Task- and Network-level Schedule Co-Synthesis of Ethernet-based Time-triggered Systems

Licong Zhang[1], Dip Goswami[2], Reinhard Schneider[1], Samarjit Chakraborty[1]

[1] *Institute for Real-Time Computer Systems, TU Munich, Germany*

[2] *Electrical Engineering Department, TU Eindhoven,* Netherlands

ASP-DAC 2014
Jan. 21, 2014

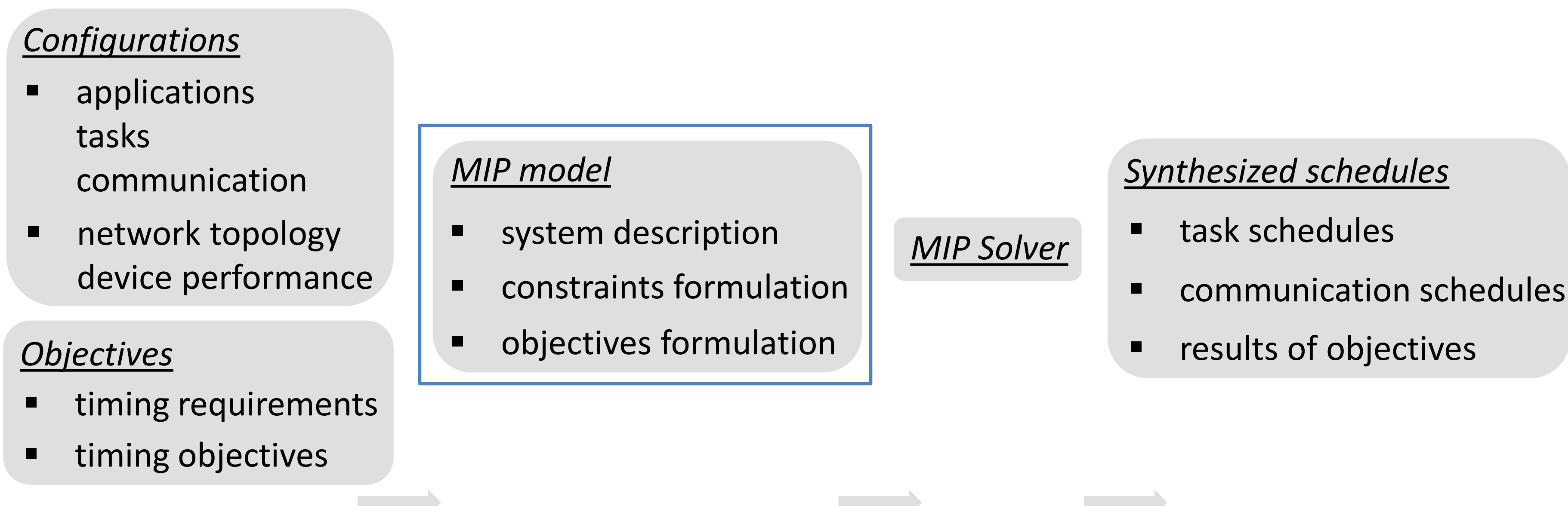
Technische Universität München

Overview

- **Problem**
  - Ethernet-based time-triggered system
  - Co-synthesis of task and communication schedule
  - Application-level (multi-)objectives

Overview

- **Problem**
  - Ethernet-based time-triggered system
  - Co-synthesis of task and communication schedule
  - Application-level (multi-)objectives
- **Approach**
  - Formulation of the problem in Mixed Integer Programming model
    - ➢ System description, constraints and objectives formulation



Outline

- Motivation
- Ethernet-based Time-triggered System
- Constraints Formulation
- Multi-objective Optimization
- Experimental Results
- Concluding Remarks

Motivation

- **Ethernet in safety-critical domain**
  - Safety-critical domains: avionics, automotive, industrial automation
  - Increased complexity and load on communication
  - Conventional buses reaching limits (e.g. CAN, FlexRay in automotive)
  - Progress in Ethernet offers better determinism and QoS

Motivation

- **Ethernet in safety-critical domain**
  - Safety-critical domains: avionics, automotive, industrial automation
  - Increased complexity and load on communication
  - Conventional buses reaching limits (e.g. CAN, FlexRay in automotive)
  - Progress in Ethernet offers better determinism and QoS
- **Need for performance guarantees**
  - Safety-critical applications (e.g. vehicle/plane dynamics control)
  - Need for ultra-low latency, jitter and determinism

Motivation

- **Ethernet in safety-critical domain**
  - Safety-critical domains: avionics, automotive, industrial automation
  - Increased complexity and load on communication
  - Conventional buses reaching limits (e.g. CAN, FlexRay in automotive)
  - Progress in Ethernet offers better determinism and QoS

- **Need for performance guarantees**
  - Safety-critical applications (e.g. vehicle/plane dynamics control)
  - Need for ultra-low latency, jitter and determinism

- **Time-triggered systems**
  - Offers determinism
  - Schedules can be synthesized to minimize latency

Motivation

- **Task- and communication-level schedule co-synthesis**
  - Application-level timing more important (e.g. feedback control loop)
  - Schedules of tasks and communication must be synchronized
  - Separate task or communication schedule synthesis

    -> not leading to optimal application-level timing properties

Motivation

- **Task- and communication-level schedule co-synthesis**
  - Application-level timing more important (e.g. feedback control loop)
  - Schedules of tasks and communication must be synchronized
  - Separate task or communication schedule synthesis

    -> not leading to optimal application-level timing properties

- **Related work**
  - On general time-triggered architecture [6]
  - Schedule synthesis of FlexRay-based time-triggered system[7,8,9]
  - Communication schedule synthesis of time-triggered Ethernet [10,11,12,13]

Motivation

- **Task- and communication-level schedule co-synthesis**
  - Application-level timing more important (e.g. feedback control loop)
  - Schedules of tasks and communication must be synchronized
  - Separate task or communication schedule synthesis

    -> not leading to optimal application-level timing properties

- **Related work**
  - On general time-triggered architecture [6]
  - Schedule synthesis of FlexRay-based time-triggered system[7,8,9]
  - Communication schedule synthesis of time-triggered Ethernet [10,11,12,13]

- **Contributions**
  - Task and communication schedule co-synthesis in Ethernet-based time-triggered system (problem formulation in Mixed Integer Programming)
  - Multi-objective optimization according to application-level objectives

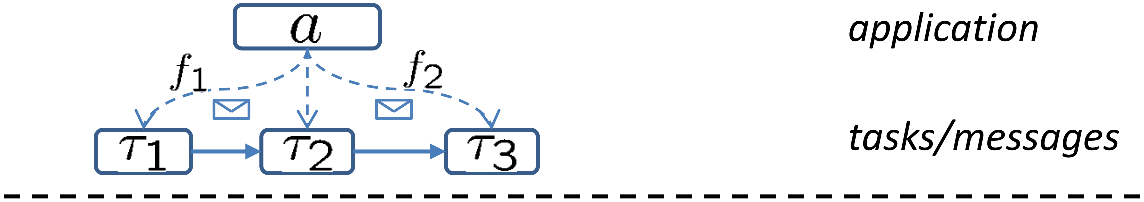
Time-triggered Distributed System

- **Distributed system**
  - Task partition and mapping onto different processing units
  - Data sent through a network (e.g. CAN, Ethernet)
  - Application-level timing -> interplay between tasks and communication

a *application*

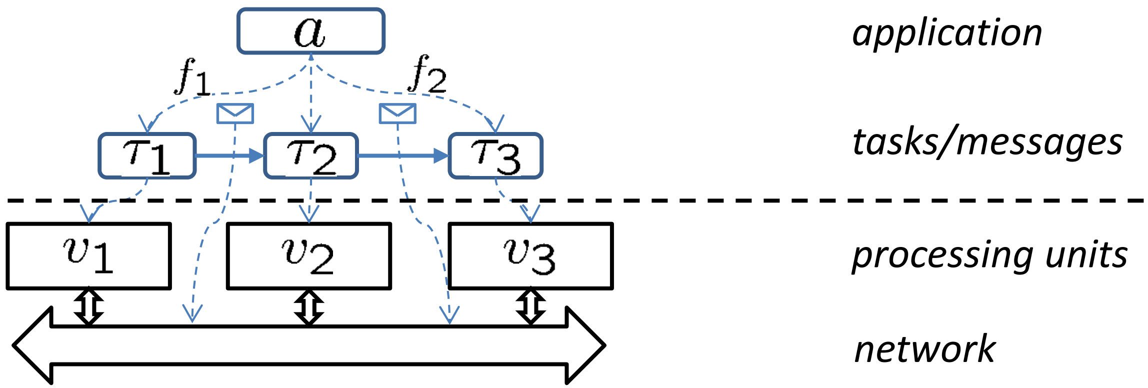
Time-triggered Distributed System

- **Distributed system**
  - Task partition and mapping onto different processing units
  - Data sent through a network (e.g. CAN, Ethernet)
  - Application-level timing -> interplay between tasks and communication



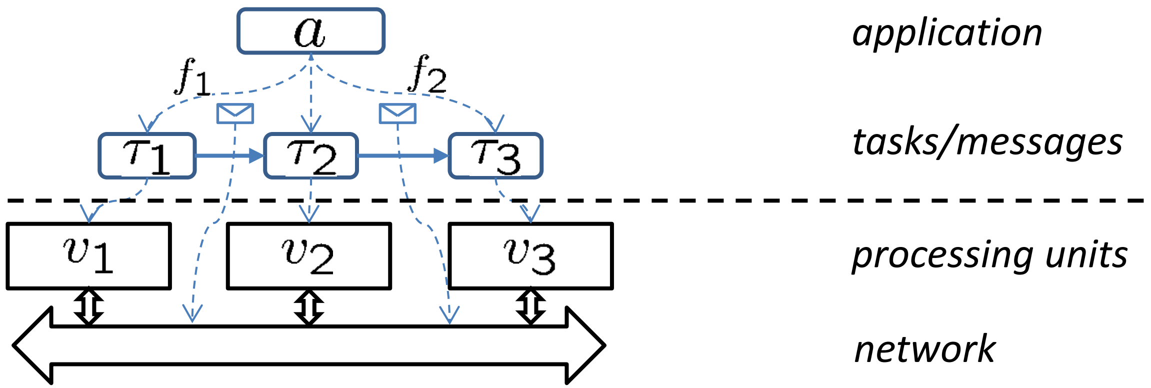
Time-triggered Distributed System

- **Distributed system**
  - Task partition and mapping onto different processing units
  - Data sent through a network (e.g. CAN, Ethernet)
  - Application-level timing -> interplay between tasks and communication



Time-triggered Distributed System

- **Distributed system**
  - Task partition and mapping onto different processing units
  - Data sent through a network (e.g. CAN, Ethernet)
  - Application-level timing -> interplay between tasks and communication

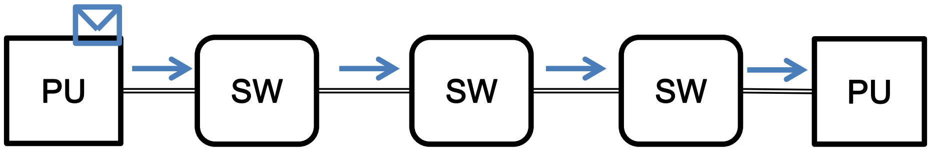


- **Time-triggered non-preemptive task scheduling**
  - Pre-defined static schedule / a task can not be preempted (e.g. eCos)

- **Time-triggered communication scheduling**
  - Pre-defined static schedule for message transmission (e.g. FlexRay static seg., TTP)

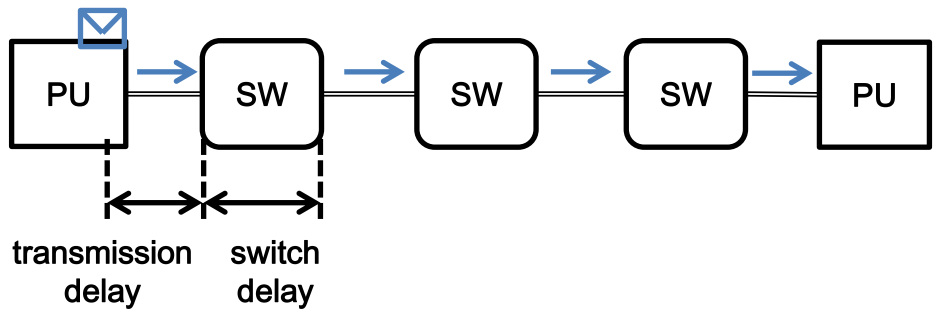
Time-triggered Ethernet Communication

- **Switched Ethernet**
  - Processing units connected through switches
  - Commonly with full-duplex links
  - Ethernet frames forwarded switch by switch



Time-triggered Ethernet Communication

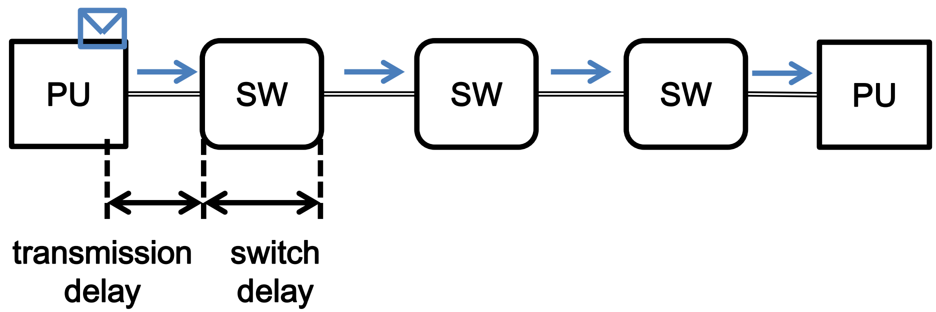
- **Switched Ethernet**
  - Processing units connected through switches
  - Commonly with full-duplex links
  - Ethernet frames forwarded switch by switch



- **Network latency**
  - Propagation delay (negligible)
  - Transmission delay
  - Switch delay

Time-triggered Ethernet Communication

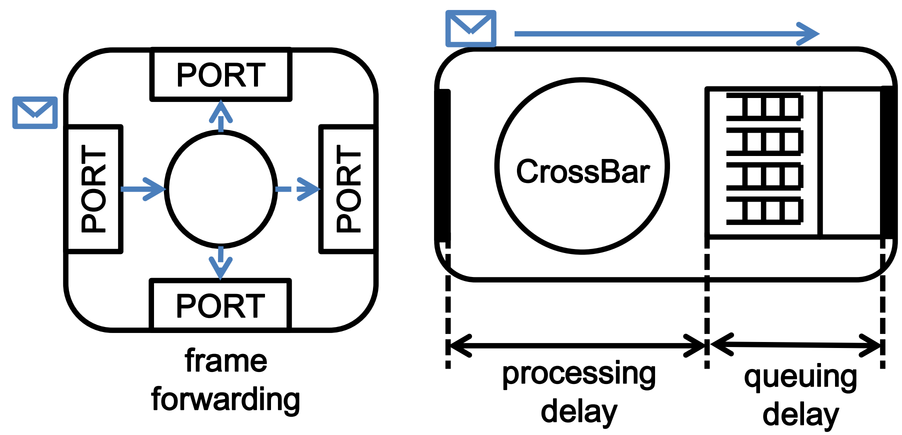
- **Switched Ethernet**
  - Processing units connected through switches
  - Commonly with full-duplex links
  - Ethernet frames forwarded switch by switch



- **Network latency**
  - Propagation delay (negligible)
  - Transmission delay
  - Switch delay
    - Processing delay
    - <u>Queuing delay</u>

      -> not deterministic

      -> can be relatively large

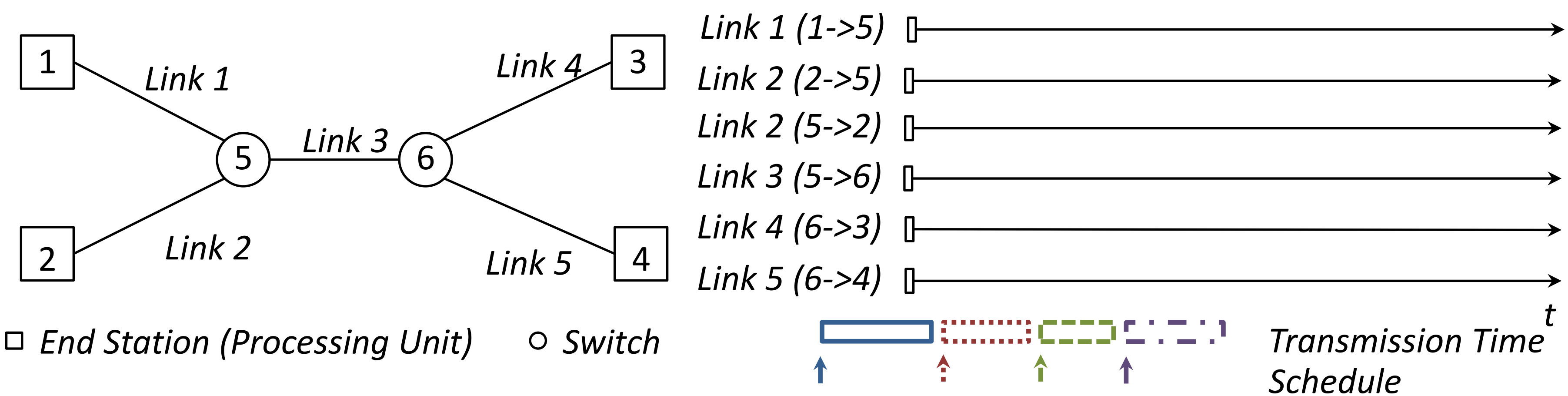


Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule

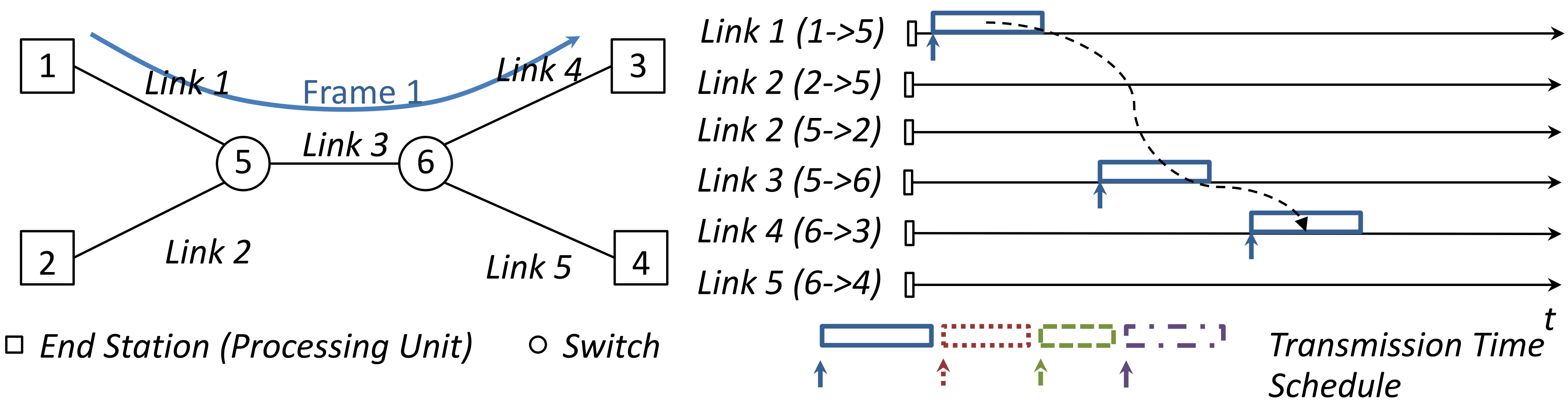
Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule



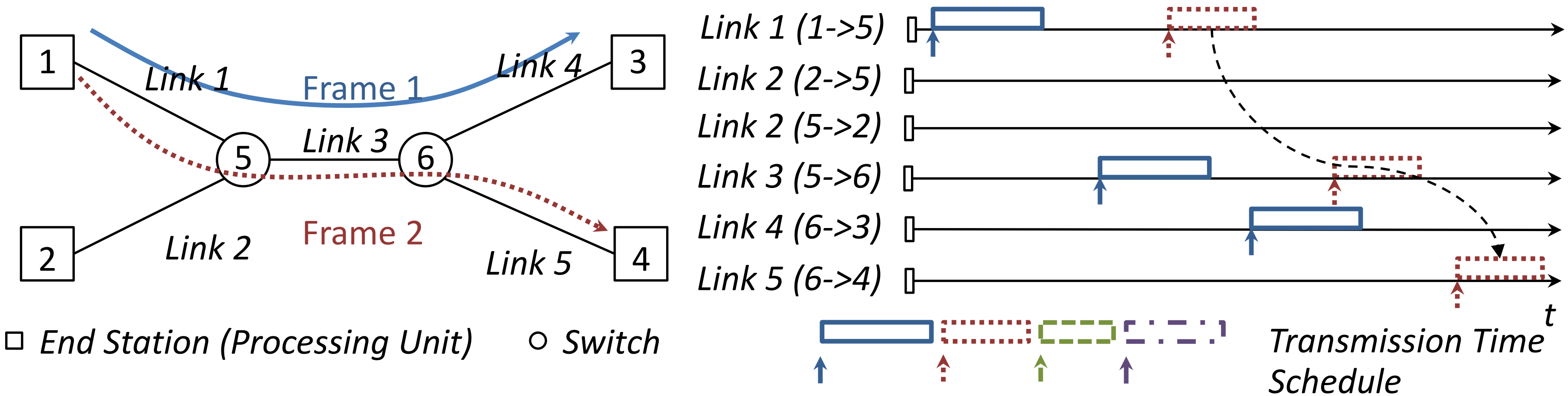
Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule



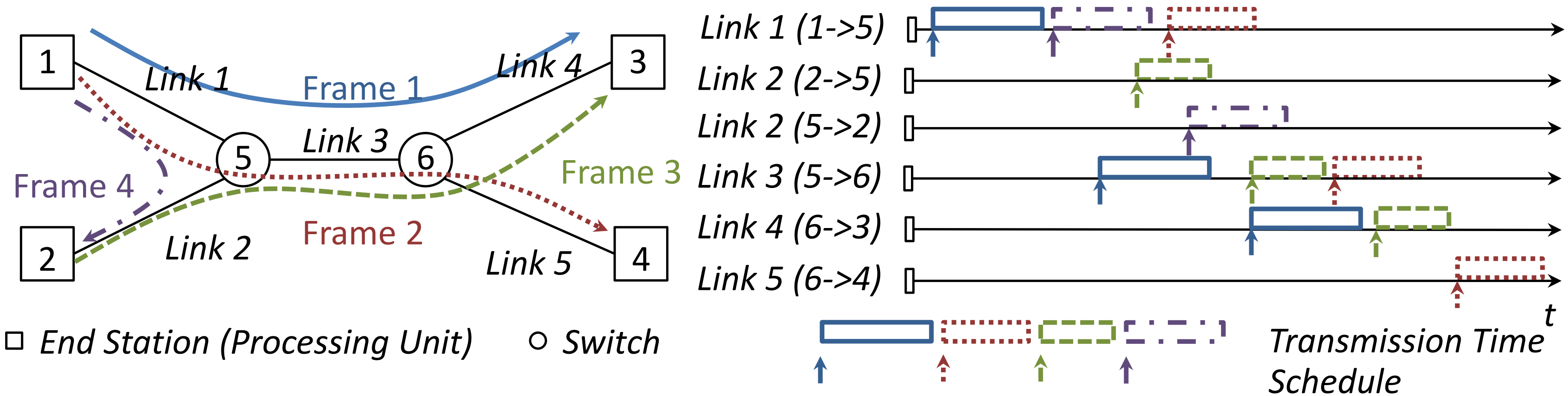
Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule



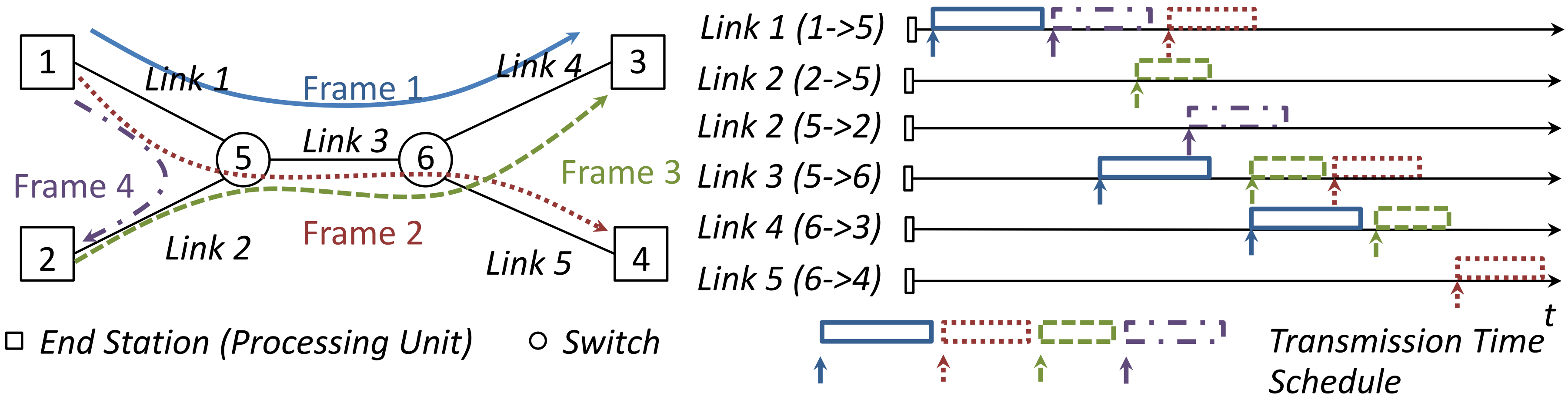
Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule



Time-triggered Ethernet Communication

- **Time-triggered Ethernet communication**
  - Frames are scheduled to avoid queuing delay
  - Frames are not queued at the output port
  - Frame transmission on each link according to static schedule



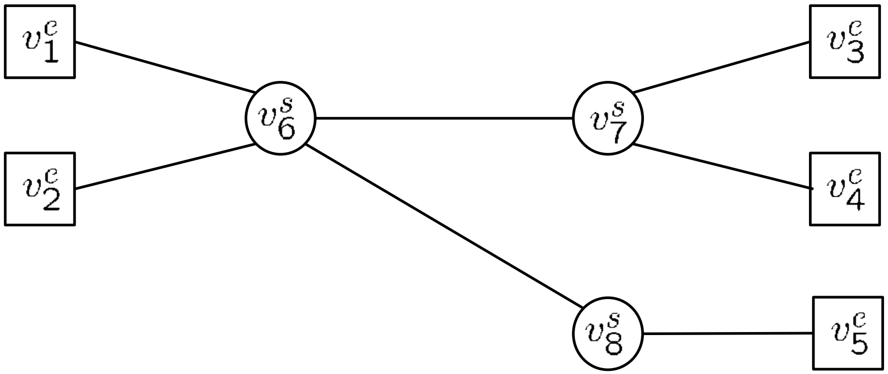
- **Ethernet-based protocols with time-triggered traffic**
  - Profinet IRT [1]
  - Time-triggered traffic in TT Ethernet [3]
  - IEEE802.1Qbv (not yet released) [5]

Problem Formulation

- **Topology** $G(\mathcal{V}, \mathcal{E})$

  $v_i \in \mathcal{V}$ ⟶ *processing units or switches*

  $l_{m,n} \in \mathcal{E}$ ⟶ *Ethernet links*



Problem Formulation

- **Topology** $G(\mathcal{V}, \mathcal{E})$

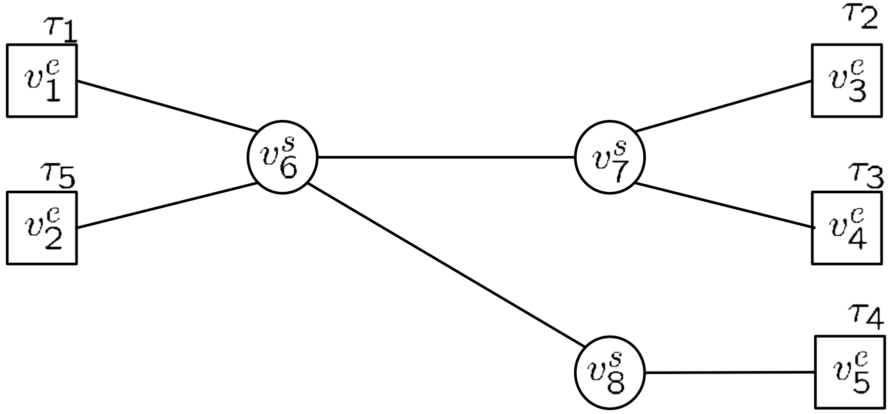
  $v_i \in \mathcal{V}$ ⟶ *processing units or switches*

  $l_{m,n} \in \mathcal{E}$ ⟶ *Ethernet links*

- **Application task** $\mathcal{T}$

  $\tau_i = \{\tau_i.p, \quad \tau_i.o, \quad \tau_i.e\}$

  period, offset, WCET



Problem Formulation

- **Topology** $G(\mathcal{V}, \mathcal{E})$

  $v_i \in \mathcal{V} \longrightarrow$ *processing units or switches*

  $l_{m,n} \in \mathcal{E} \longrightarrow$ *Ethernet links*

- **Application task** $\mathcal{T}$

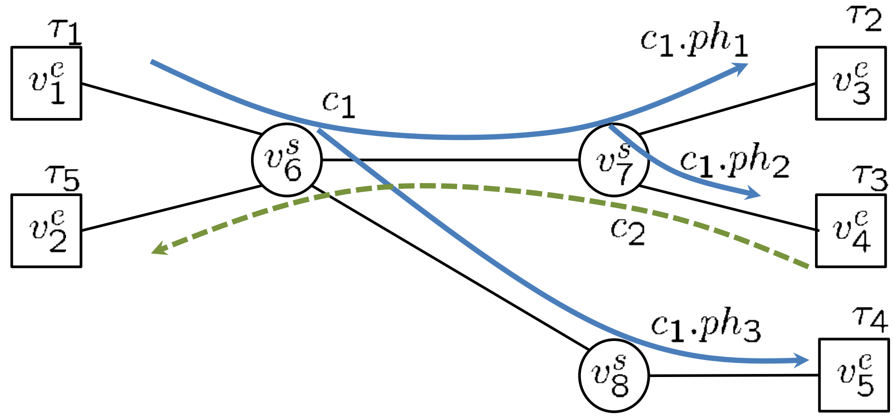
  $\tau_i = \{\tau_i.p, \quad \tau_i.o, \quad \tau_i.e\}$

  ↓ *period,* ↓ *offset,* ↓ *WCET*

- **Communication task** $\mathcal{C}$

  $c_i = \{f_i, \quad c_i.tr, \quad c_i.o, \quad c_i.p\}$

  ↓ *frame,* ↓ *path tree,* ↓ *offsets,* ↓ *period*



Problem Formulation

- **Topology** $G(\mathcal{V}, \mathcal{E})$

  $v_i \in \mathcal{V}$ $\longrightarrow$ *processing units or switches*

  $l_{m,n} \in \mathcal{E}$ $\longrightarrow$ *Ethernet links*

- **Application task** $\mathcal{T}$

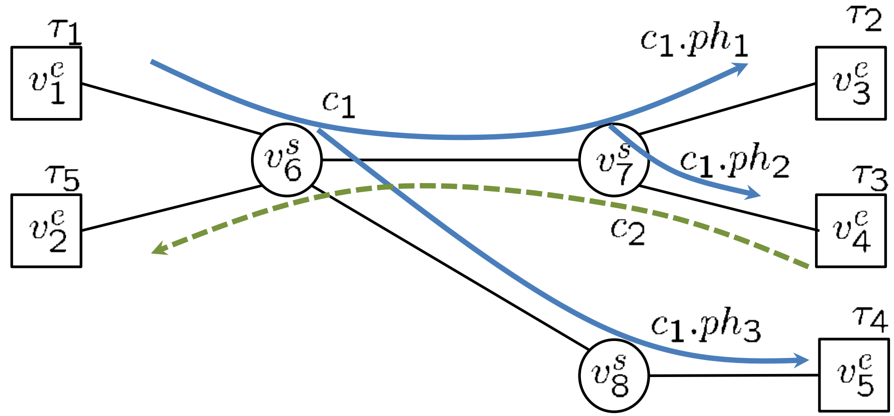
  $\tau_i = \{\tau_i.p, \quad \tau_i.o, \quad \tau_i.e\}$

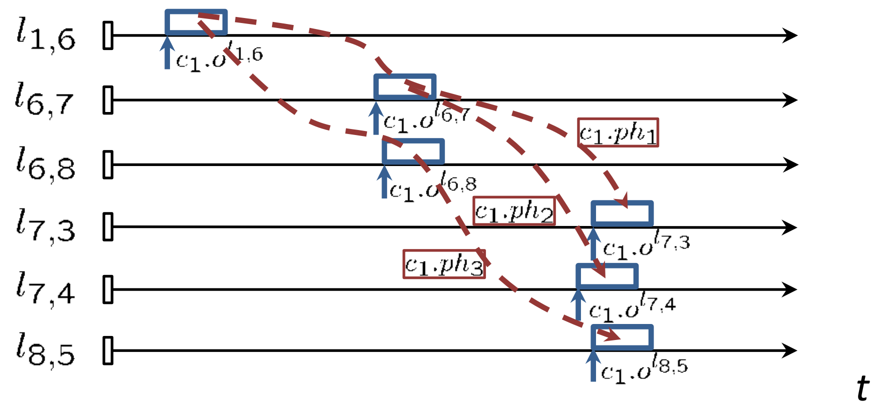
  period, offset, WCET

- **Communication task** c

  $c_i = \{f_i, \quad c_i.tr, \quad c_i.o, \quad c_i.p\}$

  frame, path tree, offsets, period





path $\longleftarrow$ $c_i.ph_j$

from sender to one receiver

path tree $\longleftarrow$ $c_i.tr = \{c_i.ph_1, c_i.ph_2 \ldots\}$

all paths in a communication task

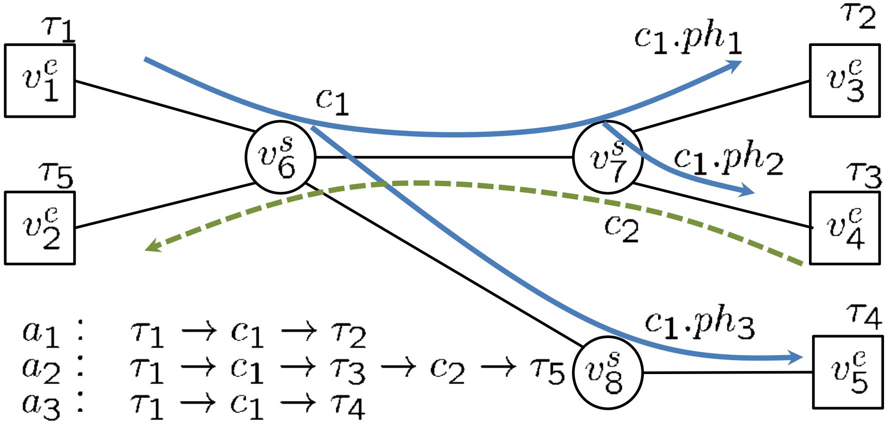
Problem Formulation

- **Application** a

$$a_i = \{a_i.tc,\ \ a_i.p,\ \ a_i.rt,\ \ a_i.lz\}$$

task chain, period

response time, latency



Problem Formulation

- **Application** a

$$a_i = \{a_i.tc,\ a_i.p,\ a_i.rt,\ a_i.lz\}$$

task chain, period

response time, latency

task chain $\leftarrow$ $a_i.tc$

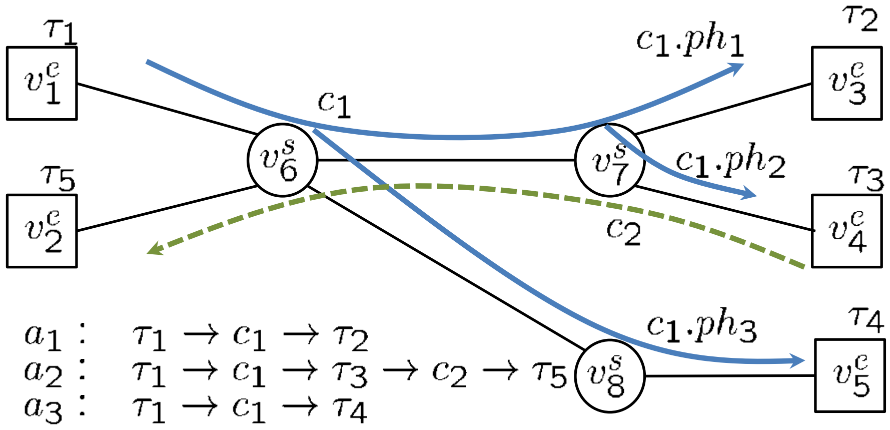
all application and communication tasks in temporal order

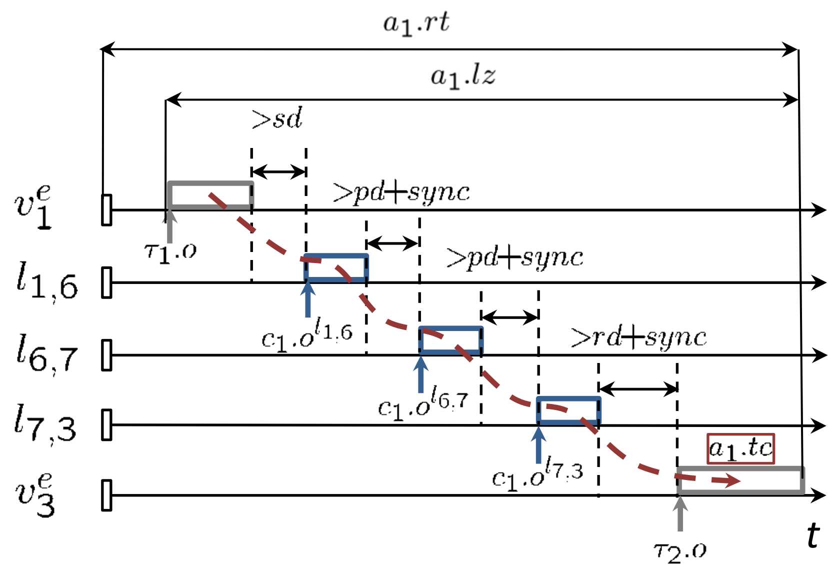
response time $\leftarrow$ $a_i.rt$

time from period begin to the end of last task in task chain

end-to-end latency $\leftarrow$ $a_i.lz$

time from begin of first task to the end of last task in task chain





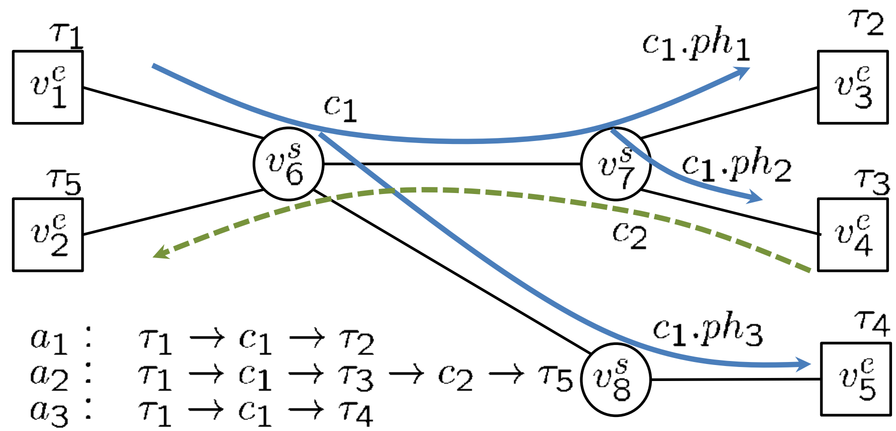
Problem Formulation

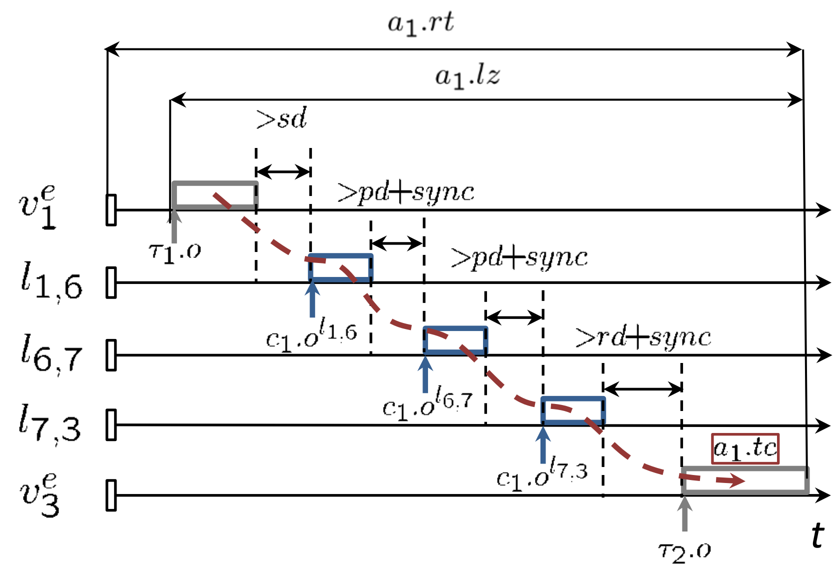
- **Application** a

$$a_i = \{a_i.tc, \quad a_i.p, \quad a_i.rt, \quad a_i.lz\}$$

task chain, period

response time, latency





- **Schedule co-synthesis problem**
  - To co-synthesize
    - task schedules $\{\tau_i.o\}$
    - communication schedules $\{c_i.o\}$

  according to **application-level objectives** (e.g. end-to-end latency, response time)

Mixed Integer Programming (MIP)

- **Mixed Integer (Linear) Programming :**

minimize	$c^T x$
subject to	$Ax \leq b$
	$lb \leq x \leq ub$
	some variables in x must take integer values

- **Model formulation**
  - Formulate system constraints of the co-synthesis problem into a MIP problem

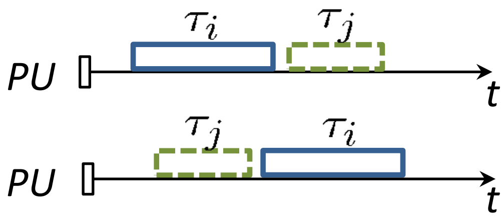
Constraints

- **(C1) Collision-free application tasks**
  - *no overlap between execution of two instances of tasks*

$$\tau_i.p \times k_i + \tau_i.o + \tau_i.e < \tau_j.p \times k_j + \tau_j.o$$

or

$$\underbrace{\tau_j.p \times k_j + \tau_j.o + \tau_j.e}_{\text{end of } \tau_j} < \underbrace{\tau_i.p \times k_i + \tau_i.o}_{\text{begin of } \tau_i}$$



k_i, k_j ⟶ *enumerate instances if periods are not equal*

Constraints

- **(C1) Collision-free application tasks**
  - *no overlap between execution of two instances of tasks*

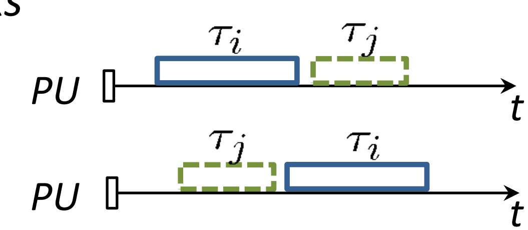
$$\tau_i.p \times k_i + \tau_i.o + \tau_i.e < \tau_j.p \times k_j + \tau_j.o$$

or

$$\tau_j.p \times k_j + \tau_j.o + \tau_j.e < \tau_i.p \times k_i + \tau_i.o$$

end of τ_j *begin of* τ_i

$k_i, k_j \longrightarrow$ *enumerate instances if periods are not equal*



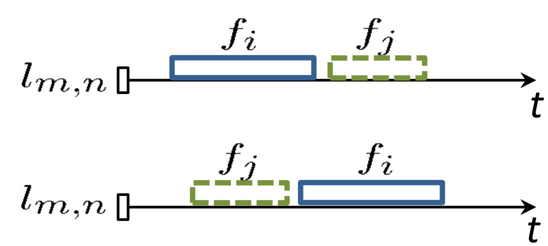
- **(C2) Collision-free communication tasks**
  - *no overlap between transmission of two frames*

$$c_i.p \times k_i + c_i.o^{l_{m,n}} + f_i.fl/bw + ifg < c_j.p \times k_j + c_j.o^{l_{m,n}}$$

or

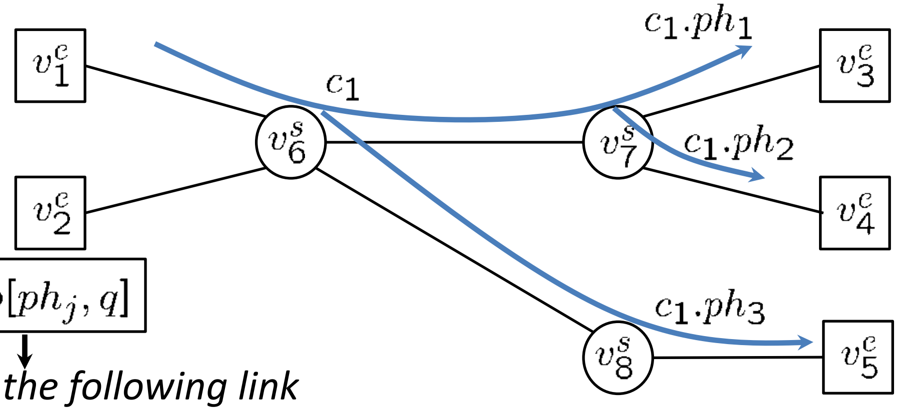
$$c_j.p \times k_j + c_j.o^{l_{m,n}} + f_j.fl/bw + ifg < c_i.p \times k_i + c_i.o^{l_{m,n}}$$

end of f_j *Inter-frame gap* *begin of* f_i



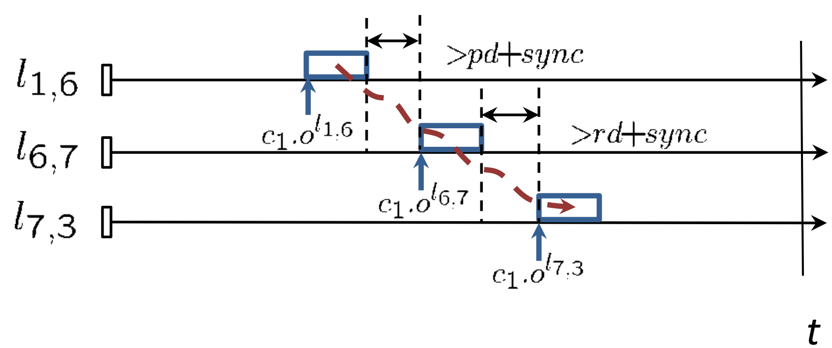
Constraints

- **(C3) Path dependency**
  - Communication schedules
    - -> *correct temporal order in the path*



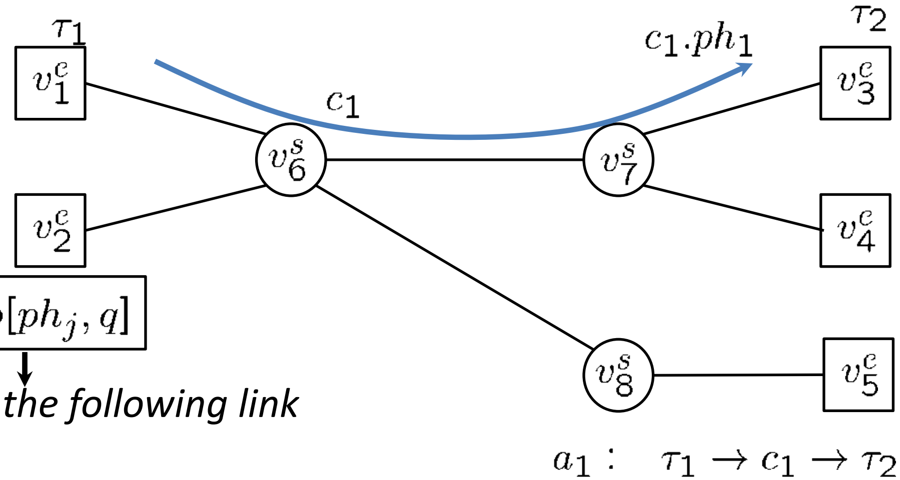
$$c_i.o[ph_i, q-1] + f_i.fl/bw + pd + sync < c_i.o[ph_j, q]$$

schedule on one link — *schedule on the following link*



Constraints

- **(C3) Path dependency**
  - Communication schedules
    -> correct temporal order in the path



$$c_i.o[ph_i, q-1] + f_i.fl/bw + pd + sync < c_i.o[ph_j, q]$$

schedule on one link — *schedule on the following link*

- **(C4) Data dependency**
  - task and communication schedules
    -> correct temporal order in task chain

if τ_i *followed by* τ_j

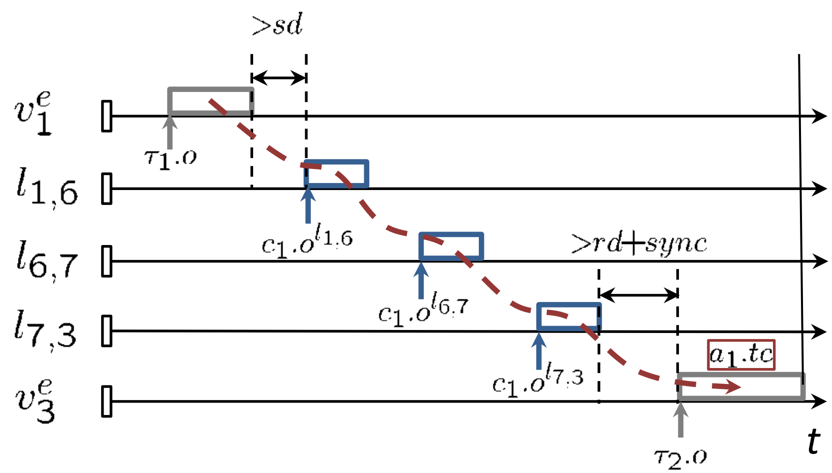
$$\tau_i.o + \tau_i.e < \tau_j.o$$

if τ_i *followed by* c_j

$$\tau_i.o + \tau_i.e + sd < c_j.o[first]$$

if c_i *followed by* τ_j

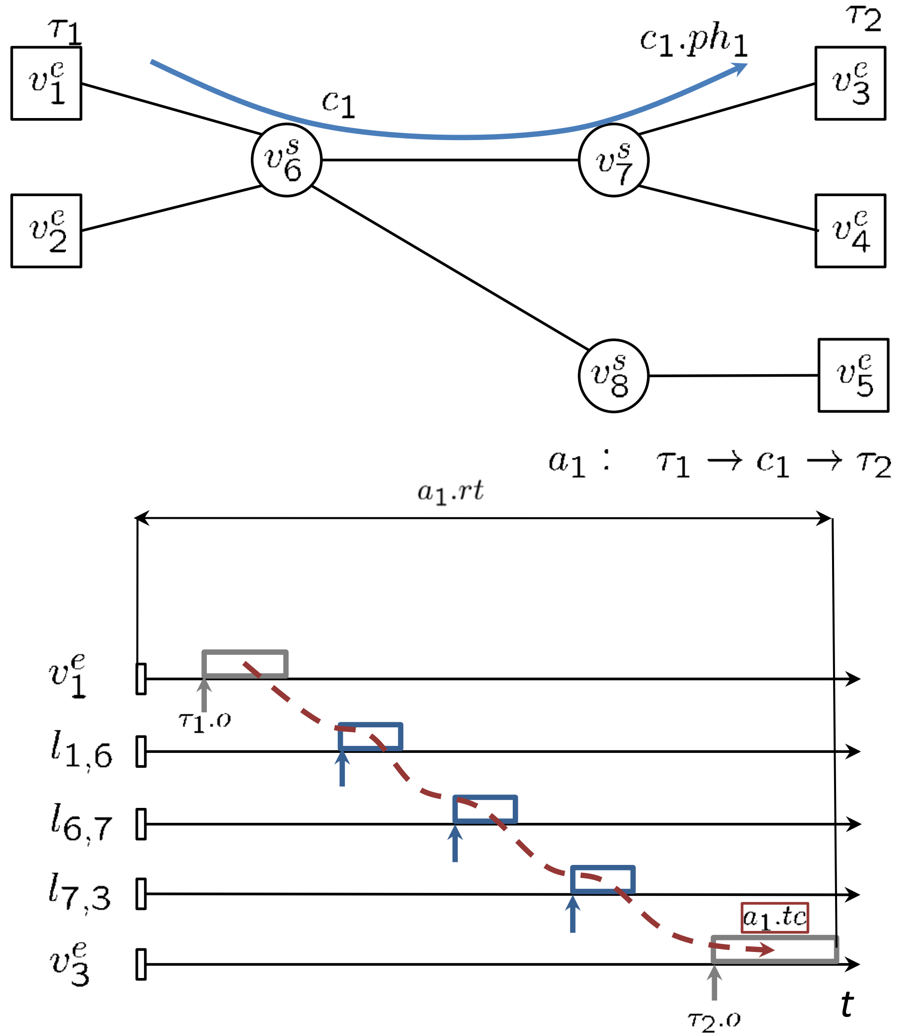
$$c_i.o[last] + f_i.fl/bw + sync + rd < \tau_j.o$$



Constraints

- **(C5) Application response time**
  - *Response time < upper bound*

$$a_i.rt < a_i.rt_{max}$$



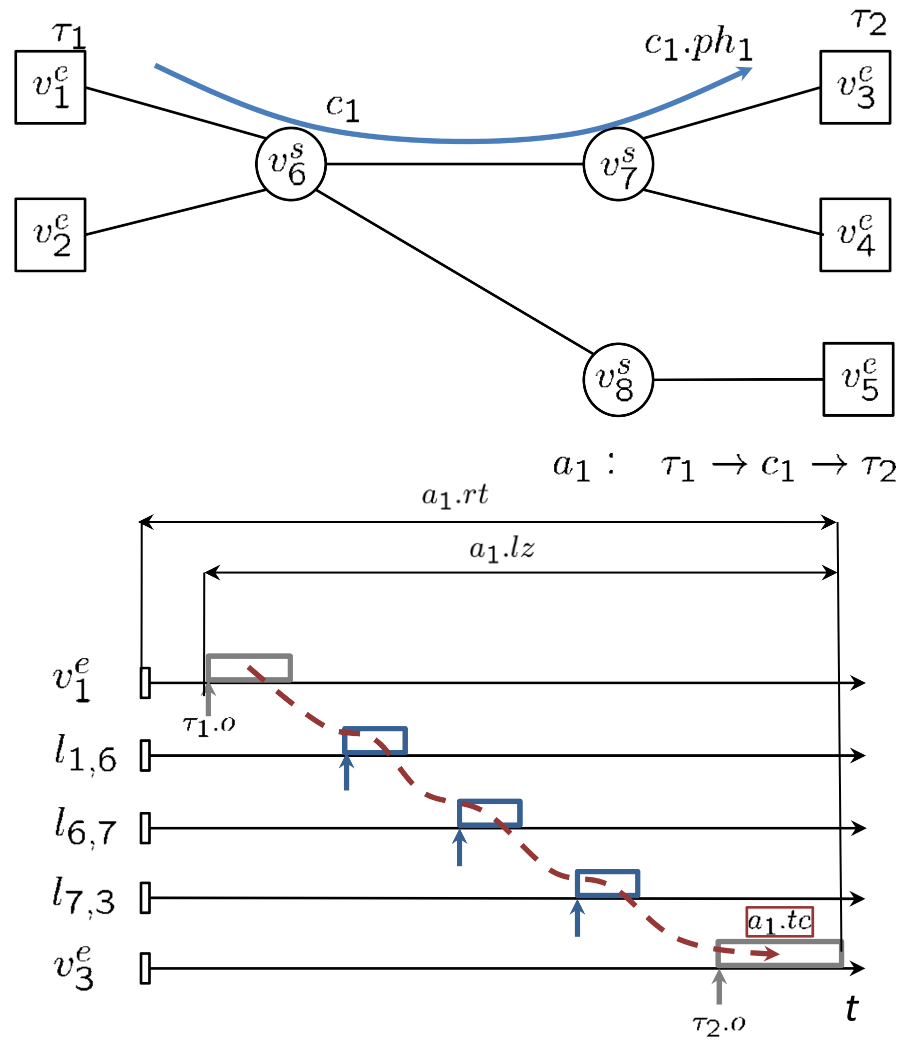
Constraints

- **(C5) Application response time**
  - *Response time < upper bound*

$$a_i.rt < a_i.rt_{max}$$

- **(C6) Application end-to-end latency**
  - *End-to-end latency < upper bound*

$$a_i.lz < a_i.lz_{max}$$



Multi-Objective Optimization

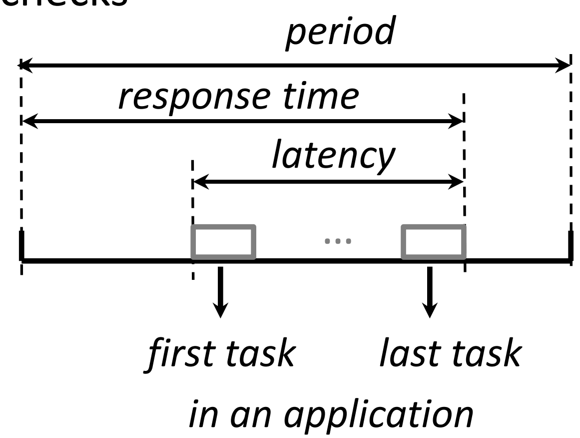
- **Application-level objectives**
  - Response time

    -> Applications that need to be finished as soon as possible in a period

    -> E.g. platform/system states, data/state integrity checks

For a set of applications $\mathcal{A}(obj) \quad \forall i, a_i \in \mathcal{A}(obj)$

Max. response time:	$obj = max(a_i.rt)$
AVG. response time:	$obj = \sum a_i.rt/N$



Multi-Objective Optimization

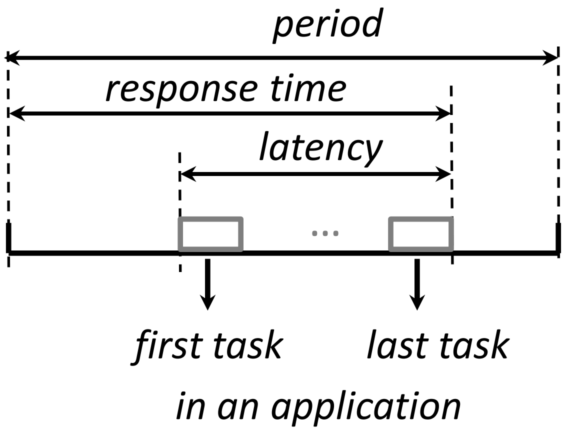
- **Application-level objectives**
  - Response time

    -> Applications that need to be finished as soon as possible in a period

    -> E.g. platform/system states, data/state integrity checks

For a set of applications $\mathcal{A}(obj) \quad \forall i, a_i \in \mathcal{A}(obj)$

Max. response time:	$obj = max(a_i.rt)$
AVG. response time:	$obj = \sum a_i.rt/N$



  - End-to-end latency

    -> Applications that need to have a low latency

    -> E.g. feedback control loops

Max. latency:	$obj = max(a_i.lz)$
AVG. latency:	$obj = \sum a_i.lz/N$

Multi-Objective Optimization

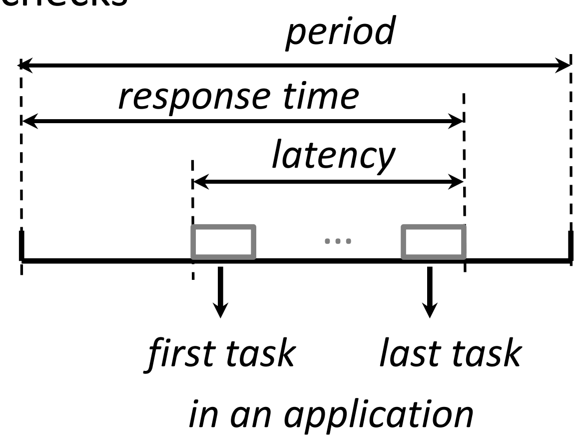
- **Application-level objectives**
  - Response time

    -> Applications that need to be finished as soon as possible in a period

    -> E.g. platform/system states, data/state integrity checks

For a set of applications $\mathcal{A}(obj)$ $\forall i, a_i \in \mathcal{A}(obj)$

Max. response time:	$obj = max(a_i.rt)$
AVG. response time:	$obj = \sum a_i.rt/N$



  - End-to-end latency

    -> Applications that need to have a low latency

    -> E.g. feedback control loops

Max. latency:	$obj = max(a_i.lz)$
AVG. latency:	$obj = \sum a_i.lz/N$

- **Multi-objective optimization**
  - Optimize according to several objectives

For all objectives $\{obj_i\}$

$$obj_M = \sum obj_i \times \omega_i$$

MIP Model Formulation/Solving

- **Constraints and objective formulation MIP**
  - Simple inequity constraints:

    -> straight forward constraint formulation
  - Either-or constraints (e.g. collision free constraints) :

    -> introduce a binary decision variable and formulate the constraint with two inequities [15]
  - Mini-max objective (e.g. max. latency of N applications):

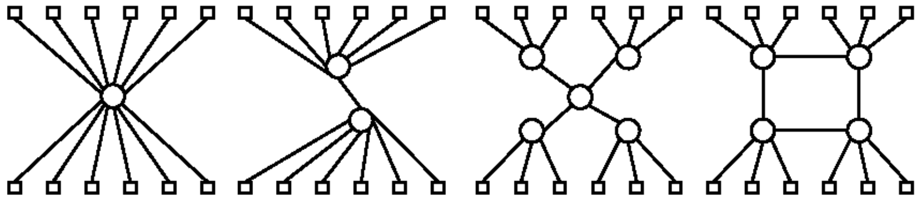
    -> introduce a continuous variable in the objective function and N inequities in the constraints [15]

- **Solving the MIP models**
  - Commercial or non-commercial solvers (e.g. Gurobi, Cplex)

Case Study

- **System description**
  - 30 applications: a_1 to a_{30} , 53 application tasks, 23 communication tasks (frames)
  - Harmonic periods – {4,5,10,20} ms, various WCETs and frame lengths

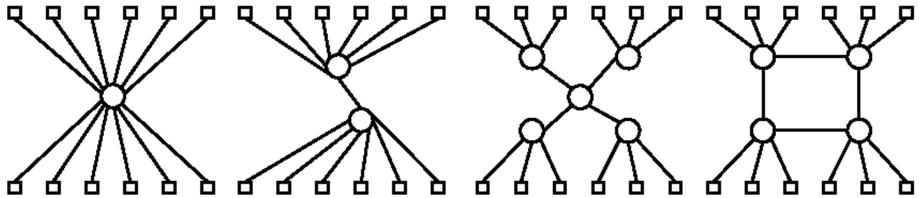
- **Network topologies**
  - 12 processing units
  - 4 topologies



Case Study

- **System description**
  - 30 applications: a_1 to a_{30} , 53 application tasks, 23 communication tasks (frames)
  - Harmonic periods – {4,5,10,20} ms, various WCETs and frame lengths

- **Network topologies**
  - 12 processing units
  - 4 topologies

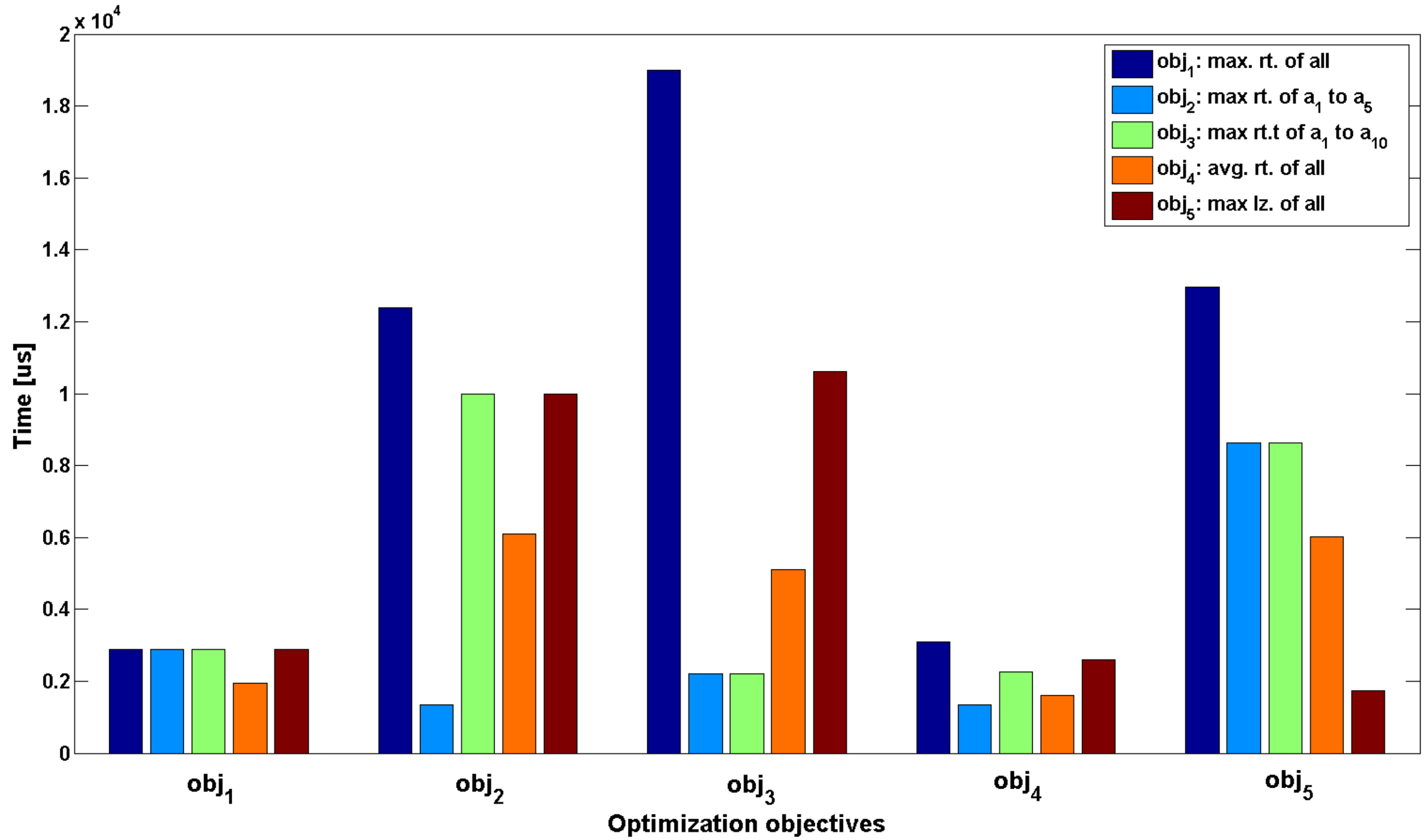


- **Optimization Objectives**

obj_1	*max. response time of* a_1 *to* a_{30}
obj_2	*max. response time of* a_1 *to* a_5
obj_3	*max. response time of* a_1 *to* a_{10}
obj_4	*avg. response time of* a_1 *to* a_{30}
obj_5	*max. end-to-end latency of* a_1 *to* a_{30}

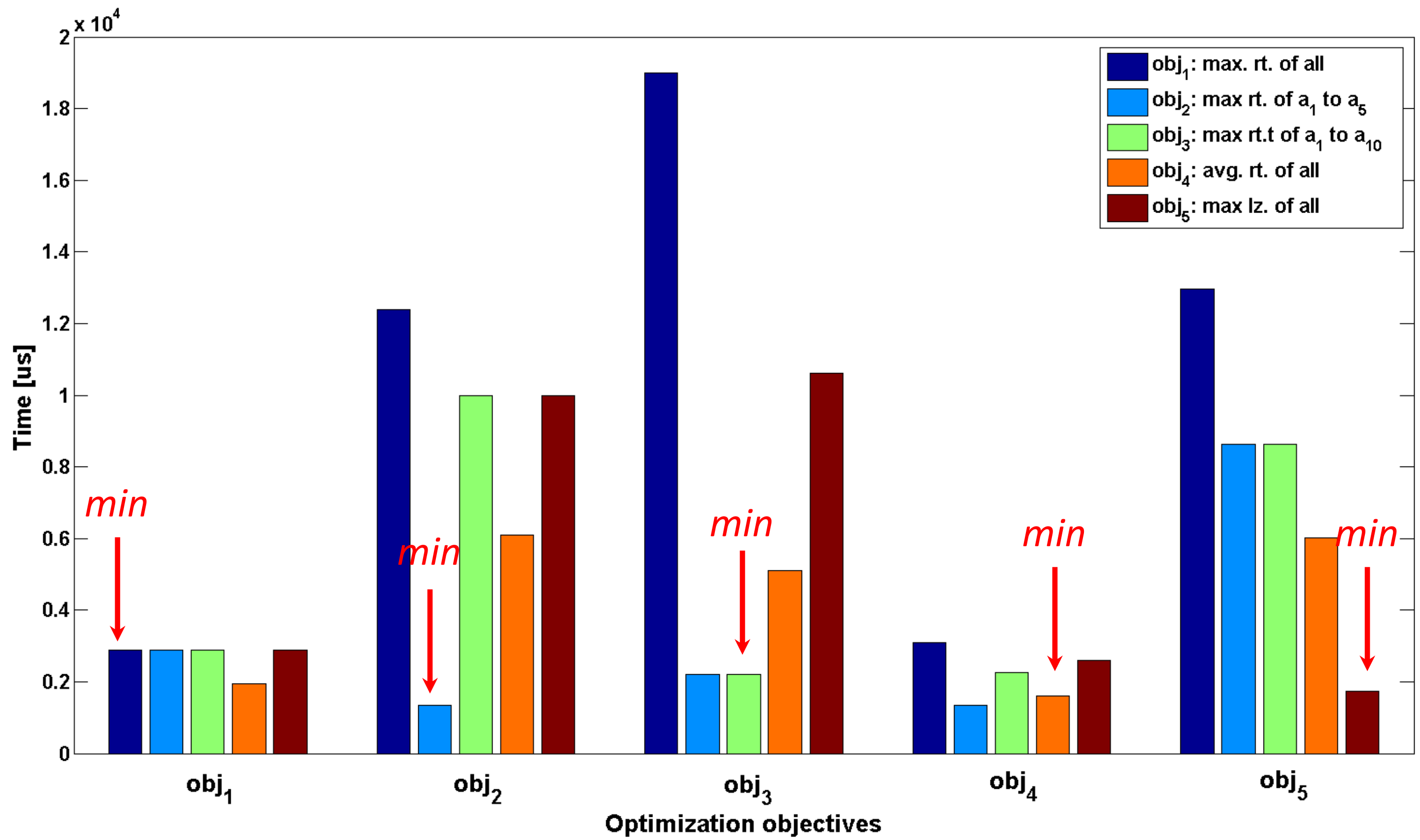
Experimental Results

- **Experimental Results**
  - Comparison of different single-objective optimizations in tree topology



Experimental Results

- **Experimental Results**
  - Comparison of different single-objective optimizations in tree topology

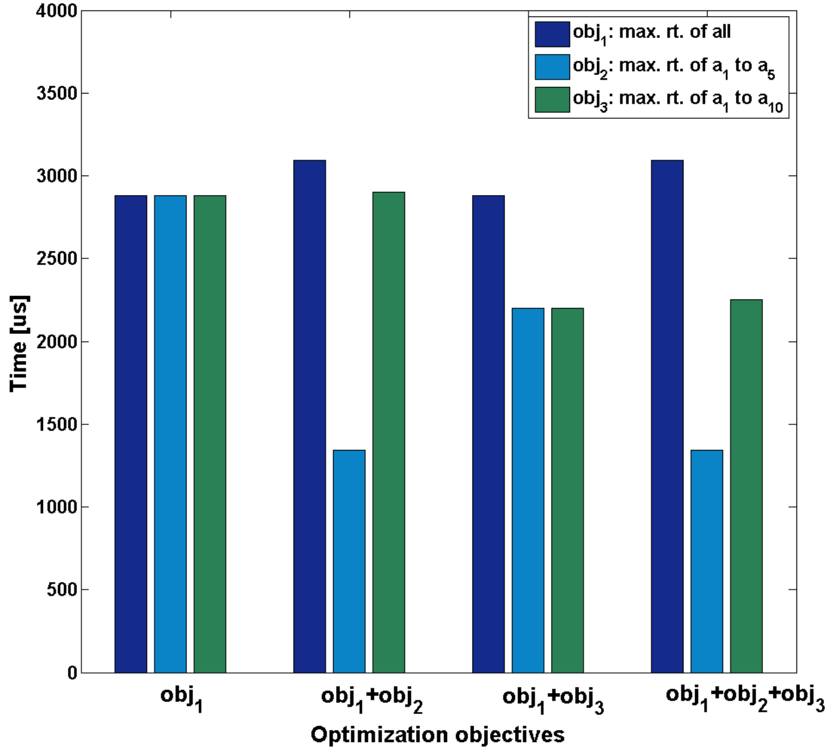


Experimental Results

- **Experimental Results**
  - Comparison of different multi-objective optimizations in tree topology

multi-objective case

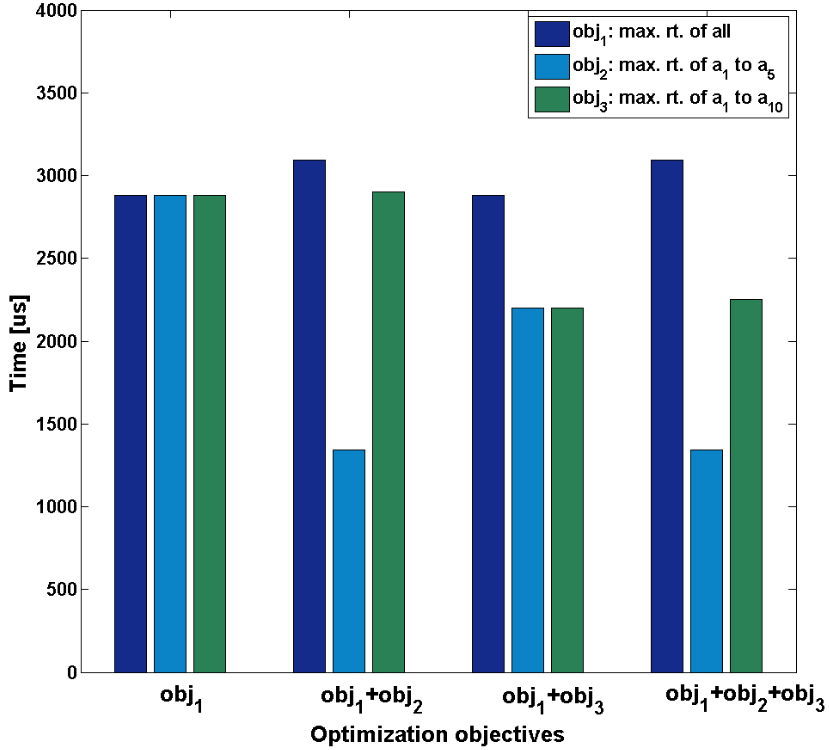
obj_1, obj_2, obj_3



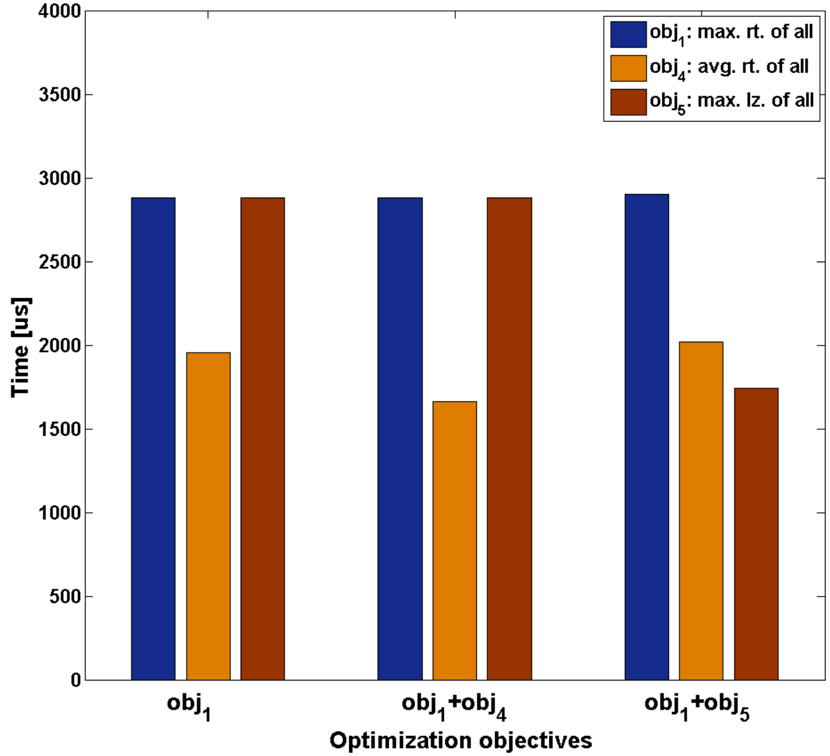
Experimental Results

- **Experimental Results**
  - Comparison of different multi-objective optimizations in tree topology

multi-objective case
obj_1, obj_2, obj_3



multi-objective case
obj_1, obj_4, obj_5

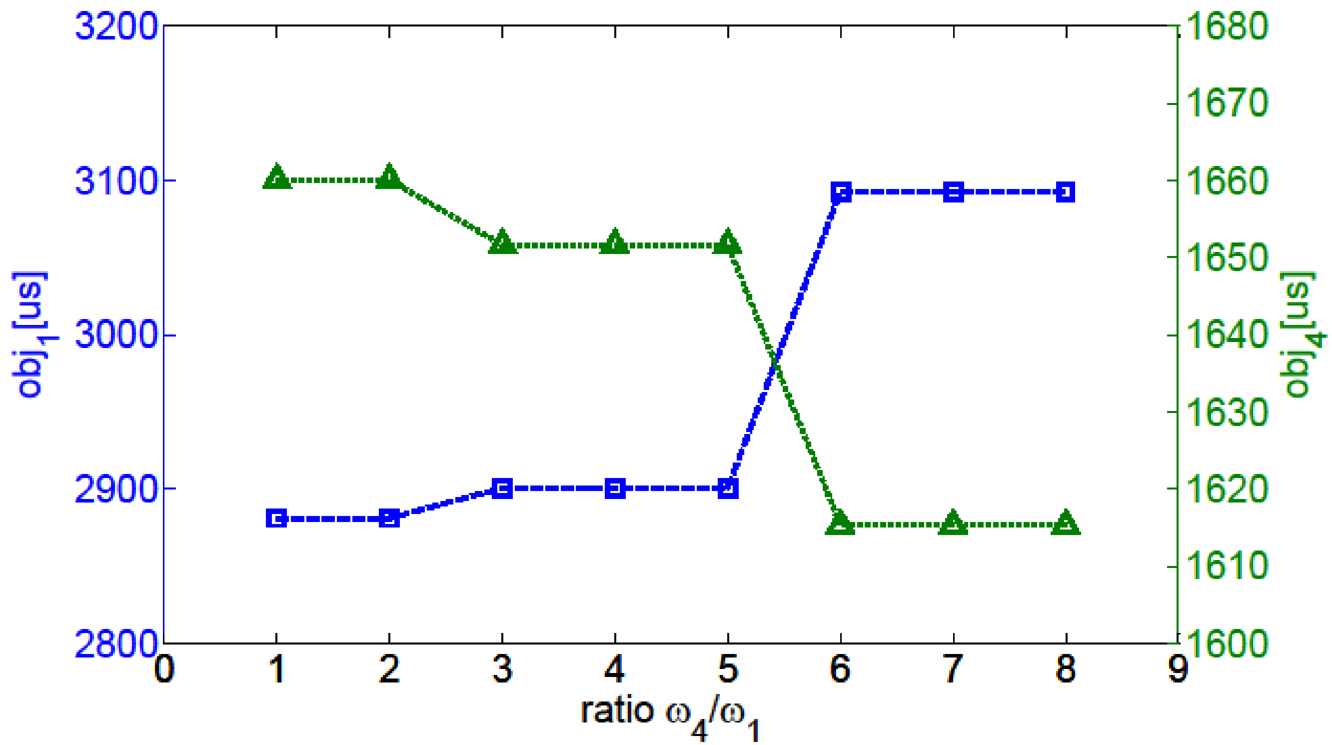


Experimental Results

- **Influence of weight in multi-objective optimization**

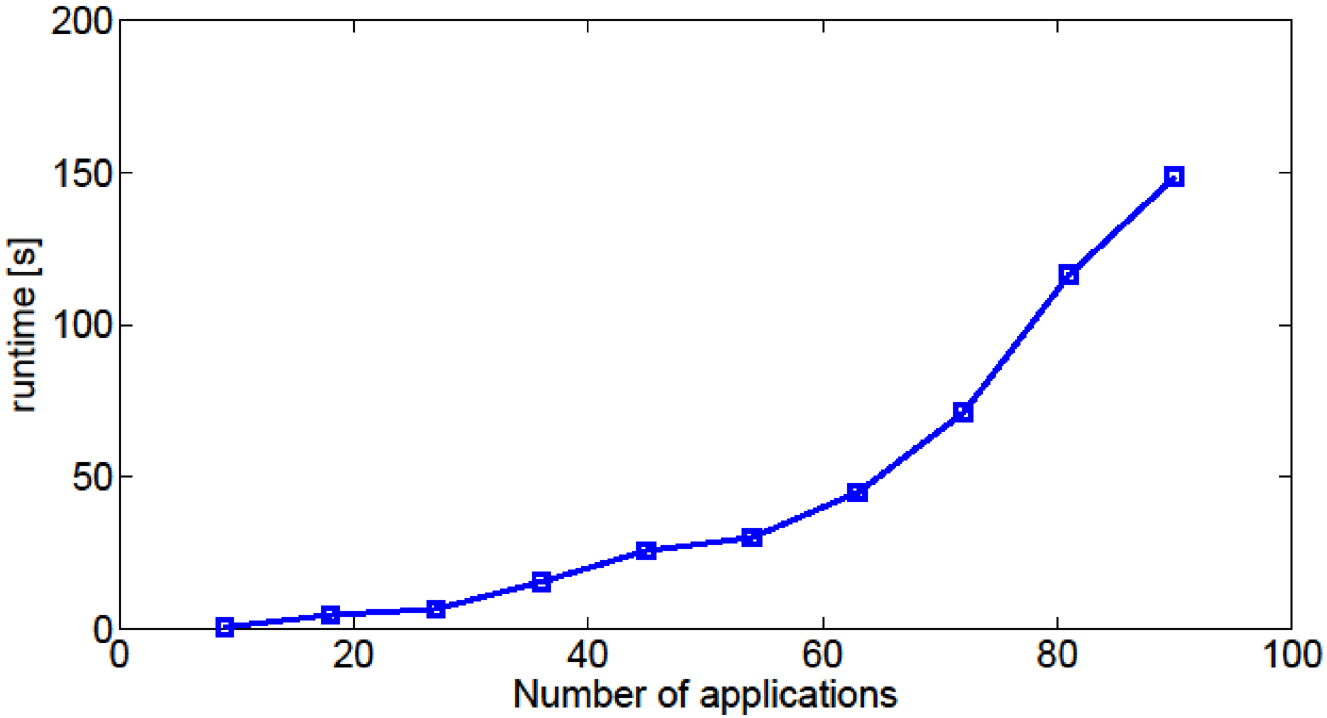
multi-objective case with different weight ratio for obj_1, obj_4

$$obj_M = obj_1 \times \omega_1 + obj_4 \times \omega_4$$



Computational Cost/ Scalability

- **Scalability analysis**
  - Synthetic test configurations from size of 9 application to 90 applications
  - Setup: 1.87GHz dual core CPU, 4 GB memory, MATLAB 2010 with Gurobi 5.10



Concluding Remarks

- **Approach**
  - Schedule co-synthesis problem for Ethernet-based time-triggered system
  - Formulation of constraints in such a system
  - Multi-objective optimization

Concluding Remarks

- **Approach**
  - Schedule co-synthesis problem for Ethernet-based time-triggered system
  - Formulation of constraints in such a system
  - Multi-objective optimization

  - Co-synthesis of task and communication schedules according to application-level objectives
  - Independent of task and communication configuration, network topologies and device performance

Concluding Remarks

- **Approach**
  - Schedule co-synthesis problem for Ethernet-based time-triggered system
  - Formulation of constraints in such a system
  - Multi-objective optimization

  - Co-synthesis of task and communication schedules according to application-level objectives
  - Independent of task and communication configuration, network topologies and device performance

- **Outlook**
  - Extensibility and sustainability of synthesized schedules
  - Local sub-optimal searches for plug-in schedules
  - Schedule synthesis according to function-level properties

References

[1] R. Pigan, M. Metter, Automating with PROFINET, Publicis Publishing, 2nd edition, .

[2] “664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network”, ARINC, 2009.

[3] “AS6802: Time-triggered Ethernet”, SAE International, 2011.

[4] “Media access contro (MAC) bridges and virtual bridge local area networks”, IEEE Computer Society, 2011.

[5] “http://www.ieee802.org/1/pages/802.1bv.html”.

[6] H. Kopetz, G. Bauer, “The time-triggered architecture”, Proceedings of the IEEE, vol. 91, iss. 1, 112-126, 2003.

[7] M. Lukasiewycz, M.Glass, J. Teich, P. Milbredt, “FlexRay schedule optimization of the static segment”, CODES+ISSS, 2009.

[8] H. Zeng, M. Di Natale, A. Ghosal, A. Sandiovanni-Vincentelli, “Schedule optimization of time-triggered systems communicating over the FlexRay static segment”, Industrial Infomatics, IEEE Transactions on, vol. 7, iss. 1, 1-17, 2011.

[9] D. Goswami, M. Lukasiewycz, R. Schneider, S. Chakraborty, “Time-triggered implementations of mixed-criticality automotive software”, DATE, 2012.

[10] W. Steiner, “An evaluation of SMT-based schedule synthesis for time-triggered multi-hop netoworks”, RTSS, 2010.

[11] W. Steiner, “Synthesis of static communication schedules for mixed-criticality systems”, ISORCW, 2011.

[12] Z. Hanzalek, P. Burget, P. Sucha, “Profinet IO IRT message scheduling with temporal constraints”, Industrial Informatics , IEEE Transactions on, vol. 6, iss. 3, 269-380, 2010.

[13] D. Tamas-Selicean, P. Pop, W. Steiner, “Synthesis of communication schedules for TTEthernet-based mixed-criticality sytems”, CODES+ISSS, 2012.

[14] R. Seifert, J. Edwards, The All-New Switch Book: The Complete Guide to LAN Switching Technology, Wiley Publishing, Inc., 2008.

[15] H. P. Williams, Model Building in Mathematical Programming, 4th edition, John Wiley and Sonds, LTD, 1999.

[16] “www.gurobi.com”

The End

Many thanks

Q/A