DESIGN, AUTOMATION & TEST IN EUROPE
25 – 29 March, 2019 · Firenze Fira · Florence · Italy
The European Event for Electronic System Design & Test

INSTITUTE OF COMPUTER AND NETWORK ENGINEERING
ONERA THE FRENCH AEROSPACE LAB

Improving Accuracy of Timing Models: From CPA to CPA+

Leonie Köhler, Borislav Nikolic, Rolf Ernst (TU Braunschweig, Germany)

Marc Boyer (ONERA, France)

Motivation

Verification of real-time systems and networks

ONERA, Toulouse

Network Calculus (NC)

TU Braunschweig

Compositional performance analysis (CPA)

- Formal timing analysis provides safe but pessimistic bounds on real-time properties for complex embedded computing systems.
- In industrial practice, formal analysis is often replaced by simulation and test.

➡ Unsatisfactory situation

Motivation

Potential causes of pessimism?

Comparison of NC and CPA

- **One interesting observation:**
  - NC is good at analyzing workload-dependent scheduling scenarios
  - CPA is good at analyzing event-dependent scheduling scenarios
- **Underlying reasons:**
  - Different system modeling

Can we find a common input model of NC and CPA?

- Boyer et al. "Embedding network calculus and event stream theory in a common model." *ETFA* 2016.

What does it mean to integrate this common input model in CPA?

- **This paper at *DATE* 2019, leading to CPA+.**

Outline

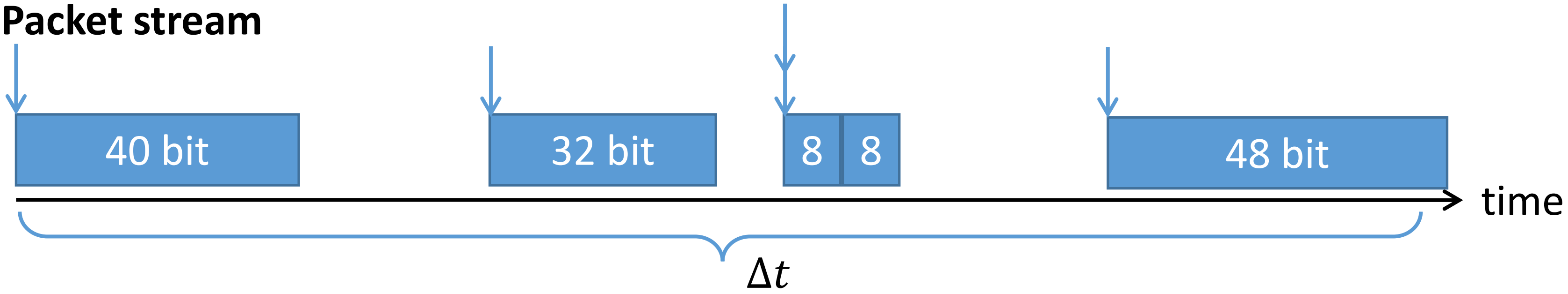
- Input models of CPA and NC
  - Input model of CPA
  - Input model of NC
  - Common input model for CPA and NC [ETFA 2016]
  - Related work
- Introducing CPA+
  - CPA principle
  - Impact of new input model: interfaces, composition, analysis
  - Use cases
- Conclusion

Outline

- Input models of CPA and NC
  - Input model of CPA
  - Input model of NC
  - Common input model for CPA and NC [ETFA 2016]
  - Related work
- Introducing CPA+
  - CPA principle
  - Impact of new input model: interfaces, composition, analysis
  - Use cases
- Conclusion

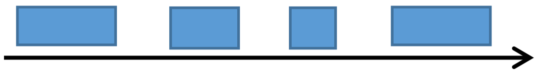
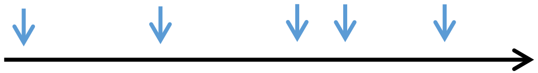
Informal model of a packet stream in NC / CPA

Packet stream

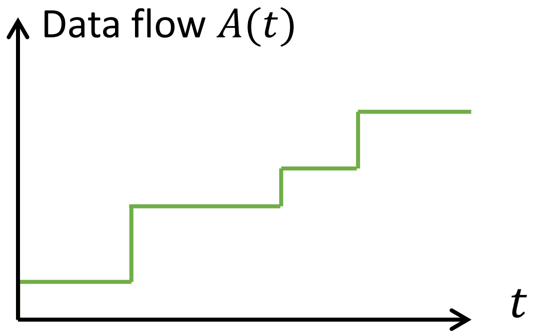
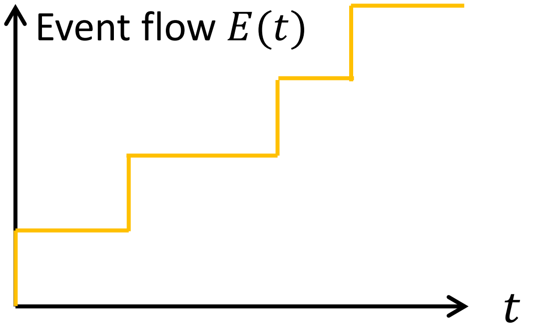
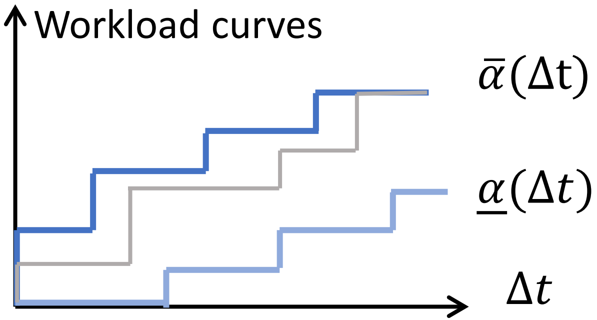
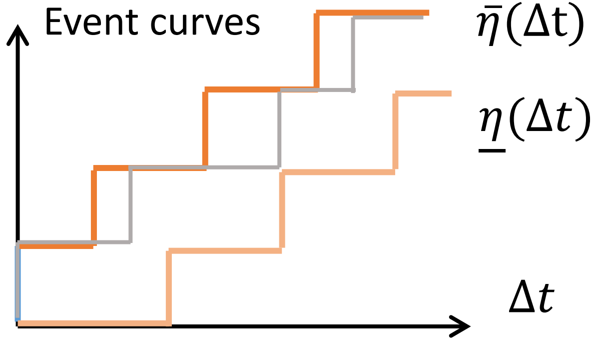


Number of packets

Variance of packet length

	NC	CPA
Characterizing the packet stream	Amount of data sent over time  136 bits in Δt	Number of packets over time  5 packets in Δt

Formal model of a packet stream in NC / CPA

<table>
<tr><th></th><th>NC (workload-based model)</th><th>CPA (event-based model)</th></tr>
<tr><td>Trace (also: flow)</td><td>
</td><td>
</td></tr>
<tr><td>Interval Bounding Pair (IBP)</td><td>
</td><td>
</td></tr>
</table>

Common input model for NC/CPA

- Addition of a **packet function $\boldsymbol{P}$**: number of packets in the first x bits of the data flow
- Interval bounding pair for P: **packet curves $(\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}})$**

NC [ETFA 2016]	CPA [ETFA 2016]
Workload curves $(\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}})$ bounding data flow $A(t)$	Workload curves $(\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}})$ bounding data flow $A(t)$ are **implicit**.
Packet curves $(\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}})$ bounding packet function P	Packet curves $(\underline{\boldsymbol{\pi}}, \overline{\boldsymbol{\pi}})$ bounding packet function P
Event curves $(\underline{\boldsymbol{\eta}}, \overline{\boldsymbol{\eta}})$ bounding event flow E(t) are **implicit**.	Event curves $(\underline{\boldsymbol{\eta}}, \overline{\boldsymbol{\eta}})$ bounding event flow E(t)

compatibility

Common input model for NC/CPA

What is the benefit of the common input model?

- **NC** needs to **infer information on events** during analysis.
- **CPA** needs to **infer information on workload** during analysis.

NC: From workload to events

Simple transformation

$$\underline{\eta}(\Delta t) = \frac{\underline{\alpha}(\Delta t)}{p^+}$$

$$\overline{\eta}(\Delta t) = \frac{\overline{\alpha}(\Delta t)}{p^-}$$

Transformation [ETFA 2016]

$$\underline{\eta}(\Delta t) = (\overline{\pi} \circ \underline{\alpha})(\Delta t)$$

$$\overline{\eta}(\Delta t) = (\underline{\pi} \circ \overline{\alpha})(\Delta t)$$

lossless

CPA: From events to workload

Simple transformation

$$\underline{\alpha}(\Delta t) = p^- \cdot \underline{\eta}(\Delta t)$$

$$\overline{\alpha}(\Delta t) = p^+ \cdot \overline{\eta}(\Delta t)$$

Transformation [ETFA 2016]

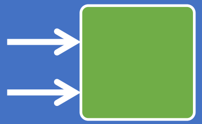
$$\underline{\alpha}(\Delta t) = (\underline{\pi}^{\overline{-1}} \circ \underline{\eta})(\Delta t)$$

$$\overline{\alpha}(\Delta t) = (\overline{\pi}^{\underline{-1}} \circ \overline{\eta})(\Delta t)$$

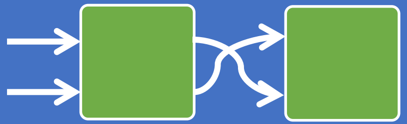
bit arrival rates are lost

Related work

component
with workload-event
input model



Composition
of components



Add variation in packet sizes

- *Mok et al. 1997*
- *Quinton et al. 2012*

Add information on events

- *Wandeler et al. 2005*
- *Bouillard et al. 2011 + 2012*

Compatible input model for NC/CPA

- *Boyer et al. 2016*

NC

- *Wandeler et al. 2005*
- *Bouillard et al. 2011 + 2012*
- *Boyer et al. 2016*

CPA

- Contribution DATE 2019

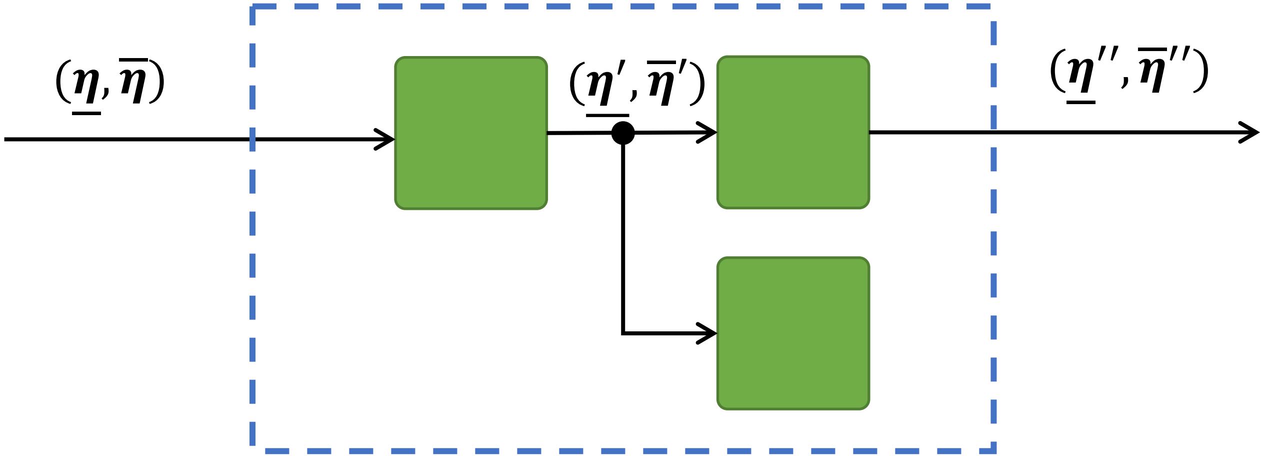
Outline

- Input models of CPA and NC
  - Input model of CPA
  - Input model of NC
  - Common input model for CPA and NC [ETFA 2016]
  - Related work
- **Introducing CPA+**
  - **CPA principle**
  - **Impact of new input model: interfaces, composition, analysis**
  - **Use cases**
- Conclusion

CPA principle

Compositional approach:

- **Global analysis problem** is decomposed into dependent **local analysis problems**
- Dependencies are represented by **event streams**

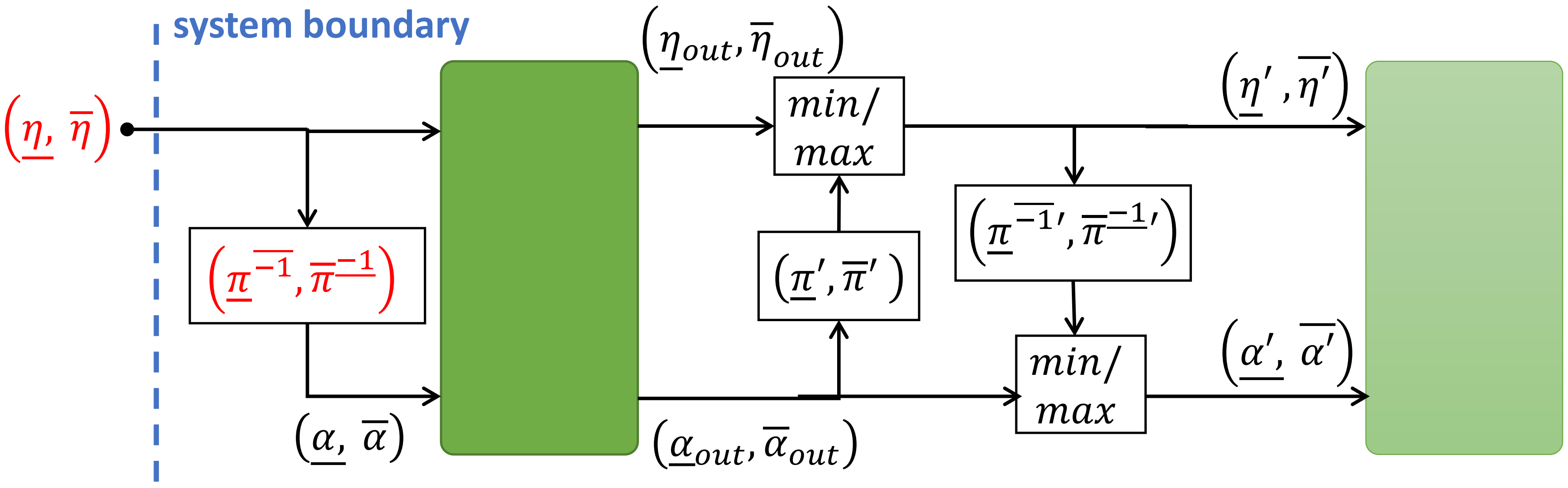


New input model will change interfaces of local analysis problems.

Composition of local analysis problems?

Internals of local analysis problems?

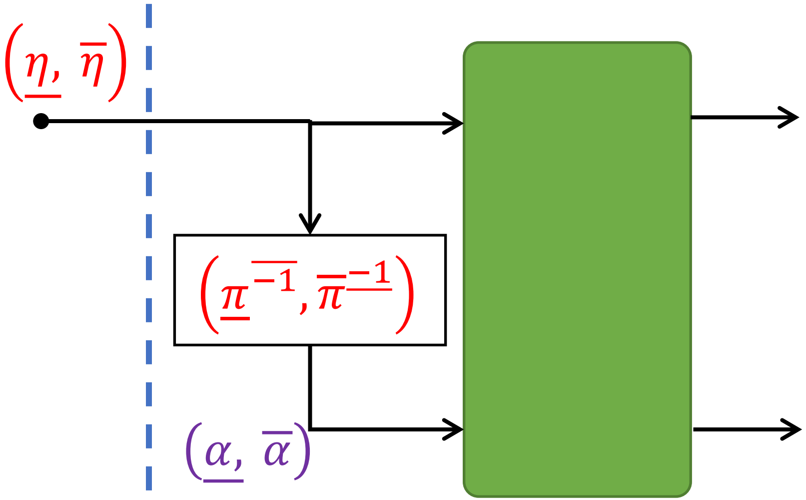
Naïve interface and composition



- **Lossy event-to-workload transformation at the input**
- **Composition allows lossy chained transformations**

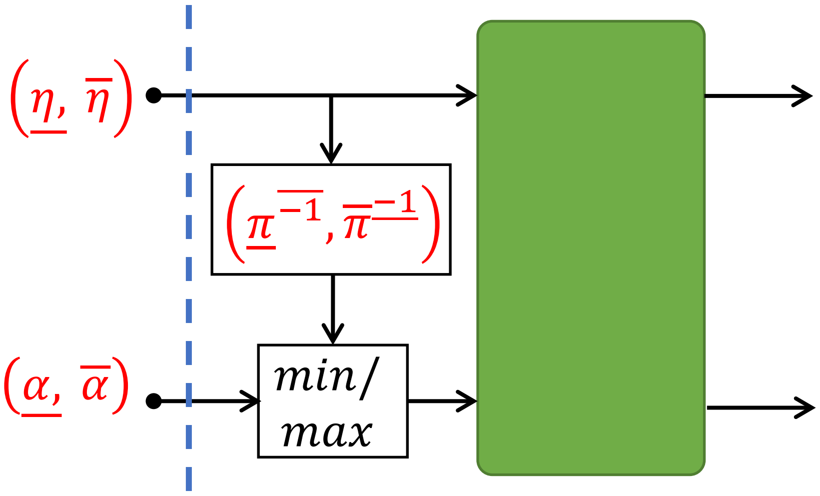
Interface issues

Lossy event-to-workload transformation at the input



2 primary inputs: events + packet sizes
1 implicit input: workload

Better:

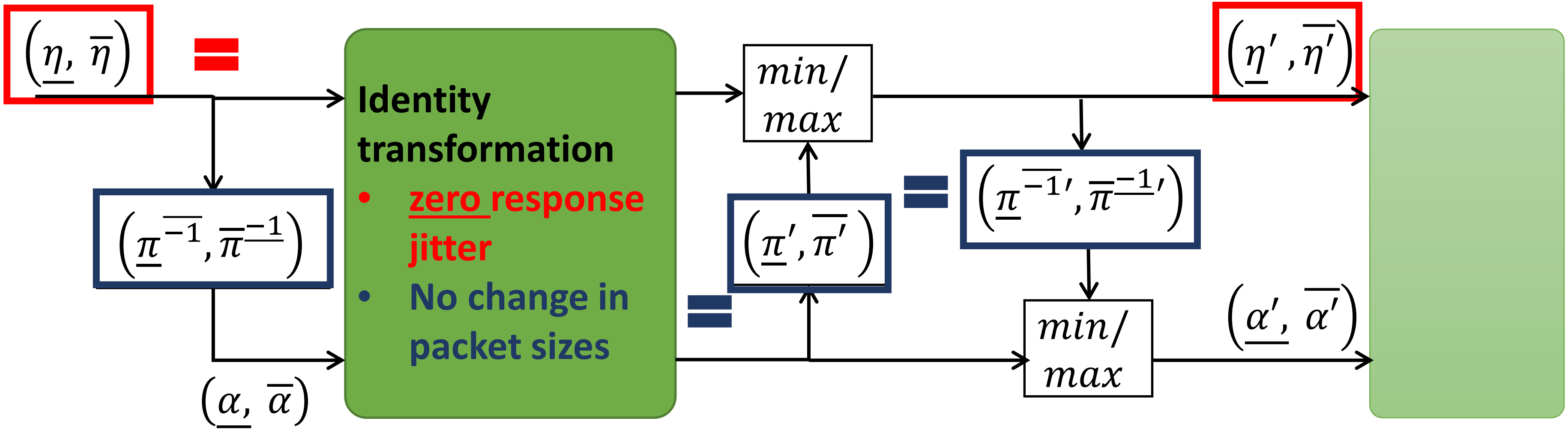


3 primary inputs:
Events, workload and packets

Composition issues

Chained transformations in the naïve composition

Example:

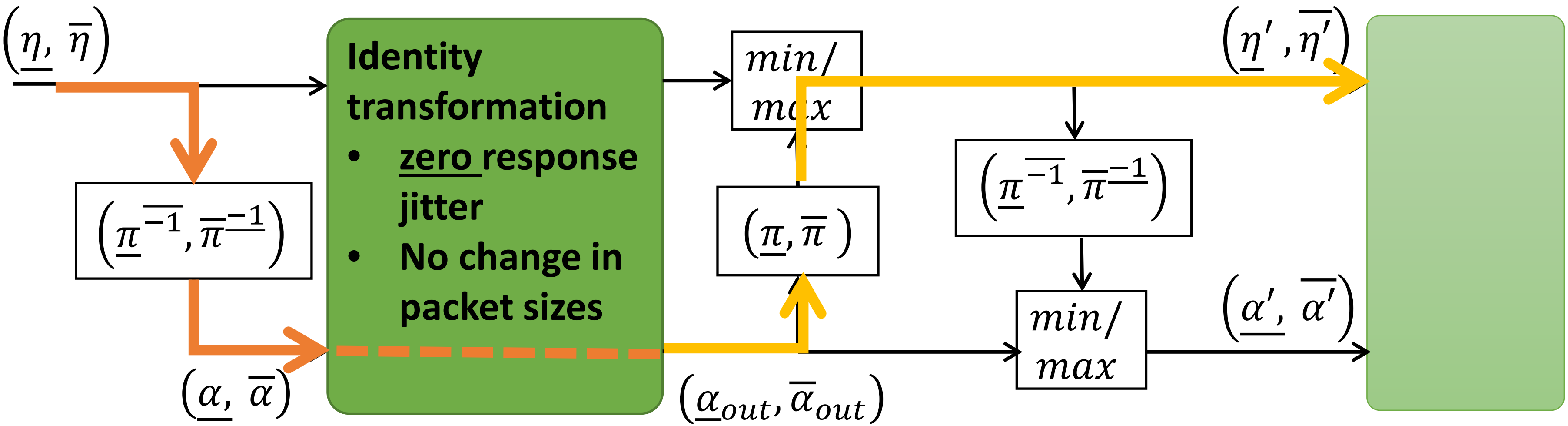


1st transf.: event-to-workload
$\overline{\alpha}_{out}(\Delta t) = (\overline{\pi}^{-1} \circ \overline{\eta})(\Delta t)$

Maximum workload = largest packets combined with maximum event number

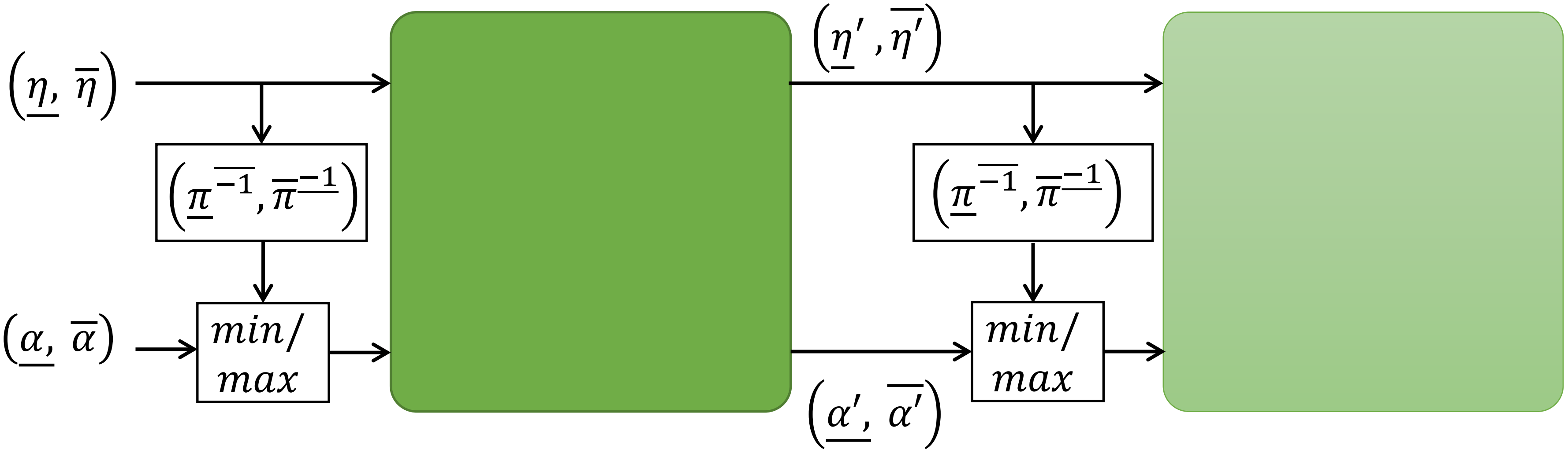
2nd transf.: workload-to-event
$\overline{\eta}'(\Delta t) = (\underline{\pi} \circ \overline{\alpha}_{out})(\Delta t)$

Maximum number of events = smallest packets combined with maximum workload



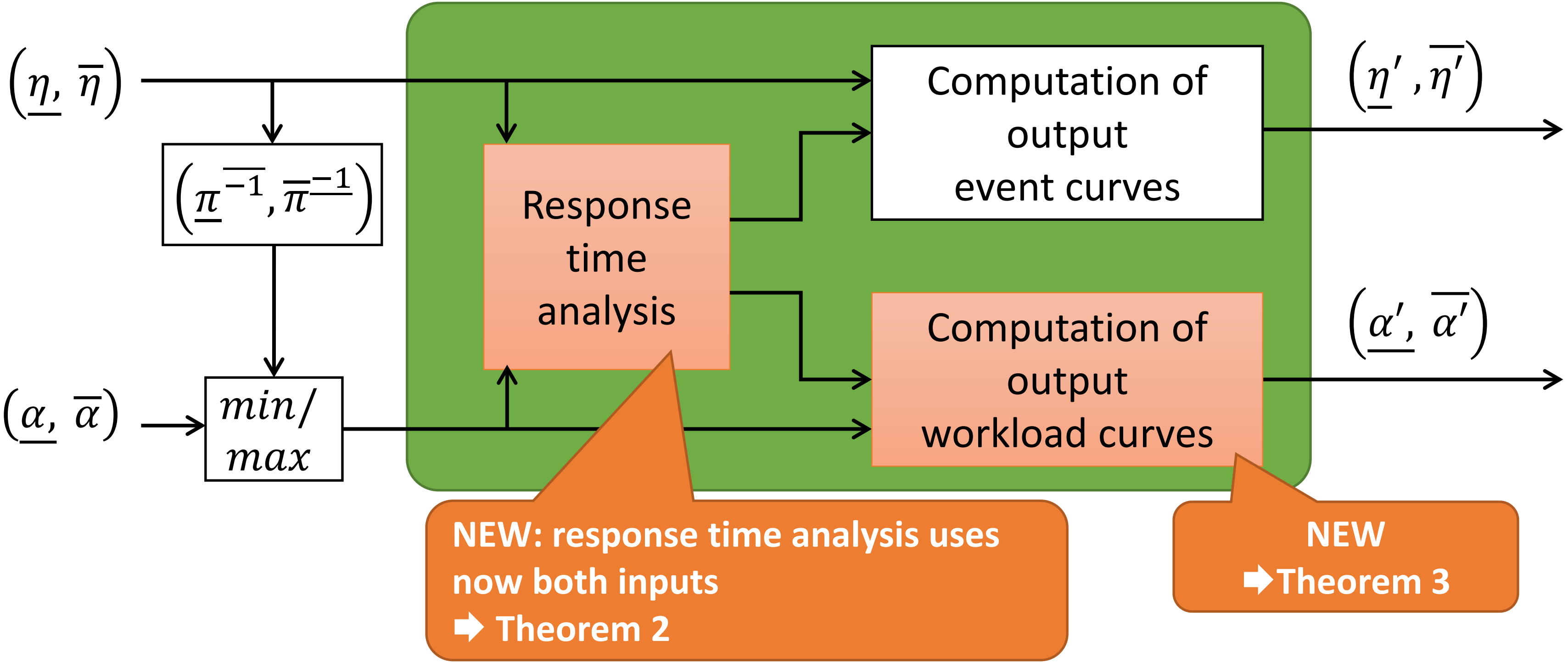
$$\overline{\eta}'(\Delta t) = (\underline{\pi} \circ \overline{\pi}^{-1} \circ \overline{\eta})(\Delta t) \quad \Rightarrow \quad \overline{\eta}'(\Delta t) > \overline{\eta}(\Delta t)$$

Improved interface and composition



- ✓ Three primary inputs
- ✓ Elimination of chained transformations:
  only event-to-workload transformations are allowed.

Impact on the local analysis problem



Use Cases for Event-Workload-Dualism in NC and CPA

Multi-component system with complex schedulers

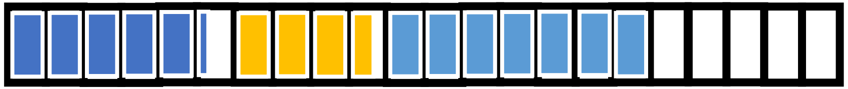
- event-centered scheduler, e.g., round robin
  - Distinction of individual packets (events) is important!
- workload-centered scheduler, e.g., budget-based scheduling
  - Sent data over time is important (workload), not number of packets!
- scheduler depending on events and workload, e.g., packetizer
  - Packet dispatch on time-out, buffer-full, high-priority event, …

- **Multi-component system with different service specifications**
  - per-event service specification, e.g., decoder
  - per-workload service specification, e.g., processor

Use Cases for Event-Workload-Dualism in NC and CPA

- **Computation of buffer filling**
  - **per-packet (event-based)**

  - **per-byte (workload-based)**



Outline

- Input models of CPA and NC
  - Input model of CPA
  - Input model of NC
  - Common input model for CPA and NC [ETFA 2016]
  - Related work
- Introducing CPA+
  - CPA principle
  - Impact of new input model: interfaces, composition, analysis
  - Use cases
- **Conclusion**

Conclusion

Problem:

- Apply a **dual (event-workload) input model** to CPA

Solution: CPA+

- Define and compute a **dual (event-workload) output model** for CPA components
- Define **new composition rules** for CPA components
- Update **response-time analysis** within CPA components.

Benefits:

- Accurate system model and analysis for complex systems
- Compatability of NC and CPA+

Future work:

- joint tool of NC and CPA+