Seminar in Networked Embedded Systems

Summer 2024
https://nes-lab.org/nes-seminar-summer2024/

Prof. Dr. Marco Zimmerling

NES | NETWORKED EMBEDDED SYSTEMS LAB

TECHNISCHE UNIVERSITÄT DARMSTADT

About me: Marco Zimmerling

- Since 04/2023: Full Professor at TU Darmstadt
- Previously:
  - Studies at TU Dresden, Uppsala University, and Swedish Institute of Computer Science
  - Internship at IBM Research, Hawthorne, NY, USA
  - PhD at ETH Zurich, Switzerland (2009-2015)
  - Research group leader at TU Dresden (2015-2022)
  - Full Professor at University of Freiburg (2022-2023)

About the Networked Embedded Systems Lab




Max

- Research group established in 2015
- Focus: Methods and tools for building *dependable*, *efficient*, and *sustainable* cyber-physical and Internet of Things systems
  - Low-power wireless protocol design and analysis
  - Battery-free, energy-harvesting embedded systems
  - Wireless control systems, design for predictability
- Thesis and job opportunities (see https://nes-lab.org/)

Plan for today

1. Teaching goals

2. Organization

3. Available topics

Plan for today

1. Teaching goals

2. Organization

3. Available topics

Why attend this seminar?

- Learn fundamentals of doing research
  - Read and understand papers
  - Summarize and present complex concepts
  - Give constructive feedback
- Learn about cutting-edge research in cyber-physical systems, wireless sensor networks, and the Internet of Things
  - Maybe your future thesis or HiWi topic
- Learn LaTeX and how to design a poster

Plan for today

1. Teaching goals

2. Organization

3. Available topics

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster

  Typical format of international scientific conferences
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

ACM SenSys 2023

November 13-15, 2023, Istanbul, Turkiye

https://sensys.acm.org/2023/

Welcome to ACM SenSys 2023

The 21th ACM Conference on Embedded Networked Sensor Systems (SenSys 2023) introduces a highly selective, single-track forum for research on systems issues of sensors and sensor-enabled smart systems, broadly defined. Systems of smart sensors will revolutionize a wide array of application areas by providing an unprecedented density and fidelity of instrumentation. They also present various systems challenges because of resource constraints, uncertainty, irregularity, mobility, and scale. This conference provides an ideal venue to address research challenges facing the design, development, deployment, use, and fundamental limits of these systems. Sensing systems require contributions from many fields, from wireless communication and networking, embedded systems and hardware, energy harvesting and management, distributed systems and algorithms, data management, and applications, so we welcome cross-disciplinary work.

From submission to conference

- Prepare and submit manuscript (i.e., a PDF file) following the instructions in the Call for Papers
- Technical Program Committee reviews all manuscripts that comply with the requirements
  - Typically 3-5 reviews per manuscript
- 15-20% of submissions are accepted to appear in the proceedings and to be presented at the conference
  - 35 accepted out of 179 submissions at ACM SenSys 2023

submission: 06/2023 → notification: 09/2023 → conference: 11/2023

Talks (10–20 min) + Q&A (5 min)

"Our"
Carsten

Poster and demo session (2–3 hours)

Posters

Demos

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

Advice today
Help from mentor

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

How does a paper typically look like?

- 10-14 pages in total, font size 9 or 10, double column for conferences papers and single column for journal articles
- Mostly text, but also (illustrative) figures and tables
  - “A picture is worth a thousand words”
- Basic structure:
  - Beginning: Abstract, introduction
  - Middle: Up to the authors
  - End: Conclusions, acknowledgements, bibliography

Advice on reading a paper

- “How to read a paper” by S. Keshav (2 pages) http://ccr.sigcomm.org/online/files/p83-keshavA.pdf

- “How to read a research paper” by M. Mitzenmacher and N. Ramsey (2 pages) https://ccc.inaoep.mx/~esucar/Clases-semdr/Lecturas/ramsey00.pdf

Basic approach: Three passes

- Skim abstract, conclusions, section titles, figures (10 min)
  - What is the general problem area? Is it interesting for me?

Basic approach: Three passes

- Skim abstract, conclusions, section titles, figures (10 min)
  - What is the general problem area? Is it interesting for me?
- Quickly read rest of the paper (1 hour, at most 2 hours)
  - What are the paper's contributions (i.e., basic idea of proposed method, how is it analyzed/evaluated, how well does it work, …)
  - Don't be afraid to skip certain parts (e.g., detailed descriptions) and note things you have not yet fully understood

Basic approach: Three passes

- Skim abstract, conclusions, section titles, figures (10 min)
  - What is the general problem area? Is it interesting for me?
- Quickly read rest of the paper (1 hour, at most 2 hours)
  - What are the paper's contributions (i.e., basic idea of proposed method, how is it analyzed/evaluated, how well does it work, …)
  - Don't be afraid to skip certain parts (e.g., detailed descriptions) and note things you have not yet fully understood
- Carefully read entire paper again (3-6 hours or even more)
  - How does it really work? Need to consult other relevant literature?

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

Poster Abstract: Exploiting Protocol Models for Generating Feasible Communication Stack Configurations

Marco Zimmerling, Federico Ferrari, Matthias Woehrle, and Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{zimmerling,ferrari,woehrle,thiele}@tik.ee.ethz.ch

ABSTRACT

Communication stacks are composed of distinct layers that, in principle, operate independently and interact through well-defined interfaces. However, resource constraints in sensor networks typically necessitate optimizations, leading to implicit assumptions and dependencies among layers (*e.g.*, a collection protocol assumes the MAC protocol provides sufficient bandwidth). These dependencies are often tracked manually, yet become extremely complex as protocols evolve and requirements change. We propose to model assumptions and dependencies explicitly, as constraints on protocol parameters. This allows for using standard tools to generate feasible protocol configurations. We demonstrate the effectiveness of our approach using the example of FTSP running on top of a low-power listening MAC protocol.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: *Modeling techniques*; D.2.1 [**Software Engineering**]: Requirements/Specifications—*Methodologies*

General Terms

Performance, Reliability

Keywords

Protocol Configuration, Constraint Programming

1. INTRODUCTION

Keeping track of the various interactions and dependencies between the different layers of a communication stack is a critical yet inherently complex task. Choi *et al.* [1] tackle the problem of interprotocol interference on the network layer and propose an isolation layer to avoid collisions and guarantee protocol fairness. Indeed, unintended interactions are even more likely between protocols on different layers: protocols frequently make assumptions on the behavior of other protocols to meet the application requirements [6].

We propose to explicitly model assumptions of protocols to expose hidden dependencies between individual layers. While creating such a model is a considerable initial overhead, it frees the designer from managing the various dependencies manually. Moreover, it allows for automatically generating feasible and consistent configurations of the complete stack for *i)* different protocol combinations, *ii)* different deployment settings, *iii)* different protocol parametrizations, and *iv)* when requirements change.

IPSN'10, April 12–16, 2010, Stockholm, Sweden.
ACM 978-1-60558-988-6/10/04.

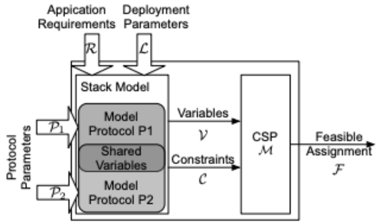


Figure 1: Overview of the approach.

2. APPROACH

Figure 1 outlines our approach. The stack model comprises different protocol models and their individual parameters $\mathcal{P}_i$ (*e.g.*, the sleep interval of the MAC protocol). Shared variables appearing in more than one protocol model signal dependencies and interactions among protocols. In addition, the stack model takes as inputs deployment parameters $\mathcal{L}$ (*e.g.*, the number of nodes) and a set of application requirements $\mathcal{R}$ (*e.g.*, the required data yield). From this model, we extract a set of constraints $\mathcal{C}$ on variables $\mathcal{V}$ for which we seek to find feasible assignments $\mathcal{F}$, the protocol configurations.

In particular, we can formulate a constraint satisfaction problem (CSP) $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{C} \rangle$, where $\mathcal{V}$ is an n-tuple of decision variables $\mathcal{V} = \langle v_1, v_2, \ldots, v_n \rangle$, $\mathcal{D}$ is a corresponding n-tuple of domains $\mathcal{D} = \langle D_1, D_2, \ldots, D_n \rangle$ such that $v_i \in D_i$, and $\mathcal{C}$ is a k-tuple of constraints $\mathcal{C} = \langle C_1, C_2, \ldots, C_k \rangle$ [3]. A constraint C_i is a pair $\langle R_i, S_i \rangle$, where S_i is a k_i-tuple of variables in $\mathcal{V}$ and R_i is a relation of arity k_i over the domains of the variables in S_i. Solving the CSP $\mathcal{M}$ involves finding a value for each variable in $\mathcal{V}$, where the constraints $\mathcal{C}$ specify that some subsets of their domains $\mathcal{D}$ cannot be used together. We can use standard CSP solvers for this task, which combine interference (constraint propagation) and search algorithms. If we are interested in an optimal protocol configuration, $\mathcal{M}$ can be transformed into a constraint optimization problem (COP) by adding an objective function.

Our approach is modular and allows for fast iterations when application requirements or deployment characteristics change. Since every protocol parameter can also be specified as a constraint, we can analyze the effect of deviating conditions. Most importantly, the stack model and thus the generated protocol configurations capture all protocol dependencies, which reduces the risk of unintended interactions at runtime.

Poster Abstract: All-to-all Communication in Multi-hop Wireless Networks with Mixer

Fabian Mager*, Johannes Neumann*, Carsten Herrmann†, Marco Zimmerling*, Frank Fitzek†
* Networked Embedded Systems Group, TU Dresden, Germany
†Deutsche Telekom Chair of Communication Networks, TU Dresden, Germany
firstname.lastname@tu-dresden.de johannes.neumann@mailbox.tu-dresden.de

ABSTRACT

Cyber-physical systems (CPS) use distributed feedback loops to control physical processes. Designing practical distributed CPS controllers often benefits from a logically centralized approach, where each node computes the control law locally based on global knowledge of the system state. We present Mixer, an all-to-all communication scheme that enables all nodes in a multi-hop low-power wireless network to exchange sizable packets with one another. Mixer's design integrates synchronous transmissions with random linear network coding, harnessing the broadcast nature of the wireless medium. Results from testbed experiments with an early Mixer prototype show that our design reduces latency by 1.1–2.6× for 16–96-byte packets compared with the state of the art, while providing a reliability above 99.9 % in most settings we test.

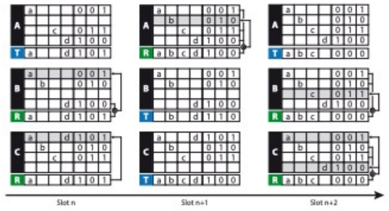


Figure 1: Example execution of Mixer in a 2-hop network with 4 nodes *A*, *B*, *C*, and *D*. *Using network coding, Mixer exchanges four different payloads within three slots—without network coding, at least one more slot would be needed.*

1. OVERVIEW

Wireless cyber-physical systems (CPS) bring unprecedented opportunities by integrating sensing, control, and actuation into feedback loops for controlling physical processes. Computing the control law locally at each node rather than at a centralized controller offers better scalability and fault tolerance. To ease the design of practical distributed CPS controllers, it is often beneficial if one can assume that all nodes have global knowledge of the current system state [5]. To this end, a communication scheme is needed that supports exchanging sizable packets among all nodes in a multi-hop wireless network in a fast, reliable, and efficient manner.

State-of-the-art solutions, such as Chaos [2], partially meet these requirements. However, Chaos was primarily designed for data aggregation, and scales sub-optimally when nodes need to exchange larger chunks of raw data, called *payloads.*

Mixer in a nutshell. We present Mixer, an all-to-all communication scheme for multi-hop wireless networks. By integrating synchronous transmissions with random linear network coding [1], Mixer nodes combine several payloads before relaying. To exploit the former, we built Mixer on top of Chaos [2], but using the latter, Mixer needs fewer transmissions than Chaos to convey the same amount of information, significantly reducing latency especially for larger payloads.

SenSys '16 November 14-16, 2016, Stanford, CA, USA

ACM ISBN 978-1-4503-4263-6/16/11...$15.00
DOI: http://dx.doi.org/10.1145/2994551.2996706

Network coding. The basic idea of network coding is to apply coding operations on incoming payloads at every node in the network. In Mixer, a coding operation is a linear combination of received payloads. This translates into a system of linear equations, which is maintained by every node.

$$\underbrace{\overset{E}{\begin{pmatrix} e_{11} & \ldots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{n1} & \ldots & e_{nn} \end{pmatrix}}}_{\text{Encoding Matrix}} * \underbrace{\overset{p}{\begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}}}_{\text{Payloads}} = \underbrace{\overset{c}{\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}}}_{\text{Encoded Payloads}} \quad (1)$$

Each row in the *encoding matrix E* represents an *encoding vector e*, determining which payloads are combined to create an *encoded payload.* The elements of *E* must be from a finite field, in our case GF(2), so encoding means to XOR several payloads. But how should a node choose *e*? One way, known as random linear network coding [1], is to choose *e* randomly each time before relaying. This approach is advantageous as it does not require the nodes to know the network topology.

After choosing a random encoding vector *e*, a node computes an encoded payload and puts it together with *e* into a packet. According to (1), nodes receiving the packet store the encoded payload in vector *c* and *e* in matrix *E* if this increases the rank of *E*. The system of linear equations has a single unique solution when *E* reaches full rank; that is, a

Abstract: Format and rules

- Use LaTeX template available on course website
- In English (British or American, not a mix of both)
- 2 pages + additional page for references

- Summarize the paper in your own words
  - No copy & paste from the paper, except for figures (e.g., results)
  - Must be understandable without consulting the original paper

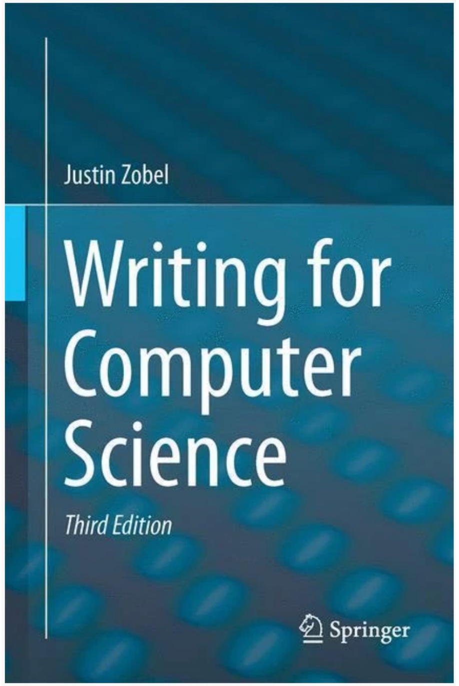
Abstract: Specific advice

- Content
  - Imagine you want to persuade someone to read the full paper
  - Focus on the core problem, key idea, main result, etc.

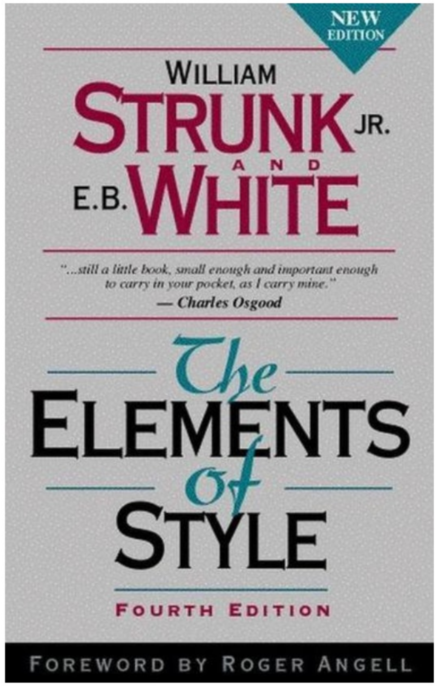
- Presentation
  - Write in a neutral way (“the authors” or ”the experiments”, not “we”)
  - Bad English distracts from good content
  - Revise, revise, revise

Abstract: General advice on writing

- ”Writing for computer science” by J. Zobel



- ”The elements of style” by W. Strunk and E.B. White



Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

Talk: Format and rules

- Present a recent research paper published at a top venue
- Slides and speech in English
- 15 minutes + questions

- Prepare your own slides
  - No copy & paste from existing slides (e.g., authors' slide deck)
  - You may use results (e.g., plots) and examples from the paper

Talk: Specific advice

- Structure
  - Motivation: What is the problem? Why is it important? *high-level*
  - Contribution: What is the main novel idea? *some details*
  - Conclusion: 1-slide summary *high-level*
- Presentation
  - Examples are your secret weapon
  - Stick to the time limit
  - Practice, practice, practice

Talk: General advice

- “How to give a great research talk” by S.P. Jones
  https://www.microsoft.com/en-us/research/academic-program/give-great-research-talk/

- “Creating effective slides” by J. Doumont *highly recommended*
  http://youtu.be/meBXuTIPJQk

Your tasks

- Read a research paper
- Present the paper to your peers
  - Write a 2-page abstract
  - Give a 15-minute talk
  - Prepare and present a poster
- Active participation
  - Try to attend all sessions
  - Ask critical questions, provide constructive feedback, etc.

Poster: Format and rules

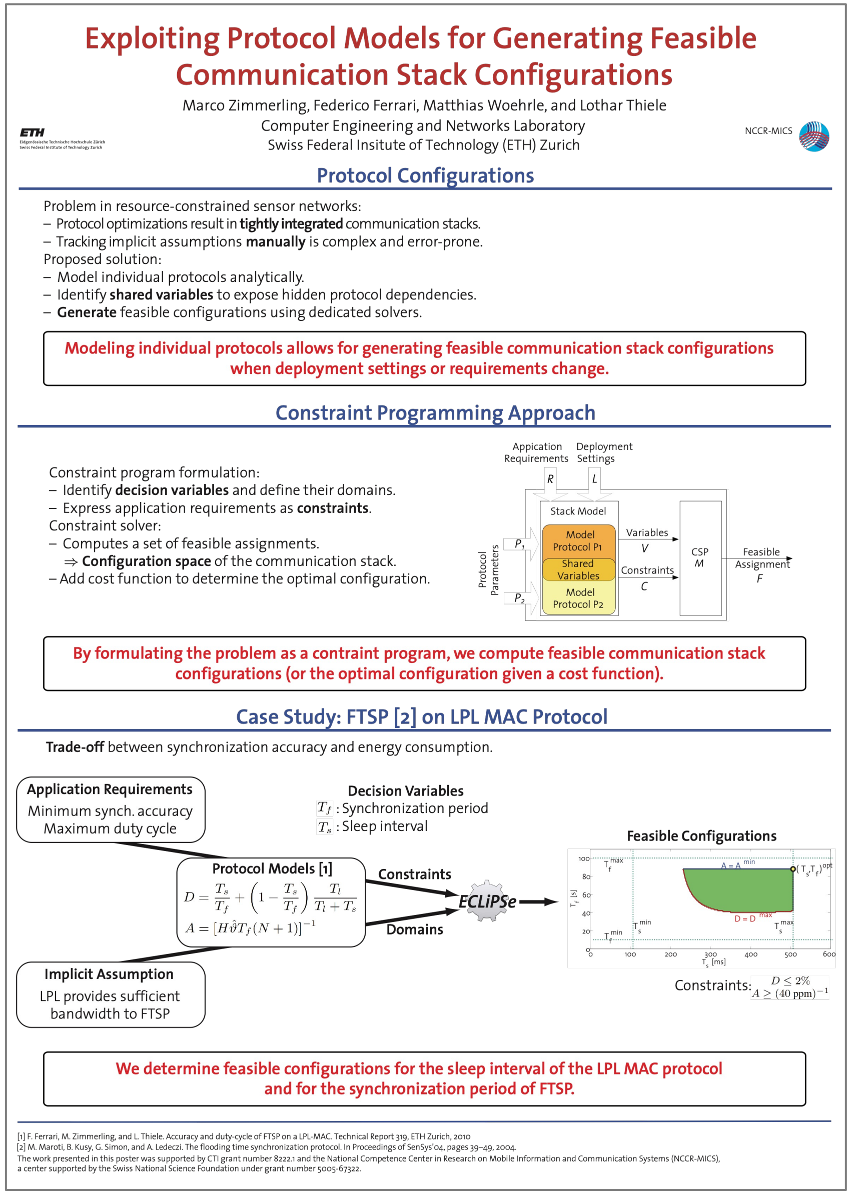
- May use PowerPoint template available on course website
  - Or Adobe Illustrator, LaTeX `beamerposter`, etc.
- In English (British or American, not a mix of both)
- A0 format with sufficient margins (i.e., about 3cm)

- Summarize the paper in your own words
  - No copy & paste from the paper, except for figures (e.g., results)

Poster: Specific advice

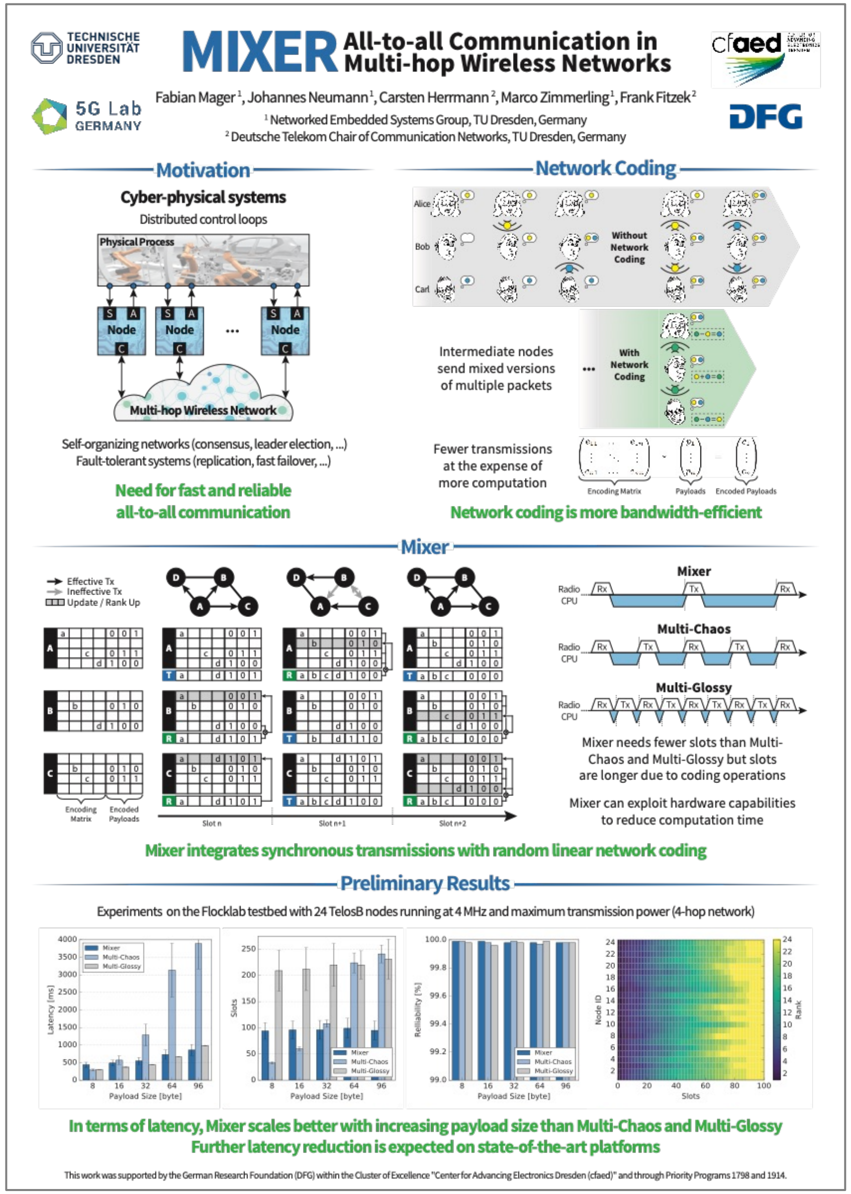
- Content
  - Imagine using a graphical illustration or data plot to explain a complex concept in words to a visitor in 2 minutes
  - Focus on the core problem, key idea, main result, etc.

- Presentation
  - Prefer (simple) figures and examples over (long) text
  - Still enough text to catch people's interest (no typos allowed)
  - Revise, revise, revise



Exploiting Protocol Models for Generating Feasible Communication Stack Configurations
Marco Zimmerling, Federico Ferrari, Matthias Woehrle, and Lothar Thiele
Computer Engineering and Networks Laboratory
Swiss Federal Institute of Technology (ETH) Zurich
NCCR-MICS
Protocol Configurations
Problem in resource-constrained sensor networks:
– Protocol optimizations result in tightly integrated communication stacks.
– Tracking implicit assumptions manually is complex and error-prone.
Proposed solution:
– Model individual protocols analytically.
– Identify shared variables to expose hidden protocol dependencies.
– Generate feasible configurations using dedicated solvers.
Modeling individual protocols allows for generating feasible communication stack configurations when deployment settings or requirements change.
Constraint Programming Approach
Constraint program formulation:
– Identify decision variables and define their domains.
– Express application requirements as constraints.
Constraint solver:
– Computes a set of feasible assignments.
⇒ Configuration space of the communication stack.
– Add cost function to determine the optimal configuration.
By formulating the problem as a contraint program, we compute feasible communication stack configurations (or the optimal configuration given a cost function).
Case Study: FTSP [2] on LPL MAC Protocol
Trade-off between synchronization accuracy and energy consumption.
Application Requirements
Minimum synch. accuracy
Maximum duty cycle
Decision Variables
Synchronization period
Sleep interval
Protocol Models [1]
Constraints
Domains
ECLiPSe
Feasible Configurations
Implicit Assumption
LPL provides sufficient bandwidth to FTSP
We determine feasible configurations for the sleep interval of the LPL MAC protocol and for the synchronization period of FTSP.

mmh



TECHNISCHE UNIVERSITÄT DRESDEN
MIXER All-to-all Communication in Multi-hop Wireless Networks
5G Lab GERMANY
DFG
Fabian Mager, Johannes Neumann, Carsten Herrmann, Marco Zimmerling, Frank Fitzek
Motivation
Cyber-physical systems
Distributed control loops
Need for fast and reliable all-to-all communication
Network Coding
Network coding is more bandwidth-efficient
Mixer
Mixer integrates synchronous transmissions with random linear network coding
Preliminary Results
In terms of latency, Mixer scales better with increasing payload size than Multi-Chaos and Multi-Glossy
Further latency reduction is expected on state-of-the-art platforms

better

Schedule

- April 23: Send paper preferences (1st/2nd/3rd choice) via email to marco.zimmerling@tu-darmstadt.de
- April 24: Papers assigned and talks (tentatively) scheduled
- June 11: Submit final version of your abstract and poster as 2 separate PDF files via email to your mentor
- Starting June 18: Talks in S4|14, room 2.1.01
- End of summer term: Poster session

Interaction with mentor

- Each student will be assigned a mentor
  - Communication with mentor primarily via email
- Your mentor will
  - Help you clarify technical questions about the research paper
  - Provide feedback on your artifacts (i.e., abstract, poster, slides)
- Rules:
  - Send artifact at least 7 days in advance
  - Feedback is provided at most once for each artifact

Grading

- Abstract: 30%
- Talk: 30%
- Poster and poster presentation: 30%
- Active participation: 10%

Plan for today

1. Teaching goals

2. Organization

3. Available topics

Example applications

precision agriculture

smart cities

disaster mitigation

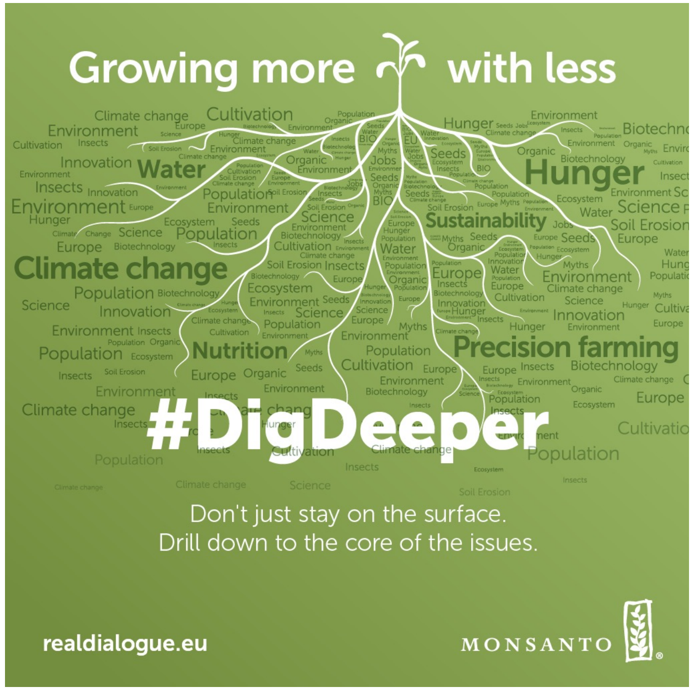
Example applications

precision
agriculture

smart
cities

disaster
mitigation

Precision agriculture



- Current agricultural practices waste water, pesticides, etc. causing environmental impact and economic losses
- Use distributed sensing, decision making, and actuation to reduce impact and losses while increasing crop yield

Example applications

precision
agriculture

smart
cities

disaster
mitigation

Smart cities

- Smart city applications include intelligent transportation, power grid, waste water management, home automation
- Use distributed sensing, decision making, and actuation to, e.g., reduce air pollution and improve quality of life

Example applications

precision
agriculture

smart
cities

disaster
mitigation

Disaster mitigation

Disaster mitigation

- Understand, e.g., thawing permafrost in high-alpine regions to predict impeding rockfall events
- Use distributed sensing, decision making, and actuation to mitigate harm to humans and critical infrastructure

Cyber-physical systems

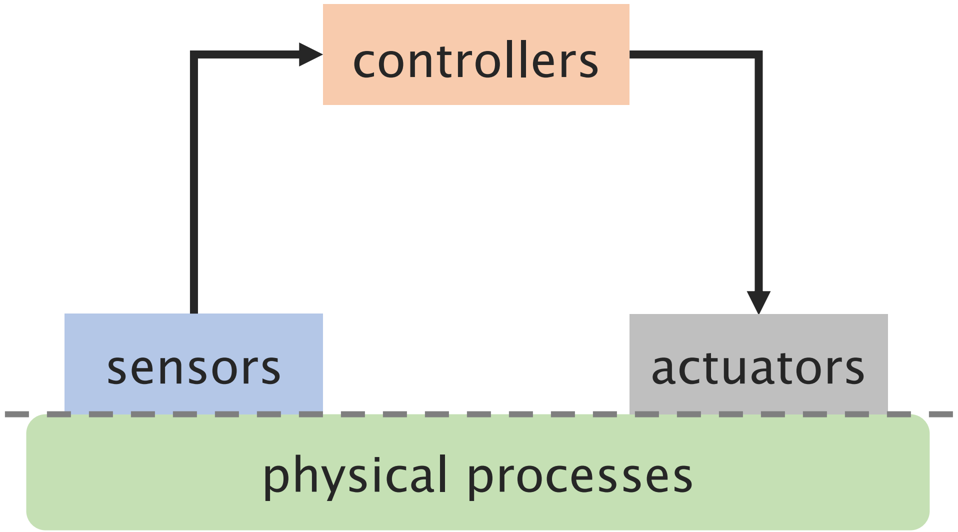
physical world
governed by nature

physical processes

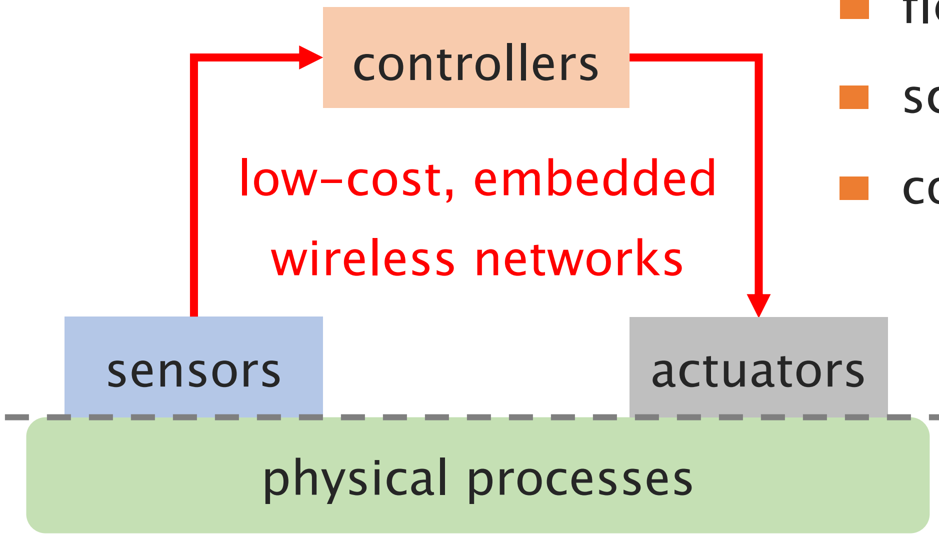
Cyber-physical systems

cyber world
governed by hardware + software

physical world
governed by nature



Cyber-physical systems

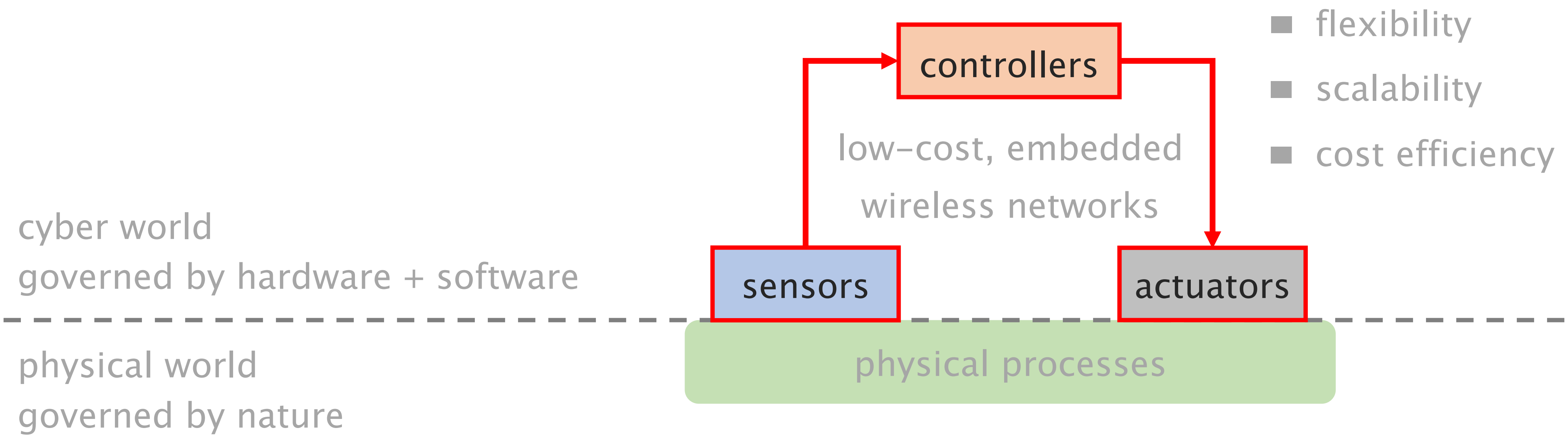


- flexibility
- scalability
- cost efficiency

cyber world
governed by hardware + software

physical world
governed by nature

Networked embedded systems (NES)



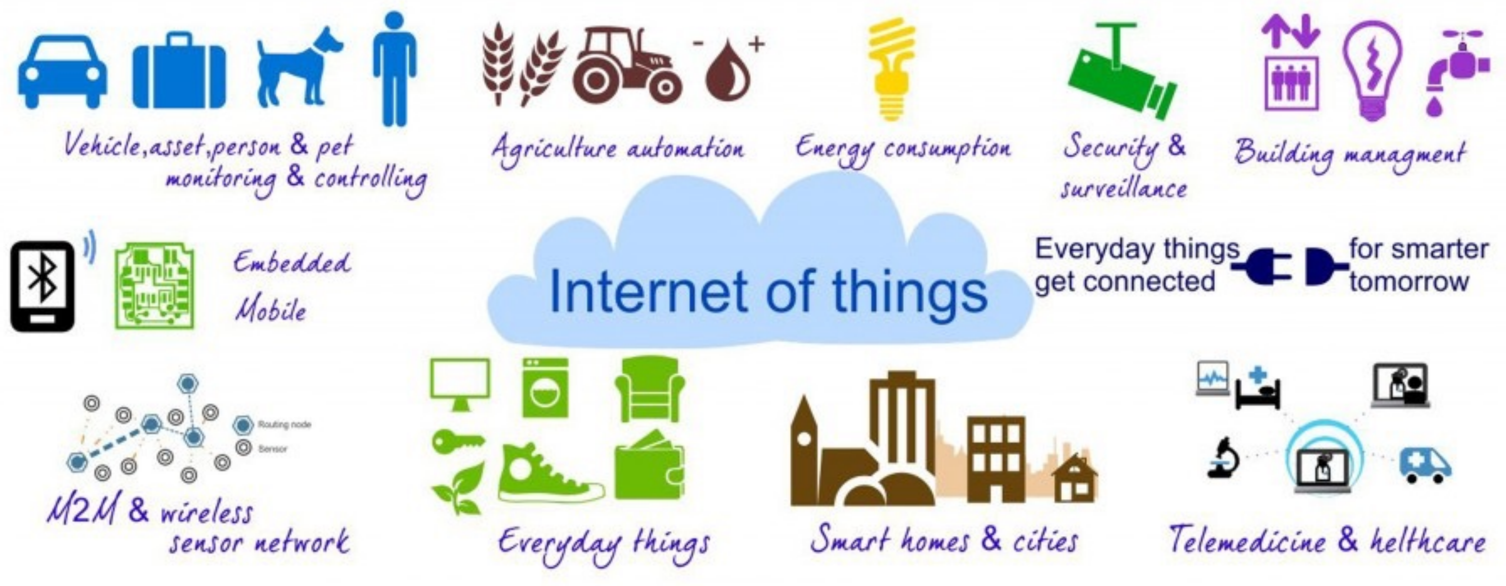
Hardware and software components for sensing, communication, computation, and actuation that enable cyber-physical systems

Traditional NES

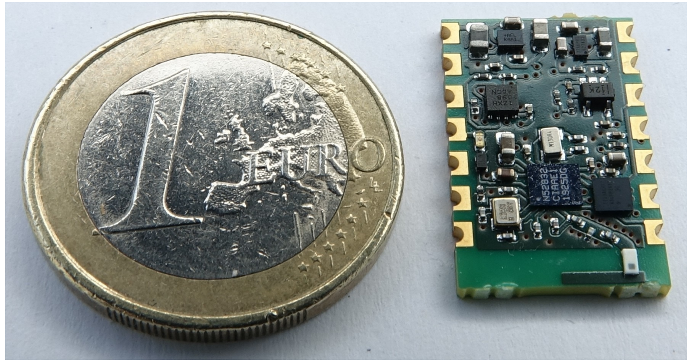
- Anti-lock braking system
- Electronic stability control
- Airbag
- Automatic gearbox
- Smart keys

- Flight control system
- Anti-collision control
- Pilot information system
- Flap control system
- Entertainment system

Emerging NES



Everywhere Wireless



- Trends: Battery-free devices, embedded machine learning, low-power wireless communication, edge computing, etc.

Topics to choose from

- >20 papers published in top international venues: https://nes-lab.org/nes-seminar-summer2024/
  - Battery-free and energy-harvesting systems
  - Embedded machine learning
  - Wireless communication and networking
  - Localization
  - Sensing and edge computing
  - Emerging applications (e.g., space)

Topics to choose from

- Submit your preferences by April 23, 11:59pm
  - You pick three papers (1st/2nd/3rd choice)
  - By email to marco.zimmerling@tu-darmstadt.de
  - Please use the paper numbers listed on the course website
  - We assign one paper and a mentor to each of you