

LINCS and ERC-NEMO Workshop  
“Performance Guarantees in Wireless  
Networks”, March 8-9, Palaiseau, France

# Time Sensitive Networks, Network Calculus and Clock Non-idealities

Jean-Yves Le Boudec

EPFL I&C, Lausanne, Switzerland

Joint work with Thomas Ludovic, Ehsan Mohammadpour and Hossein Tabatabaee



Title: Time Sensitive Networks, Network Calculus and Clock Non-idealities

Abstract: Time Sensitive Networks offer guarantees on worst-case delay, worst-case delay variation and zero congestion loss; in addition, they provides mechanisms for packet duplication in order to hide residual losses due to transmission errors. They find applications in many areas such as factory automation, embedded and vehicular networks, audio-visual studio networks, and in the front-hauls of cellular wireless networks. In this talk we will describe how network calculus can be used to analyze time sensitive networks with components such as packet ordering and duplicate removal functions, schedulers, regulators and dampers. We will also explain why clock non-idealities matter, and will describe how to take them into account.

# Contents

1. Time Sensitive Networks, Network Calculus and Single Node Analysis
2. Network Analysis
3. Regulators
4. Clock Non Idealities
5. Other Bells and Whistles

# 1. Time Sensitive Networks

= deterministic service: **upper bounds** on end-to-end **delay** and **delay-jitter** + **zero congestion loss**.

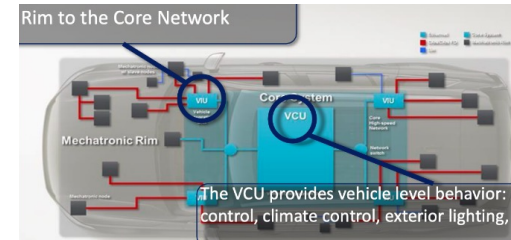
Congestion control with feedback is not an option here.

Proven bounds are required.

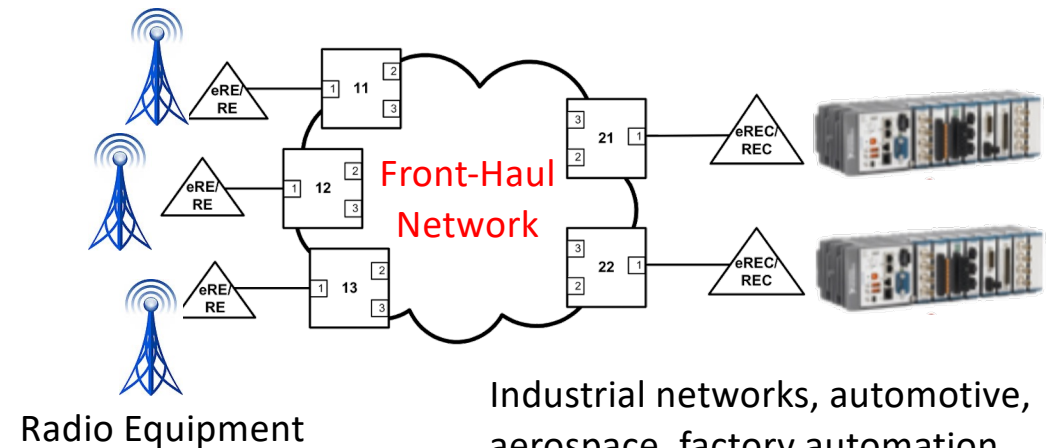
## Standardization:

MAC-layer networks: IEEE TSN (Time Sensitive Networking)

IP and MPLS networks: IETF Detnet (Deterministic Networking)



From [Navet et al,2020]



Industrial networks, automotive, aerospace, factory automation.  
Studio networking  
Front-haul of cellular networks  
Distributed games  
Low latency on-demand video

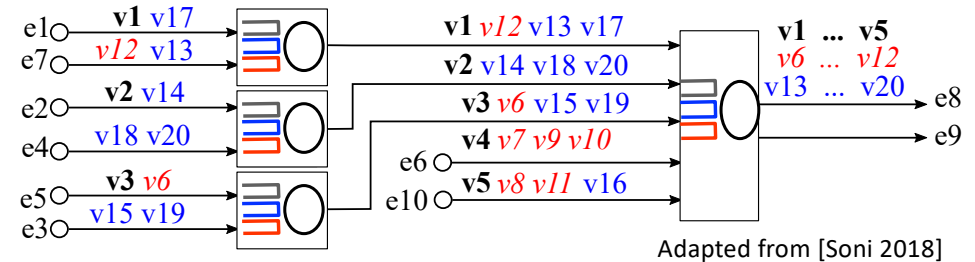
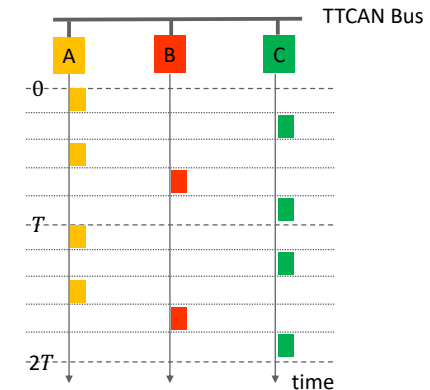
# How can a Network Offer a Deterministic Service ?

1. Every flow is **constrained at source** (*arrival curve constraint*) (T-SPEC)
2. The network nodes offer a guaranteed service to flows or classes of flows

**synchronous**: e.g Time Triggered CAN bus: every flow is scheduled on bus (not our focus today)

**asynchronous**: e.g. switch/router network

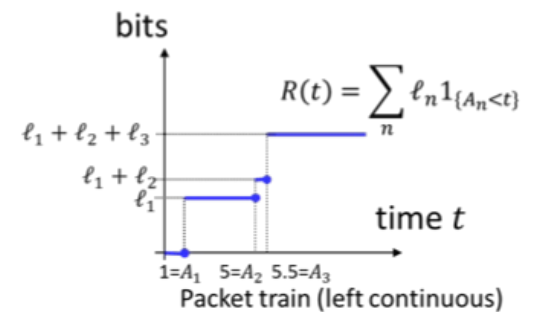
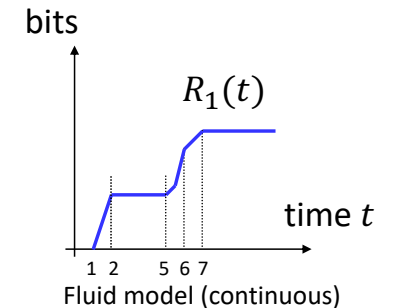
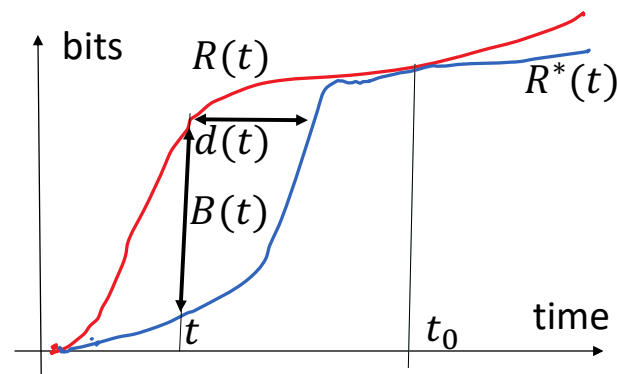
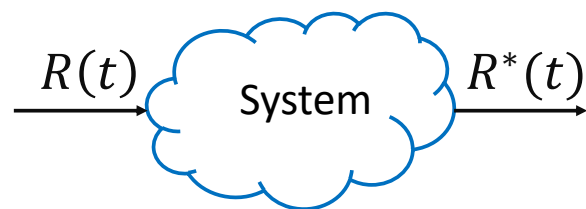
- a) Flows are assigned to a small number of **classes** with different quality of service requirements
- b) At every node, traffic of a given class is FIFO; a **scheduler** shares bandwidth and buffer between classes (e.g. Deficit Round Robin (DRR) [Shreedhar 1996], Credit Based Shaper (CBS, TSN))



# Analysis of Deterministic Networks uses Network Calculus

Given source constraints and schedulers, what are the worst-case delay, jitter and backlog?

- Flows are modelled with cumulative arrival functions,  $R(t)$ , non-decreasing with  $R(0) = 0$ , or, for packetized flows, with point processes (packet trains)  $(A, L)$
- Delay and backlog are derived



$$d(t) = \inf \{d \text{ s.t. } R(t) \leq R^*(t + d)\}$$

(horizontal deviation)

# Arrival Curves

Flow with cumulative function  $R(t)$  has  $\alpha$  as (maximal) **arrival curve** if

$$R(t) - R(s) \leq \alpha(t - s) \text{ for any } t \geq s \geq 0$$

where  $\alpha$  is a monotonic nondecreasing function  $\mathbb{R}^+ \rightarrow [0, +\infty]$

This is equivalent to  $R \leq R \otimes \alpha$ , where  $\otimes$  denotes min-plus convolution:

$$(f_1 \otimes f_2)(t) = \inf_{s \geq 0} (f_1(s) + f_2(t - s))$$

and, for a point process model, to

$$A_n - A_m \geq \alpha^\downarrow(\ell_m + \dots + \ell_n), \forall m, n, 1 \leq m \leq n$$

where  $\alpha^\downarrow$  is the lower-pseudo inverse of  $\alpha$ .

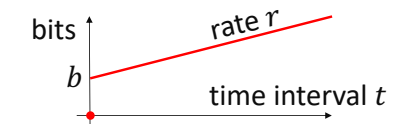
E.g. for  $\alpha(t) = rt + b$ ,  $\alpha^\downarrow(x) = \frac{(x-b)^+}{r}$

[Le Boudec 2018]

token bucket constraint  $(r, b)$

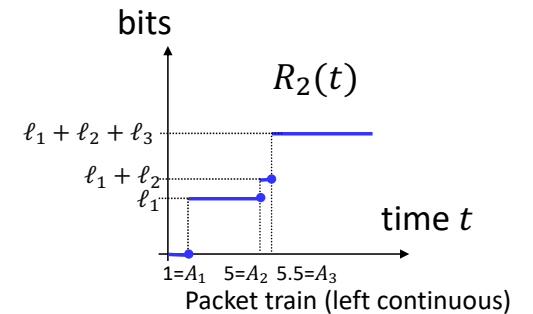
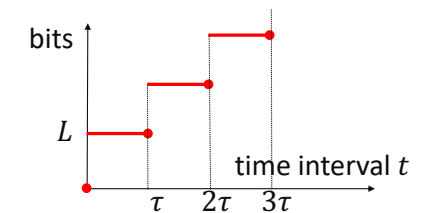
with rate  $r$  and burst  $b$ :

$$\alpha(t) = rt + b$$



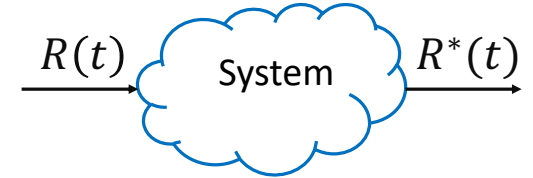
periodic stream of packets of size  $\leq L$ :

$$L: \alpha(t) = L \left\lceil \frac{t}{\tau} \right\rceil$$



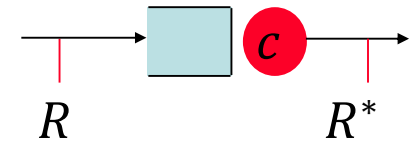
## Service Curves

System offers to  $(R, R^*)$  a (minimal) **service curve**  $\beta$  if  $R^* \geq R \otimes \beta$   
 i.e.  $\forall t \geq 0, \exists s \in [0, t]: R^*(t) \geq R(s) + \beta(t - s)$ , where  $\beta$  is a  
 function  $\mathbb{R}^+ \rightarrow \mathbb{R} \cup \{+\infty\}$ .



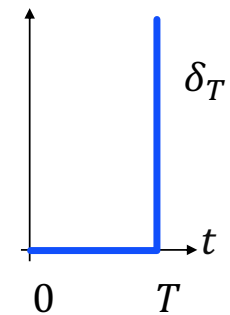
System  $\mathcal{S}$  offers to a flow a **strict service curve**  $\beta$  if for any  $s < t$  in a  
 backlogged period, we have  $R^*(t) - R^*(s) \geq \beta(t - s)$

$\mathcal{S}$  is typically a single queuing point; example: constant rate server  
 with line rate  $c$  has strict service curve  $\beta(t) = ct$



$\beta$  is a strict service curve  $\Rightarrow \beta$  is a service curve; converse is not true.

Sometimes it is convenient to model a system as a black  
 box with known delay upper bound  $T$ . For a node that is FIFO for this  
 flow:  $\text{delay} \leq T \Leftrightarrow$  node offers to this flow a service curve  $\delta_T$ . Not a  
 strict service curve.



$$\begin{aligned} \delta_T(t) &= 0 \text{ if } t \leq T \\ \delta_T(t) &= +\infty \text{ if } t > T \end{aligned}$$



# Examples of Service Curves

**Static Priority** (non-pre-emptive).

High prio:  $\beta_H(t) = (ct - MTU_L)^+$  (strict service curve)

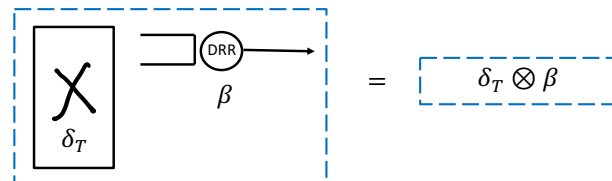
Low prio: when high priority constrained by  $\alpha(t) = rt + b, r < c$ :

$\beta_L(t) = ((c - r)t - b)^+$  (not a strict service curve)

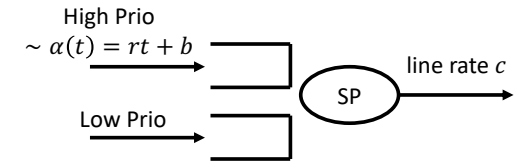
$\beta'_L(t) = ((c - r)t - b - MTU_L)^+$  (strict service curve)

Service curve characterizations exist for most schedulers used in time-sensitive networks: CBS, **DRR**, IEEE AVB, etc.

**Concatenation** of service curve elements  $\beta_1, \beta_2$  has service curve  $\beta_1 \otimes \beta_2$

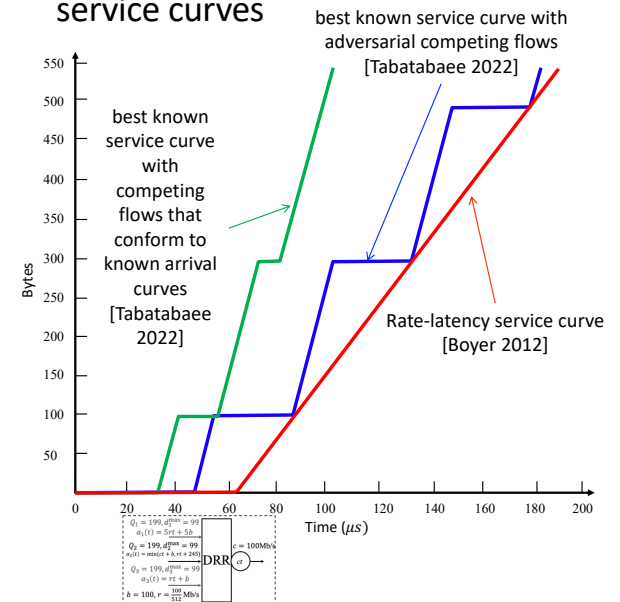


[De Azua 2014] [Bouillard 2018] [Tabatabaee 2022]



$MTU_L$  = max packet size, low prio

**Deficit Round Robin (DRR)**  
service curves



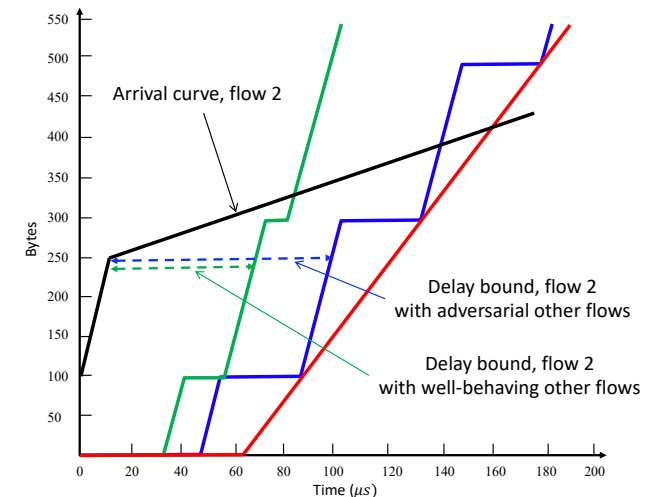
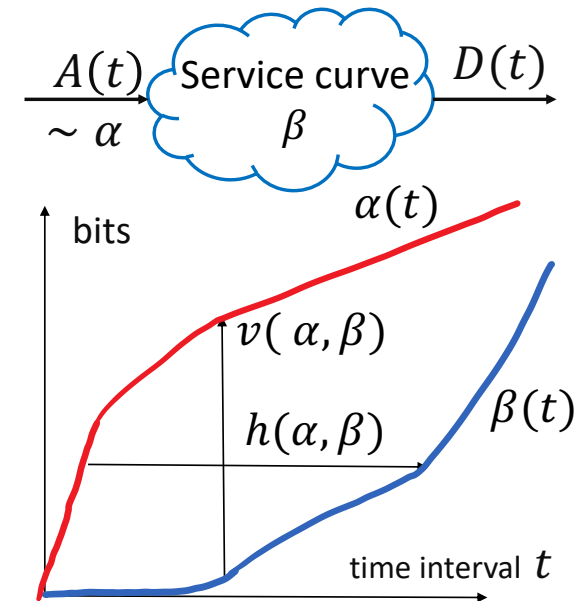
## Three Tight Bounds

1. **backlog**  $\leq v(\alpha, \beta) = \sup_t (\alpha(t) - \beta(t))$
2. if FIFO for this flow, **delay**  $\leq h(\alpha, \beta)$
3. **output** is constrained by arrival curve  

$$\alpha^*(t) = \sup_{u \geq 0} (\alpha(t+u) - \beta(u))$$
 i.e.  $\alpha^* = \alpha \oslash \beta$  (deconvolution)

Jitter bound =  $h(\alpha, \beta)$  – delay lower bound

Delay bound can be improved to  $h(\alpha - L_{min}, \beta) + \frac{L_{min}}{c}$   
 if we know line rate of server [Mohammadpour 2019]



## 2. Network Analysis: Per-flow Networks

**Per-flow** network: network nodes offer guarantees to individual flows, e.g. IETF IntServ.

Solution: apply concatenation of service curves.

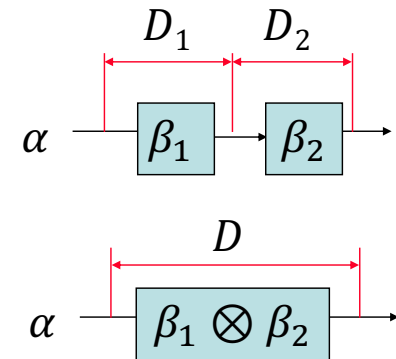
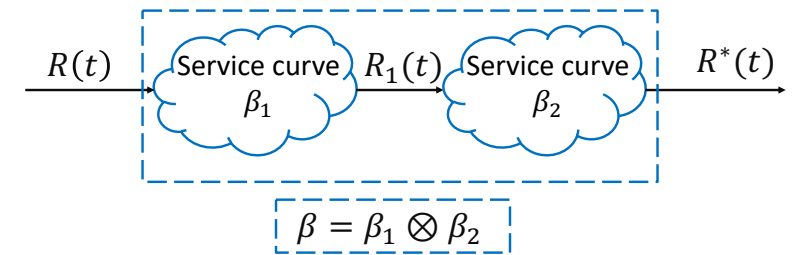
**Pay bursts only once**: end-to-end delay bound computed *node-by-node* (also accounting for increased burstiness at node 2):

$$D_1 + D_2 = \frac{2b + rT_1}{R} + T_1 + T_2$$

computed *by concatenation*:

$$D = \frac{b}{R} + T_1 + T_2$$

i.e. worst cases cannot happen simultaneously



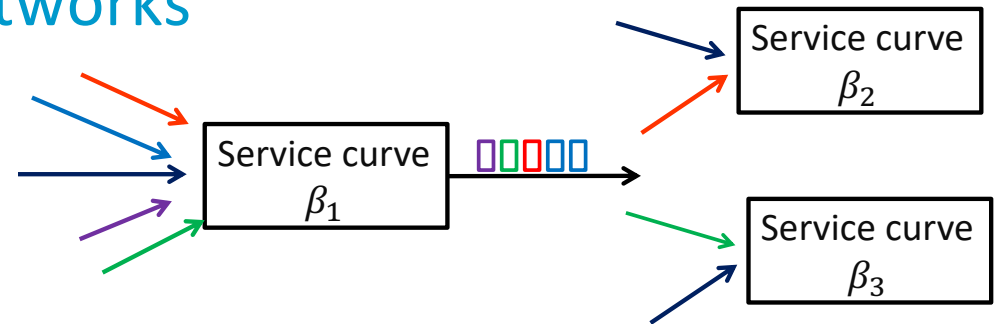
$$\begin{aligned}\alpha(t) &= rt + b \\ \beta_1(t) &= R(t - T_1)^+ \\ \beta_2(t) &= R(t - T_2)^+ \\ r &\leq R\end{aligned}$$

## Network Analysis: FIFO Per-Class Networks

Most time-sensitive networks are **FIFO per class**; schedulers (such as DRR) separate classes; service guarantee is to the aggregate of all flows of this class.

Service curves → analyze one separate FIFO network model per class.

Analysis of a FIFO (single class) network is NP-hard [Bouillard 2010]. Worst-case delays can be computed with ELP (Exponential Linear Programming) [Bouillard 2014].



# Total Flow Analysis (TFA) and Polynomial Linear Programming (PLP)

**Total Flow Analysis** (TFA [Schmitt 2006], TFA++ [Mifdaoui 2017])

a) **Propagated arrival curve** of flow inside the network is computed by

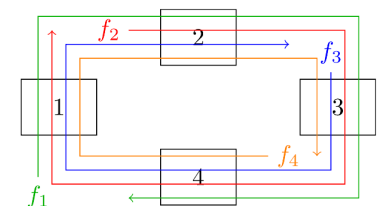
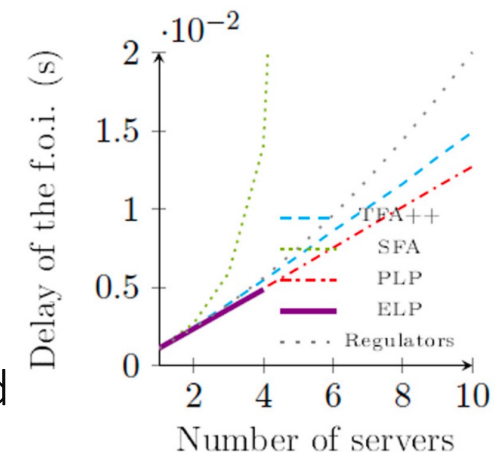
$$\alpha^*(t) = \alpha(t + \text{delay jitter bound between source and here})$$

b) **Delay** at every node is computed from propagated arrival curves.

End-to-end delay bound is sum of nodal bounds on path.

In a feedforward network of depth  $d$ , start at edge nodes and stop in  $d$  iterations. In a generic network, iterate a) and b) at all nodes until convergence to a **fixpoint** or move to infinity. If convergence, the bound are valid. If divergence, we don't know. [Thomas 2019, Plassart 2022, Tabatabaee 2023].

**PLP** (Polynomial Linear Programming): relaxation of ELP, with polynomial complexity, uses TFA (and other) bounds as constraints, applies to generic topologies. Generally provides better bounds and stability region than TFA but is more complex and limited to concave arrival curves and convex service curves. [Bouillard 2022].



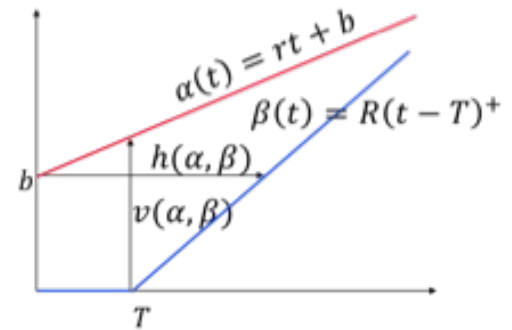
From [Bouillard 2022]

## Stability of a FIFO Network

Every flow  $f \in \mathcal{F}$  constrained by  $\alpha_f(t) = r_f t + b_f$  at source. Route of flow  $f$  is fixed.  $F_i \subset \mathcal{F}$  is the set of flows passing through node  $i$ . Every node  $i \in \mathcal{I}$  is FIFO and offers to the aggregate of flows  $f \in F_i$  a service curve  $\beta_i(t) = R_i(t - T_i)^+$ . Load factor  $u = \max_i \left( \frac{\sum_{f \in F_i} r_f}{R_i} \right)$ .  $\mathcal{F}, \mathcal{I}$  finite. Network **underloaded**:  $u < 1$ ; **overloaded**:  $u > 1$ ; **critical**:  $u = 1$ .

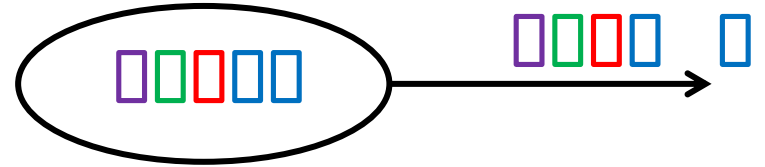
One network instance  $= (\mathcal{F}, r, b, F, \mathcal{I}, R, T)$  is **stable** if there is a bound on all delays (or backlogs), that is valid for any execution trace of the network.

- An overloaded FIFO network is not stable. A feed-forward network that is underloaded or critical is stable.
- For any  $\varepsilon > 0$  there is an **unstable underloaded** FIFO network with load factor  $u < \varepsilon$  [Andrews 2009]
- Every underloaded **ring** is stable [Tassiulas 1996].



When PLP or TFA does not converge, it may be that network is truly unstable or not. Stability conditions are still an open research issue.

### 3. Regulators



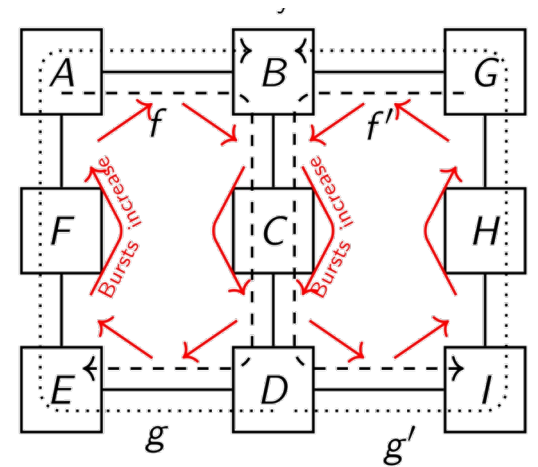
**Regulator** (= shaper) delays packets in order to limit burstiness to a prescribed value (i.e. enforces an arrival curve constraint).

Example: Token Bucket regulator.

Typically placed at network **edge** to protect from misbehaving sources.

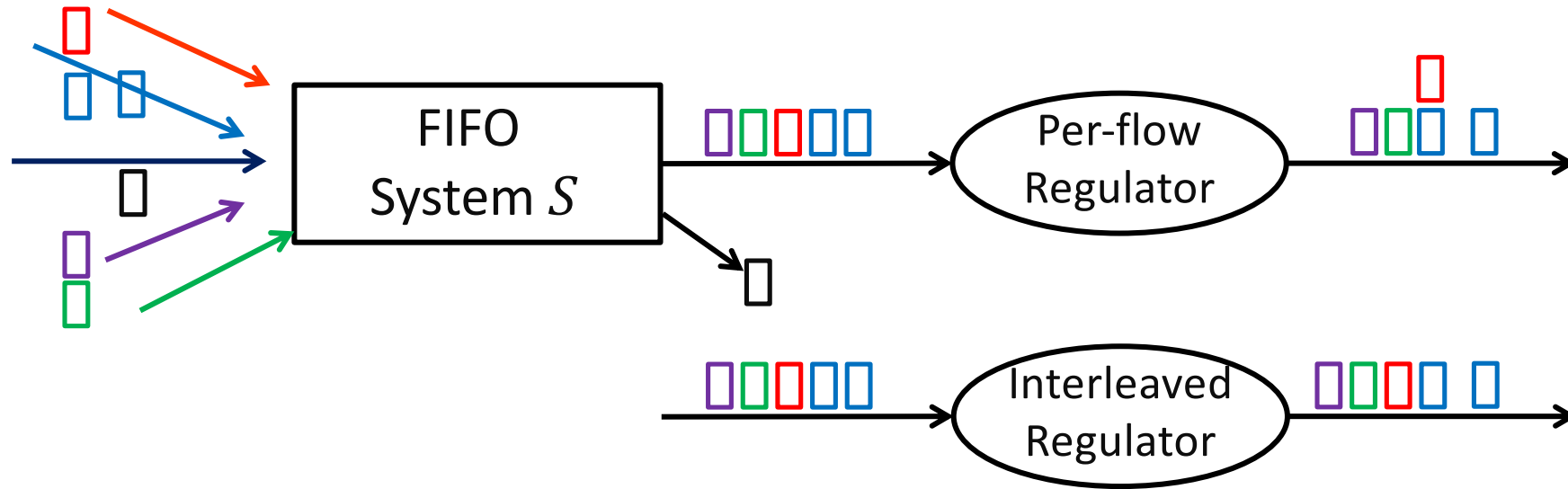
In per-class networks, burstiness of every flow increases at every hop as a function of other flows' burstiness. Increased burstiness causes increased burstiness (**cascade**).

Propagated burstiness is computed by PLP / TFA as solution to a fixpoint problem. Cyclic dependencies may cause bad delays.



Regulators can be used to reduce burstiness or even break cyclic dependencies [Thomas 2019].

## Regulators Avoid Cascading Burstiness in Per-Class Networks



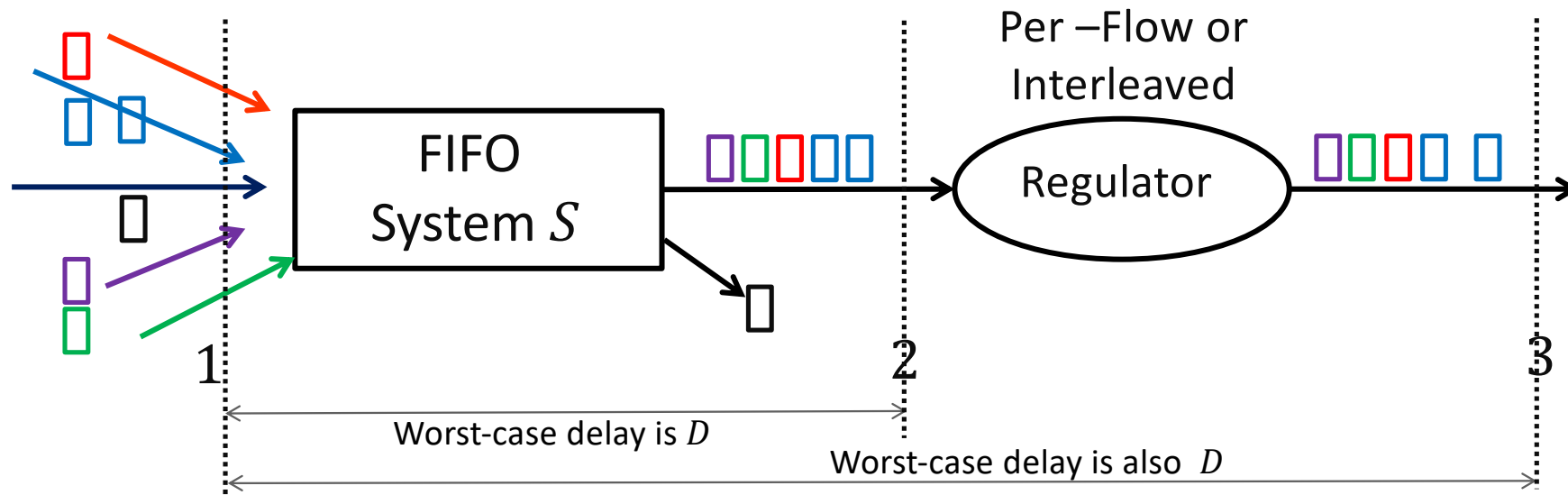
**Per flow regulator:** one state + one queue per flow.

**Interleaved regulator:** one state per flow + one global queue:

- packet at head of queue is examined against the arrival constraint (e.g. rate  $r_f$  and burstiness  $b_f$ ) of its flow  $f$ ; this packet is delayed if it came too early; different flows in same queue can have different arrival constraints;
- packets not at head of queue wait for their turn to come [Specht 2016].



## Regulators do not Increase Worst Case Delay



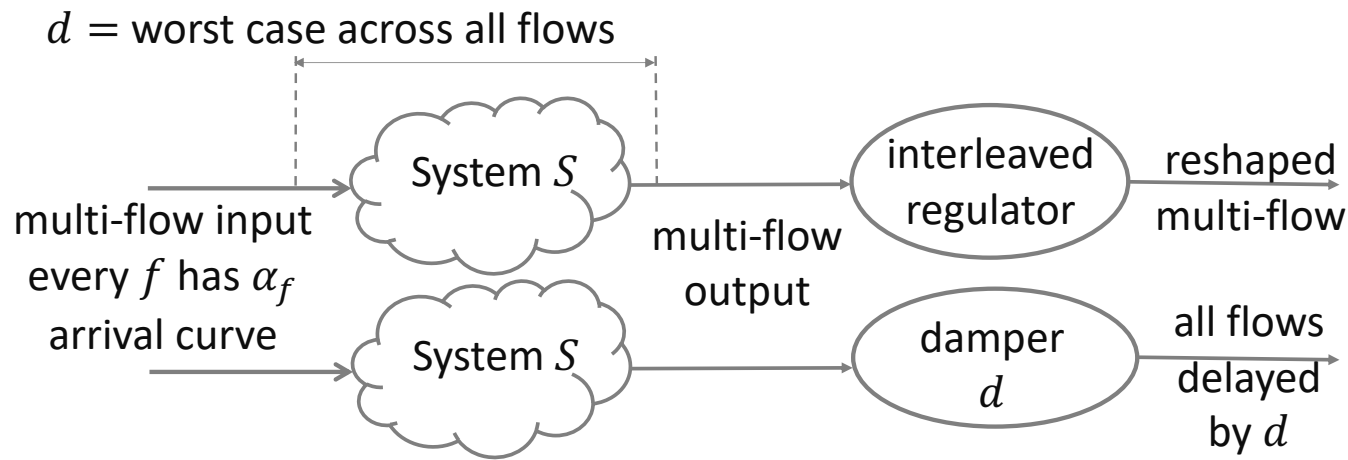
Assume  $S$  is FIFO per flow (per-flow regulator) or globally (interleaved regulator).

Assume every flow satisfies some arrival constraint at 1 (e.g. rate and burstiness) and regulators enforces same constraint at 3.

The worst case delay 1 – 3 is the same as the worst-case delay 1 – 2 [Le Boudec 2018].  
(Reshaping-for-free property).

⇒ Used in Time Sensitive Networks where regulators break cyclic dependencies

## Proof



- The interleaved regulator is the **minimal** FIFO system that delivers output flows  $\sim \alpha_f, \forall f$ .
- Replace minimal regulator by **damper** [Verma et al 1991]:  
Damper forces total delay of input to be exactly  $d$ ; Damper is causal and FIFO if  $d$  is  $\geq$  worst-case delay through  $S$ .
- Damper delivers output flows  $\sim \alpha_f, \forall f \Rightarrow$  multi-flow output delayed by  $d$  is no earlier than reshaped multi-flow.

## 4. Clock Non Idealities

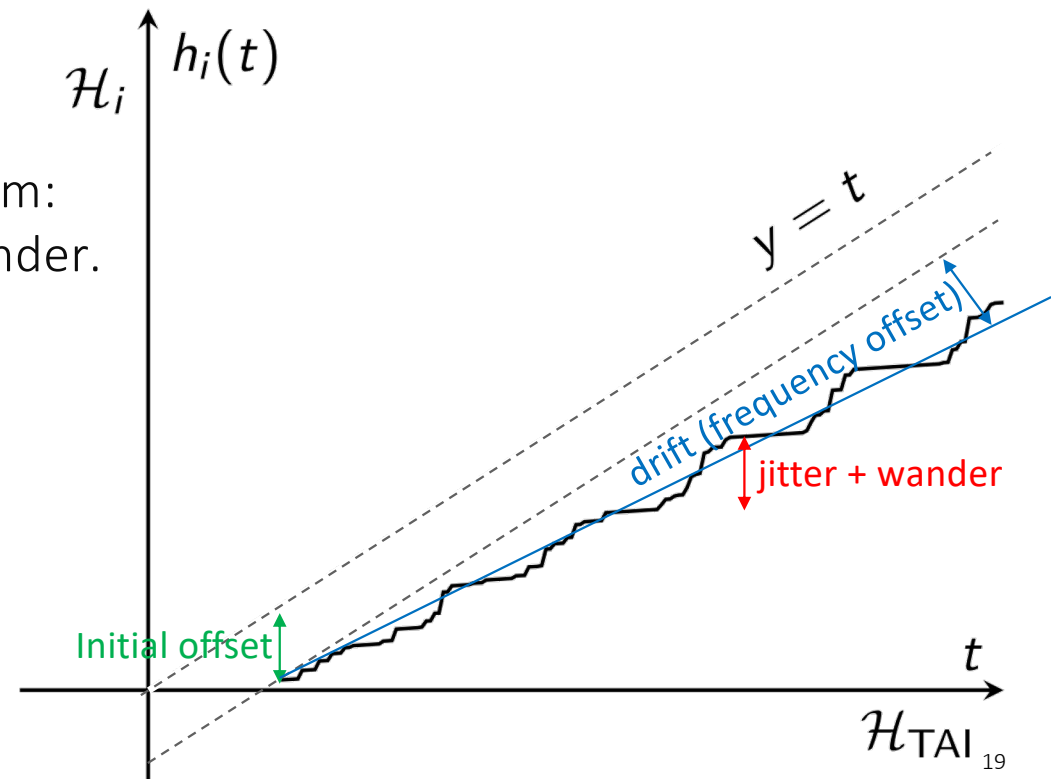
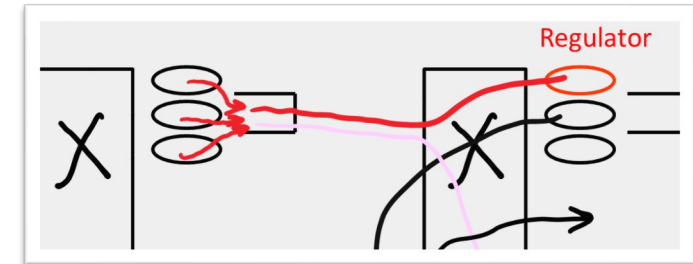
Previous theory assumes perfect time everywhere.  
In reality, nodes use local clocks that are not ideal.

- **tight sync** (PTP, White Rabbit, GPS) :  
timestamping error  $\leq \omega \approx 10\text{ns} - 1\mu\text{s}$
- **loose sync** (NTP):  $\omega \approx 1\text{ms} - 1\text{s}$
- **no sync**: timestamping error  $\omega$  unbounded;  
measurement of time interval on same system:  
error is bounded by clock drift, jitter and wander.

[ITU-T 1996]

Regulators use time measurements to decide  
when a packet can be released.

What is the effect of clock non ideality ?



## Clock Model in Network Calculus [Thomas 2020]

Measurement of a time interval is performed with one clock  $\rightarrow d$   
and with another clock  $\rightarrow d'$

Time synchronization error:  $d' - d \leq 2\omega$

Clock jitter and wander:  $d' \leq \rho d + \eta$

This gives the **change-of-clock inequalities**

$$\max\left(0, \frac{d - \eta}{\rho}, d - 2\omega\right) \leq d' \leq \min(\rho d + \eta, d + 2\omega)$$

Model is symmetric, i.e. same inequalities if we exchange  $d' \leftrightarrow d$

Relative error on estimation of delays is, in general,  $\approx 10^{-4}$ , i.e. negligible. However there are some corner cases.

$\omega$  = time error bound  
=  $1\mu s$  in TSN with PTP;  
=  $+\infty$  if no  
synchronization

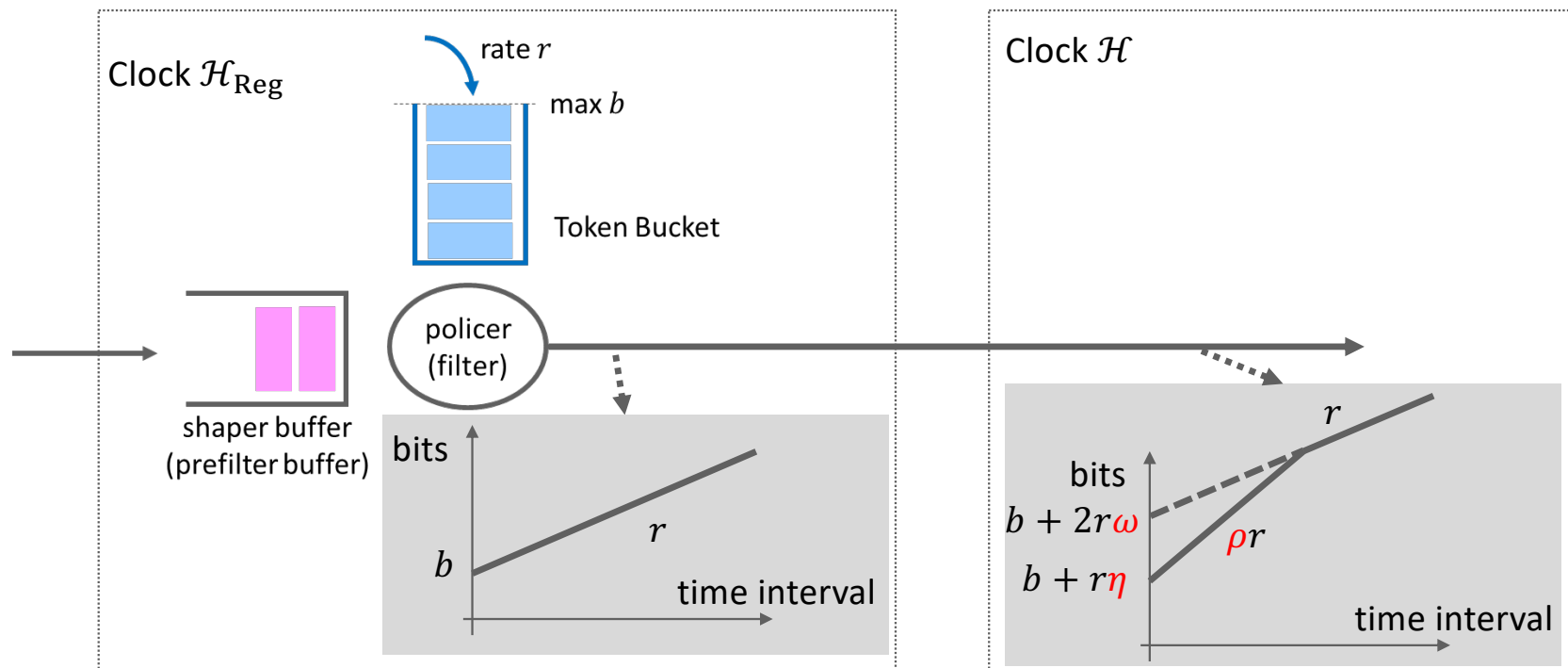
$\rho$  = clock-stability  
bound  
= 1.0001 (e.g. in TSN)

$\eta$  = timing-jitter bound  
= 2ns (e.g. in TSN)

## Change of Clock: Arrival Curves

Assume a flow satisfies a token bucket constraint  $(r, b)$  when observed with clock  $\mathcal{H}_{\text{Reg}}$   
 i.e. arrival curve constraint  $\alpha^{\mathcal{H}_{\text{Reg}}}(t) = rt + b$

When observed with some other clock  $\mathcal{H}$ , it satisfies the arrival curve constraint  
 $\alpha^{\mathcal{H}}(t) = \min(\rho r t + b + r\eta, rt + b + 2r\omega)$



# Consequences for Non-Adapted Regulators [Thomas 2020]

*Non adapted* regulator : uses same nominal arrival curve as at source.

Perfect clocks:

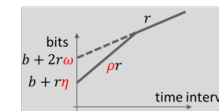
- Regulator does not increase worst-case delay

Non-synchronized network:

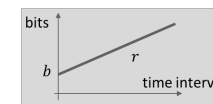
- Per-flow and interleaved regulator unstable (unbounded delay).

Synchronized network:

- Per-flow regulator incurs delay penalty up to  $4\omega$ ;
- Interleaved regulator is **unstable**  
 $\Rightarrow$  must be adapted,  
 e.g. with rate-and-burst cascade



In regulator's clock,  
flow satisfies this  
constraint at source



Source

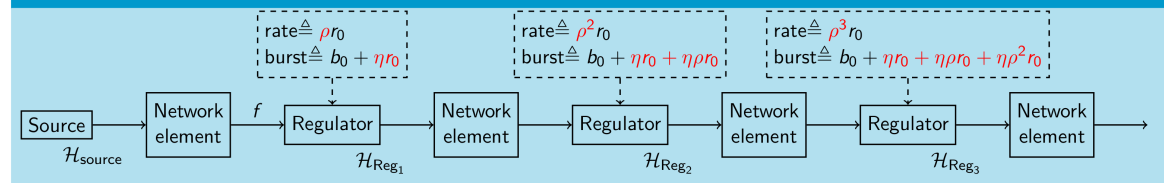
Flow constrained by  $\alpha(t) = rt + b$   
in local clock

Network  
Elements

Non adapted  
Regulator  
(PFR or IR)

Implements constraint  
 $\alpha(t) = rt + b$   
in local clock

## Rate-and-burst cascade Works with PFR or IR



## 5. Other Bells and Whistles

Packet **re-ordering** occurs due to multi-paths and packet replication.

⇒ Re-sequencing buffers are used. Network calculus was extended to account for them [Mohammadpour 2021]

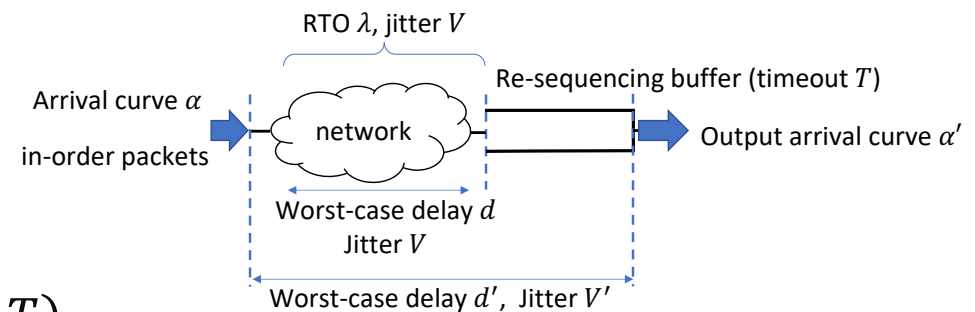
- Lossless network:

$$d' = d, V' = V \text{ and } \alpha'(t) = \alpha(t + V)$$

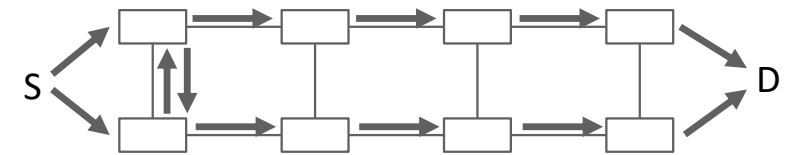
(re-sequencing is for free)

- Lossy network:

$$d' = d + T, V' = V + T \text{ and } \alpha'(t) = \alpha(t + V + T).$$



**Packet replication and removal** is used to repair non-congestion losses. It causes mis-ordering and increases burstiness. Network calculus was extended to account for this [Thomas 2022]



Any combination of failures that leaves at least one path up is masked ("0 msec repair") [IEEE 802.1CB]

## Towards Stochastic Network Calculus ...

Stochastic arrival curves [Ciucu 2012]

$$\text{SBB: } \forall s \leq t, \sigma > 0: \mathbb{P}(A(s, t) > f(t - s) + \sigma) \leq \varepsilon(\sigma)$$

$$\text{S}^2\text{BB: } \forall t, \sigma > 0: \mathbb{P}\left(\sup_{s \leq t} A(s, t) > f(t - s) + \sigma\right) \leq \varepsilon(\sigma)$$

$$\text{S}^3\text{BB: } \forall \sigma > 0: \mathbb{P}\left(\sup_{s \leq t} A(s, t) > f(t - s) + \sigma\right) \leq \varepsilon(\sigma)$$

$\text{S}^2\text{BB}$  can be obtained from Hoeffding bounds for independent, arrival-curve constrained sources [Vojnovic 2003] – metric is  $\mathbb{P}(Q(t) \leq b)$ , obtains significant statistical multiplexing gain.

$\text{S}^3\text{BB}$  cannot apply nontrivially to ergodic processes, but applies for example to periodic sources, can apply to  $\mathbb{P}(\forall t, Q(t) \leq b)$

Stochastic service [Jiang 2008, Fidler 2015, Nikolaus 2019] uses MGF bounds. Could be used to model wireless links. Hard to obtain tight bounds.



# Tools

- The [DiscoDNC](#) is an academic Java implementation of the network calculus framework.<sup>[10]</sup>
- The [RTC Toolbox](#) is an academic Java/[MATLAB](#) implementation of the Real-Time calculus framework, a theory quasi equivalent to network calculus.<sup>[4][11]</sup>
- The [CyNC](#)<sup>[12]</sup> tool is an academic [MATLAB](#)/Symulink toolbox, based on top of the [RTC Toolbox](#). The tool was developed in 2004-2008 and it is currently used for teaching at [Aalborg university](#).
- The [RTaW-PEGASE](#) is an industrial tool devoted to timing analysis tool of switched Ethernet network (AFDX, industrial and automotive Ethernet), based on network calculus.<sup>[13]</sup>
- The [WOPANets](#) is an academic tool combining network calculus based analysis and optimization analysis.<sup>[14]</sup>
- The DelayLyzer is an industrial tool designed to compute bounds for Profinet networks.<sup>[15]</sup>
- [DEBORAH](#) is an academic tool devoted to FIFO networks.<sup>[16]</sup>
- [NetCalBounds](#) is an academic tool devoted to blind & FIFO tandem networks.<sup>[17][18]</sup>
- [NCBounds](#) is a network calculus tool in Python, published under BSD 3-Clause License. It considers rate-latency servers and token-bucket arrival curves. It handles any topology, including cyclic ones.<sup>[19]</sup>
- The Siemens Network Planner ([SINETPLAN](#)) uses network calculus (among other methods) to help the design of a [PROFINET](#) network.<sup>[20]</sup>
- [experimental modular TFA](#) (xTFA) is a Python code, support of the PhD thesis of Ludovic Thomas<sup>[21]</sup>

## Conclusion

Time Sensitive Networks require deterministic, proven bounds on delay, jitter, backlog and re-ordering.

Network Calculus provides theory and software tools for computing such bounds and for understanding operation of regulators, dampers, re-sequencing buffers or packet elimination functions.

Clock non-idealities can easily be incorporated. Regulators are dramatically affected, other systems not.

Stochastic Network calculus promises to apply to wireless networks.

# Thank You !

References are in the online version.

# References

- [Andrews 2009] Andrews, M., 2009. Instability of FIFO in the permanent sessions model at arbitrarily small network loads. *ACM Transactions on Algorithms (TALG)*, 5(3), pp.1-29.
- [Bennett 2002] Bennett, J.C., Benson, K., Charny, A., Courtney, W.F. and Le Boudec, J.Y., 2002. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. *IEEE/ACM Transactions on networking*, 10(4), pp.529-540.
- [Bondorf 2017] Bondorf, S., 2017, May. Better bounds by worse assumptions—Improving network calculus accuracy by adding pessimism to the network model. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1-7). IEEE.
- [Bouillard 2010] Bouillard, A., Jouhet, L. and Thierry, E., 2010, March. Tight performance bounds in the worst-case analysis of feed-forward networks. In *2010 Proceedings IEEE INFOCOM* (pp. 1-9). IEEE.
- [Bouillard 2014] Bouillard, A. and Stea, G., 2014. Exact worst-case delay in FIFO-multiplexing feed-forward networks. *IEEE/ACM Transactions on Networking*, 23(5), pp.1387-1400.
- [Bouillard 2018] Bouillard, A. Boyer, M. and Le Coronc, E. *Deterministic Network Calculus: from Theory to Practical Implementation*, ISTE Editions, 2018
- [Bouillard 2022] Bouillard, A., 2022. Trade-off between accuracy and tractability of network calculus in FIFO networks. *Performance Evaluation*, 153, p.102250.
- [Boyer 2010] Boyer, M., Navet, N., Olive, X. and Thierry, E., 2010, October. The PEGASE project: precise and scalable temporal analysis for aerospace communication systems with network calculus. In *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation* (pp. 122-136). Springer, Berlin, Heidelberg.
- [Chang 1997] Chang, C.S., 1997, April. A filtering theory for deterministic traffic regulation. In *Proceedings of INFOCOM'97* (Vol. 2, pp. 436-443). IEEE.
- [Ciucu 2012] Ciucu, F. and Schmitt, J., 2012, August. Perspectives on network calculus: no free lunch, but still good value. *SIGCOMM 2012*.
- [Cruz 1998] Cruz, R.L., 1998, March. SCED+: Efficient management of quality of service guarantees. In *Proceedings. IEEE INFOCOM'98*, IEEE
- [De Azua 2014] De Azua, J.A.R. and Boyer, M., 2014, October. Complete modelling of AVB in Network Calculus Framework. In *RTNS*(p. 55).
- [Fidler 2003] Fidler, M., 2003, February. Extending the network calculus pay bursts only once principle to aggregate scheduling. In *International Workshop on Quality of Service in Multiservice IP Networks* (pp. 19-34). Springer, Berlin, Heidelberg.
- [Geyer 2022] Geyer, F., Scheffler, A. and Bondorf, S., 2022. Network Calculus with Flow Prolongation—A Feedforward FIFO Analysis enabled by ML. *arXiv preprint arXiv:2202.03004*.
- [Grieu 2004] Grieu, Jérôme (Sept. 24, 2004). “Analyse et évaluation de techniques de commutation Ethernet pour l’interconnexion des systems avioniques.” PhD thesis. url: <http://ethesis.inp-toulouse.fr/archive/00000084>
- [Grigorjew 2020] Grigorjew, A., Metzger, F., Hoßfeld, T., Specht, J., Götz, F.J., Chen, F. and Schmitt, J., 2020. Asynchronous Traffic Shaping with Jitter Control. <https://opus.bibliothek.uni-wuerzburg.de>
- [IEEE 802.1CB] “IEEE Standard for Local and Metropolitan Area Networks—Frame Replication and Elimination for Reliability” (Oct. 2017). In: *IEEE Std 802.1CB-2017*, pp. 1–102. doi: 10.1109/IEEESTD.2017.8091139.
- [ITU-T 1996] Definitions and terminology for synchronization networks. ITU-T G.810.
- [ITU-T 2020 ] ITU-T, “ITU-T Y.3000-series - representative use cases and key network requirements for network 2030,” vol. Y.Sup67, 2020.
- [Jiang 2008] Jiang, Y. and Liu, Y., 2008. *Stochastic network calculus*. London: Springer.
- [Le Boudec 1996] Le Boudec, JY, “Network Calculus Made Easy”, preprint, Technical report EPFL-DI 96/2018, December 14, 1996
- [Le Boudec-Thiran 2001] Le Boudec, J.Y. and Thiran, P., 2001. Network calculus: a theory of deterministic queuing systems for the internet (Vol. 2050). Springer Science & Business Media. <https://leboudec.github.io/netcal/>

- [Le Boudec 2018] Le Boudec, J.Y., 2018. A theory of traffic regulators for deterministic networks with application to interleaved regulators. *IEEE/ACM Transactions on Networking*, 26(6), pp.2721-2733.
- [Lenzini 2006] Lenzini, L., Martorini, L., Mingozzi, E. and Stea, G., 2006. Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks. *Performance Evaluation*, 63(9-10), pp.956-987.
- [Maile 2020] Maile, L., Hielscher, K.S. and German, R., 2020, May. Network calculus results for TSN: An introduction. In *2020 Information Communication Technologies Conference (ICTC)* (pp. 131-140). IEEE.
- [Mifdaoui 2010] Mifdaoui, A. and Ayed, H., 2010, December. WOPANets: a tool for Worst case Performance Analysis of embedded Networks. In *2010 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD)* (pp. 91-95). IEEE.
- [Mifdaoui 2017] Mifdaoui, A. and Leydier, T., 2017, December. Beyond the accuracy-complexity tradeoffs of compositional analyses using network calculus for complex networks. In *10th International Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (co-located with RTSS 2017)* (pp. pp-1).
- [Mohammadpour 2018] Mohammadpour, E., Stai, E., Mohiuddin, M. and Le Boudec, J.Y., 2018, September. Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping. In *2018 30th International Teletraffic Congress (ITC 30)* (Vol. 2, pp. 1-6). IEEE.
- [Mohammadpour 2019] Mohammadpour, E., Stai, E. and Le Boudec, J.Y., 2019. Improved delay bound for a service curve element with known transmission rate. *IEEE Networking Letters*, 1(4), pp.156-159.
- [Mohammadpour 2021] Mohammadpour, E. and Le Boudec, J.Y., 2021. On Packet Reordering in Time-Sensitive Networks. *IEEE/ACM Transactions on Networking*.
- [Mohammadpour 2022] Mohammadpour, E. and Le Boudec, J.Y., 2022. Analysis of Dampers in Time-Sensitive Networks With Non-Ideal Clocks. *IEEE/ACM Transactions on Networking*.
- [Navet et al, 2020] Navet, N., Bengtsson, H.H. and Migge, J., 2020. Early-stage bottleneck identification and removal in TSN networks, <https://orbi.lu.uni/bitstream/10993/46282/1/AEC2020-UL-Volvo-RTaW-web.pdf>
- [Nikolaus 2019] Nikolaus, P., Schmitt, J. and Schütze, M., 2019. h-Mitigators: Improving your stochastic network calculus output bounds. *Computer Communications*, 144, pp.188-197.
- [Plassart 2022] Plassart, S. and Boudec, J.Y.L., 2021. Equivalent Versions of Total Flow Analysis. *arXiv preprint arXiv:2111.01827*.
- [RFC 4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S. and Perser, J., 2006. *Packet reordering metrics* (IETF RFC 4737).
- [Schmitt 2006] Schmitt, J.B. and Zdarsky, F.A., 2006, October. The disco network calculator: a toolbox for worst case analysis. In *Proceedings of the 1st international conference on Performance evaluation methodologies and tools* (pp. 8-es).
- [Shoushou 2020] R. Shoushou, L. Bingyang, M. Rui, W. Chuang, J.-Y. Le Boudec, E. Mohammadpour, and A. El Fawal, "A method for sending data packets and network equipment," China Patent, Jul., 2020
- [Shreedhar 1996] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, 1996
- [Soni 2018]: A. Soni, X. Li, J. Scharbarg, and C. Fraboul, "Optimizing network calculus for switched ethernet network with deficit round robin," in *2018 IEEE Real-Time Systems Symposium (RTSS)*, 2018.
- [Specht 2016] Specht, J. and Samii, S., 2016, July. Urgency-based scheduler for time-sensitive switched ethernet networks. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)* (pp. 75-85). IEEE.
- [Tabatabaee 2022] Tabatabaee SM, Le Boudec JY. Deficit round-robin: A second network calculus analysis. *IEEE/ACM Transactions on Networking*. 2022 Apr 13.
- [Tabatabaee 2023] S. Tabatabaee, M. Boyer, J.-Y. Le Boudec, J. Migge Efficient and Accurate Handling of Periodic Flows in Time-Sensitive Networks, accepted at RTAS 2023.

- [Tassiulas 1996] Tassiulas, L. and Georgiadis, L., 1996. Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACM transactions on networking*, 4(2), pp.205-208.
- [Thomas 2019] Thomas, L., Le Boudec, J.Y. and Mifdaoui, A., 2019, December. On cyclic dependencies and regulators in time-sensitive networks. In 2019 IEEE Real-Time Systems Symposium (RTSS) (pp. 299-311). IEEE.
- [Thomas 2020] Thomas, L. and Le Boudec, J.Y., 2020. On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2), pp.1-41.
- [Thomas 2022] Thomas, L., Mifdaoui, A. and Le Boudec, J.Y., 2022. Worst-Case Delay Bounds in Time-Sensitive Networks With Packet Replication and Elimination. *IEEE/ACM Transactions on Networking*.
- [Vojnovic 2003] Vojnovic, M. and Le Boudec, J.Y., 2003. Bounds for Independent Regulated Inputs Multiplexed in a Service Curve Network Element. *IEEE Transactions on Communications*, 51, pp.735-740.
- [Zhang 1993] Zhang, H. and Ferrari, D., 1993, March. Rate-controlled static-priority queueing. In *IEEE INFOCOM'93 The Conference on Computer Communications*, Proceedings (pp. 227-236). IEEE.
- [Zhao 2018] Zhao, L., Pop, P., Zheng, Z. and Li, Q., 2018, April. Timing analysis of AVB traffic in TSN networks using network calculus. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (pp. 25-36). IEEE.
- [Zhou 2020] B. Zhou, I. Howenstine, S. Limprapaipong and L. Cheng, "A survey on network calculus tools for network infrastructure in real-time systems," *IEEE Access*, vol. 8, pp. 223588–223605, 2020. doi: 10.1109/ACCESS. 2020.3043600.
- [Nikolaus 2019] Nikolaus, P., Schmitt, J. and Schütze, M., 2019. h-Mitigators: Improving your stochastic network calculus output bounds. *Computer Communications*, 144, pp.188-197.