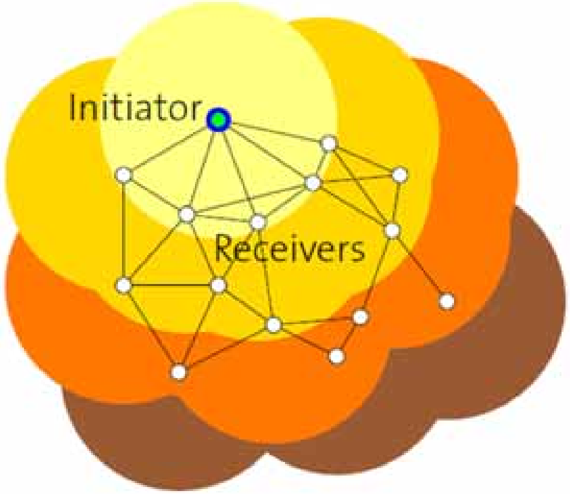
- Nodes communicate using only network floods (one-to-all)
- At any point in time:
  - At most one node (master) initiates a flood
  - Remaining nodes (slaves) receive and relay flooding packet
- Time-triggered communication
  - Nodes are time-synchronized
  - Global schedule distributes master role over time
  - Dedicated host node announces the schedule

Is there a flooding architecture that is sufficiently reliable and efficient?

Glossy I

Efficient network flooding architecture for wireless sensor networks

- Temporally decouples flooding from other activities
- Exploits synchronous transmissions of the same packet



Nodes within the same color area relay the packet at the same time

- Reliability often higher than 99.99 %
- A few milliseconds to flood a packet

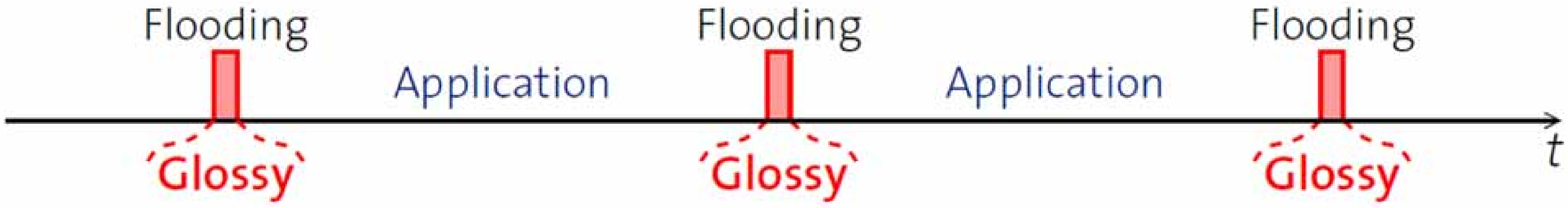
} Sufficiently reliable and efficient

- Time synchronization at no additional cost
- Requires no knowledge of network topology

} Exploited by the LWB

Glossy II

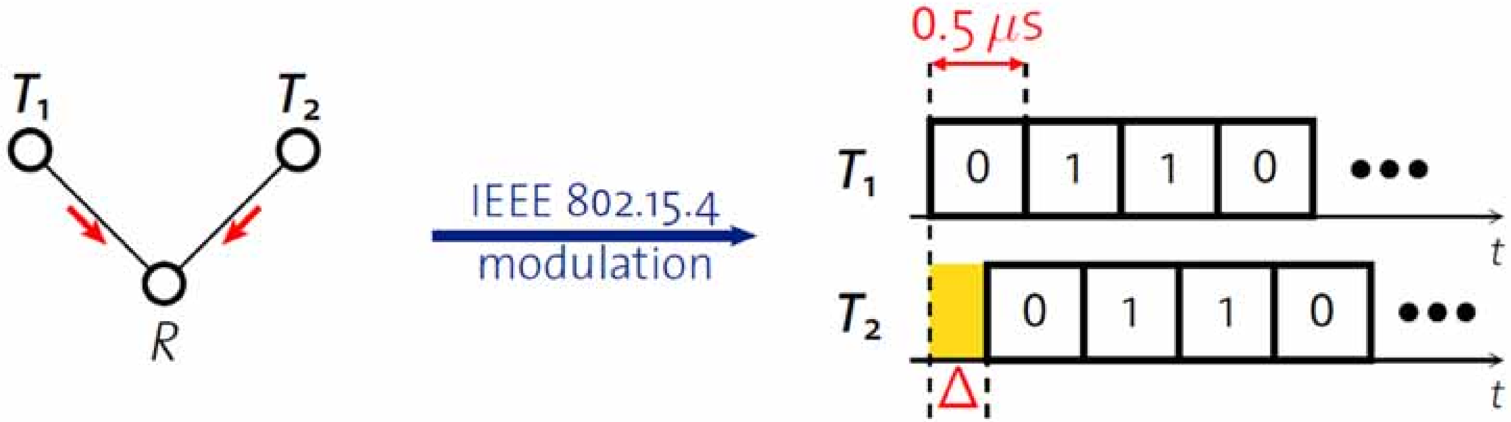
- Temporally decouple network flooding from application tasks



- Exploit synchronous transmissions for fast network flooding

Glossy III

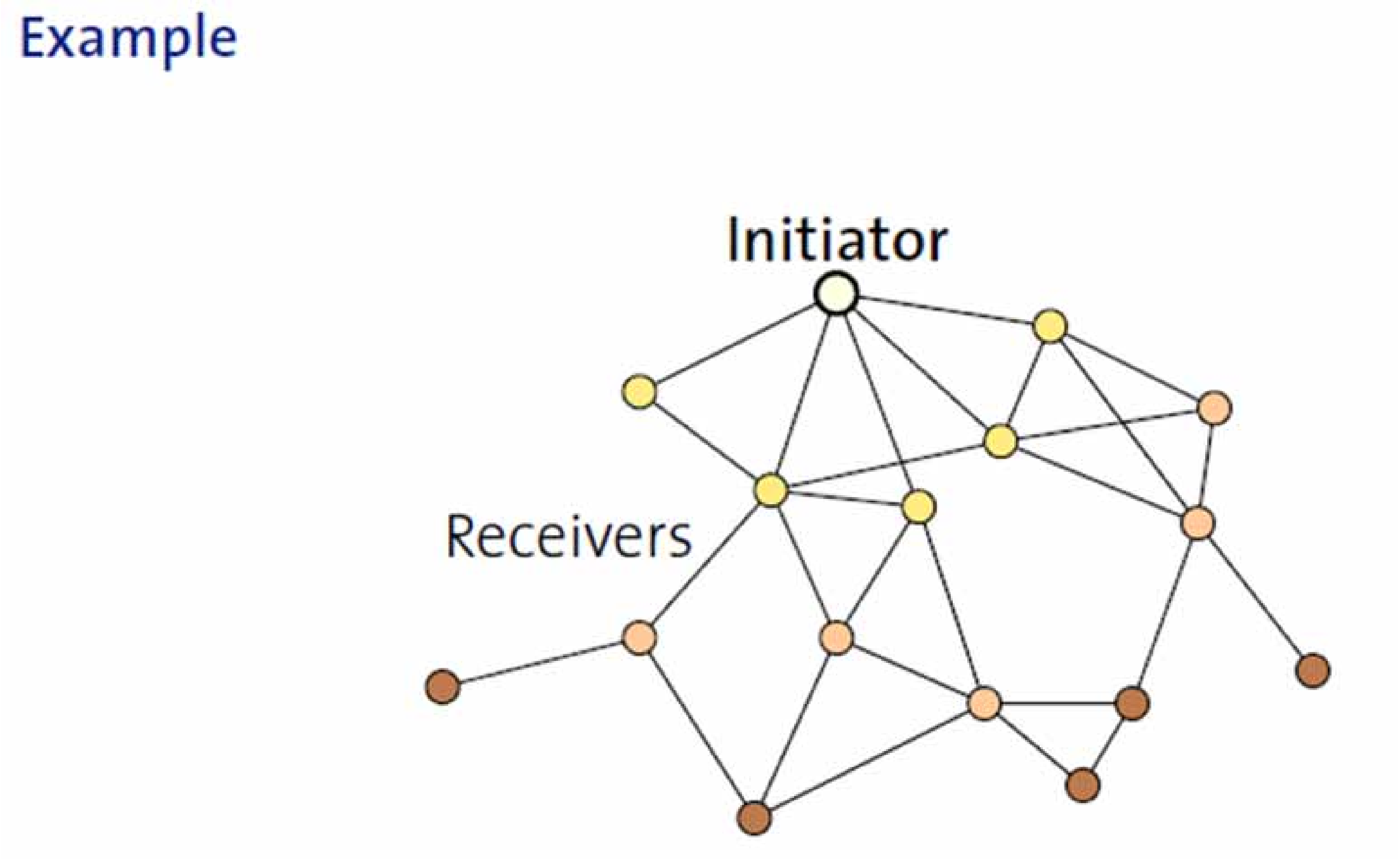
- Multiple nodes transmit same packet at same time



- R receives packet with high probability if $\Delta \leq 0.5\ \mu s$

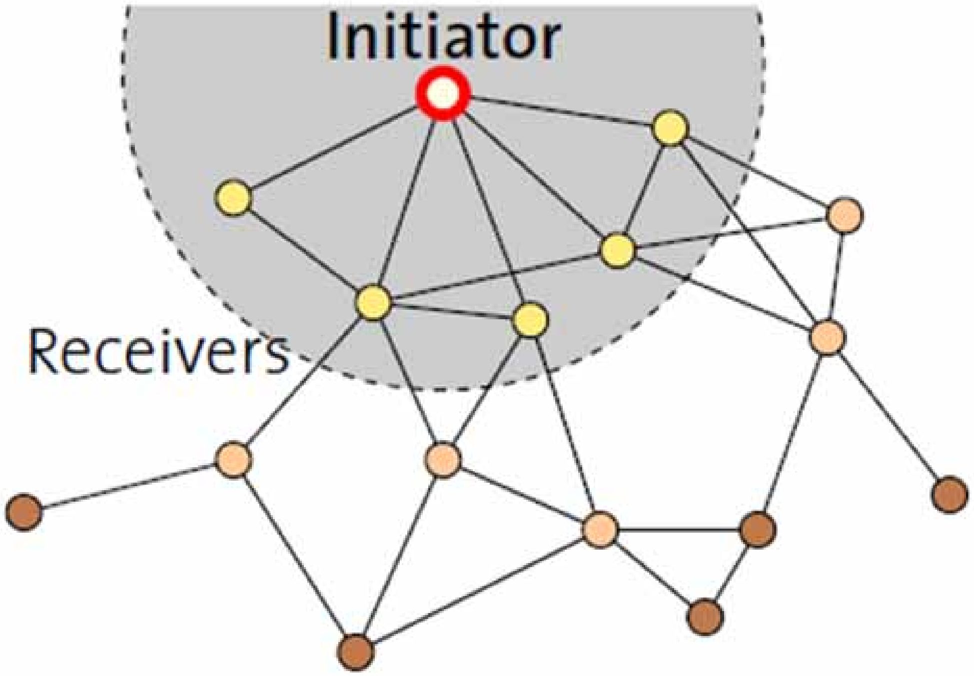
Property exploited also in **A-MAC** [Dutta et al., SenSys '10]

Glossy IV



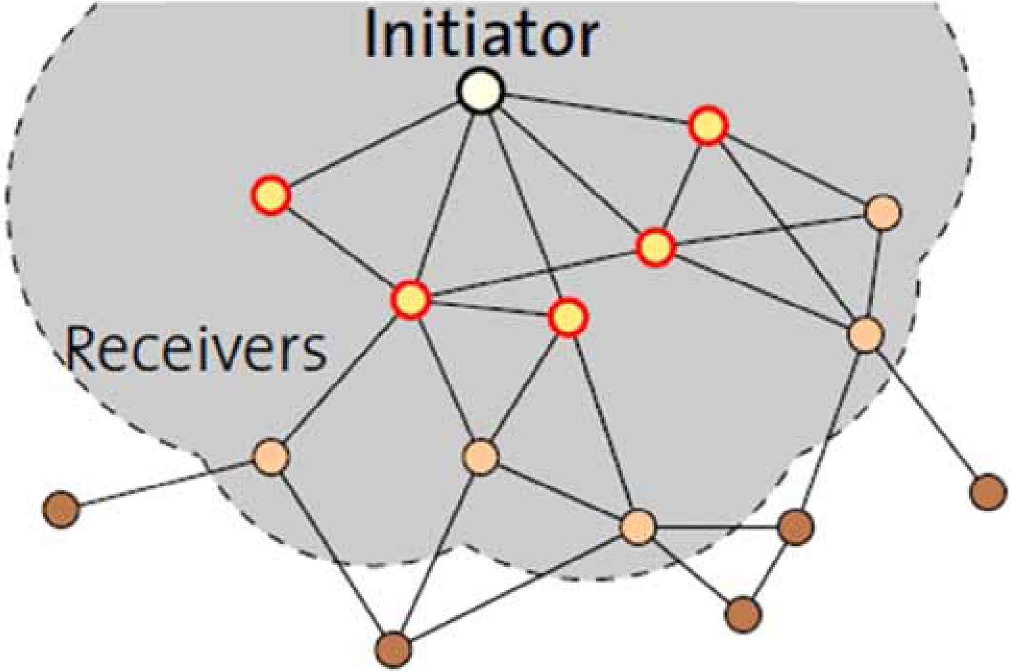
Glossy IV

Example



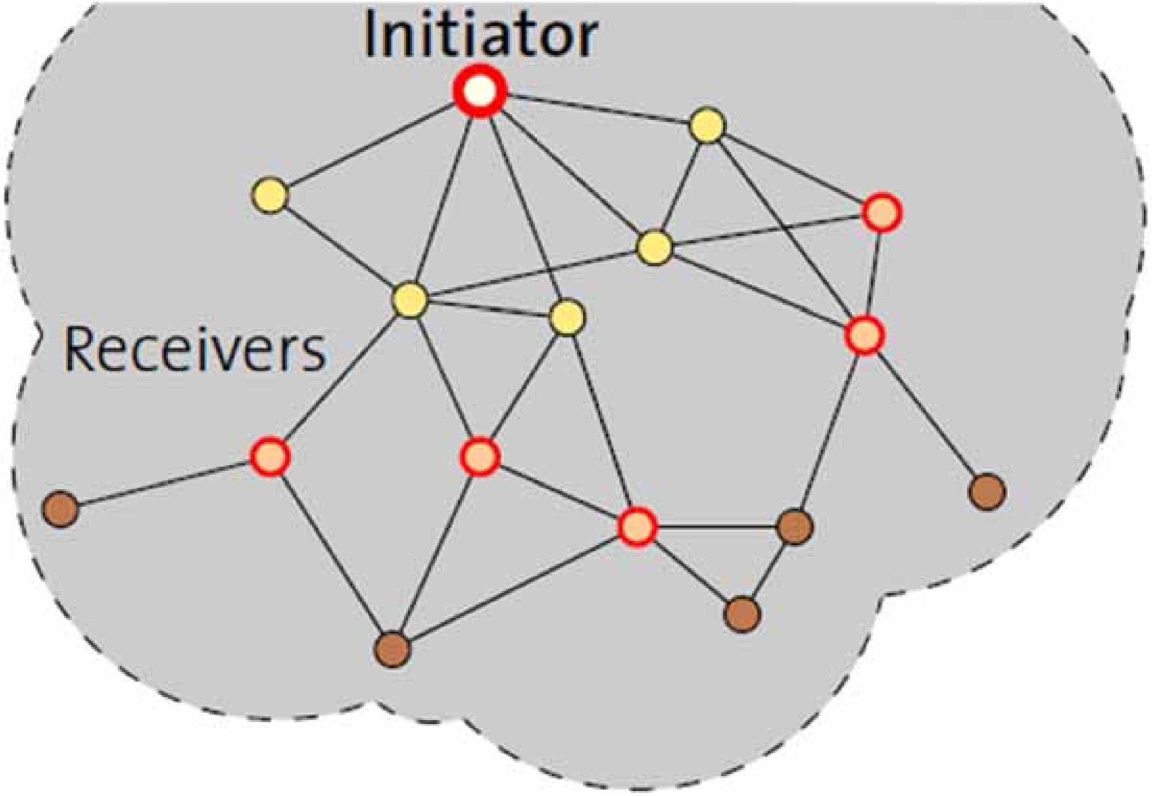
Glossy IV

Example



Glossy IV

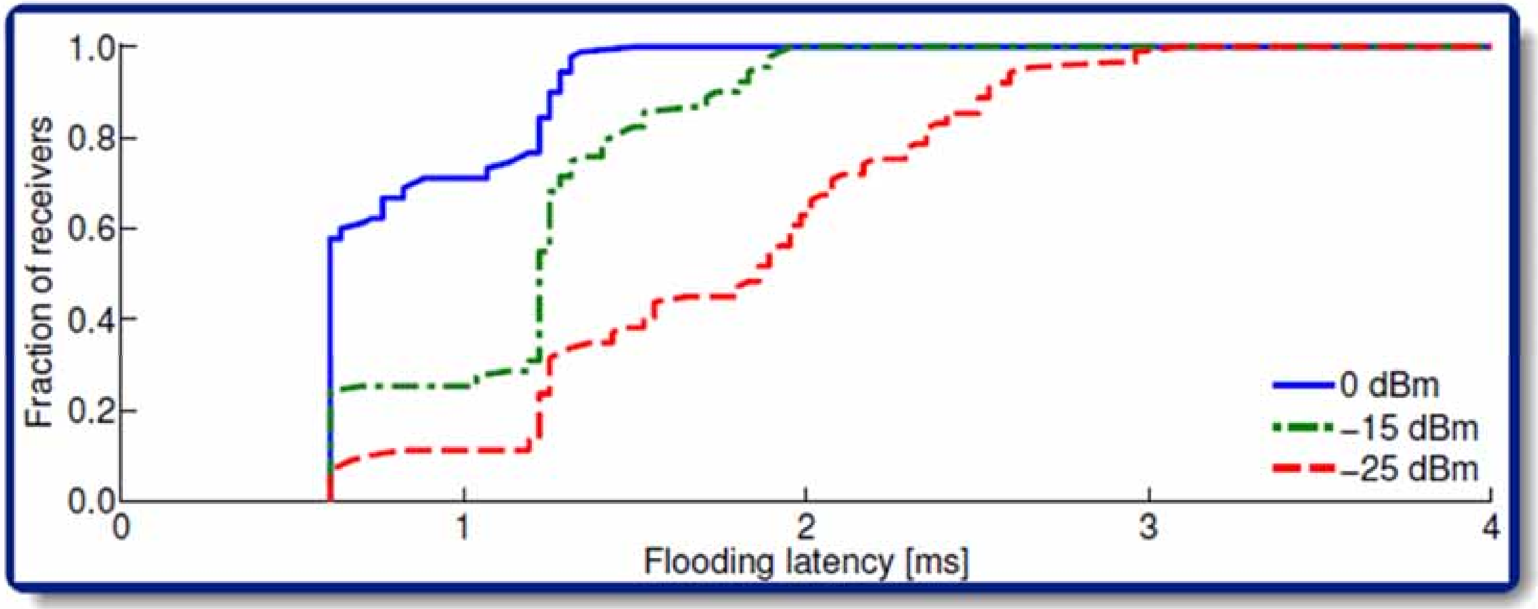
Example



Impact of Network Characteristics

Flooding latency

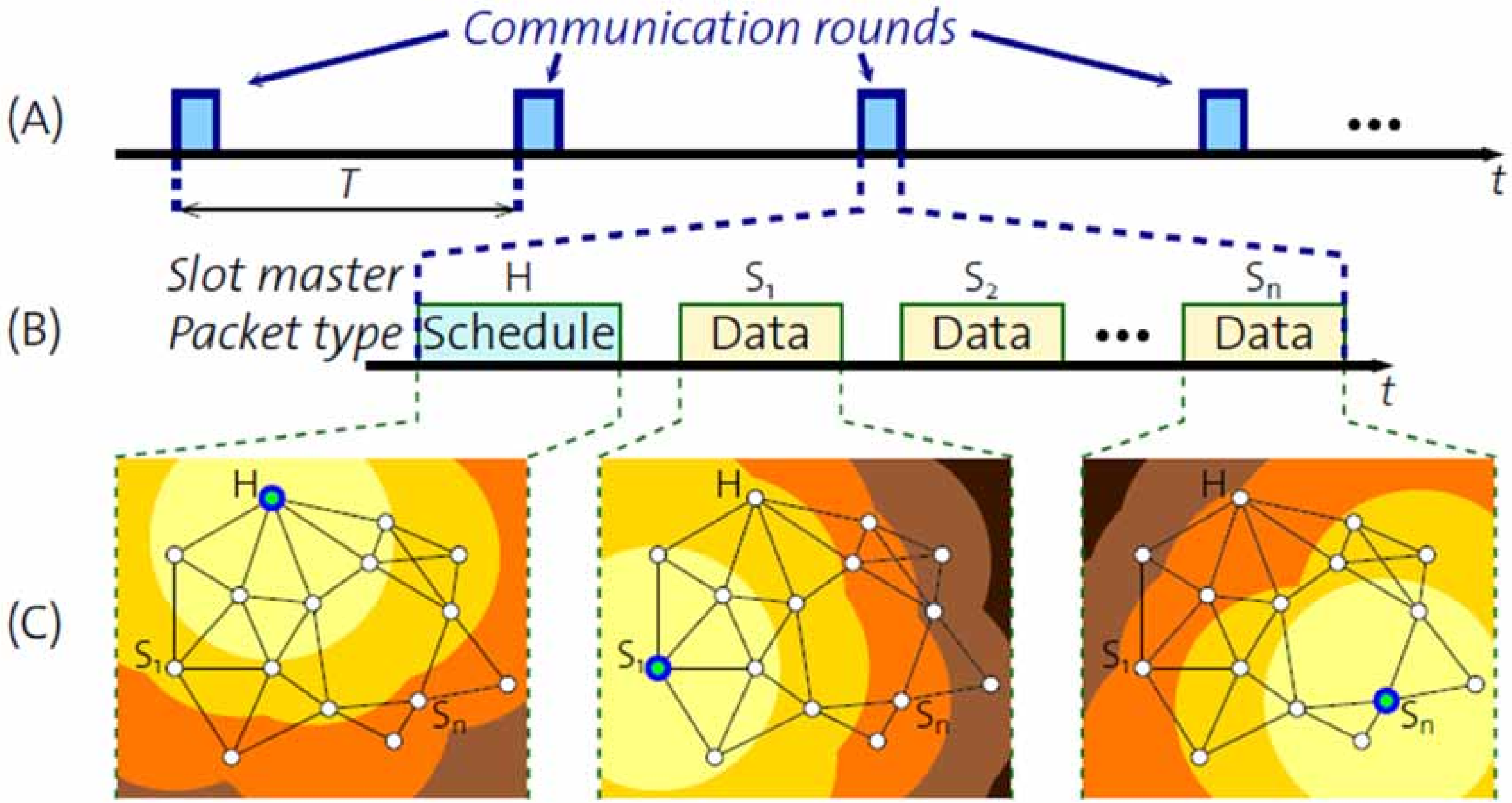
- Less than 3 ms to flood 8-byte packet to 91 receivers
- Average reliability > 99.99 % in all settings



maximum number of transmissions N: 3

Low-Power Wireless Bus

Communication organized in TDMA rounds

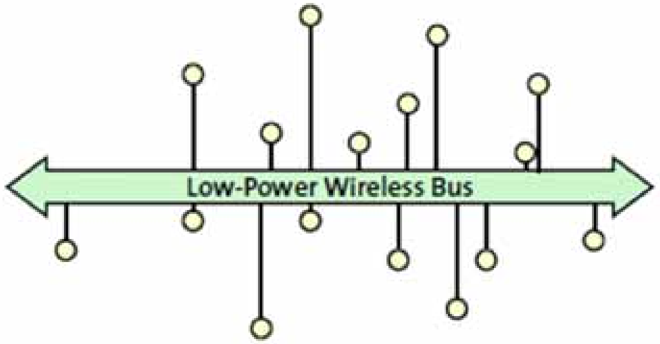


Only nodes that received the schedule participate in communication

Implications of the LWB

Routing-free communication

- No network state
  - Does not rely on knowledge of network topology (*e.g.*, neighbors)
- Simpler than existing multi-hop TDMA protocols
  - Schedules nodes instead of links
- Inherent mobility support
- Inherent support for multiple sinks



Virtual single-hop communication

- Schedule computed based only on application requirements
- Simplifies verification of system properties

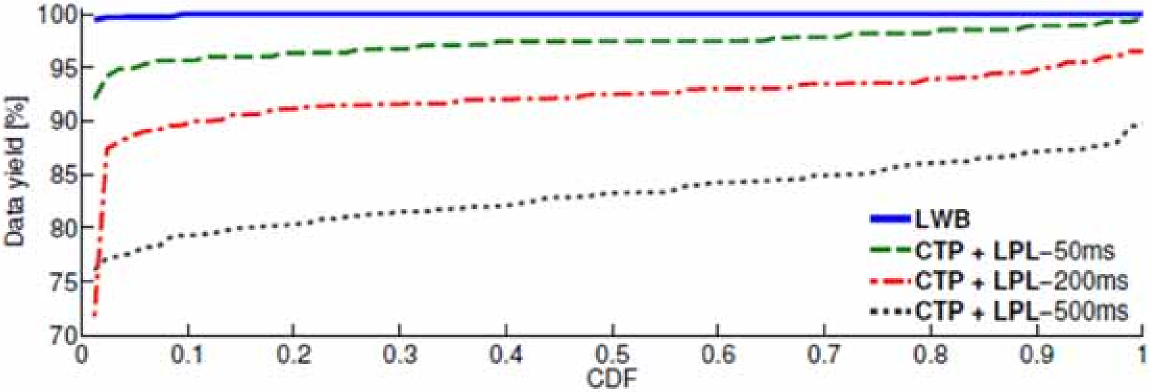
Evaluation I

One sink, low data rate, static nodes

- Twist: 85 nodes, 1 packet every minute

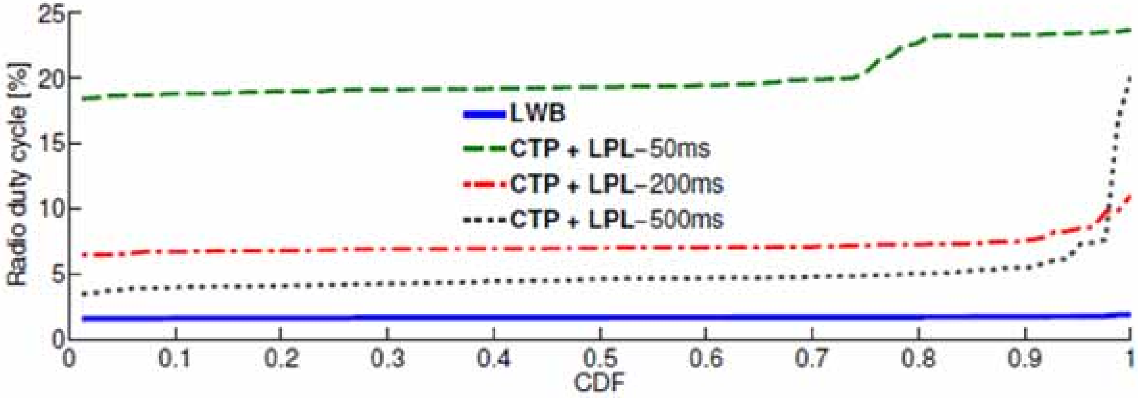
Data yield

- Avg: 99.97 %
- Min: 99.45 %

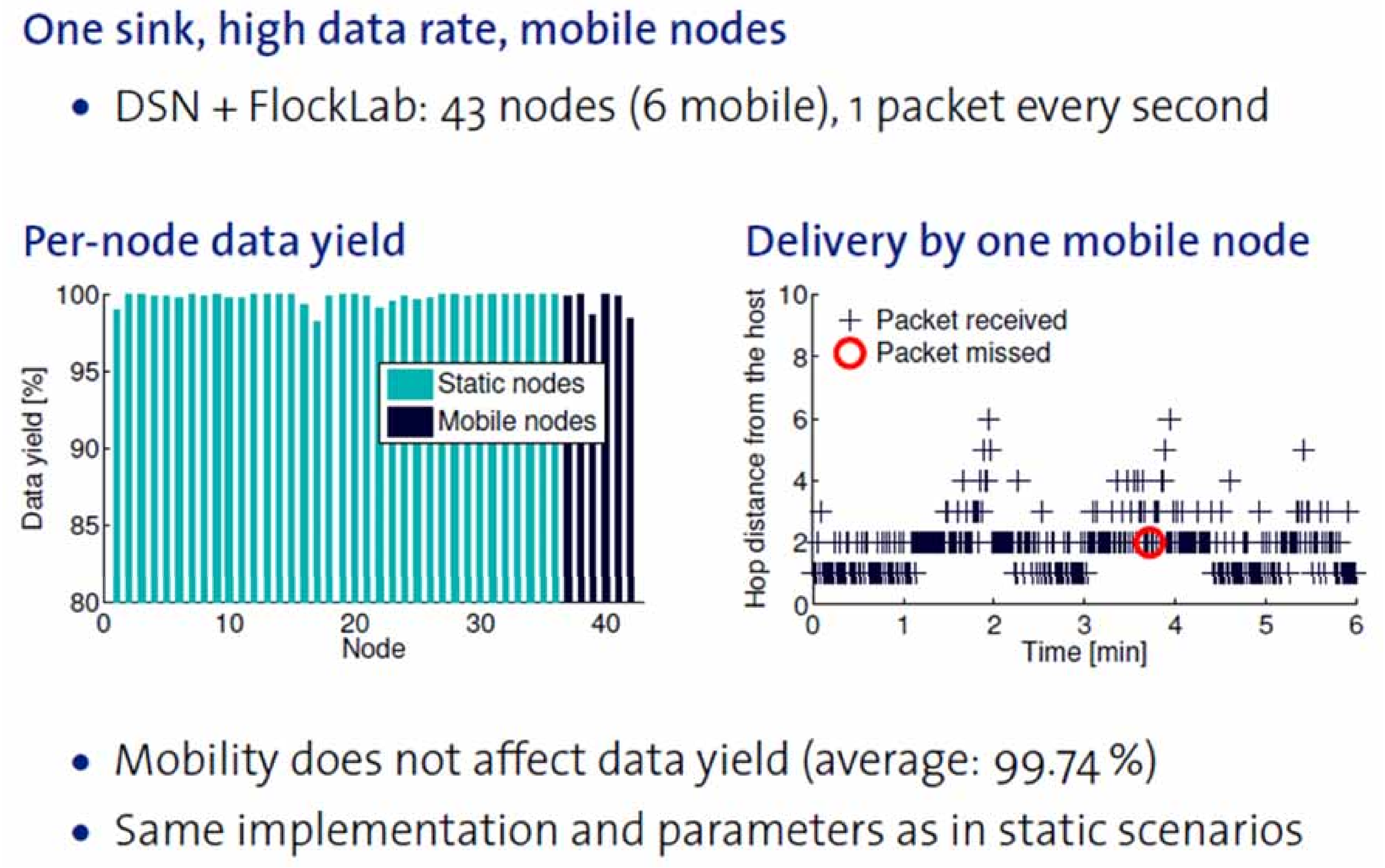


Radio duty cycle

- Avg: 1.69 %
- Max: 1.90 %



Evaluation II



One sink, high data rate, mobile nodes

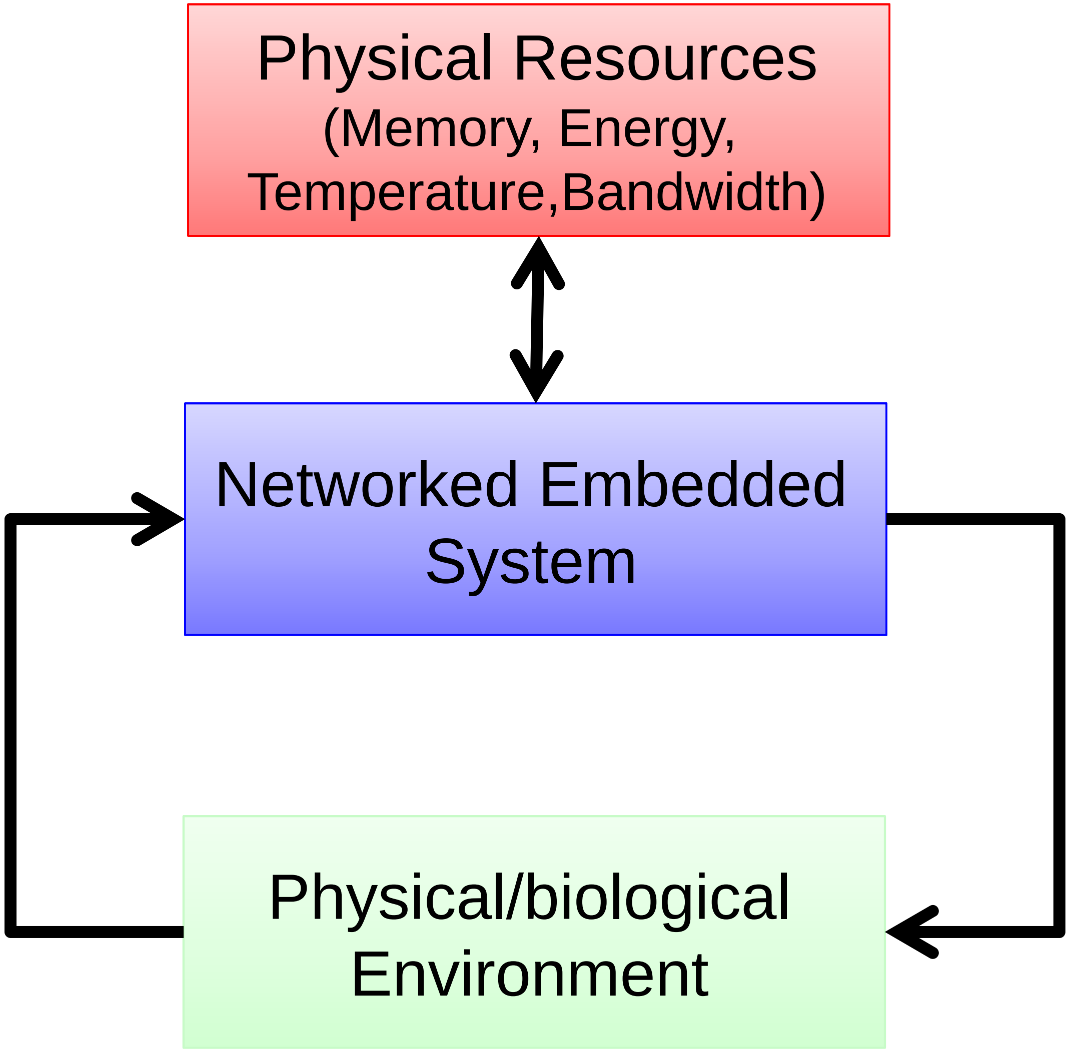
- DSN + FlockLab: 43 nodes (6 mobile), 1 packet every second

Per-node data yield

Delivery by one mobile node

- Mobility does not affect data yield (average: 99.74 %)
- Same implementation and parameters as in static scenarios

Interacting with the Environment



Resource Interaction

Physical Resources (Memory, Energy, Temperature,Bandwidth)

Networked Embedded System

Functional Interaction

Physical/biological Environment

Conclusion - Part I

- *Interesting scientific questions* arise from serious applications.

- *Serious applications* involve tremendous effort in
  - understanding environment and constraints
  - related science

- Not all extracted scientific questions found directly their way back to the application, but they helped a lot in understanding the basic problems.

Acknowlegement

- People:
  - *Jan Beutel*, Olga Saukh
  - Bernhard Buchli, Federico Ferrari, Tonio Gsell, David Hasenfratz, Matthias Keller, Roman Lim, Clemens Moser, Mustafa Yuecel, Christoph Walser, Matthias Woehrle, Marco Zimmerling
- Funding:
  - SNF Switzerland (via the NCCR MICS program)
  - BAFU (Federal Office for the Environment)
  - Competence Centre Environment and Sustainability (CCES)
  - Siemens Building Technologies
  - CTI: The Swiss Innovation Promotion Agency
  - Nano-Tera (X-Sense, OpenSense)

References

Clemens Moser, Lothar Thiele, Davide Brunelli, Luca Benini: **Adaptive Power Management for Environmentally Powered Systems**

Matthias Woehrle, Kai Lampka, Lothar Thiele: **Exploiting timed automata for conformance testing of power measurements**

Federico Ferrari, Marco Zimmerling, Lothar Thiele, Luca Mottola: **The Bus goes Wireless: Routing-Free Data Collection with QoS Guarantees in Sensor Networks**

Federico Ferrari, Marco Zimmerling, Lothar Thiele, Olga Saukh: **Efficient Network Flooding and Time Synchronization with Glossy**

Jan Beutel, Bernhard Buchli, Federico Ferrari, Matthias Keller, Lothar Thiele, Marco Zimmerling: **X-Sense: Sensing in Extreme Environments**

Karl Aberer, Saket Sathe, Dipanjan Chakraborty, Alcherio Martinoli, Guillermo Barrenetxea, Boi Faltings, Lothar Thiele: **OpenSense: open community driven sensing of environment**