Exploiting Minimal Arrival Curves To Deal With Negative Service Curves

How MinAC Saves The Day

Anja Hamscher, Vlad-Cristian Constantin, Jens B. Schmitt

April 4th 2024

Many of the results will be published in IEEE RTAS 2024.

Outline

Motivation

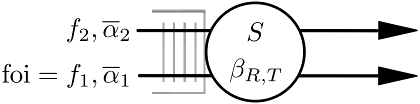
1 Motivation

2 Extension of NC Performance Bounds for (Partially) Negative Service Curves

3 Applications

4 Conclusion

What We're Up To



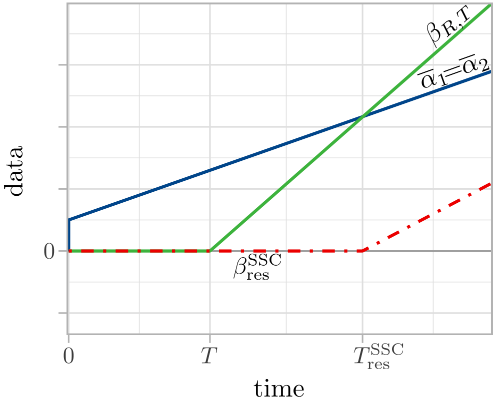
What We're Up To

- Assume $\mathcal{S}$ offers a strict service curve $\beta_{R,T}$

What We're Up To

- Assume $\mathcal{S}$ offers a strict service curve $\beta_{R,T}$
- Residual service curve $\beta_{\text{res}}^{\text{SSC}}$ [Le Boudec and Thiran, 2001] is calculated as

$$\beta_{\text{res}}^{\text{SSC}} \coloneqq [\beta_{R,T} - \overline{\alpha}_2]^+$$



What We're Up To

- Assume $\mathcal{S}$ offers a strict service curve $\beta_{R,T}$
- Residual service curve $\beta_{\mathrm{res}}^{\mathrm{SSC}}$ [Le Boudec and Thiran, 2001] is calculated as

$$\beta_{\mathrm{res}}^{\mathrm{SSC}} := [\beta_{R,T} - \overline{\alpha}_2]^+$$

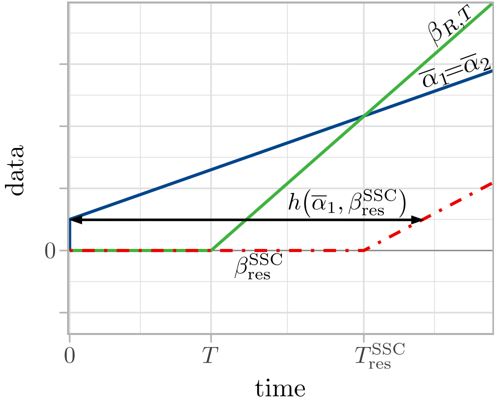
$\Rightarrow$ Delay bound for f_1 is calculated as $h(\overline{\alpha}_1, \beta_{\mathrm{res}}^{\mathrm{SSC}})$

What We're Up To

- Assume $\mathcal{S}$ offers a strict service curve $\beta_{R,T}$
- Residual service curve $\beta_{\text{res}}^{\text{SSC}}$ [Le Boudec and Thiran, 2001] is calculated as

$$\beta_{\text{res}}^{\text{SSC}} := [\beta_{R,T} - \overline{\alpha}_2]^+$$

$\Rightarrow$ Delay bound for f_1 is calculated as $h(\overline{\alpha}_1, \beta_{\text{res}}^{\text{SSC}})$



The Presumably "Hopeless" Case

- Assume $\mathcal{S}$ offers a min-plus service curve $\beta_{R,T}$

The Presumably "Hopeless" Case

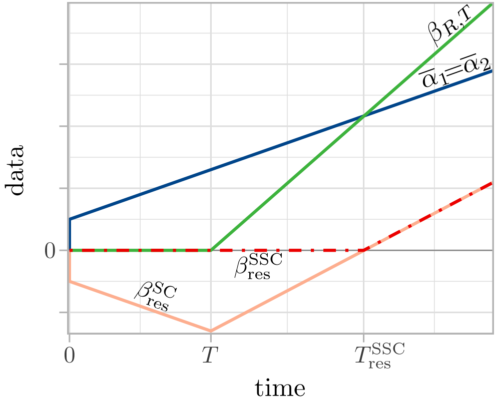
- Assume $\mathcal{S}$ offers a min-plus service curve $\beta_{R,T}$
- Residual service curve candidate $\beta_{\text{res}}^{\text{SC}}$ [Bouillard et al., 2018] is calculated as

$$\beta_{\text{res}}^{\text{SC}} := \beta_{R,T} - \overline{\alpha}_2$$

The Presumably "Hopeless" Case

- Assume $\mathcal{S}$ offers a min-plus service curve $\beta_{R,T}$
- Residual service curve candidate $\beta_{\mathrm{res}}^{\mathrm{SC}}$ [Bouillard et al., 2018] is calculated as

$$\beta_{\mathrm{res}}^{\mathrm{SC}} := \beta_{R,T} - \overline{\alpha}_2$$

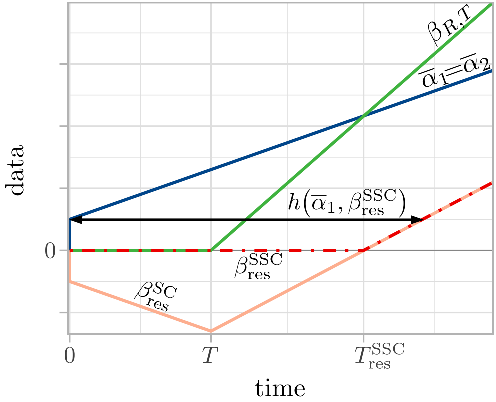


- $D_1 \geq A_1 \otimes \beta_{\mathrm{res}}^{\mathrm{SC}}$

The Presumably "Hopeless" Case

- Assume $\mathcal{S}$ offers a min-plus service curve $\beta_{R,T}$
- Residual service curve candidate $\beta_{\mathrm{res}}^{\mathrm{SC}}$ [Bouillard et al., 2018] is calculated as

$$\beta_{\mathrm{res}}^{\mathrm{SC}} := \beta_{R,T} - \overline{\alpha}_2$$



- $D_1 \geq A_1 \otimes \beta_{\mathrm{res}}^{\mathrm{SC}}$, **HOWEVER,** $\beta_{\mathrm{res}}^{\mathrm{SC}} \notin \mathcal{F}_0^{\uparrow}$!

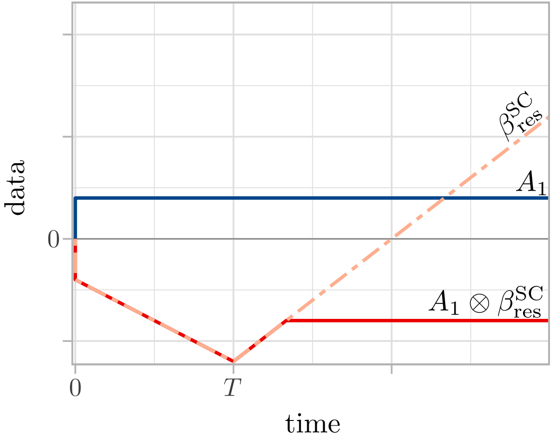
The Presumably "Hopeless" Case

- Assume $\mathcal{S}$ offers a min-plus service curve $\beta_{R,T}$
- Residual service curve candidate $\beta_{\mathrm{res}}^{\mathrm{SC}}$ [Bouillard et al., 2018] is calculated as

$$\beta_{\mathrm{res}}^{\mathrm{SC}} := \beta_{R,T} - \overline{\alpha}_2$$

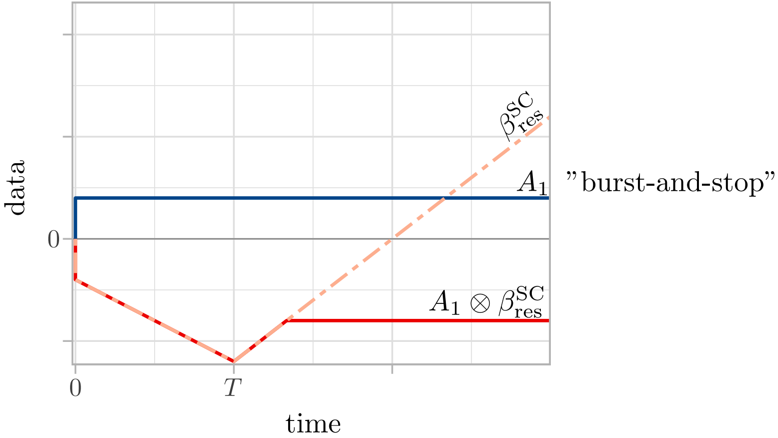
- $D_1 \geq A_1 \otimes \beta_{\mathrm{res}}^{\mathrm{SC}}$, **HOWEVER,** $\beta_{\mathrm{res}}^{\mathrm{SC}} \notin \mathcal{F}_0^{\uparrow}$!
- Performance bounds are only defined for $\beta \in \mathcal{F}_0^{\uparrow}$
  $\Rightarrow$ We have no valid delay bound

What's The Issue? [Bouillard et al., 2018]



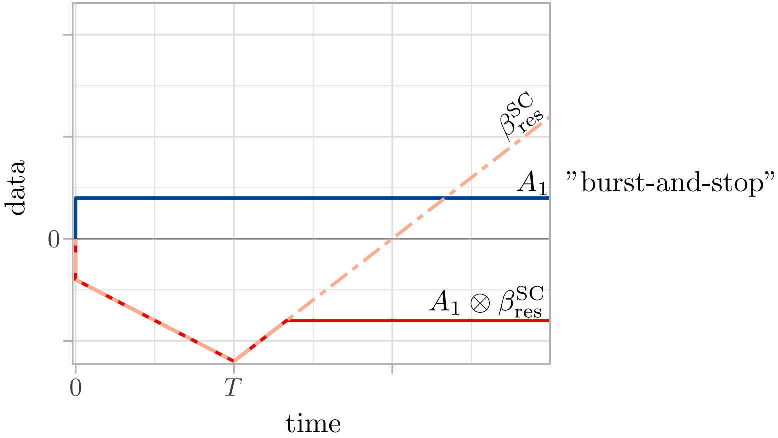
"burst-and-stop"

What's The Issue? [Bouillard et al., 2018]



- $A_1 \otimes \beta^{\mathrm{SC}}_{\mathrm{res}}$ never becomes positive

What's The Issue? [Bouillard et al., 2018]



- $A_1 \otimes \beta_{\mathrm{res}}^{\mathrm{SC}}$ never becomes positive

$\Rightarrow$ $\mathcal{S}$ can delay f_1 infinitely without violating min-plus SC property

Enter The Minimal Arrival Curve

- Issue: Not enough traffic in the system, allowing a "burst-and-stop" pattern

Enter The Minimal Arrival Curve

- Issue: Not enough traffic in the system, allowing a "burst-and-stop" pattern
- Idea: guarantee a lower bound on incoming traffic

Enter The Minimal Arrival Curve

- Issue: Not enough traffic in the system, allowing a "burst-and-stop" pattern
- Idea: guarantee a lower bound on incoming traffic

Definition

Let $\overline{\alpha}, \underline{\alpha} \in \mathcal{F}_0^{\uparrow}$. We say that $\overline{\alpha}$ is a maximal arrival curve for arrival process A, and $\underline{\alpha}$ is a minimal arrival curve for A, if it holds for all $0 \leq s \leq t$ that

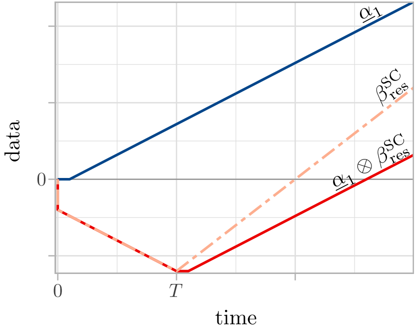
$$\underline{\alpha}(t-s) \leq A(t) - A(s) \leq \overline{\alpha}(t-s).$$

Enter The Minimal Arrival Curve

Definition

Let $\overline{\alpha}, \underline{\alpha} \in \mathcal{F}_0^{\uparrow}$. We say that $\overline{\alpha}$ is a maximal arrival curve for arrival process A, and $\underline{\alpha}$ is a minimal arrival curve for A, if it holds for all $0 \leq s \leq t$ that

$$\underline{\alpha}(t-s) \leq A(t) - A(s) \leq \overline{\alpha}(t-s).$$

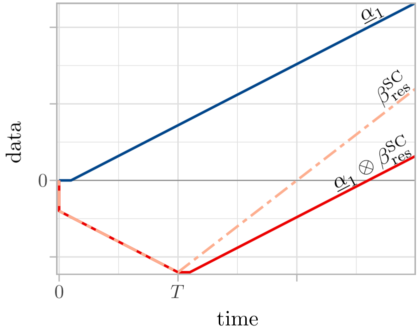


Enter The Minimal Arrival Curve

Definition

Let $\overline{\alpha}, \underline{\alpha} \in \mathcal{F}_0^{\uparrow}$. We say that $\overline{\alpha}$ is a maximal arrival curve for arrival process A, and $\underline{\alpha}$ is a minimal arrival curve for A, if it holds for all $0 \leq s \leq t$ that

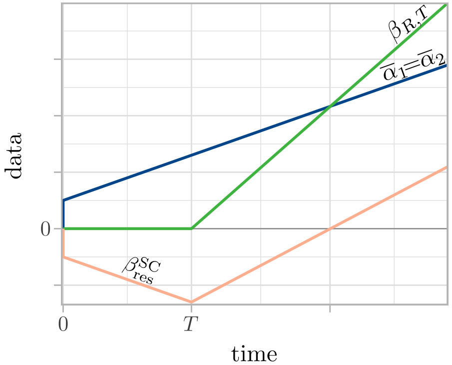
$$\underline{\alpha}(t-s) \leq A(t) - A(s) \leq \overline{\alpha}(t-s).$$



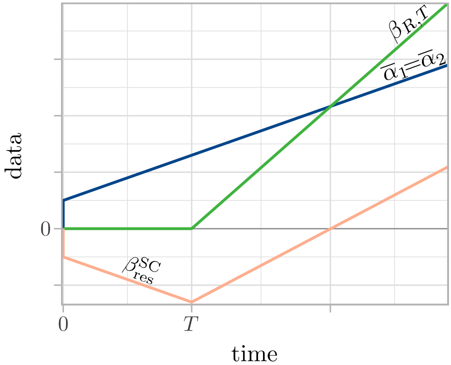
$\Rightarrow$ $\mathcal{S}$ has to eventually serve f_1

Extension of NC Performance Bounds for (Partially) Negative Service Curves

Monotony Preprocessing

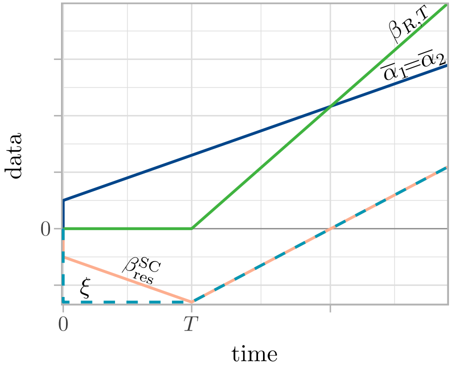


Monotony Preprocessing



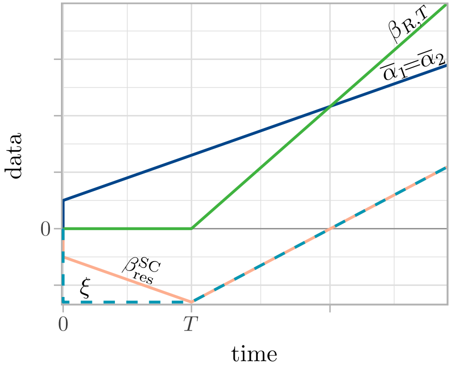
- We start with $\beta^{SC}_{res} =: \beta \in \mathcal{F}_0^{\cancel{\uparrow}}$

Monotony Preprocessing



- We start with $\beta^{SC}_{res} =: \beta \in \mathcal{F}_0^{\not\uparrow}$
- Then safely perform the lower non-decreasing closure $\xi(t) := \beta_{\downarrow}(t) = \beta \overline{\oslash} 0(t) = \inf_{s \geq 0}\{\beta(t+s)\}$

Monotony Preprocessing



- We start with $\beta^{SC}_{res} =: \beta \in \mathcal{F}_0^{\cancel{\uparrow}}$
- Then safely perform the lower non-decreasing closure
  $\xi(t) := \beta_{\downarrow}(t) = \beta \overline{\oslash} 0(t) = \inf_{s \geq 0}\{\beta(t+s)\}$

$\Rightarrow$ $\xi \in \mathcal{F}^{\uparrow}_{\leq 0}$

Generalized Delay Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$

Generalized Delay Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$

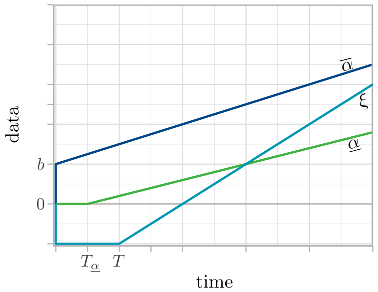
- For proof details, please refer to our technical report [Hamscher et al., 2024, arXiv]

Generalized Delay Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$

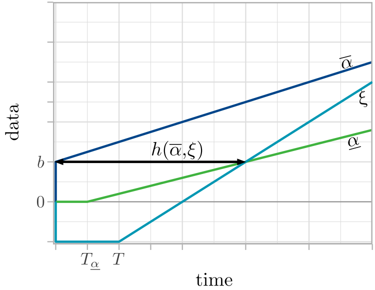


Generalized Delay Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$

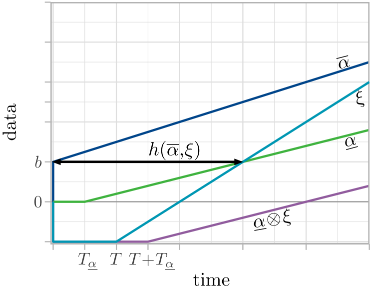


Generalized Delay Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$

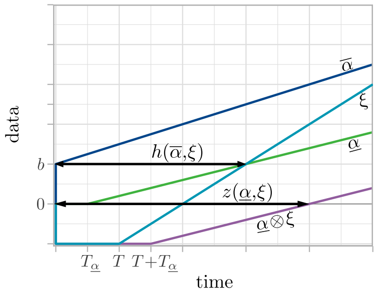


Generalized Delay Bound

Theorem

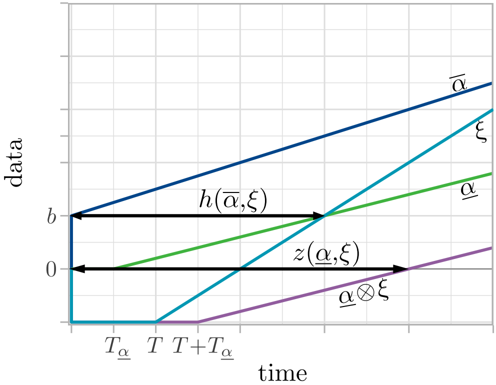
Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$ and minimal arrival curve $\underline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The virtual delay $d(t)$ satisfies for all $t \geq 0$

$$d(t) \leq z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi).$$



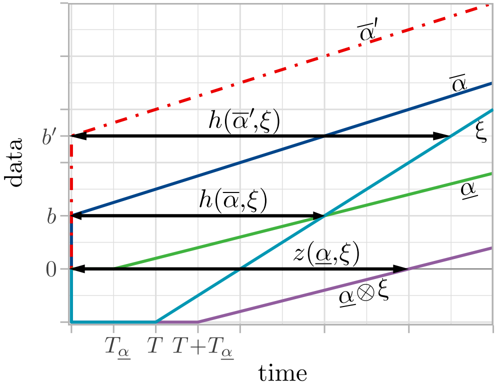
Different Cases of the Delay Bound

- If $\overline{\alpha}$ is not large enough, $z(\underline{\alpha}, \xi)$ is the delay bound



Different Cases of the Delay Bound

- If $\overline{\alpha}$ is not large enough, $z(\underline{\alpha}, \xi)$ is the delay bound
- If $\overline{\alpha}$ is large, then the delay bound is $h(\overline{\alpha}, \xi)$



Is the Generalized Delay Bound Tight?

Is the Generalized Delay Bound Tight?

- YES!

Is the Generalized Delay Bound Tight?

- YES!
- The worst-case delay $\mathit{WCD} = z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi)$

Is the Generalized Delay Bound Tight?

- YES!
- The worst-case delay $WCD = z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi)$
- A worst-case sample path is set as:
  - If $h(\overline{\alpha}, \xi) > z(\underline{\alpha}, \xi)$
    - $A^{\mathrm{WC}} := \overline{\alpha}$
    - $D^{\mathrm{WC}} := [\overline{\alpha} \otimes \xi]^{+}$

Is the Generalized Delay Bound Tight?

- YES!
- The worst-case delay $WCD = z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi)$
- A worst-case sample path is set as:
  - If $h(\overline{\alpha}, \xi) > z(\underline{\alpha}, \xi)$
    - $A^{\mathrm{WC}} := \overline{\alpha}$
    - $D^{\mathrm{WC}} := [\overline{\alpha} \otimes \xi]^+$
  - If $h(\overline{\alpha}, \xi) \leq z(\underline{\alpha}, \xi)$
    - $A^{\mathrm{WC}}(t) := \underline{\alpha}(t) + \overline{\alpha}(0_+) \cdot \mathbb{1}_{\{t>0\}}$
    - $D^{\mathrm{WC}}(t) := \left[\underline{\alpha} \otimes \xi(t) + \overline{\alpha}(0_+) \cdot \mathbb{1}_{\{t>0\}}\right]^+$

Is the Generalized Delay Bound Tight?

- YES!
- The worst-case delay $WCD = z(\underline{\alpha}, \xi) \vee h(\overline{\alpha}, \xi)$
- A worst-case sample path is set as:
  - If $h(\overline{\alpha}, \xi) > z(\underline{\alpha}, \xi)$
    - $A^{\mathrm{WC}} := \overline{\alpha}$
    - $D^{\mathrm{WC}} := [\overline{\alpha} \otimes \xi]^+$
  - If $h(\overline{\alpha}, \xi) \leq z(\underline{\alpha}, \xi)$
    - $A^{\mathrm{WC}}(t) := \underline{\alpha}(t) + \overline{\alpha}(0_+) \cdot \mathbb{1}_{\{t>0\}}$
    - $D^{\mathrm{WC}}(t) := \left[\underline{\alpha} \otimes \xi(t) + \overline{\alpha}(0_+) \cdot \mathbb{1}_{\{t>0\}}\right]^+$
- For proof details, please refer to our technical report [Hamscher et al., 2024, arXiv]

Backlog Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The backlog $q(t)$ satisfies for all t

$$q(t) \leq v(\overline{\alpha}, \xi)$$

Backlog Bound

Theorem

Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The backlog $q(t)$ satisfies for all t

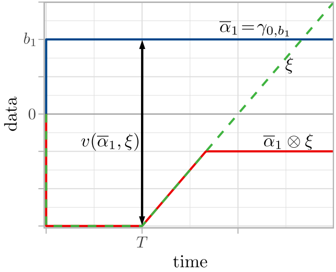
$$q(t) \leq v(\overline{\alpha}, \xi) \wedge \sup_{s \geq 0} \{\overline{\alpha}(s)\}.$$

Backlog Bound

Theorem

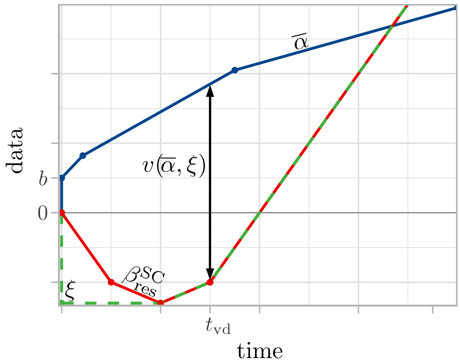
Let an arrival process A traverse a system $\mathcal{S}$. Further, let the arrivals be constrained by maximal arrival curve $\overline{\alpha} \in \mathcal{F}_0^{\uparrow}$, and let the system offer a service curve $\xi \in \mathcal{F}_{\leq 0}^{\uparrow}$. The backlog $q(t)$ satisfies for all t

$$q(t) \leq v(\overline{\alpha}, \xi) \wedge \sup_{s \geq 0} \{\overline{\alpha}(s)\}.$$



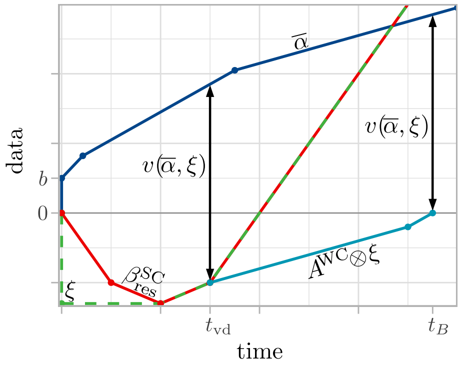
Tightness of the Backlog Bound: The Interesting Case

- The vertical deviation $v(\overline{\alpha}, \xi)$ is attained when ξ is negative



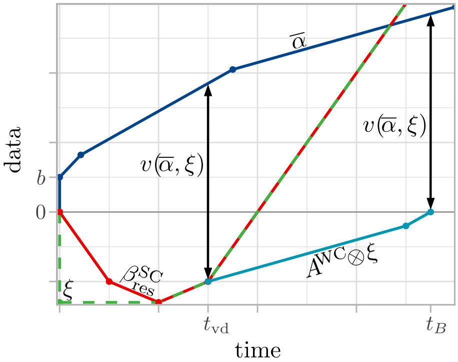
Tightness of the Backlog Bound: The Interesting Case

- The vertical deviation $v(\overline{\alpha}, \xi)$ is attained when ξ is negative



Tightness of the Backlog Bound: The Interesting Case

- The vertical deviation $v(\overline{\alpha}, \xi)$ is attained when ξ is negative



- For proof details, please refer to our technical report [Hamscher et al., 2024, arXiv]

Applications

Application Areas

- So far: have to choose between
  either using (1) residual SC or (2) concatenation

Application Areas

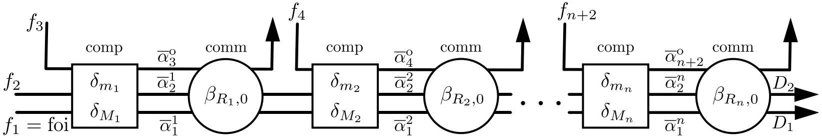
- So far: have to choose between
  either using (1) residual SC or (2) concatenation
- With new results, we can safely use both operations without requiring SSC

Application Areas

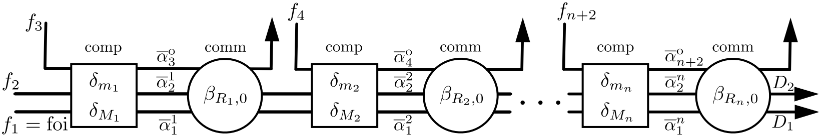
- So far: have to choose between
  either using (1) residual SC or (2) concatenation
- With new results, we can safely use both operations without requiring SSC
- Some network elements cannot provide a SSC

Application Areas

- So far: have to choose between either using (1) residual SC or (2) concatenation
- With new results, we can safely use both operations without requiring SSC
- Some network elements cannot provide a SSC

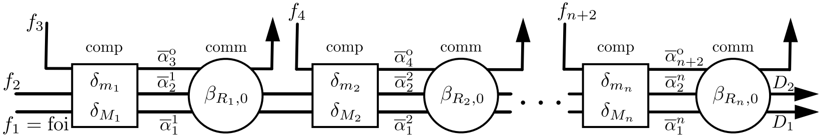


The MinAC Approach



- Idea: calculate end-to-end min-plus service curve and residual service curve, then calculate delay bound for f_1

The MinAC Approach



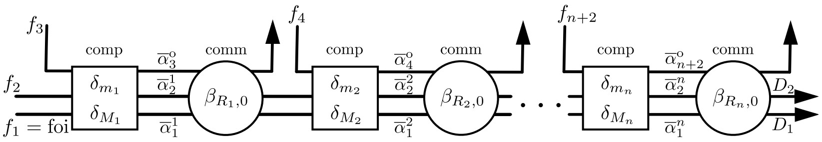
- Idea: calculate end-to-end min-plus service curve and residual service curve, then calculate delay bound for f_1
- Determine residual service curve:

$$\beta_{\mathrm{res}}^{\mathrm{mac}} := \left(\left(\bigotimes_{i=1}^{n} (\beta_{R_i,T_i} - \overline{\alpha}_{i+2}) \right) - \overline{\alpha}_2 \right)_{\downarrow}$$

$$= \xi_{b_2 + b_{i+2} + (r_2 + r_{i+2}) \sum_{i=1}^{n} T_i, \bigwedge_{i=1}^{n} (R_i - r_{i+2}) - r_2, \sum_{i=1}^{n} T_i}$$

with $\xi_{b_N,R,T}(t) := \beta_{R,T}(t) - b_N$

The MinAC Approach



- Idea: calculate end-to-end min-plus service curve and residual service curve, then calculate delay bound for f_1
- Determine residual service curve:

$$\beta_{\text{res}}^{\text{mac}} := \left(\left(\bigotimes_{i=1}^{n} (\beta_{R_i,T_i} - \overline{\alpha}_{i+2}) \right) - \overline{\alpha}_2 \right)_{\downarrow}$$

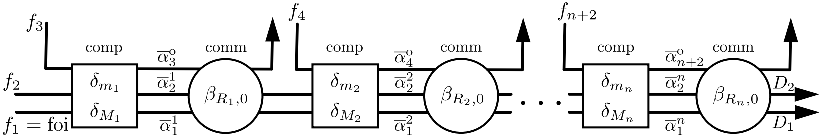
$$= \xi_{b_2 + b_{i+2} + (r_2 + r_{i+2}) \sum_{i=1}^{n} T_i, \bigwedge_{i=1}^{n} (R_i - r_{i+2}) - r_2, \sum_{i=1}^{n} T_i}$$

with $\xi_{b_N,R,T}(t) := \beta_{R,T}(t) - b_N$

- Calculate end-to-end delay bound:

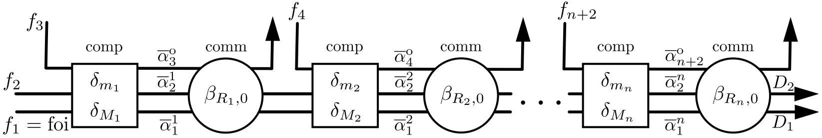
$$d_{\text{e2e}}^{\text{mac}} = h(\overline{\alpha}_1, \beta_{\text{res}}^{\text{mac}}) \vee z(\underline{\alpha}_1, \beta_{\text{res}}^{\text{mac}})$$

The Conventional Approach



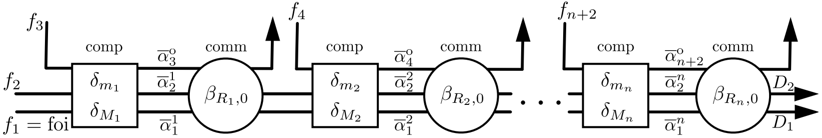
- Idea: determine input to comm by using output bound and calculate residual service curve for each component, then sum up the delay bound at each component to get end-to-end delay for f_1

The Conventional Approach



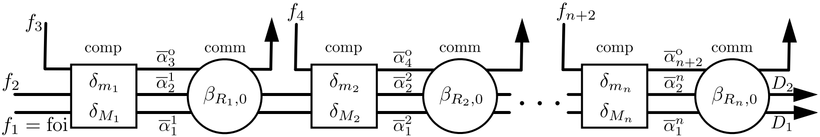
- Idea: determine input to comm by using output bound and calculate residual service curve for each component, then sum up the delay bound at each component to get end-to-end delay for f_1
- Determine input flows

The Conventional Approach



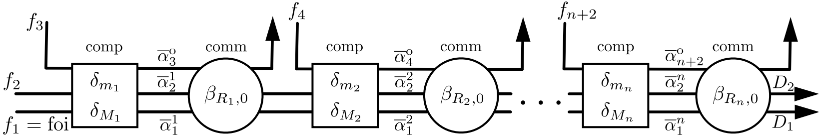
- Idea: determine input to comm by using output bound and calculate residual service curve for each component, then sum up the delay bound at each component to get end-to-end delay for f_1
- Determine input flows
- Determine residual service curve at comm i

The Conventional Approach



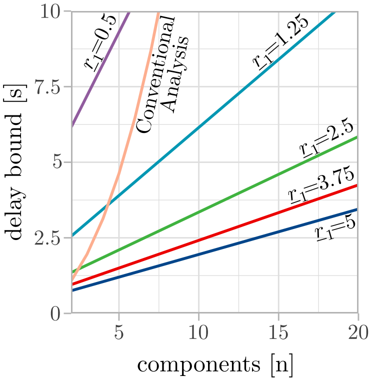
- Idea: determine input to comm by using output bound and calculate residual service curve for each component, then sum up the delay bound at each component to get end-to-end delay for f_1
- Determine input flows
- Determine residual service curve at comm i
- Calculate nodal delay bound d_i for f_1

The Conventional Approach



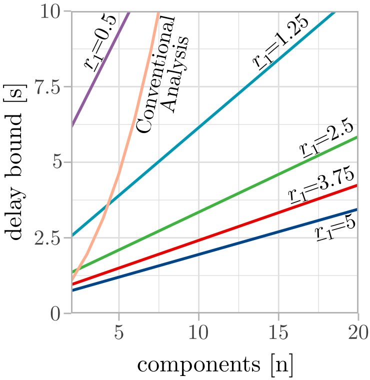
- Idea: determine input to comm by using output bound and calculate residual service curve for each component, then sum up the delay bound at each component to get end-to-end delay for f_1
- Determine input flows
- Determine residual service curve at comm i
- Calculate nodal delay bound d_i for f_1
- Calculate end-to-end delay bound

How Do They Compare?



- $b_1 = b_2 = b_3 = 1\,\mathrm{Mbit}$,
- $r_1 = r_2 = r_3 = 5\,\frac{\mathrm{Mbit}}{\mathrm{s}}$,
- $R_i = 20\,\frac{\mathrm{Mbit}}{\mathrm{s}} =: R,\ T_i = 50\,\mathrm{ms}$, $i \in \{1, \ldots, n\}$,
- $T_{\underline{\alpha_1}} = \frac{b_1}{R}$,
- $\underline{r}_1 \in \{0.5, 1.25, 2.5, 3.75, 5\}\,\frac{\mathrm{Mbit}}{\mathrm{s}}$

How Do They Compare?



- $b_1 = b_2 = b_3 = 1\,\mathrm{Mbit}$,
- $r_1 = r_2 = r_3 = 5\,\frac{\mathrm{Mbit}}{\mathrm{s}}$,
- $R_i = 20\,\frac{\mathrm{Mbit}}{\mathrm{s}} =: R,\ T_i = 50\,\mathrm{ms}$, $i \in \{1, \dots, n\}$,
- $T_{\underline{\alpha_1}} = \frac{b_1}{R}$,
- $\underline{r}_1 \in \{0.5, 1.25, 2.5, 3.75, 5\}\,\frac{\mathrm{Mbit}}{\mathrm{s}}$

- Different scaling of bounds
  - Min AC approach stays linear in n, while conventional analysis has super-linear scaling

Conclusion

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC
- Opens the scope for new applications that were previously difficult or unable to be analyzed

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC
- Opens the scope for new applications that were previously difficult or unable to be analyzed
  - Presented C/C networks

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC
- Opens the scope for new applications that were previously difficult or unable to be analyzed
  - Presented C/C networks
  - Not presented: finite buffer calculation

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC
- Opens the scope for new applications that were previously difficult or unable to be analyzed
  - Presented C/C networks
  - Not presented: finite buffer calculation
- What's next?

Closing Remarks

- Extended NC to be able to handle $\beta \in \mathcal{F}_{\leq 0}$
  - Solved issue of requiring SSCs to obtain residual SCs
  - Enables simultaneous use of concatenation and calculation of residual SCs
  - Delay bound calculation is enabled by minimal AC, backlog bound is generalized without a minimal AC
- Opens the scope for new applications that were previously difficult or unable to be analyzed
  - Presented C/C networks
  - Not presented: finite buffer calculation
- What's next?
  - Investigate accuracy of LNDC preprocessing
  - Explore further applications
  - Build lower traffic shaper
  - ...

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

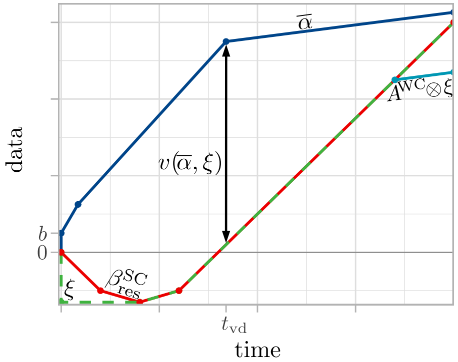
Hamscher, A., Constantin, V.-C., and Schmitt, J. B. (2024).
Extending Network Calculus To Deal With Partially Negative And Decreasing Service Curves.
arXiv preprint arXiv:2403.18042.

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

Any questions?

Tightness of the Backlog Bound: The Standard Case

- The negativity of ξ essentially plays no role



Tightness of the Backlog Bound: The Interesting Case

- Let $t_B := \overline{\alpha}^{-1}(v(\overline{\alpha}, \xi))$
- Set a worst-case sample path as:

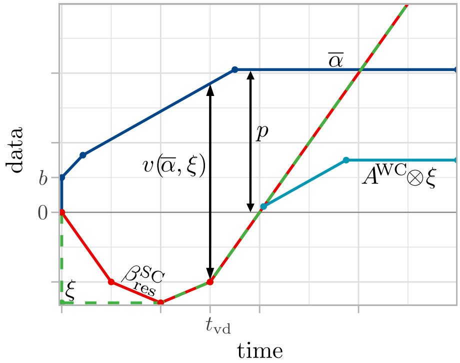
$$A^{\mathrm{WC}}(t) := \begin{cases} \overline{\alpha}(t_B) - \overline{\alpha}(t_B - t), & \text{if } t \leq t_B, \\ \overline{\alpha}(t_B), & \text{otherwise,} \end{cases}$$

and $D^{\mathrm{WC}} := \left[A^{\mathrm{WC}} \otimes \xi\right]^{+}$.

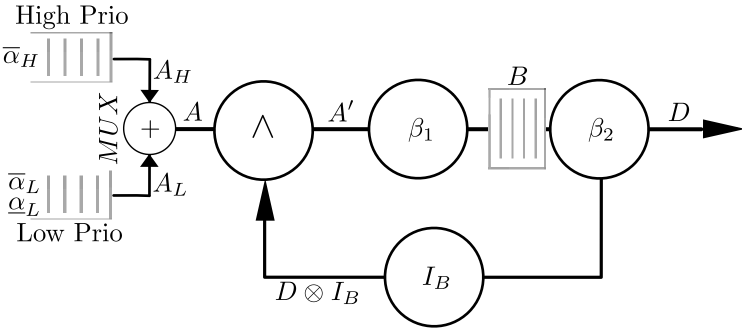
- Then $q(t_B) = v(\overline{\alpha}, \xi) \wedge \sup_{s \geq 0} \{\overline{\alpha}(s)\}$

Tightness of the Backlog Bound: The Plateau Case

- The backlog may never attain $v(\overline{\alpha}, \xi)$



Finite Buffers: Network Model

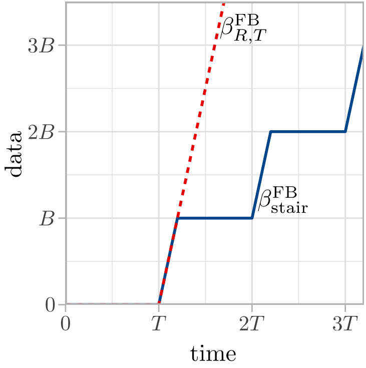


Finite Buffers: Buffer Dimensioning

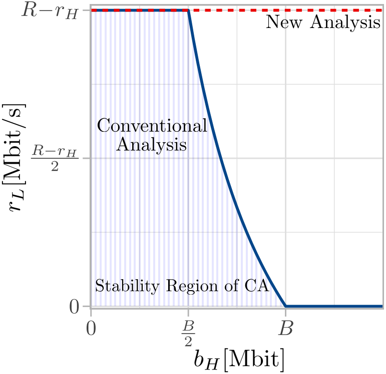
- High prio stays the same: $v(\overline{\alpha}_H, \beta^{\mathrm{FB}}) = b_H$
- Low prio changes with analysis method
  - MinAC: $v(\overline{\alpha}_L, \beta_{\mathrm{res}}^{\mathrm{mac}}) = b_H + b_L$

Finite Buffers: Stability Regions of Conventional Analysis

- Check whether $R^{\text{res}} T^{\text{res}} \leq B^{\text{res}}$
  - True? $\Rightarrow v(\overline{\alpha}, \beta^{\text{ca}}_{\text{res}}) = b_L + r_L \frac{b_H}{R - r_H}$
  - False? $\Rightarrow$ Follow shape of $\beta^{\text{FB}}_{\text{stair}}$ and watch for r_L



a) Shapes of $\beta^{\text{FB}}_{R,T}$ and $\beta^{\text{FB}}_{\text{stair}}$.



b) Stability regions of the analyses.

Finite Buffers: Delay Bounds

- High prio stays the same: $h(\overline{\alpha}_H, \beta^{\mathrm{FB}}) = \frac{b_H}{R}$
- Low prio changes with analysis method
  - MinAC uses previously shown delay bound
  - CA: Analogous to backlog; check for bandwidth delay product and r_L