Universität Rostock

Traditio et Innovatio

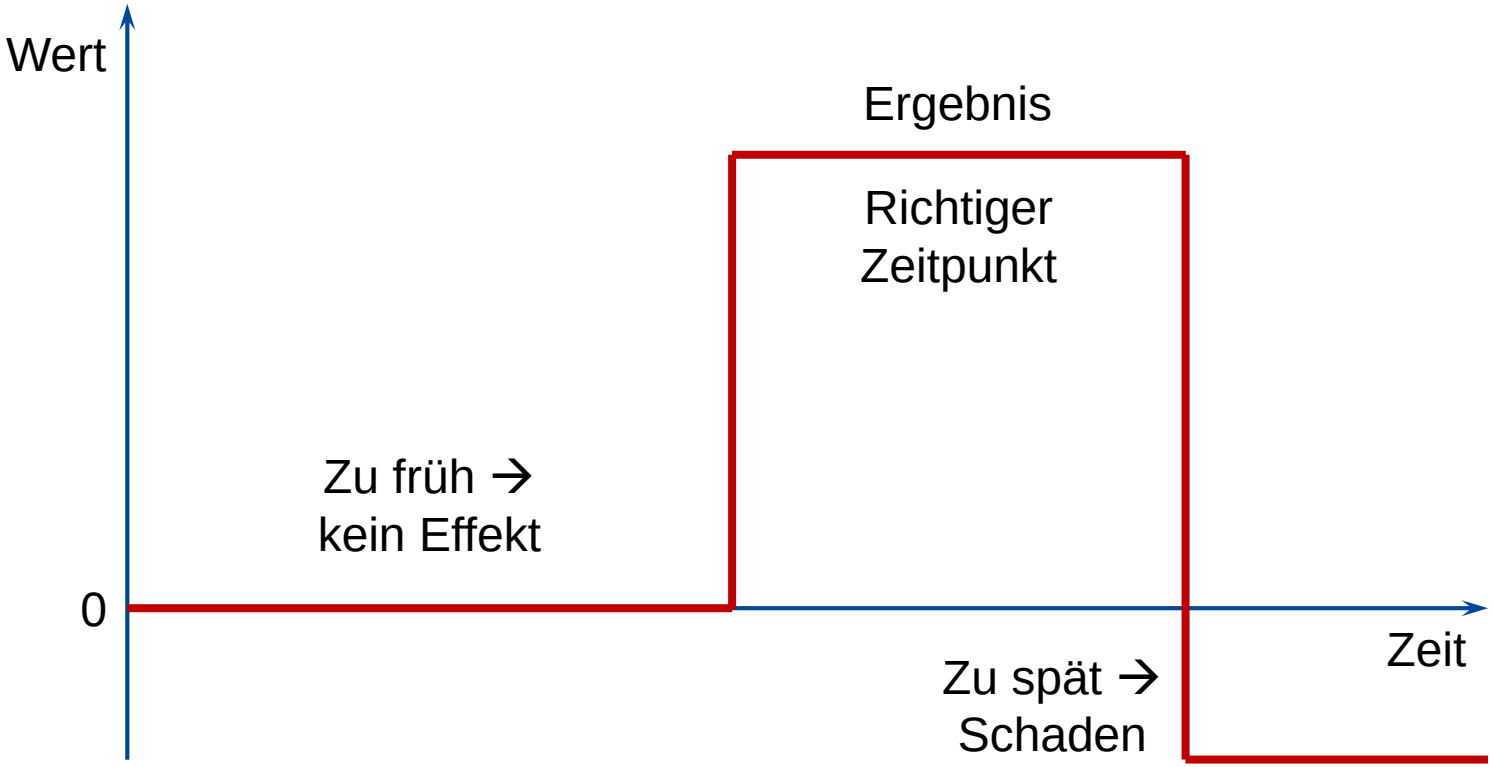
Realtime Publish/Subscribe für Cyber-Physische Systeme

KSWS / Projekt

Peter Danielis
Parallele Systeme (ParSys)

Helge Parzyjegla
Architektur von Anwendungssystemen (AVA)

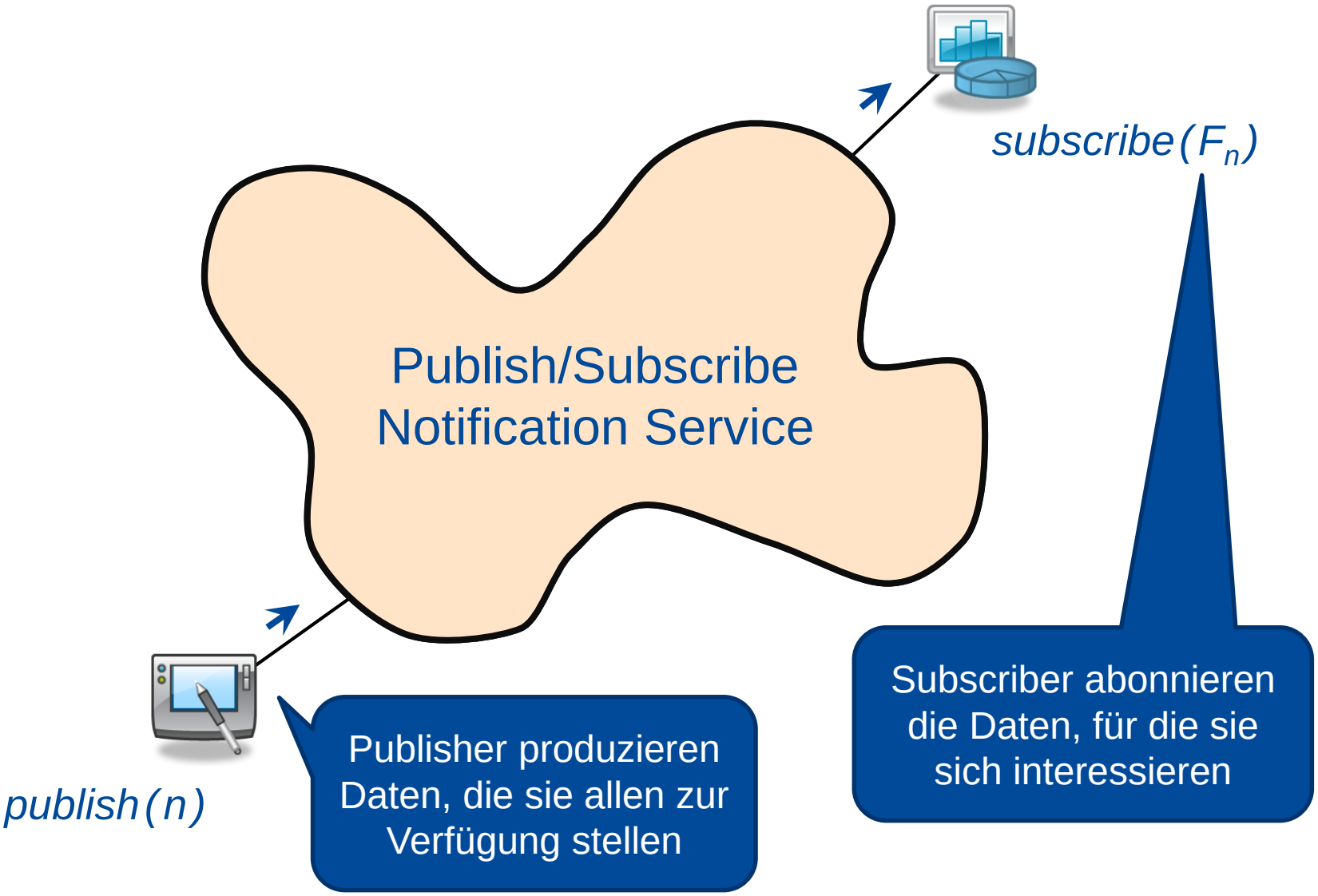
Was bedeutet Realtime/Echtzeit?

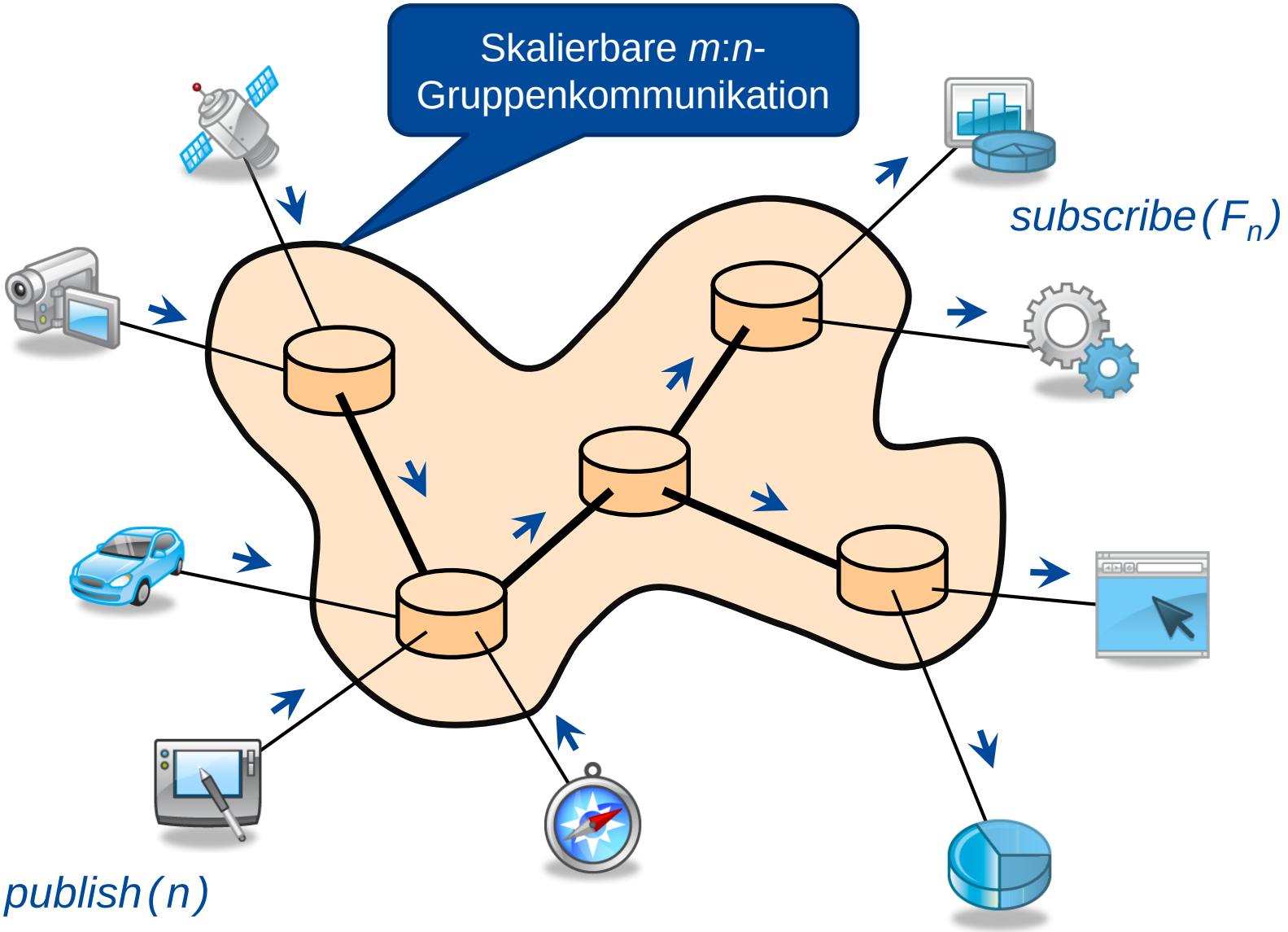


Nicht notwendigerweise schnell, sondern vorhersagbar!

→ Das Richtige zum richtigen Zeitpunkt tun.

Was ist Publish/Subscribe?




Skalierbare m:n-Gruppenkommunikation
subscribe(F_n)
publish(n)

Was sind Cyber-Physische Systeme (CPS)?

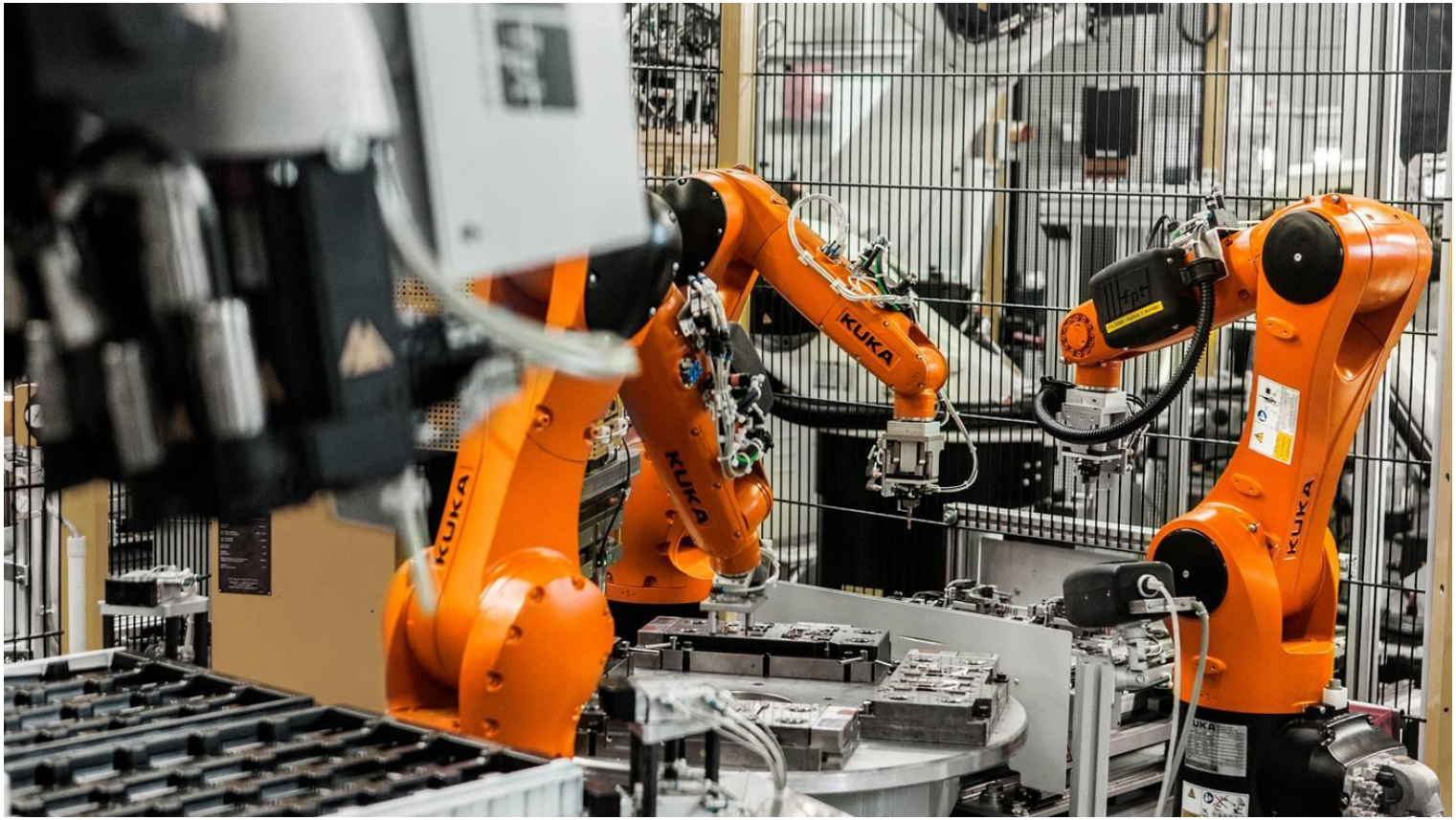
- Systeme bestehend aus Software-Komponenten und mechanischen bzw. elektronischen Teilen verbunden über ein Kommunikationsnetz
- Wirken auf die reale, physische Welt ein
  → unterliegen physikalischen Gesetzen
  → haben zeitliche Anforderungen (Echtzeit)
- Beispiele
  - Industrieroboter
    - Fertigungsstraße in der Smart Factory
    - Rekonfigurierbare Produktionszelle einer Smart Factory
  - Moderne (autonome) Fahrzeuge
    - Steer/Fly-By-Wire
    - Autopilotfunktionen jeglicher Art

Industrieroboter in der Smart Factory

Fertigungsroboter von Kuka

Zeitkritische Kommunikation bei Übergabe eines Werkstücks.

Rekonfigurierbare Produktionszelle

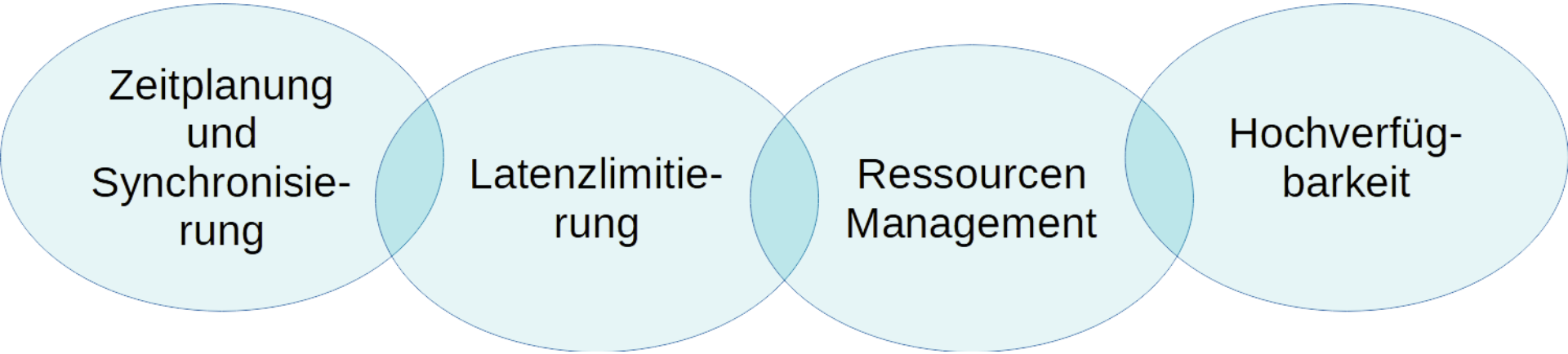


Fertigungsroboter von Kuka

Flexible Kommunikation bei Aufgabenänderung.

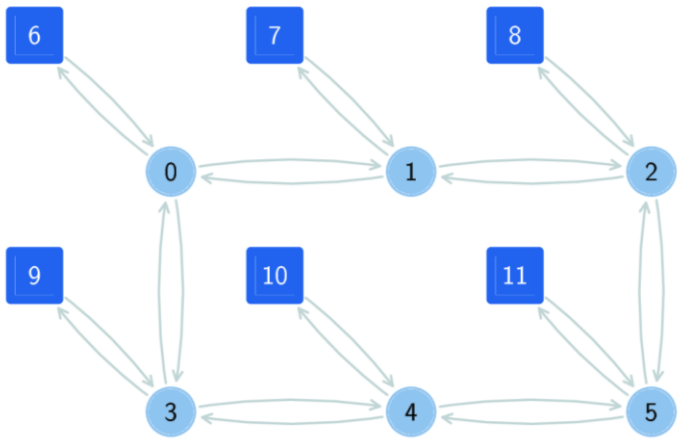
Echtzeit-Ethernet: Time-Sensitive Networking

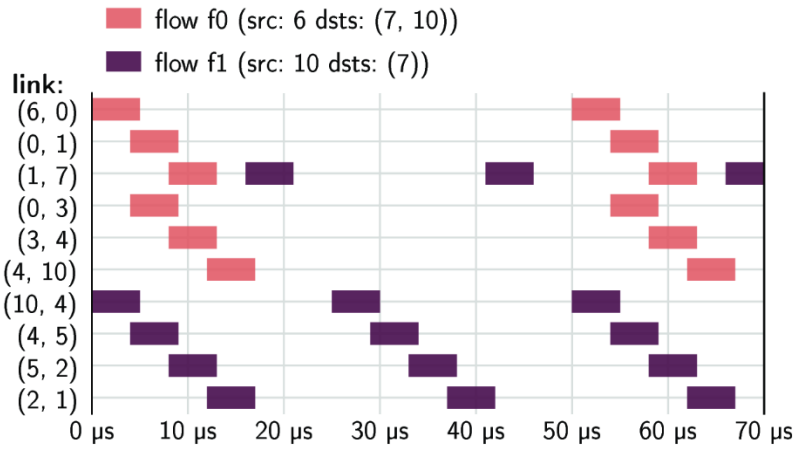
- Drahtgebundene Kommunikation nach IEEE 802.1Q
- Erweiterung von Ethernet
- Anpassung auf ISO/OSI-Schicht 2 (Sicherungsschicht)
- Baukastenprinzip



- TSN-Netze müssen konfiguriert werden: hierfür benötigen wir eine Planung!

Geplante Echtzeitkommunikation



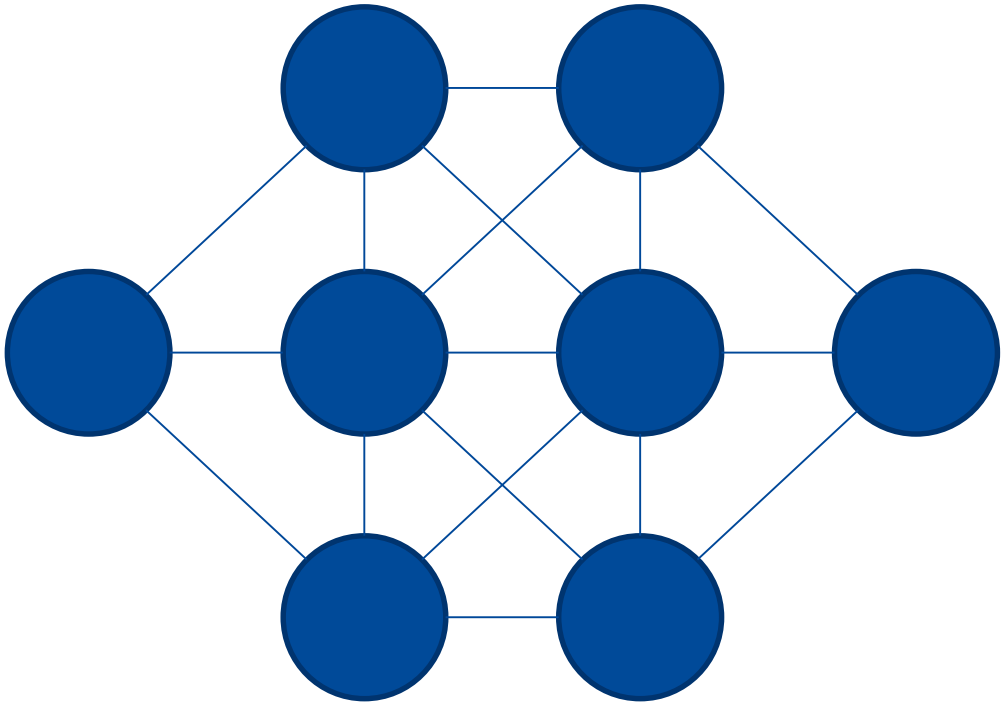


- Streams
  - Von Knoten 6 zu Knoten 7 und 10 (Multicast)
  - Von Knoten 10 zu Knoten 7
- Ablaufplan (engl. Schedule)
  - Bestimmt, wann welches Paket über welchen Link gesendet wird
  - Stets ohne Konflikte → nachweisbar korrekt
  - Anpassung bei Änderungen des Kommunikationsmusters
  - Zusätzlicher, weniger wichtiger Datenverkehr möglich

Constraint-basierte Programmierung

- Kommunikationspläne fürs Netzwerk werden mittels Constraint-basierter Programmierung erstellt
  - Beispiel Integer Linear Programming (ILP)
- Nutzer beschreibt das Problem
  - Angabe von einzuhaltenden Bedingungen → Constraints
  - Angabe von Zielfunktionen → (Lineare) Optimierung
- Computer/Solver löst das Problem
  - Finden von Variablenbelegungen
    - die Constraints erfüllen
    - Zielfunktionen optimieren

Puzzle

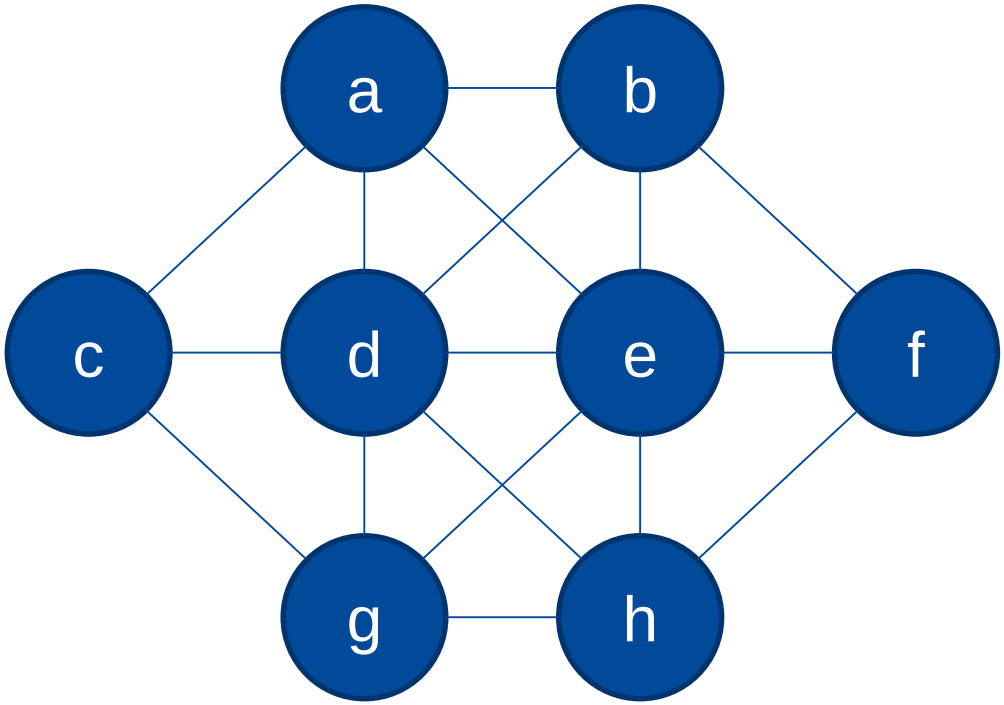


Die Zahlen 1 bis 8 auf die Felder verteilen, so dass benachbarte Felder keine aufeinanderfolgenden Zahlen beinhalten!

Lösungsideen

- Erschöpfende Suche
  - Alle Belegungen durchprobieren
  - Computer können viele Varianten schnell testen
  - Suchraum kann auch für Computer sehr groß sein
- Heuristiken
  - Felder mit den meisten Einschränkungen identifizieren
  - Zahlen mit den wenigsten Einschränkungen identifizieren
  - Kombinieren und Folgern
- Constraint-basierte Programmierung
  - Modellierung der Einschränkungen
  - Verständlichkeit für den Solver

Modellierung mittels Constraints (i)



Variablen und Constraints.

Modellierung mittels Constraints (ii)

Unterschiedliche Werte

- $a \neq b, a \neq c, a \neq d, \ldots$
- $b \neq c, b \neq d, \ldots$
- $\ldots$
- $g \neq h$

Benachbarte Knoten nicht aufeinanderfolgende Werte

- $|a - b| \geq 2$
- $|a - c| \geq 2$
- $\ldots$

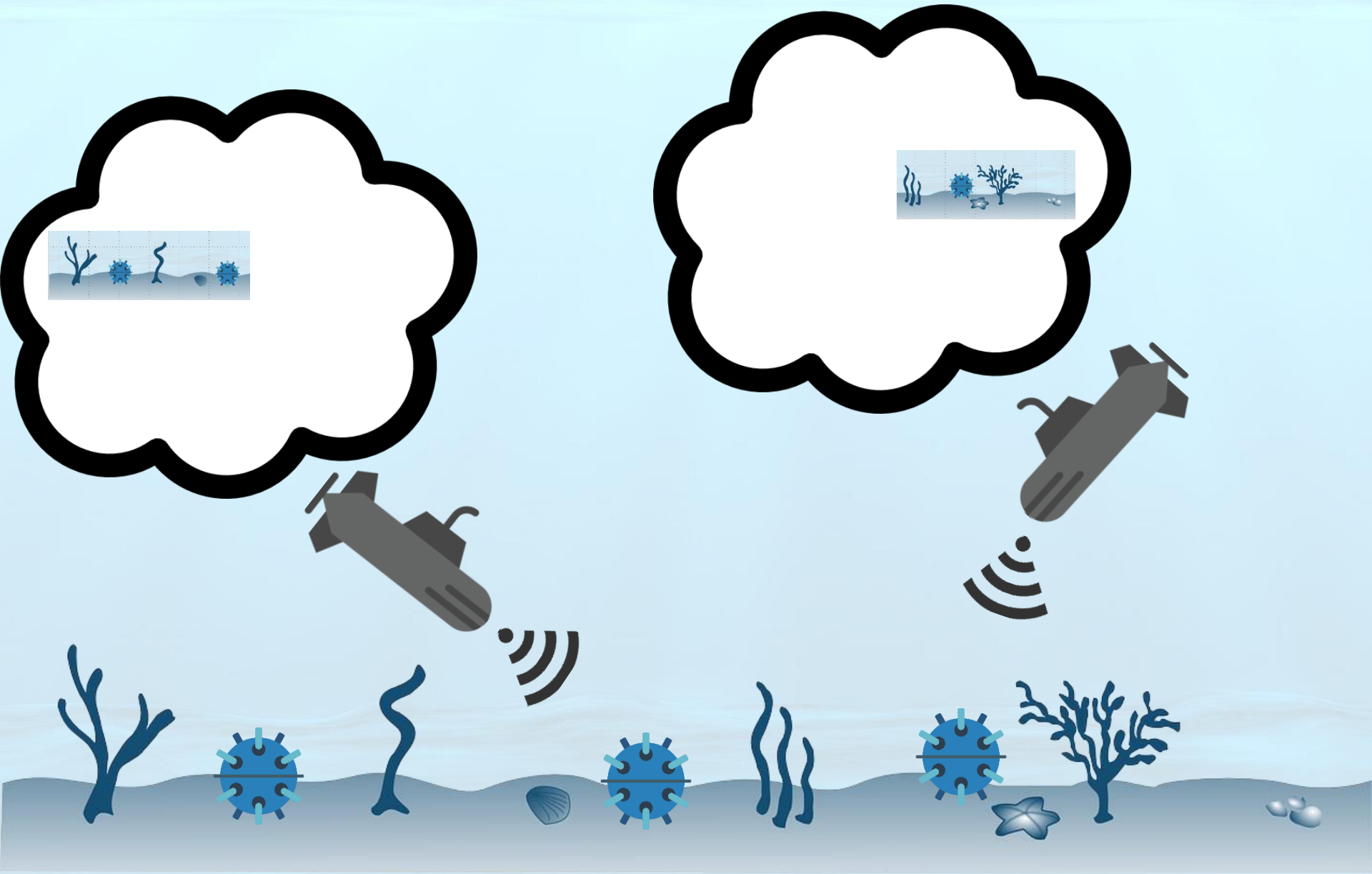
Danach einfach Solver.solve()!

Drahtlose Kommunikation

- Kommunikationsvielfalt in CPS
  - Kombination aus drahtgebundener Stabilität und drahtloser Flexibilität, um verschiedene Anwendungsanforderungen zu erfüllen
- 5G als Schlüsselfaktor
  - Ultra-latenzarme und hochzuverlässige drahtlose Kommunikation für mobile und dynamische CPS-Anwendungen
- Herausforderungen durch 5G
  - Sicherheitsrisiken, Interoperabilitätsfragen und Energieeffizienz als zentrale Themen bei der Integration in CPS

Unterwasser-Szenario 1:
Wartung der Fundamente von Offshore-Windkraftanlagen

Unterwasser-Szenario 2: Räumung von Altlasten aus dem 2. Weltkrieg (Blindgänger, Unexploded Ordnance - UXO)



Projekte und Kooperationen

- Echtzeitfähige Publish/Subscribe-Kommunikation
  - Teil eines DFG-Projektes
  - Planung flexibler Kommunikationsmuster und Reservierung notwendiger Zeitslots auf den Kommunikationsverbindungen
  - Abschätzung der Worst-Case-Laufzeit einer Publikation und deren (ggf. inhaltsbasierter) Filterung und Auslieferung
  - Einsatzgebiet in der Smart Factory
- Autonome Unterwasserfahrzeuge (AUVs)
  - Zusammenarbeit mit dem Institut für den Schutz maritimer Infrastrukturen, Abteilung Resilienz maritimer Systeme, Deutsches Zentrum für Luft- und Raumfahrt (DLR) Bremerhaven
  - Kooperative Navigation mehrerer AUVs
  - Begrenzte Energie limitiert Sensoren und Bewegung
  - Opportunistische Kommunikation durch Akustikmodems

Aufgaben: TSN-Standards in OMNeT++

- OMNeT++-Framework INET umfasst Simulationsmodelle für TSN-Standards
- INET bietet diverse TSN-Showcases zur Demonstration der Funktionalität dieser Standards

- Aufgabe TSN.1: Showcases
  - Inbetriebnahme eines Showcases
  - Nachbau des Showcases in eigenem Use Case
  - Ggf. Integration fehlender TSN-Features in die Simulationsmodelle
- Aufgabe TSN.2: Tutorial
  - Entwurf eines Tutorials für eine wissenschaftliche Konferenz
  - Theoretische Erklärung und praktische Demo eines TSN-Standards

Aufgaben: 5G-Kommunikation in OMNeT++

- Simu5G: Simulator für 5G NR und LTE/LTE-A Netzwerke, integriert in OMNeT++ und INET
- Funktionen: Simulation der Datenebene in 5G RAN und Kernnetz mit FDD/TDD, heterogene gNBs, D2D-Kommunikation und Dual Connectivity

- Aufgabe 5G.1: Showcases
  - Inbetriebnahme von Simu5G
  - Analyse und Test von Simu5G
  - Entwicklung eigener Showcases

Aufgaben: Autonome Fahrzeuge in OMNeT++

- Nutzung eines OMNeT++-Simulationsmodells für autonome Fahrzeuge (Autonomous Vehicles, AVs)
- Simulationsmodell ist modular aufgebaut und lässt sich leicht anpassen

- Aufgabe AV.1: Energiemodell
  - Inbetriebnahme des Simulationsmodells in OMNeT++
  - Analyse verschiedener Energiemodelle und Verbrauchskennlinien
  - Integration eines Energiemodells
- Aufgabe AV.2: Kooperative Missionen
  - Formation mehrerer AUVs
  - Kartierung des Meeresbodens
  - Kooperative Jagd

Aufgaben: ILP/Constraint-Programmierung

- Berechnung von TSN-Schedules mittels ILP-Solver
  - Modellierung der Problemstellung als ILP
  - Programmierung des Problems
  - Eingabe, Gleichungen und Ausgabe
  - Optimierungen und Evaluation
- Gurobi als ILP-Solver
  - Industriestandard bei linearer Optimierung
  - Kommerziell mit akademischer Lizenz
  - Eventuell Nutzung einer Cloud-Installation
- Programmierung in Python
  - Python zur Verarbeitung der Ein- und Ausgabe
  - Python als Programmiersprache für Gurobi

Organisatorisches

- Wöchentliche Treffen Donnerstag, 13:00 Uhr, R 301 (AE22)

- Bis zu zwei Teams
  - Team A: TSN, 5G und AUVs (Peter)
    (wahrscheinlich feingliedrigere Aufgabenaufteilung)
  - Team B: ILP/Constraint-Programmierung (Helge)
- Entwicklungsmethodik
  - Agile Entwicklung
  - Drei Meilensteine bzgl. Entwurf, Implementierung, Bericht

Art und Umfang der Aufgaben nach Anzahl
und Interessen der Teilnehmer!

Anmeldung und Kontakt

- Eintrag in die richtige Stud.IP-Veranstaltung

1.  23850 Vorlesung: KSWS: AVA
2.  23851 Projekt: Projekt: AVA

- Fragen an Peter Danielis und Helge Parzyjegla per E-Mail
  - peter.danielis@uni-rostock.de
  - helge.parzyjegla@uni-rostock.de