Catching Corner Cases in Network Calculus – Flow Segregation Can Improve Accuracy

Steffen Bondorf

NUS National University of Singapore

Paul Nikolaus, Jens B. Schmitt

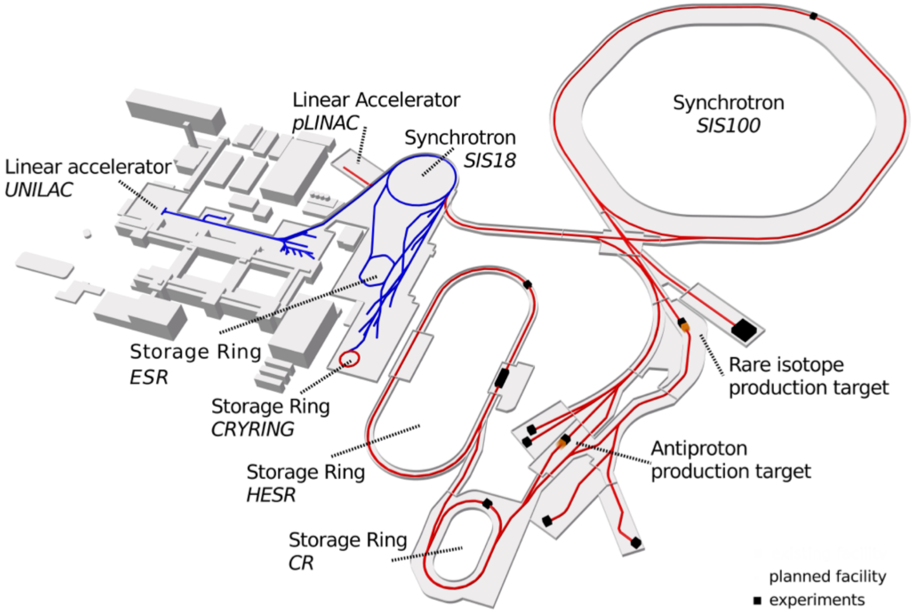
disco DISTRIBUTED COMPUTER SYSTEMS @ Technische Universität Kaiserslautern

MMB 2018 / WoNeCa-4

Worst-Case Performance Analysis

- Hard real-time systems have to respond within finite and specified deadlines
  - Crucial for certification of safety-critical systems

Source: nextreflexdc.com

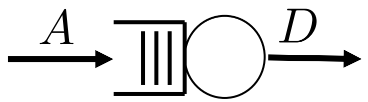


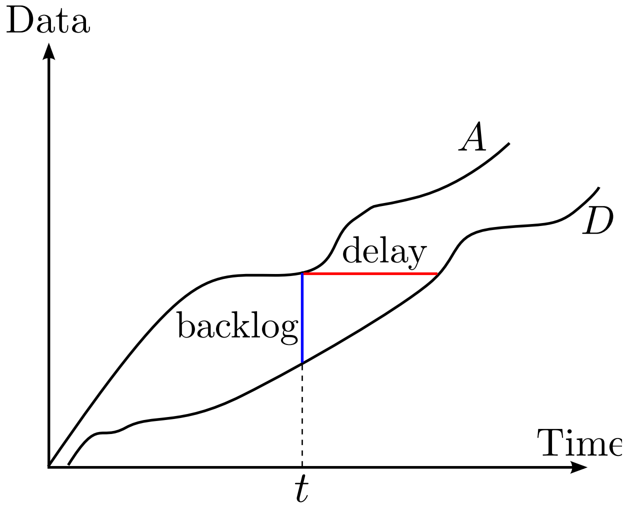
The FAIR Accelerator. Source: [Fitzek17]

- Integration into the design phase of a system
- Requirements:
  - Results should be accurate to prevent over-provisioning
  - Analyzing/Ranking of many different configurations should be fast

Deterministic Network Calculus: Arrivals and Service

- Worst-case bounds on the behavior: cumulative arrivals and service [Cruz91]



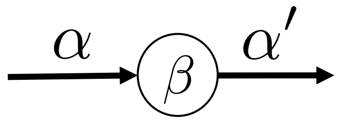


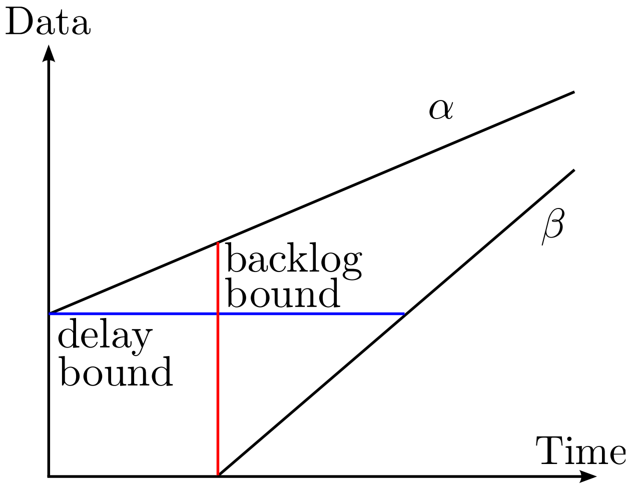
Arrival Curve α:

$$\alpha(s) \geq A(t) - A(t-s)\,\forall s \leq t$$

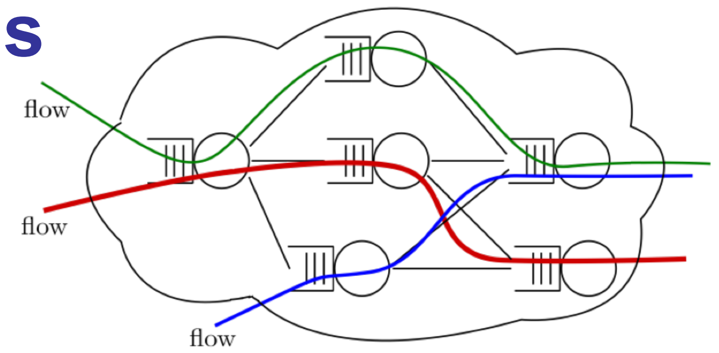
Strict Service Curve β:

A server is said to offer a strict service curve β if, during any backlogged period of duration u, the output of the system is at least equal to $\beta(u)$.





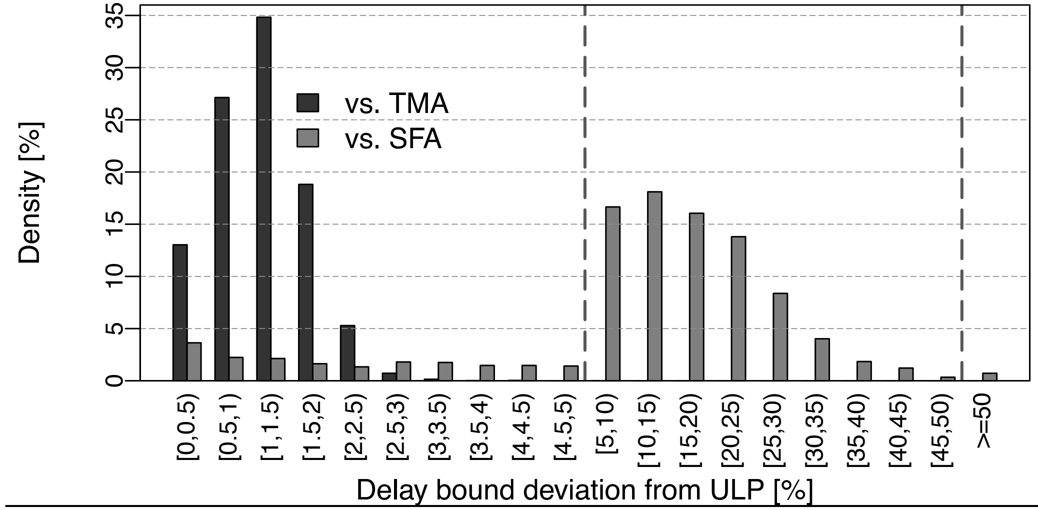
Deterministic Network Calculus: Network Analysis



TFA [Cruz91]	SFA [LeBoudec01]	PMOO [Fidler03]	OBA [Schmitt08]	LP, ULP [Bouillard10]	TMA [Bondorf17]
1991	2001	2003	2008	2010	2017
algebraic DNC			optimization-based DNC		algebraic DNC

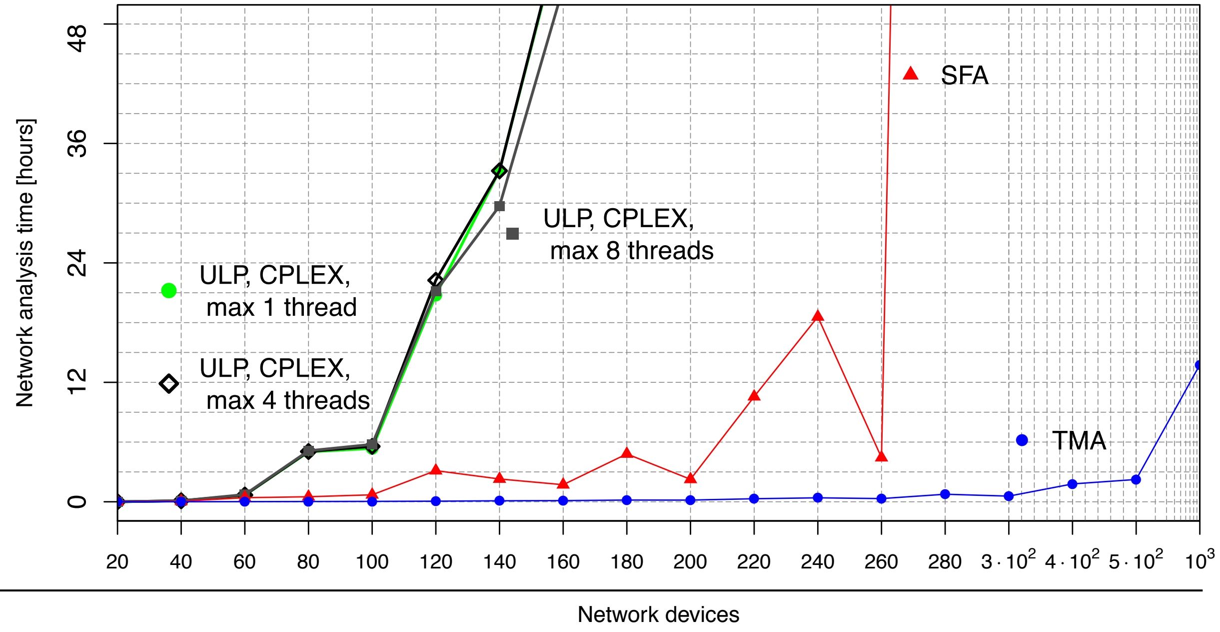
Quality

- Median deviation from ULP 1.142%
- Some outliers at double that ☹



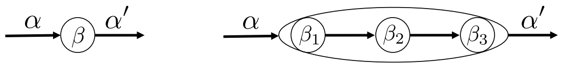
Cost

- TMA is the fastest, scales best



DNC: Compositional Feed-Forward Analysis

Algebraic DNC's Composition of Local Results



- Basic (min,+)-algebraic operations
  1. Output bound $(\alpha \oslash \beta)(d) = \sup_{u \geq 0} \{\alpha(d+u) - \beta(u)\} =: \alpha'(d)$
  2. Aggregation of flows $(\alpha_1 + \alpha_2)(d) = \alpha_1(d) + \alpha_2(d)$
  3. Concatenation of servers (= **tandems**) $(\beta_1 \otimes \beta_2)(d) = \inf_{0 \leq s \leq d} \{\beta_1(d-s) + \beta_2(s)\} = \beta_{\langle 1,2 \rangle}$
  4. Left-over service curve (**server**) $(\beta \ominus \alpha)(d) = \sup_{0 \leq u \leq d} \{(\beta - \alpha)(u)\} =: \beta^{\text{l.o.}}$
  5. Left-over service curve (**tandem**) (considers entanglement of cross-flows) $\beta_{\langle 1,2 \rangle} \ominus \alpha = \beta^{\text{l.o.}}_{\langle 1,2 \rangle}$

- Rules for composition of operations on curves
  - Retain the worst case
  - Impose a *composition penalty*
  - Leave some degrees of freedom

Principle	Tandem Analysis				Network Analysis		
	TFA	SFA	PMOOA	OBA	TMA	ULP	LP
Agg	✓	(✓)	(✓)	(✓)	(✓)	✓	✓
PBOO	✗	✓	✓	✓	✓	✓	✓
PMOO	✗	✗	✓	✓	(✓)	✓	✓
Order	✗	✓	✗	✓	(✓)	✓	✓
OBC	✓	✗	*NA*		✓	*NA*	
PSOO	*NA*	✗[1]	*NA*		(✓)	(✓)	✓
SegrAB	*NA*				✗	*NA*	
AggrAB	*NA*				✓	*NA*	
good scaling	✓	✓	✓	✗	✓	✗	✗

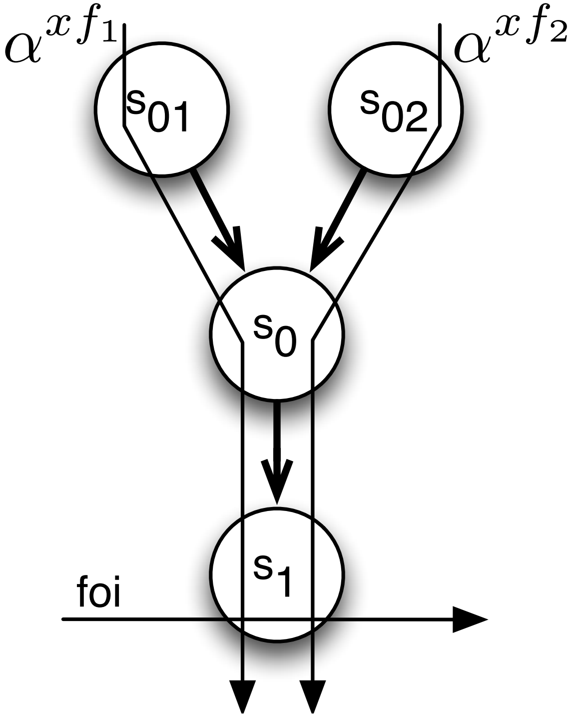
Feature matrix of all current, mutually exclusive DNC analyses.
Principle implementation: ✓ full, (✓) partial/selective, ✗ none, *NA* not applicable.

- TMA tries to minimize the composition penalty
  - exhaustively search for the best solution among alternatives
  - The paper provides an overview

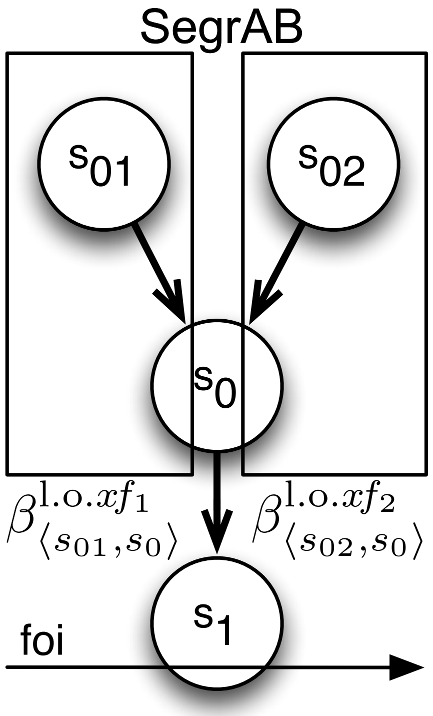
Is TMA Really an Exhaustive Search?

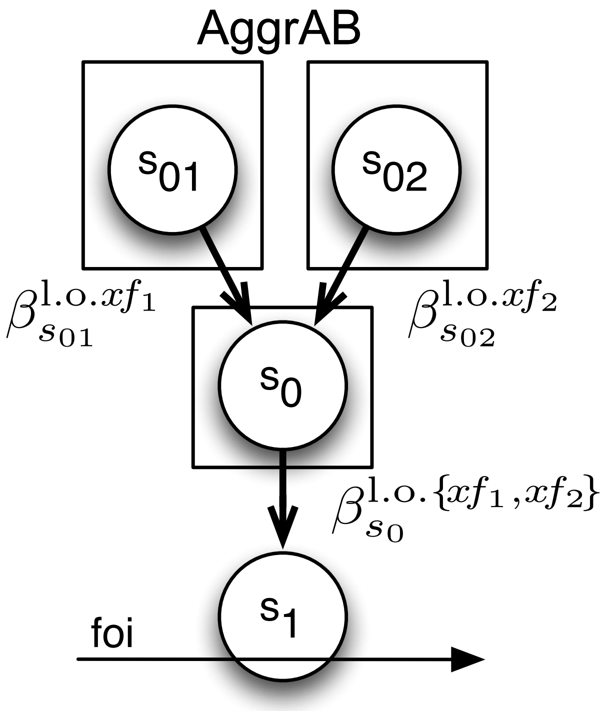
Principle	Network Analysis		
	TMA	ULP	LP
SegrAB	✗	*NA*	
AggrAB	✓	*NA*	

- Why is there an ✗ ?
- When bounding the arrivals of cross-flows, we prefer aggregation (AggrAB) over segregation (SegrAB).
- Reason: [Bondorf15]



Basic Building Block



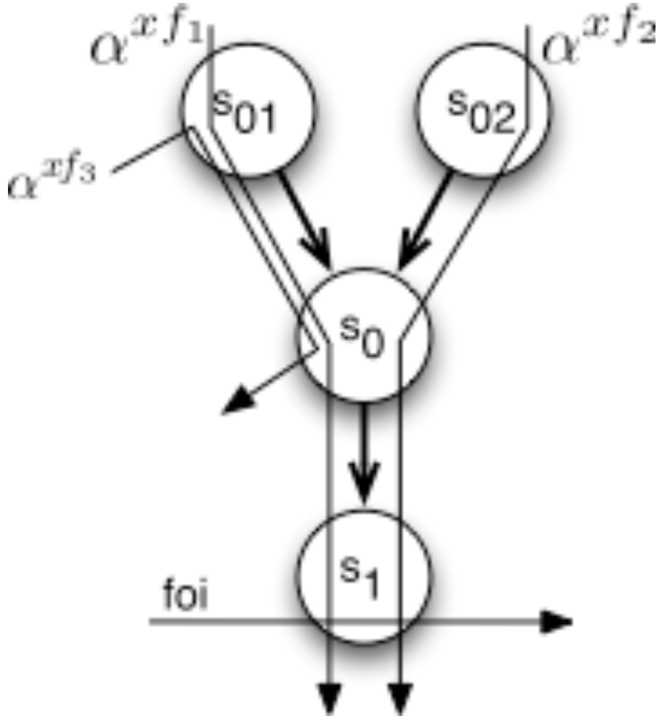


Segregation: mutually incompatible priority assignment at s_0

Is TMA Really an Exhaustive Search?

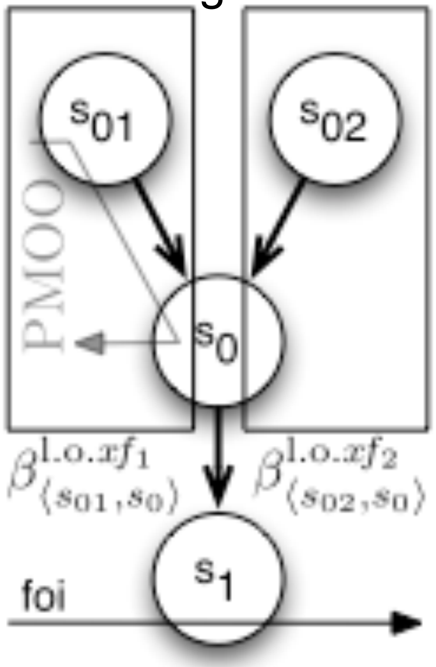
Principle	Network Analysis		
	TMA	ULP	LP
SegrAB	✗	*NA*	
AggrAB	✓	*NA*	

- Why is there an ✗ ?
- When bounding the arrivals of cross-flows, we prefer aggregation (AggrAB) over segregation (SegrAB).
- Insight:

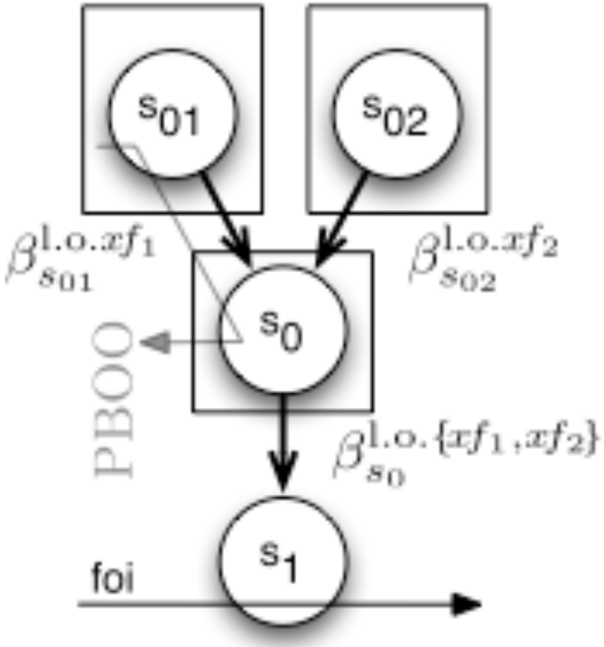


Better Basic Building Block

SegrAB



AggrAB



Segregation: mutually incompatible priority assignment at s_0
but the PMOO principle can be used [Schmitt08]

Can Segregation Improve the Delay Bound Accuracy?

- **Yes.** *Proof:*

  (in the paper)

Proposition 1. *Cross-flow segregation paired with a PMOO analysis is able to obtain lower bounds on flow arrivals than its aggregating counterpart. That is, none of these arrival bounding alternatives is a dominating approach.*

Proof. The superiority of AggrAB employing PBOO over the segregated version has been discussed in [5]. For the case that AggrAB implements either PBOO or PMOO and SegrAB implements PMOO, we give an example where cross-flow segregation yields a better result. Let us therefore consider the setting as in Figure 1 with token-bucket arrivals ($\mathcal{F}_{\mathrm{TB}}$) and rate-latency service ($\mathcal{F}_{\mathrm{RL}}$). First, we derive the arrival bound when aggregating cross flows:

$$\begin{aligned}
&\alpha_{S_1}^{\mathrm{Aggr}\{xf_1,xf_2\}}\\
&\quad= \alpha_{S_0}^{\{xf_1,xf_2\}} \oslash \beta_{S_0}^{\mathrm{l.o.}\{xf_1,xf_2\}}\\
&\quad= \left(\left(\alpha^{xf_1} \oslash \beta_{S_{01}}^{\mathrm{l.o.}xf_1}\right) + \left(\alpha^{xf_2} \oslash \beta_{S_{02}}^{\mathrm{l.o.}xf_2}\right)\right) \oslash \left(\beta_{S_0} \ominus \alpha_{S_0}^{xxf_1}\right)\\
&\quad= \left(\left(\alpha^{xf_1} \oslash \left(\beta_{S_{01}} \ominus \alpha^{xf_3}\right)\right) + \left(\alpha^{xf_2} \oslash \beta_{S_{02}}\right)\right) \oslash \left(\beta_{S_0} \ominus \left(\alpha^{xf_3} \oslash \beta_{S_{01}}^{\mathrm{l.o.}xf_3}\right)\right)\\
&\quad= \left(\left(\alpha^{xf_1} \oslash \left(\beta_{S_{01}} \ominus \alpha^{xf_3}\right)\right) + \left(\alpha^{xf_2} \oslash \beta_{S_{02}}\right)\right) \oslash \left(\beta_{S_0} \ominus \left(\alpha^{xf_3} \oslash \beta_{S_{01}}\right)\right)\\
&\quad= \left(\gamma_{r_1,b_1} \oslash \left(\beta_{R_{01},T_{01}} \ominus \gamma_{r_3,b_3}\right)\right)\\
&\qquad\quad + \left(\left(\gamma_{r_2,b_2} \oslash \beta_{R_{02},T_{02}}\right) \oslash \left(\beta_{R_0,T_0} \ominus \left(\gamma_{r_3,b_3} \oslash \beta_{R_{01},T_{01}}\right)\right)\right).
\end{aligned}$$

We continue with

$$\begin{aligned}
&\alpha_{S_1}^{\mathrm{Aggr}\{xf_1,xf_2\}}\\
&= \left(\left(\gamma_{r_1,b_1} \oslash \beta_{R_{01}-r_3,\frac{R_{01}\cdot T_{01}+b_3}{R_{01}-r_3}}\right) + \gamma_{r_2,b_2+r_2\cdot T_{02}}\right) \oslash \left(\beta_{R_0,T_0} \ominus \gamma_{r_3,b_3+r_3\cdot T_{01}}\right)\\
&= \left(\gamma_{r_1,b_1+r_1\cdot\frac{R_{01}\cdot T_{01}+b_3}{R_{01}-r_3}} + \gamma_{r_2,b_2+r_2\cdot T_{02}}\right) \oslash \beta_{R_0-r_3,\frac{R_0\cdot T_0+b_3+r_3\cdot T_{01}}{R_0-r_3}}\\
&= \gamma_{r_1+r_2,b_1+b_2+r_1\cdot\frac{R_{01}\cdot T_{01}+b_3}{R_{01}-r_3}+r_2\cdot T_{02}} \oslash \beta_{R_0-r_3,\frac{R_0\cdot T_0+b_3+r_3\cdot T_{01}}{R_0-r_3}}\\
&= \gamma_{r_1+r_2,b_1+b_2+r_1\cdot\frac{R_{01}\cdot T_{01}+b_3}{R_{01}-r_3}+r_2\cdot T_{02}} + (r_1+r_2)\cdot\frac{R_0\cdot T_0+b_3+r_3\cdot T_{01}}{R_0-r_3}.
\end{aligned}$$

At this point, please note that the PBOO property is preserved as b_1 and b_2 occur only once. The PMOO property, on the other hand, does not hold anymore, as b_3 is included twice.

The segregated version yields

$$\begin{aligned}
\alpha_{S_1}^{\mathrm{Segr}\{xf_1,xf_2\}} &= \alpha_{S_1}^{xf_1} + \alpha_{S_1}^{xf_2}\\
&= \left(\alpha^{xf_1} \oslash \beta_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1}\right) + \left(\alpha^{xf_2} \oslash \beta_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2}\right)\\
&= \left(\gamma_{r_1,b_1} \oslash \beta_{R_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1},T_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1}}\right) + \left(\gamma_{r_2,b_2} \oslash \beta_{R_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2},T_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2}}\right)\\
&= \gamma_{r_1,b_1+r_1\cdot T_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1}} + \gamma_{r_2,b_2+r_2\cdot T_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2}}\\
&= \gamma_{r_1+r_2,b_1+b_2+r_1\cdot T_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1}+r_2\cdot T_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2}}.
\end{aligned}$$

Using

$$\begin{aligned}
T_{\langle S_{01},S_0\rangle}^{\mathrm{l.o.}xf_1} &= T_{01} + T_0 + \frac{b_2+b_3+r_3\cdot T_{01}+(r_2+r_3)\cdot T_0}{(R_{01}-r_3)\wedge(R_0-r_2-r_3)},\\
T_{\langle S_{02},S_0\rangle}^{\mathrm{l.o.}xf_2} &= T_{02} + T_0 + \frac{b_1+b_3+(r_1+r_3)\cdot T_0}{R_{02}\wedge(R_0-r_1-r_3)}
\end{aligned}$$

computed with [17] gives us

$$\begin{aligned}
\alpha_{S_1}^{\mathrm{Segr}\{xf_1,xf_2\}} &= \gamma_{r_1+r_2,b_1+b_2+r_1\cdot\left(T_{01}+T_0+\frac{b_2+b_3+r_3\cdot T_{01}+(r_2+r_3)\cdot T_0}{(R_{01}-r_3)\wedge(R_0-r_2-r_3)}\right)+r_2\cdot\left(T_{02}+T_0+\frac{b_1+b_3+(r_1+r_3)\cdot T_0}{R_{02}\wedge(R_0-r_1-r_3)}\right)}\\
&= \gamma_{r_1+r_2,b_1+b_2+r_1\cdot T_{01}+r_1\cdot T_0+r_1\cdot\frac{b_2+b_3+r_3\cdot T_{01}+(r_2+r_3)\cdot T_0}{(R_{01}-r_3)\wedge(R_0-r_2-r_3)}+r_2\cdot T_{02}+r_2\cdot T_0+r_2\cdot\frac{b_1+b_3+(r_1+r_3)\cdot T_0}{R_{02}\wedge(R_0-r_1-r_3)}}
\end{aligned}$$

where the PMOO principle is implemented per flow xf_1 and xf_2. Yet, overall, b_3 appears twice. We bound all arrivals with equal token buckets and continue by comparing burst terms. As we are free to choose parameters, we set $T_0 = T_{01} = T_{02} = b_1 = b_2 = 0$ and the arrival rates to be homogeneous ($r_1 = r_2 = r_3 =: r > 0$). We further assume the burst term b_3 to be >0. Assume now that the claim does not hold true yielding for the burst term

$$b_{S_1}^{\mathrm{Aggr}\{xf_1,xf_2\}} < b_{S_1}^{\mathrm{Segr}\{xf_1,xf_2\}} \tag{1}$$

$$\begin{aligned}
\Leftrightarrow\quad & r\cdot\frac{b_3}{R_{01}-r} + r\cdot\frac{b_3}{R_0-r} + r\cdot\frac{b_3}{R_0-r}\\
&< r\cdot\frac{b_3}{(R_{01}-r)\wedge(R_0-2r)} + r\cdot\frac{b_3}{R_{02}\wedge(R_0-2r)}\\
\Leftrightarrow\quad & \frac{1}{R_{01}-r} + \frac{2}{R_0-r} < \frac{1}{(R_{01}-r)\wedge(R_0-2r)} + \frac{1}{R_{02}\wedge(R_0-2r)}.
\end{aligned}$$

In order to contradict the claim and prove the proposition, it is sufficient to give an example where Equation (1) cannot hold. For this, see Example 1 below.
□

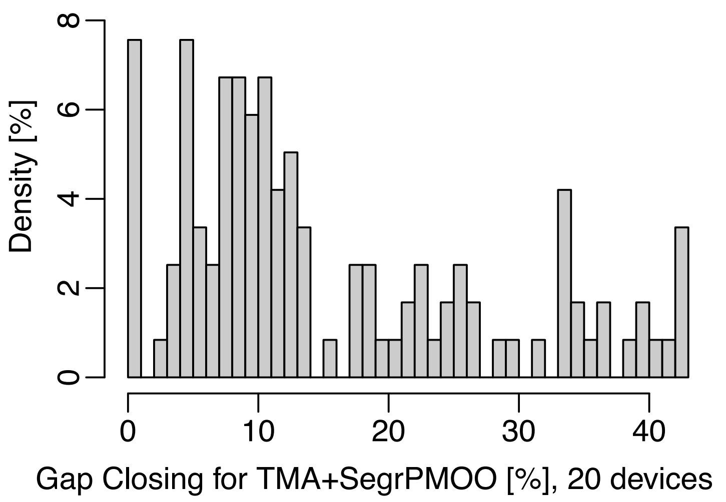
Can Segregation Close the Gap Between TMA and ULP?

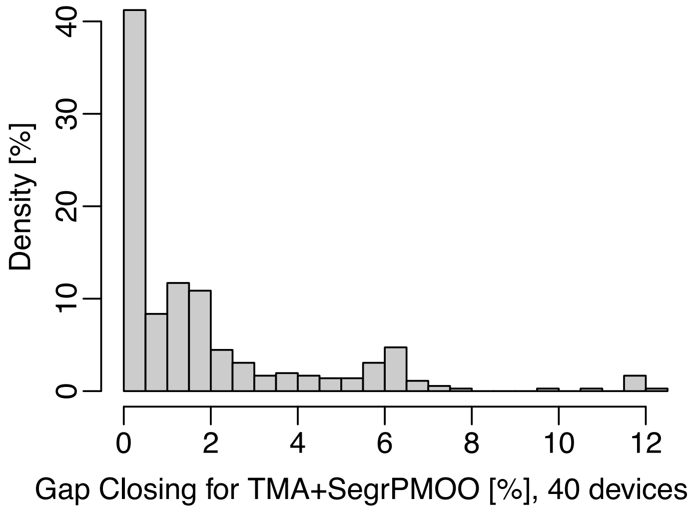
- 2 sample networks: 152 (left), 472 flows (right)
- 78% and 72% of delay bounds improved
- Reduction of the gap between TMA and ULP

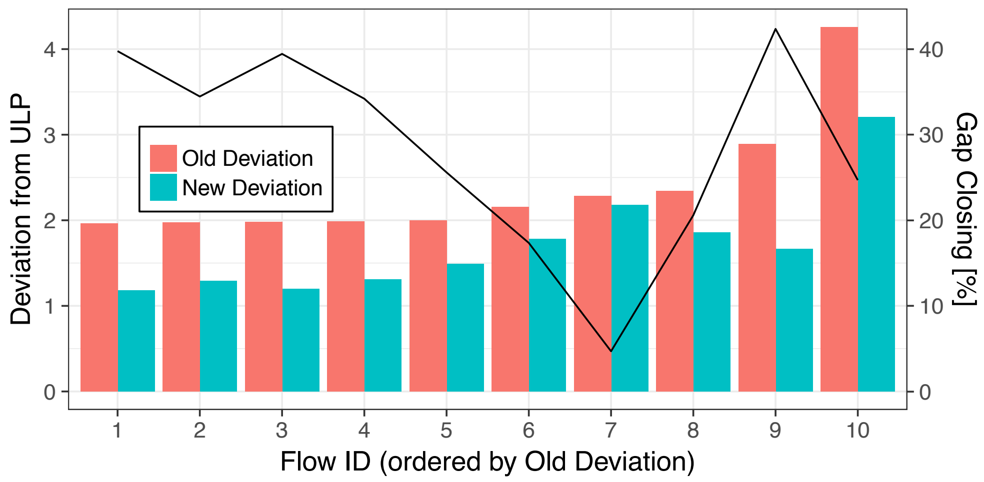
  - Max: 49.92%
  - Mean: 15.5%
  - Distribution:

  - Max: 12.0%
  - Mean: 1.87%
  - Distribution:

- Catching Outliers?
- 10 flows TMA performed worse compared to ULP
  - 8/10 see >20% improvement
  - 1/10 improves <10%

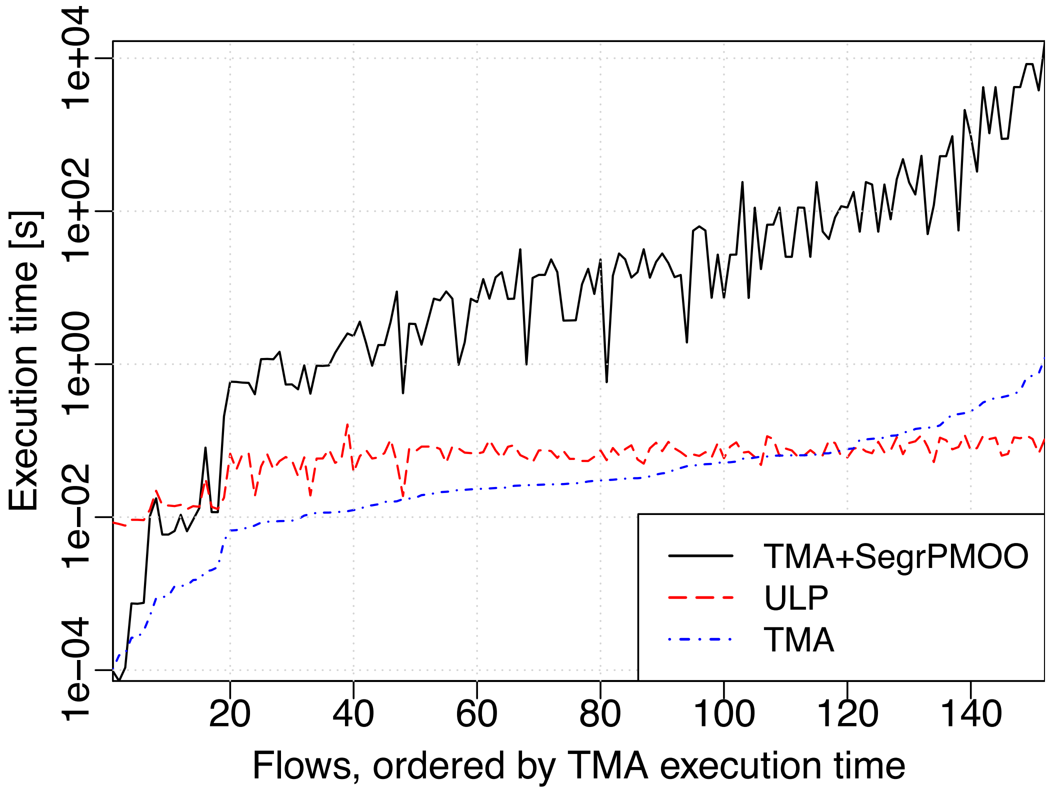


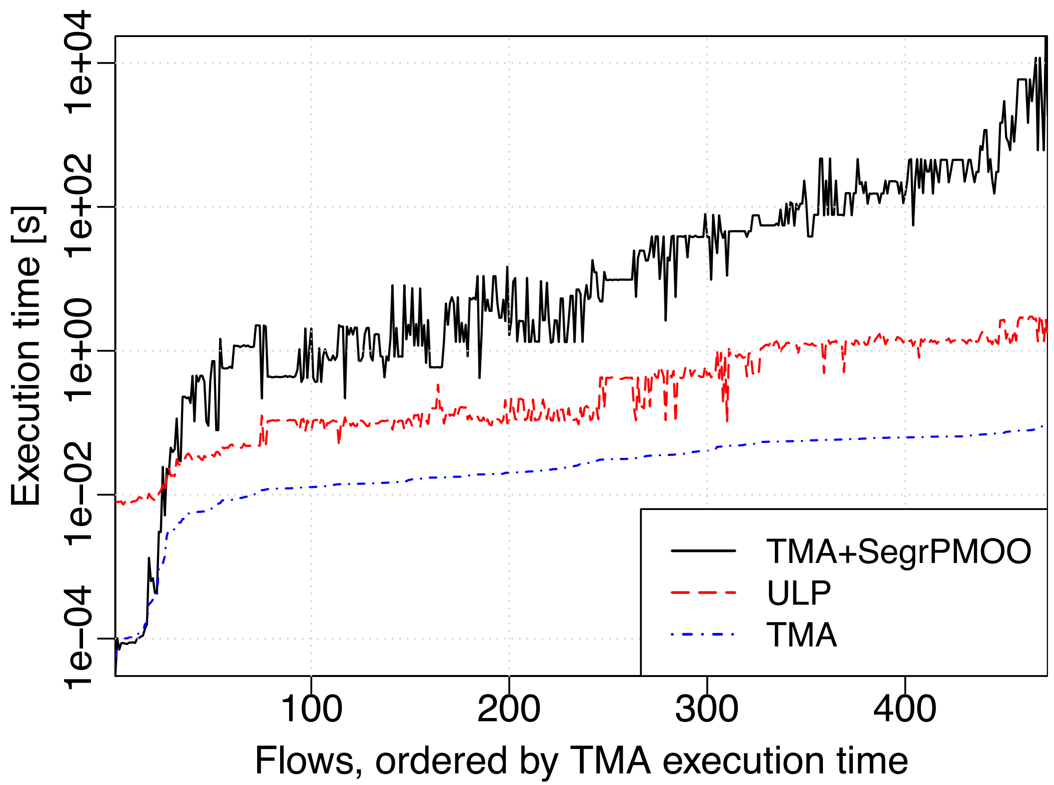




Can Segregation Close the Gap Between TMA and ULP?

- Computational Effort?





Conclusion

- Improving accuracy of algebraic DNC is still possible
- Catching outliers that are highly impacted by special cases in the analysis

- Cost of the current exhaustive search is too high
  - SegrAB is not the only extension that suffers from this problem [Bondorf17-1]
- Be smarter and identify the best approach in advance? ☺

Thank You!

References

[Bouillard10] A. Bouillard, L. Jouhet, and E. Thierry. Tight Performance Bounds in the Worst- Case Analysis of Feed-Forward Networks. In Proc. of IEEE INFOCOM, 2010.

[Bouillard14] A. Bouillard. Algorithms and Efficiency of Network Calculus. Habilitation thesis, ENS, 2014.

[Bondorf15] S. Bondorf and J. B. Schmitt. Calculating Accurate End-to-End Delay Bounds – You Better Know Your Cross-Traffic. In Proc. of ValueTools, 2015.

[Bondorf17] S. Bondorf, P. Nikolaus, and J. B. Schmitt. Quality and Cost of Deterministic Network Calculus – Design and Evaluation of an Accurate and Fast Analysis. In Proc. of ACM SIGMETRICS, 2017.

[Bondorf17-1] S. Bondorf. Better Bounds by Worse Assumptions – Improving Network Calculus Accuracy by Adding Pessimism to the Network Model. In Proc. of IEEE ICC, 2017.

[Cruz91] R. L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. In IEEE Transactions on Information Theory, 1991, and R. L. Cruz. A Calculus for Network Delay, Part II: Network Analysis. In IEEE Transactions on Information Theory, 1991.

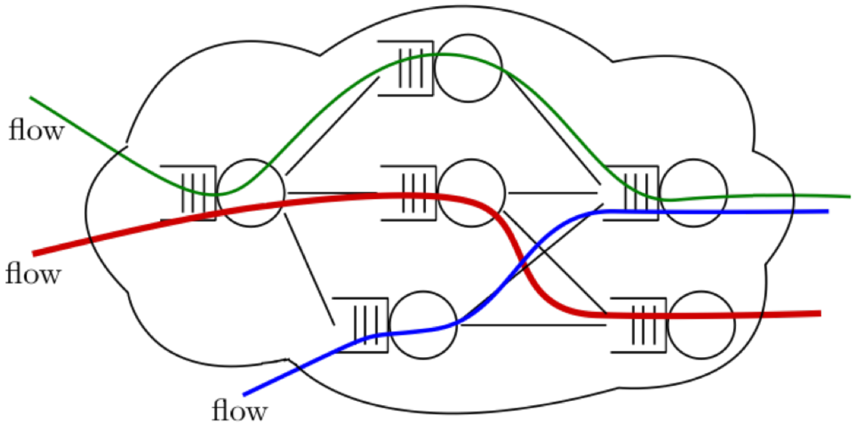
[Fidler03] M. Fidler. Extending the Network Calculus Pay Bursts Only Once Principle to Aggregate Scheduling. In Proc. of QoS-IP, 2003.

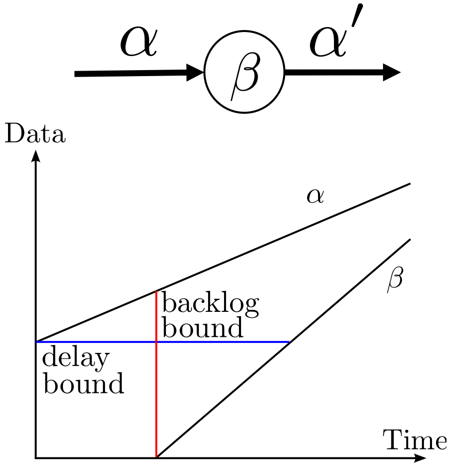
[Fitzek17] J. Fitzek, H. Hüther, R.Müller and A. Schaller. First Production Use of the New Settings Management for FAIR. In Proc. ICALEPCS 2017.

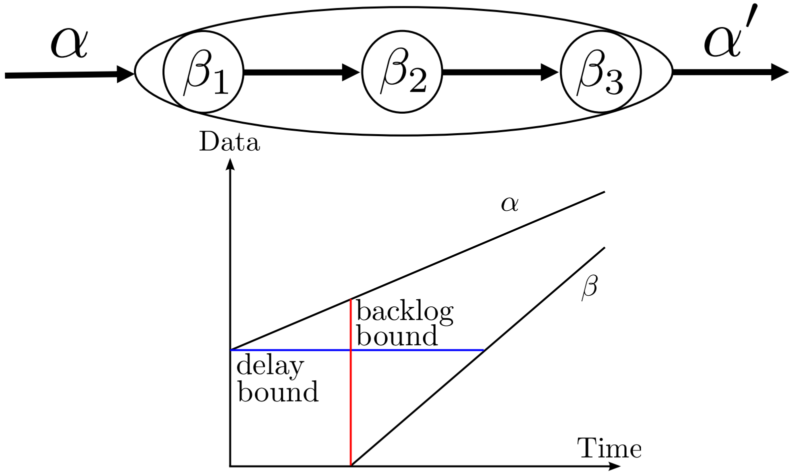
[LeBoudec01] J.-Y. Le Boudec and P. Thiran. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. Springer, 2001.

[Schmitt08] J. B. Schmitt, F. A. Zdarsky, and M. Fidler. Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch ... In Proc. of IEEE INFOCOM, 2008.

Algebraic DNC’s Composition of Local Results







tandem is our basic unit of operation

- Basic (min,+)-algebraic operations
  1. Output bound $(\alpha \oslash \beta)(d) = \sup_{u \geq 0} \{\alpha(d+u) - \beta(u)\} =: \alpha'(d)$
  2. Aggregation of flows $(\alpha_1 + \alpha_2)(d) = \alpha_1(d) + \alpha_2(d)$
  3. Concatenation of servers (= **tandems**) $(\beta_1 \otimes \beta_2)(d) = \inf_{0 \leq s \leq d} \{\beta_1(d-s) + \beta_2(s)\} = \beta_{\langle 1,2 \rangle}$
  4. Left-over service curve (**server**) $(\beta \ominus \alpha)(d) = \sup_{0 \leq u \leq d} \{(\beta - \alpha)(u)\} =: \beta^{\text{l.o.}}$
  5. Left-over service curve (**tandem**) (considers entanglement of cross-flows) $\beta_{\langle 1,2 \rangle} \ominus \alpha = \beta^{\text{l.o.}}_{\langle 1,2 \rangle}$

(4. and 5.) Arbitrary multiplexing: no FIFO assumptions