distributed computer systems

Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

Using Minimal Arrival Curves to Derive Per-Flow Performance Bounds in (Tandem) Networks with Complex Feedback Structures [1]

Anja Hamscher, Lukas Wildberger, Jens B. Schmitt

June 5th, 2025

[1] to be published in ECRTS 2025

Outline

1 Previous NC Work on Feedback

2 Extending Analysis to Interdependent Feedback Constraints

3 Deriving Per-Flow Performance Bounds

4 Conclusion

Previous NC Work on Feedback

1 Previous NC Work on Feedback

2 Extending Analysis to Interdependent Feedback Constraints

3 Deriving Per-Flow Performance Bounds

4 Conclusion

Motivation

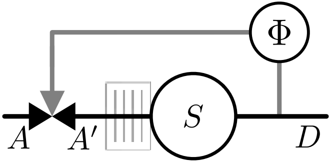


Figure: System with feedback

- System with some sort of signaling mechanism (finite buffer, ...)

Motivation

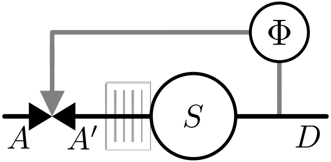


Figure: System with feedback

- System with some sort of signaling mechanism (finite buffer, ...)
- Signaling is modeled as backwards loop that feeds back into dataflow

Motivation

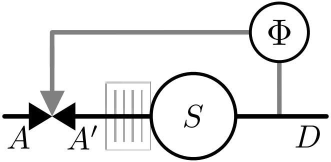


Figure: System with feedback

- System with some sort of signaling mechanism (finite buffer, ...)
- Signaling is modeled as backwards loop that feeds back into dataflow
- Feedback is modeled by some function Φ

Motivation

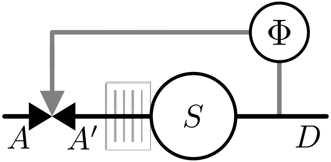


Figure: System with feedback

- System with some sort of signaling mechanism (finite buffer, ...)
- Signaling is modeled as backwards loop that feeds back into dataflow
- Feedback is modeled by some function Φ
- Arrivals to the system are throttled according to this function

Motivation

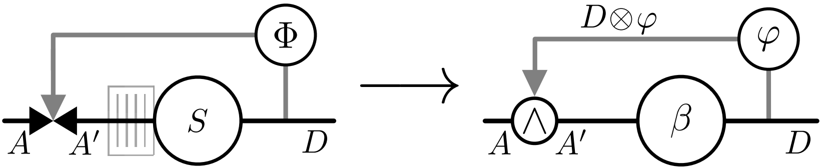


Figure: System with feedback

- System with some sort of signaling mechanism (finite buffer, ...)
- Signaling is modeled as backwards loop that feeds back into dataflow
- Feedback is modeled by some function Φ
- Arrivals to the system are throttled according to this function
- Call φ with $D \circ \Phi \geq D \otimes \varphi$ a feedback curve

What Do We Want To Analyze?

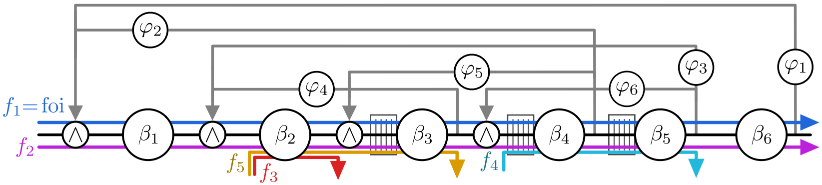


Figure: Example network with various feedback constraints and flows.

- Network with a multitude of feedback constraints that potentially have interdependencies
- Arbitrary number of interfering flows

What Do We Want To Analyze?

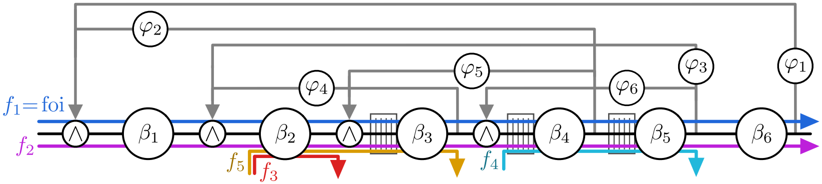


Figure: Example network with various feedback constraints and flows.

- Network with a multitude of feedback constraints that potentially have interdependencies
- Arbitrary number of interfering flows
- Goal: Obtain per-flow performance bounds for any scheduling policy
  - Not just FIFO...

Closed-Loop System

- System with feedback is a *closed-loop system*

Closed-Loop System

- System with feedback is a *closed-loop system*
- Issue: not min-plus linear [Boudec and Thiran, 2001], hence cannot be (directly) analyzed using NC

Closed-Loop System

- System with feedback is a *closed-loop system*
- Issue: not min-plus linear [Boudec and Thiran, 2001], hence cannot be (directly) analyzed using NC
- Solution: transformation into *open-loop system* [Chang, 2000]

Theorem

Let an arrival process A traverse a system $\mathcal{S}$ offering service curve $\beta \in \mathcal{F}$. The system is subject to a feedback curve φ. If

$$D \geq (A \wedge (D \otimes \varphi)) \otimes \beta,$$

then

$$D \geq A \otimes \beta \otimes (\beta \otimes \varphi)^{*}.$$

Transformation Into Open-Loop System

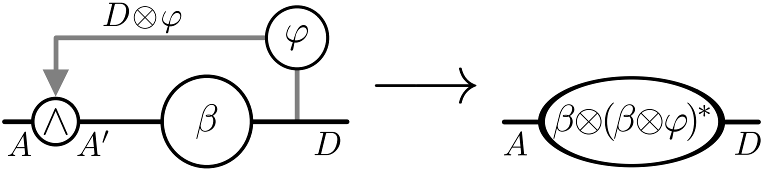


Figure: Open-loop transformation using Theorem.

- Using the Theorem, we obtain an open-loop system

Transformation Into Open-Loop System

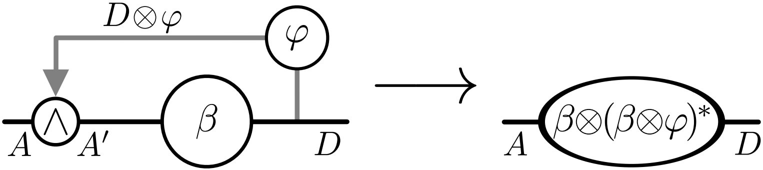


Figure: Open-loop transformation using Theorem.

- Using the Theorem, we obtain an open-loop system
- Feedback is still captured in the system, but encoded into the service curve directly

Transformation Into Open-Loop System

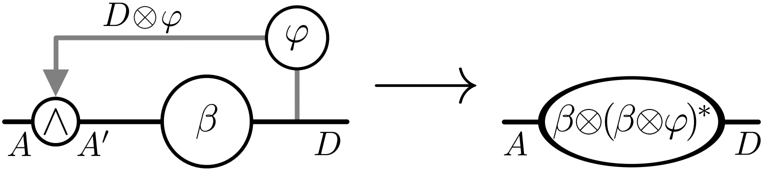


Figure: Open-loop transformation using Theorem.

- Using the Theorem, we obtain an open-loop system
- Feedback is still captured in the system, but encoded into the service curve directly
- Resulting system is min-plus linear → can be analyzed with normal NC methods

Generalization To Arbitrary Length

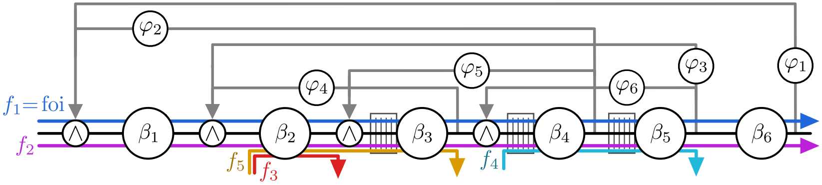


Figure: Example network with various feedback constraints and flows.

Generalization To Arbitrary Length

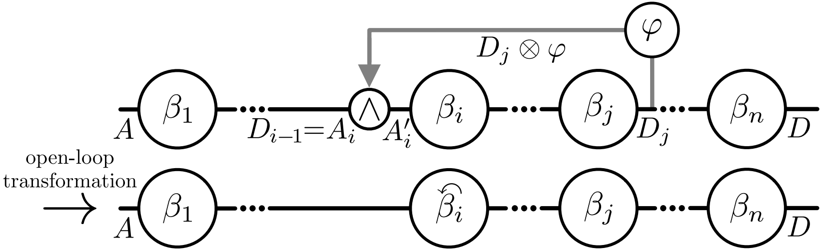


Figure: Open-loop transformation for a feedback constraint of arbitrary length.

- Generalize to feedback constraint $F = (i, j, \varphi)$

Generalization To Arbitrary Length

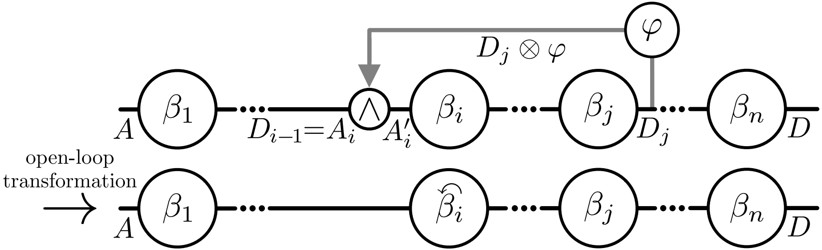


Figure: Open-loop transformation for a feedback constraint of arbitrary length.

- Generalize to feedback constraint $F = (i, j, \varphi)$
- Transformed service curve $\overset{\frown}{\beta}_i := \beta_i \otimes \left(\bigotimes_{k=i}^{j} \beta_k \otimes \varphi \right)^*$

Generalization To Arbitrary Length

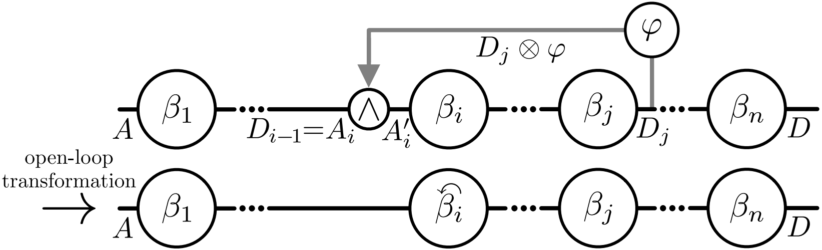


Figure: Open-loop transformation for a feedback constraint of arbitrary length.

- Generalize to feedback constraint $F = (i, j, \varphi)$
- Transformed service curve $\widehat{\beta}_i := \beta_i \otimes \left(\bigotimes_{k=i}^{j} \beta_k \otimes \varphi\right)^*$
- Call this Proposition for remainder of the talk

Extending Analysis to Interdependent Feedback Constraints

1 Previous NC Work on Feedback

2 Extending Analysis to Interdependent Feedback Constraints

3 Deriving Per-Flow Performance Bounds

4 Conclusion

Complex Feedback Structures

- What about having multiple constraints in a single system?

Complex Feedback Structures

- What about having multiple constraints in a single system?
- What if the constraints have an interdependency?

Complex Feedback Structures

- What about having multiple constraints in a single system?
- What if the constraints have an interdependency?
- Some existing results on overlapping interdependency [Chang, 2000, Bose et al., 2006, Bouillard et al., 2009]

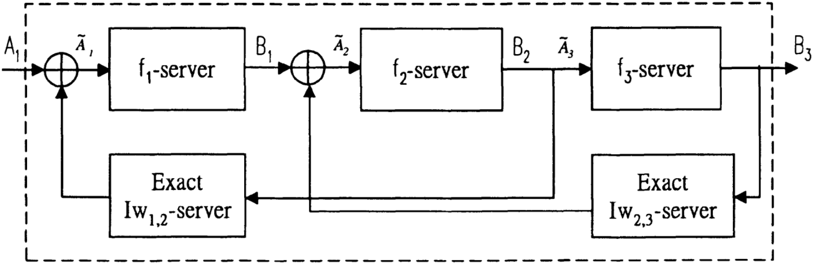


Figure: Overlapping interdependency [Chang, 2000].

Complex Feedback Structures

- What about having multiple constraints in a single system?
- What if the constraints have an interdependency?
- Some existing results on overlapping interdependency [Chang, 2000, Bose et al., 2006, Bouillard et al., 2009]
- But, not the only interdependence type!

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order

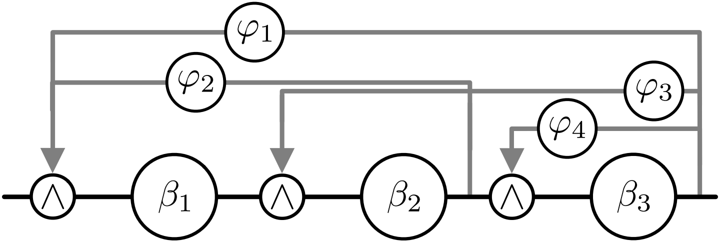


Figure: Network with interdependent feedback constraints.

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints
- Can then specify the three interdependence types between two constraints F_l and F_m with $l < m$

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints
- Can then specify the three interdependence types between two constraints F_l and F_m with $l < m$
  - If $j_l < i_m$, no interdependency exists

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints
- Can then specify the three interdependence types between two constraints F_l and F_m with $l < m$
  - If $j_l < i_m$, no interdependency exists

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints
- Can then specify the three interdependence types between two constraints F_l and F_m with $l < m$
  - If $j_l < i_m$, no interdependency exists
  - If $j_l \geq i_m$, interdependency exists

Interdependence Types

- Establish lexicographical order over all existing feedback constraints $F_l = (i_l, j_l, \varphi_l), l = 1, \ldots, k$, in the system as follows
  - First criterion: order by i_l in ascending order
  - Second criterion: order by j_l in descending order
- Result is ordered set of feedback constraints
- Can then specify the three interdependence types between two constraints F_l and F_m with $l < m$
  - If $j_l < i_m$, no interdependency exists
  - If $j_l \geq i_m$, interdependency exists

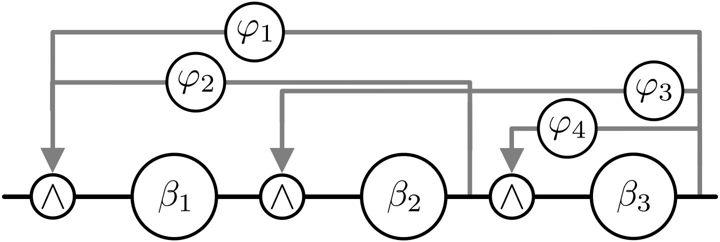


Figure: Network with interdependent feedback constraints.

Contained Interdependency

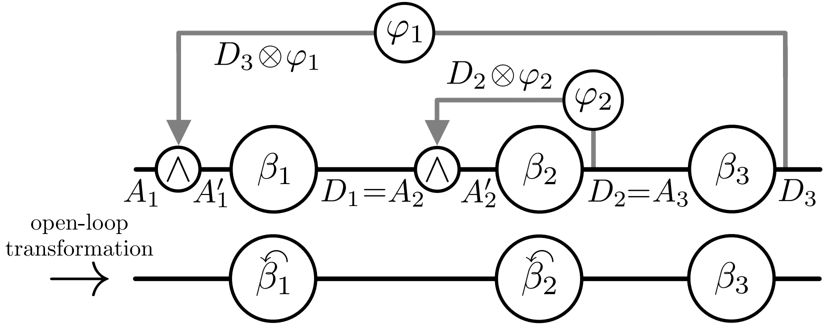


Figure: Feedback constraints with contained interdependency.

- Is the case for $i_l < i_m \Rightarrow F_l$ is fully contained within F_m

Contained Interdependency

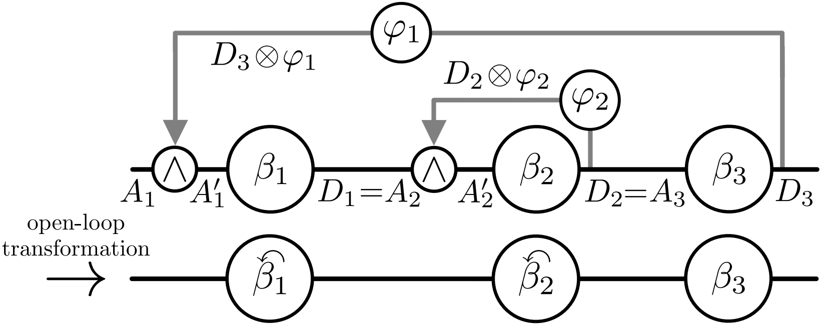


Figure: Feedback constraints with contained interdependency.

- Is the case for $i_l < i_m \Rightarrow F_l$ is fully contained within F_m
- Canonical example: $F_1 = (1, 3, \varphi_1), F_2 = (2, 2, \varphi_2)$

Contained Interdependency

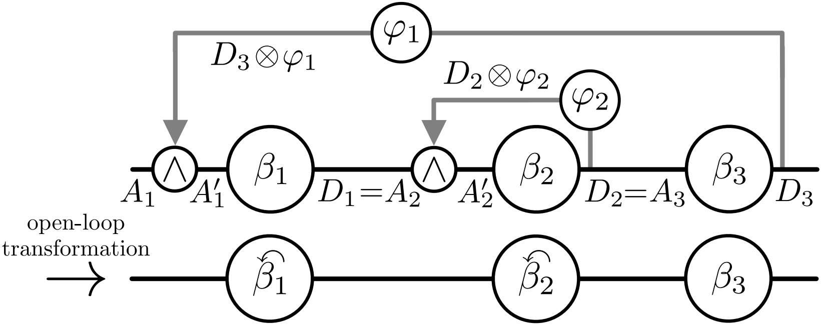


Figure: Feedback constraints with contained interdependency.

- Is the case for $i_l < i_m \Rightarrow F_l$ is fully contained within F_m
- Canonical example: $F_1 = (1, 3, \varphi_1), F_2 = (2, 2, \varphi_2)$
- Use Theorem to resolve $F_2 \rightarrow \overset{\curvearrowleft}{\beta_2} = \beta_2 \otimes (\beta_2 \otimes \varphi_2)^*$

Contained Interdependency

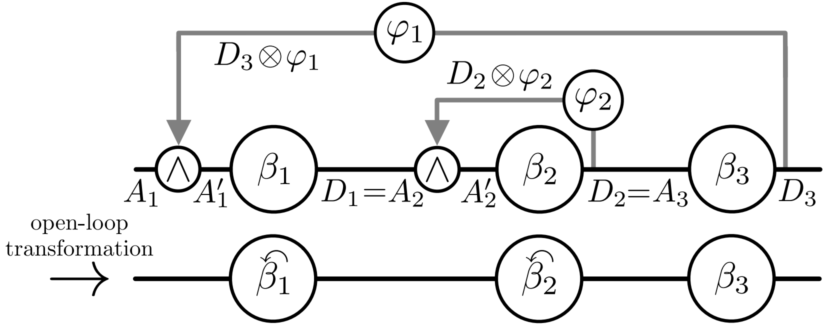


Figure: Feedback constraints with contained interdependency.

- Is the case for $i_l < i_m \Rightarrow F_l$ is fully contained within F_m
- Canonical example: $F_1 = (1, 3, \varphi_1), F_2 = (2, 2, \varphi_2)$
- Use Theorem to resolve $F_2 \rightarrow \overset{\frown}{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \varphi_2)^*$
- Use Proposition to resolve F_1
  $\rightarrow \overset{\frown}{\beta}_1 = \beta_1 \otimes (\beta_2 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \varphi_1)^*$

Overlapping Interdependency

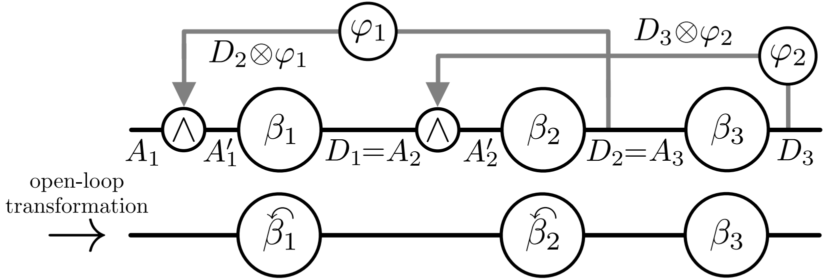


Figure: Feedback constraints with overlapping interdependency.

- "Known" case; occurs for $j_l < j_m$

Overlapping Interdependency

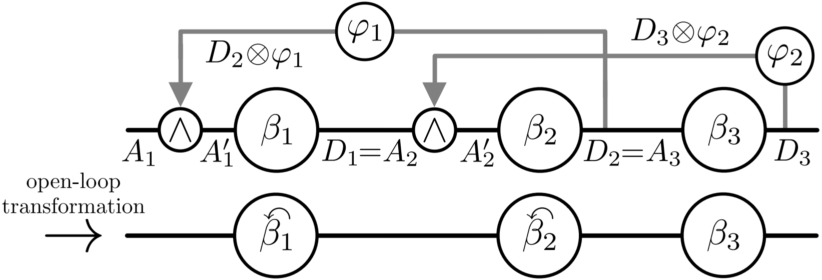


Figure: Feedback constraints with overlapping interdependency.

- "Known" case; occurs for $j_l < j_m$
- Canonical example: $F_1 = (1, 2, \varphi_1), F_2 = (2, 3, \varphi_2)$

Overlapping Interdependency

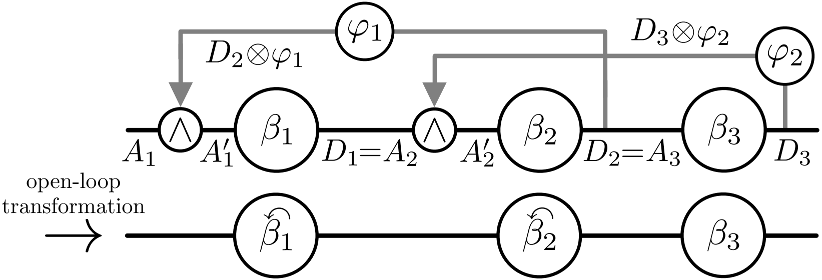


Figure: Feedback constraints with overlapping interdependency.

- "Known" case; occurs for $j_l < j_m$
- Canonical example: $F_1 = (1, 2, \varphi_1), F_2 = (2, 3, \varphi_2)$
- Use Prop. to resolve $F_2 \rightarrow \overset{\frown}{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*$

Overlapping Interdependency

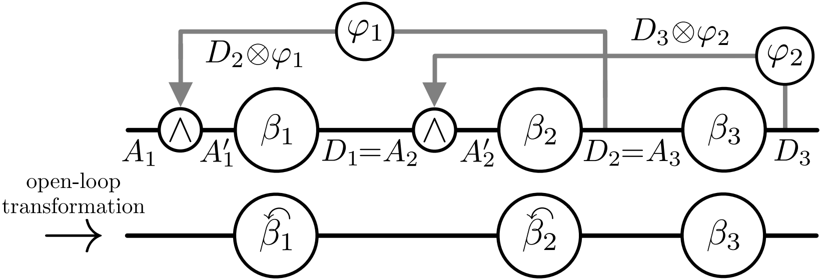
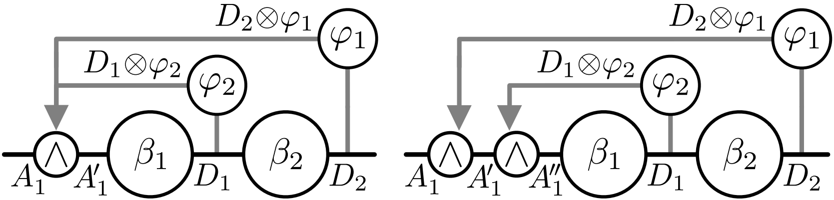


Figure: Feedback constraints with overlapping interdependency.

- "Known" case; occurs for $j_l < j_m$
- Canonical example: $F_1 = (1, 2, \varphi_1), F_2 = (2, 3, \varphi_2)$
- Use Prop. to resolve $F_2 \rightarrow \overset{\frown}{\beta}_2 = \beta_2 \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*$
- Use Prop. to resolve F_1
  $\rightarrow \overset{\frown}{\beta}_1 = \beta_1 \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^* \otimes (\beta_2 \otimes \beta_3 \otimes \varphi_2)^*$

Compounded Interdependence



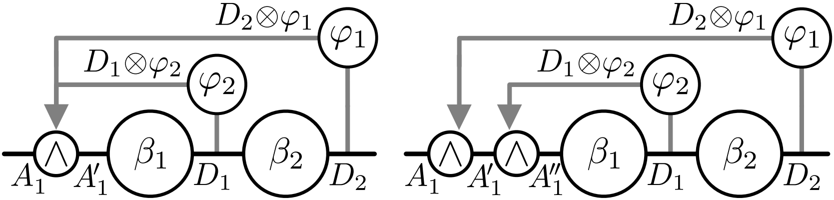
(a) Combined minimum

(b) Separated minima

Figure: Equivalent canonical compounded feedback structures.

- Occurs if $i_l = i_m$

Compounded Interdependence



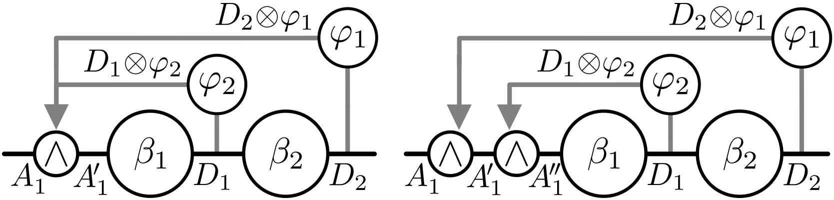
(a) Combined minimum

(b) Separated minima

Figure: Equivalent canonical compounded feedback structures.

- Occurs if $i_l = i_m$
- Canonical example: $F_1 = (1, 2, \varphi_1), F_2 = (1, 1, \varphi_2)$

Compounded Interdependence



(a) Combined minimum (b) Separated minima

Figure: Equivalent canonical compounded feedback structures.

- Occurs if $i_l = i_m$
- Canonical example: $F_1 = (1, 2, \varphi_1), F_2 = (1, 1, \varphi_2)$
- Can show that structure (a) is equivalent to structure (b)
  - (a) $A'_1 = A_1 \wedge (D_2 \otimes \varphi_1) \wedge (D_1 \otimes \varphi_2)$
  - (b) $A'_1 = A_1 \wedge (D_2 \otimes \varphi_1)$,
    $A''_1 = A'_1 \wedge (D_1 \otimes \varphi_2) = A_1 \wedge (D_2 \otimes \varphi_1) \wedge (D_1 \otimes \varphi_2)$

Compounded Interdependency

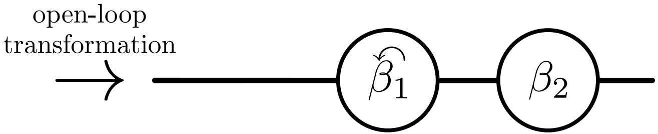


Figure: Resulting open-loop system.

- Recover contained interdependency, but with $i_l = i_m$

$\Rightarrow$ β_1 has to be transformed twice

Compounded Interdependency

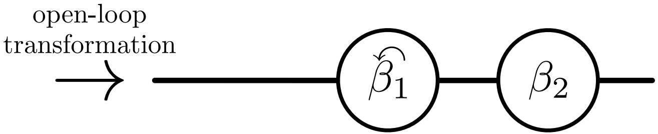


Figure: Resulting open-loop system.

- Recover contained interdependency, but with $i_l = i_m$

$\Rightarrow$ β_1 has to be transformed twice

- $\overset{\curvearrowleft}{\beta_1} = \beta_1 \otimes (\beta_1 \otimes \varphi_2)^* \otimes (\beta_1 \otimes \beta_2 \otimes \varphi_1)^*$

Order of Evaluation

- Using Prop., we can use the insights from the canonical examples for feedback constraints of arbitrary lengths

Order of Evaluation

- Using Prop., we can use the insights from the canonical examples for feedback constraints of arbitrary lengths
- Have seen in all canonical example that there is one feedback constraint that can be safely resolved first

Order of Evaluation

- Using Prop., we can use the insights from the canonical examples for feedback constraints of arbitrary lengths
- Have seen in all canonical example that there is one feedback constraint that can be safely resolved first
- Using the established ordering of feedback constraints, this is always the feedback with the largest ordering index

Order of Evaluation

- Using Prop., we can use the insights from the canonical examples for feedback constraints of arbitrary lengths
- Have seen in all canonical example that there is one feedback constraint that can be safely resolved first
- Using the established ordering of feedback constraints, this is always the feedback with the largest ordering index
- General procedure:
  1. Transform feedback constraint F_l with largest index l and exchange β_{i_l} with $\overset{\curvearrowleft}{\beta}_{i_l}$
  2. Remove F_l from set of feedback constraints
  3. Repeat until no feedback constraints are left in the system

Order of Evaluation

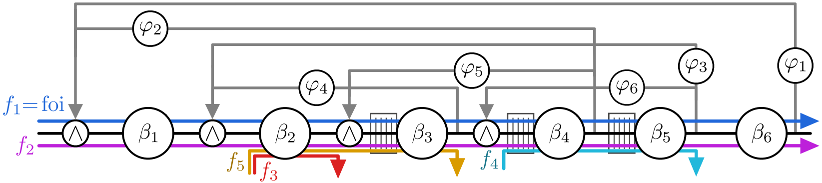


Figure: Example network before transformation.

Order of Evaluation

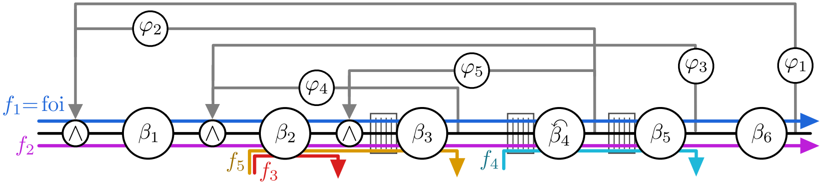


Figure: First transformation.

Order of Evaluation

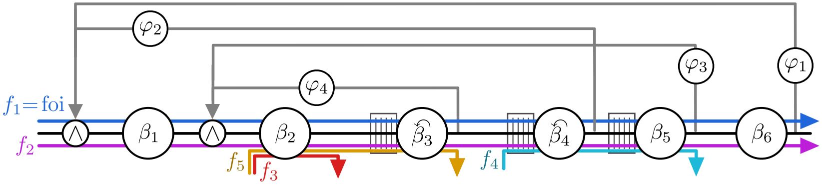


Figure: Second transformation.

Order of Evaluation

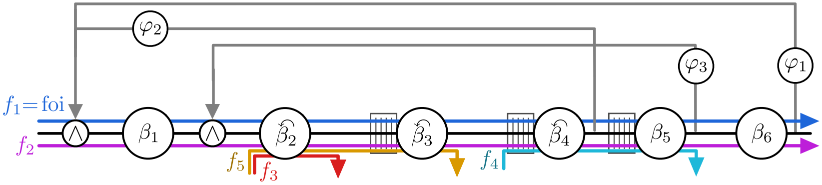


Figure: Third transformation.

Order of Evaluation

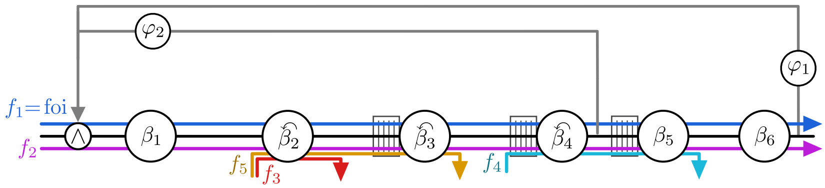


Figure: Fourth transformation.

Order of Evaluation

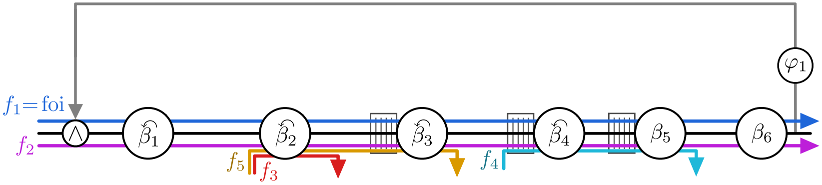


Figure: Fifth transformation.

Order of Evaluation

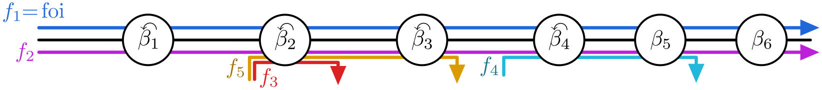


Figure: Transformed open-loop system.

Deriving Per-Flow Performance Bounds

1 Previous NC Work on Feedback

2 Extending Analysis to Interdependent Feedback Constraints

3 Deriving Per-Flow Performance Bounds

4 Conclusion

Obtaining A Residual Service Curve

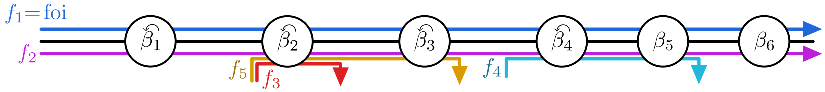


Figure: Open-loop system of the introductory example.

- Start with an open-loop system, where each transformed node offers a *min-plus* service curve

Obtaining A Residual Service Curve

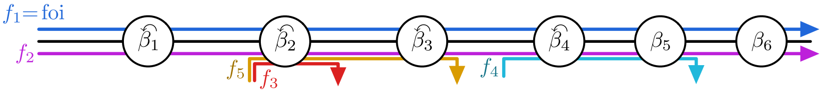


Figure: Open-loop system of the introductory example.

- Start with an open-loop system, where each transformed node offers a *min-plus* service curve
- Interested in performance bounds for a flow of interest f_1

Obtaining A Residual Service Curve

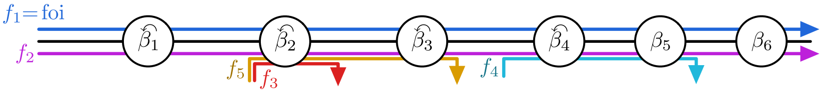


Figure: Open-loop system of the introductory example.

- Start with an open-loop system, where each transformed node offers a *min-plus* service curve
- Interested in performance bounds for a flow of interest f_1
- Use general residual service curve formula from [Bouillard et al., 2018]

$$\beta_i^{\mathrm{res}} = \beta_i - \overline{\alpha}_i^{\mathrm{c}}$$

with $\overline{\alpha}_i^c$ the maximal arrival curve for the cross flows

Obtaining Per-Flow Performance Bounds

- Use result from [Hamscher et al., 2024] to obtain performance bounds

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 min-plus service curve $\beta \in \mathcal{F}$, with $\xi := \beta_{\downarrow}$ its lower non-decreasing closure. The backlog $q(t)$ satisfies for all t

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

The virtual delay $d(t)$ satisfies for all $t \geq 0$

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

with $z(\underline{\alpha}, \xi) := \inf\{\tau \geq 0 | \underline{\alpha} \otimes \xi(\tau) \geq 0\}$

Conclusion

1 Previous NC Work on Feedback

2 Extending Analysis to Interdependent Feedback Constraints

3 Deriving Per-Flow Performance Bounds

4 Conclusion

Closing Remarks

- Identified and showed how different interdependence types between feedback constraints are resolved

Closing Remarks

- Identified and showed how different interdependence types between feedback constraints are resolved
- Here, considered tandem networks
  - Lexicographical order of feedback constraints is enough to identify order of evaluation

Closing Remarks

- Identified and showed how different interdependence types between feedback constraints are resolved
- Here, considered tandem networks
  - Lexicographical order of feedback constraints is enough to identify order of evaluation
- Order of evaluation for feed-forward networks should be straightforward, but left to be proven

Closing Remarks

- Identified and showed how different interdependence types between feedback constraints are resolved
- Here, considered tandem networks
  - Lexicographical order of feedback constraints is enough to identify order of evaluation
- Order of evaluation for feed-forward networks should be straightforward, but left to be proven
- Idea: build precedence graph for the feedback constraints and find its topological order

⇒ Transform feedback constraints in the order provided by the topological order

Closing Remarks

- Identified and showed how different interdependence types between feedback constraints are resolved
- Here, considered tandem networks
  - Lexicographical order of feedback constraints is enough to identify order of evaluation
- Order of evaluation for feed-forward networks should be straightforward, but left to be proven
- Idea: build precedence graph for the feedback constraints and find its topological order

$\Rightarrow$ Transform feedback constraints in the order provided by the topological order

- Applying this method to tandem networks provides the correct order of evaluation

Bose, A., Jiang, X., Liu, B., and Li, G. (2006).
Analysis of manufacturing blocking systems with network calculus.
Perform. Evaluation, 63(12):1216–1234.

Boudec, J. L. and Thiran, P. (2001).
Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, volume 2050 of *Lecture Notes in Computer Science*.
Springer.

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

Bouillard, A., Phan, L. T. X., and Chakraborty, S. (2009).
Lightweight modeling of complex state dependencies in stream processing systems.
In *15th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2009, San Francisco, CA, USA, 13-16 April 2009*, pages 195–204. IEEE, IEEE Computer Society.

Chang, C. (2000).
Performance Guarantees in Communication Networks.
Telecommunication Networks and Computer Systems. Springer.

Hamscher, A., Constantin, V., and Schmitt, J. B. (2024).
Extending network calculus to deal with min-plus service curves in multiple flow scenarios.
In *30th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2024, Hong Kong, May 13-16, 2024*, pages 95–107. IEEE, IEEE.

Any questions?

Generalization To Arbitrary Length

- Can generalize the Theorem to transform a system with a feedback constraint of arbitrary length

Proposition

Let a flow f traverse a tandem of n nodes, each offering a service curve $\beta_k, k \in \{1, \ldots, n\}$, with a single feedback constraint $F = (i, j, \varphi)$. The tandem offers an end-to-end service curve to the flow f

$$D \geq A \otimes \beta_{\mathrm{e2e}}, \text{ with } \beta_{\mathrm{e2e}} := \bigotimes_{k=1}^{i-1} \beta_k \otimes \overset{\curvearrowleft}{\beta_i} \otimes \bigotimes_{l=i+1}^{n} \beta_l, \tag{1}$$

and a transformed service curve at node i

$$\overset{\curvearrowleft}{\beta_i} := \beta_i \otimes \left(\bigotimes_{k=i}^{j} \beta_k \otimes \varphi \right)^{*}. \tag{2}$$

Compounded Interdependency

Note that in the case of $i_k = i_l$ and $j_k = j_l$, i.e., the extreme case of compounded feedback spanning the same set of nodes, we only need to evaluate a single feedback loop, however, with feedback constraint $\varphi_k \wedge \varphi_l$. Using distributivity,

$$A' = A \wedge (D \otimes \varphi_k) \wedge (D \otimes \varphi_l) = A \wedge D \otimes (\varphi_k \wedge \varphi_l).$$