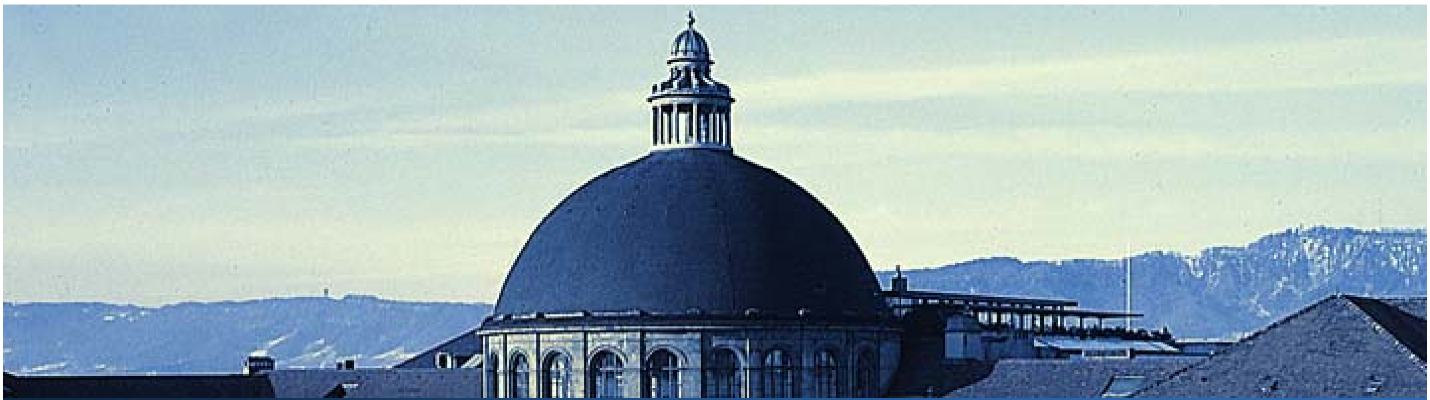
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Real-world Sensor Networks: Experiences in Design and Deployment

Jan Beutel

Summer School on Wireless Sensor Networks and Smart Objects
Schloss Dagstuhl



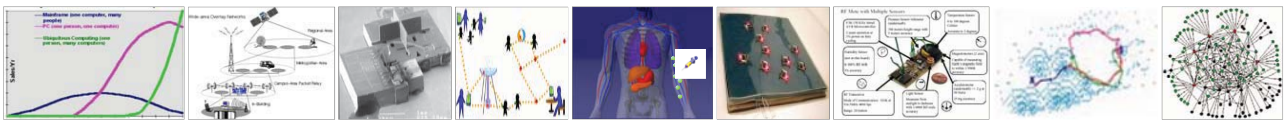
1-Sep-05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Wireless Sensor Networks

Visions 1991 1996 1999 2000 2001 2003 2004

Ubiquitous Vision, Smart Dust, Directed Diffusion, COTS Dust, Scale Free Networks

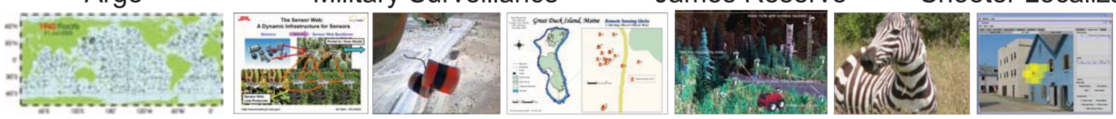


Wireless Overlay, PicoRadio, Paintable Computing, Terminodes

Applications 2000 2001 2003 2004

Argo, Military Surveillance, James Reserve, Shooter Localization

.....



Sensor Webs, Duck Island, ZebraNet

WSN – The Systems Perspective

Wireless sensor networks are not a fundamentally new area of research

- New application domain for wireless
- Limited node resources are leveraged by node collaboration and the amount of nodes
- Tight coupling of nodes, application, environment
- Broad usage profile (non-expert users)

Drawing best from other established areas/technology

- Cross-layer development

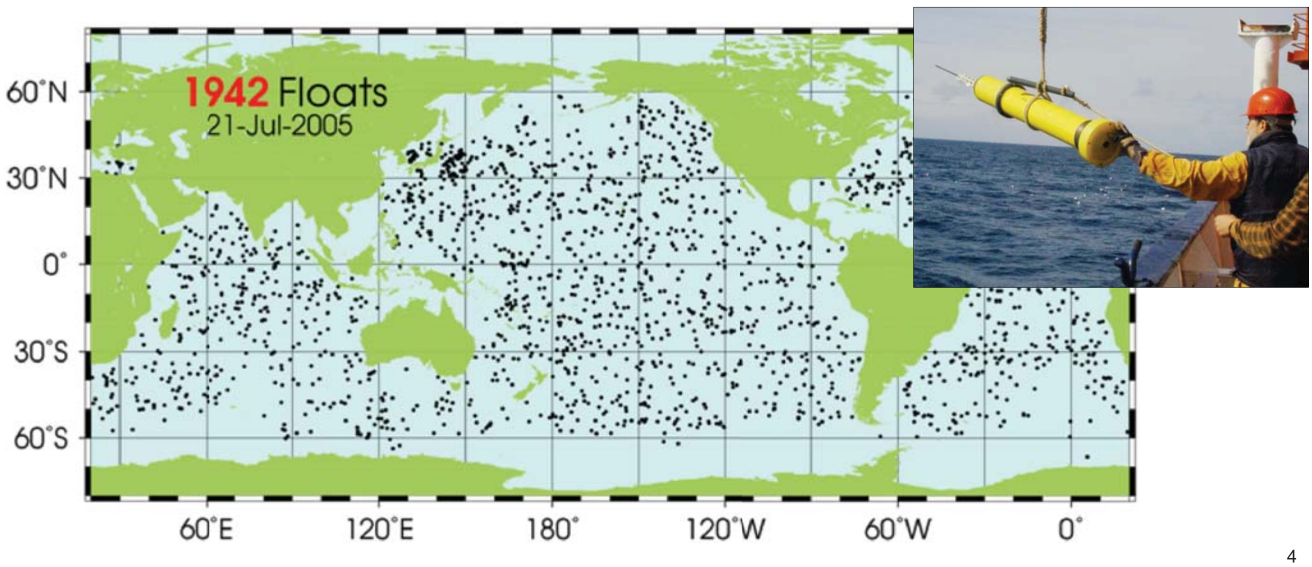
Sensor network applications have quite a long tradition.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Argo – Global Ocean Observation Strategy

Global array of temperature/salinity profiling floats

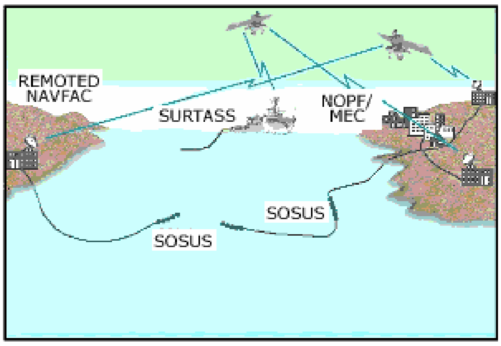
- Satellite data relay to data centers on shore
- Operational since 2000
- Developed and maintained mainly by oceanographers

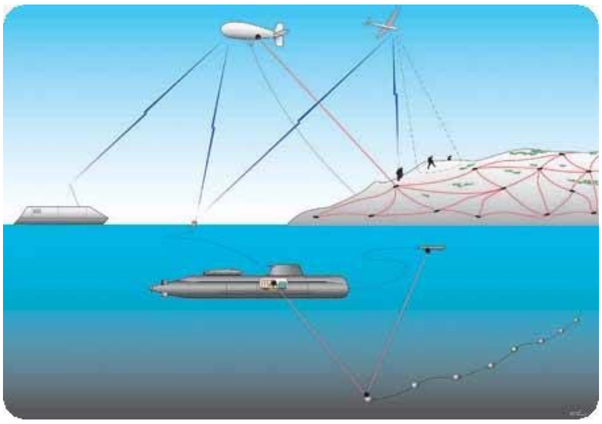


Anti-Submarine Surveillance

Distributed acoustic monitoring and surveillance

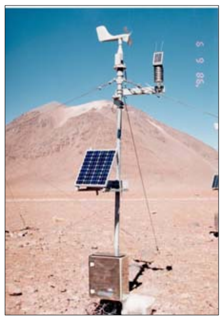
- Advanced signal processing
- Mostly wireline and analog
- Fixed installations and mobile units
- Military development since the cold war

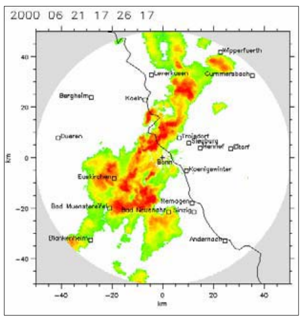




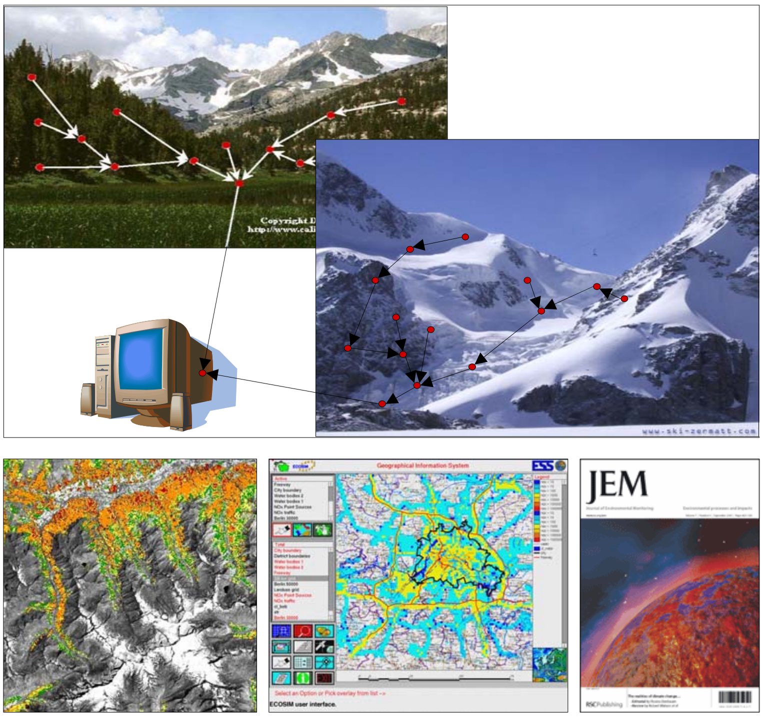
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

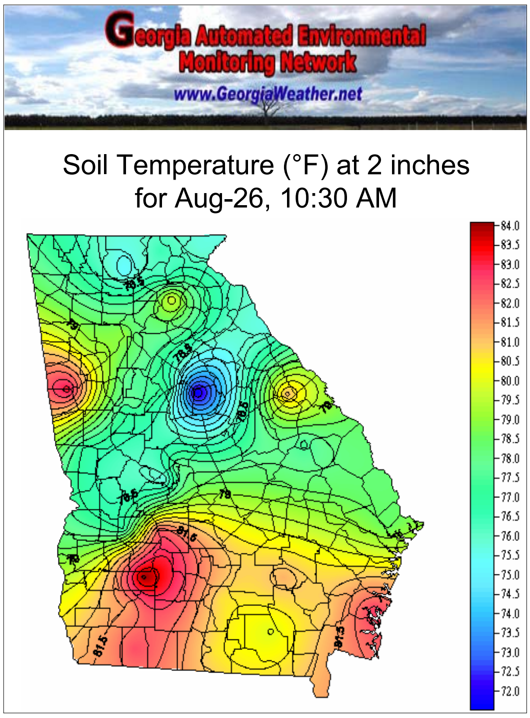
Globally Networked Weather Stations



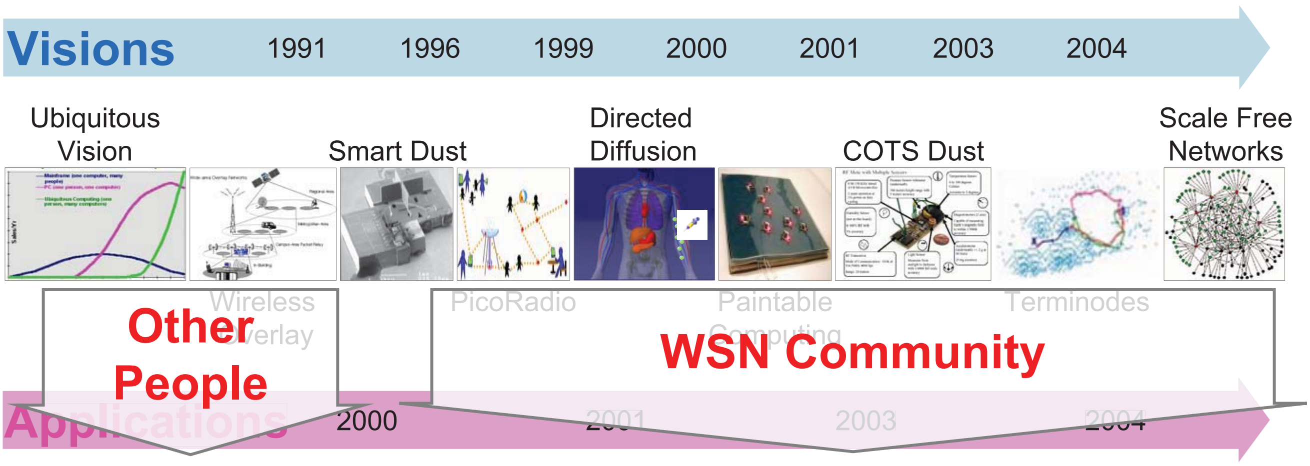


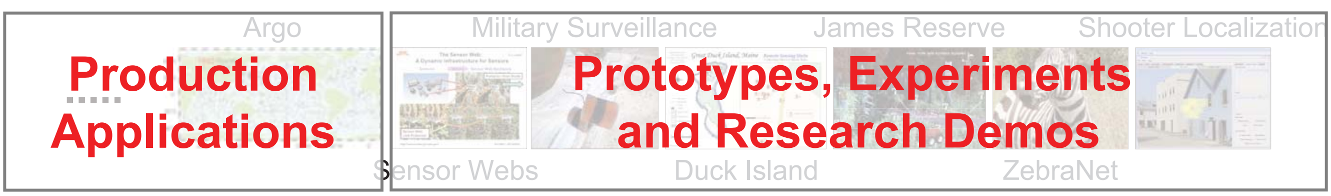
Environmental Monitoring





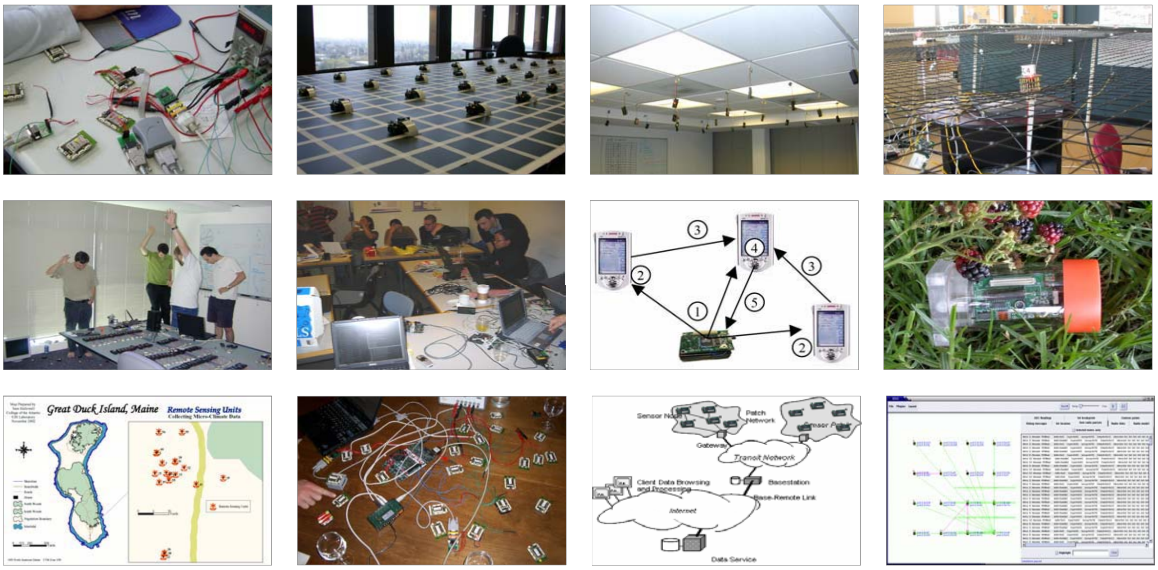
Wireless Sensor Network Applications





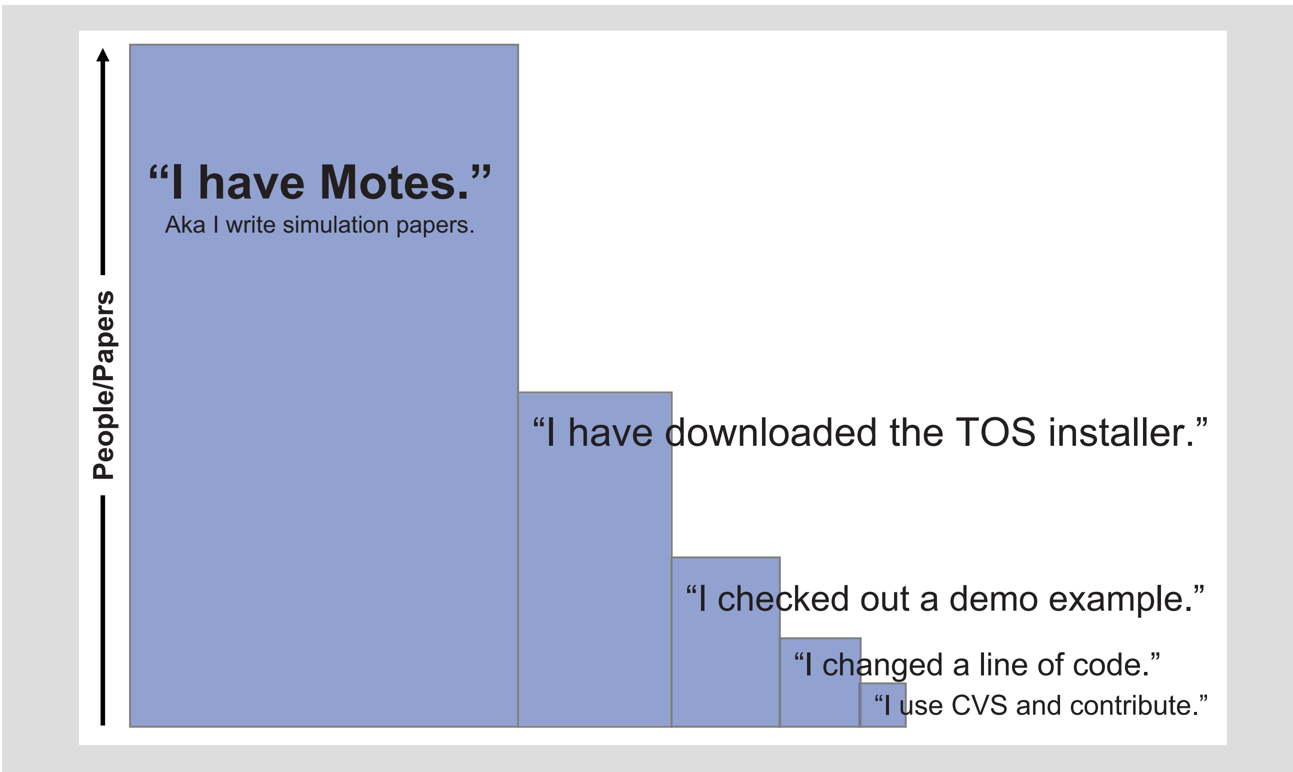
WSN Development Reality

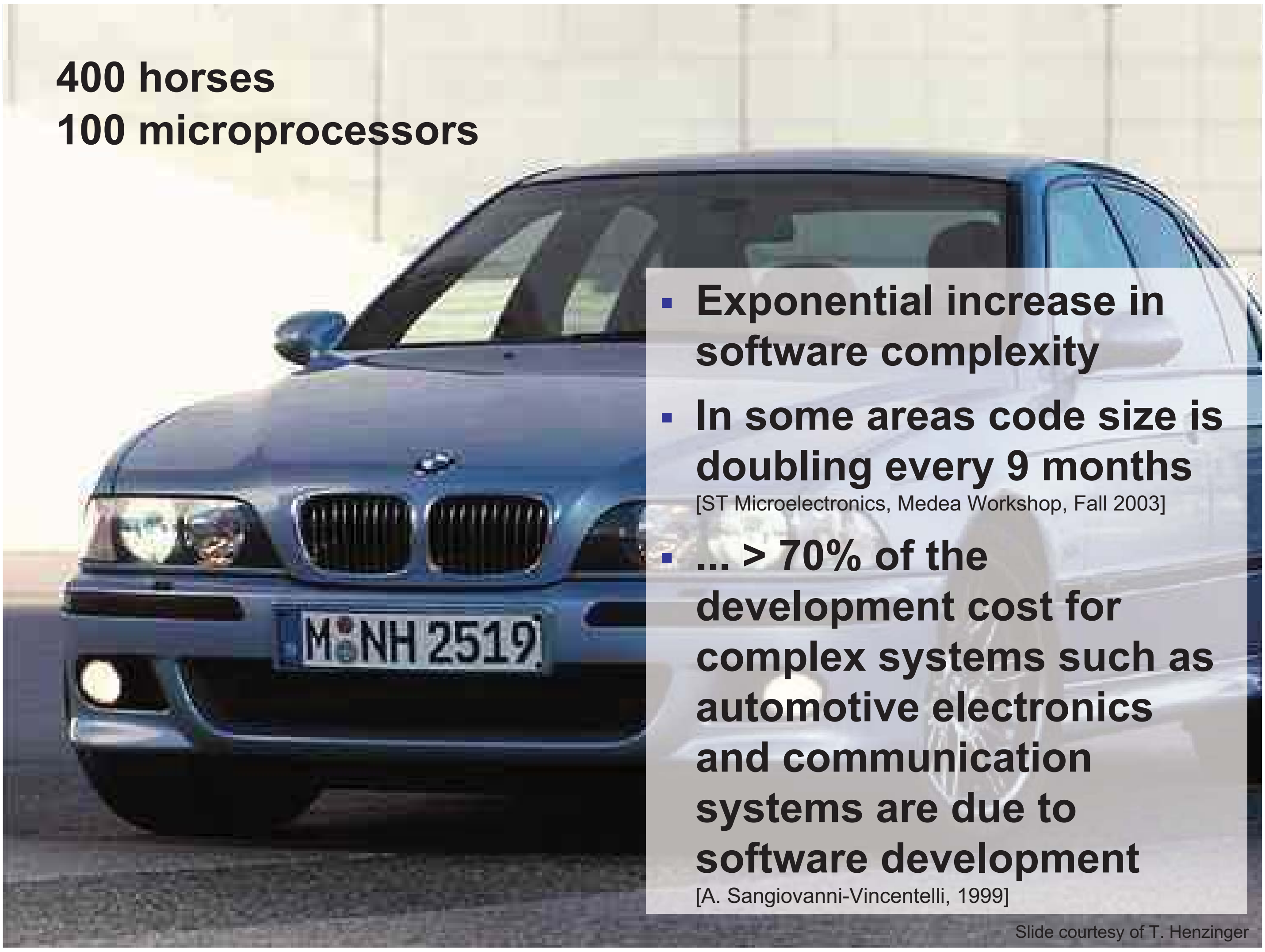
It is hard to deploy anywhere beyond 10-20 nodes today.



Coordinated methods and tools are missing today.

The WSN Evolution – Empirical Backup




400 horses
100 microprocessors
Exponential increase in software complexity
In some areas code size is doubling every 9 months
[ST Microelectronics, Medea Workshop, Fall 2003]
... > 70% of the development cost for complex systems such as automotive electronics and communication systems are due to software development
[A. Sangiovanni-Vincentelli, 1999]
M NH 2519
Slide courtesy of T. Henzinger


777
BOEING 777-300
$4 billion development effort
40% system integration & validation cost
Slide courtesy of T. Henzinger

From Proof-of-concept to Real-world WSNs

Simulation Tools

- Of theory and practice

System Design

- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

BTnode platform for fast-prototyping

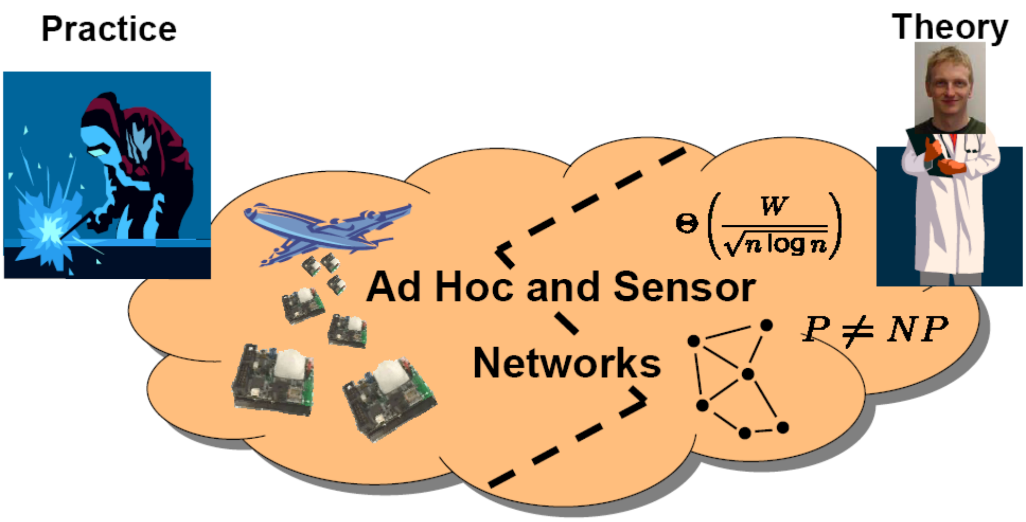
- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment

- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Of Theory and Practice...



There is often a big gap between theory and practice in the field of wireless ad hoc and sensor networks.

Slide courtesy of R. Wattenhofer

Problems of Theoretical Work and Simulation

Typical simulation papers use

- Flawed assumptions, simplifications, wrong models
  [Kotz03/04,Min2003,Heidemann2001,Ganesan2002]
- Limited comparability/reproducibility [Cavin2002]

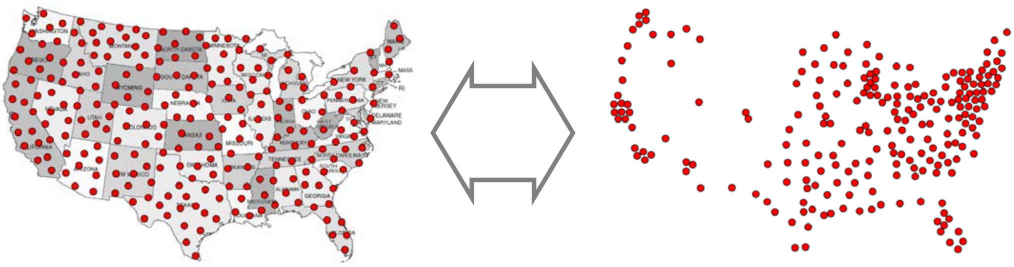
☑ **Theoreticians try to understand the fundamentals.**
☑ **Need to abstract away a few “technicalities”.**
☑ **This allows nice formulas.**

☒ **Abstracting away too many “technicalities“ renders theory useless for practice!**

Material courtesy of R. Wattenhofer

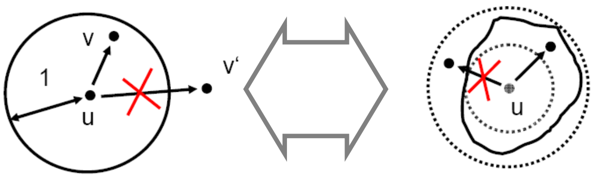
Common Assumptions in Theory and Simulation

Random, uniform node distribution



Circular radio propagation

- Unit disk graph model



Simplistic algorithms

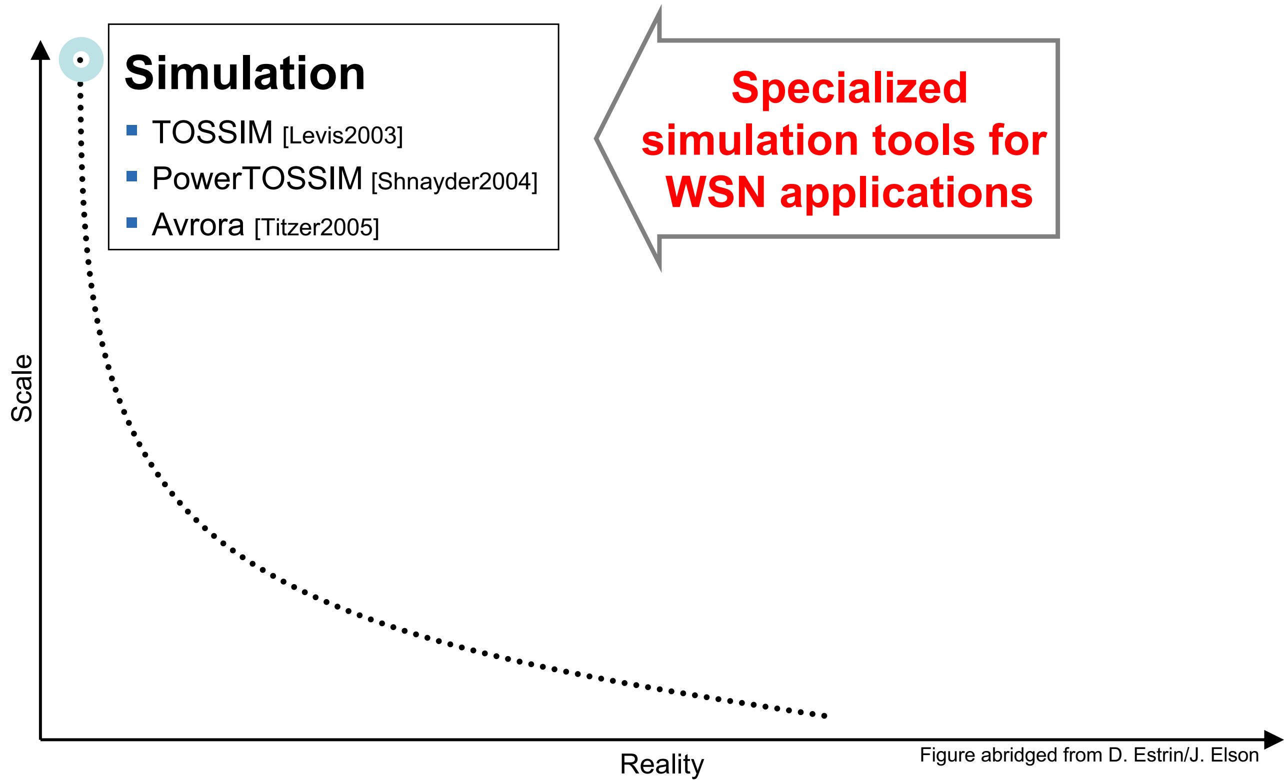
- Mac layer already in place
- Global time synchronization

6: **send** color$_i$ to all neighbors;

2: **for** $\ell := k-1$ **to** 0 **by** -1 **do**

Material courtesy of R. Wattenhofer

Today's WSN Design and Development



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

From Proof-of-concept to Real-world WSNs

Simulation Tools

- Of theory and practice

System Design

- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

BTnode platform for fast-prototyping

- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment

- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

Characteristics of Embedded Systems – (1)

Must be *dependable*:

- ***Reliability:*** *R(t)* = probability of system working correctly provided that is was working at *t*=0
- ***Maintainability:*** *M(d)* = probability of system working correctly *d* time units after error occurred.
- ***Availability:*** probability of system working at time *t*
- ***Safety:*** no harm to be caused
- ***Security:*** confidential and authentic communication

Even perfectly designed systems can fail if the assumptions about the workload and possible errors turn out to be wrong.

Making the system dependable must not be an after-thought, it must be considered from the very beginning.

Characteristics of Embedded Systems – (2)

Must be *efficient:*

- ***Energy*** efficient
- ***Code-size*** efficient (especially for systems on a chip)
- ***Run-time*** efficient
- ***Weight*** efficient
- ***Cost*** efficient

Dedicated towards a certain *application*

- Knowledge about behavior at design time can be used to minimize resources and to maximize robustness.

Characteristics of Embedded Systems – (3)

Many ES must meet real-time constraints:

- A real-time system must **react to stimuli** from the controlled object (or the operator) within the time interval dictated by the environment.
- For real-time systems, right answers arriving too late (or even too early) are wrong.

„A real-time constraint is called hard, if not meeting that constraint could result in a catastrophe.“ [Kopetz, 1997]

- All other time-constraints are called soft.
- A **guaranteed system response** has to be explained without statistical arguments.

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Characteristics of Embedded Systems – (4)

Frequently *connected to physical environment*

- Through sensors and actuators
- ***Hybrid systems*** (analog + digital parts).

Typically, ES are *reactive systems*:

„A reactive system is one which is in continual interaction with is environment and executes at a pace determined by that environment“ [Bergé, 1995]

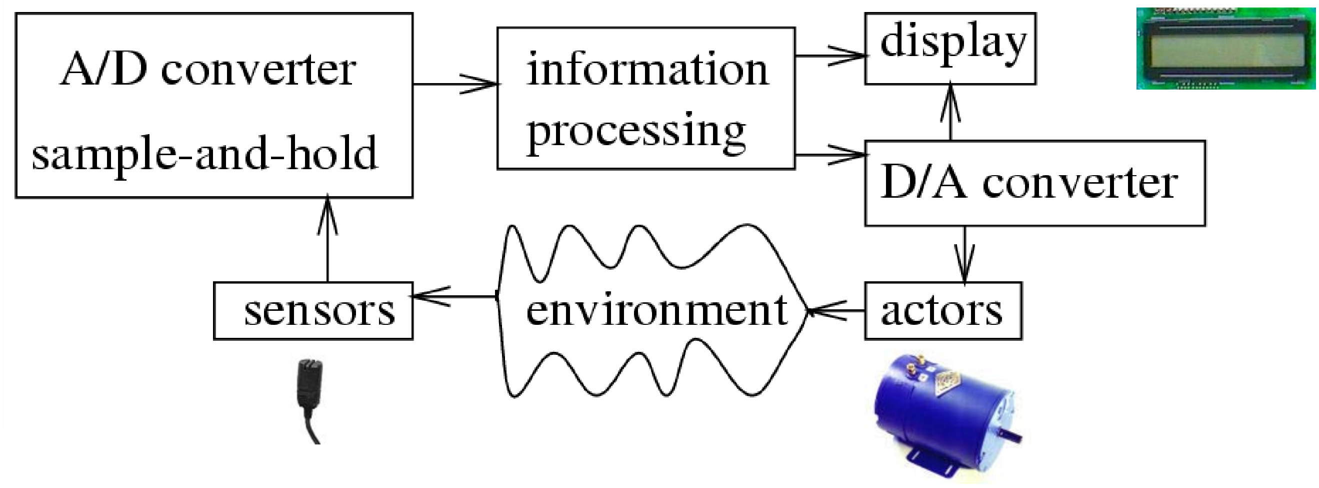
- Behavior depends on input and current state.

  ☞ automata model are often appropriate.

Embedded System Hardware-in-the-loop

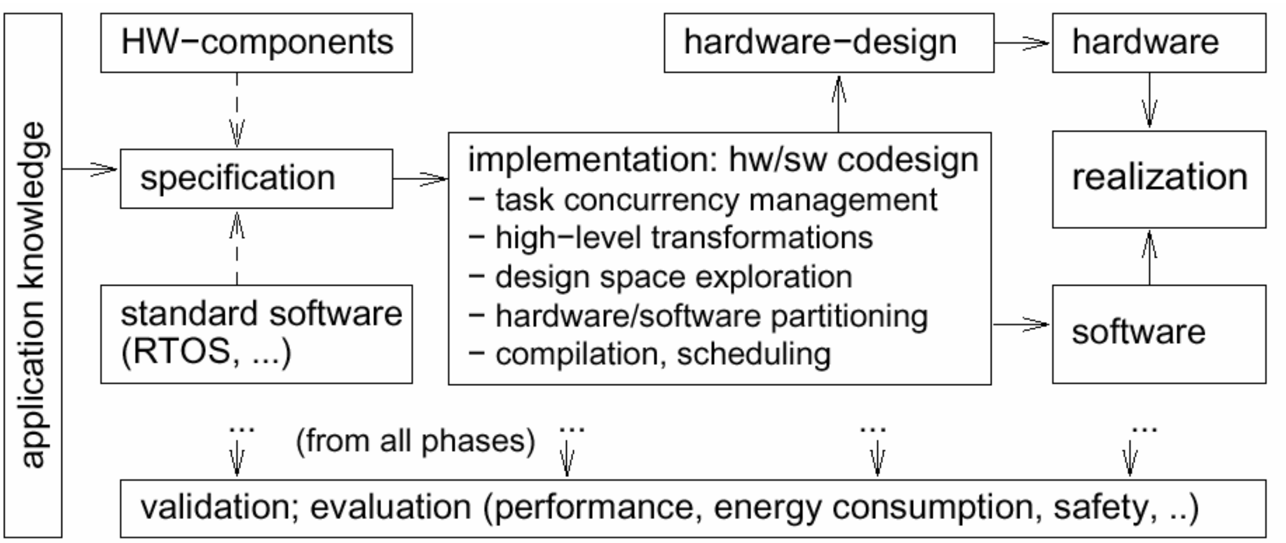
Embedded system hardware is frequently used as

- Hardware-in-a-loop



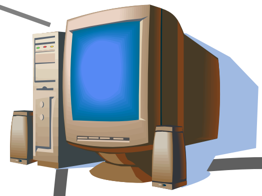
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Simple Embedded Systems Design Flow

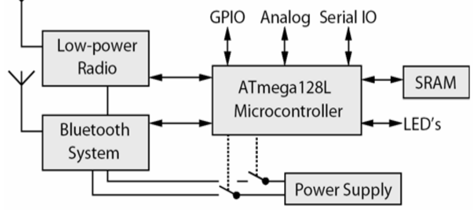


Classical Embedded Development

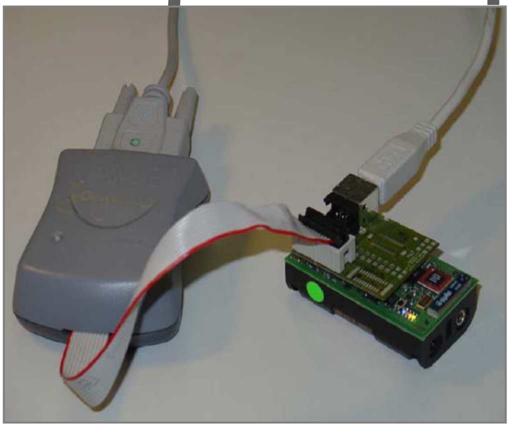
Developer Workstation
- Windows XP
- GNU GCC
- AVR libc
- Eclipse

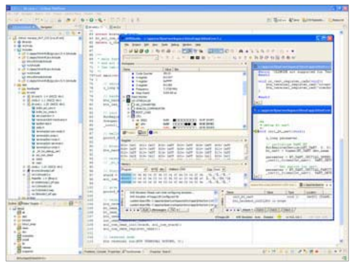


Datasheets



A Platform
- Devel Kit
- Nodes
- Programmers
- Adapter boards
- Sensors
- Housings
- Batteries
- Chargers
- …
- …
- Serial cables





Software/Tools

From Proof-of-concept to Real-world WSNs

Simulation Tools
- Of theory and practice

System Design
- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

BTnode platform for fast-prototyping
- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment
- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

The MARS Pathfinder problem – (1)

- VxWorks provides preemptive priority scheduling of threads. Tasks on the Pathfinder spacecraft were executed as threads with priorities assigned in the usual manner reflecting the relative urgency of tasks.
- Pathfinder contained an “information bus”, which you can think of as a shared memory area used for passing information between different components of the spacecraft.

A bus management task ran frequently with high priority to move certain kinds of data in and out of the information bus. Access to the bus was synchronized with mutual exclusion locks (mutexes).

The MARS Pathfinder problem – (2)

- The meteorological data gathering task ran as an infrequent, low priority thread… When publishing its data, it would acquire a mutex, do writes to the bus, and release the mutex.
- The spacecraft also contained a communications task that ran with medium priority.

High priority:	retrieval of data from shared memory
Medium priority:	communications task
Low priority:	thread collecting meteorological data

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The MARS Pathfinder problem – (3)

- Most of the time this combination worked fine. However, very infrequently it was possible for an interrupt to occur that caused the (medium priority) communications task to be scheduled during the short interval while the (high priority) information bus thread was blocked waiting for the (low priority) meteorological data thread. In this case, the long-running communications task, having higher priority than the meteorological task, would prevent it from running, consequently preventing the blocked information bus task from running.

After some time had passed, a watchdog timer would go off, notice that the data bus task had not been executed for some time, conclude that something had gone drastically wrong, and initiate a total system reset.

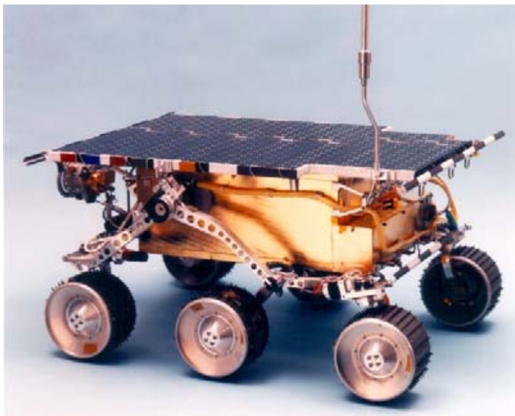
This scenario is a classic case of priority inversion.

Priority inversion on Mars

Priority inheritance solved the Mars Pathfinder problem

- The VxWorks operating system used in the pathfinder implements a flag for the calls to mutex primitives. This flag allows priority inheritance to be set to “on”.
- When the software was shipped, it was set to “off”.

The problem on Mars was corrected by using the debugging facilities of VxWorks to change the flag to “on”, while the Pathfinder was already on the Mars.
[Jones, 1997]



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

From Proof-of-concept to Real-world WSNs

Simulation Tools
- Of theory and practice

System Design
- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

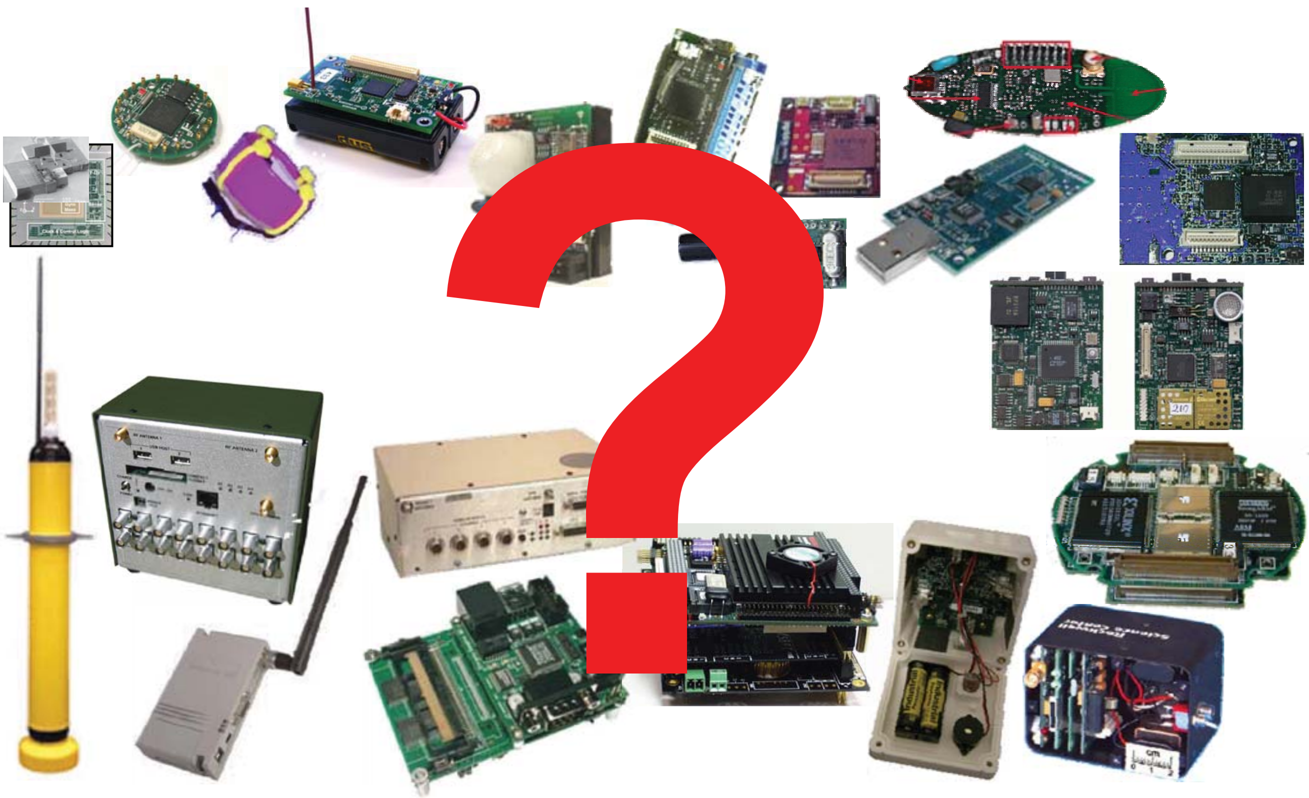
BTnode platform for fast-prototyping
- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment
- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

Wireless Sensor Network Systems Today

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Selecting the Best Platform?



Selecting the Best Platform?

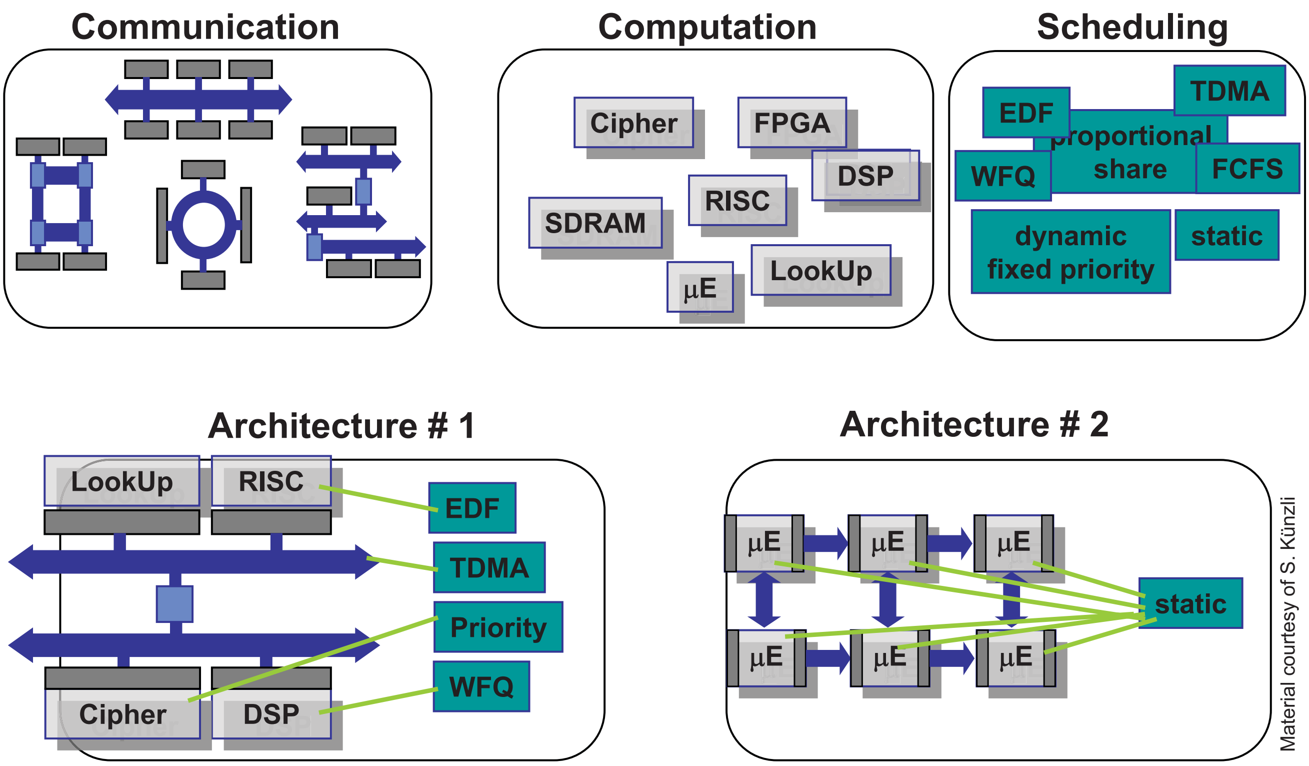
Semi-automatic Design Space Exploration

Finding the best set of resources for a given application.

Can be used
- Before building a platform
- To select from available platforms
- For many multi-criteria systems optimization problems

Results
- Set of pareto-optimal design variants that meet the specification

Design Space Exploration – Example NP

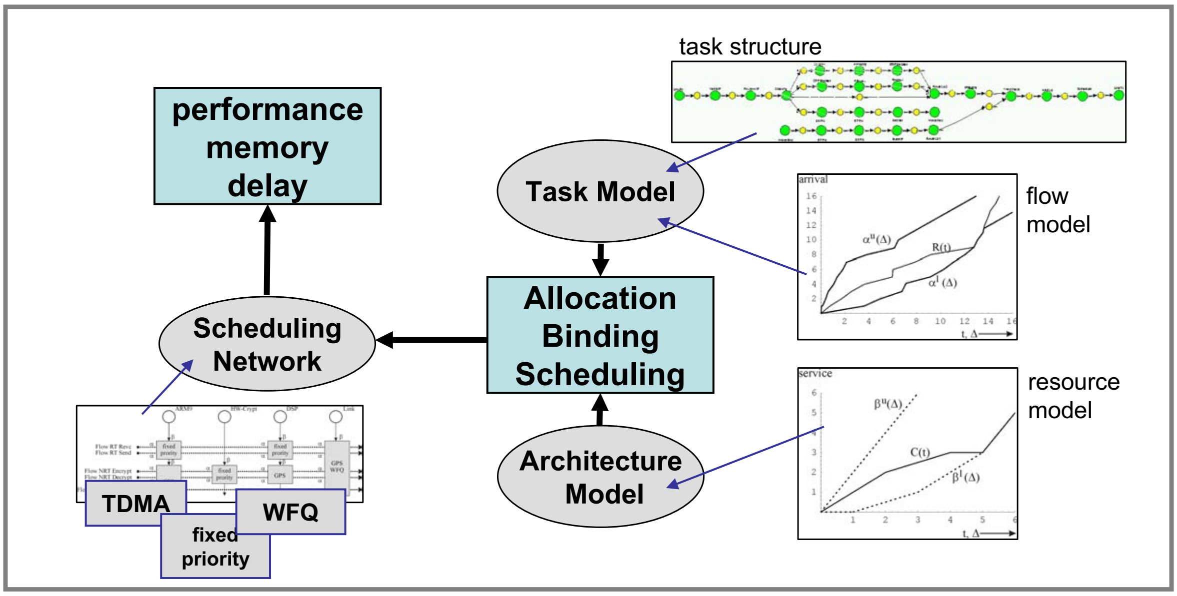


Material courtesy of S. Künzli

EXPO – Design Evaluation Cycle Example

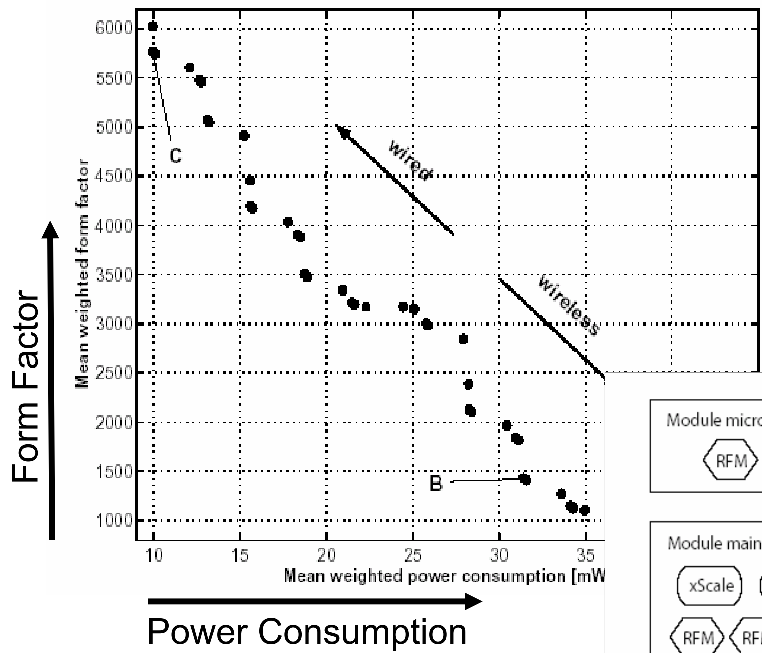
Semi-auto design space exploration

- Application to network processors [Künzli2005]



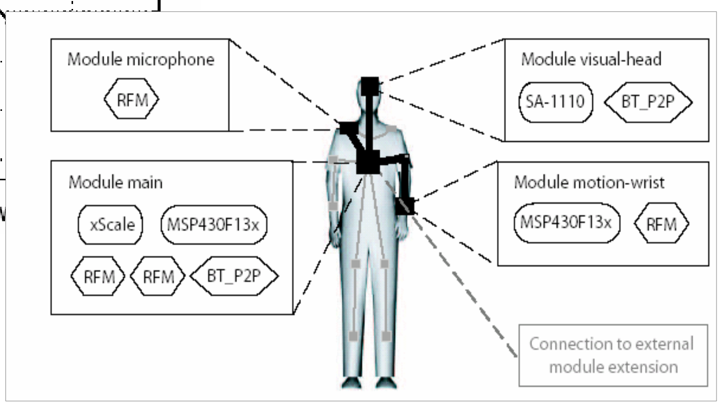
Material courtesy of S. Künzli

Design Space Exploration – Results



Example results

- From wearable computing [Anliker2004]
- Fast search using evolutionary algorithms



From Proof-of-concept to Real-world WSNs

Simulation Tools
- Of theory and practice

System Design
- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

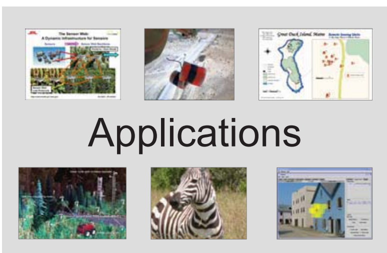
BTnode platform for fast-prototyping
- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment
- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

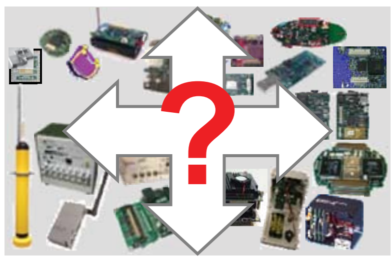
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Metrics of WSN Platforms

Large application domain
- No unified one-size-fits-all solution [Römer2004]



Good platform?
Suitable solution?
Optimum match?



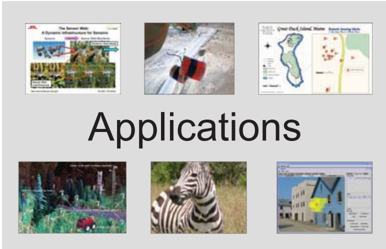
Automated tools common in EDA community
- E.g. semi-automatic design space exploration [Künzli2005,Anliker2004]

Current WSN community approach
- Device characterization, e.g. Mote family [Polastre2005,Shnayder2004]
- Tiered architectures [Estrin2003], WSN device classes [Hill2004]

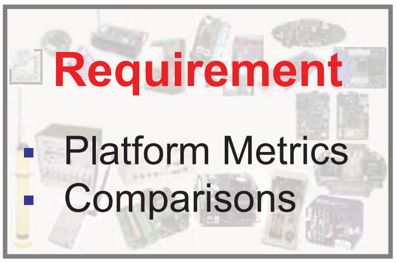
Metrics of WSN Platforms

Large application domain

- No unified one-size-fits-all solution [Römer2004]



Good platform?
Suitable solution?
Optimum match?



Automated tools common in EDA community

- E.g. semi-automatic design space exploration [Künzli2005,Anliker2004]

Current WSN community approach

- Device characterization, e.g. Mote family [Polastre2005,Shnayder2004]
- Tiered architectures [Estrin2003], WSN device classes [Hill2004]

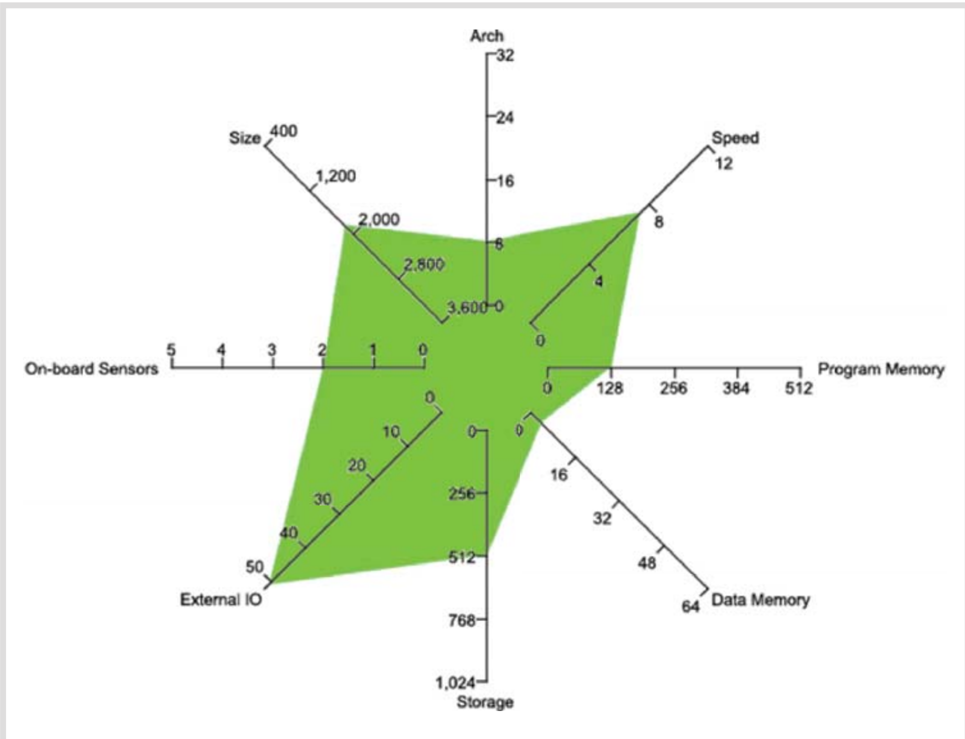
State-of-the-Art Platforms – System Core


Mica2


Mica2Dot


Tmote Sky


Imote



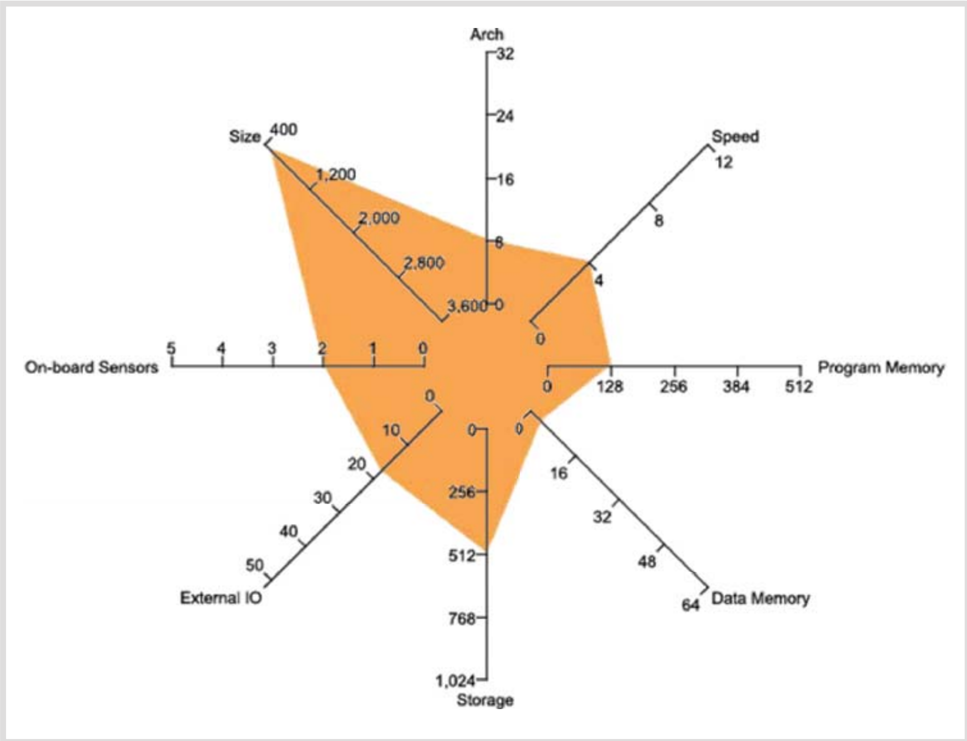
State-of-the-Art Platforms – System Core


Mica2


Mica2Dot


Tmote Sky


Imote



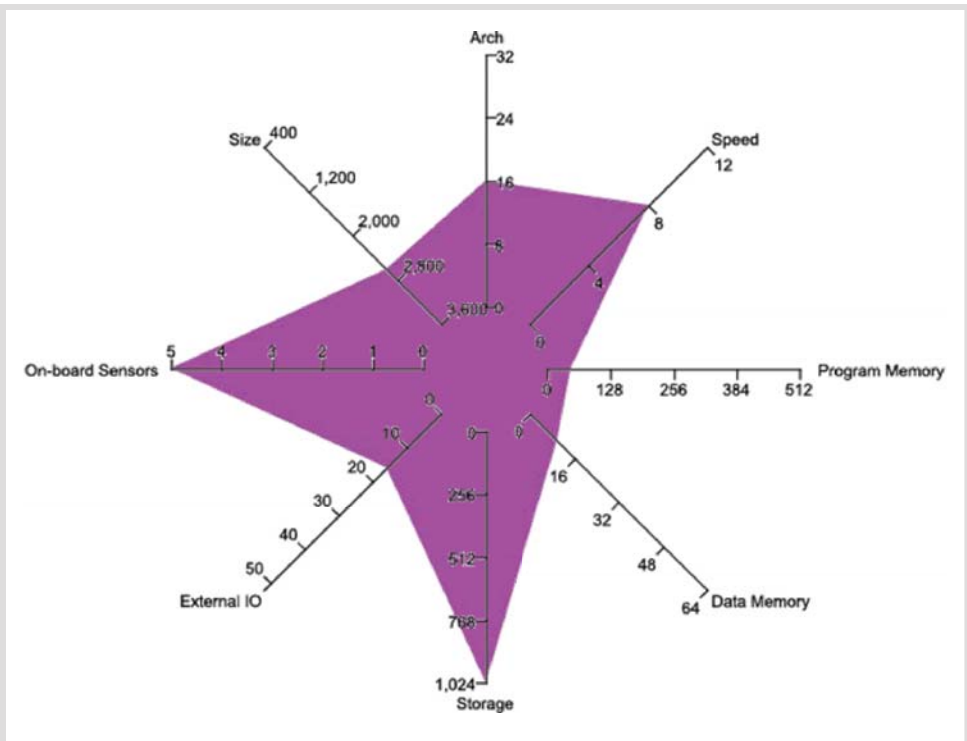
State-of-the-Art Platforms – System Core


Mica2


Mica2Dot

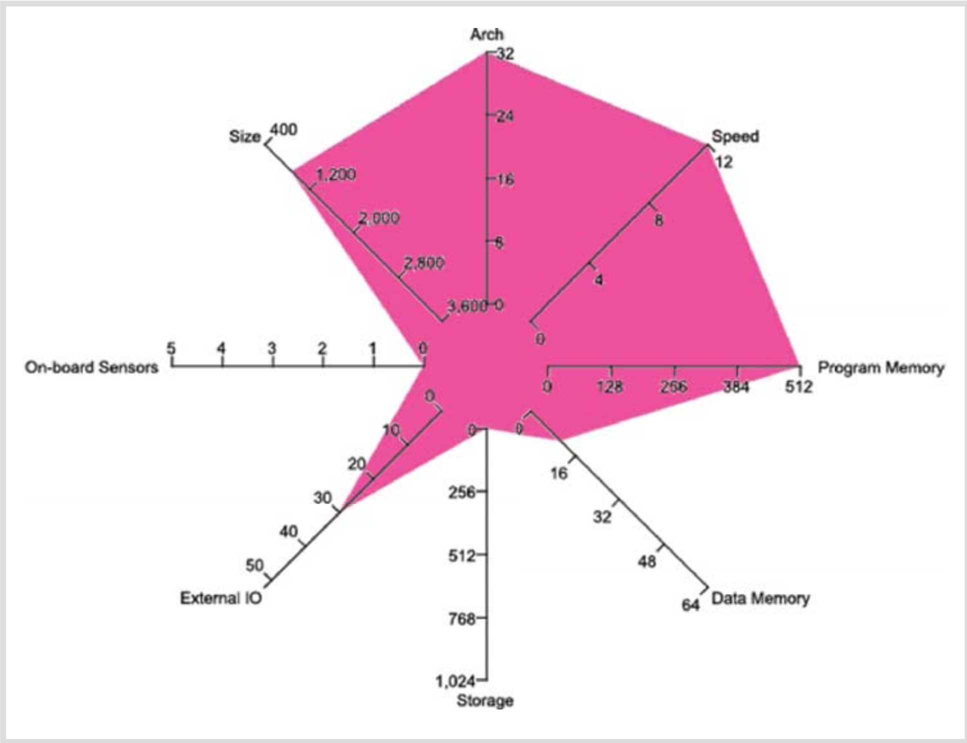

Tmote Sky


Imote



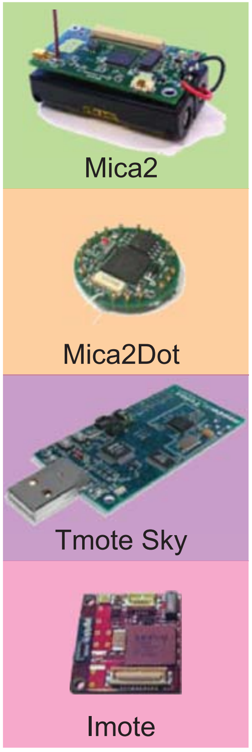
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

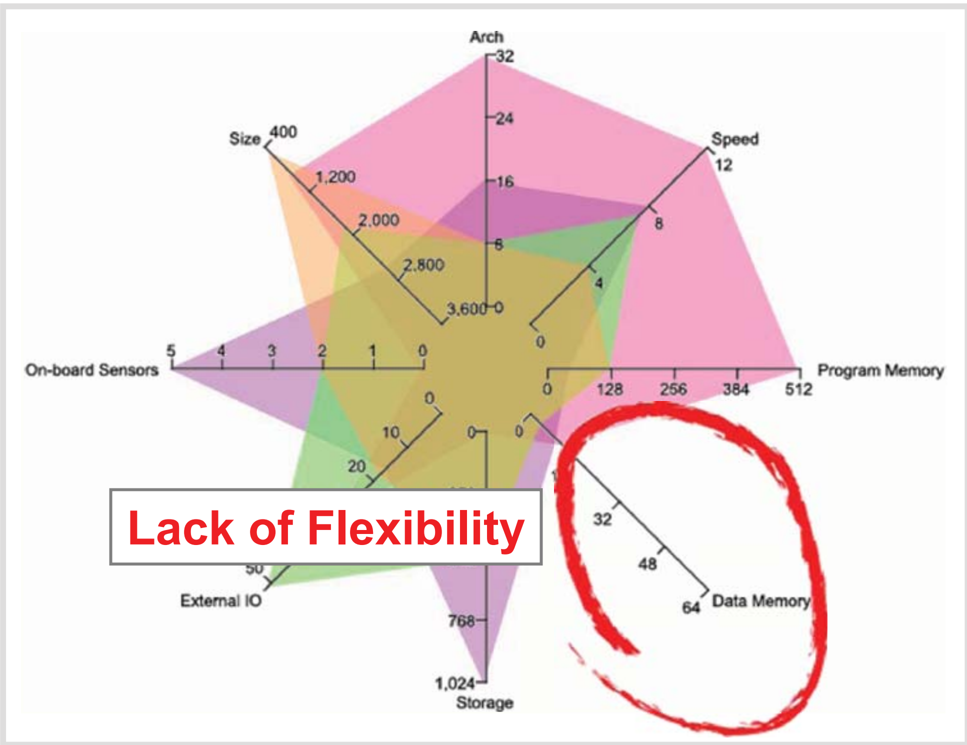
State-of-the-Art Platforms – System Core



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

State-of-the-Art Platforms – System Core





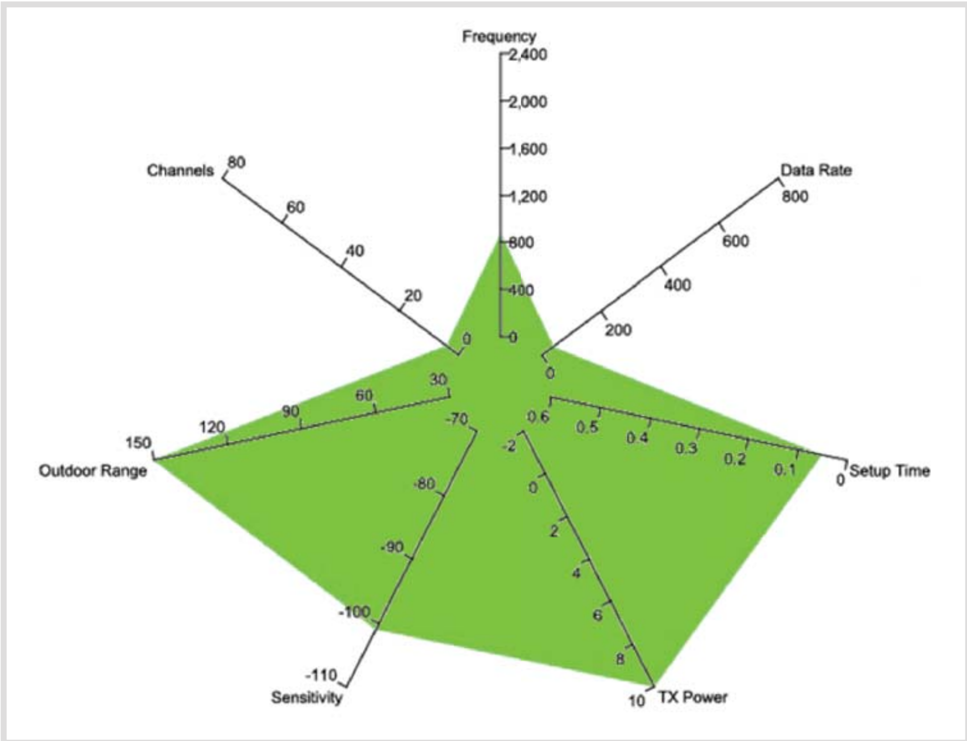
State-of-the-Art Platforms – Radio Systems

Mica2

Mica2Dot

Tmote Sky

Imote



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

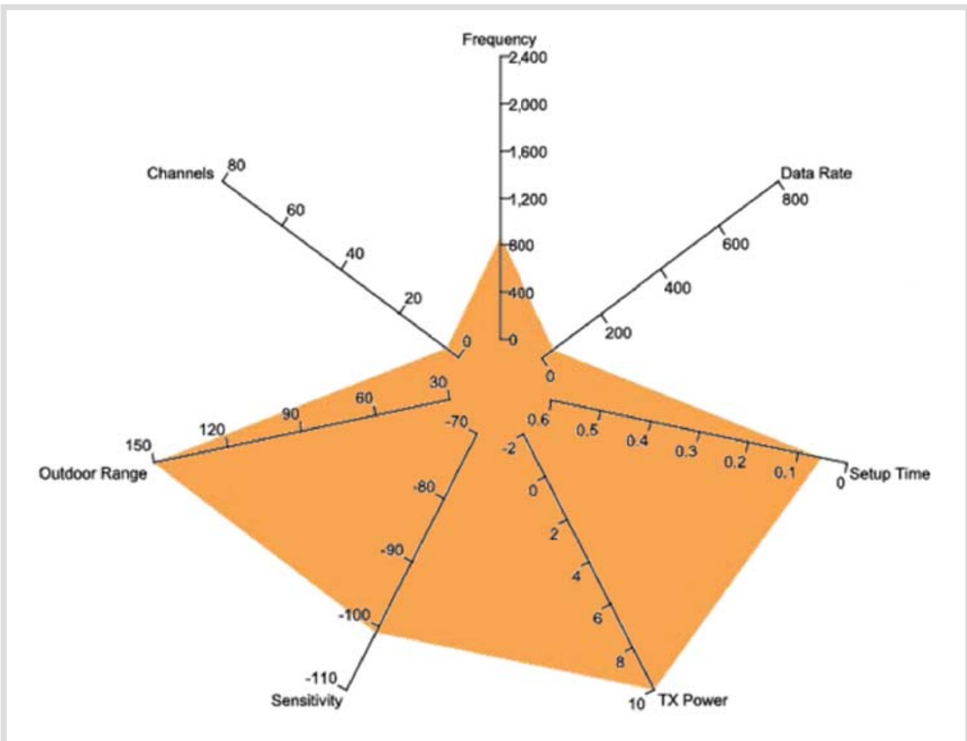
State-of-the-Art Platforms – Radio Systems

Mica2

Mica2Dot

Tmote Sky

Imote



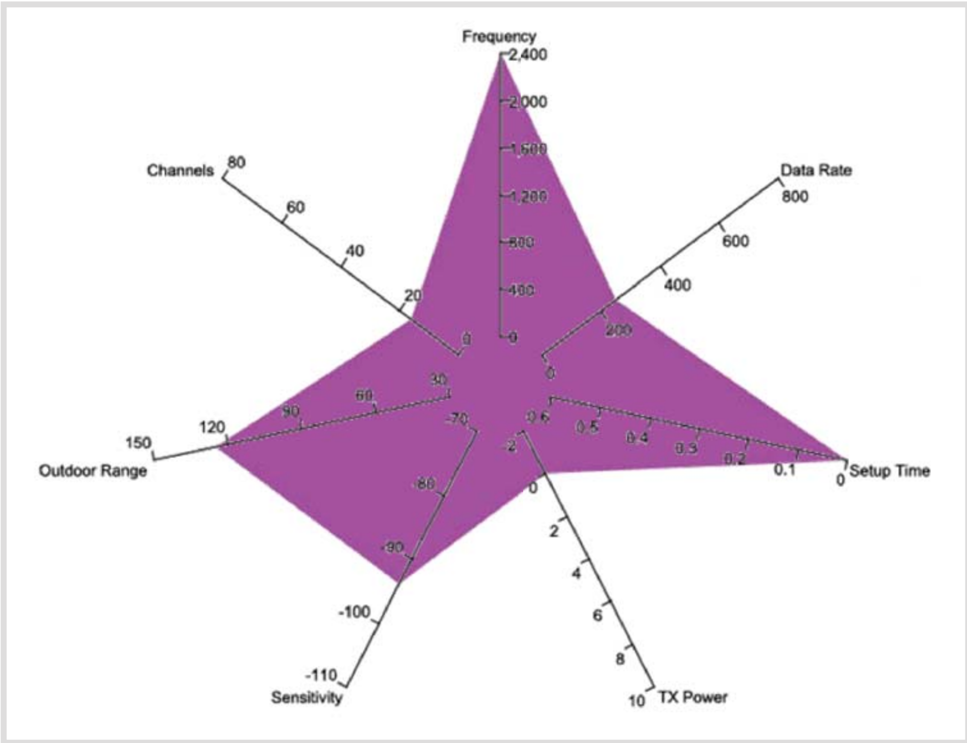
State-of-the-Art Platforms – Radio Systems

Mica2

Mica2Dot

Tmote Sky

Imote



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

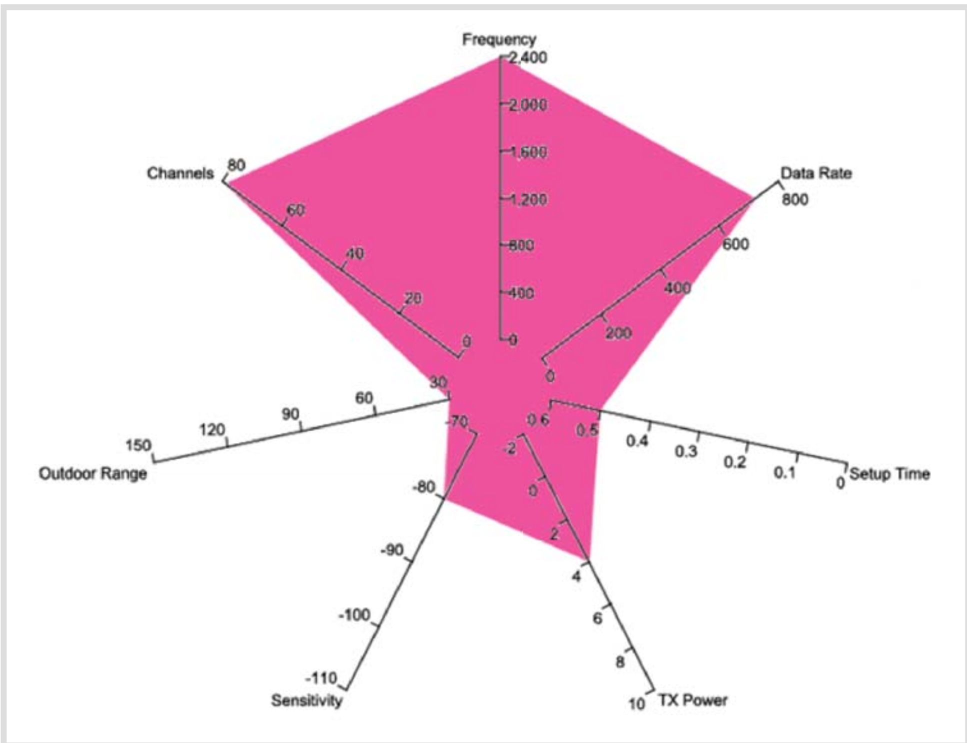
State-of-the-Art Platforms – Radio Systems

Mica2

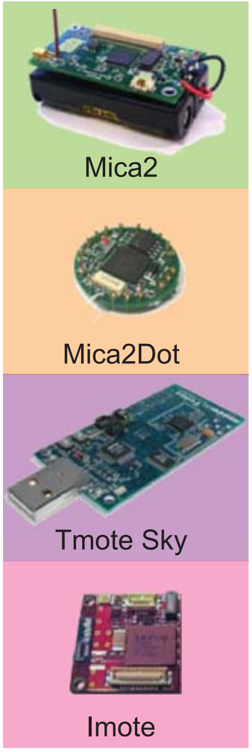
Mica2Dot

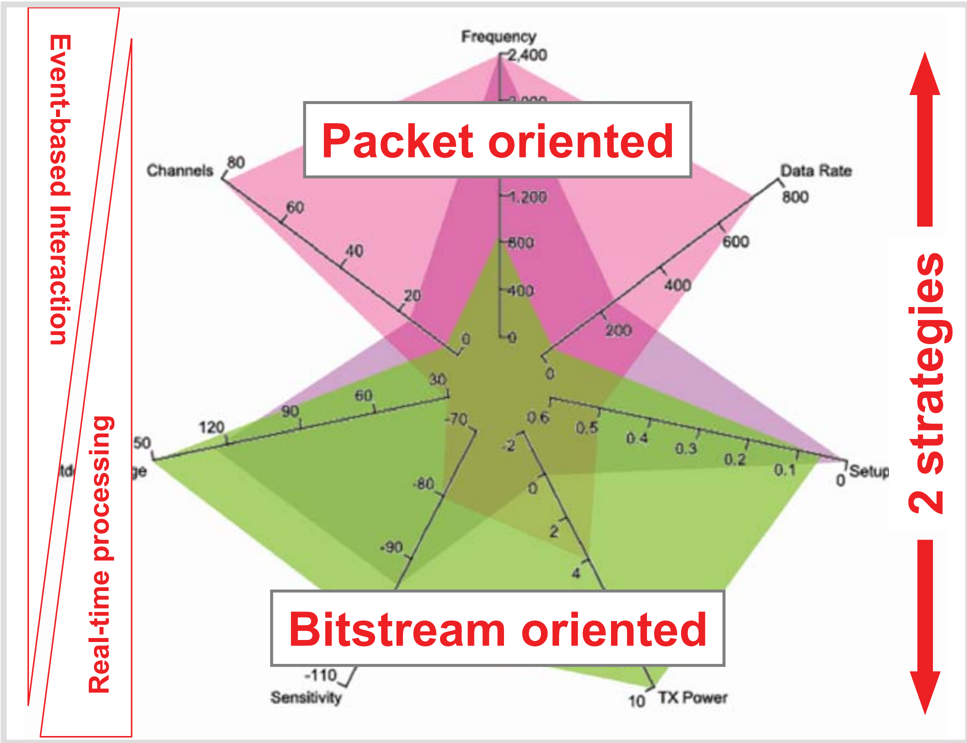
Tmote Sky

Imote



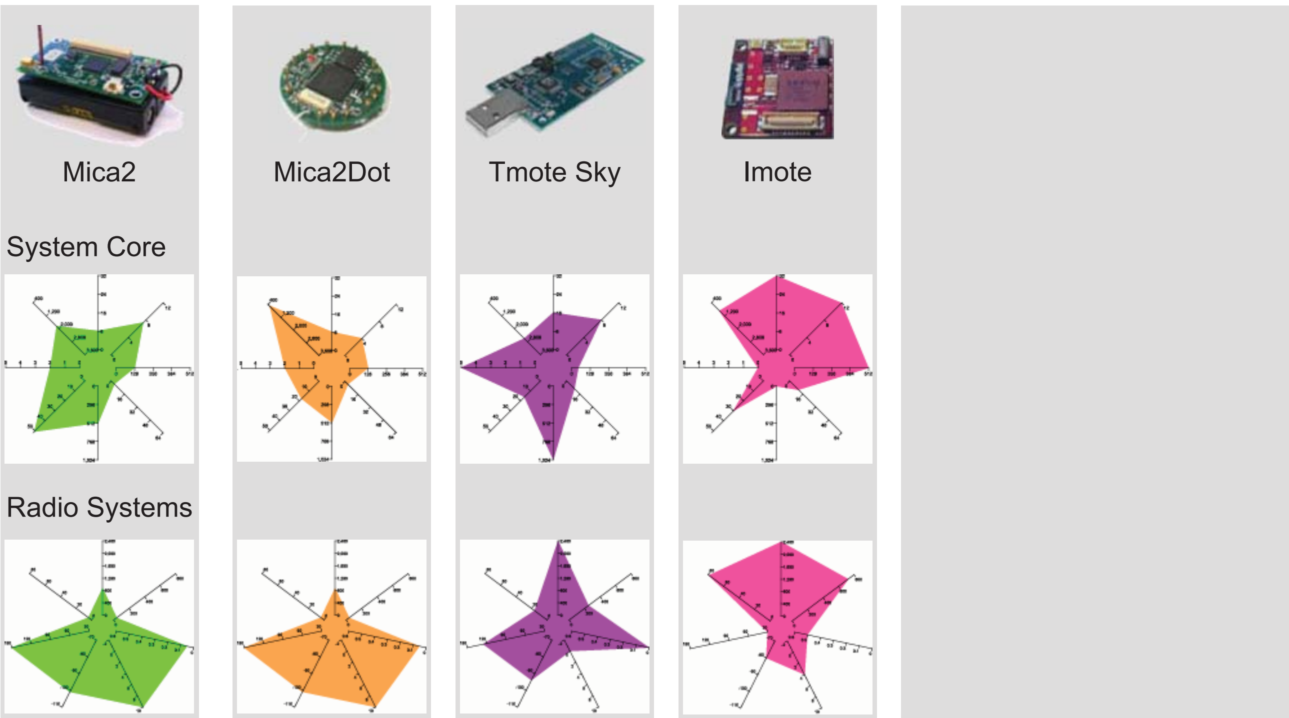
State-of-the-Art Platforms – Radio Systems



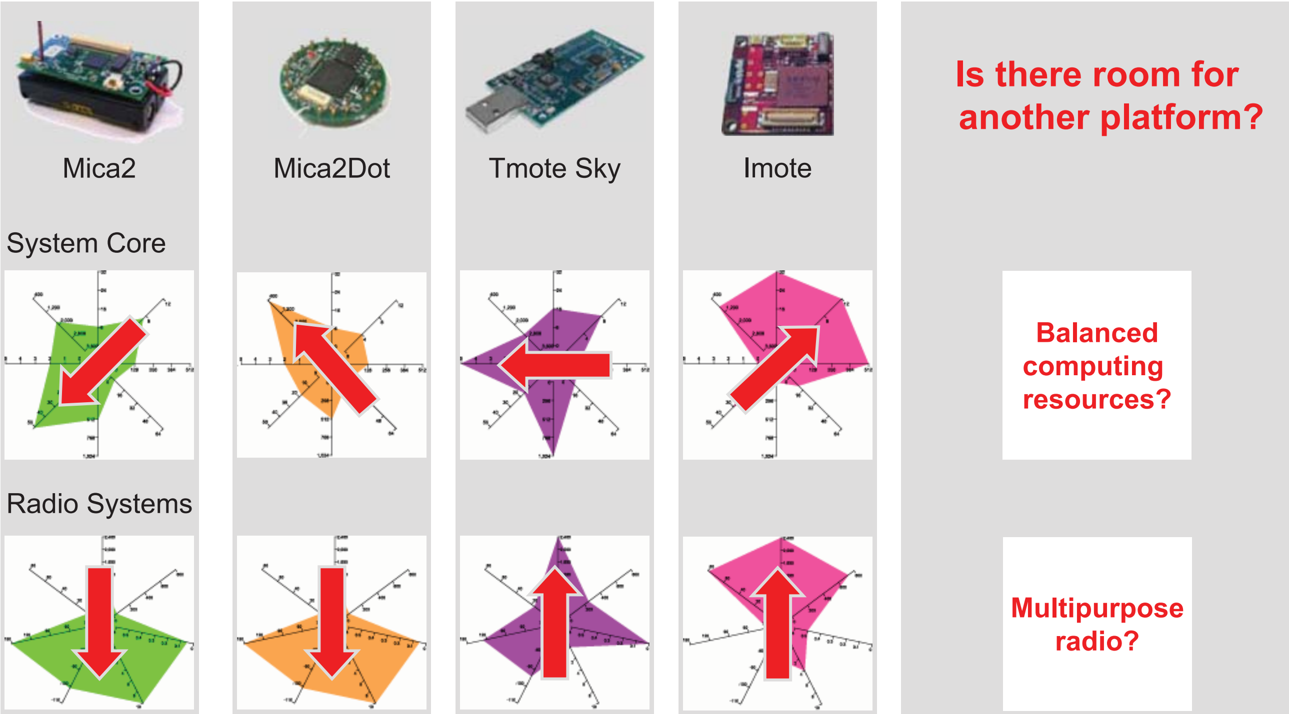


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

State-of-the-Art Platform Comparison



State-of-the-Art Platform Comparison



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

From Proof-of-concept to Real-world WSNs

Simulation Tools

- Of theory and practice

System Design

- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

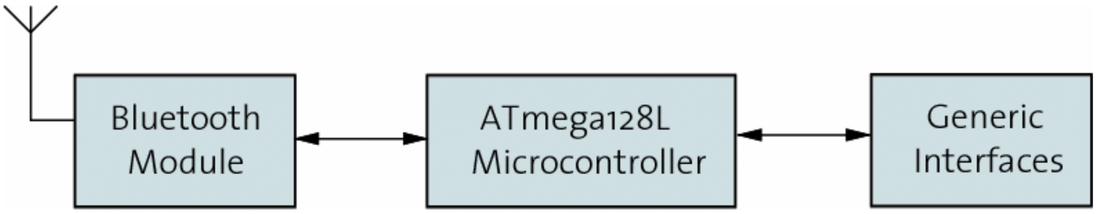
BTnode platform for fast-prototyping

- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

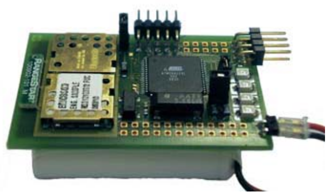
Development and Deployment

- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

The BTnode Platform



Communication — Computation — IO/Peripherals



Prototype → 2nd Generation → 3rd Generation

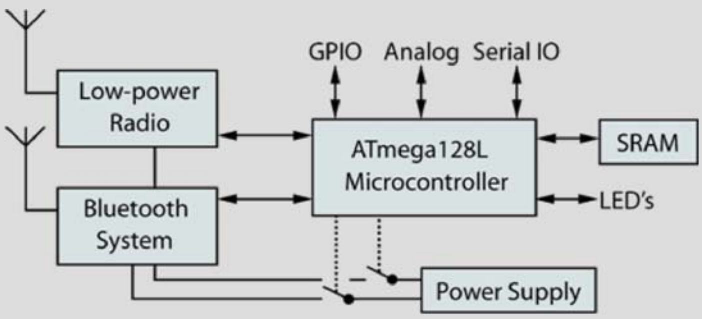
BTnode rev3 Architecture Details

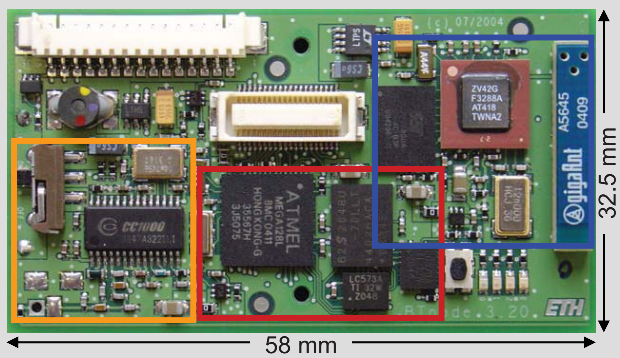
System core
- Atmel ATmega128
- 256 kB SRAM
- Generic IO/Peripherals
- Switchable power supplies

Dual radio system

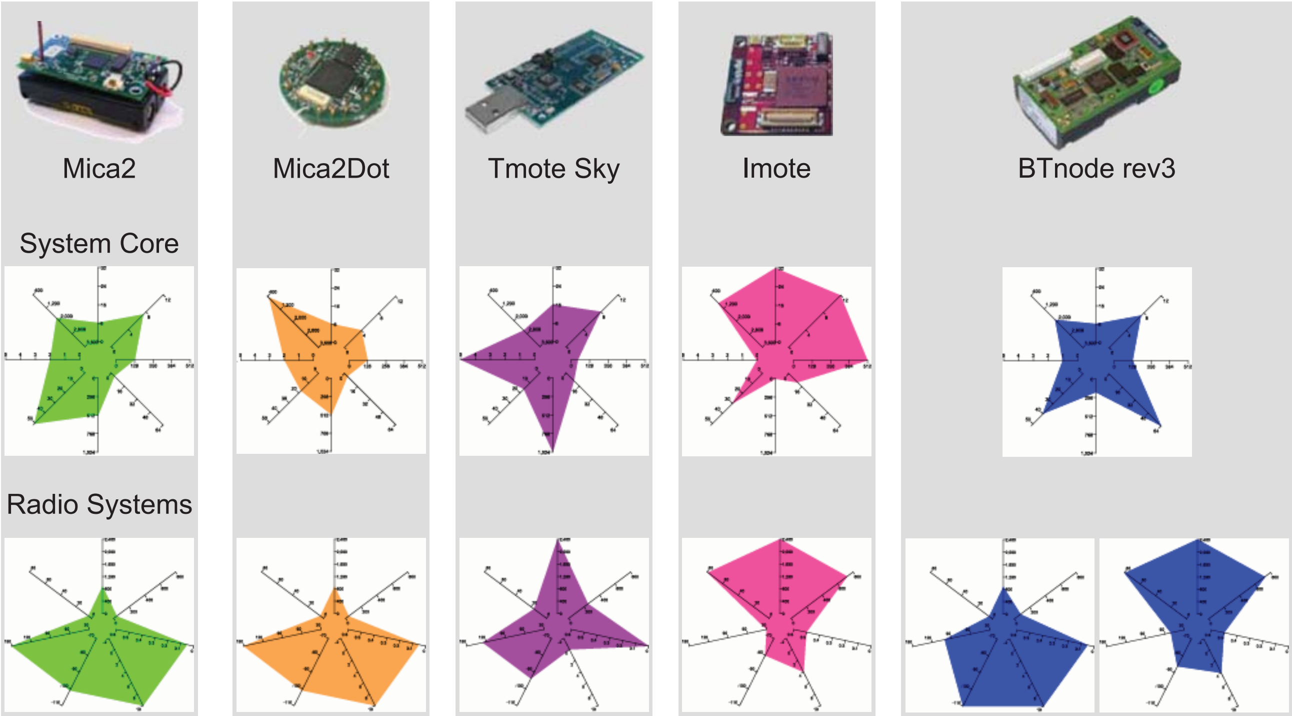
Bluetooth radio
- 2.4 GHz Zeevo ZV4002

Low-power radio
- 433-915 MHz ISM Chipcon CC1000

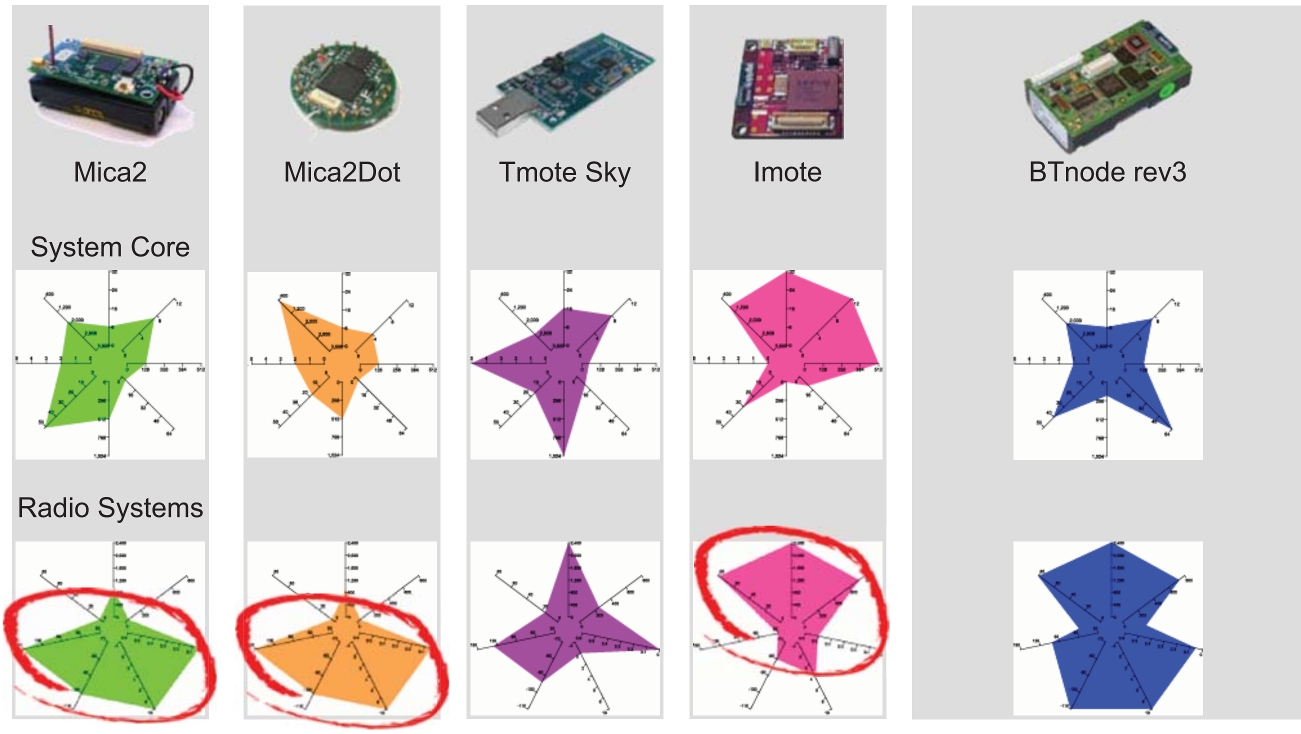




State-of-the-Art Platforms Comparison



State-of-the-Art Platforms Comparison



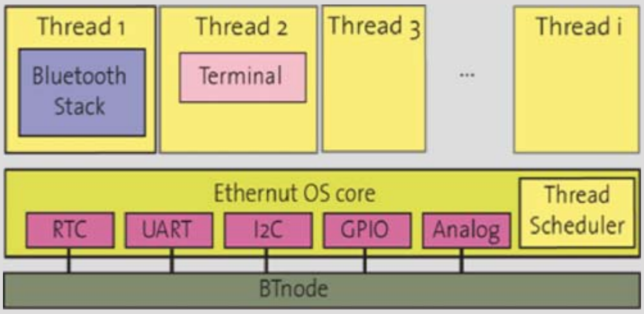
BTnut System Software

Versatile and flexible fast-prototyping

- Lightweight operating system support in plain C
- Linux-to-AVR embedded emulation
- Demo applications and tutorial

Built on top of multi-threaded Nut/OS framework

- Non-preemptive, cooperative multi-threading
- Events, timers
- Priorities for thread
- Dynamic heap allocation
- Interrupt driven streaming I/O



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

How Much Does System Software Help?

Pros

- Quick jumpstart (design kit, demo examples, tutorial)
- Community effort: exchange, collaboration, debugging
- Standardized interfaces, (modularity, reuse)
- Cleaner specifications, standards

Cons

- Overhead, learning curve (TinyOS CVS tree is ~200MB)
- Other peoples bugs/features make life hard

Bottom line

- Until it finally works you know your system so well, you might as well have started from scratch on your own…

BTnode Platform Success

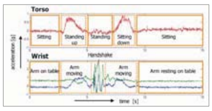
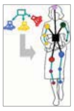
BTnode dev kit € 500

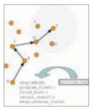
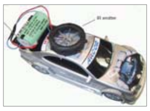
Vitronics Cobalt Blue™ Bluetooth Board

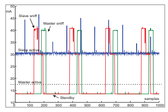
Industrial technology transfer
- Commercialization with ETH spin-off "Art of Technology"
- Commercial replicas resulting from open source policy

BTnodes in Education
- Different labs and demos
- Graduate lab in embedded systems (120 participants)
- 30-40 successfully completed student projects

BTnodes in Research Domains
- 25+ wearable and ubiquitous computing applications and demos
- Wireless (sensor) network research
- 40+ scientific publications based on or related to BTnodes

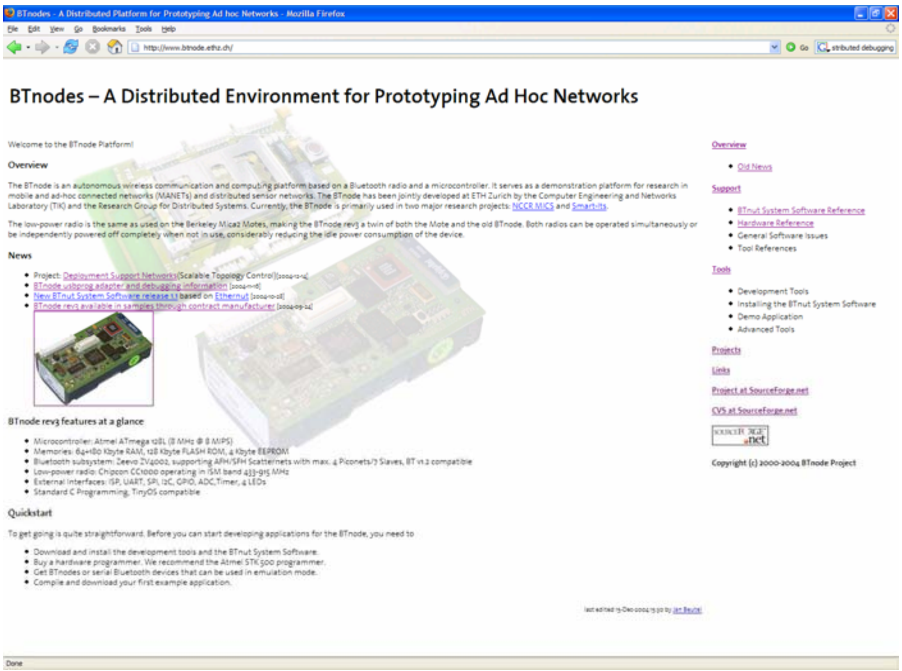






ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

To probe further…



http://www.btnode.ethz.ch

From Proof-of-concept to Real-world WSNs

Simulation Tools

- Of theory and practice

System Design

- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

BTnode platform for fast-prototyping

- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

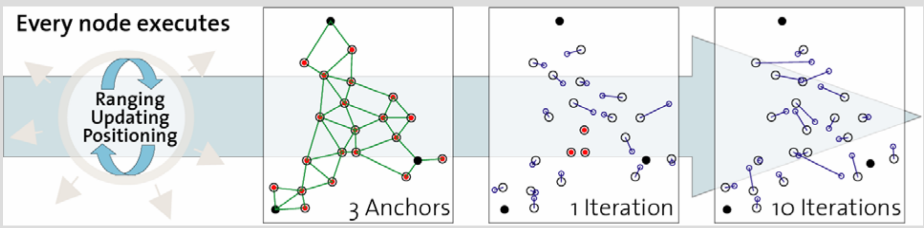
Development and Deployment

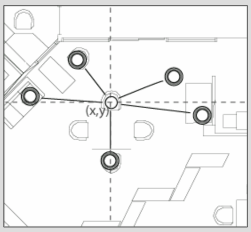
- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

Deployment of Network Services

Example: Location Management

- Finding position based on radionavigation
- Robust network-based trilateration





Service deployment functions

- Re-programming
- Supervision, control and monitoring
- Measurements, benchmarking

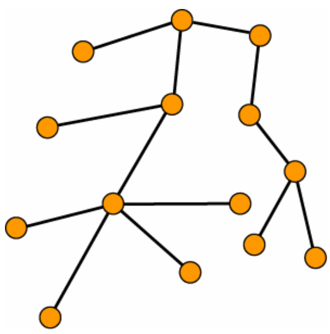
Requirement

- Robust connectivity
- Reliable data link layer

Bluetooth Multihop Network Topologies

Constructing ad hoc network topologies

- Large networks, many devices
- All devices connected
- Transparent multihop transport



Scatternet formation algorithms

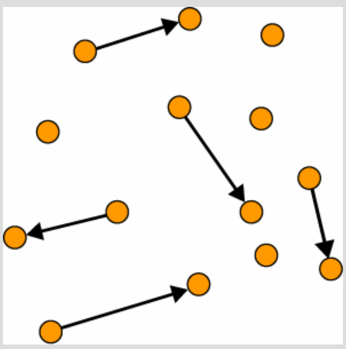
- BlueMesh [Petrioli2002], BlueStars [Petrioli2003], BlueRings [Foo2002], BlueTrees [Zaruba2001], mesh topologies [Guerin2003]
- Single-hop connectivity [Law2003]
- Complexity analysis [Law2003,Vergetis2003], comparative study [Basagni2004]

Mostly static, no (large-scale) implementation reports

Bluetooth Multihop Network Topologies

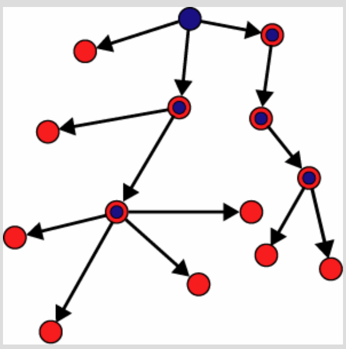
XHOP

- Initial experiments
- Time-multiplexed, dumbbell-like connections



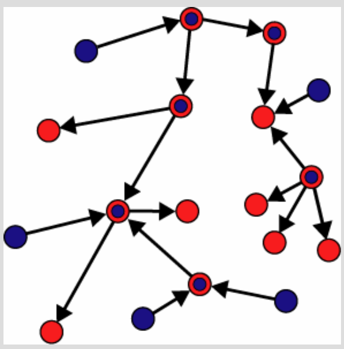
TreeNet

- Large, connected topologies
- Simple, top-down tree-building



DSNtrees

- Distributed tree topology formation
- Random connection points
- Streaming data

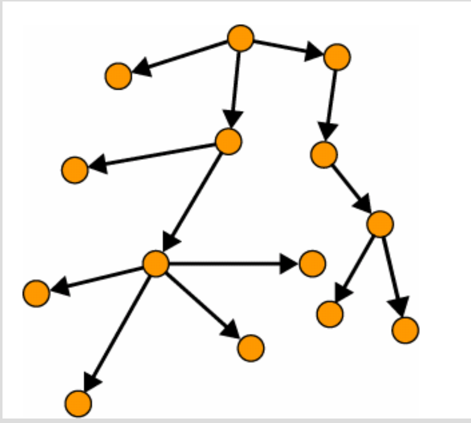


Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```



Distributed coordination

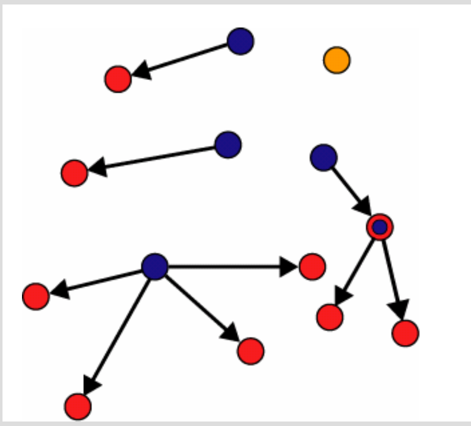
- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```



Distributed coordination

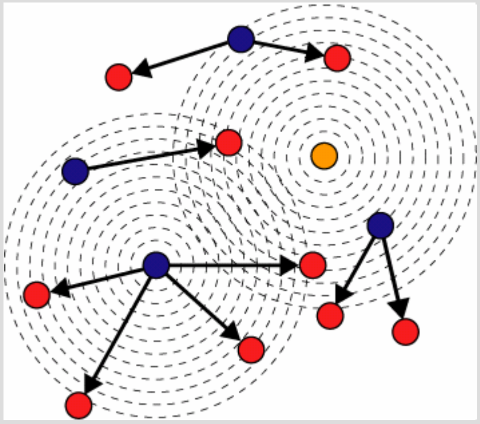
- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```



Distributed coordination

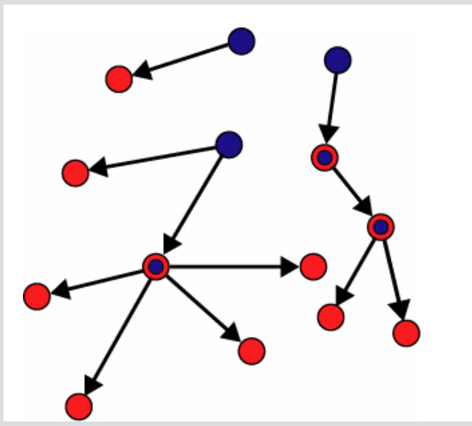
- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```

Distributed coordination

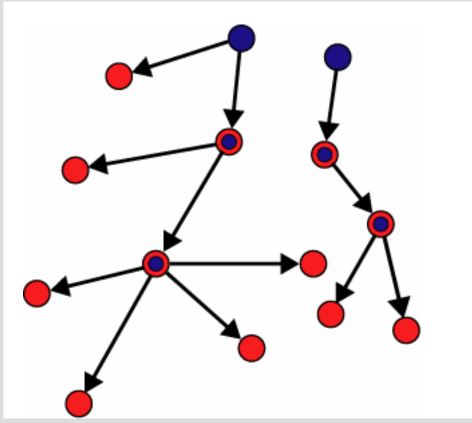
- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```



Distributed coordination

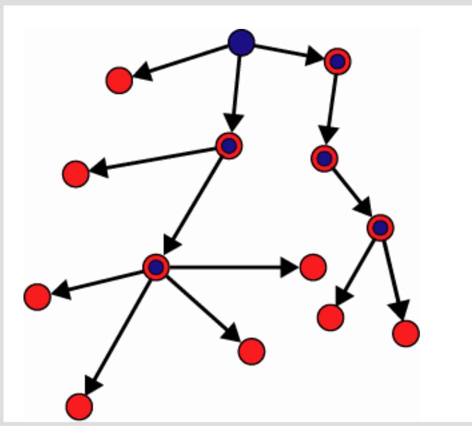
- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Simple Scatternet Tree Construction

Link layer connectivity

- Random search and connect

```
loop {
 while (my_slaves < max_degree) do
   found_nodes = inquiry();
   forall nodes in found_nodes do
     connect();
  }
}
```



Distributed coordination

- Inquiry() and connect() operations can exhibit long delays
- No a priori guarantee for success
- Serialization of parallel processes

Making a Seven Line Algorithm Work

```
loop {
 while (my_slaves < max_degree) do
     found_nodes = inquiry();
   forall nodes in found_nodes do
         connect();
  }
}
```

```
#define HEX2BYTE(c) ((u_char)(((c)<='9') ? (c)-'0' : tolower(c) - 'a' +
10))
typedef struct _jaws_stack {
    FILE *uart_terminal;
    HANDLE table_changed_event;
    bt_addr_t my_addr;
    struct btstack* bt_stack;
    bt_l2cap_stack_t *l2cap_stack;
} jaws_stack_t;
jaws_stack_t* _jaws_stack;
//int foo __attribute__ ((section (".noinit")));
//int foo2 __attribute__ ((section (".eeprom")));
void bt_print_bt_addr(bt_addr_t addr)
    DEBUG("%.2x:%.2x:%.2x:%.2x:%.2x:%.2x", addr[5], addr[4], addr[3],
addr[2], addr[1], addr[0]);
const char *bt_addr_to_string( char *buf, bt_addr_t addr)
    sprintf_P( buf, PSTR("%.2x:%.2x:%.2x:%.2x:%.2x:%.2x"),
             addr[5], addr[4], addr[3], addr[2], addr[1], addr[0]);
    return buf;
}
u_char* string_to_bt_addr(u_char* str, u_char* addr)
{
    char i;
    u_char *strp = str;
    // skip whitespace
    while(*strp == ' ')
        strp++;
    for(i = BD_ADDR_LEN-1; i >= 0; i--){
        if(isxdigit(strp[0]) && isxdigit(strp[1])){
            addr[(u_char) i] = HEX2BYTE(strp[0]) << 4 |
HEX2BYTE(strp[1]);
            strp+=2;
        }else{
            break;
        }
        // skip ':'
        if(i > 0){
            if(*strp == ':')
                strp++;
            else
                break;
        }
    }
u_char get_uart_errors(FILE* stream){
    u_long parameter;
    u_char errors;
    // check driver status
    _ioctl(_fileno(stream), UART_GETSTATUS, &parameter);

    if (parameter & UART_ERRORS) {
       errors = (u_char) (parameter & UART_ERRORS);
       // set error flags back to normal
       parameter = UART_ERRORS;
       _ioctl(_fileno(stream), UART_SETSTATUS, &parameter);
```

+ Adaptation to devices
- Root lockup, cycle elimination

+ Error handling
- Deadlocks, timeouts

+ Robustness, performance
- Greedy behavior, heuristics

+ Application support
- Basic OS functions
- Debugging, visualization, monitoring
- Stepwise testing + deployment

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Making a Seven Line Algorithm Work

```
loop {
 while (my_slaves < max_degree) do
     found_nodes = inquiry();
   forall nodes in found_nodes do
         connect();
  }
}
```

Seven lines

2000 lines
~87 kbyte

+ Adaptation to devices
- Root lockup, cycle elimination

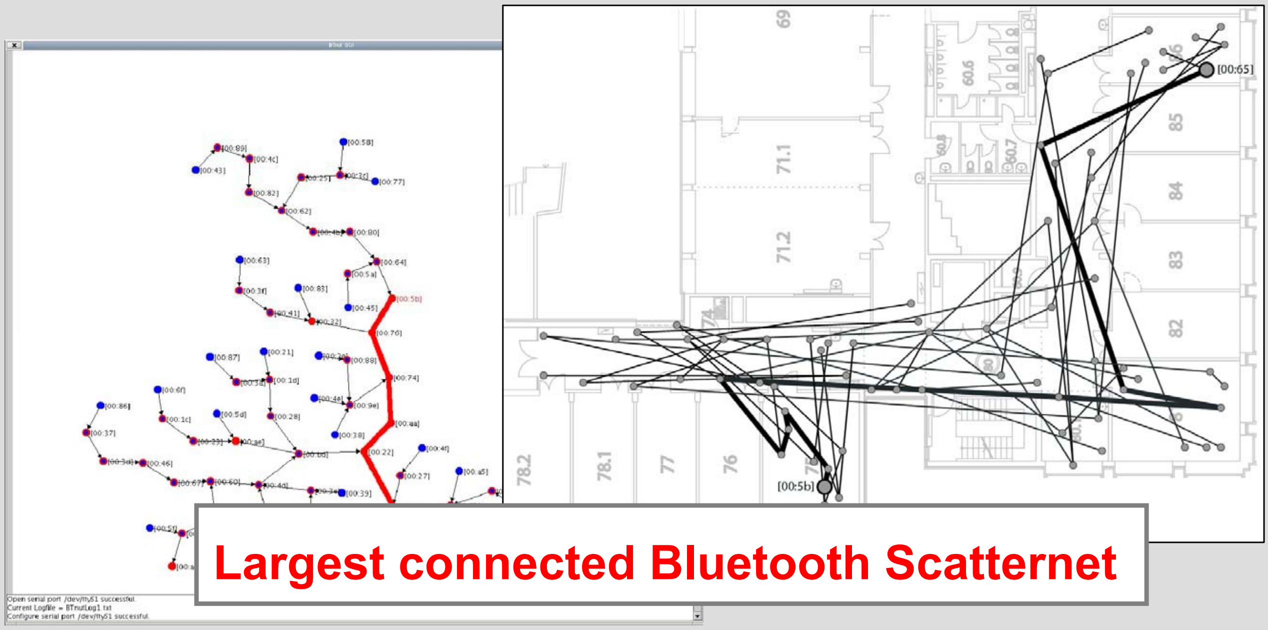
+ Error handling
- Deadlocks, timeouts

+ Robustness, performance
- Greedy behavior, heuristics

+ Application support
- Basic OS functions
- Debugging, visualization, monitoring
- Stepwise testing + deployment

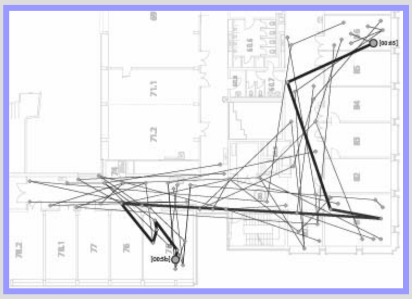
DSNtrees – Field Experiments

Deployment using 70+ nodes on an office floor

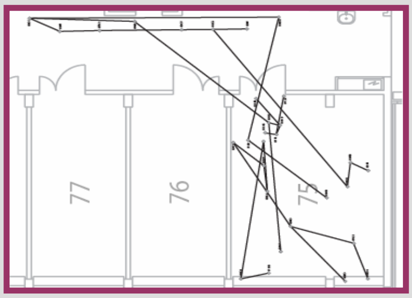


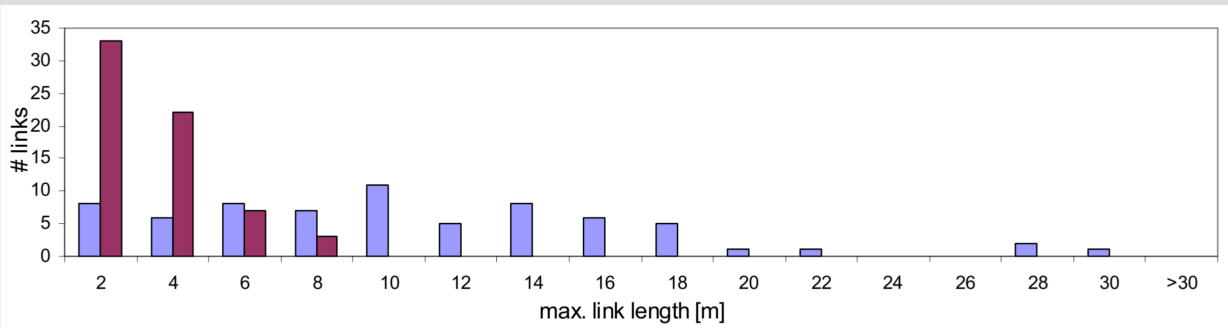
DSNtrees – Connection Manager Variants

Random selection



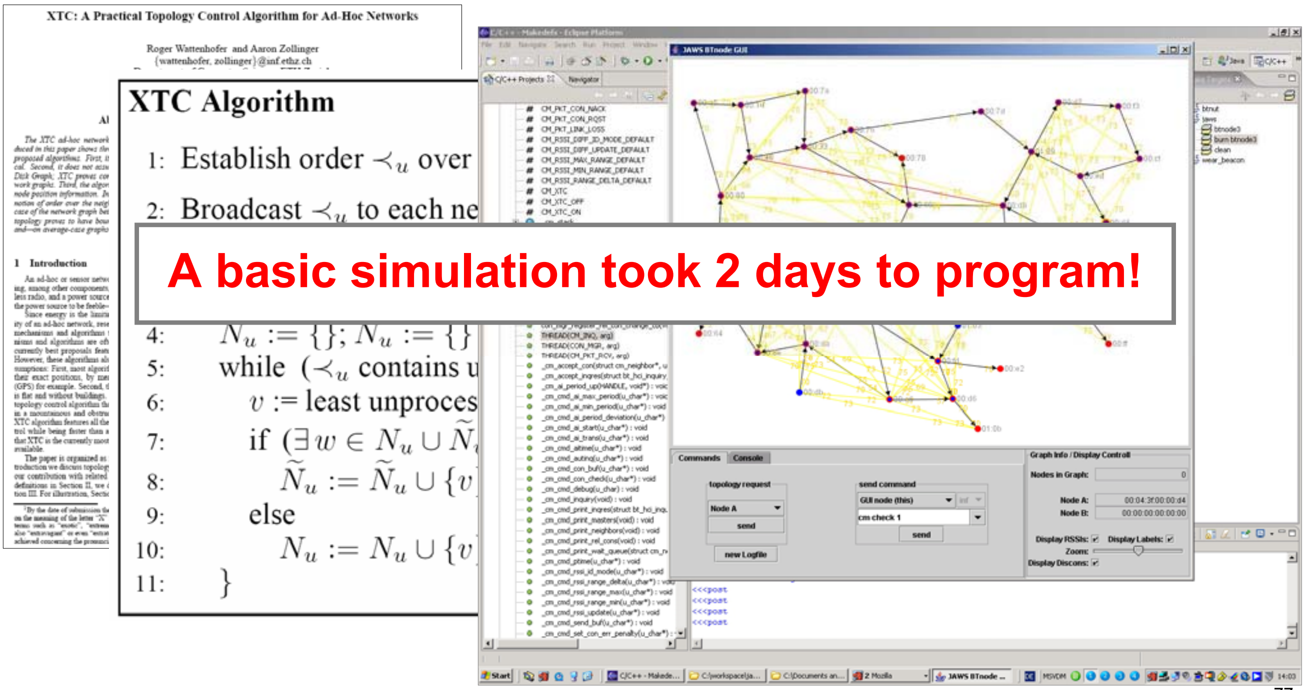
RSSI-limited selection





XTC – Bluetooth Mesh Networking

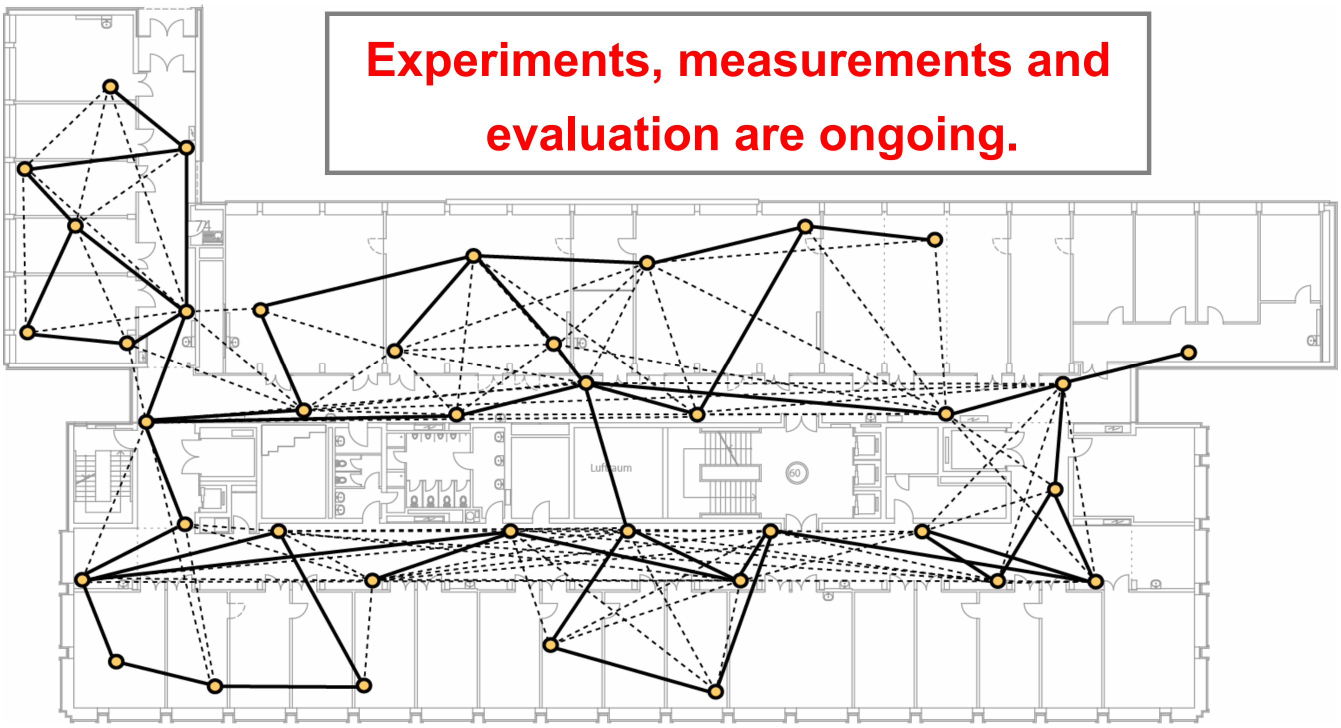
Theory paper-grade algorithm to implementation in 6 months



A basic simulation took 2 days to program!

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

XTC – Bluetooth Networking Revisited



Experiments, measurements and evaluation are ongoing.

From Proof-of-concept to Real-world WSNs

Simulation Tools
- Of theory and practice

System Design
- Embedded systems design
- Case study: Mars pathfinder
- Tools: Design space exploration

Concept/Theory → Design and Development → Prototype and Pilot → Launch and Ramp → Production → Service and Support

BTnode platform for fast-prototyping
- Tools: Metrics and comparison of existing platforms
- Design of hard- and software based on standardized interfaces and event driven interaction

Development and Deployment
- Application experience
- Case study: Bluetooth Scatternets
- Deployment-support concept

Today's WSN Design and Development

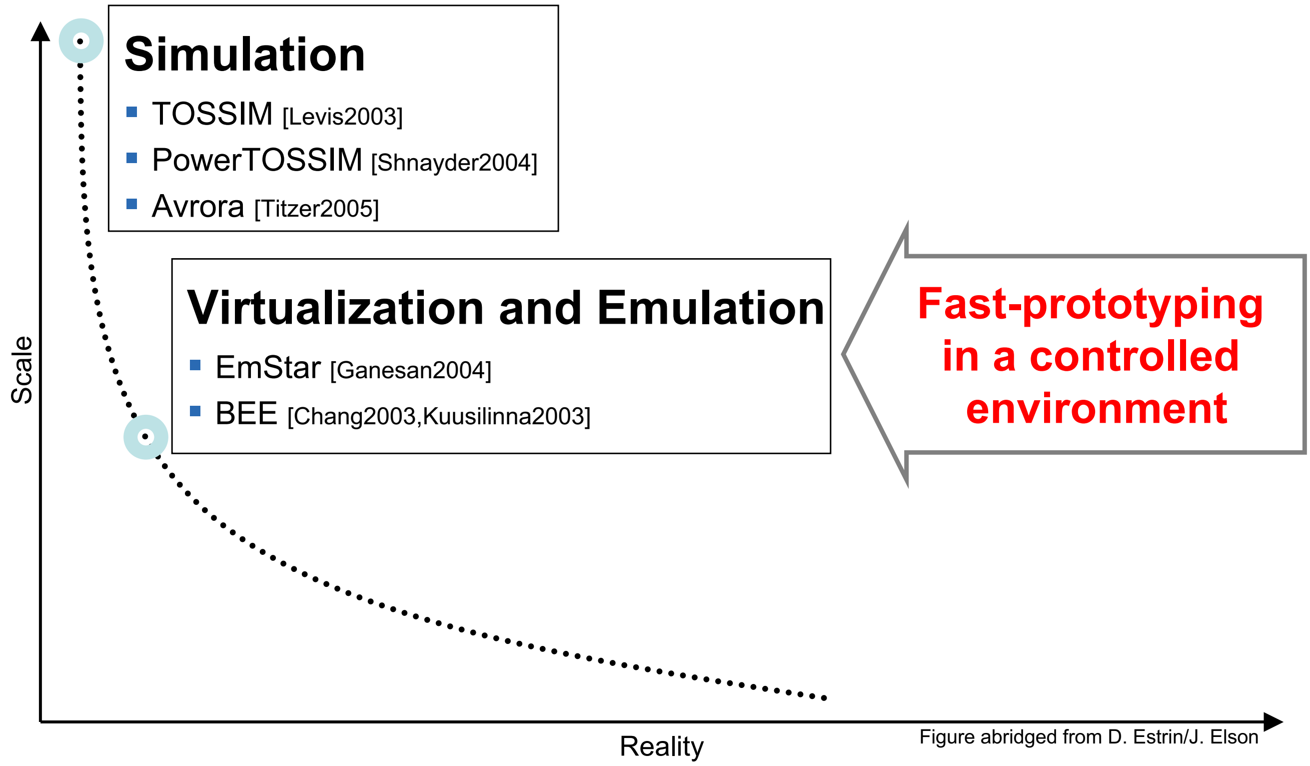
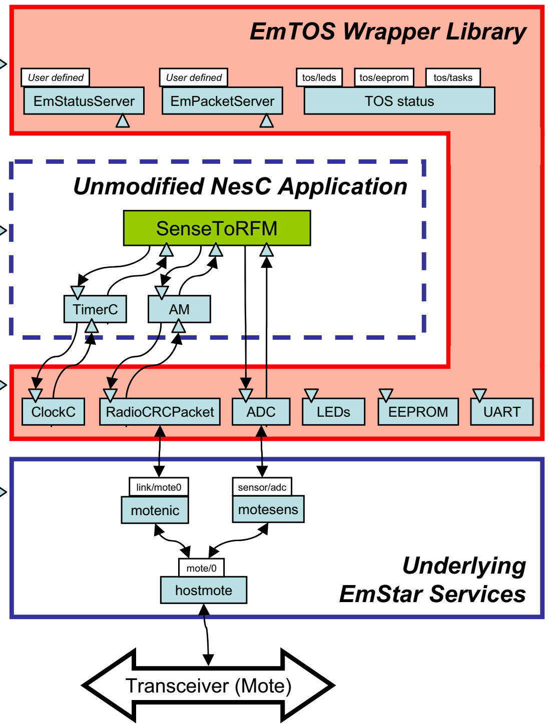


Figure abridged from D. Estrin/J. Elson

EmStar – Emulation on Backend Servers

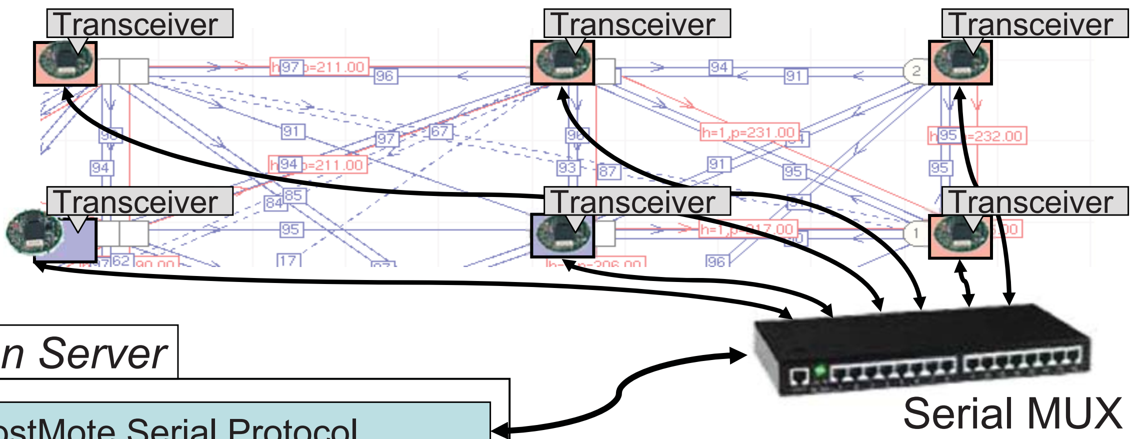
- Enables NesC applications to provide *new* EmStar services
- Wraps an unmodified NesC app into an EmStar module
- Implements TinyOS API and low-level components…
- By connecting to existing EmStar services

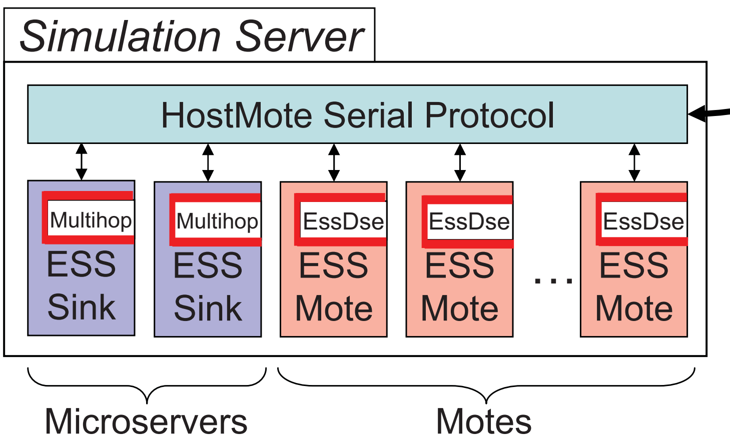


This fairly small investment of effort -- the wrapper library and some TinyOS components that make up a new TinyOS platform -- simultaneously addresses *both* goals of heterogeneous simulation *and* integration!

Material courtesy of L. Girod

EmStar – Emulation Array





Emulation Array

- Emulation/Real/Hybrid Mode
- Real motes installed in environment
- Serial multiplexer connects server to real motes, replaces channel model
- Limits scale – improves reality

Material courtesy of L. Girod

CENS

The Ceiling Array: A Real Wireless Channel

UCLA USC UCR CALTECH CSU

CENTER FOR EMBEDDED NETWORKED SENSING

Slide courtesy of D. Estrin

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CENS

Portable Array: In-Situ, so Smaller Scale

UCLA USC UCR CALTECH CSU

CENTER FOR EMBEDDED NETWORKED SENSING

Slide courtesy of D. Estrin

Today's WSN Design and Development

Simulation
- TOSSIM [Levis2003]
- PowerTOSSIM [Shnayder2004]
- Avrora [Titzer2005]

Virtualization and Emulation
- EmStar [Ganesan2004]
- BEE [Chang2003,Kuusilinna2003]

Test Grids
- moteLab [Werner-Allen2005]
- Emstar arrays [Cerpa03/04]
- Kansei [Dutta2005]

Closing in on the "real" experience

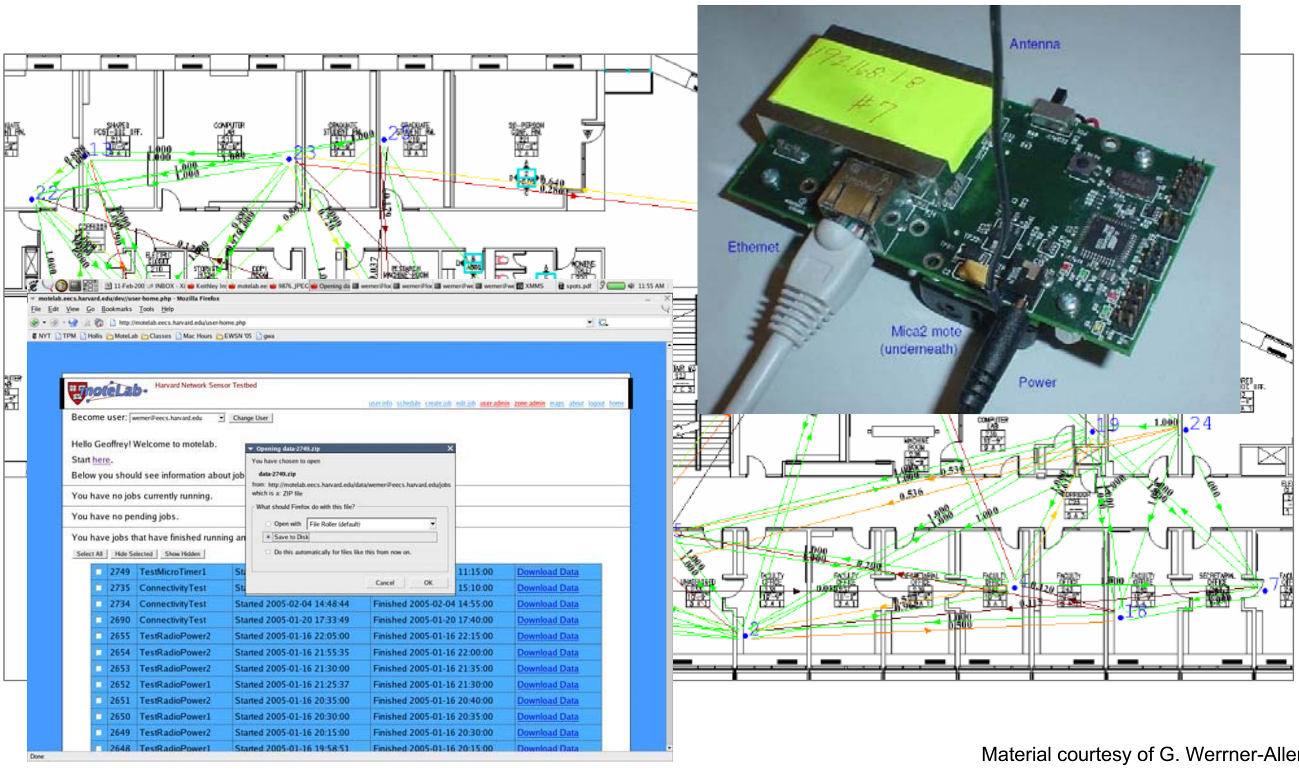
Scale

Reality

Figure abridged from D. Estrin/J. Elson

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

MoteLab – Test Bed and Compute Server



Material courtesy of G. Werrner-Allen

Today's Design and Development

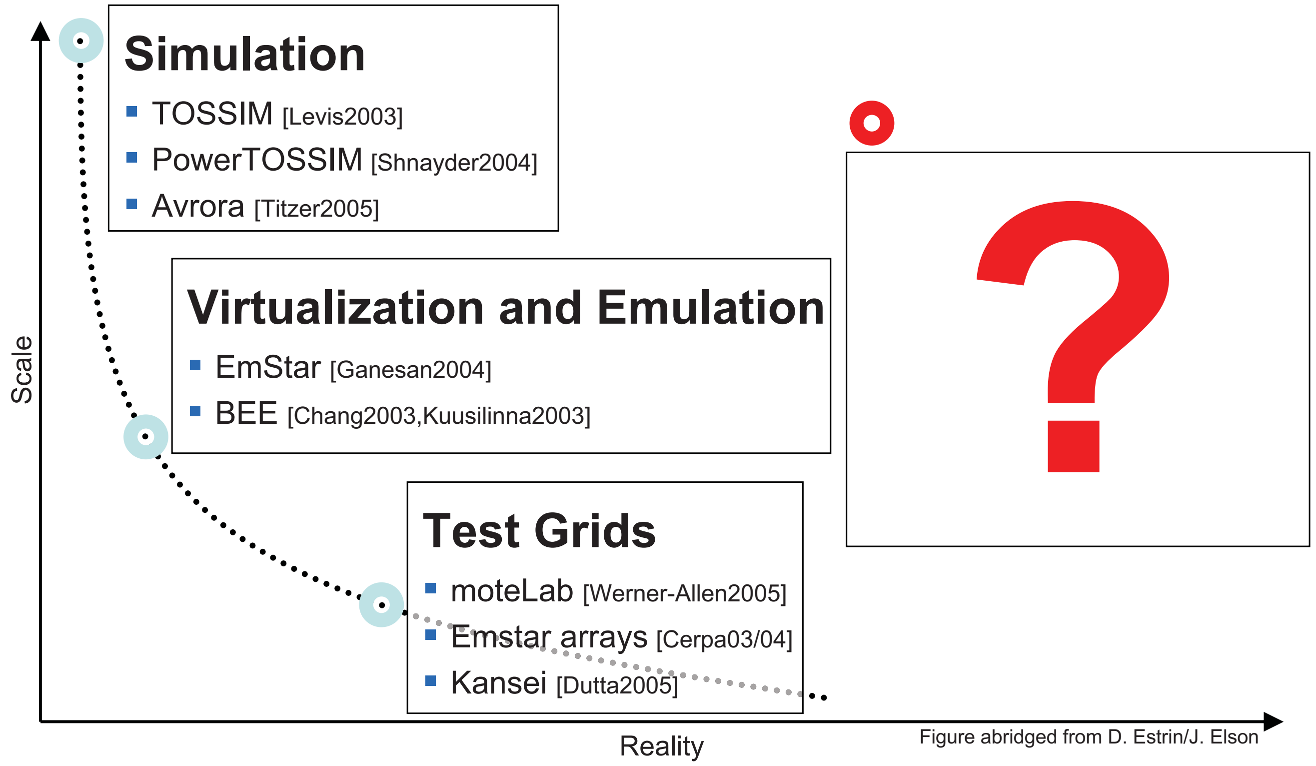


Figure abridged from D. Estrin/J. Elson

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Today's Design and Development

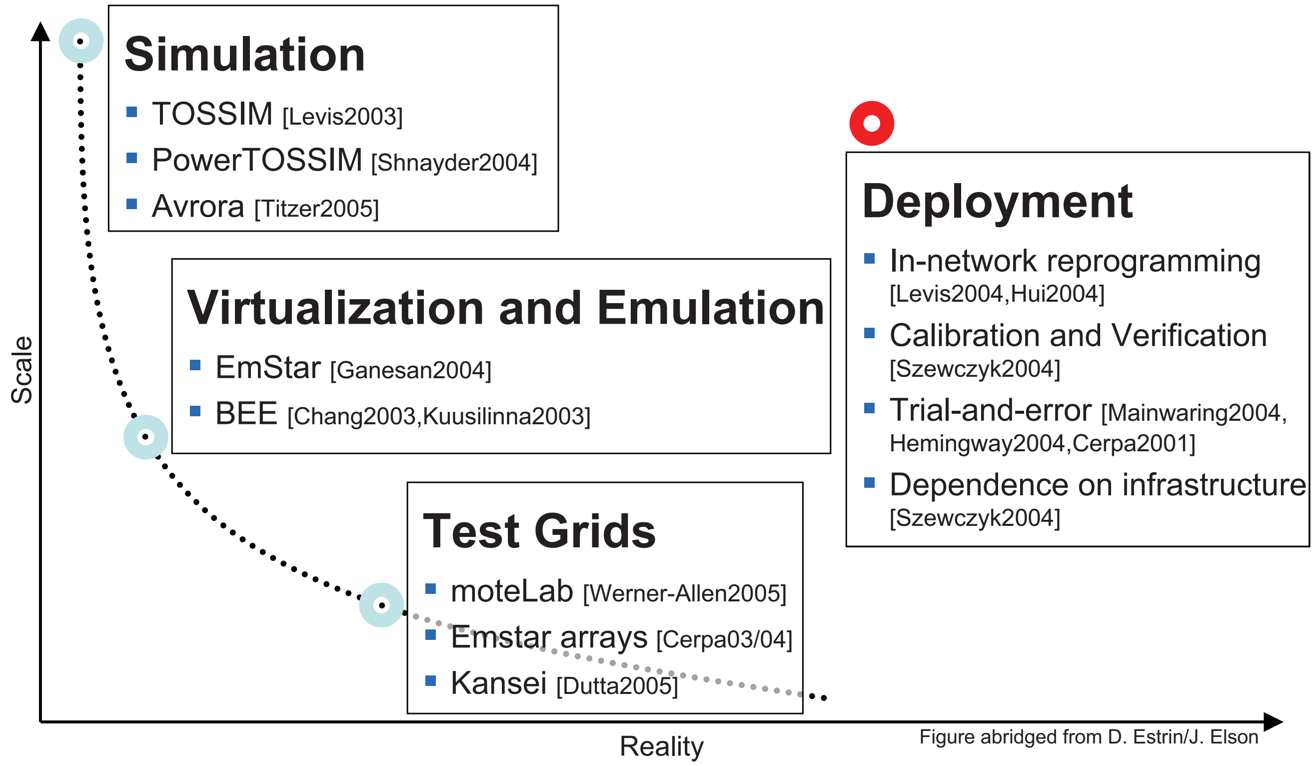
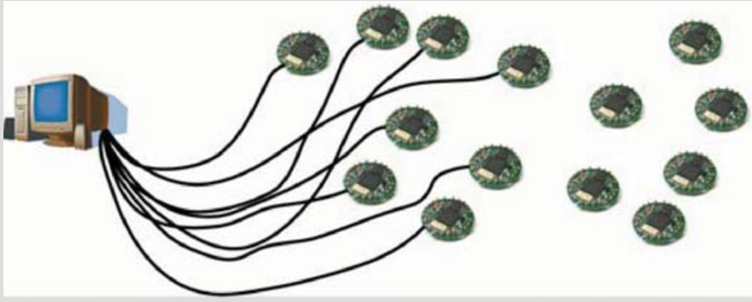


Figure abridged from D. Estrin/J. Elson

Next-Generation Deployment-Support

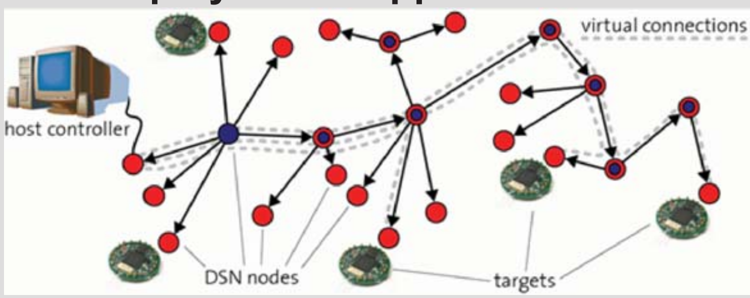
Traditional test grid
- Wired
- Immobile
- Not scalable



In-network tools
- Unreliable

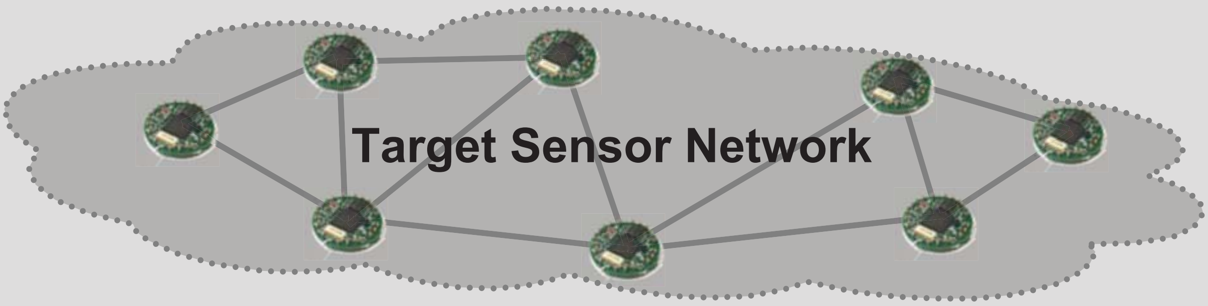
Self-organizing backbone network with deployment-support services

Deployment-Support Network

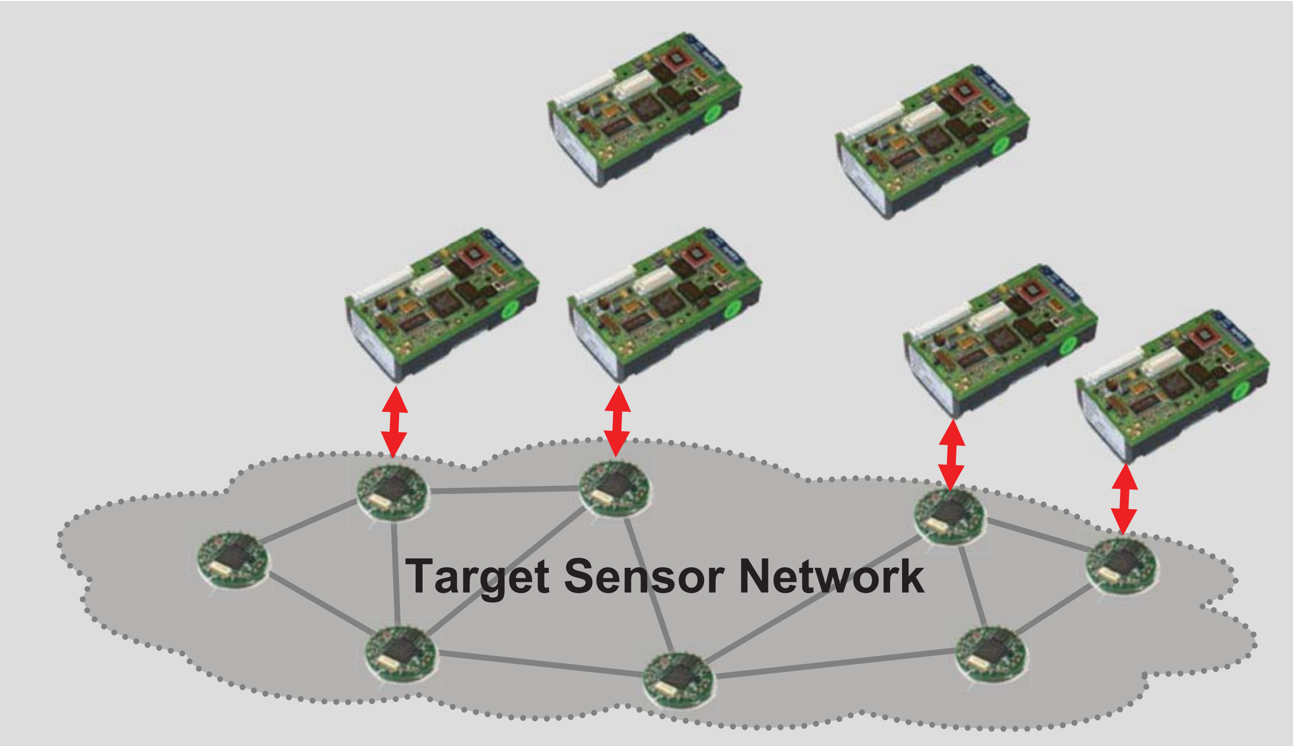


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Next-Generation Deployment-Support

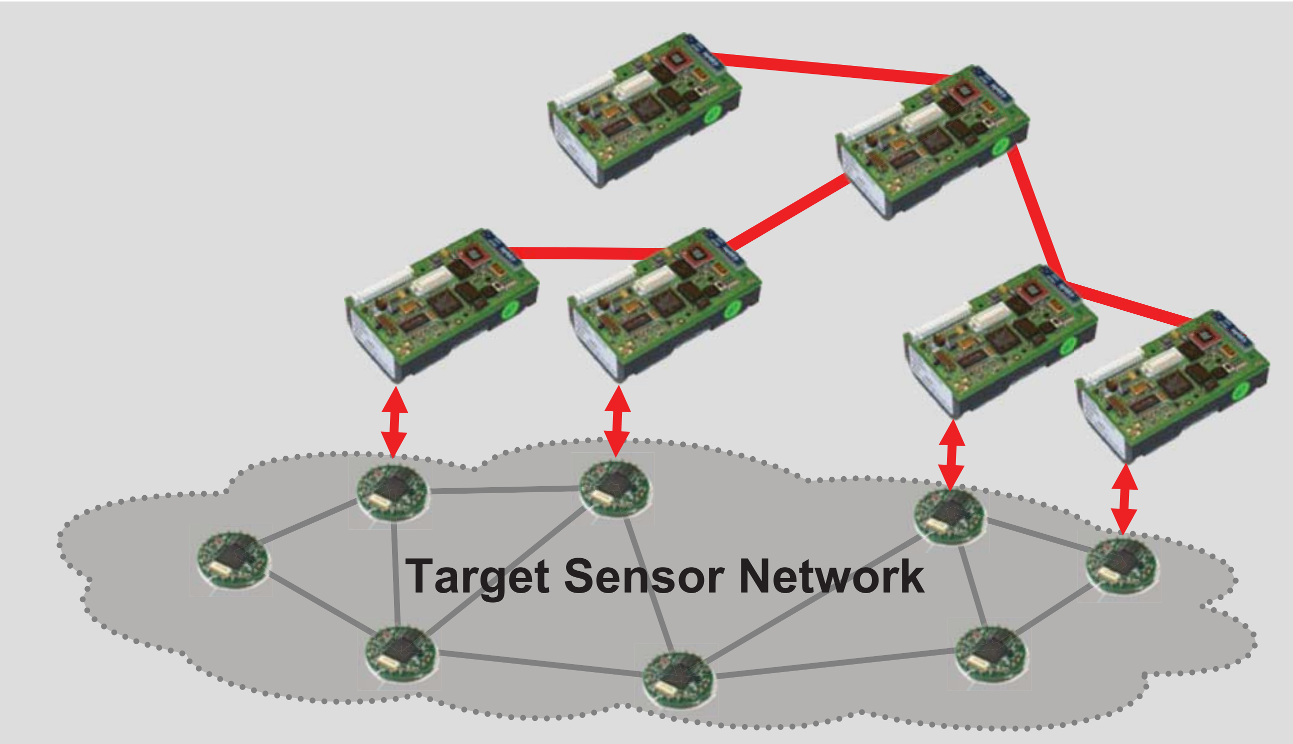


Next-Generation Deployment-Support

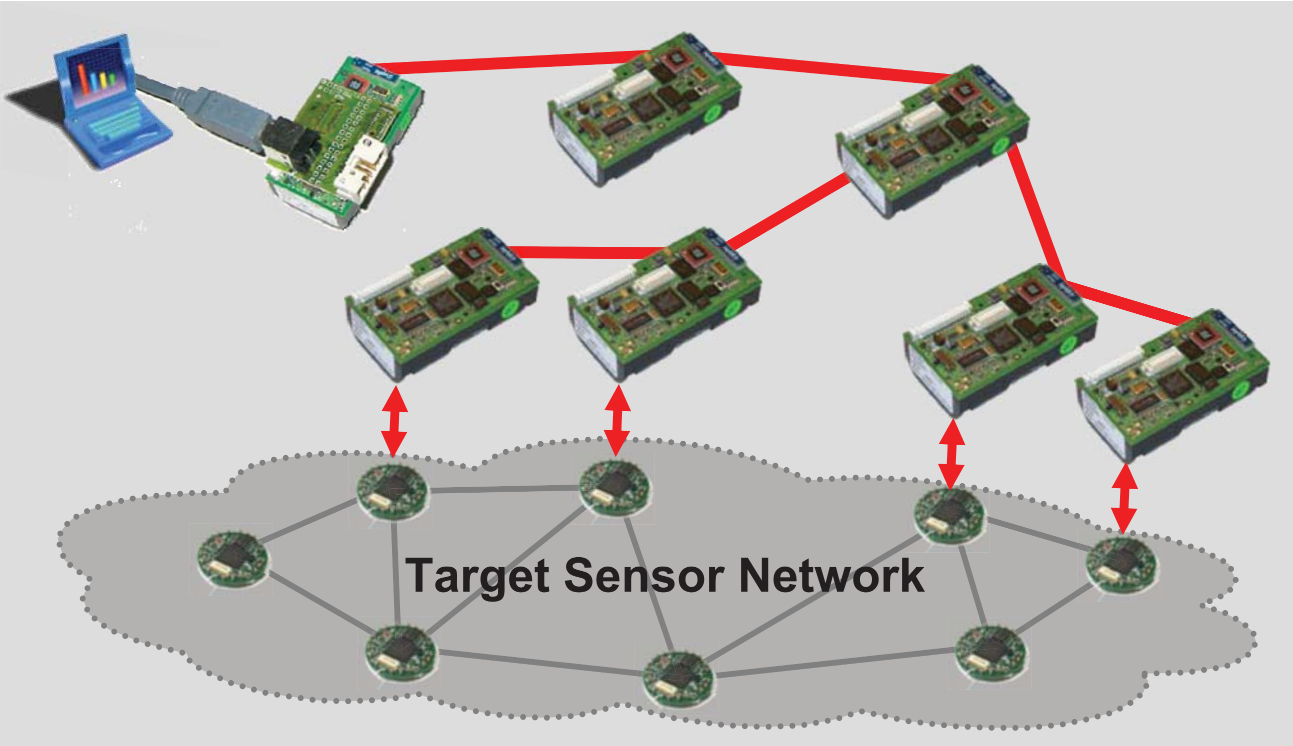


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Next-Generation Deployment-Support

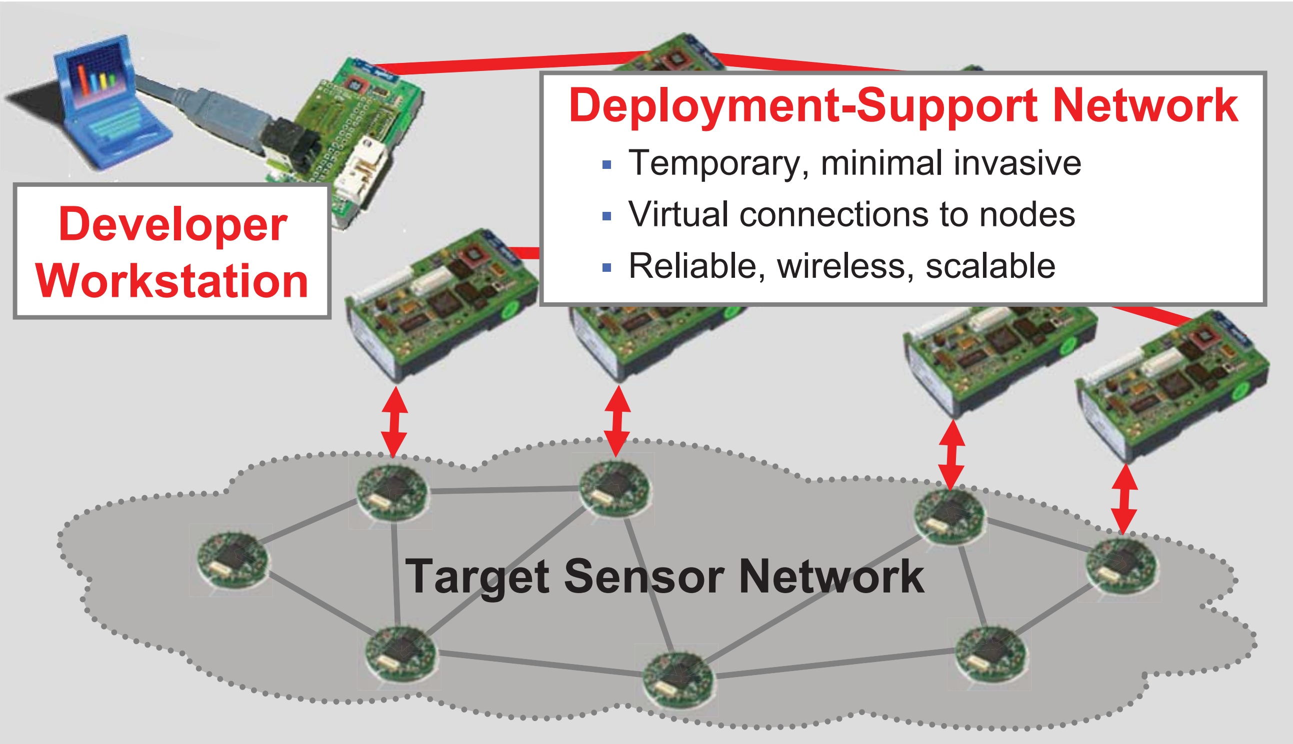


Next-Generation Deployment-Support



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Next-Generation Deployment-Support



Vision: Full Life-Cycle Support for WSNs

Stepwise refinement

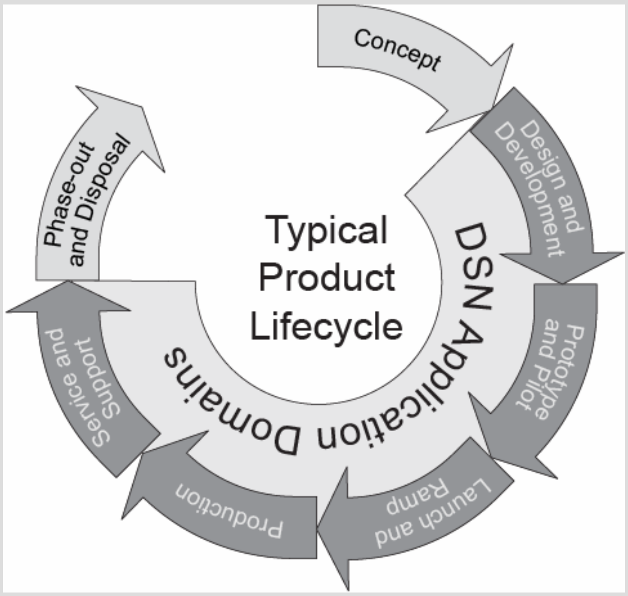
Feedback to

- Design
- Development

Monitoring of

- Functionality
- Quality

Validation and Verification



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Further Reading

Suggested Papers (in this order)

- R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004), pages 214–226. ACM Press, New York, November 2004.
- J. Gray. Why do computers stop and what can be done about it? In Proc. 5th Symp. Reliability in Distributed Software and Database Systems (SRDS 86), pages 3–12, January 1986.
- D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dartmouth College Computer Science, July 2003.
- G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: A wireless sensor network testbed. In Proc. 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05), pages 483–488. IEEE, Piscataway, NJ, April 2005.
- L. Girod, T. Stathopoulos, N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer, "A System for Simulation, Emulation, and Deployment of Heterogeneous Sensor Networks", In Proc. of SenSys 2004.
- J. Beutel, M. Dyer, M. Hinz, L. Meier, and M. Ringwald. Next-generation prototyping of sensor networks. In Proc. 2nd ACM Conf. Embedded Networked Sensor Systems (SenSys 2004), pages 291–292. ACM Press, New York, November 2004.

Further Reading

Further Papers and Reports

- L. Girod, J. Elson, A. Cerpa, T. Stathapopoulos, N. Ramananthan, and D. Estrin. *EmStar: A software environment for developing and deploying wireless sensor networks.* In Proc. USENIX 2004 Annual Tech. Conf., pages 283–296, June 2004.
- R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. *Lessons from a sensor network expedition.* In Proc. 1st European Workshop on Sensor Networks (EWSN 2004), volume 2920 of Lecture Notes in Computer Science, pages 307–322. Springer, Berlin, January 2004.
- J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K.C. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. *Effects of Detail in Wireless Network Simulation.* In Proc. SCS Multiconference Distributed Simulation 2001, pages 3–11. USC/Information Sciences Institute, Society for Computer Simulation, Los Angeles, CA, January 2001.
- D. Kotz, C. Newport, R.S. Gray, J. Liu, Y. Yuan, and C. Elliott. *Experimental evaluation of wireless simulation assumptions.* In Int'l Workshop Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM 04), pages 78–82. ACM Press, New York, October 2004.
- D. Cavin and Y. Sasson. *On the accuracy of MANET simulators.* In ACM Workshop Principles Of Mobile Computing (POMC 02), pages 38–43. ACM Press, New York, October 2002.
- J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund, and L. Thiele. *Prototyping wireless sensor network applications with BTnodes.* In Proc. 1st European Workshop on Sensor Networks (EWSN 2004), volume 2920 of Lecture Notes in Computer Science, pages 323–338. Springer, Berlin, January 2004.
- J. Beutel, M. Dyer, L. Meier, and L. Thiele. *Scalable topology control for deployment-sensor networks.* In Proc. 4th Int'l Conf. Information Processing in Sensor Networks (IPSN '05), pages 359–363. IEEE, Piscataway, NJ, April 2005.
- ESA, *Ariane 501 - Presentation of Inquiry Board report*, press release N° 33-1996.
- Yeh, Y.C.; *Design considerations in Boeing 777 fly-by-wire computers*, HASE 1998.
- Richard Feynman, *PRESIDENTIAL COMMISSION ON THE SPACE SHUTTLE CHALLENGER ACCIDENT*, Appendix F.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Further Reading

Books

- *What Do You Care What Other People Think?*, Richard P. Feynman, W. W. Norton & Company, 2001, ISBN 0-393-32092-8
- *Ambient Intelligence*, Editors: W. Weber, E. Aarts and J.M. Rabaey, Springer, Berlin, 2004, ISBN 3-540-23867-0
- *Embedded System Design*, Peter Marwedel. Kluwer Academic Publishers, Nov. 2003, ISBN 1-4020-7690-8, 258 pp.

Part of the material used in this tutorial originates from other authors.

- L. Thiele, S. Künzli, R. Wattenhofer, ETH Zurich
- P. Marwedel, U. Dortmund
- R. Szewczyk, UC Berkeley
- T. Henzinger, EPF Lausanne
- D. Estrin, L. Girod, UCLA
- G. Werner-Allen, Harvard

Think,

Try hard,

Talk to the community,

Use simple solutions,

Share your work.

You are not alone.

Have fun…