DISTRIBUTED COMPUTER SYSTEMS

Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Per-Flow Backlog Bounds for a FIFO Server Beyond Token-Bucket-Constrained Arrivals

Lukas Wildberger, Anja Hamscher, Jens B. Schmitt

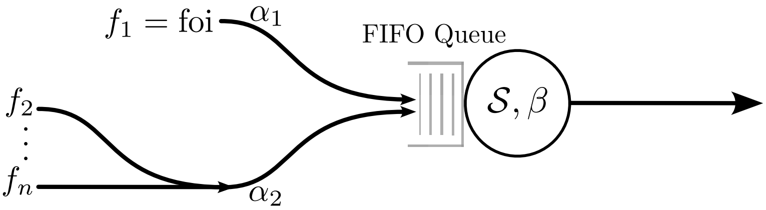
June 5th 2025

These results will be published in QEST + FORMATS 2025

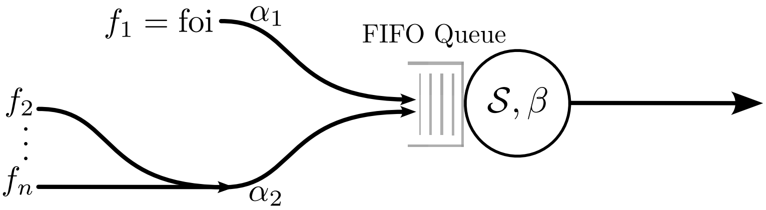
Outline

Motivation

What We're Up To



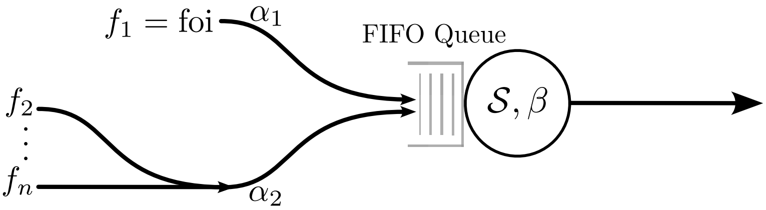
What We're Up To



How to choose the free parameter θ in order to minimize the

- Backlog bound

What We're Up To



How to choose the free parameter θ in order to minimize the

- Backlog bound
- We have to solve the mathematical problem given by: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ v(\alpha_1, \beta_\theta^1) \}$

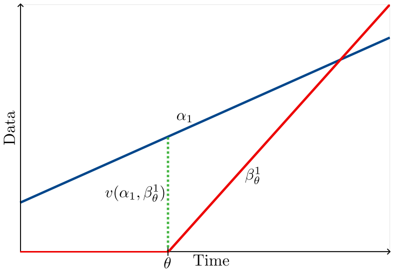
Function Set

Function Set

- Literature results primarily for token-bucket arrival and rate-latency service curves [Le Boudec and Thiran, 2001]

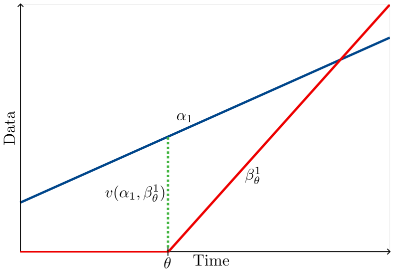
Function Set

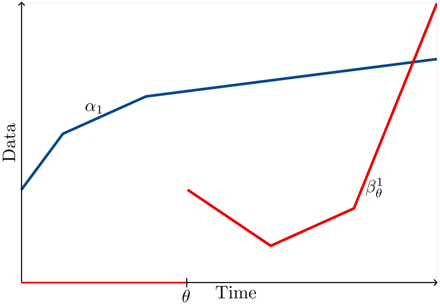
- Literature results primarily for token-bucket arrival and rate-latency service curves [Le Boudec and Thiran, 2001]



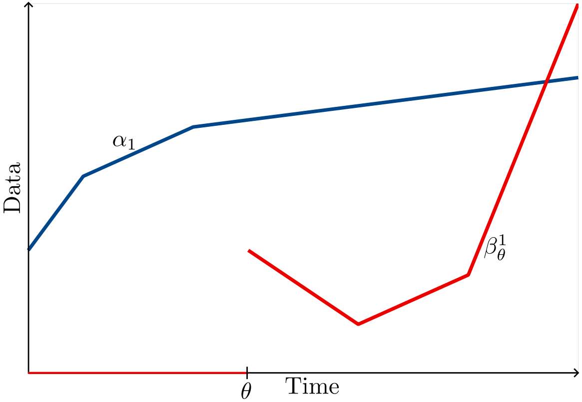
Function Set

- Literature results primarily for token-bucket arrival and rate-latency service curves [Le Boudec and Thiran, 2001]

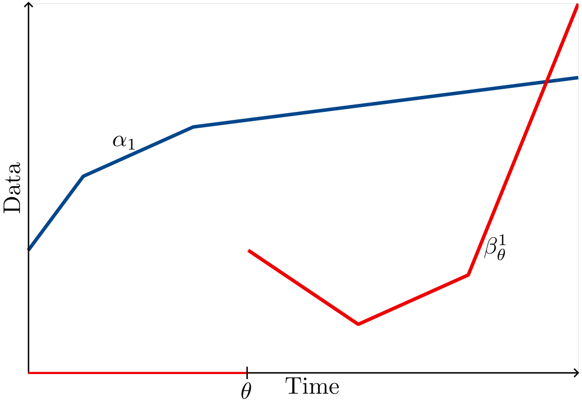




Function Set - PWL Concave/Convex Curves

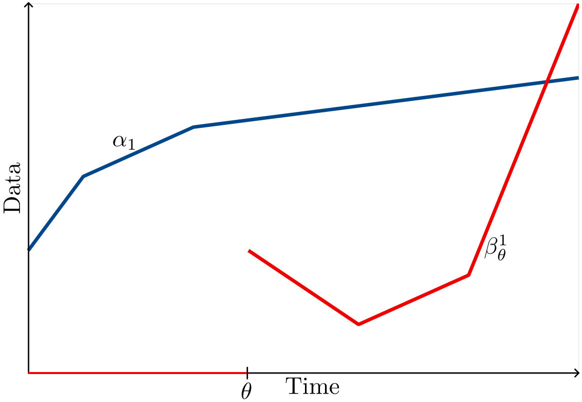


Function Set - PWL Concave/Convex Curves

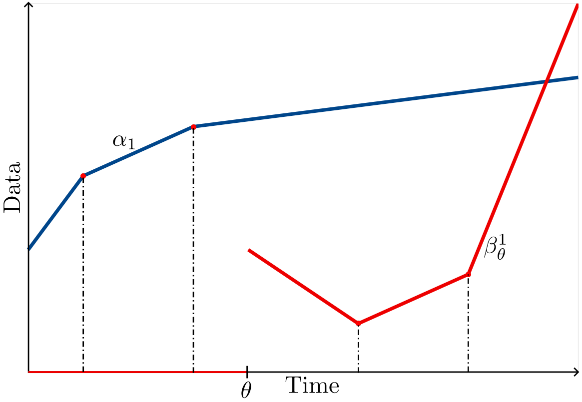


- FIFO residual service curve is also a service curve for the foi **even if it is not wide-sense increasing** [Bouillard et al., 2018]

Terminology



Terminology



- We call time where the slope of the linear segment changes "breakpoint"
  - In the literature also referred to as "intersetion point" or "changepoint"

Backlog Bound

Backlog Bound: Non-Wide-Sense Increasing Property

Backlog Bound: Non-Wide-Sense Increasing Property

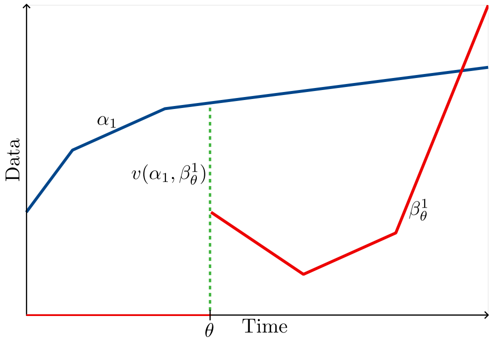
Lemma

Let $\alpha, \beta \in \mathcal{F}$ be given. Then it holds that $v(\alpha, \beta) = v(\alpha, \beta_\downarrow)$, with $\beta_\downarrow := \beta \overline{\oslash} 0$.

Backlog Bound: Non-Wide-Sense Increasing Property

Lemma

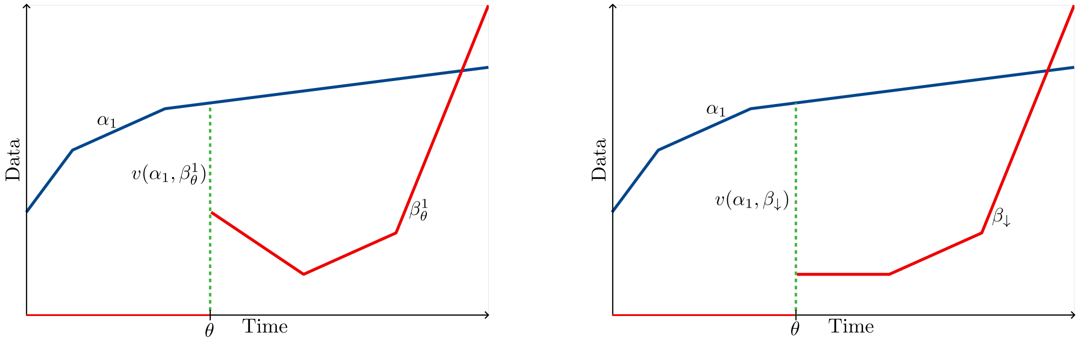
Let $\alpha, \beta \in \mathcal{F}$ be given. Then it holds that $v(\alpha, \beta) = v(\alpha, \beta_\downarrow)$, with $\beta_\downarrow := \beta \overline{\oslash} 0$.



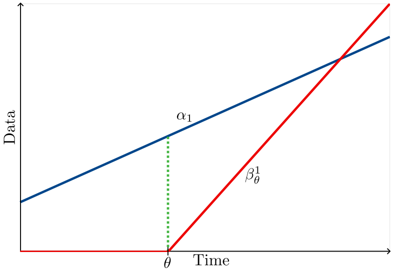
Backlog Bound: Non-Wide-Sense Increasing Property

Lemma

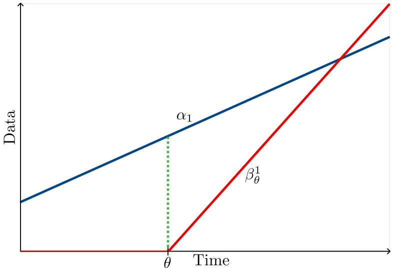
Let $\alpha, \beta \in \mathcal{F}$ be given. Then it holds that $v(\alpha, \beta) = v(\alpha, \beta_{\downarrow})$, with $\beta_{\downarrow} := \beta \overline{\oslash} 0$.

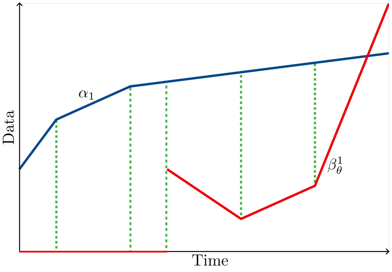


What Is The Challenge With PWL Curves?

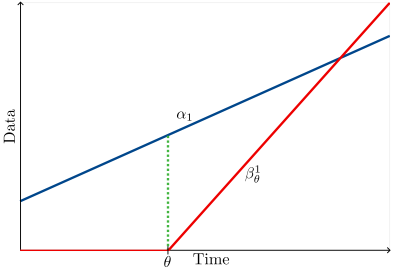


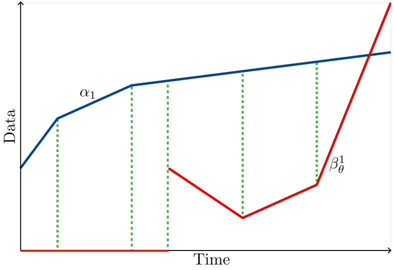
What Is The Challenge With PWL Curves?





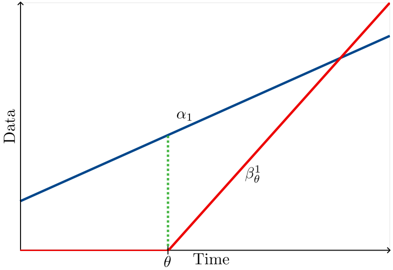
What Is The Challenge With PWL Curves?

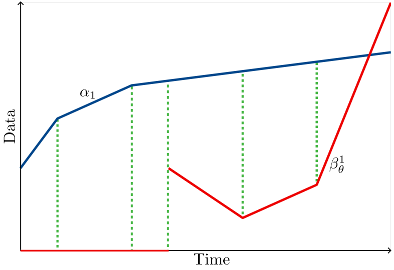




- If we choose smaller/larger θ values the vertical distance will increase at some breakpoints and decrease at others

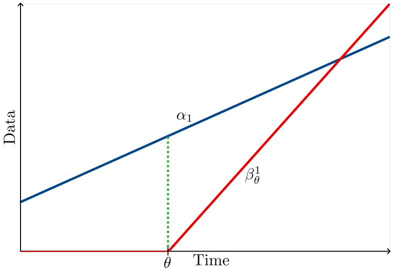
What Is The Challenge With PWL Curves?

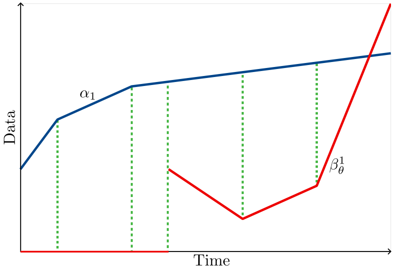




- If we choose smaller/larger θ values the vertical distance will increase at some breakpoints and decrease at others
- We cannot split the problem into token-bucket cases and take the minimum

What Is The Challenge With PWL Curves?





- If we choose smaller/larger θ values the vertical distance will increase at some breakpoints and decrease at others
- We cannot split the problem into token-bucket cases and take the minimum
- We have to take care of all linear sections at once

Backlog-Optimal θ

- Transform the initial problem until we are able to determine a solution

Backlog-Optimal θ

- Transform the initial problem until we are able to determine a solution
- Initial Problem:

$$\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ v\left(\alpha_1, \beta_\theta^1\right) \}$$

Backlog-Optimal θ

- Transform the initial problem until we are able to determine a solution
- Initial Problem:

$$\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ v\left(\alpha_1, \beta_\theta^1\right) \}$$

- Transformed into equivalent one:

$$\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ \max_{t \geq 0} \{ v_t(\theta) \} \}$$

Backlog-Optimal θ

- Transform the initial problem until we are able to determine a solution
- Initial Problem:

$$\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ v\left(\alpha_1, \beta_\theta^1\right) \}$$

- Transformed into equivalent one:

$$\theta_{\text{opt}} = \arg\min_{\theta \geq 0} \{ \max_{t \geq 0} \{ v_t(\theta) \} \}$$

Definition

$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta_\theta^1(t), & \text{if } h(\alpha_2, \beta) \leq \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \geq t > h(\alpha_2, \beta). \end{cases}$$

$v_t(\theta)$ Curves

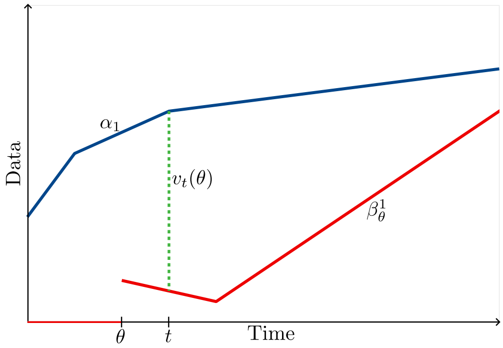
Definition

$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta_\theta^1(t), & \text{if } h(\alpha_2, \beta) \leq \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \geq t > h(\alpha_2, \beta). \end{cases}$$

$v_t(\theta)$ Curves

Definition

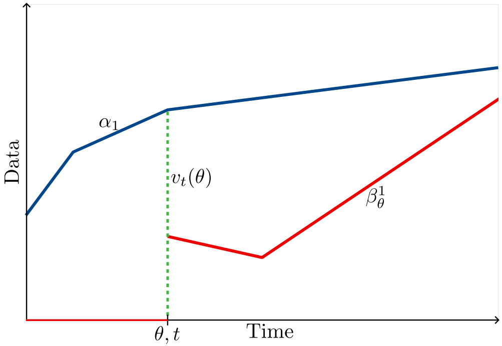
$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta_\theta^1(t), & \text{if } h(\alpha_2, \beta) \le \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \ge t > h(\alpha_2, \beta). \end{cases}$$



$v_t(\theta)$ Curves

Definition

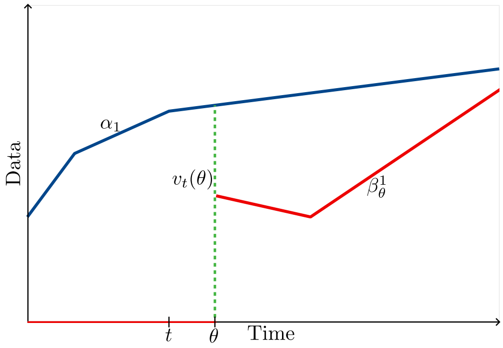
$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta^1_\theta(t), & \text{if } h(\alpha_2, \beta) \le \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \ge t > h(\alpha_2, \beta). \end{cases}$$



$v_t(\theta)$ Curves

Definition

$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta^1_\theta(t), & \text{if } h(\alpha_2, \beta) \le \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \ge t > h(\alpha_2, \beta). \end{cases}$$



Reduce And Discretize The Search Space

- Problem to solve: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0}\{\max_{t \geq 0}\{v_t(\theta)\}\}$

Reduce And Discretize The Search Space

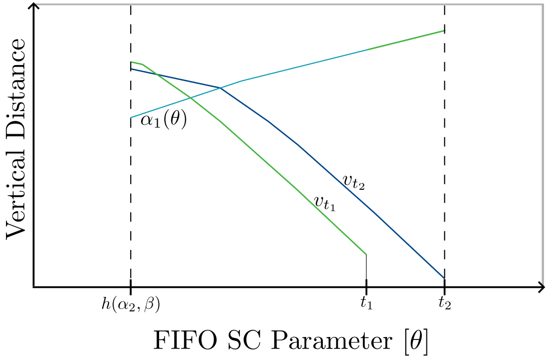
- Problem to solve: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0}\{\max_{t \geq 0}\{v_t(\theta)\}\}$
- Get a lower bound (> 0) and an upper bound for the choice of θ

Reduce And Discretize The Search Space

- Problem to solve: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0}\{\max_{t \geq 0}\{v_t(\theta)\}\}$
- Get a lower bound (> 0) and an upper bound for the choice of θ
- Discretize t, s.t. we have a finite set of $v_t(\theta)$ curves
    - A finite set, since only the $v_t(\theta)$ curves of the breakpoints are relevant

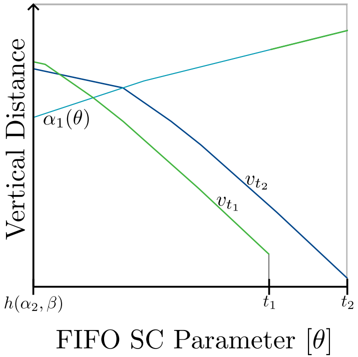
Reduce And Discretize The Search Space

- Problem to solve: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0}\{\max_{t \geq 0}\{v_t(\theta)\}\}$
- Get a lower bound (> 0) and an upper bound for the choice of θ
- Discretize t, s.t. we have a finite set of $v_t(\theta)$ curves
  - A finite set, since only the $v_t(\theta)$ curves of the breakpoints are relevant



Reduce And Discretize The Search Space

- Problem to solve: $\theta_{\text{opt}} = \arg\min_{\theta \geq 0}\{\max_{t \geq 0}\{v_t(\theta)\}\}$
- Get a lower bound (> 0) and an upper bound for the choice of θ
- Discretize t, s.t. we have a finite set of $v_t(\theta)$ curves
  - A finite set, since only the $v_t(\theta)$ curves of the breakpoints are relevant

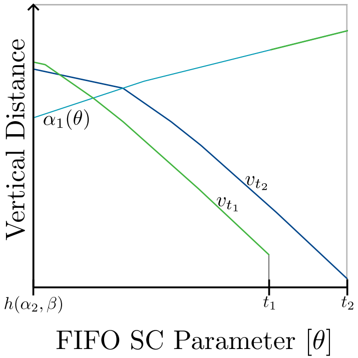


Backlog-Optimal θ

- Problem to solve: $\theta_{\text{opt}} = \arg\min\{\max\{v_t(\theta)\}\}$
- Solution is given by the last of all intersections of $v_t(\theta)$ curves and $\alpha_1(\theta)$

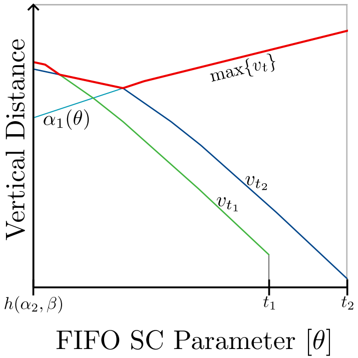
Backlog-Optimal θ

- Problem to solve: $\theta_{\text{opt}} = \arg\min\{\max\{v_t(\theta)\}\}$
- Solution is given by the last of all intersections of $v_t(\theta)$ curves and $\alpha_1(\theta)$



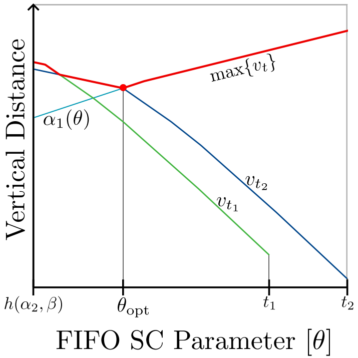
Backlog-Optimal θ

- Problem to solve: $\theta_{\text{opt}} = \arg\min\{\max\{v_t(\theta)\}\}$
- Solution is given by the last of all intersections of $v_t(\theta)$ curves and $\alpha_1(\theta)$



Backlog-Optimal θ

- Problem to solve: $\theta_{\text{opt}} = \arg\min\{\max\{v_t(\theta)\}\}$
- Solution is given by the last of all intersections of $v_t(\theta)$ curves and $\alpha_1(\theta)$



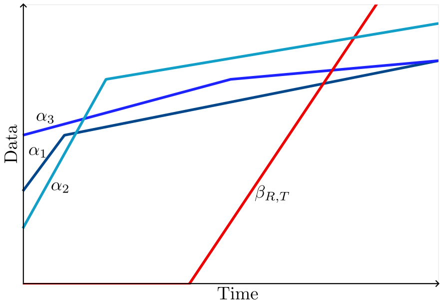
Numerical Evaluation

Numerical Evaluation: Setting

- Setting:
  - Rate-latency service curve $\beta_{R,T}$
  - Multiple cross flows and foi constraint by T-Spec arrival curves $\alpha_i(t) = \min\{b_i^1 + r_i^1 t, b_i^2 + r_i^2 t\}$

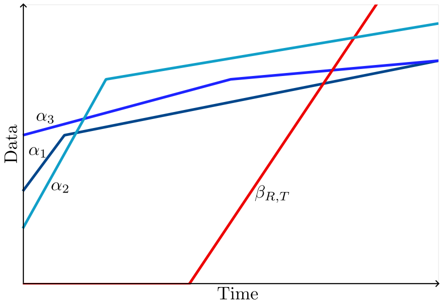
Numerical Evaluation: Setting

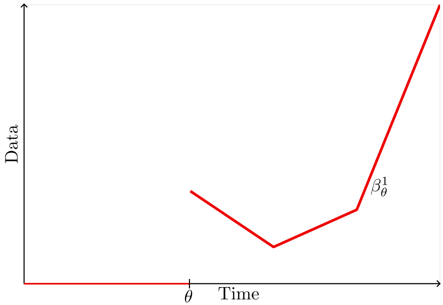
- Setting:
  - Rate-latency service curve $\beta_{R,T}$
  - Multiple cross flows and foi constraint by T-Spec arrival curves $\alpha_i(t) = \min\{b_i^1 + r_i^1 t, b_i^2 + r_i^2 t\}$



Numerical Evaluation: Setting

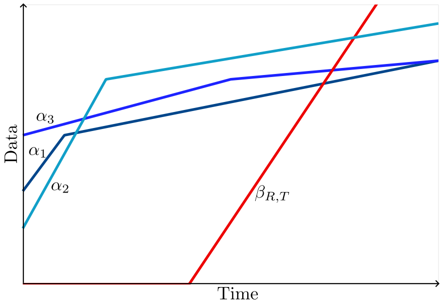
- Setting:
  - Rate-latency service curve $\beta_{R,T}$
  - Multiple cross flows and foi constraint by T-Spec arrival curves $\alpha_i(t) = \min\{b_i^1 + r_i^1 t, b_i^2 + r_i^2 t\}$

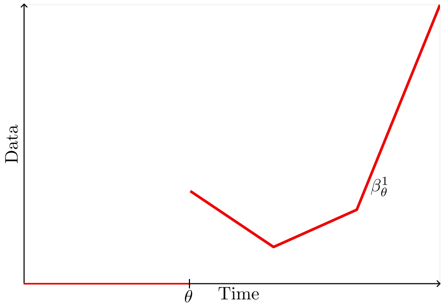




Numerical Evaluation: Setting

- Setting:
  - Rate-latency service curve $\beta_{R,T}$
  - Multiple cross flows and foi constraint by T-Spec arrival curves $\alpha_i(t) = \min\{b_i^1 + r_i^1 t, b_i^2 + r_i^2 t\}$





- The parameters used, b_i^1, r_i^1, b_i^2 and r_i^2, are randomly chosen out of a predefined interval

Numerical Evaluation: Comparison

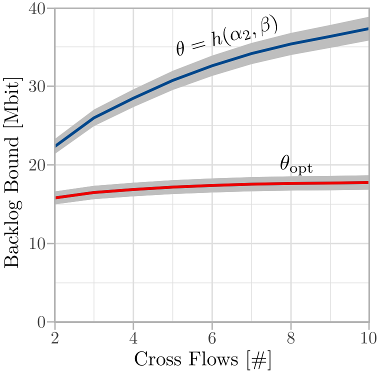
- We compare the backlog bound using our θ_{opt} with our lower bound $\theta = h(\alpha_2, \beta)$

Numerical Evaluation: Comparison

- We compare the backlog bound using our θ_{opt} with our lower bound $\theta = h(\alpha_2, \beta)$
- Choosing $\theta = h(\alpha_2, \beta)$ could be a reasonable choice, since it results in a continuous function

Numerical Evaluation: Comparison

- We compare the backlog bound using our θ_{opt} with our lower bound $\theta = h(\alpha_2, \beta)$
- Choosing $\theta = h(\alpha_2, \beta)$ could be a reasonable choice, since it results in a continuous function



Conclusion

Closing Remarks

- We are able to calculate backlog-optimal values of θ for PWL concave/convex curves

Closing Remarks

- We are able to calculate backlog-optimal values of θ for PWL concave/convex curves
- Using θ_{opt} leads to reduced backlog bounds (compared to the choice of existing tools)

Closing Remarks

- We are able to calculate backlog-optimal values of θ for PWL concave/convex curves
- Using θ_{opt} leads to reduced backlog bounds (compared to the choice of existing tools)

What's next?

Closing Remarks

- We are able to calculate backlog-optimal values of θ for PWL concave/convex curves
- Using θ_{opt} leads to reduced backlog bounds (compared to the choice of existing tools)

What's next?

- Explore (further) applications
- Consider multi-node case
- ...

References

Bouillard, A., Boyer, M., and Le Corronc, E. (2018).
Deterministic Network Calculus: From Theory to Practical Implementation.
John Wiley & Sons.

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

Delay-Optimal θ (Theoretical Result)

Theorem

Let $\mathcal{S}$ be a system that multiplexes two flows f_1 and f_2 according to FIFO, where the arrivals of f_1 and f_2, A_1 and A_2, are constrained by α_1 and α_2 respectively. Further, assume that $\mathcal{S}$ offers a service curve β to the aggregate of the flows. If $\alpha_1 \in \mathcal{F}$ and $\alpha_2 \in \mathcal{F}$ are concave and $\beta \in \mathcal{F}$ is convex, then the optimal θ, that minimizes the delay bound, is given by

$$\theta_{opt} = \arg\min_{\theta \geq 0}\{h(\alpha_1, \beta_\theta^1)\} = h(\alpha_{\mathrm{agg}}, \beta).$$

- Under FIFO scheduling the delay bound of the aggregate equals the per-flow delay bound
- It can be directly shown that $h(\alpha_{\mathrm{agg}}, \beta) = h\left(\alpha_1, \beta^1_{h(\alpha_{\mathrm{agg}}, \beta)}\right)$

$v_t(\theta)$ Curves

Definition

$$v_t(\theta) := \begin{cases} \alpha_1(t) - \beta_\theta^1(t), & \text{if } h(\alpha_2,\beta) \le \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \ge t > h(\alpha_2,\beta), \end{cases}$$
$$= \begin{cases} \alpha_1(t) - \beta(t) + \alpha_2(t-\theta), & \text{if } h(\alpha_2,\beta) \le \theta < t, \\ \alpha_1(\theta), & \text{if } \theta \ge t > h(\alpha_2,\beta). \end{cases}$$

By definition we have $\beta_\theta^1 = [\beta(t) - \alpha_2(t-\theta)]^+ \wedge \delta_\theta(t)$. Since for $\theta < t$ it holds that $\delta_\theta(t) = 0$, and because $h(\alpha_2,\beta) \le \theta$ ensures that $\beta(t) \ge \alpha_2(t-\theta)$ for all t, both the positive part and $\delta_\theta(t)$ can be omitted.

Application

We are now able to properly choose flow queue sizes for a setting with flows, having their own flow queue, getting FIFO scheduled