Telecooperation Lab
Prof. Dr. Max Mühlhäuser

TK3: Ubiquitous Computing

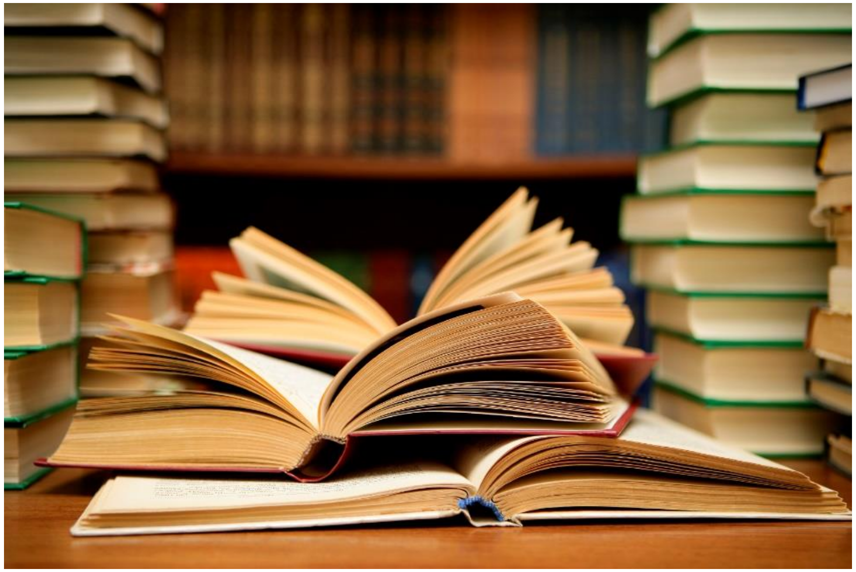
Chapter 1: Devices & Sensing

Part 2: Sensors & Actuators

Lecturer: Dr. Immanuel Schweizer

Sources

- http://web.cse.ohio-state.edu/~anish/788-12.html
- *www.cs.berkeley.edu/~prabal/teaching/cs294-11-f05*
- *Book slides: 6LoWPAN – The Wireless Embedded Internet*
- *Book / Lecture slides: Embedded System Design & CPS, TU Dortmund*
- http://www.ecse.rpi.edu/homepages/abouzeid/6962-05/monet12-coverage-xiang.pdf



Recap: Tagging

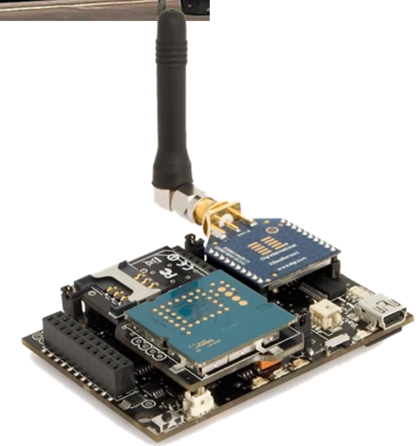
- Tagging
  - Augmenting people, objects, and the environment with digital information
- Technologies
  - RFID
  - NFC
  - iBeacons
- Automatic interaction between virtual and real world

Recap: iBeacon

- Tagging of the environment
  - Distribute beacons in stores / airports
- Bluetooth LE
  - Years of operation
  - Longer Range (~60-350 meters)
- RSSI measurements
  - Measure distance
  - Additional sensor data
  - Location & Direction
- Use cases:
  - Indoor location
  - Proximity services (Ads…)

Sensors & Actuators

- Embedded Systems
  - Sensor Nodes
- Wireless Sensor Networks
- Cyper-Physical Systems

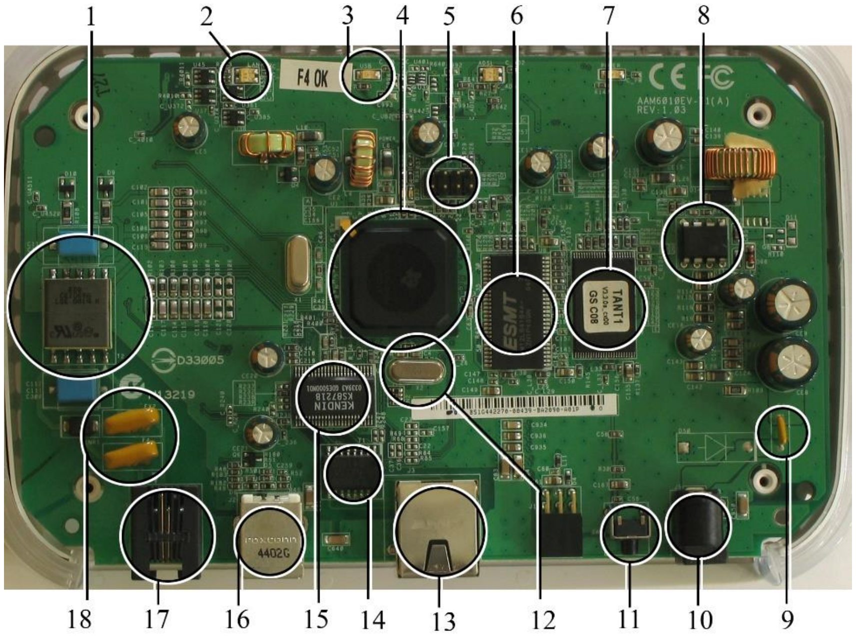


Remember

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”

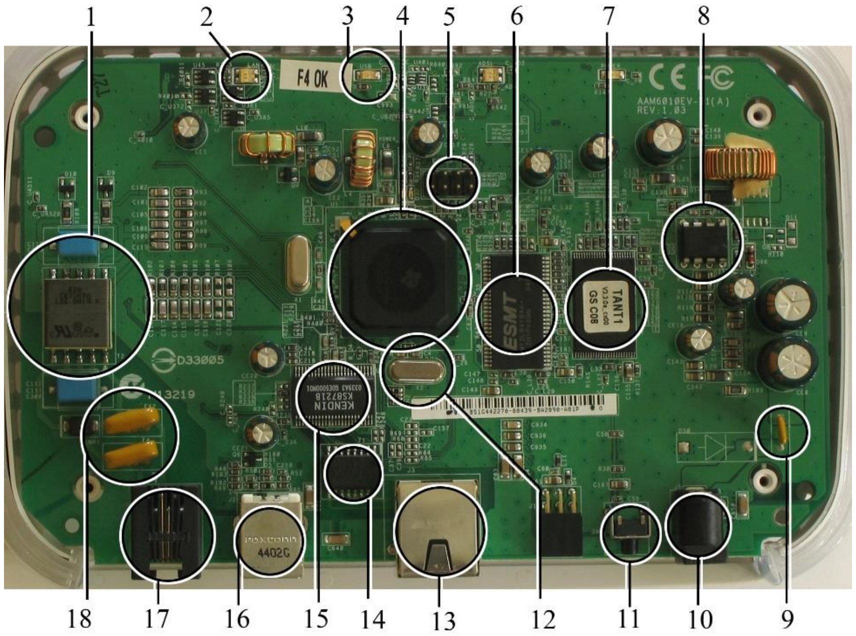
Embedded Systems

An **embedded system** is a computer **system** with a dedicated function within a larger mechanical or electrical **system**, often with real-time computing constraints. It is **embedded** as part of a complete device often including hardware and mechanical parts. **Embedded systems** control many devices in common use today.



Embedded Systems

- Embedding computer systems anywhere
- Dedicated function
- Control devices



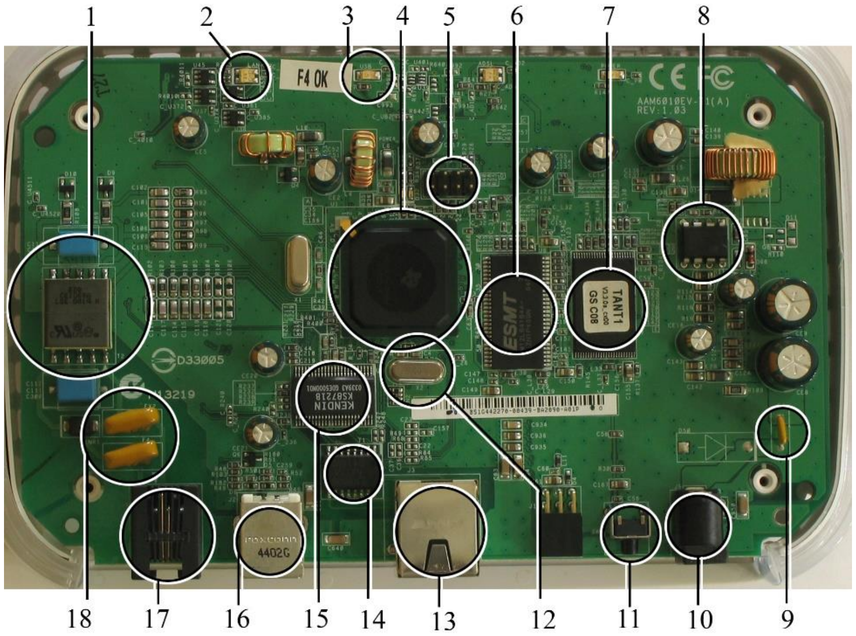
Example: Car

- EPS
- ABS
- Airbags
- Efficient automatic gearbox
- Blind-Angle alert systems

Embedded Systems

- Embedding computer systems anywhere
- Dedicated function
- Control devices

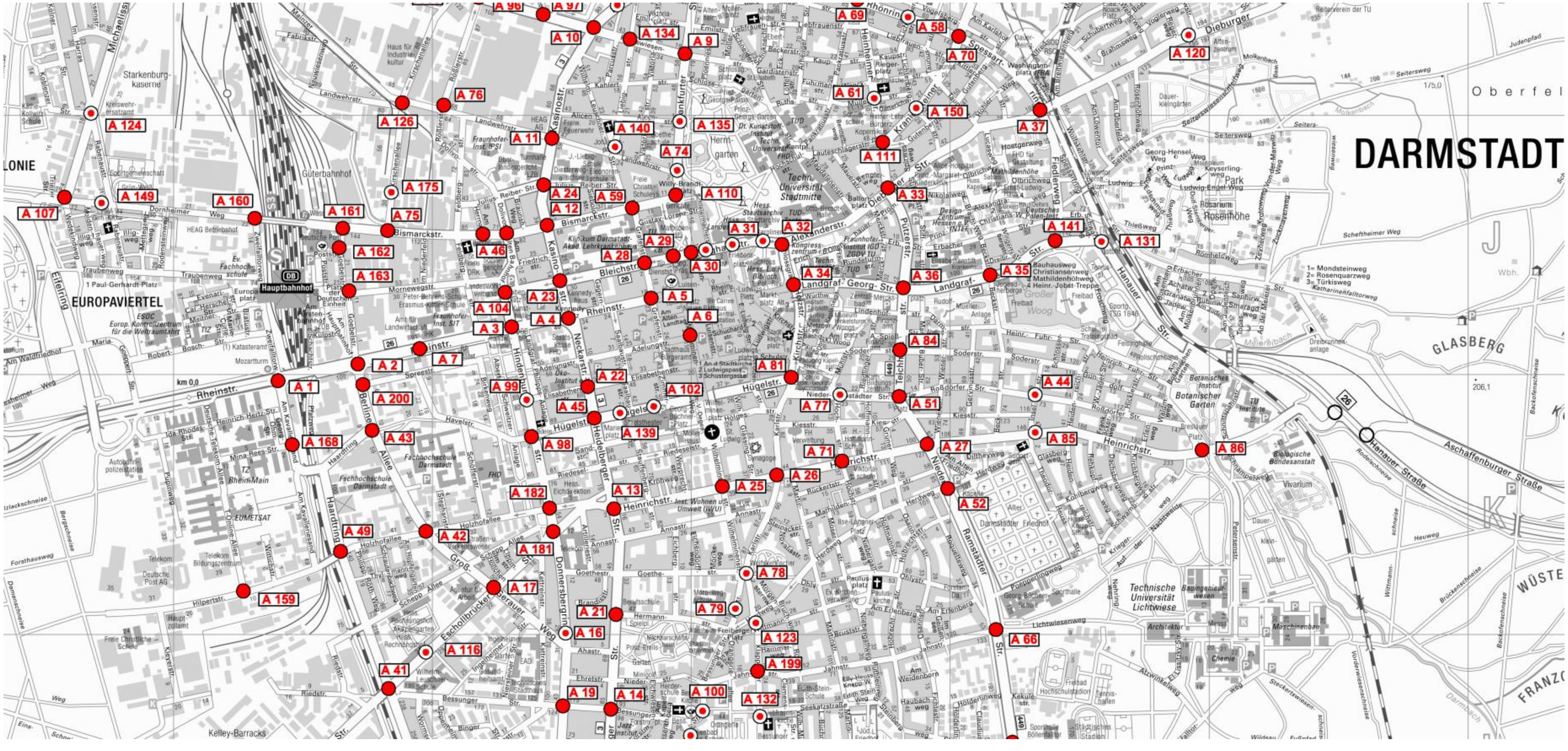
- Stand-alone or closed systems
- Connectivity?



Embedded Infrastructure

Traffic sensors

- 171 traffic lights equipped with inductive loops

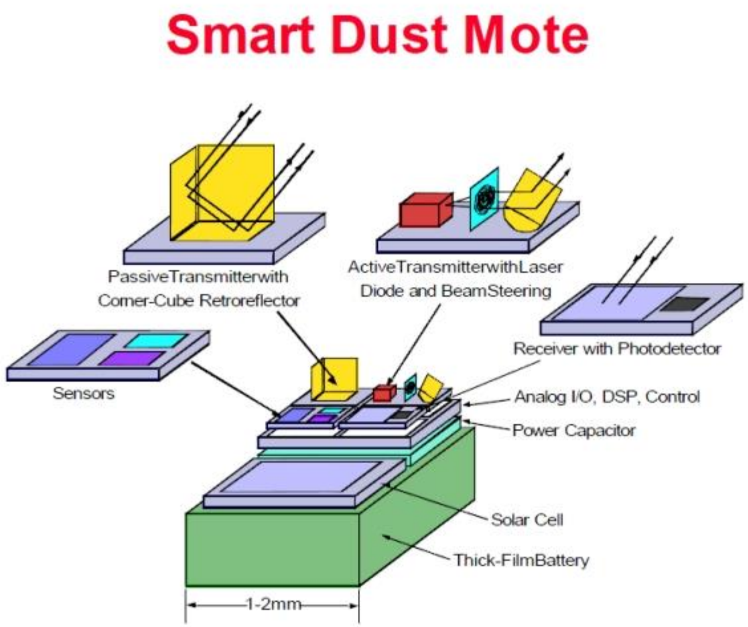


Smart Dust

- Micro fabrication and integration of low-cost sensors, actuators and computer controllers, MEMS (Micro Electro-Mechanical Systems)
- Can be sprayed & embedded throughout the digital environment
- Creating a digital skin that senses physical & chemical phenomena

Sensor Nodes

- Connect wirelessly
  - WiFi, Zigbee, 802.15.4., NFC
  - GPRS, 3G, 4G
- Small Sensors
  - Smart dust (<1mm)
- Battery
  - Months, Years
  - Infinite?
- Cheap
  - Today: 50 – 700€ per node



Kahn et al., Next Century Challenges: Mobile Networking for „Smart Dust“, 1999

Sensor Nodes

- Small & Cheap
- Battery-powered
- Wireless

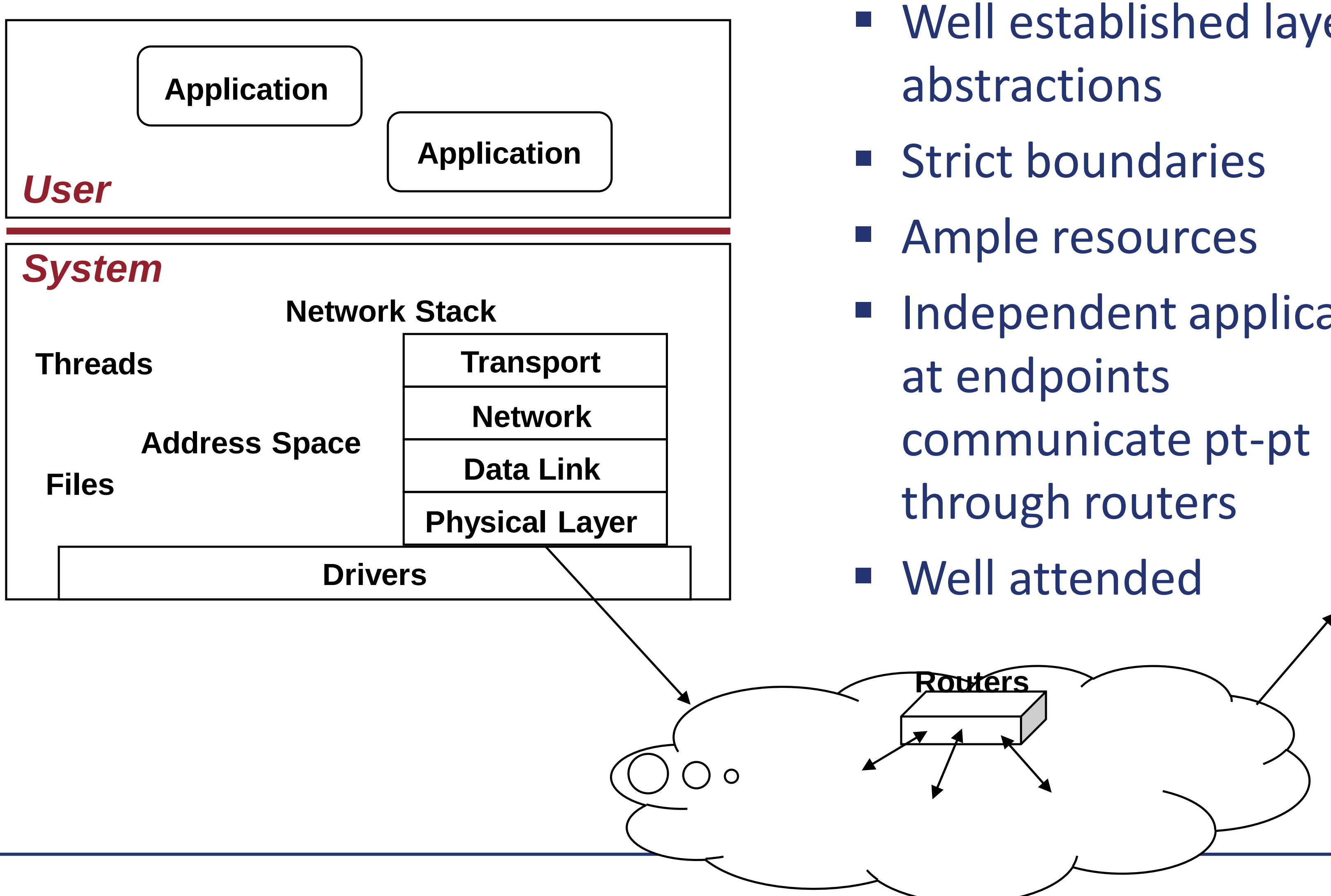
Combination of:

- Breakthroughs in MEMS technology
- Development of low power radio technologies
- Advances in low-power embedded microcontrollers

Challenges

- Deployment
  - Self configuration
  - (Real-time) Operating System
  - Sensor installation
- Wireless Communication
  - Standards
    - Zigbee, IPv6 (6LowPan, RPL)
  - Routing, Clustering, Topology, etc.
- Energy
  - Energy-efficient algorithms
  - Energy harvesting

OS - Traditional Systems



- Well established layers of abstractions
- Strict boundaries
- Ample resources
- Independent applications at endpoints communicate pt-pt through routers
- Well attended

Choice of Programming Primitives

- Traditional approaches
  - Command processing loop (wait, request, act, respond)
  - Monolithic event processing
  - Full thread/socket posix regime

- Alternative
  - Provide framework for concurrency and modularity
  - Never poll, never block
  - Interleaving flows, events

Threads / Events

Threads

- Thread stacks must be allocated at creation time
- Stack Memory Over-Provisioned
  - No Virtual Memory System
  - No Memory Protection Mechanisms
- Locking for state exclusion
- Stack memory is inaccessible to other threads

Events

- No locking
- Only one running at a time
  - Only one stack
  - Locking rare
- Hard to express some things as state machine
- Problem: Cryptography takes 2 seconds, everything else blocked.
- Solution: Preemptive Threads

TinyOS

- Microthreaded OS (lightweight thread support) and efficient network interfaces
- Two level scheduling structure
  - Long running tasks that can be interrupted by hardware events
- Small, tightly integrated design that allows crossover of software components into hardware

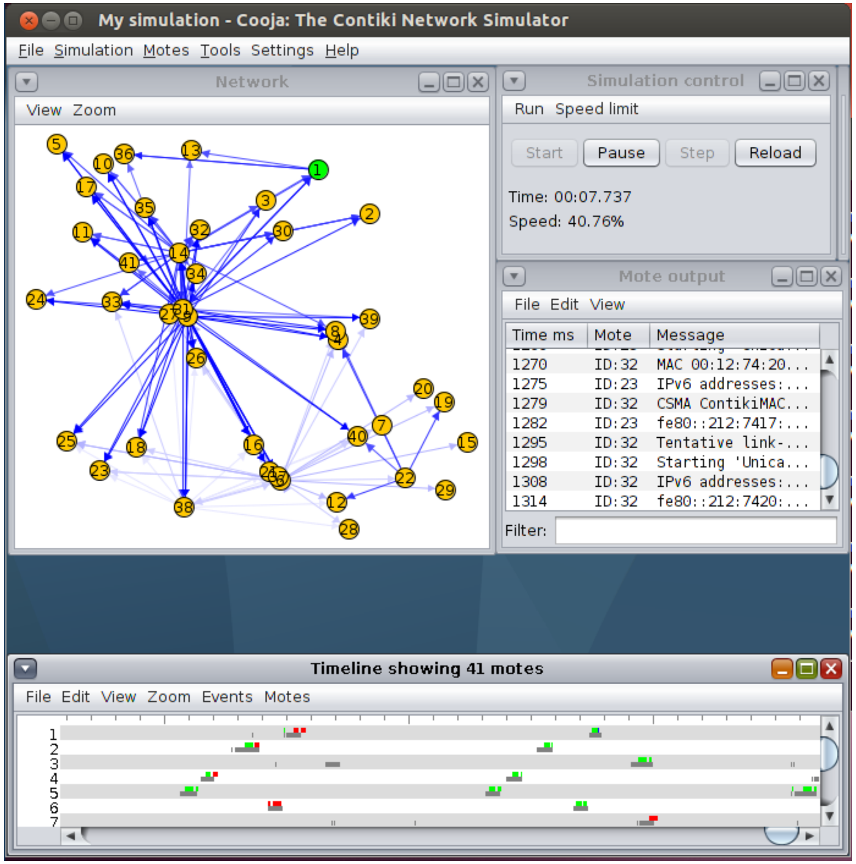
Event-Driven Sensor Access Pattern

- Clock event handler initiates data collection
- Sensor signals data ready event
- Data event handler calls output command
- Device sleeps or handles other activity while waiting
- Conservative send/ack at component boundary

```
command result_t
StdControl.start() {
    return call
Timer.start(TIMER_REPEAT, 200);
  }
event result_t Timer.fired() {
    return call
sensor.getData();
  }
event result_t
sensor.dataReady(uint16_t data)
{
    display(data)
    return SUCCESS;
  }
```

Contiki

- Dynamic loading of programs (vs. static)
- Multi-threaded concurrency managed execution
  - In addition to event driven
- Available on MSP430, AVR, HC12, Z80, 6502, x86, ...
- Simulation environment available for BSD/Linux/Windows



Event-based kernel with threads

- Kernel is event-based
  - Most programs run directly on top of the kernel
- Multi-threading implemented as a library
- Threads only used if explicitly needed
  - Long running computations, ...
- Preemption possible
  - Responsive system with running computations

Multi-Thread API

`mt_yield();`

Yield from the running thread.

`mt_post(id, event, dataptr);`

Post an event from the running thread.

`mt_wait(event, dataptr);`

Wait for an event to be posted to the running thread.

`mt_exit();`

Exit the running thread.

`mt_start(thread, functionptr, dataptr);`

Start a thread with a specified function call.

`mt_exec(thread);`

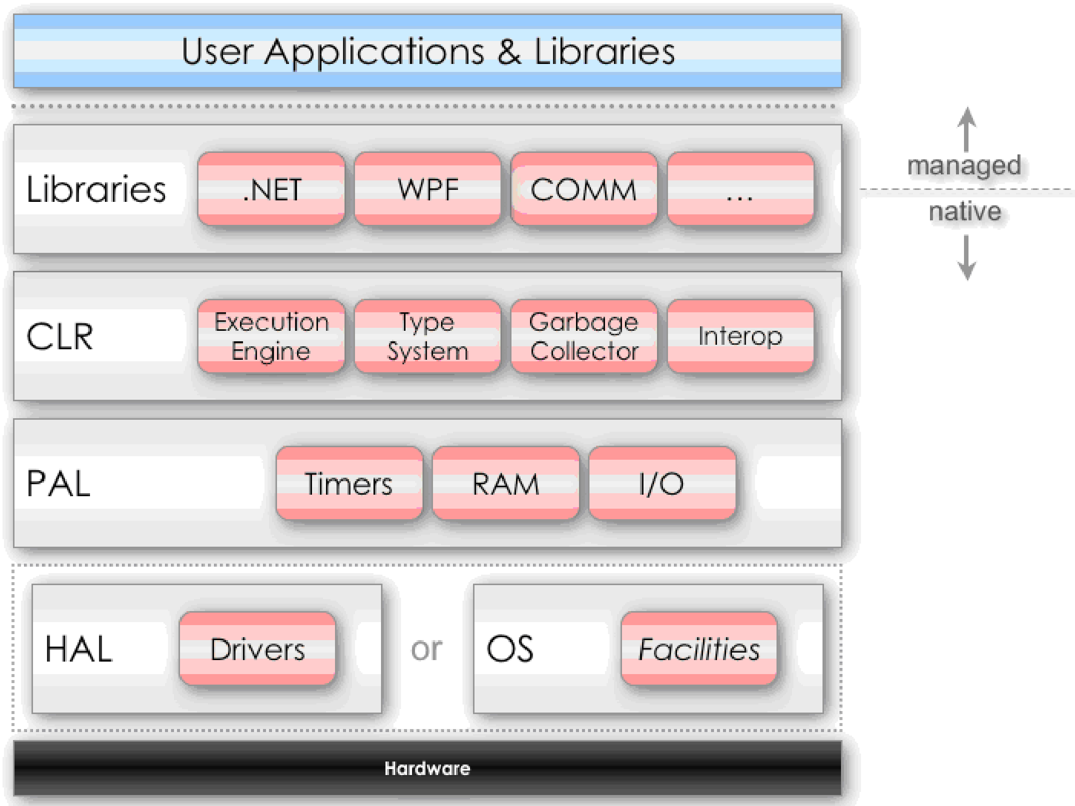
Execute the specified thread until it yields or is preempted.

Figure 4. The multi-threading library API.

.NET MicroFramework (MF)

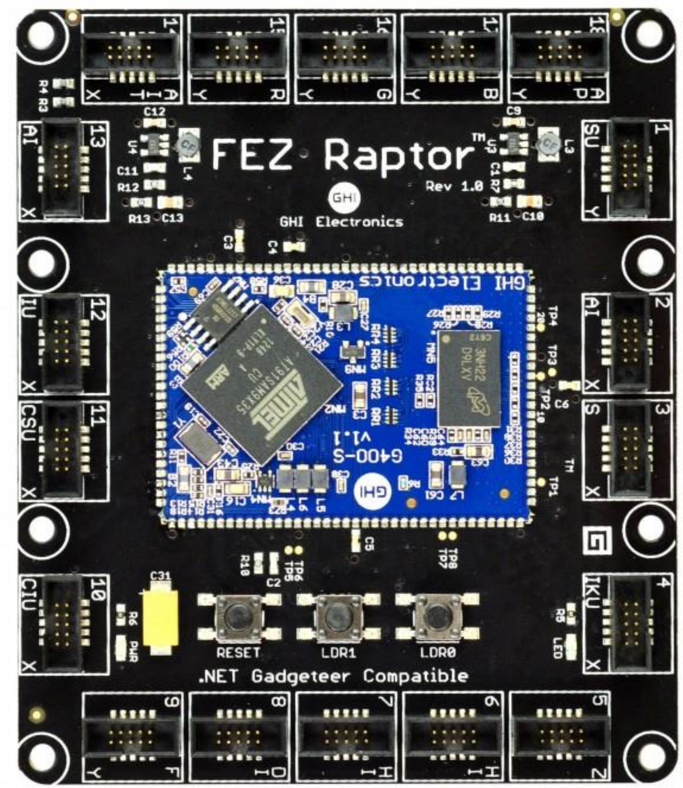
- .NET MF is a bootable runtime environment tailored for embedded development

- MF services include:
  - Boot Code
  - Code Execution
  - Thread Management
  - Memory Management
  - Hardware I/O



Hardware Abstraction Layer (HAL)

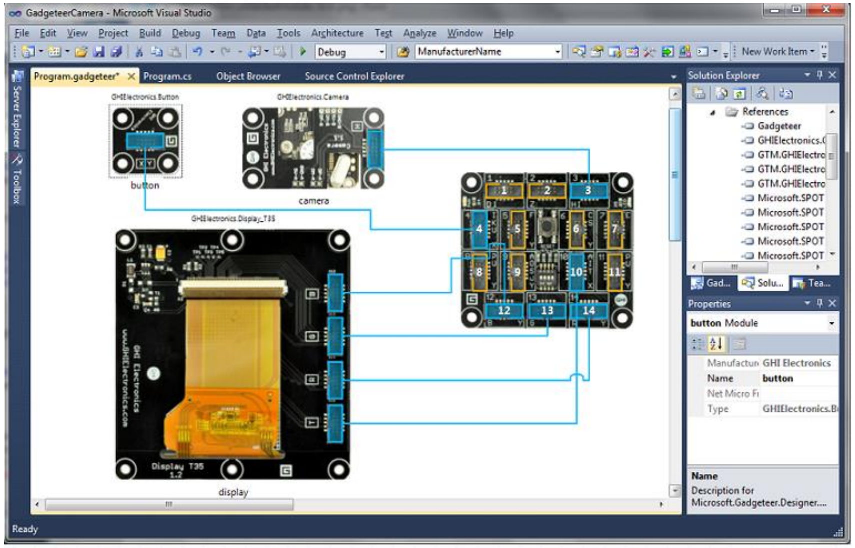
- Provides an interface to access hardware and peripherals
  - Relevant only for system, not application developers
- Does not require operating system
  - Can run on top of one if available
- Interfaces include:
  - Clock Management
  - Core CPU
  - Communications
  - External Bus Interface Unit (EBIU)
  - Memory Management
  - Power
  - Watchdog Timer
  - Security



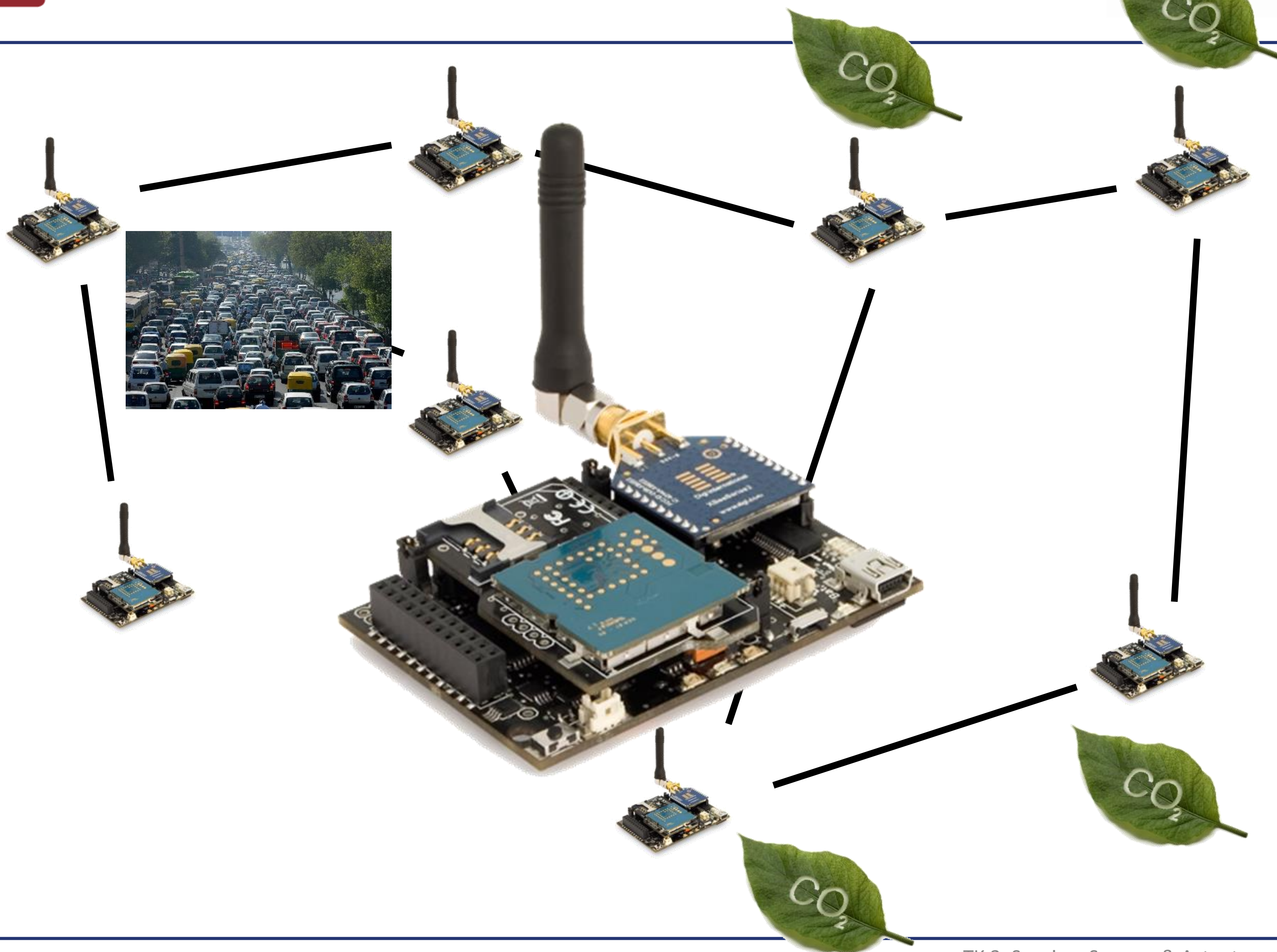
Platform Abstraction Layer (PAL)

- Provides hardware independent abstractions
  - Used by application developers to access system resources
  - Application calls to PAL managed by Common Language Runtime (CLR)
  - In turn calls HAL drivers to access hardware

- PAL interfaces include:
  - Time
  - PAL Memory Management
  - Input/Output
  - Events
  - Debugging
  - Storage
  - PAL Communications
  - Asynchronous Procedure Call
  - Bootstrap

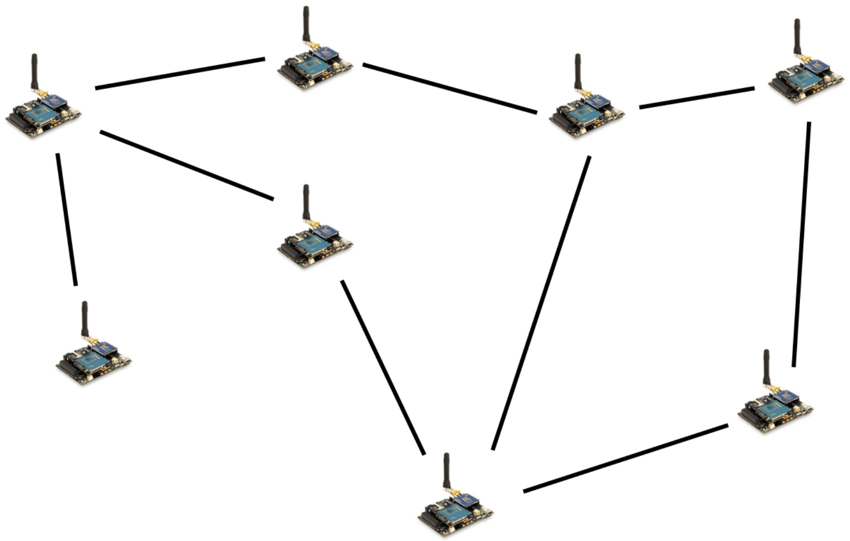


Sensor Nodes



Definition

- Wireless Sensor Networks (WSNs):
  - Highly distributed networks of small, lightweight wireless nodes,
  - Deployed in large numbers,
  - Monitors the environment or system by measuring physical parameters such as temperature, pressure, humidity.
- Node:
  - sensing + processing + communication



Wireless Sensor Networks

- Animal tracking & monitoring
  - Zebranet (2004) / Great Duck Island (2002) / SWIM (2003)

- Vehicle & people tracking
  - Exscale (2004)

- Data centers
  - RACNet (2009)
- Wearable WSNs
  - Mercury (2009)

- Enviromental Sensing
  - da_sense (2011)

Wireless Sensor Networks

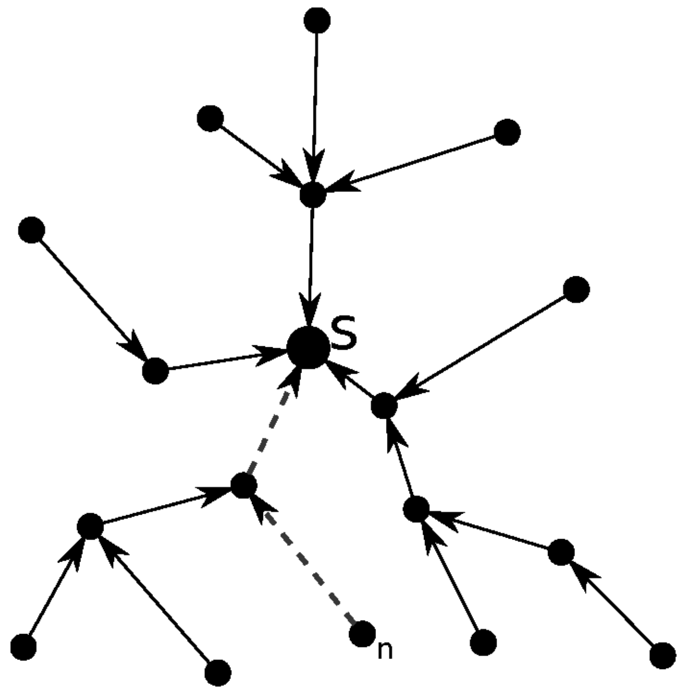
- Machine and vehicle monitoring
  - Sensor nodes in moveable parts
  - Monitoring of temperatures, fluid levels, …
  - Calculation of maintenance intervals
- Smart Cities
  - Intelligent buildings, building monitoring
  - Intrusion detection, mechanical stress detection
  - Smart heating, ventilating, air conditioning
- Health & medicine
  - Long-term monitoring of patients with minimal restrictions
  - Assisted Living
- Environmental monitoring, person tracking
  - Monitoring of wildlife and national parks
  - Cheap and (almost) invisible person monitoring
  - Monitoring demilitarized zones, ...
- Logistics
  - Dataloggers: Temperature, Humidity, Acceleration/Shock
  - … and many more

Wireless Sensor Networks

- Periodic Reporting
  - Plant Monitoring
  - Environmental Monitoring
- Event Detection
  - Fire, Chemicals, etc.
- Tracking
  - Person Tracking
  - Vehicle Tracking

Wireless Sensor Networks

- Most common application: Periodic reporting
- Two quality metrics
  - Frequent data samples
    - Sampling rate should be high /Energy consumption should be low
  - Coverage problems

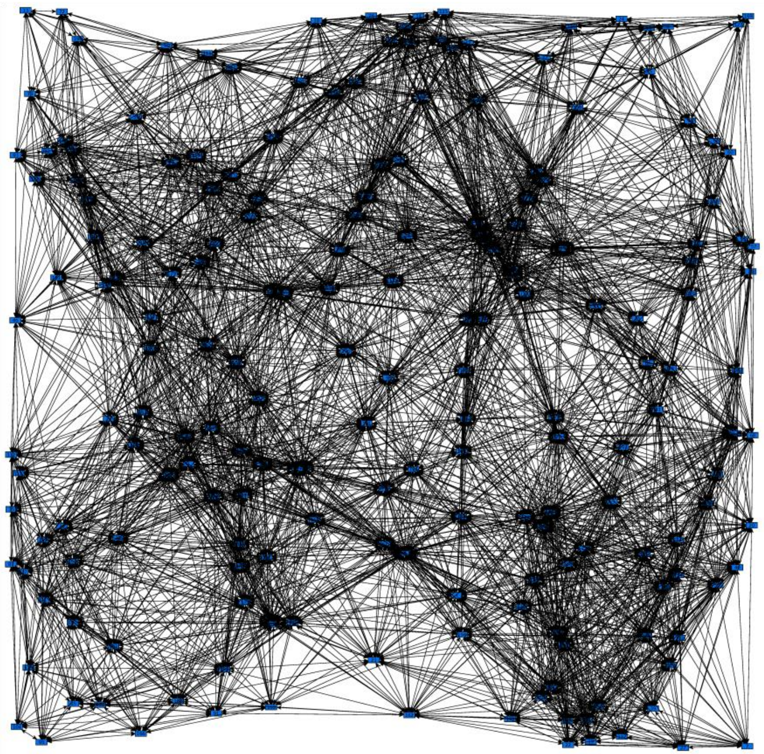


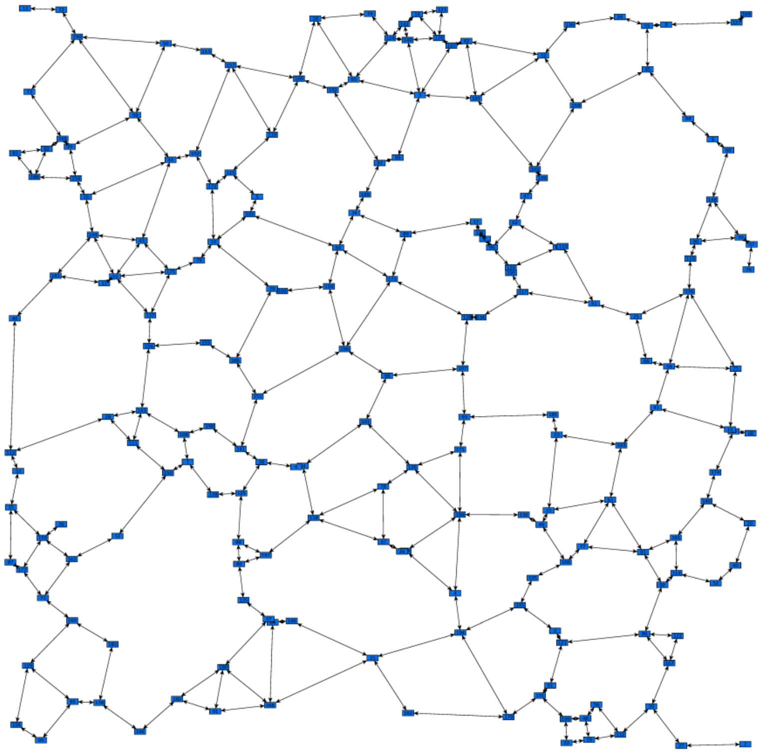
Coverage problem

- Definition: the measurements of quality of service that can be provided by a particular sensor network

- Area coverage: Cover a given area
- Point coverage: Cover a set of (static) targets
- Detectability: Determine the maximal support / breach paths that traverse a sensor field

Topology control





Frequent data sampling

- Most common application: Periodic reporting
  - Frequent data samples
  - Data density should be high, therefore high frequency
  - Energy consumption should be low, therefore low frequency
- Energy-efficiency
  - Given a sampling rate s how can the network survive as long as possible
- Energy-harvesting
  - Use solar cells (or other energy sources)
  - Now we might reach perpetual operation at sampling rate s
  - New question: How high can s be, without depleting one sensor node!

Solar-aware distributed flow

- A node needs energy for:
  - Own sampling
  - Own messages
  - Messages of other nodes!

- How can we determine the other nodes routed through us?

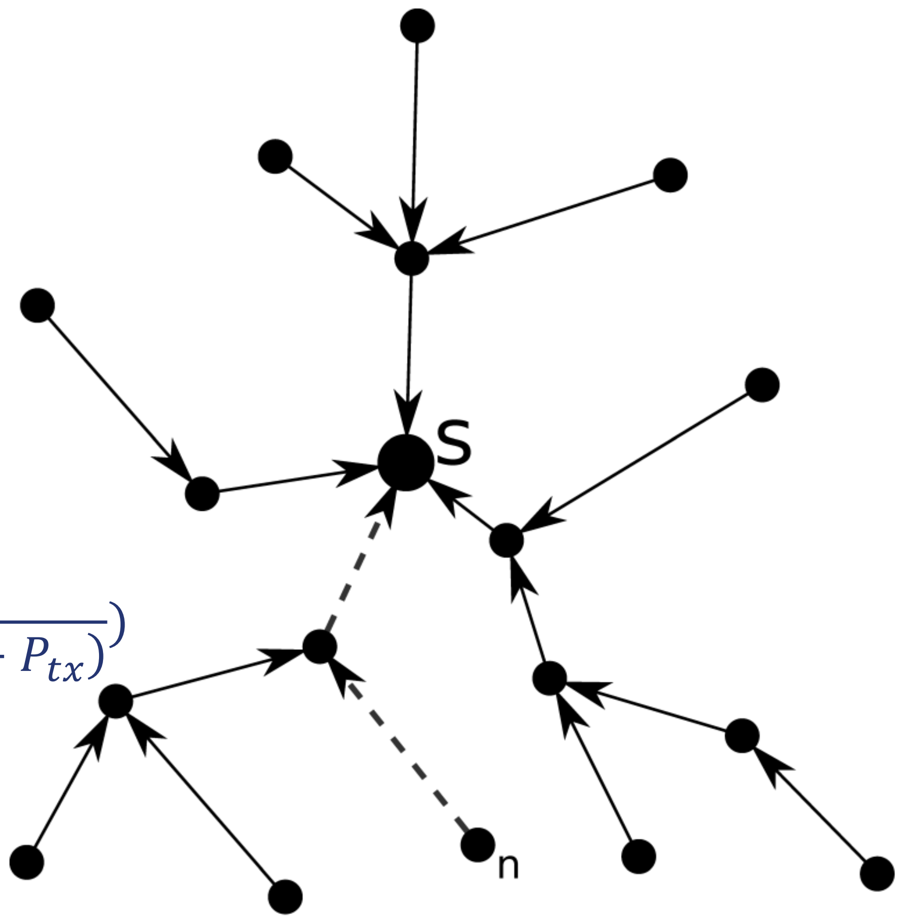
- How can we determine the energy available to us over time?

- How can we calculate a sampling rate for us and all nodes routed through us?

Solar-aware distributed flow

- Predict available energy c
  - Solar-energy is predictable!
- Energy consumption
  - Sensor sampling (P_{sense})
  - Sending messages (P_{tx})
  - Receving predecessor messages (P_{rx})
- Two bounds
  - Own available energy
  - Allowed flow from successor

$$x = \min\left(\frac{f_a}{n+1}, \frac{c}{n * (P_{rx} + P_{tx}) + (P_{sense} + P_{tx})}\right)$$

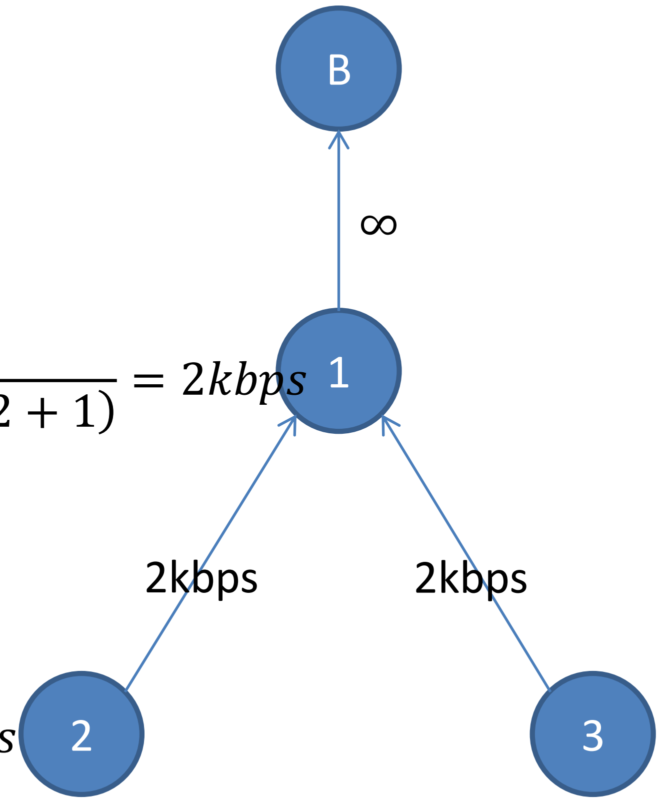


- Allocate energy
  - Set own sampling rate
  - Send control messages to direct predecessors

Solar-aware distributed flow

- P_{rx} = 1,5 mW/kbps
- P_{tx} = 2 mW/kbps
- P_{sense} = 1 mW/kbps

- c_1 = 20mW
- c_2 = 9mW
- c_3 = 3mW



$$x_1 = \frac{20}{2(2+1{,}5)+(2+1)} = 2kbps$$

$$x_2 = \min(2, \frac{9}{(2+1)}) = 2kbps$$

$$x = \min(\frac{f_a}{n+1}, \frac{c}{n*(P_{rx}+P_{tx})+(P_{sense}+P_{tx})})$$

$$x_3 = \min(2, \frac{3}{(2+1)}) = 1kbps$$

Energy Harvesting

- Solar Energy
  - Predictable, efficient
- Wind Energy
  - Bulky, less predictable
- Vibrations
  - Machines, unpredictable
- ...

Challenges

- Deployment
  - Self configuration
  - (Real-time) Operating System
  - Sensor installation
- Wireless Communication
  - Standards
    - Zigbee, IPv6 (6LowPan, RPL)
  - Routing, Clustering, Topology, etc.
- Energy
  - Energy-efficient algorithms
  - Energy harvesting

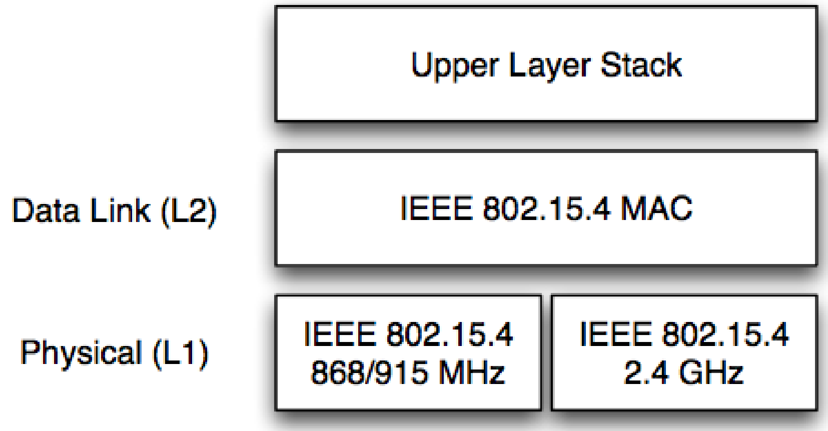
Zigbee

- Bluetooth - Desktop / Personal Area Net: few, „valued“ devices
- ZigBee – scales up to sensor networks („smart dust“) in terms of power, #of nodes, management …

Market Name Standard	GPRS/UMTS (TDMA/CDMA)	Wi-Fi™ 802.11b	Bluetooth™ 802.15.1	ZigBee™ 802.15.4
Application Focus	LongDist. Voice/Data	Web, Email, Video	Cable Replacement	Monitoring & Cntrl
System Resources	16MB+	1MB+	250KB+	4KB - 32KB
Battery Life (days)	1-7	.5 - 5	1 - 7	100 - 1,000+
Network Size	(1)	(32)	7	255 / 65,000
Bandwidth (kb/s)	14 - 2000	11,000+	720	20 - 250
Transmission Range (m)	1,000+	1 - 100	1 - 10+	1 - 100+
Success Metrics	Reach, Quality	Speed, Flexibility	Cost, Convenience	Reliab., Power, Cost

IEEE 802.15.4

- Important standard for home networking, industrial control and building automation
- Three PHY modes
  - 20 kbps at 868 MHz
  - 40 kbps at 915 MHz
  - 250 kbps at 2.4 GHz (DSSS)
- Beaconless mode
  - Simple CSMA algorithm
- Beacon mode with superframe
  - Hybrid TDMA-CSMA algorithm
- Up to 64k nodes with 16-bit addresses
- Extensions to the standard
  - IEEE 802.15.4a, 802.15.4e, 802.15.5



ZigBee Node Types, Protocol Stack

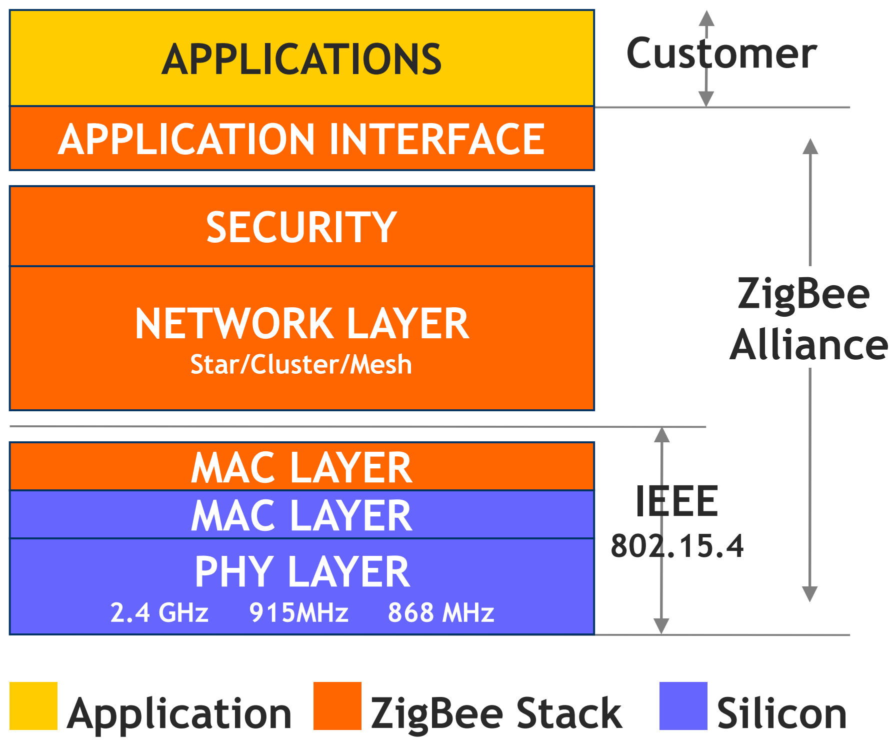
TECHNISCHE UNIVERSITÄT DARMSTADT

a) FFN vs. RFN: Full Function Nodes– Reduced Function Nodes

b) Coordinator vs. Router vs. EndNode

Only EndNode may (!) be RFN

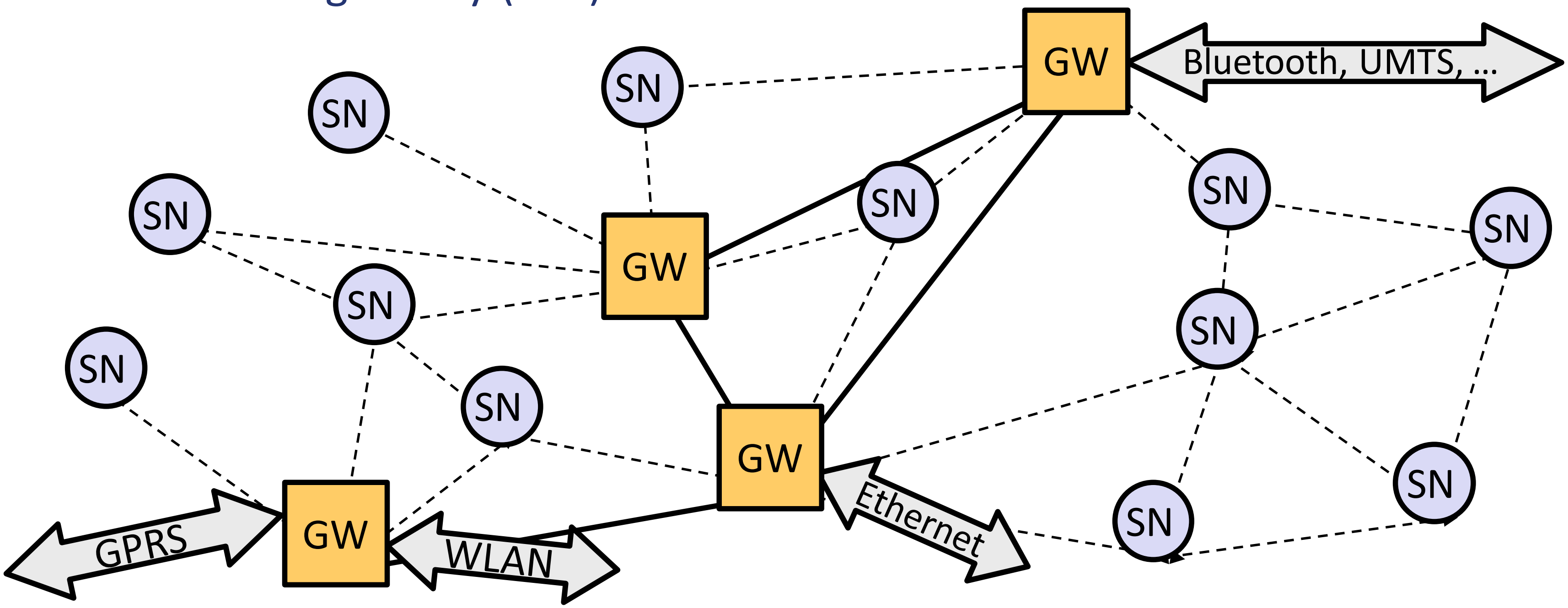
- Microcontroller utilized
- FFN protocol stack <32 k
- RFN protocol stack ~4k
- Coordinators: extra RAM
  (DBs f. nodes/transactions/pairing)
- PHY: OQPSK (2.4GHz); CDMA
- MAC: CSMA/CA



ZigBee Net can be replaced w/ IEEE802 MAC + IP

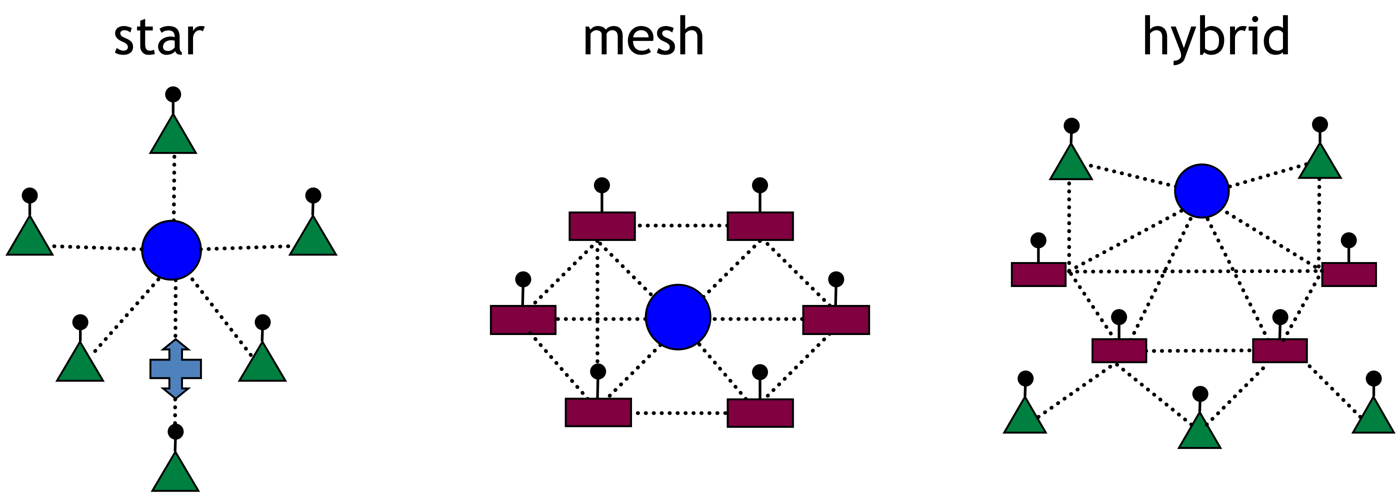
Wireless Sensor Networks

- Typical topology
  - Sensor nodes (SN) monitor and control the environment
  - Nodes process data and forward data via radio
  - Integration into the environment, typically attached to other networks over a gateway (GW)



ZigBee: More Characteristics

- 3 Bands: ISM-Europe (868 MHz) / ISM US (915) / 2.4 GHz →
  - Different no. of available channels (1/10/16), speeds (20/40/250 kbps) no. of chips/symbol in CDMA (15/15/32 – note: WLAN has 11)
  - Other limitations e.g. duty cycling for 868 MHz in Europe
- 3 network types, made of:



- *star:* low battery → low reliability; *mesh:* opposite (+routg complex!)
- *hybrid:* tradeoff possible

ZigBee: MAC Options

- Two channel access mechanisms

- Non-beacon network: unslotted CSMA/CA
  - Acknowledgement for successfully received packets

- Beacon-enabled network
  - ZigBee router transmits periodic beacons
    - 15ms to 252sec (15.38ms*2n where 0 < n < 14)
    - 16 equal-width time slots between beacons
    - Channel access in each time slot is contention free
  - Nodes can sync with beacons and may sleep between beacons

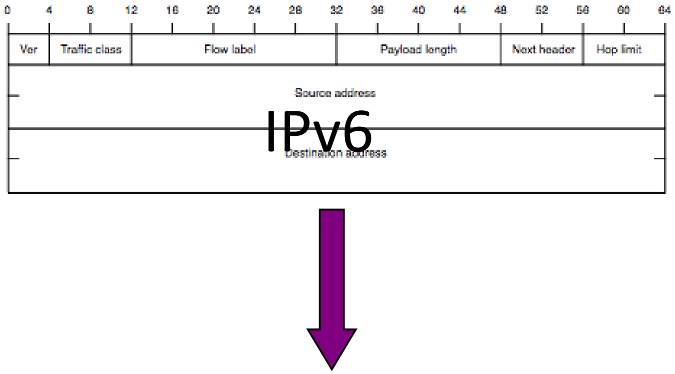
ZigBee

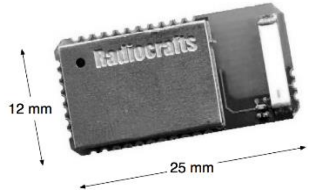
- Low power
- Many Nodes
- Based on 802.15.4
- Some security

- Proprietary stack up to the application layer
- Special nodes
- Limited choice of network protocols
- Interoperability?

What is 6LoWPAN?

- IPv6 over Low-Power wireless Area Networks
- Defined by IETF standards
  - RFC 4919, 4944
  - draft-ietf-6lowpan-hc and -nd
  - draft-ietf-roll-rpl
- Stateless header compression
- Enables a standard socket API
- Minimal use of code and memory
- Direct end-to-end Internet integration
  - Multiple topology options





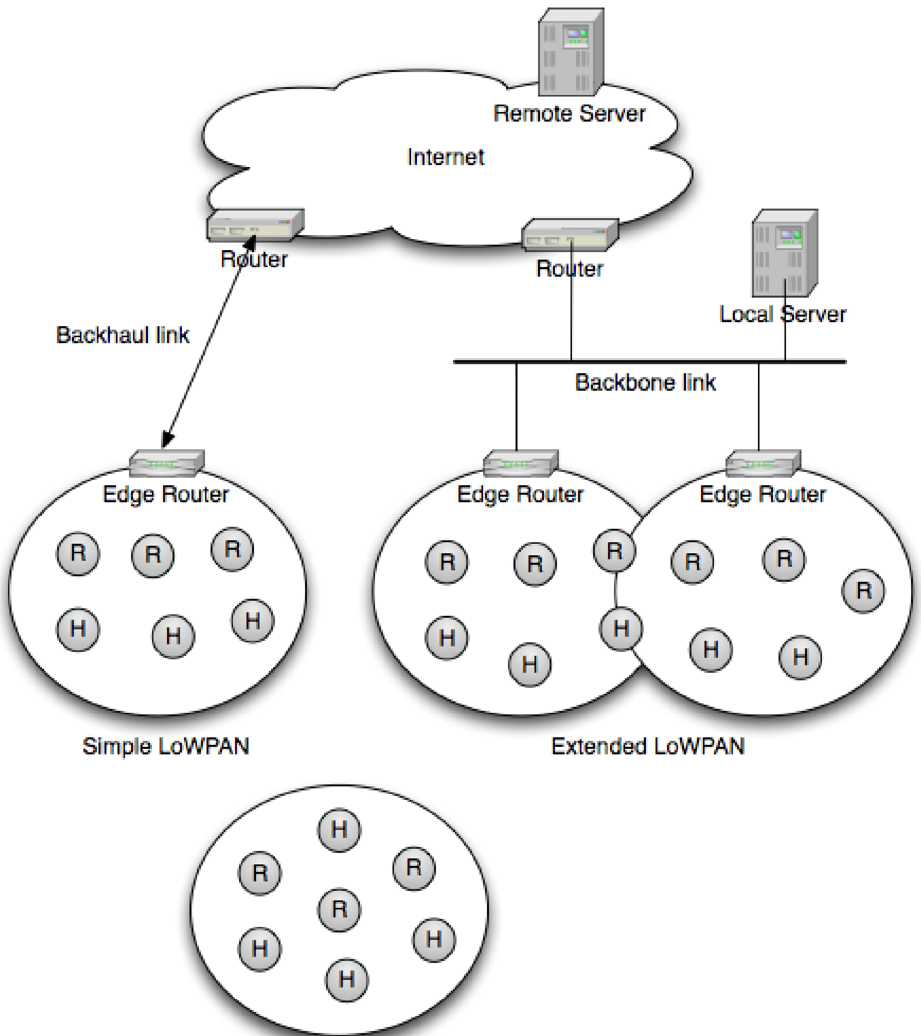
Protocol Stack

TCP/IP Protocol Stack		6LoWPAN Protocol Stack
HTTP / RTP	Application	Application
TCP / UDP / ICMP	Transport	UDP / ICMP
IP	Network	IPv6 with LoWPAN
Ethernet MAC	Data Link	IEEE 802.15.4 MAC
Ethernet PHY	Physical	IEEE 802.15.4 PHY

Features

- Support for e.g. 64-bit and 16-bit 802.15.4 addressing
- Useful with low-power link layers such as IEEE 802.15.4, narrowband ISM and power-line communications
- Efficient header compression
  - IPv6 base and extension headers, UDP header
- Network autoconfiguration using neighbor discovery
- Unicast, multicast and broadcast support
  - Multicast is compressed and mapped to broadcast
- Fragmentation
  - 1280 byte IPv6 MTU -> 127 byte 802.15.4 frames
- Support for IP routing (e.g. IETF RPL)
- Support for use of link-layer mesh (e.g. 802.15.5)

Architecture



Architecture

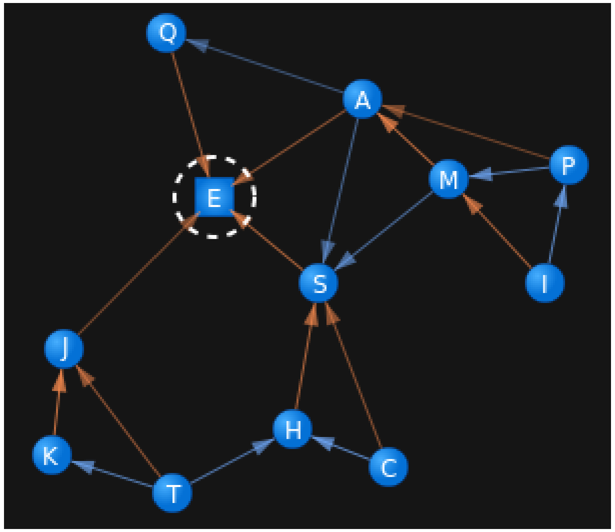
- LoWPANs are stub networks
- Simple LoWPAN
  - Single Edge Router
- Extended LoWPAN
  - Multiple Edge Routers with common backbone link
- Ad-hoc LoWPAN
  - No route outside the LoWPAN
- Internet Integration issues
  - Maximum transmission unit
  - Application protocols
  - IPv4 interconnectivity
  - Firewalls and NATs
  - Security

IPv6	
Ethernet MAC	LoWPAN Adaptation
	IEEE 802.15.4 MAC
Ethernet PHY	IEEE 802.15.4 PHY

IPv6-LoWPAN Router Stack

Routing Protocol for Low-Power and Lossy Networks (RPL)

- Works on top of 6LoWPAN / IPv6
- IETF Standard RFC 6550
- RPL builds a DODAG
  - One per base station
- Parent is picked based on weight
  - Hop to destination + custom function



Challenges

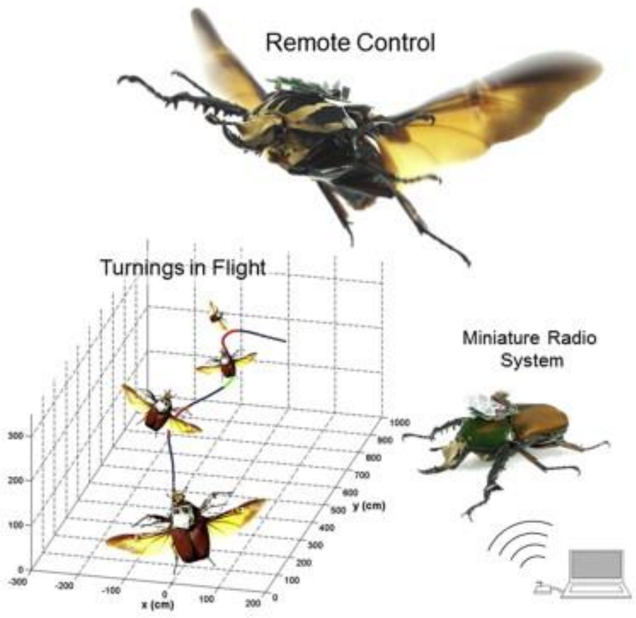
- Deployment
  - Self configuration
  - (Real-time) Operating System
  - Sensor installation
- Wireless Communication
  - Standards
    - Zigbee, IPv6 (6LowPan, RPL)
  - Routing, Clustering, Topology, etc.
- Energy
  - Energy-efficient algorithms
  - Energy harvesting

Wireless Sensor Networks

- Periodic Reporting
  - Plant Monitoring
  - Environmental Monitoring
- Event Detection
  - Fire, Chemicals, etc.
- Tracking
  - Person Tracking
  - Vehicle Tracking

Stuff we haven‘t talked about

- Personal Sensor Networks
  - Connected wearables
- Miniature Air Vehicles (MAV)
  - Drones
  - Insects
  - …

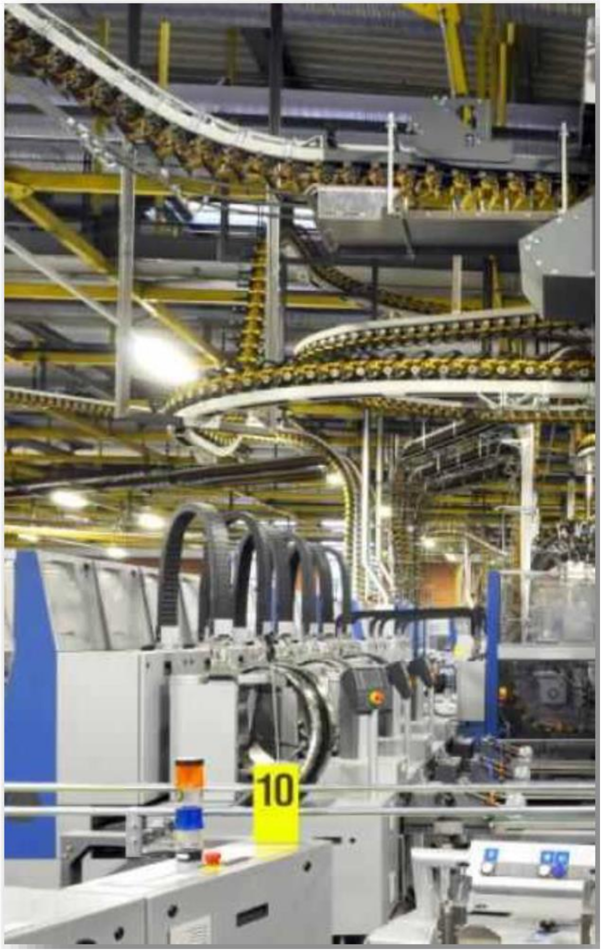


Cyper Phyisical Systems

Cyber-Physical (cy-phy) Systems (CPS) are integrations of computation with physical processes [Edward Lee, 2006].

Fabrication

- Production resources are self-configuring and distributed social machines
- Personalization
- 3D-Printers



Structural safety

- Sensors + data analysis

Möhne lake dam

Kilauea, Hawaii

Bridge at Vancouver

Taipeh 101

Smart Home

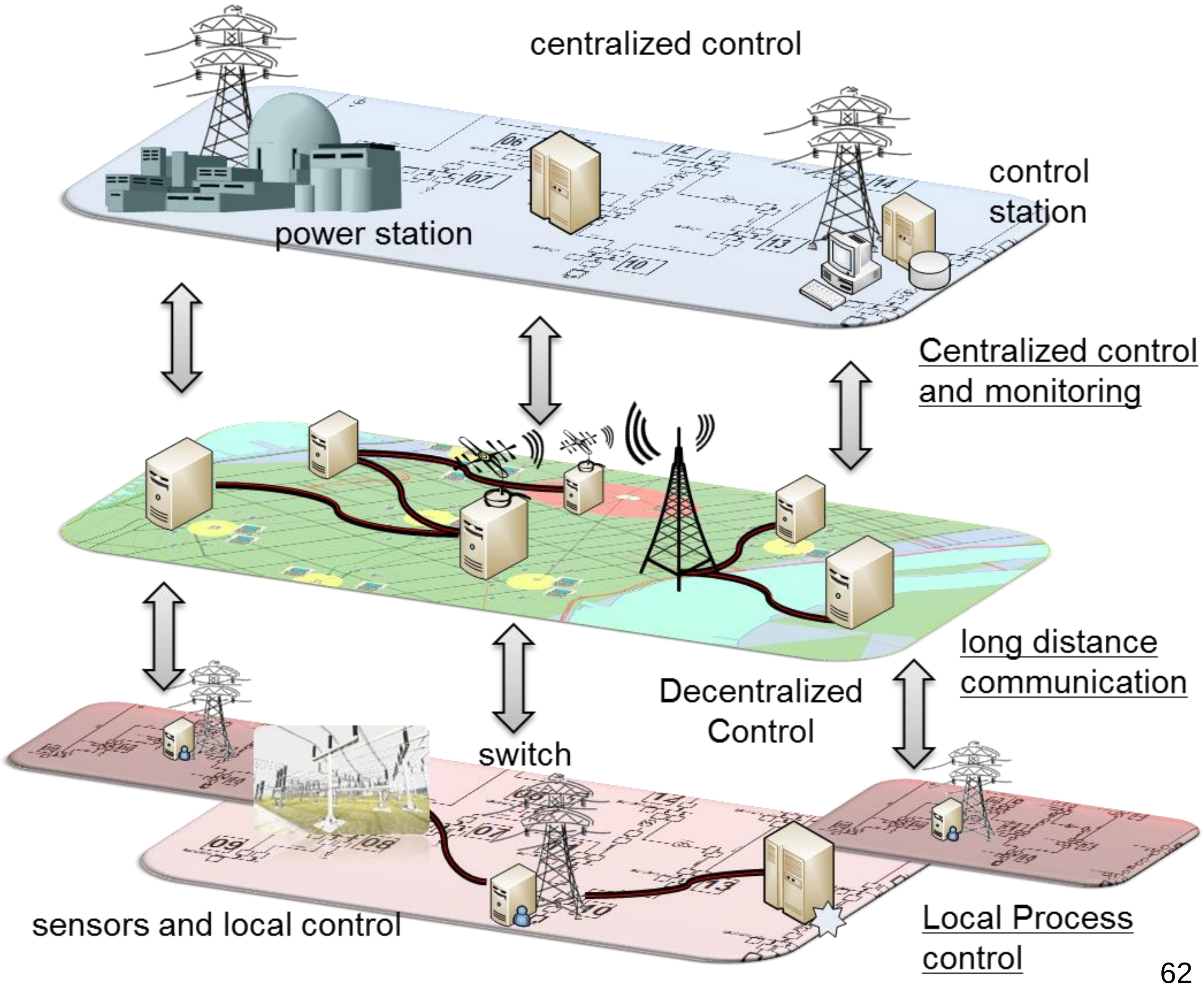
- Zero energy building, generates as much energy as it consumes
- Provides safety and security
- Supports owners
- Provides maximum comfort
- ambient assisted living

Smart Medicine

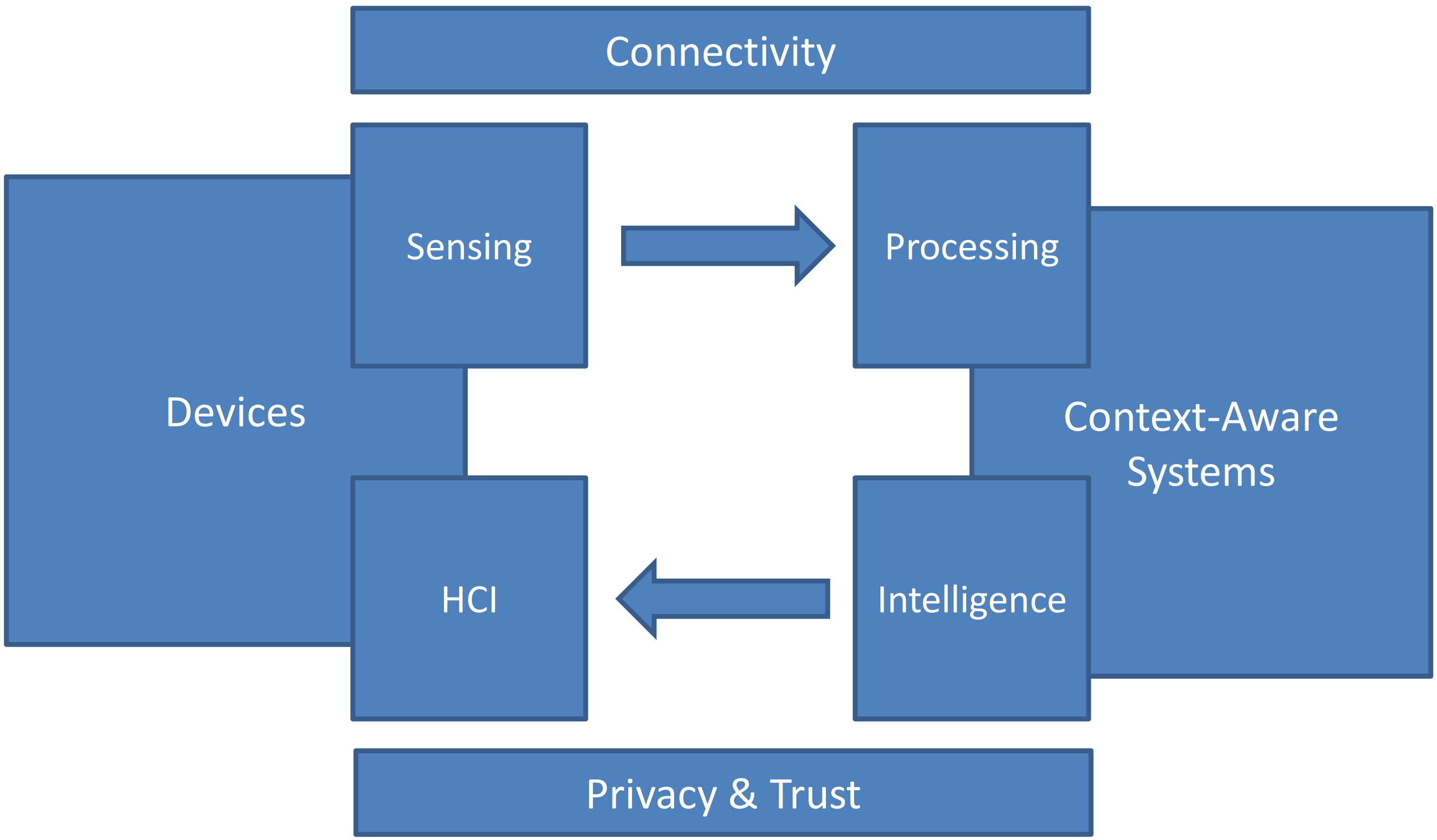
- Diagnosis
- Support of therapy
- Evaluation
- Risk analysis
- Information about patients

- Personalized Medicine

Smart Grid


centralized control
control
station
power station
Centralized control
and monitoring
long distance
communication
Decentralized
Control
switch
sensors and local control
Local Process
control
62

CPS



Connectivity
Sensing
Processing
Devices
Context-Aware Systems
HCI
Intelligence
Privacy & Trust

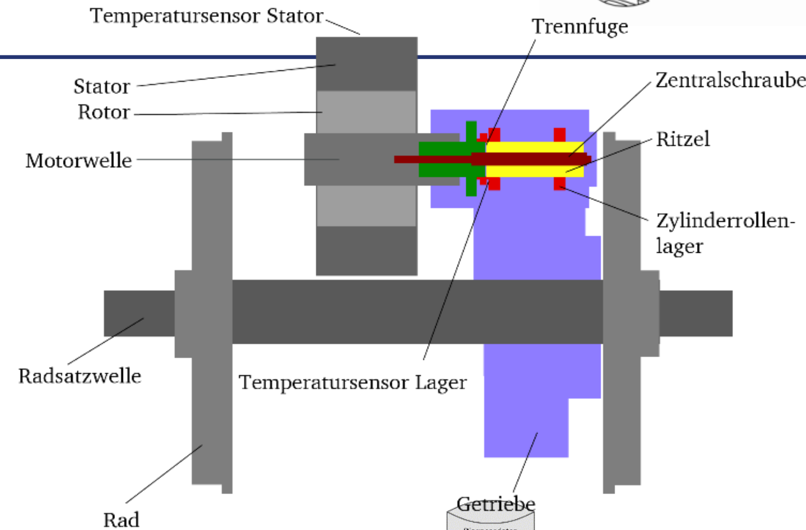
CPS & UC

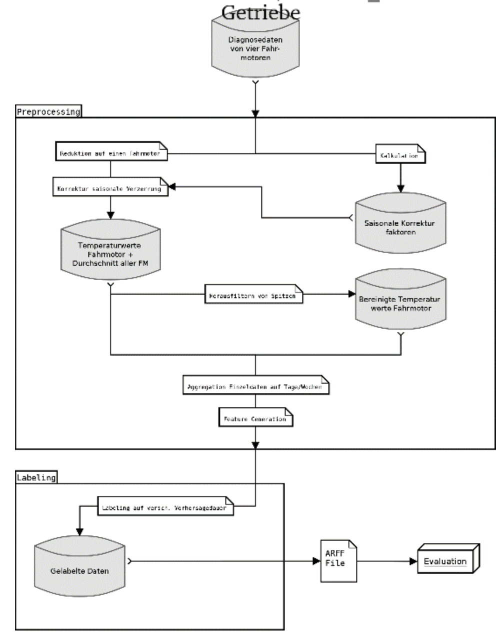
- Fast reaction times
  - Real-time constraints
- Humans out of the loop
- Safety, security

- Same building blocks
- Enabled by connectivity & smaller, more powerful devices

Predictive Maintenance

- Predictive Analytics: Using diagnosis, sensor, and maintenance logs to predict future failures
- Example: Failure of central screw breaks the complete engine
  - Cost ration (Repair / Maintenance) 50:1
  - Two temperature sensors





Summary

- Embedded Systems
- Sensor Nodes

- Wireless Sensor Networks

- Cyper-Physical Systems