Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
D-70569 Stuttgart

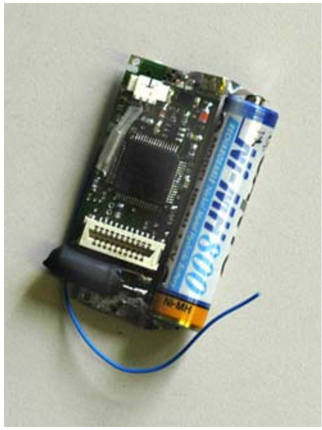
System Aspects of Sensor Networks

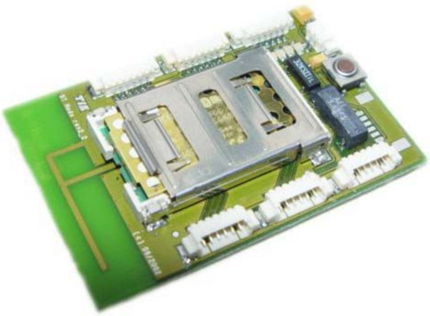
Kurt Rothermel
Universität Stuttgart
kurt.rothermel@informatik.uni-stuttgart.de

Existing Hardware Platforms

- Processor technology
  - Atmel ATmega128L: BTnodes, MICA2
  - PIC 18F6720: Smart-Its
  - TI MSP 430F149: ESB
- Memory capacities:
  - RAM 2kB (ESB) - 64kB (BTnodes)
  - Flash 60kB (ESB) - 640kB (MICA2)
  - EEPROM 4kB (BTnodes, MICA2) - 33kB (Smart-Its)
- Radio technology with 19.2, 38.4, or 125 kbps

BUT: also sensors in cars, shopping malls, ...

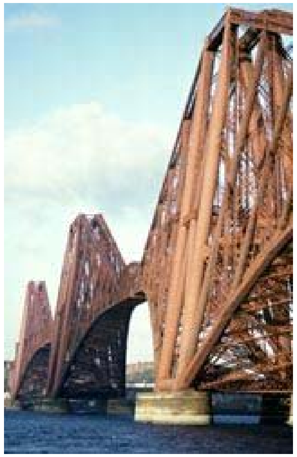




Application: „Sustainable Bridges“

Goal: Continuous monitoring of bridges using sensors (vibration, acoustic emission) to detect cracks

- Current inspections are done visually or using sensors connected with kilometers of cable
- Acoustic emission techniques need data with sampling rate of approx. 40 kHz!
- Localization of cracks needs synchronized clocks within 15-25µsec!
- Minimum required sensor lifetime is at least 3 years!



Application: Habitat monitoring (e.g. bird watching)

Goal: Non-disruptive monitoring of animals in their natural environment

- Use of wide range of sensors, e.g., light, temperature, humidity,barometric pressure, infrared detector, etc.
- Need for standard database functions
  - Which nests are occupied?
  - What is the MIN/MAX/AVR temperature/humidity of the nests?

Abstractions

	Sustainable Bridges	Habitat Monitoring
Communication	Typically Push: Event triggered	Typically Pull: Query-based
Programming paradigm	Publish/Subscribe Specialized functions	Generic query-based interface Standard functions
Distribution transparency	No (Triangulation requires topology information)	Yes (sensor network looks like a DB)

QoS Requirements

	Sustainable Bridges	Habitat Monitoring
Lifetime	> 3 years	Several months
Real-Time	Data synchronization and precise event reporting are critical	No hard real-time constraints
Reliability	Reliability is crucial: people might be in danger if data is lost	Reliability is desired, but not critical

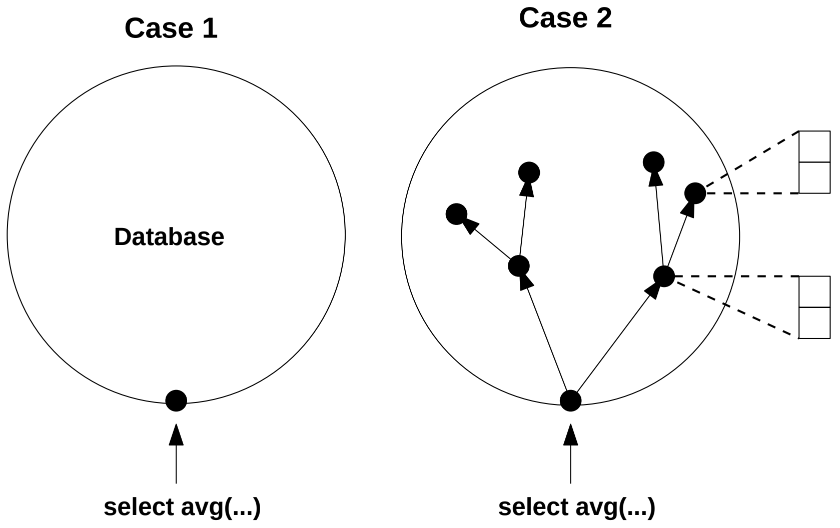
Functions

	Bridges	Habitat
Communication (Pull/ Push) ◦ unicast, multicast, broadcast, geocast ◦ single-hop ◦ multi-hop (routing often application-specific) ◦ scoped	+++	+++
Aggregation ◦ diffusion queries/results ◦ optimizations (often depending on aggregate function)	+	+++
Time Synchronization	+++	+
Security	+++	o
Systems Management (Code Deployment)	+++	+
Data Storage	o	+++
Energy management	+++	++
Integration of a wide range of sensors / sensor fusion	++	+++

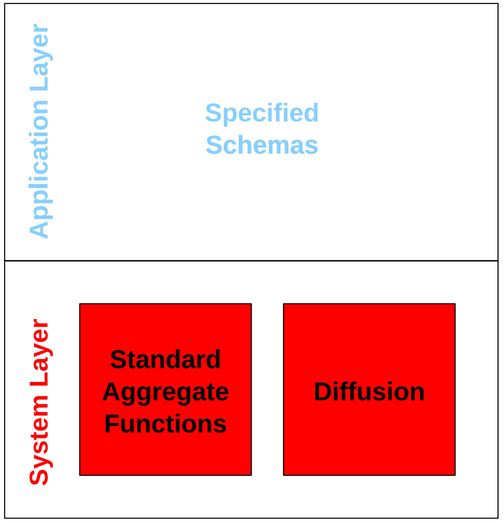
Specific Issues of Aggregation

Two extremes:

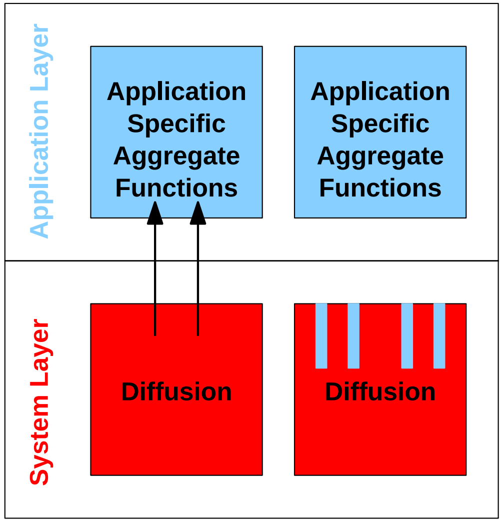
- Programmer defines schema
- Programmer develops aggregation and diffusion algorithms
  - external aggregation
  - in-network aggregation



In-Network Aggregation

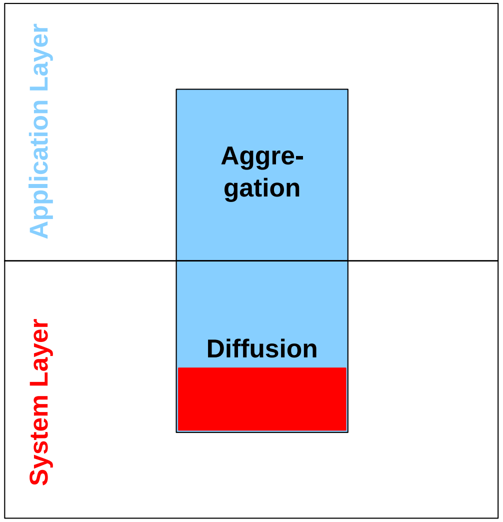


(Standard) Aggregation performed entirely by the system


Application Layer
Application Specific Aggregate Functions
Application Specific Aggregate Functions
System Layer
Diffusion
Diffusion

Application specific aggregate functions

Aggr.Type specific diffusion algorithms provided by system



Application specific aggregate functions

(Portions of) diffusion algorithms application specific

Middleware Issues

Must support a wide range of applications

- appropriate abstractions & functions
- different algorithms for the same function, choice depends on
  - type of application
  - QoS requirements
  - system parameters (node density, mobility, resource availability, ...)

Must run on resource-limited devices

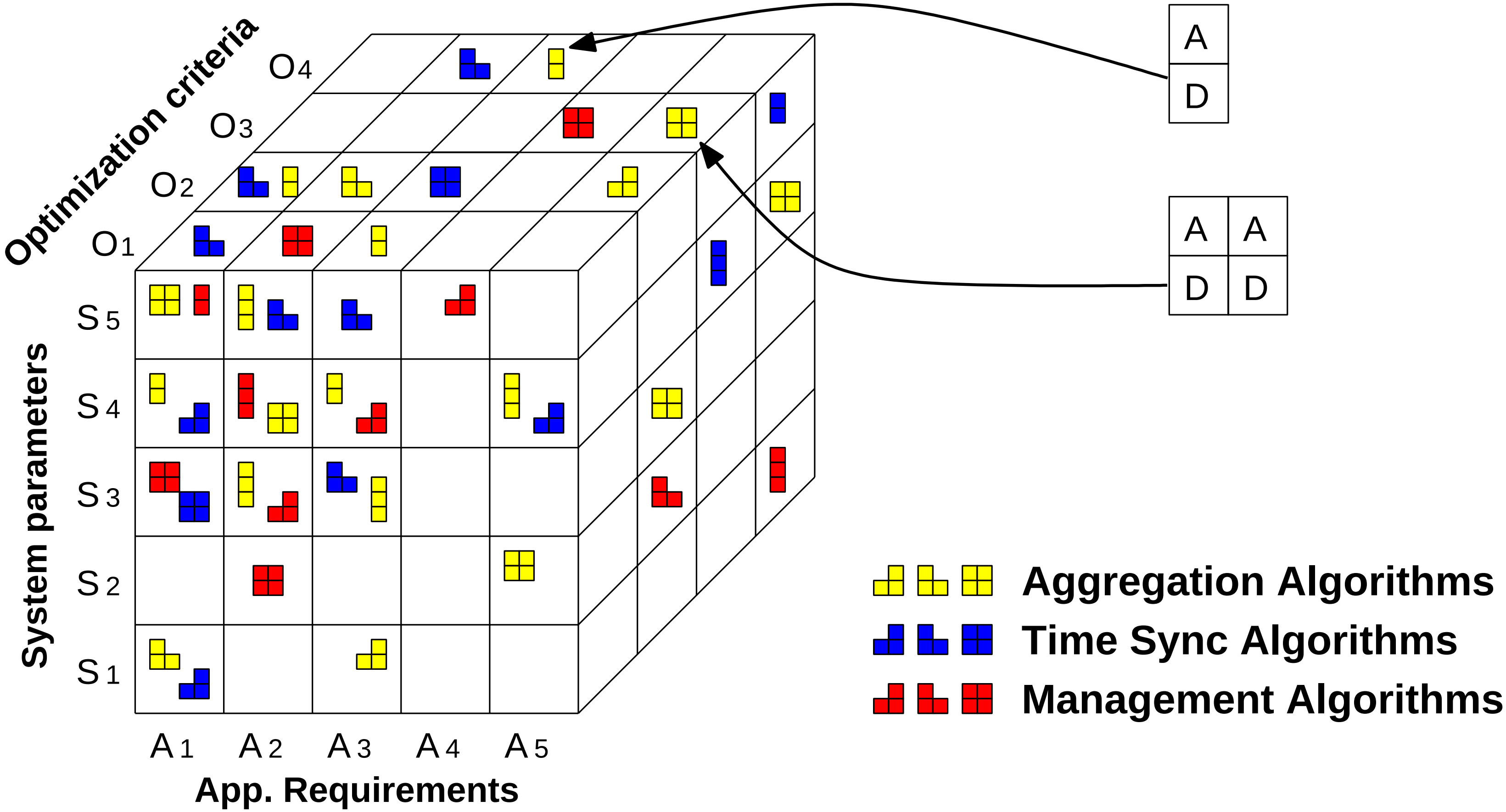
⇒Should provide a minimal core functionality

⇒Other functions should be dynamically configured

⇒Appropriate middleware architecture

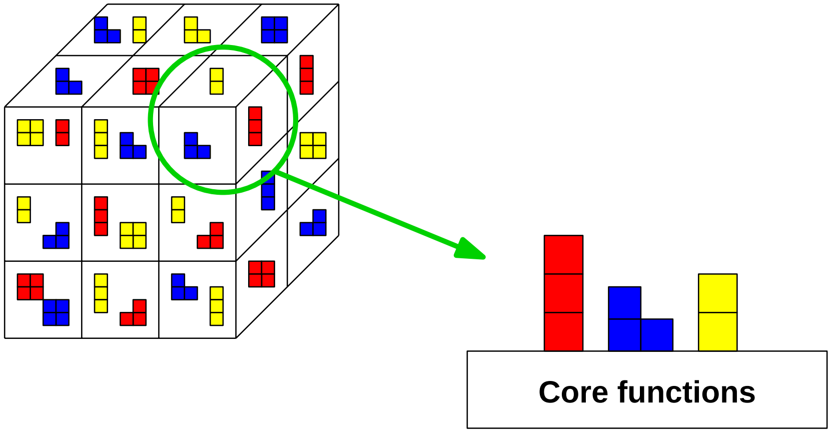
- dynamically configurable / extensible
- concepts for cross-layer design

Middleware Architecture (1)



Middleware Architecture (2)

- Building blocks selected depending on application, system model, required QoS
  - before deployment
  - after deployment (dynamic reconfiguration)
- Based on generic Core Functions, e.g.,
  - Scheduling of block execution
  - Communication
  - Configuration
  - ???

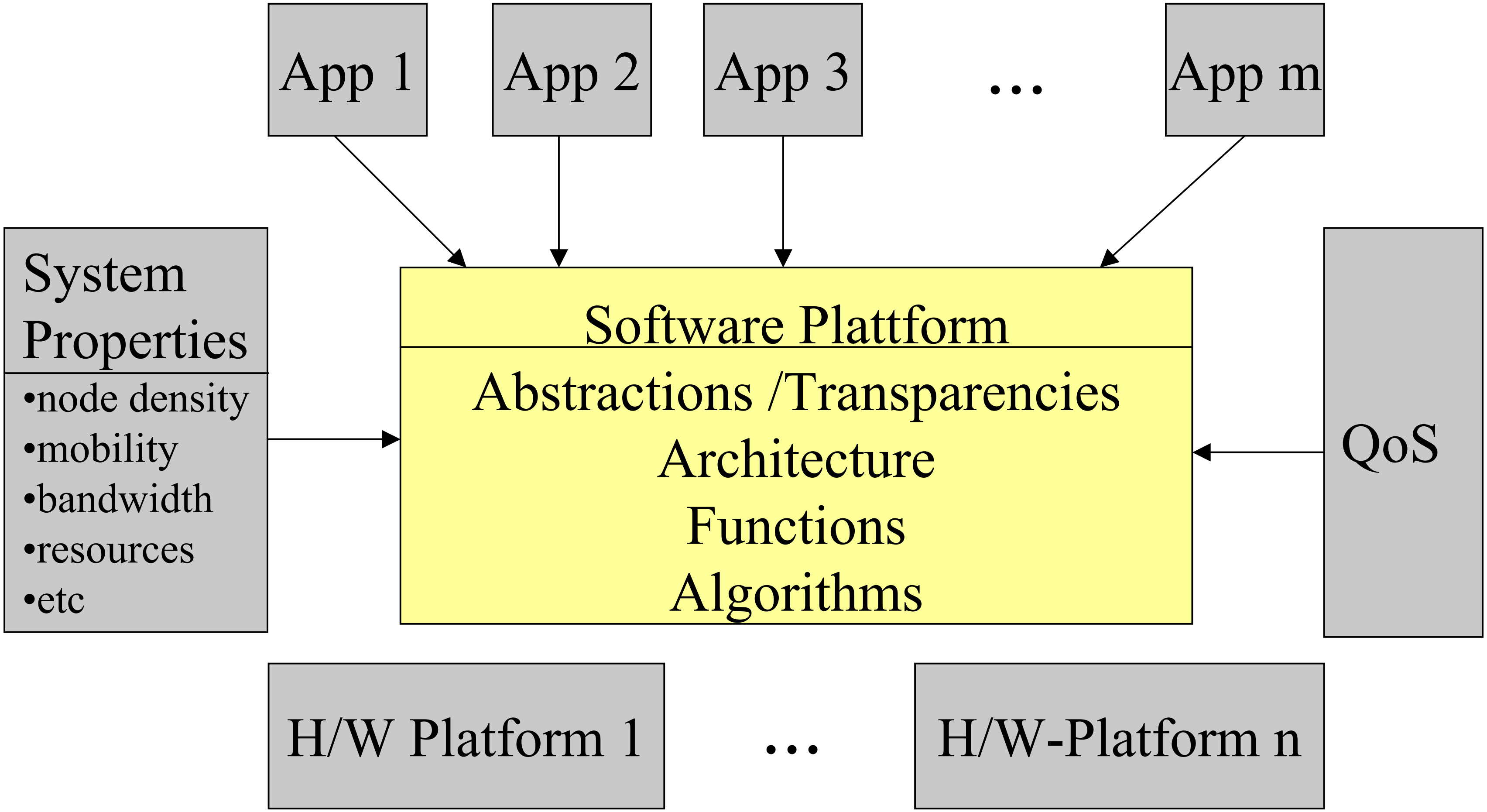


Fundmental Questions (just a few)

- What is the appropriate set of programming abstractions and QoS concepts?
- What level(s) of distribution transparency needed?
- What are appropriate QoS concepts
- What functions should be supported by a platform?
- How does an appropriate architecture look like?
- What kind of algorithm in which setting?
- What kind of adaptation needed?
- and more ...

Discussion “Platforms and Testbeds”

Software Platforms: Requirements



App 1

App 2

App 3

…

App m

System Properties

- •node density
- •mobility
- •bandwidth
- •resources
- •etc

Software Plattform

Abstractions /Transparencies

Architecture

Functions

Algorithms

QoS

H/W Platform 1

…

H/W-Platform n

Software Platform Needed ?

YES! (more than one?)

- wide spectrum of apps
- multiple H/W platforms

European platform is important

- key technology and important market
- would bring together

  researchers, industry, regulators, user groups

  to devise and implement a common strategy for

  the development, deployment and use of sensor networks in Europe

  (See: Investing in research: an action plan for Europe)

Fundamental Research Issues (just a few)

- What is the appropriate set of programming abstractions?
  - SQL *versus* Messages/Events
- What level(s) of distribution transparency needed?
  - full transparency *versus* application knows “some aspects” of topology (node addresses, location, distance, neighborhood, ???)
- What are appropriate QoS concepts?
  - QoS concepts from client/server, multimedia, ..., domain *versus* new (application specific) concepts
- What functions should be supported by a platform?
  - subset *versus* extended subset of Corba, .NET, ...

Fundamental Research Issues (just a few)

- How does an appropriate architecture look like?
  - layered architecture
  - *but* what about cross-layer issues, (self-)configuration, adaptation
- What kind of algorithm in which setting?
  - scalability, resource consumption, reliability, ...
  - dependencies between algorithms
- What kind of adaptation needed?
  - adaptation goals (local, global)
  - system monitoring
  - adaptation algorithms