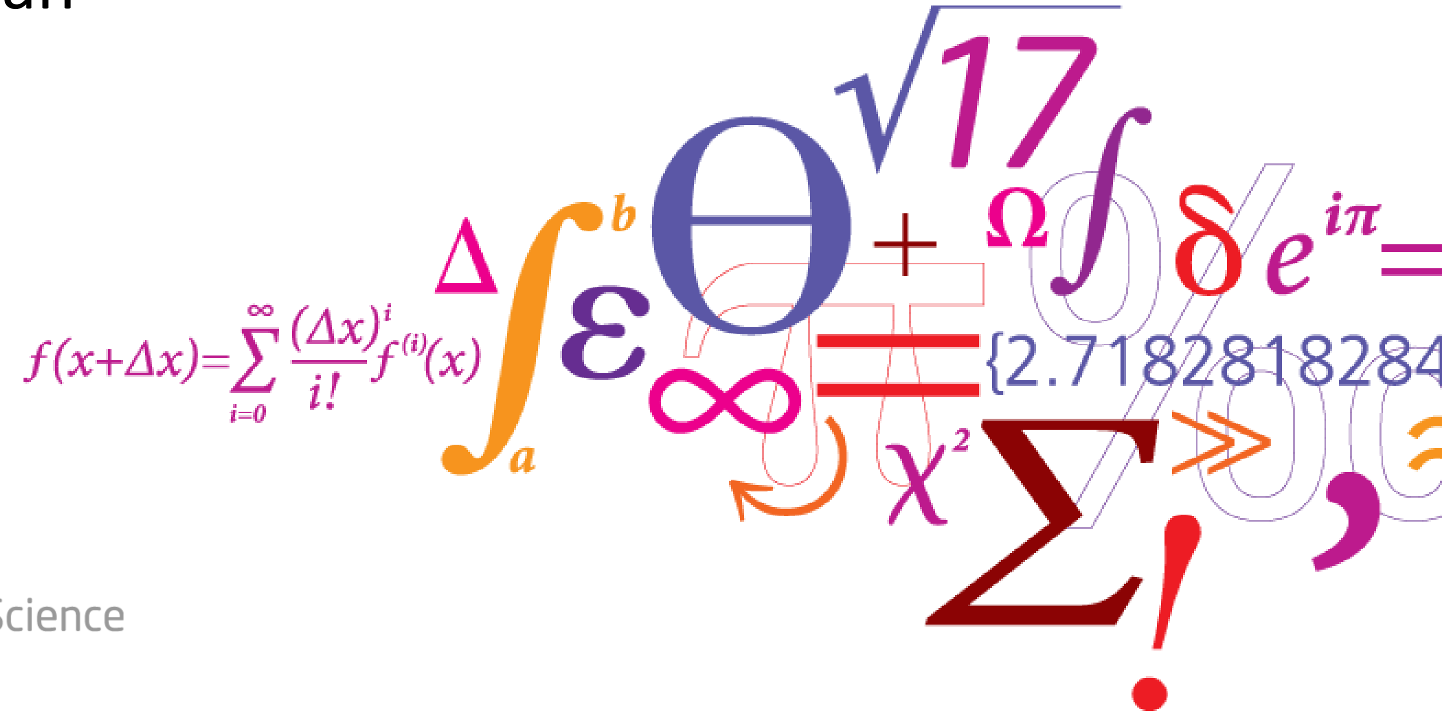
Configuration Optimization of Fog Computing Platforms for Control Applications

Mohammadreza (Reza) Barzegaran

June 14th 2021



DTU Compute
Department of Applied Mathematics and Computer Science

Outline

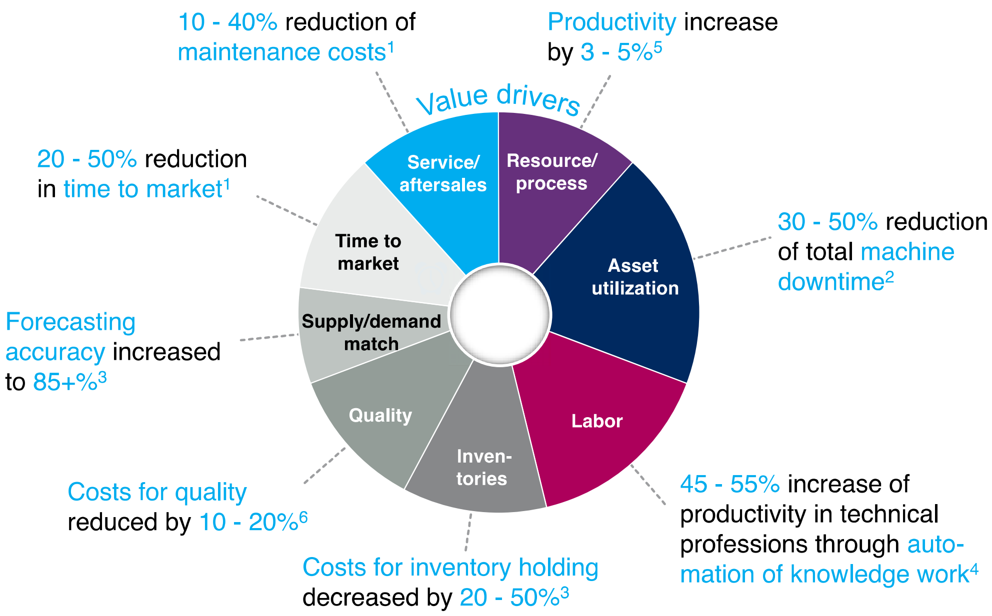
- Motivation
- Paper A: Computation-level optimizations targeting control applications
  “Performance Optimization of Control Applications on Fog Computing Platforms Using Scheduling and Isolation”
- Paper B: Communication-level optimizations targeting control applications
  “Communication Scheduling for Control Performance in TSN-Based Fog Computing Platforms”
- Paper C: Design for extensibility targeting both control applications and dynamic Fog applications
  “Extensibility-Aware Fog Computing Platform Configuration for Mixed-Criticality Applications”
- Paper D: Evaluating the methods using a realistic use case and demonstrator
  “Electric Drives as Fog Nodes in a Fog Computing-based Industrial Use Case”
- Summary
- Research output and list of publications

Motivation

Industry 4.0: *Digitalization of the manufacturing and industrial sectors, with embedded sensors in virtually all product components and manufacturing equipment, ubiquitous cyberphysical systems, and analysis of all relevant data.*

Benefits (manufacturing)

Indicative quantification of value drivers



Source: McKinsey

Digitalization of all areas

Source: Overview The Internet Of Things (IoT) System Security, Applications, Architecture And Business Models

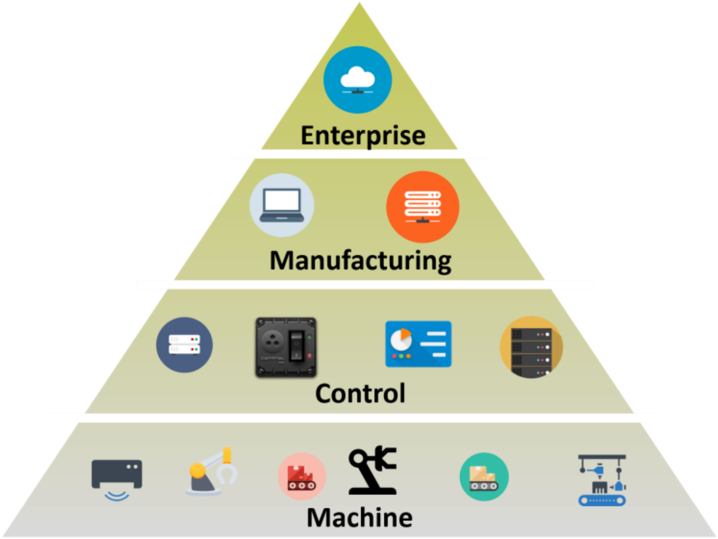
FORA: Fog Computing for Robotics and Industrial Automation

- **The FORA European Training Network**
  - Marie Skłodowska-Curie Action
  - 15 PhD students, 5 universities, 4 countries

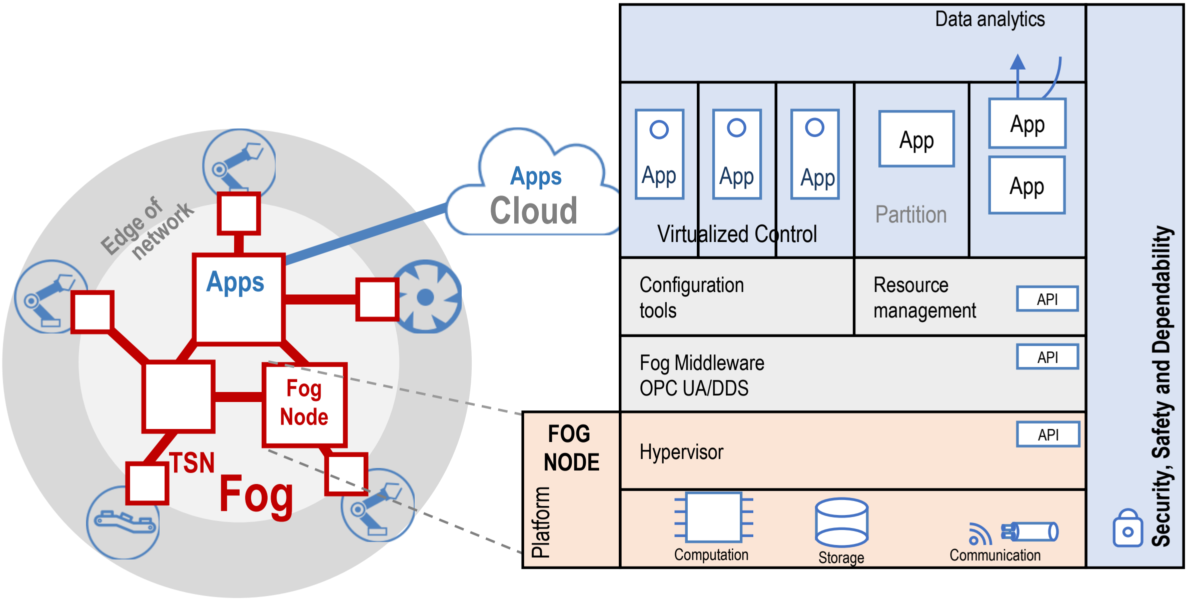
Vision: open Fog Computing-based architecture built on open source and open standards, e.g., Time-Sensitive Networking (TSN), OPC Unified Architecture (UA) and 5G.

- **My objectives in FORA: virtualization of control**
  - Develop methods and tools for the configuration of a Fog Computing Platform for critical control applications

Fog Computing *is a system-level architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from Cloud to Things*



Current closed architectures using the automation pyramid (Purdue model)



Industry 4.0 architecture based on Fog Computing and Deterministic Networking (TSN)

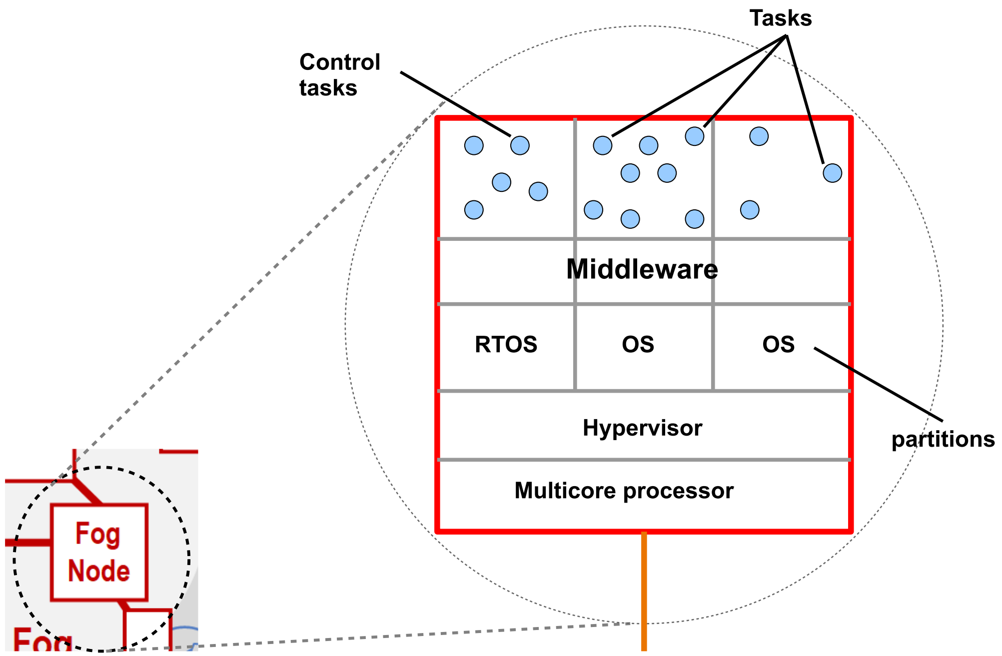
Paper A

Performance Optimization of Control Applications on Fog Computing Platforms using Scheduling and Isolation

M. Barzegaran, A. Cervin and P. Pop, "Performance Optimization of Control Applications on Fog Computing Platforms using Scheduling and Isolation," in IEEE Access, vol. 8, pp. 104085-104098, 2020.

Fog Computing Platform Architecture

- **The Fog Computing Platform (FCP) runs mixed-criticality applications, including control applications**
  - An FCP is composed of several interconnected Fog Nodes (FNs), from powerful multicore FNs to low-end FNs
  - The control applications are virtualized as tasks running on the Fog Nodes (FNs) of the FCP
  - Partitioning is used to isolate applications of different criticalities
    - Each partition can have its own operating system (OS)
    - The partitions running the control applications use a real-time operating system (RTOS)

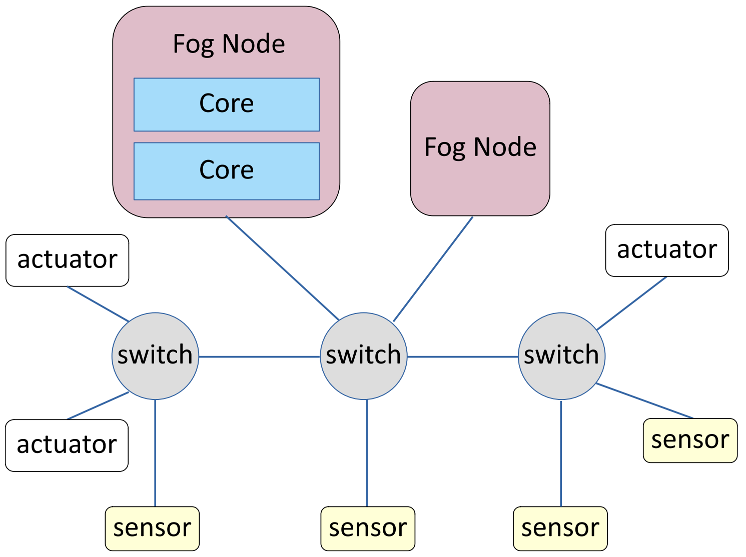


Example architecture of a Fog Node

Architecture model

Architecture model

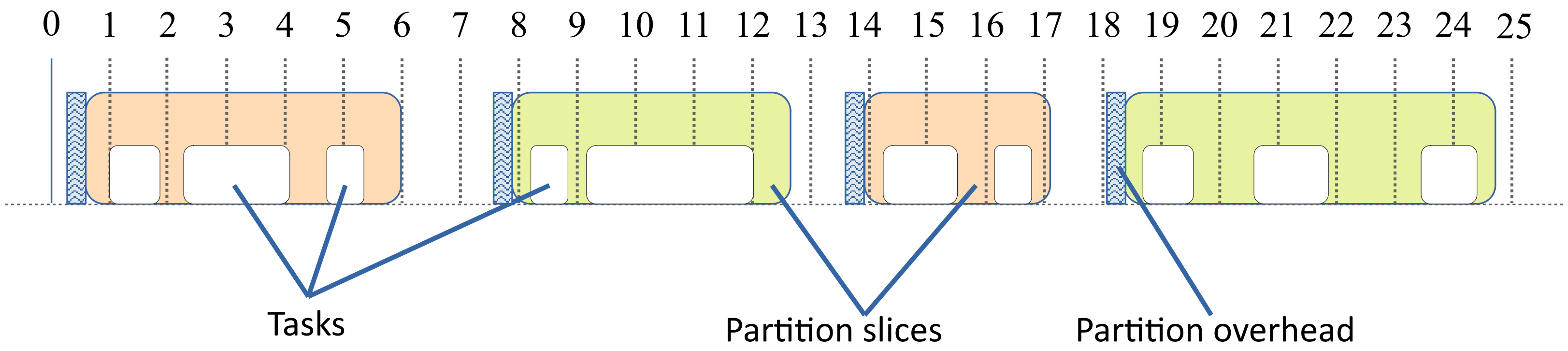
- Set of Fog nodes; each Fog Node has
  - Multiple cores
  - A hypervisor and a time-triggered scheduler
    - Isolation of mixed-criticality applications



Example architecture model

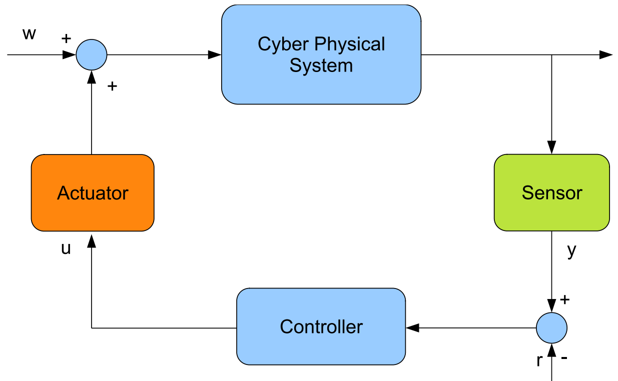
Partition tables and scheduling

- Partitions are statically scheduled using partition tables (e.g., as in PikeOS)
- Time-triggered scheduling is used to run the tasks, and communication is ignored (we'll revisit this in Paper B)



Example schedule table

Control applications and their performance



A simple feedback control system

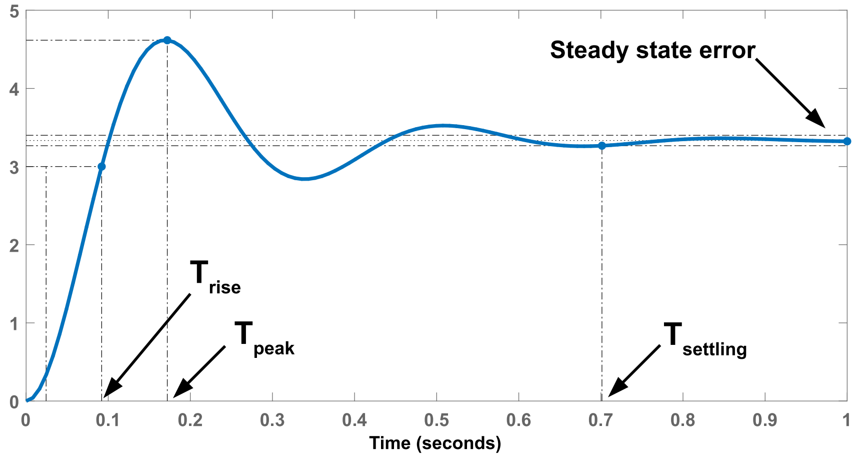
A **Feedback control system (FCS)** or control application operates and commands a dynamical system (robots and industrial machines) using a control algorithm.

- Can be implemented as a three-task application: sampling, control law, and actuation tasks

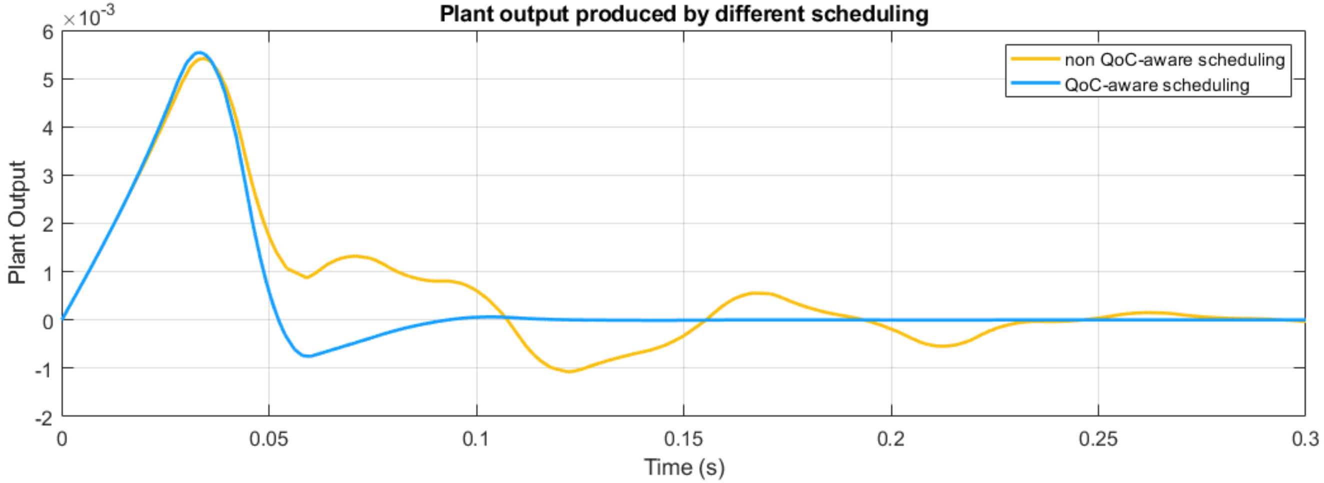
Control performance (QoC)

- Captures the trade-off between the accuracy and the rapidity of the controller
- We use a quadratic cost function (J) proposed in the literature
- QoC is calculated using the JitterTime tool, which simulates control applications

$$J = \int_0^\infty \left(x^T(t) Q_1 x(t) + u^T(t) Q_2 u(t) \right) dt$$



Step response of a control loop



Scheduling effect on the control output

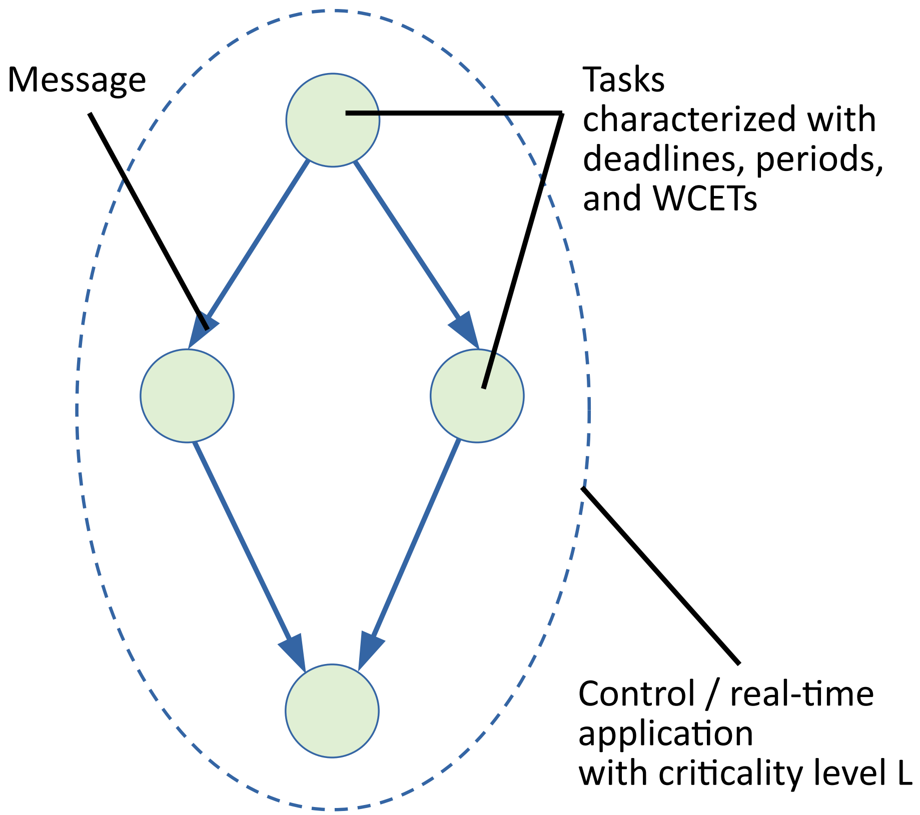
A. Cervin, P. Pazzaglia, M. Barzegaran, and R. Mahfouzi, "*Using JitterTime to analyze transient performance in adaptive and reconfigurable control systems*," In Proceeding of IEEE International Conference on Emerging Technologies and Factory Automation. pp. 1025-1032, 2019.

Application model

- Real-time systems: "*the correctness of the system behavior depends not only on the logical results of the computations, but also on the physical time when these results are produced*"
- Safety-critical systems: their failure may lead to loss of life or damage

- Control applications are real-time and may be safety-critical
  - The problem of redundancy optimization is orthogonal to our work
  - They share the same Fog Nodes with real-time applications
  - The criticality of an application is modeled via a *criticality level* L

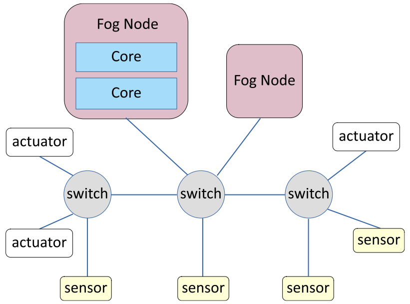
- Control and real-time applications
  - Are modeled as directed acyclic graphs (DAGs)
  - Each node in a DAG is a task
    - Task are periodic and have hard deadlines
    - We know their worst-case execution time (WCET)
    - They exchange messages

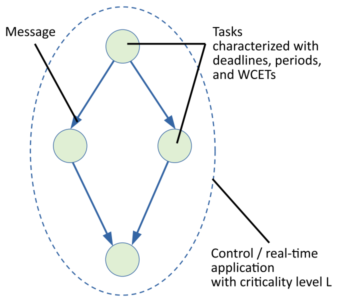


Example application model

Problem Formulation

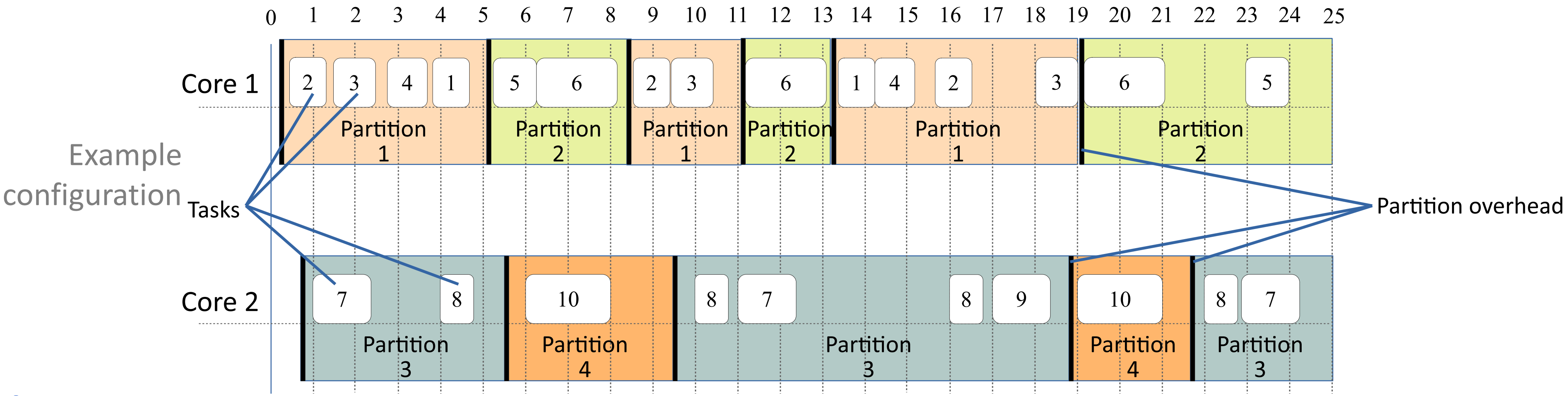
Given: Application and architecture models





Determine: An FCP configuration:

- partitions
- mapping of tasks to the cores
- assignment of tasks to partitions
- the period of control applications
- partition tables
- task schedule tables



Such that:

- The QoC of applications is maximized and “balanced” across applications
- The mixed-criticality applications are isolated within partitions, and the deadlines are satisfied

Fog Computing Platform Configuration (FCPC) optimization strategy

Simulated Annealing metaheuristic searches iteratively for *solutions*, decides in *each iteration*

Part of the FCP configuration:

- Period of the control applications
- Mapping of tasks to cores in the FCP

Parameters to control the scheduling heuristic

- Task *offsets*
- Task *relative deadlines*

EDF-based Scheduling heuristic decides

The rest of the FCP configuration

- The schedule tables
- The assignment of tasks to partitions
- The partitions

EDF simulation creates the static schedule

- Parameters: *relative deadlines*, *offsets*

The current configuration is evaluated using the cost function (JitterTime for QoC)

SA iterates until a termination criteria is satisfied

- SA uses random design transformations to generate neighboring solutions (tasks mapping, periods, deadlines and offsets)
- SA is a variant of a hill-climbing heuristic: worst solutions may be accepted depending on a "temperature" parameter

- The scheduling heuristic simulates the tasks as if they would execute up to their WCET with an "Earliest Deadline First" (EDF) scheduling policy
- EDF is a dynamic scheduling algorithm that picks the task with the "earliest" relative *deadline* (considering *offsets*) and allows preemptions

Evaluation results

Fog Computing Platform Configuration (FCPC) has 3 variants:

1. FCPC/M—ignores mapping optimization
2. FCPC/Q—ignores the QoC optimization
3. FCPC/P—ignores the control applications' period optimization

Test cases	No. of Cores	Total No. of Control Applications	Total No. of Tasks	Total No. of Tasks for Criticality level of [0-4]	Ω of FCPC	Ω for FCPC/M	Ω for FCPC/Q	Ω for FCPC/P
1	2	2	12	{1,2,2,1,6}	0,19	48%	Not Feasible	60%
2	2	3	23	{3,2,5,4,9}	0,34	5%	Not Feasible	Not Feasible
3	2	2	17	{2,2,4,3,6}	0,21	5%	Not Feasible	33%
4	2	2	23	{2,1,7,6,7}	0,29	13%	Not Feasible	13%
5	3	3	32	{1,4,8,8,11}	0,21	39%	Not Feasible	55%
6	3	3	31	{2,3,9,7,10}	0,22	85%	Not Feasible	50%
7	3	4	33	{4,4,8,7,12}	0,26	15%	Not Feasible	10%
8	4	4	44	{4,6,11,9,14}	0,20	29%	Not Feasible	72%
9	5	6	54	{7,5,14,10,18}	0,21	21%	Not Feasible	Not Feasible
10	6	7	63	{6,7,16,12,22}	0,24	21%	Not Feasible	46%

Ω is the value of the cost function; a small value means better and well-balanced QoC for control applications

Contributions of Paper A

- **Main contributions:**
  - Considering Fog-based virtualization via partitioning
  - Using a realistic model of control applications
  - Accurate measuring of QoC which simulates the behavior of a control application with JitterTime
  - Allowing preemption
  - Setting periods for control applications

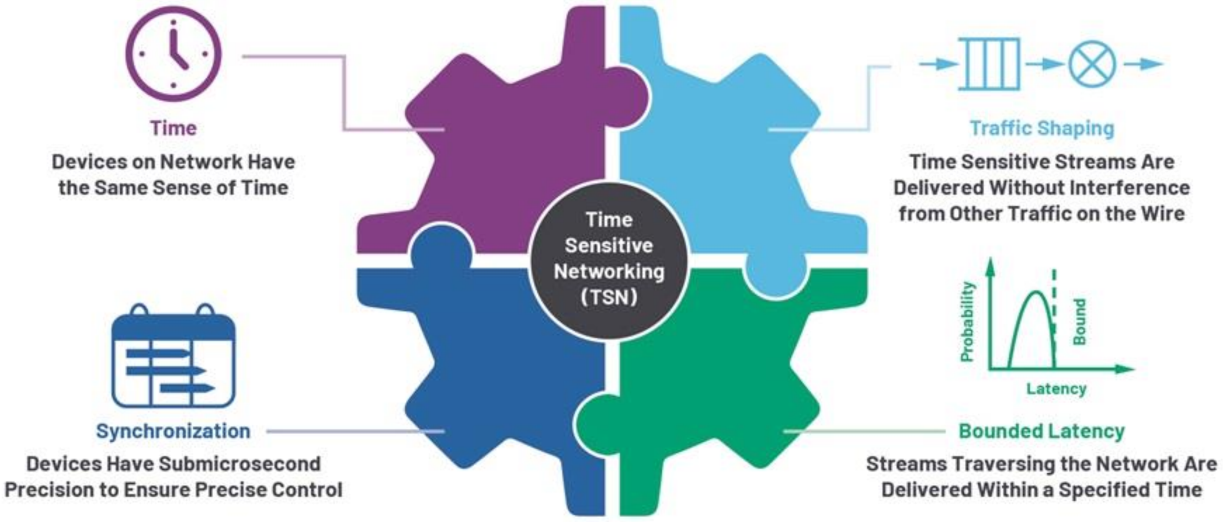
Paper B

Communication Scheduling for Control Performance in TSN-Based Fog Computing Platforms

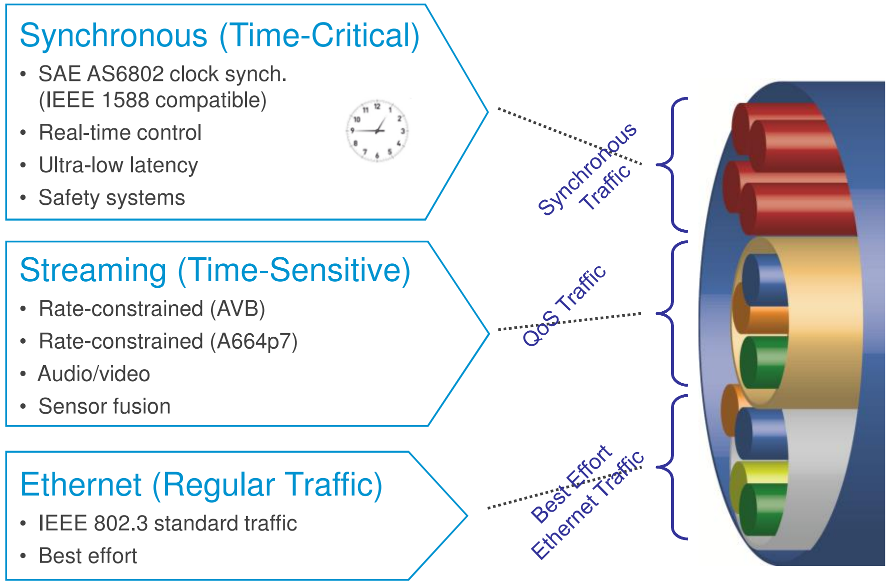
M. Barzegaran and P. Pop, "Communication Scheduling for Control Performance in TSN-Based Fog Computing Platforms," in IEEE Access, vol. 9, pp. 50782-50797, 2021.

IEEE 802.1 Time-Sensitive Networking

- **We already mentioned that the FCP runs mixed-criticality applications, including control applications**
  - Time-Sensitive Networking (TSN) is used to connect FNs to each other and to the machines
    - TSN consists of a set of amendments to the IEEE 802.3 Ethernet standard
    - TSN provides features useful for real-time and safety-critical applications
    - TSN supports multiple traffic types; Scheduled Traffic (ST), Audio-Video Bridging (AVB), and Best-Effort Traffic (BE)



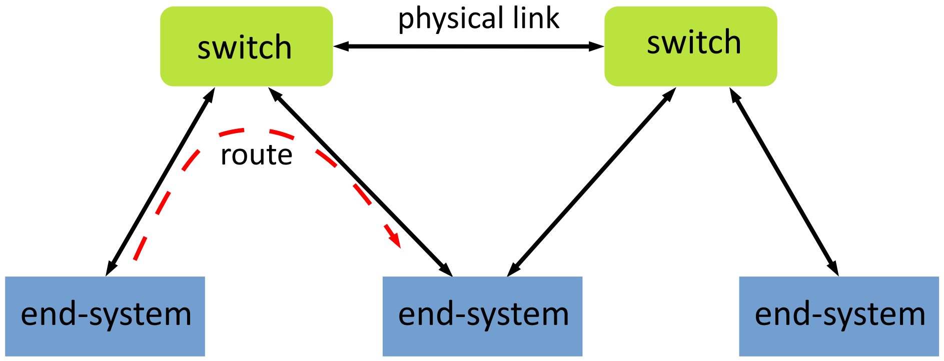
Source: Analog Devices



Architecture model

Architecture model

- A network graph
  - Vertices are network nodes (end-systems and switches)
    - End-systems send and receive messages
    - Switches forward messages
    - End-systems and switches can be integrated
  - Edges are physical links with known speed and transmission delay
    - Links are connected to the ports of switches and end-systems
- Routing; each route is an ordered list of links
- Messages are sent as streams
  - Each stream is responsible for sending frames encapsulating data
  - Each streams is sent form an end-system to one or multiple end-systems
  - Streams are transmitted via routes

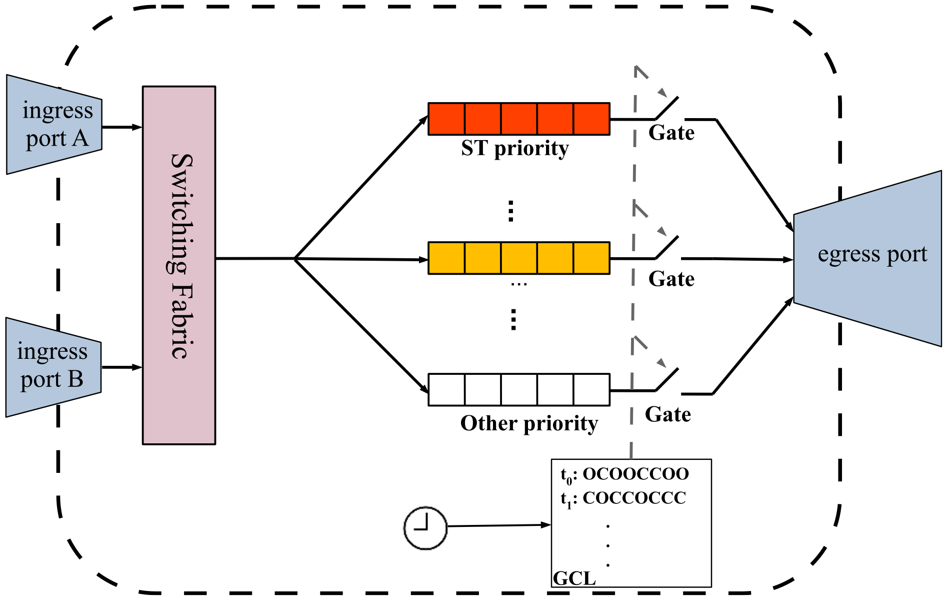


Example architecture model

GCL synthesis for IEEE 802.1Qbv

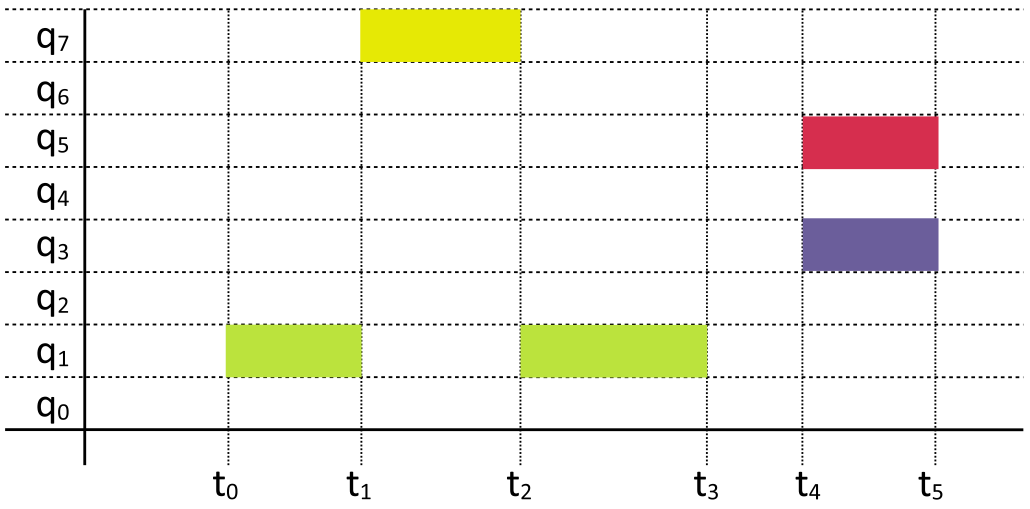
GCLs contain a cyclic timed list of updates to the gate status of queues

- Each GCL entry corresponds to a window for traffic transmission
- Each window can handle one or more network frames
- Assumption: end systems are “scheduled”, and flows are “isolated”, so frames can be individually scheduled via GCLs
  - Barzegaran et al. 2021 relaxes these assumptions



Time	q_7	q_6	q_5	q_4	q_3	q_2	q_1	q_0
t_0	C	C	C	C	C	C	O	C
t_1	O	C	C	C	C	C	C	C
t_2	C	C	C	C	C	C	O	C
t_3	C	C	C	C	C	C	C	C
t_4	C	C	O	C	O	C	C	C
t_5	C	C	C	C	C	C	C	C

Example GCL

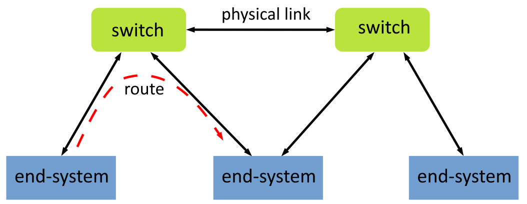


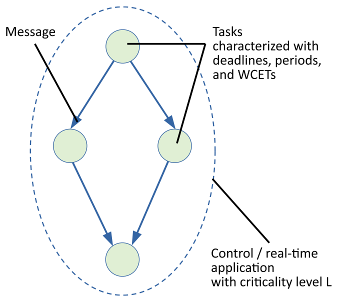
Correspondent schedule table

M. Barzegaran, N. Reusch, L. Zhao, S. Craciunas, and P. Pop. ”Real-Time Guarantees for Critical Traffic in IEEE 802.1Qbv TSN Networks with Un-scheduled End-Systems,” In arXiv, 2021.

Problem Formulation

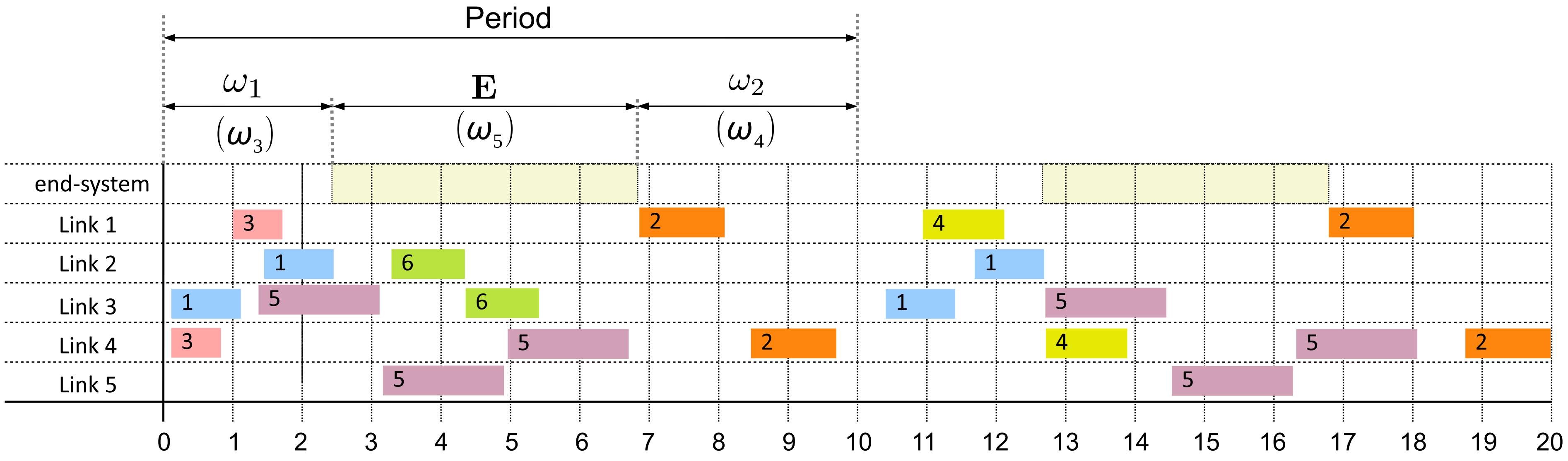
Given: Application and architecture models





Determine: An FCP configuration consists of GCLs.

Example configuration

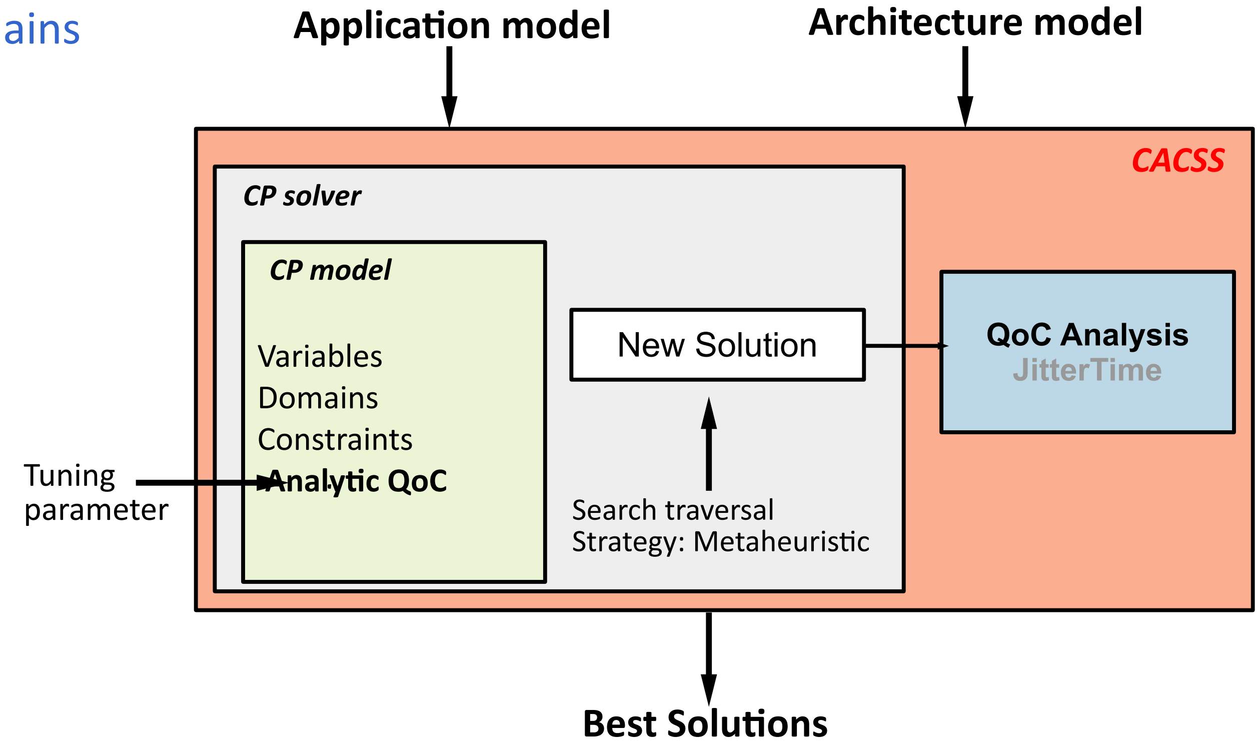


Such that:

- The QoC of control applications is maximized
  - The QoC is captured using an analytic model
- The deadlines for mixed-criticality applications are satisfied

Control-Aware Communication Scheduling Strategy (CACSS)

- CP model consist of a set of variables and their domains
  - The variable domains are related by constraints

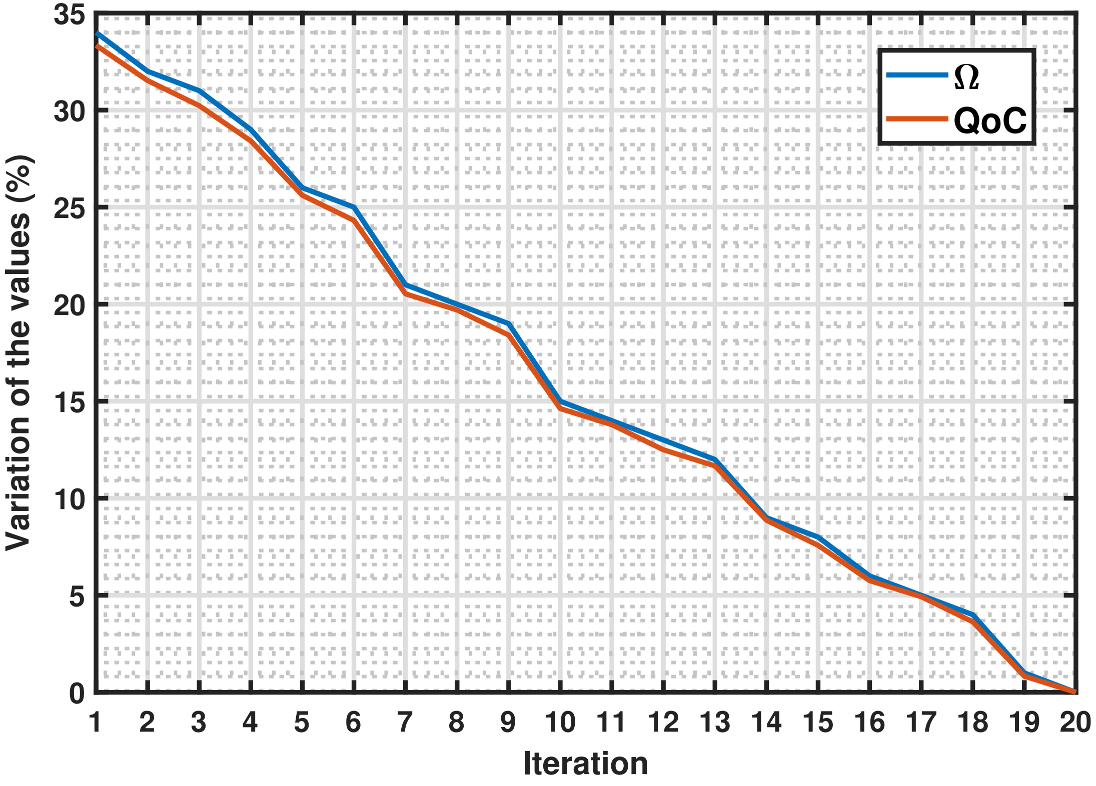


- Constraint Programming (CP) is a declarative programming paradigm
- CP visits solutions that satisfy the constraints and evaluates them using an objective function
- The visited solutions are evaluated using the analytic QoC model
- For each improving solution JitterTime calculates the accurate QoC value

Evaluation results

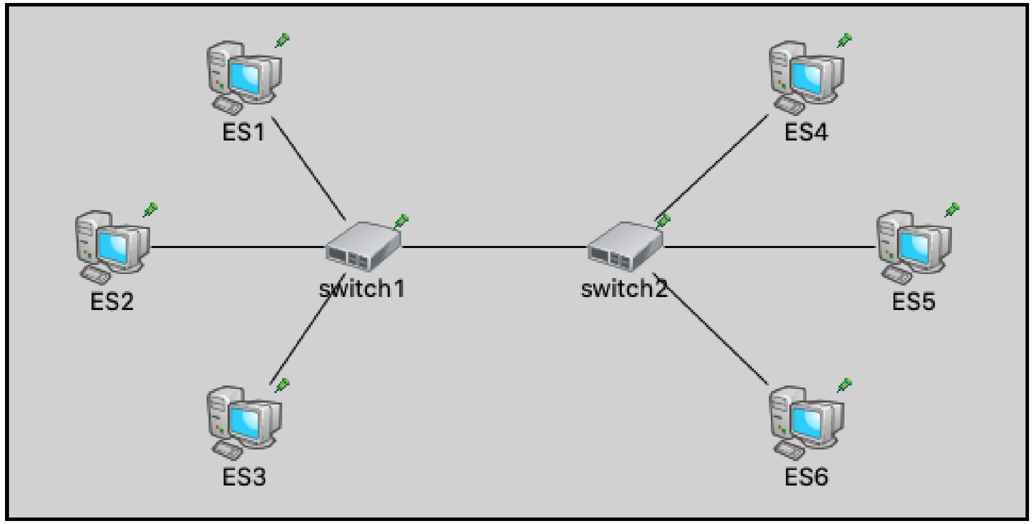
We have validated

- The ability of the analytical QoC model to drive the search



Comparison of analytical QoC model with JitterTime

- The generated GCLs using OMNET++ simulation



OMNET++ implementation of a test case

Evaluation results

Control-Aware Communication Scheduling Strategy (CACSS) has been compared to the approaches of the related work:

1. Zero-Jitter GCL (ZJGCL)—proposed by Silviu S. Craciunas (Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks)
2. Frame-to-Window GCL (FWGCL)—proposed by Ramon Serna Oliver (IEEE 802.1Qbv gate control list synthesis using array theory encoding)

Item	ZJGCL	FWGCL	CACSS
Scheduling object	frame	window	frame
Frame isolation	Yes	No	Yes
Network precision	Yes	Yes	Yes
Control app. model	No	No	Yes
Forwarding delay	No	No	Yes
PPTP flows scheduling	No	No	Yes

PPTP: Point-to-Point Tunneling Protocol

#	Total no. of flows	Total no. of control apps	Total no. of SWs	Total no. of ESes	Total no. of frames	Ω	QoC for CACSS	QoC for ZJGCL	Runtime (ms)
1	8	1	2	6	53	66.1	0.862	1.335	132
2	12	1	2	6	68	66.0	0.860	1.415	138
3	14	2	2	6	60	77.4	0.959	1.409	147
4	8	1	3	6	77	110.2	1.367	1.985	170
5	16	3	4	8	89	67.1	0.872	1.605	621
6	24	3	5	10	171	67.3	0.881	1.821	1351
7	16	2	5	8	100	58.0	0.771	1.177	743
8	20	3	6	10	149	66.5	0.867	1.187	943
9	24	4	7	10	198	90.0	1.152	1.555	1465
10	30	4	7	10	244	90.0	1.153	1.661	1793
11	27	5	20	20	1770	151.0	1.889	2.957	6225
12	27	5	20	20	1834	151.0	1.897	3.805	9153
13	27	7	20	20	1770	149.6	1.865	3.411	3553

Ω is the value of the cost function; a small value means better and well-balanced QoC for control applications

Contributions of Paper B

- **Main contributions:**
  - Considering the impact of TSN scheduling on control performance
  - Constraint Programming (CP) formulation of the optimization of communication configuration
  - Considering the effect of forwarding delay in the CP formulation
  - Development of an analytical model for QoC integrated in the CP formulation
  - More accurate measuring of QoC using JitterTime

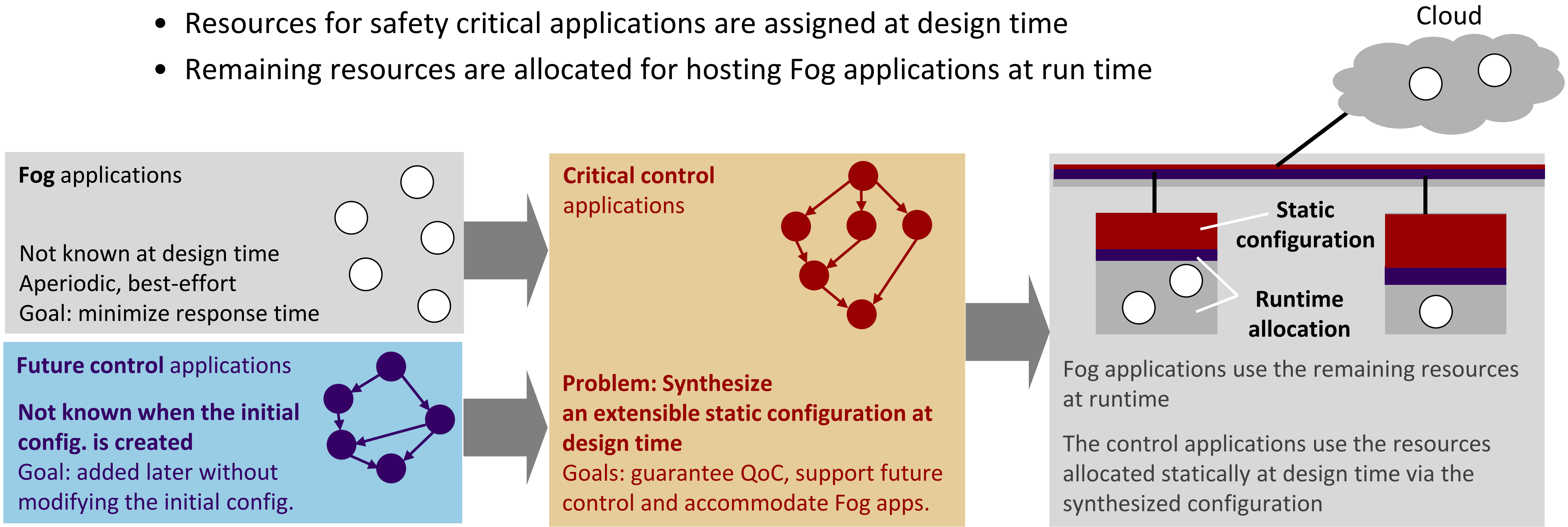
Paper C

Extensibility-Aware Fog Computing Platform Configuration for Mixed-Criticality Applications

M. Barzegaran and P. Pop. “Extensibility-Aware Fog Computing Platform Configuration for Mixed-Criticality Applications”, Submitted to IEEE Trans-actions on Services Computing, 2021.

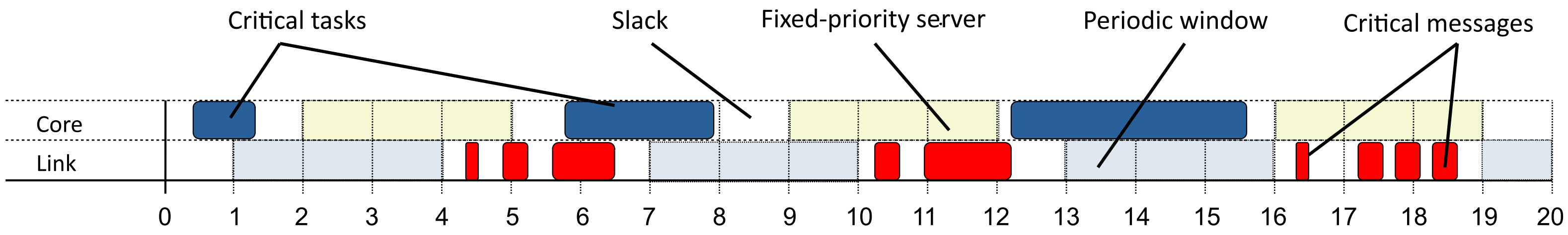
Extensibility in Fog Computing

- **The FCP runs mixed-criticality applications with safety nature, must be certified requiring static pre-release configuration**
  - The vision is Industry 4.0 is to host Fog applications, which are not considered at design time
  - A costly re-certification is required when the configuration is changed
  - An extensible configuration is used in the FCP to realize Industrial IoT
    - Resources for safety critical applications are assigned at design time
    - Remaining resources are allocated for hosting Fog applications at run time

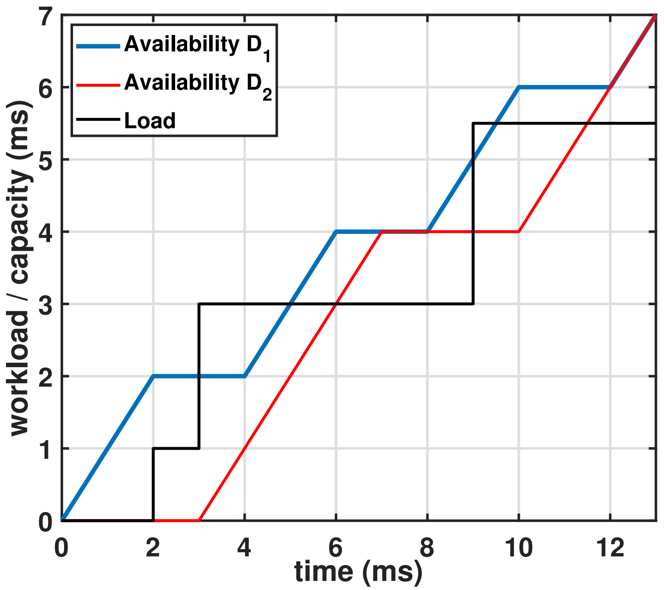


Extensible configuration

- **An extensible FCP configuration is synthesized at design time and considers changes in runtime.**
  - The changes can be future critical applications, Fog applications or both; handled by appropriate technique put together in a hierarchical scheduling model
  - The extensible schedule accommodates a larger number of future control applications and provides a shorter response time for Fog applications
- **The extensible configuration uses the uniform distribution of the short periodic slack in schedules to host the changes.**
  - The slack in the schedules is distributed uniformly at design time
  - The slack is used to allocate resources for dimensioned servers to handle applications



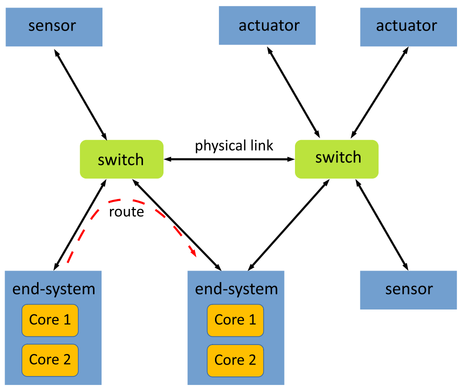
Extensible configuration

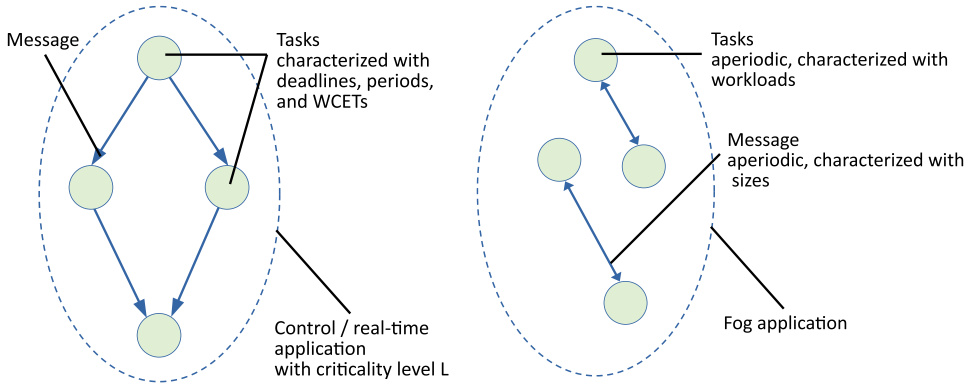


Response time analysis

Problem Formulation

Given: Application and architecture models

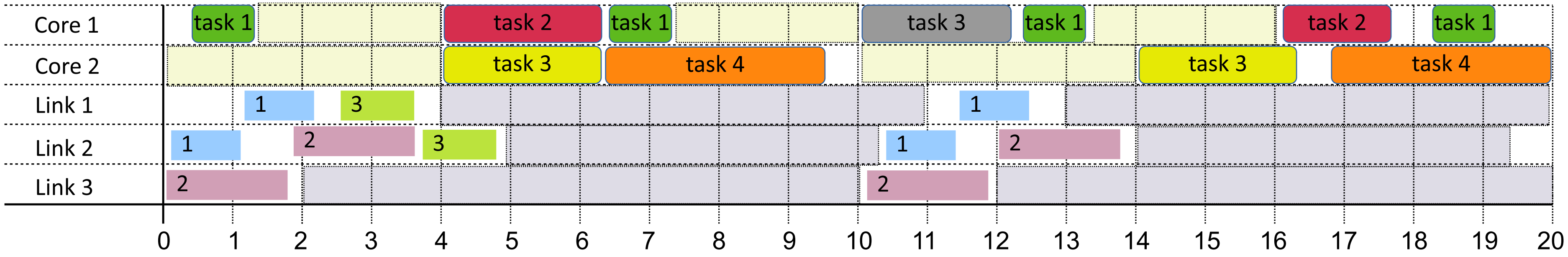




Determine: An FCP configuration:

- mapping of tasks to the cores
- routes for critical control applications
- Gate Control Lists (GCLs)
- static task schedule tables
- dimensioning of deferrable servers
- dimensioning of port windows

Example configuration



Such that:

- The QoC of applications is maximized and “balanced” across applications
- The deadlines are satisfied
- The extensibility of the FCP configuration is maximized

Extensible Configuration Optimization Strategy (ECOS)

Architecture model

Application model

ECOS models the problem as a CP model

- The CP model consists of a set of variables
    - Start time of frames
    - End time of frames
    - Start time of jobs
    - End time of jobs
    - Mapping of tasks to the cores of FNs

ECOS defines a CP formulation for the problem

- The CP formulation consists of a set of constraints
    - Link overlapping
    - Routing
    - Isolation of frames
    - Frame deadlines
    - Core utilization
    - Task overlapping
    - Task deadlines
    - Precedence in applications

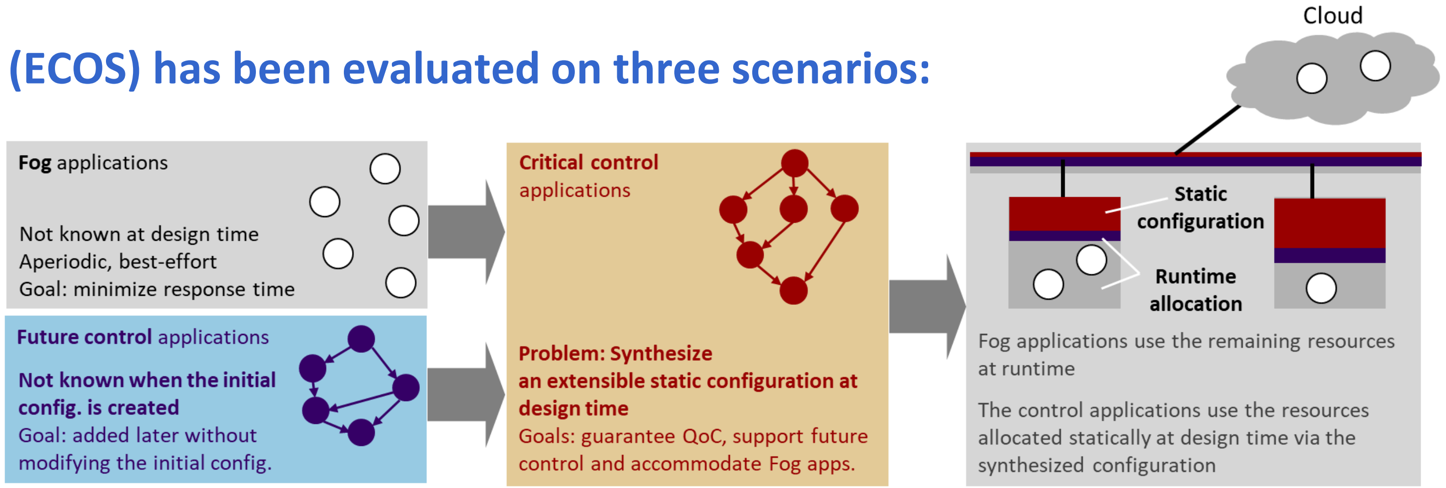
ECOSS generates a set of optimized solutions

- ECOS improves the speed of the search with a metaheuristic search traversal strategy
- The strategy uses heuristic methods for choosing variables and assigning values to the variables

Evaluation results

Extensible Configuration Optimization Strategy (ECOS) has been evaluated on three scenarios:

1. Supporting future control applications
2. Hosting Fog applications
3. Extending with upgrades



TC #	Total no. of tasks / flows in FCCAs[1]	Mean util. of FCCAs[1]	Percentage of Supported FCCAs[1]		RT[2] of Fog application 1 Tasks: 16 Flows: 15		RT[2] of Fog application 2 Tasks: 21 Flows: 20		RT[2] of Fog application 3 Tasks: 35 Flows: 38	
			ECOS	ECOS/E	ECOS	ECOS/E	ECOS	ECOS/E	ECOS	ECOS/E
1	36/58	57%	100%	78%	1.33	3.97	2.82	5.66	4.73	7.43
2	37/65	55%	100%	89%	1.42	1.91	2.75	4.92	6.74	9.56
3	30/20	50%	100%	96%	1.26	3.36	2.94	4.88	5.16	12.29
4	29/32	45%	100%	81%	2.17	3.27	3.64	5.65	5.18	7.34
5	33/40	44%	100%	96%	1.56	4.14	3.68	5.81	4.36	9.44
6	44/45	40%	100%	90%	1.47	2.96	3.14	4.17	4.96	5.82
7	37/35	39%	100%	83%	1.19	3.95	3.88	3.96	2.96	4.76
8	33/34	37%	100%	90%	2.18	2.93	3.14	5.84	2.84	9.64
9	25/28	31%	100%	98%	1.65	3.77	4.43	5.93	2.35	4.74
10	18/21	27%	100%	82%	2.37	3.79	4.82	5.97	2.85	6.68
Average			100%	87%	1.6	3.4	3.5	5.2	4.2	7.7

[1] Future Critical Control Application
[2] Response time in ms.

Contributions of Paper C

- **Main contributions:**
  - Considering both real-time control and dynamic Fog applications
  - Considering both tasks and messages in a TSN-based Fog Computing platform for QoC
  - Formulating and solving an optimization problem related to the extensibility of the Fog platform

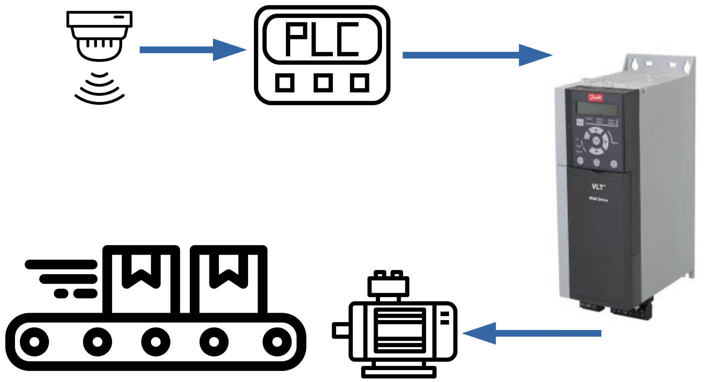
Paper D

Electric Drives as Fog Nodes in a Fog Computing-based Industrial Use Case

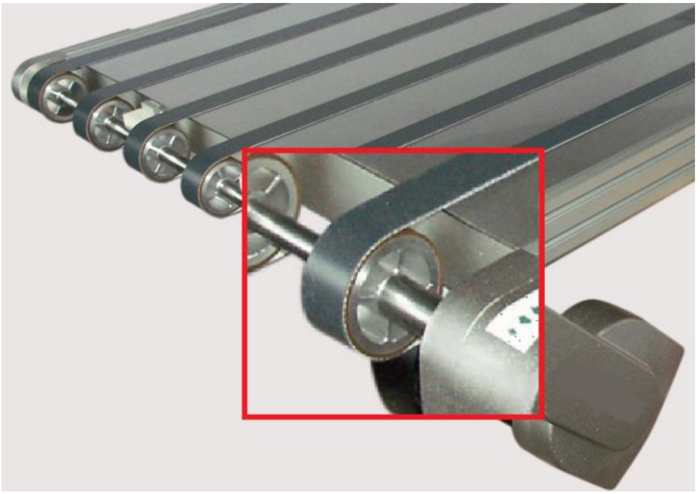
M. Barzegaran, N. Desai, J. Qian and P. Pop, “Electric Drives as Fog Nodes in a Fog Computing-based Industrial Use Case,” Submitted to IET Journal of Engineering, 2021.

Fog-based electric drives

- **Electric drives are widely used in Industry**
  - They produce data that carries vital information: they can be used as the data source
- **Electric drives are re-engineered as FNs in an FCP; that is called “fogification”**
  - Fogified drives perform data analytics; avoids sending massive data with vital information
  - Electric drives are fogified considering the FORA FCP reference architecture
  - Fogified drives are modelled using Architecture Analysis & Design Language (AADL)



An electric drive in a conveyor belt setting

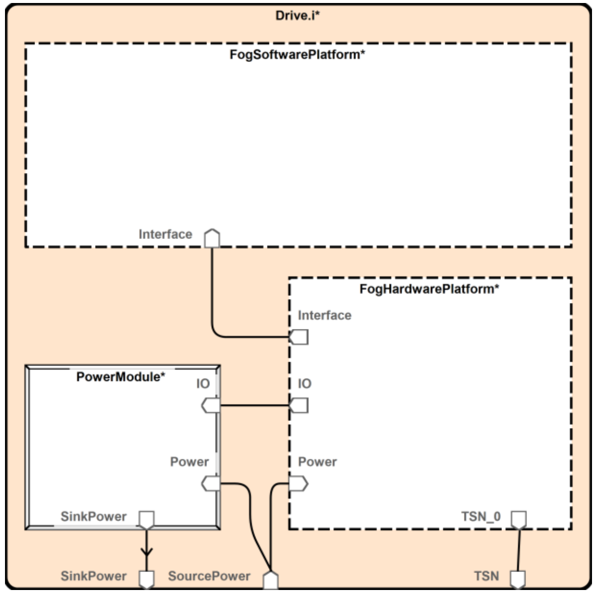


Conveyor belt

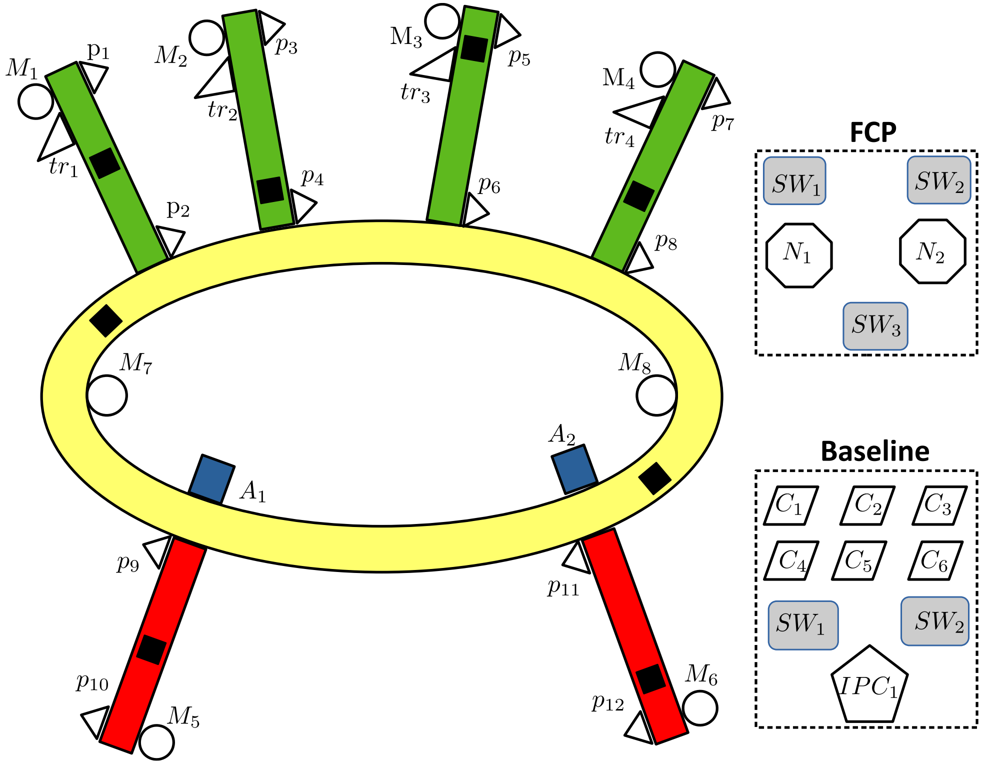
Use case architecture and model

Fogified drives are modelled using Architecture Analysis and Design Language (AADL)

- AADL is a standard language for modelling systems using a component-oriented approach
- Fogified drives are designed to deliver the drive punctualities and realize the vision of Industry 4.0



AADL model of the fogified drive



UC schematics

A self baggage drop system

Evaluation results

Fogified drives have been evaluated using several Key Performance Indicators (KPIs)

- **Safety:** Introducing partitioning for isolation of mixed-criticality applications and evaluating its overhead
- **Security:** Provision of authentication mechanisms for communication to protect high-criticality applications
- **Performance of the virtualized control:** implementing the control applications and maximizing their QoC
- **Hardware cost:** evaluating the hardware cost reduction of a fog-based implementation
- **Data analytics:** provision and evaluation of a decentralized Fog-based machine learning solution

Note: see the thesis for the numerical evaluation details.

Contributions of Paper D

- **Main contributions:**
  - Identifying and modeling with AADL a realistic use case using the FORA Fog Computing Platform
  - Definition of realistic requirements and KPIs related to the use case
  - Evaluating the appropriateness of a Fog-based solution for the use case from several perspectives

Summary

- We proposed several approaches to the design time FCP configuration optimization for mixed-criticality applications
  - The configuration guarantees the performance and timeliness of control applications
  - The configuration provides maximum Quality-of-Service for dynamic Fog applications
  - The configuration consists of:
    - Decisions on the partitions that provide temporal and spatial isolation among mixed-criticality applications
    - Mapping the tasks to the cores of multicore Fog nodes
    - Routing of flows on TSN
    - Synthesizing the task schedule tables and GCLs
- We propose approaches to handle migration and best-effort scheduling of dynamic Fog applications at runtime
- We have developed several algorithms that use heuristics, metaheuristics and Constraint Programming to solve these combinatorial optimization problems
- We have proposed analytical models for QoC and extensibility that can be integrated to optimization strategies
- We have evaluated the algorithms on several test cases

Research output and list of publications

- Publications
  - 12 peer-reviewed articles: 8 first author, 5 journal papers (2 under review)
  - 1 technical report
- Coordinated the contributions to the FORA reference architecture using the Architecture Analysis & Design Language
- Developed an Industry 4.0 demonstrator

Journal articles

- **M. Barzegaran**, A. Cervin, and P. Pop. "***Performance Optimization of Control Applications on Fog Computing Platforms Using Scheduling and Isolation***," IEEE Access, vol. 8, pp. 104085-104098, 2020.
- P. Paul, B. Zarrin, **M. Barzegaran**, S. Schulte, S. Punnekkat, J. Ruh, and W.Steiner, "***The FORA Fog Computing Platform for Industrial IoT***," In Information Systems, Elsevier, vol. 98, pp. 101727, 2021.
- **M. Barzegaran** and P. Pop, "***Communication Scheduling for Control Performance in TSN-Based Fog Computing Platforms***," In IEEE Access, vol. 9, pp.50782-50797, 2021.

Research output and list of publications

Conference and workshops

- **M. Barzegaran**, A. Cervin, and P. Pop, "***Towards quality-of-control-aware scheduling of industrial applications on fog computing platforms***," In Proceeding of the Workshop on Fog Computing and the IoT. ACM, pp. 1–5, 2019.
- A. Cervin, P. Pazzaglia, **M. Barzegaran**, and R. Mahfouzi, "***Using JitterTime to analyze transient performance in adaptive and reconfigurable control systems***," In Proceeding of IEEE International Conference on Emerging Technologies and Factory Automation. pp. 1025-1032, 2019.
- **M. Barzegaran**, N. Desai, J. Qian, K. Tange, B. Zarrin, P. Pop and J. Kuusela."***Fogification of electric drives: An industrial use case***," In Proceeding of IEEE International Conference on Emerging Technologies and Factory Automation. Vol. 1, pp. 77-84, 2020.
- **M. Barzegaran**, B. Zarrin, and P. Pop. "***Quality-of-control-aware scheduling of communication in TSN-based fog computing platforms using constraint programming***," In Workshop on Fog Computing and the IoT, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, pp. 1-4, 2020.
- **M. Barzegaran**, V. Karagiannis, C. Avasalcai, P. Pop, S. Schulte and S. Dustdar,"***Towards Extensibility-Aware Scheduling of Industrial Applications on Fog Nodes***," In Proceeding of IEEE International Conference on Emerging Technologies and Factory Automation, pp. 67-75, 2020.
- I. Murturi, **M. Barzegaran** and S. Dustdar, "***A Decentralized Approach for Determining Configurator Placement in Dynamic Edge Networks***," In Proceeding of IEEE International Conference on Cognitive Machine Intelligence, pp.147-156, 2020.

Research output and list of publications

Other

- **M. Barzegaran**, N. Reusch, L. Zhao, S. Craciunas, and P. Pop. ”***Real-Time Guarantees for Critical Traffic in IEEE 802.1Qbv TSN Networks with Un-scheduled End-Systems***,” In arXiv, 2021.
- **M. Barzegaran** and P. Pop. “***Extensibility-Aware Fog Computing Platform Configuration for Mixed-Criticality Applications***”, Submitted to IEEE Transactions on Services Computing, 2021.
- **M. Barzegaran**, N. Desai, J. Qian and P. Pop, “***Electric Drives as Fog Nodes in a Fog Computing-based Industrial Use Case***,” Submitted to IET Journal of Engineering, 2021.
- J. Qian, **M. Barzegaran**, and P. Pop “***Decomposing Deep Training Solutions on Fog Computing Platforms***,” To be submitted to ACM/IEEE Symposium on Edge Computing, 2021

Thank you for your attention