Algebraic transformations for network paths with hop-by-hop flow control

RAFFAELE ZIPPO, GIOVANNI STEA
UNIVERSITY OF PISA

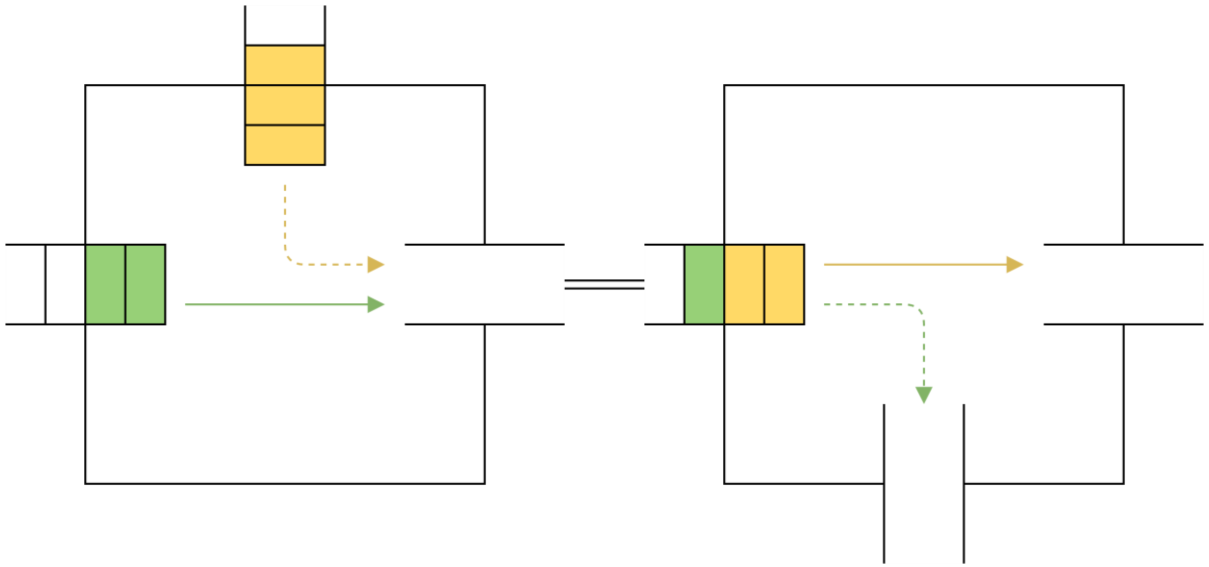
@ WONECA-5 VIRTUAL EVENT, 09/10/2020

Context: Wormhole Networks

Packets are divided in *flits*

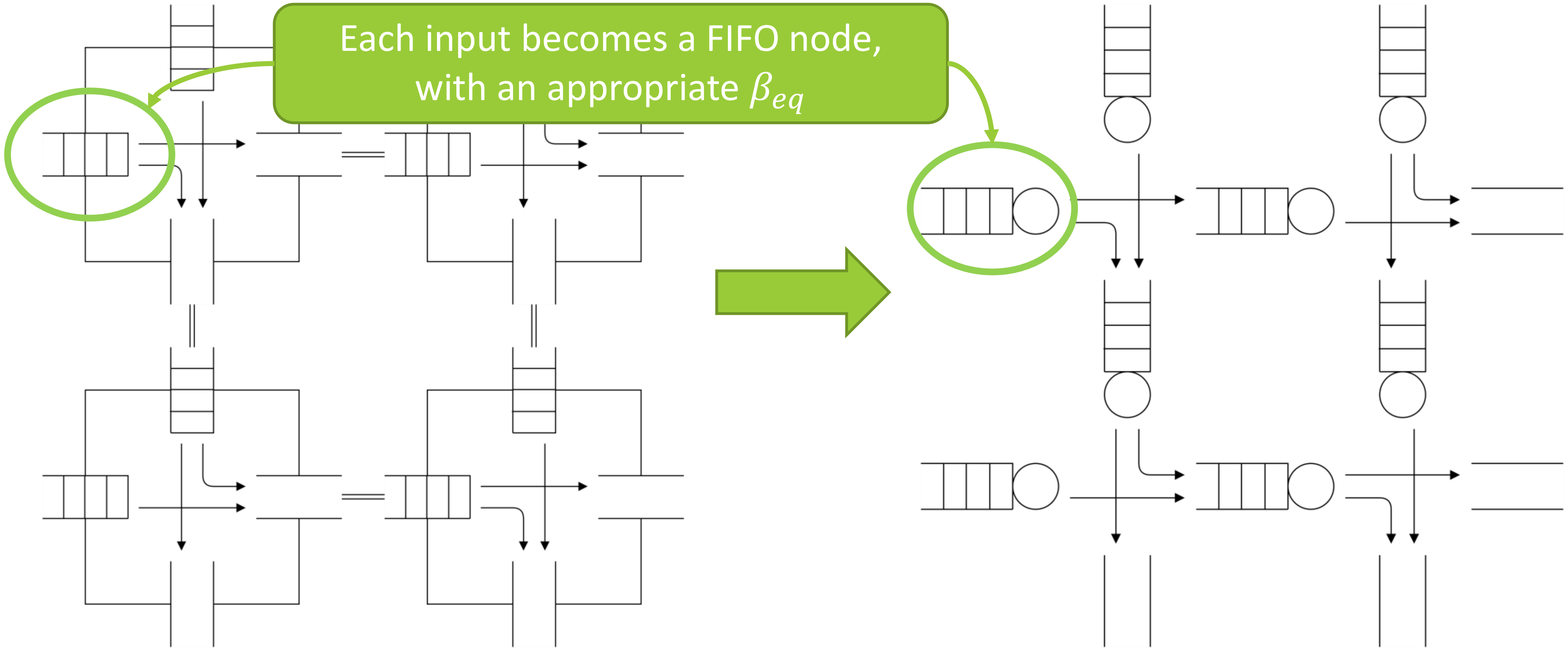
Flits move through input buffers, which are FIFO, one by one

Credit system to authorize transmission – i.e. window flow control at each hop



Context: Wormhole Networks

Approach: transform into a FIFO feed-forward network



[1] *Analysis of Worst-case Delay Bounds for Best-effort Communication in Wormhole Networks on Chip,* Yue Qian et ali, 2009

Context: Wormhole Networks

Approach: transform into a FIFO feed-forward network

$$\beta_{eq}^{A} = f(\ \textit{contention at input buffer (FIFO)}, \textit{arbitration at output port (e.g. WRR)}, \textit{flow control at ouput link}\ )$$

$\min(\beta_{f_1}, \beta_{f_2}, \dots)$

add delay, reduce rate

$\beta_{fc} = \overline{W + \beta_{eq}^{B}}$

Recursion: equivalent service at given node depends on downstream equivalent service, due to flow control

[1] *Analysis of Worst-case Delay Bounds for Best-effort Communication in Wormhole Networks on Chip,* Yue Qian et ali, 2009

Problem: sub-additive closure

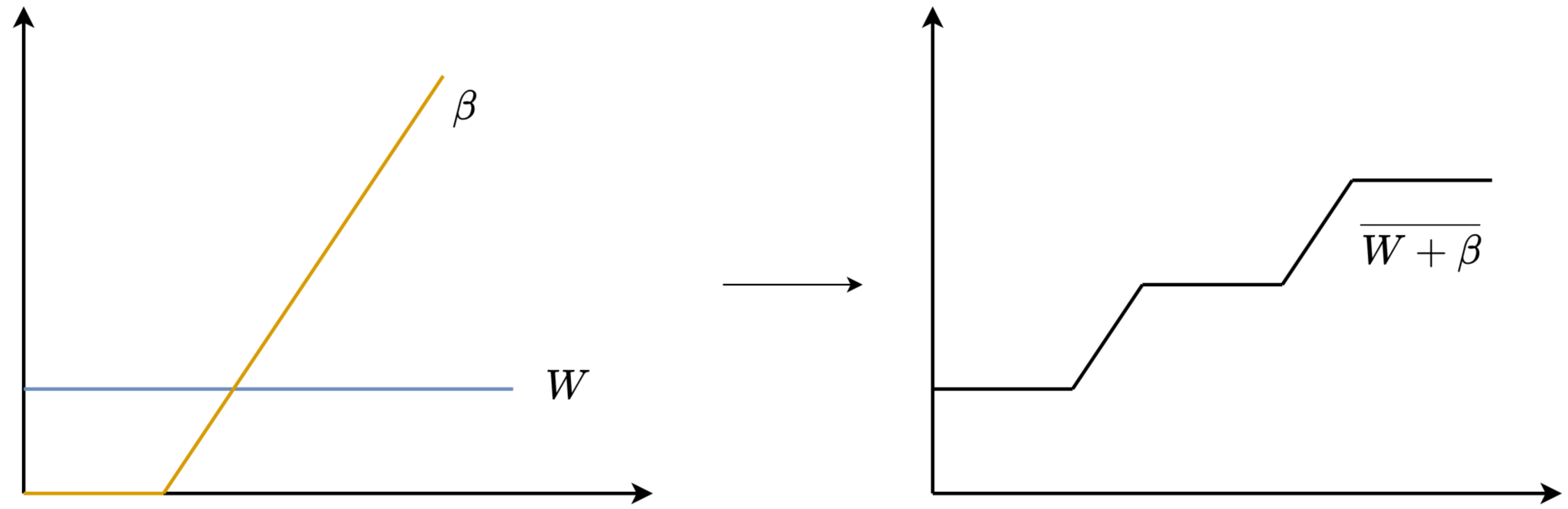
$$\beta_{eq}^{A} = [\ldots] \otimes \overline{W + \underbrace{\left[[\ldots] \otimes \overline{W + \cdots} \right]}_{\beta_{eq}^{B}}}$$

Computing generic sub-additive closures is an NP-hard problem

Solving this nested form is prohibitive

Problem: sub-additive closure

Closed-form is known for flow control on a delay-rate server

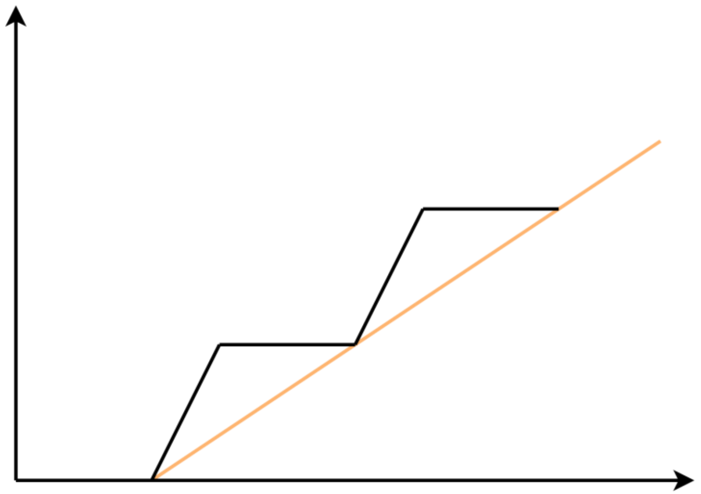


Sloppy way: lower bounding to delay-rate

Closed-form is known for flow control on a delay-rate server

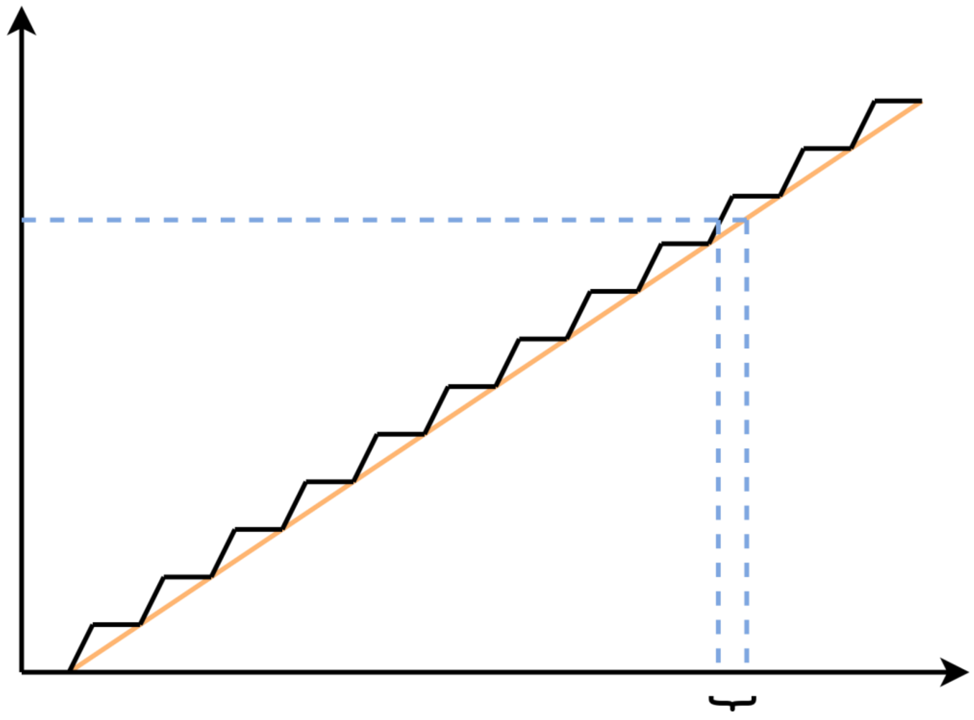
So, as an easy fix, we can lower-bound downstream β_{eq} with a delay-rate service

$$\beta_{eq}^{A} = [\dots] \otimes \overline{W + \lfloor \beta_{eq}^{B} \rfloor_{dr}}$$

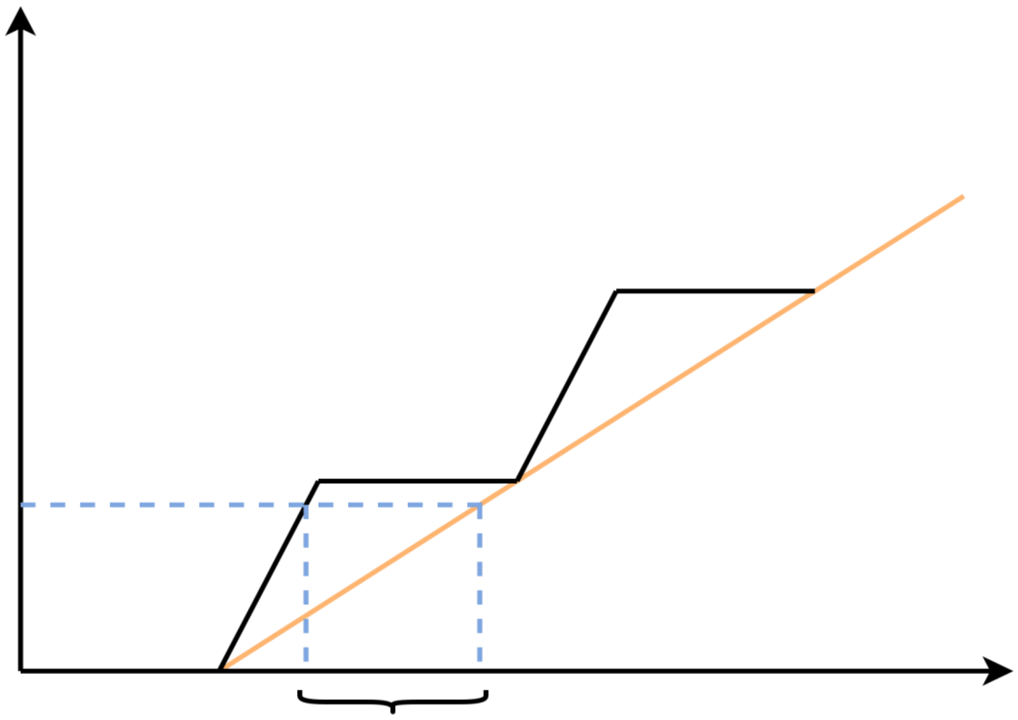


Impact on computed delay

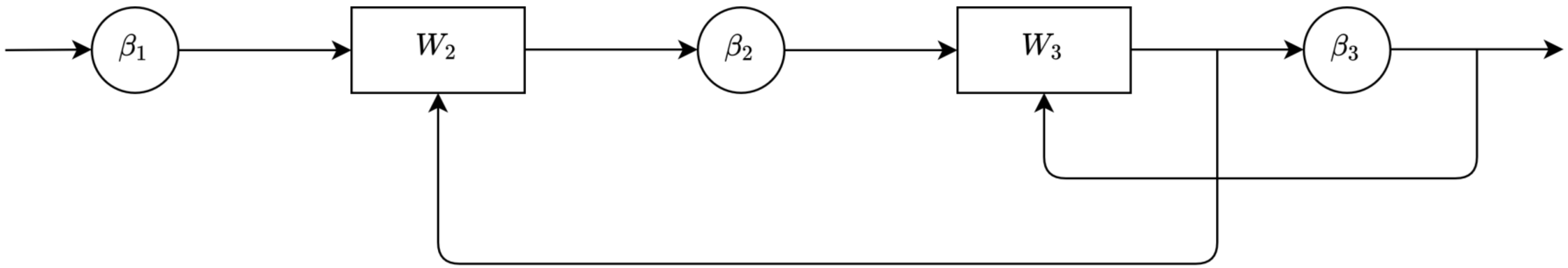
GOOD FOR LONG STREAMS



BAD FOR SHORT MESSAGES



Flow control tandem



$$\beta_1^{eq} = \beta_1 \otimes \overline{W_2 + \beta_2^{eq}}$$
$$= \beta_1 \otimes \overline{W_2 + \beta_2 \otimes \overline{W_3 + \beta_3}}$$

Flow control tandem

We can transform the recursive flow control expression from a chain of generic sub-additive closures into a series of convolutions with closed-form closures.

$$\beta_1^{eq} = \beta_1 \otimes \overline{W_2 + \beta_2 \otimes \overline{W_3 + \beta_3}}$$
$$\geq \beta_1 \otimes \overline{W_2 + \beta_2} \otimes \overline{W_3 + \beta_3}$$

Expansion to wormhole network

We extended this approach to include routing characteristics

$$\overline{FC_{i,j,k}} \geq \overline{\beta_{r_j}^{min} + W_{r_j}} \otimes \bigotimes_{f_l \in F^{p_j}} k^{f_l} \cdot \overline{FC_{r_j,l,p_j}}$$

Replaces $\overline{W + \beta_{eq}}$

Recursive, but expands to series of convolutions

$$\overline{W_a + \beta_a} \otimes \overline{W_b + \beta_b} \otimes \cdots$$

$\beta_{eq}^{A} = f($

contention at input buffer (FIFO),
arbitration at output port (e.g. WRR),
flow control at ouput link

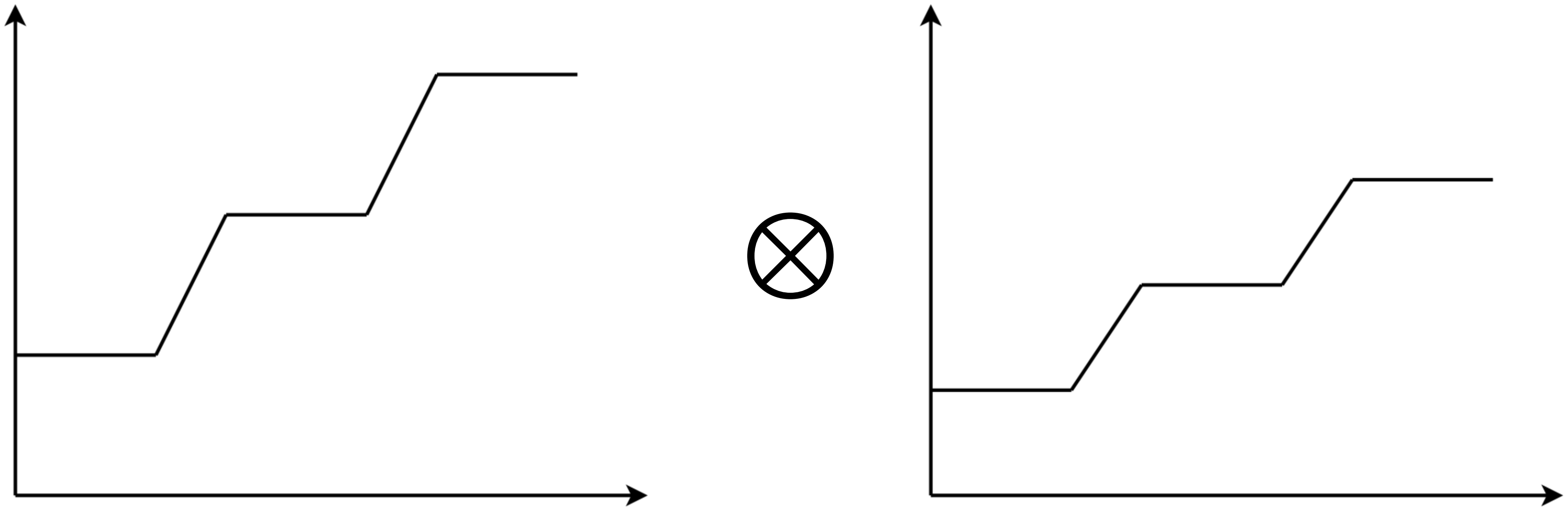
$)$

$\min(\beta_{f_1}, \beta_{f_2}, \ldots)$
add delay, reduce rate
$\beta_{fc} = \overline{W + \beta_{eq}^{B}}$

Convolution performance

$$\beta_{eq} = \beta \otimes \overline{W_a + \beta_a} \otimes \overline{W_b + \beta_b} \dots$$

$\overline{W + \beta}$ elements are not convex, need a generic algorithm to compute convolutions

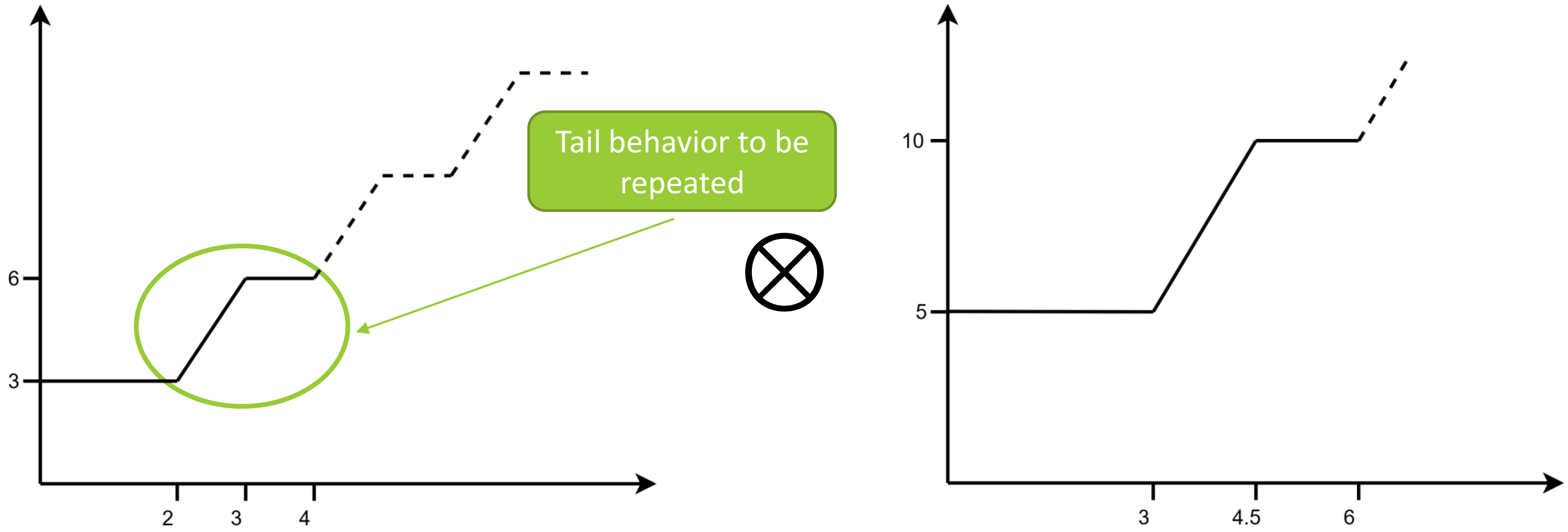


[3] *An Algorithmic Toolbox for Network Calculus, A. Bouillard and E. Thierry, 2007*

Convolution performance

Algorithms in the literature scale poorly with the number of chained operations

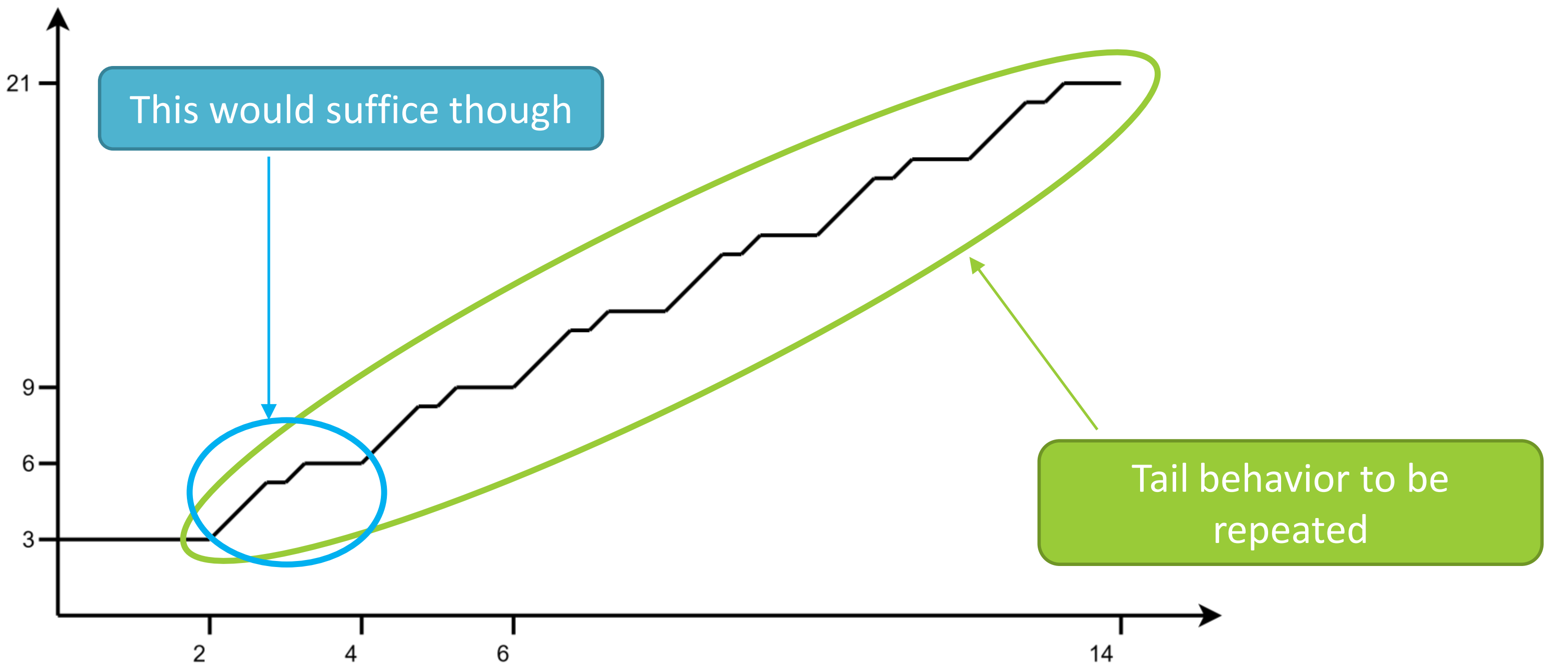
- Computation time depends on number of segments in data structure representations
- $\beta_c = \beta_a \otimes \beta_b \Rightarrow$ Data structure complexity of $\beta_c \gg$ that of β_a and β_b



Convolution performance

Algorithms in the literature scale poorly with the number of chained operations

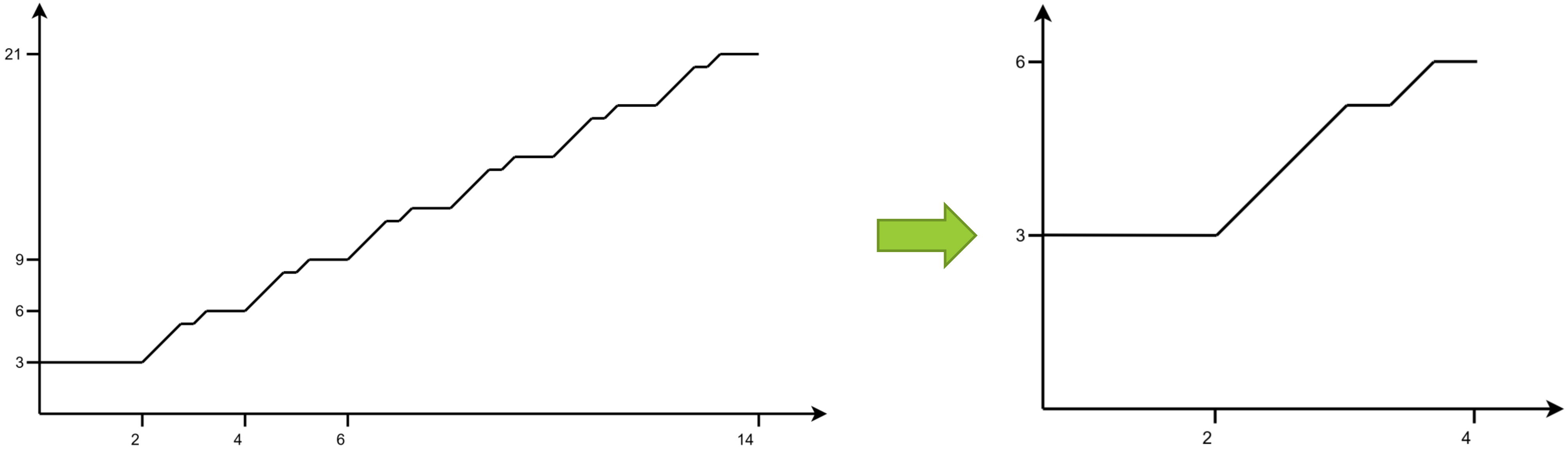
- Computation time depends on number of segments in data structure representations
- $\beta_c = \beta_a \otimes \beta_b \Rightarrow$ Data structure complexity of $\beta_c \gg$ that of β_a and β_b



Convolution performance

Algorithms in the literature scale poorly with the number of chained operations

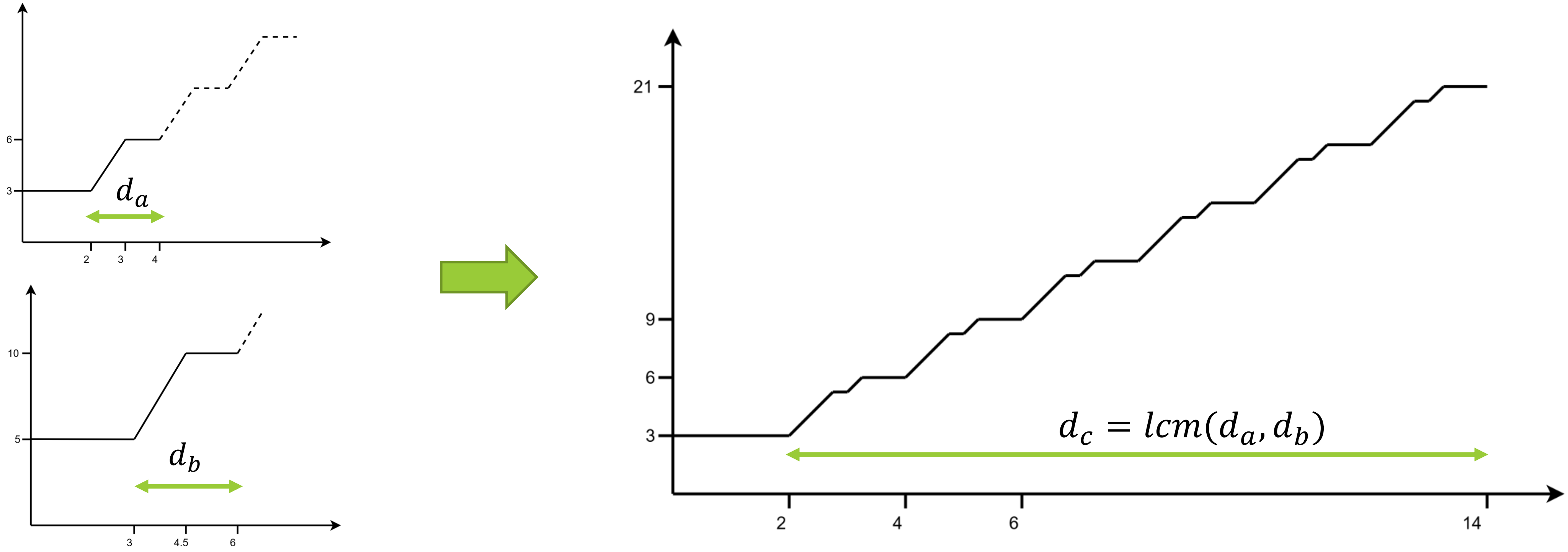
- Computation time depends on number of segments in data structure representations
- $\beta_c = \beta_a \otimes \beta_b \Rightarrow$ Data structure complexity of $\beta_c \gg$ that of β_a and β_b
- After each convolution, we try reducing the result to a minimal representation
  - We observed speed-up from 10 minutes to 10 seconds



Convolution performance

This optimization affects chains of convolutions, but not the single one

Computations can still get out of hand, as they depend on numerical properties

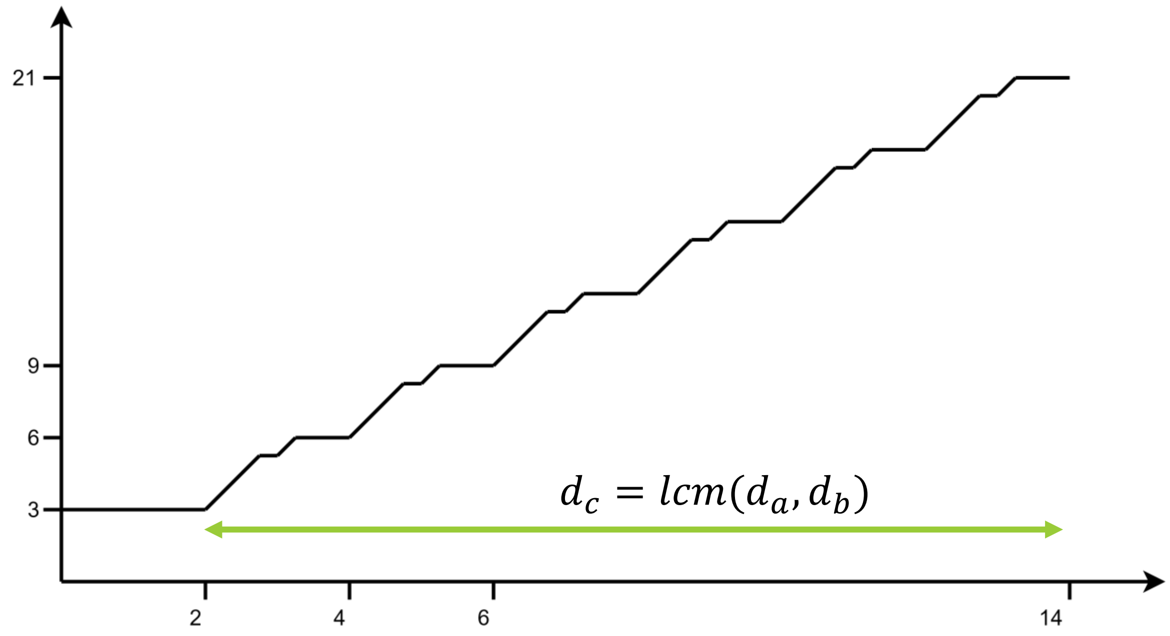


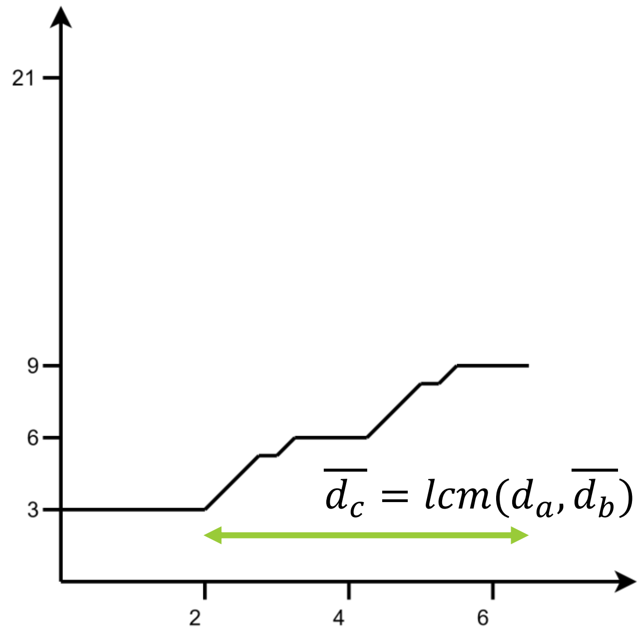
Convolution performance

This affects chains of convolutions, but not the single one

Computations can still get out of hand, as they depend on numerical properties

Clever lower-bounding may allow to avoid this





Summing up

Studying networks of flow-controlled links leads to intractable computations

Delay-rate lower-bounding fixes this, but crucial information on the system is lost in the process

Our contributions so far:

- Transform these recursive closures into a series of convolutions
- Algorithmic improvements for chains of generic convolutions

Work to be done

- How to recognize at a glance unacceptably heavy convolutions, and avoid them via approximation
- Exploit the computed curves – known methods for FIFO study assume *convex* service curves
- Systematic performance study

Thanks for the attention