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

(subscribe\(F_n\))

Subscriber subscribe those data, in which they are interested
Scalable $m:n$-group communication

$subscribe(F_n)$

$publish(n)$
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.

Industry robots made by Kuka
Reconfigurable Production Cell

Industry robots made by Kuka

Flexible communication in case of task changes.
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
Underwater-Scenario 1: Maintenance of the foundations of offshore wind turbines
Underwater-Scenario 2:
Clearance of Unexploded Ordnance (UXO) from World War II
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
  > Cooperative navigation of several AUVs
  > Limited Energy restricts movement and usage of sensors
  > Opportunistic communication via acoustic modems
Tasks: Realtime Publish/Subscribe

> Simulation models for realtime communication (TSN standards)
  > TSN configuration (IEEE 802.1Qcc)
  > Time synchronization (IEEE 802.1AS)
  > Controlled timing (IEEE 802.1Qch)
  > Reliable communication (IEEE 802.1Qca, IEEE 802.1Qci)

> TSN controller (CUC and CNC)
  > Implementation of TSN configuration option (IEEE 802.1Qcc)
  > Based on Ryu framework for SDN controller
  > Integration of a trivial planning component

> Development and test platform for prototypes
  > Scripts for configuring TSN switches
  > Generators for test data
  > Management tools for different purposes
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
Tasks: Autonomous Underwater Vehicles

- Cooperative navigation of several AUVs
  - Implementation of motion models
  - Implementation of localization algorithms
- Energy consumption caused by motion and activated sensors
  - Implementation of models for energy consumption for movement and activated sensors
  - Implementation of models for energy consumption for image processing algorithms
- Opportunistic communication via acoustic modems
  - Implementation of realistic underwater communication
- Implementations using Simulator OMNeT++ and C++
  - Python for scripting and evaluation of simulation results
Organizational Matters

> Up to three teams
  > Team A: Realtime publish/subscribe
    (probably more fine-grained distribution of tasks)
  > Team B: Realtime communication schedule (ILP)
    (only on Tuesdays 9-11)
  > Team C: Autonomous Underwater Vehicles (AUVs)
    (small team)

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

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

> Enrollement in respective Stud.IP course

1. 23850 (Vorlesung) KSW: Verteiltes Hochleistungsrechnen
2. 23848 (Vorlesung) Neueste Entwicklungen der Informatik (Verteiltes Hochleistungsrechnen)
3. 23851 (Projekt) Projekt: Verteiltes Hochleistungsrechnen

> Questions via email to Peter Danielis and Helge Parzyjegla

> peter.danielis@uni-rostock.de
> helge.parzyjegla@uni-rostock.de