Realtime Publish/Subscribe for Cyber-Physical Systems
KSWS AVA / Projekt AVA / NEidl VHR

Dr.-Ing. Peter Danielis
Verteiltes Hochleistungsrechnen (VHR)

Dr.-Ing. Helge Parzyjegla
Architektur von Anwendungssystemen (AVA)

M.Sc. Eike Björn Schweißguth
Institut für Angewandte Mikroelektronik und Datentechnik (IMD)
What is Realtime (Echtzeit)?

Not necessarily fast, but **predictable**!
→ Do the **right thing** at the **right time**.
What is Publish/Subscribe?

Publish/Subscribe Notification Service

Publisher produce data and offer them to everyone interested

publish\( (n) \)

Subscribe subscribe those data, in which they are interested

subscribe\( (F_n) \)
Scalable $m:n$-group communication

publish($n$)

subscribe($F_n$)
What are Cyber-Physical Systems?

- Systems containing software components and mechanical or electronic parts that are interconnected via network
- Interact with the real, physical world
  - are subject to physical laws
  - have requirements w.r.t (real) time
- Examples
  - Industry robots
    - Production line in the smart factory
    - Reconfigurable production cell of a smart factory
  - Modern (autonomous) vehicles
    - Steer/fly by wire
    - Autopilots of any kind
Industry Robots in a Smart Factory

Time-critical communication when handing over work pieces.
Reconfigurable Production Cell

Flexible communication in case of task changes.

Industry robots made by Kuka
Communication Schedule

Streams
- From node 6 to nodes 7 and 10 (multicast)
- From node 10 to node 7

Schedule
- Determines exactly when which packet is sent over which link
- Has to be always without conflicts → provable correct
- Needs to be adapted whenever communication pattern changes
- Additional traffic of lesser importance is possible
Projects and Collaborations

> Realtime publish/subscribe communication
  > Part of a DFG project (VHR, AVA und IMD)
  > Planning of flexible communication patterns and reservation of required time slots on communication links
  > Formal models and methods for scheduling
  > Estimation of the worst case runtime for publishing and filtering (content-based if necessary) a notification
  > Application scenario within a smart factory

> Autonomous Underwater Vehicles (AUVs)
  > Cooperation with the Institute for the Protection of Maritime Infrastructures, Resilience Department of Maritime Systems, German Aerospace Center (DLR) Bremerhaven
Tasks: Realtime Publish/Subscribe

> Data models and filter models
  > Design of suitable data models
  > Design of corresponding filter models
  > Prototypical implementation (probably) in C

> Realtime properties
  > Estimation of Worst Case Execution Time (WCET)
  > Scalability analysis (#Filter, #Daten, #Subscriber)
  > Test/measurement of implementation

> Development and test platform for prototyping
  > Scripts for configuring TSN switches
  > Traffic generators for test traffic
  > Management platform/tools of any kind
Tasks: Realtime Communication Schedule

> Integer Linear Programming (ILP) model
  > Familiarize with formal modeling of optimization problems
  > Familiarize with programming an ILP solver (→ Python)

> Porting existing ILP model from Gurobi to CPLEX and PuLP
  > Document the steps for porting models
  > Evaluation of model complexity and solver runtime

> ILP models for flow migration
  > Development of new models to reschedule a part of an existing schedule
  > Migration of flows/streams to different time slots and/or different paths
Optional: Autonomous Underwater Vehicles

- Cooperative navigation of several AUVs
  - Battery capacity limits motion and sensors
  - Opportunistic communication via acoustic modems

- Limited number of advanced tasks available
  - Implementations using Simulator OMNeT++ and C++
  - Improving motion models and 3D visualization
  - Simulation of sensors (energy consumption, coupling with other systems)
  - Modelling/simulation of underwater media and communication
Organizational Matters

> Up to three teams
  > Team A: Realtime publish/subscribe
  > Team B: Realtime Communication Schedule
  > Optional team C: Autonomous Underwater Vehicles (AUVs)

> Design methodology
  > Agile development
  > Three milestones w.r.t. design, implementation, documentation

Type and sizeSCALE of tasks depends on number
and interests of participants!
Registration and Contact

> Enrollement in respective Stud.IP course

1. 23846 (Vorlesung) KSWS: AVA

2. 23848 (Vorlesung) Neueste Entwicklungen der Informatik (Verteiltes Hochleistungsrechnen

3. 23847 (Projekt) Projekt: AVA

> Questions via email to Peter Danielis and Helge Parzyjegla

> peter.danielis@uni-rostock.de

> helge.parzyjegla@uni-rostock.de