# Time Sensitive Networks and Network Calculus

Jean-Yves Le Boudec

Dagstuhl Seminar

April 2024

Slides available at https://leboudec.github.io/leboudec/resources/lebSlides.html

Title: Time Sensitive Networks and Network Calculus

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 describe recent results that can be used to analyze time sensitive networks with components such as packet ordering and duplicate removal functions, schedulers, regulators, dampers. We explain why clock non-idealities matter and how to take them into account.

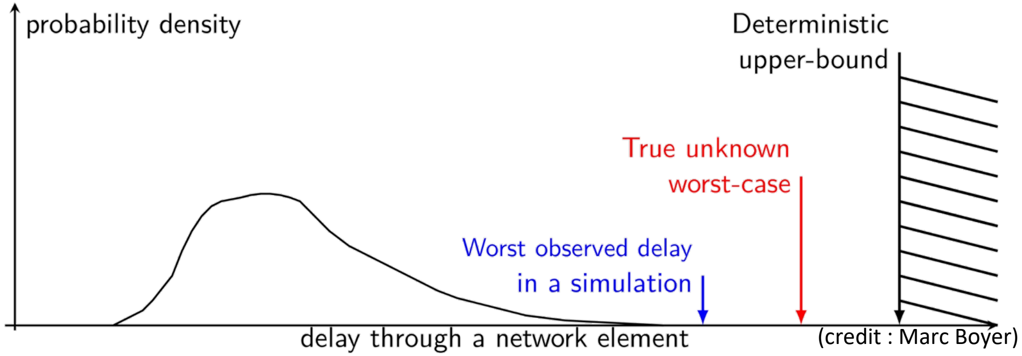
Joint work with: Anne Bouillard, Marc Boyer, Damien Guidolin-Pina, Jörn Migge, Ahlem Mifadoui, Ehsan Mohammadpour, Stephan Plasssart, Eleni Stai, Hossein Tabatabaee and Ludovic Thomas.

# Contents

1. Time sensitive Networks
2. Improved Delay Bounds
3. Clock Non-Idealities
4. Packet Re-ordering
5. Packet Replication and Elimination

# 1. Time Sensitive Networks (IEEE TSN, IETF DetNet)

Deterministic service: upper bounds on end-to-end delay and delay jitter, buffer sizing to achieve zero congestion loss. Proven bounds are required, simulation is not sufficient.

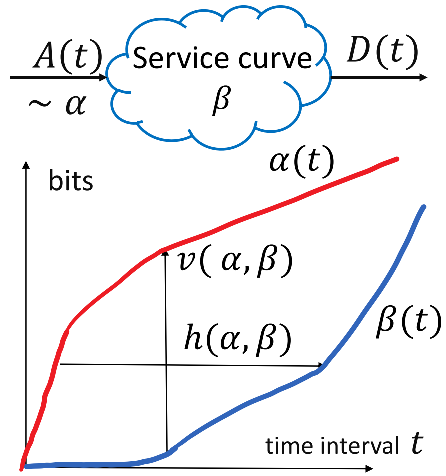



Every flow is constrained at source.

Network nodes offer a guaranteed service.

- synchronous: e.g Time Triggered CAN bus: (not our focus today)
- asynchronous: e.g. switch/router network: Flows are assigned to classes; at every node, traffic of a given class is FIFO; a scheduler shares bandwidth and buffer between classes (e.g. CBS, Burst Limiting Shaper, WRR, DRR, Static Priority, etc)

# 2. Nodal Analysis: Delay Bound



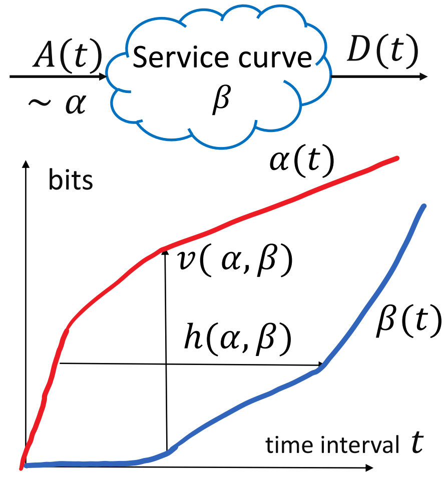

Assume

- Node is FIFO for this flow aggregate
- Service guarantee with service curve $\beta$
- Flow aggregate is constrained by arrival curve $\alpha$ at input (total number of bits in any $t$ seconds is $\leq \alpha(t)$)

Delay bound ?

- Delay $\leq h(\alpha, \beta)$
- Jitter bound = $h(\alpha, \beta)$ − delay lower bound

# Improved Delay Bound




Assume in addition

- Data is packetized at input
- Output serves entire packets, with rate $c$

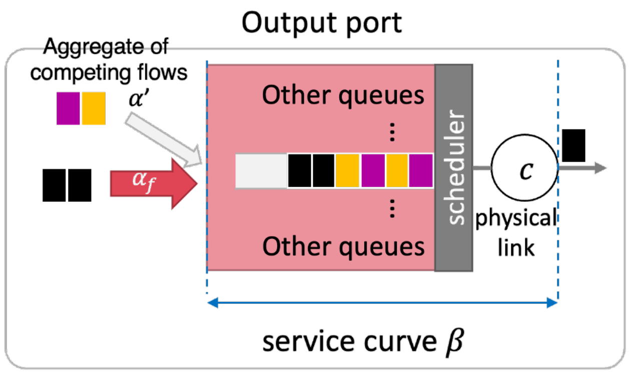
$\alpha^+$ = right-limit of $\alpha$

For a packet of size $\ell$, $\boxed{\text{delay} \leq h(\alpha^+ - \ell, \beta) + \frac{\ell}{c}}$

If $\beta$ is $c$-Lipschitz (slope$\leq c$), for a flow within the aggregate that has minimum packet size $L^{\min}$,

$$\boxed{\text{delay} \leq h\left(\alpha - L^{\min}, \beta\right) + \frac{L^{\min}}{c}}$$

[Mohammadpour 2023]




Example: $\beta(t) = R[t-T]^+$, $\alpha_f(t) = r_f t + b_f$, $\alpha'(t) = r't + b'$, $\alpha = \alpha_f + \alpha'$,

$r_f + r' \leq R \leq c$: delay bound for flow $f$ is $T + \frac{b_f + b'}{R} - L_f^{\min}\left(\frac{1}{R} - \frac{1}{c}\right)$ improvement

# How is this improvement obtained ?

1. Use arrival curve characterization for a point process model:

saying that the flow has arrival curve $\alpha$ is equivalent to

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

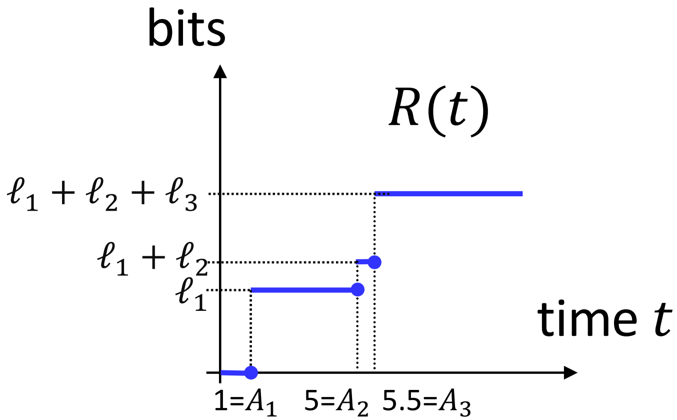
which is also equivalent to

$$\ell_m + \cdots + \ell_n \leq \alpha^{+}(A_n - A_m), \forall m, n, 1 \leq m \leq n$$

where $\alpha^{\downarrow}$ is the lower-pseudo inverse of $\alpha$ and $\alpha^{+}$is the right-limit of $\alpha$

2. An upper bound on *queuing delay* for a packet of length $\ell$ is $h(\alpha^{+} - \ell, \beta)$

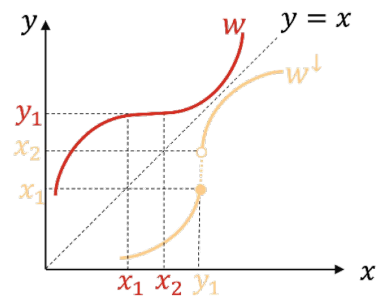
[Mohammadpour 2023]




Point process (left continuous)

$$R(t) = \sum_{n \geq 1} \ell_n 1_{\{A_n < t\}}$$

Lower pseudo-inverse



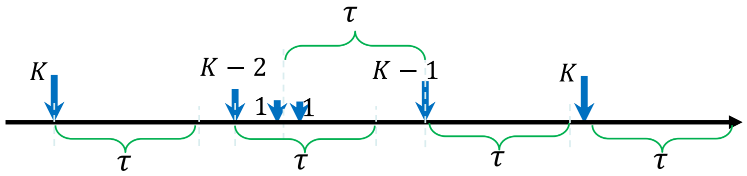
# TSN Source Constraint $(\tau, K, L)$

During an interval of duration $\tau$, up to $K$ packets of size $\leq L$ can be sent.

Is this equivalent to an arrival curve constraint ?

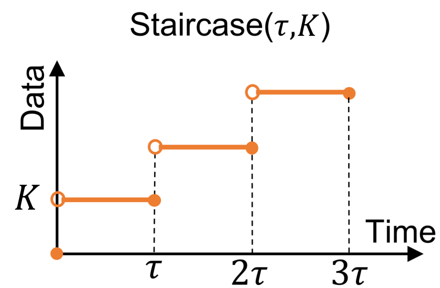
# TSN Source Constraint $(\tau, K, L)$: sliding window interpretation

During *any* interval of duration $\tau$, up to $K$ packets of size $\leq L$ can be sent.




$\Leftrightarrow$ packet-level arrival curve: at most $\alpha_P(t)$ packets in any interval of duration $t$, with

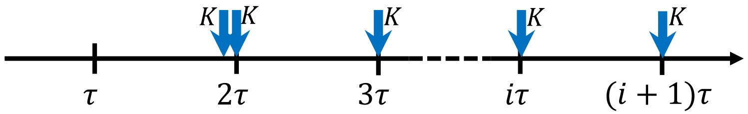
$$\alpha_P(t) = K \left\lceil \frac{t}{\tau} \right\rceil$$




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

# TSN Source Constraint $(\tau, K, L)$ : fixed window interpretation

During *repeating* intervals of duration $\tau$, up to $K$ packets of size $\leq L$ can be sent.

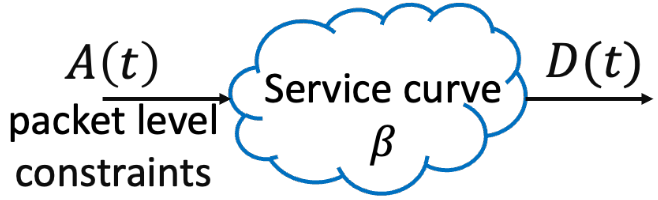



$\Longrightarrow$ packet-level arrival curve : $\alpha_P(t) = K \left\lceil \frac{t}{\tau} \right\rceil + K$

Not an equivalence.

However: for every $\varepsilon > 0$, there is a greedy source that has packet-level arrival curve $\alpha_P^{\varepsilon}(t) = K \left\lceil \frac{t-\varepsilon}{\tau} \right\rceil + K$
and satisfies the TSN constraint *fixed-window* $(\tau, K, L)$
[Mohammadpour 2023]

# Delay Bound with Packet-level Arrival Constraints




Assume

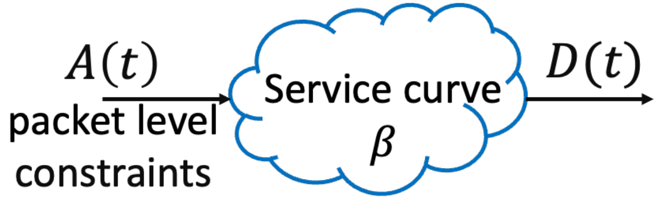
- Node is FIFO for this flow aggregate
- Service guarantee with service curve $\beta$
- Flow $i$ within aggregate is constrained by packet level arrival curve $\alpha_{P,i}$ at input
- Data is packetized at input, line rate $c$, $\beta$ is $c$-Lipschitz

Delay bound ?

- Derive bit-level arrival curve for aggregate $\alpha(t) = \sum_i L_i^{\max} \alpha_{P,i}\,(t)$
- For flow $f$, delay $\leq h\left(\alpha - L_f^{\min}, \beta\right) + \frac{L_f^{\min}}{c}$

# Improved Delay Bound with Packet-level Arrival Constraints

A better bound is [Mohammadpour 2023]:




For flow $f$, $\boxed{\text{delay} \leq h\left(\alpha - L_f^{\max}, \beta\right) + \frac{L_f^{\max}}{c}}$

- Every flow $i$ is constrained by packet level arrival curve $\alpha_{P,i}$ at input
- $\alpha(t) = \sum_i L_i^{\max} \alpha_{P,i}(t)$
- Data is packetized at input, line rate $c$, $\beta$ is $c$-Lipschitz

Example: 2 flows, $\beta(t) = R[t - T]^+$, $\alpha_{P,i}(t) = K_i \left\lceil \frac{t}{\tau_i} \right\rceil, i = 1,2;$

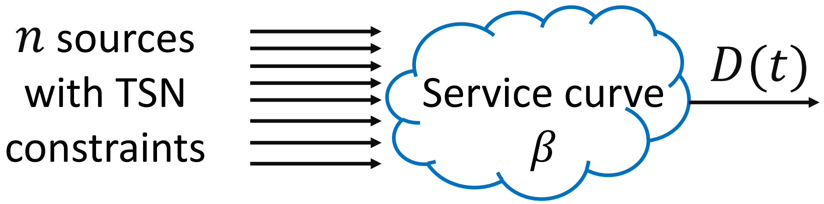
$$c \geq R \geq \frac{K_1 L_1^{\max}}{\tau_1} + \frac{K_2 L_2^{\max}}{\tau_2}$$

Delay bound for flow 1: $\underbrace{T + \frac{K_1 L_1^{\max} + K_2 L_2^{\max}}{R}}_{\text{standard NC bound}} - L_1^{\max}\left(\frac{1}{R} - \frac{1}{c}\right)$

Compare to bound derived from bit-level arrival curve: $T + \frac{K_1 L_1^{\max} + K_2 L_2^{\max}}{R} - L_1^{\min}\left(\frac{1}{R} - \frac{1}{c}\right)$

Packet-level arrival curve is more constraining than bit-level ! Better bounds can be found by exploiting packet-level constraints !

# Delay Bounds for TSN Sources




- For a flow $i$ that is constrained by TSN sliding window $(\tau_i, K_i, L_i)$ let $\alpha_{P,i}(t) = K_i \left\lceil \frac{t}{\tau_i} \right\rceil$
- For a flow $i$ that is constrained by TSN fixed window $(\tau_i, K_i, L_i)$ let $\alpha_{P,i}(t) = K_i \left\lceil \frac{t}{\tau_i} \right\rceil + K_i$
- Let $\alpha(t) = \sum_{i=1}^{n} L_i^{\max} \alpha_{P,i}\,(t)$
- Delay bound for a tagged flow $i$ is $h(\alpha - L_i^{\max}, \beta) + \frac{L_i^{\max}}{c}$

This delay bound is tight [Mohammadpour 2023].

Network Calculus delay bound can be slightly improved

The analysis combines min-plus (service curve) and max-plus (point process)

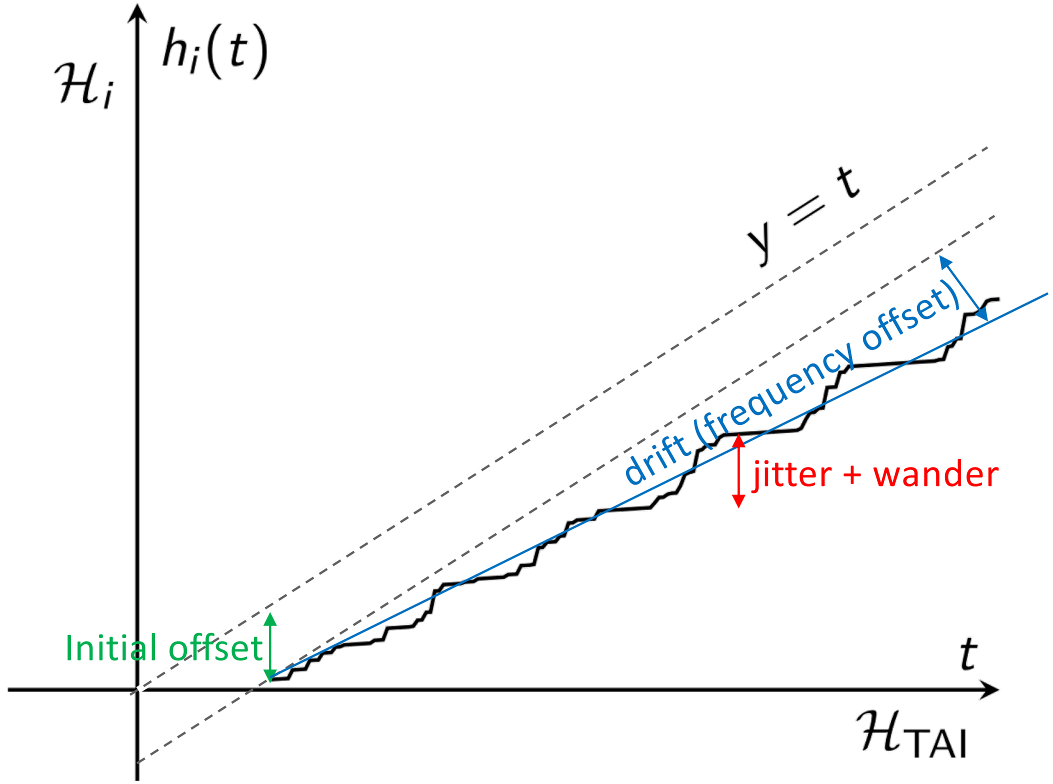
TSN T-spec is at packet level, regulators (ATS) are at bit level $\Longrightarrow$ packet-level theory ?

# 3. Clock Non-Idealities

Standard 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 unbounded; measurement of time interval on same system:
  error is bounded by clock drift, jitter and wander.

[ITU-T 1996]




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 $\rightarrow \mathrm{d}$ 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 [Thomas 2020]

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

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

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

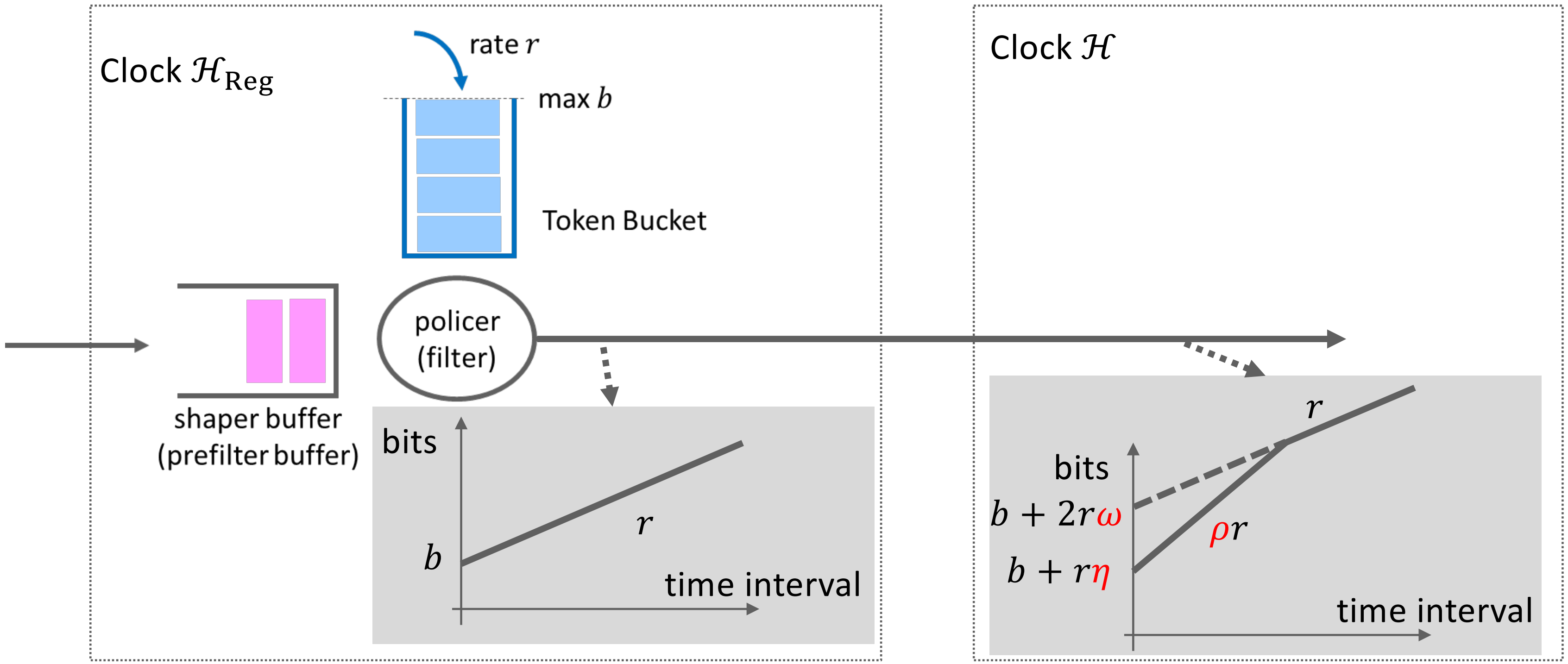
$\rho =$ clock-stability bound
=1.0001 (e.g. in TSN)

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

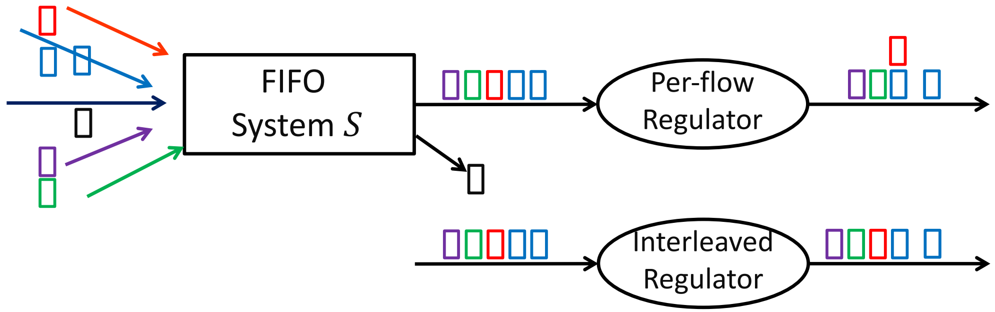
# Change of Clock: Arrival Curves

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 $\alpha^{\mathcal{H}}(t) = \min(\rho rt + b + r\eta, rt + b + 2r\omega)$ [Thomas 2020]



# Regulators (aka Shapers, ATS)




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

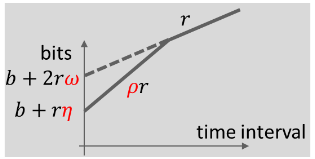
Non work-conserving.

Per flow regulator: one state + one queue per flow.
Interleaved regulator: one state per flow + one global queue, packets not at head of queue wait behind [Specht 2016].

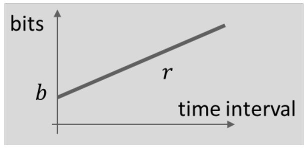
Regulators avoid burstiness cascade, do not increase worst-case end-to-end delay (in principle).

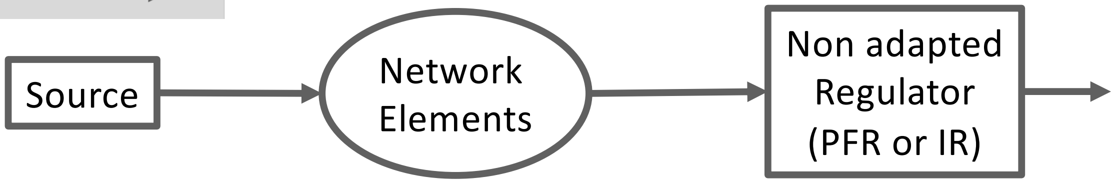
# Consequences for Regulators

- **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.




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



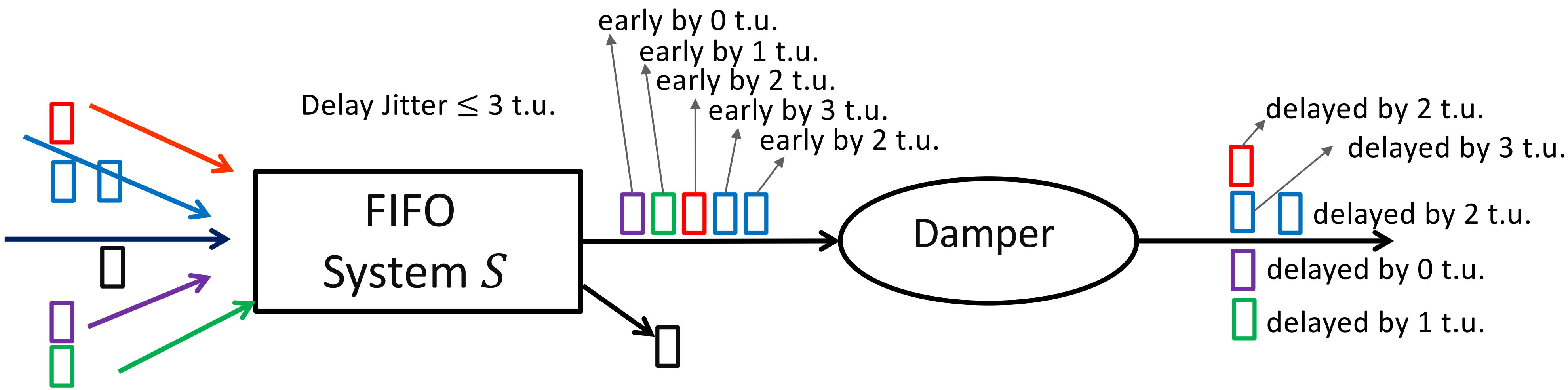




Flow constrained by $\alpha(t) = rt + b$ in local clock

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

Regulators must use safety margins (be *adapted)* using e.g. rate and burst cascade or ADAM [Thomas 2020]

# Dampers



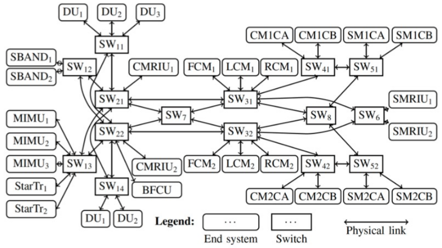

Damper delays a packet by "earliness" read from packet header.

Removes most of jitter, with some residual jitter dependent on tolerance, not on traffic $\Longrightarrow$ also removes burstiness cascade.

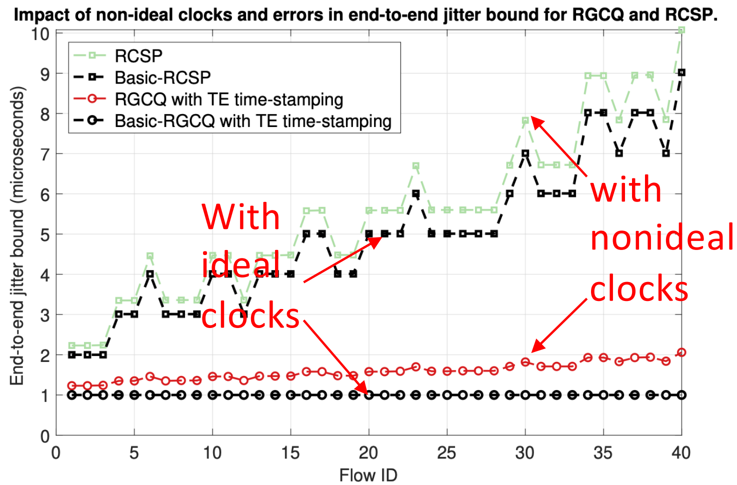
Non work-conserving. Like a per-flow regulator, does not exist in isolation, is combined with queue at next hop.

Unlike regulator, is stateless.
[Cruz 1998] RCSP [Zhang 1993], RGCQ [Shoushou 2020], ATS with Jitter Control [Grigorjew 2020].

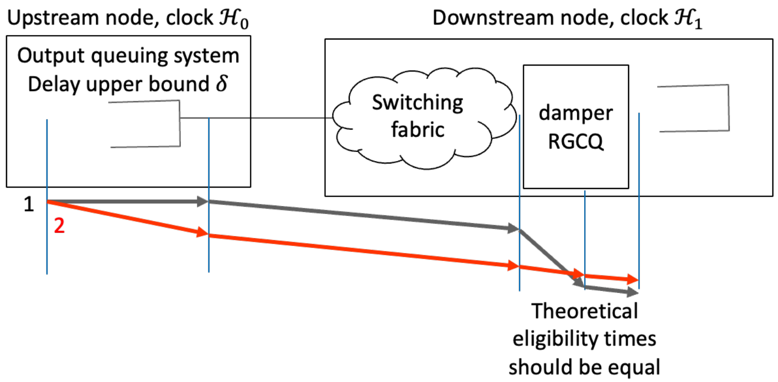
# Consequences for Dampers



Residual jitter is somehow affected by clock inaccuracies




Timing inaccuracies may lead to mis-ordering

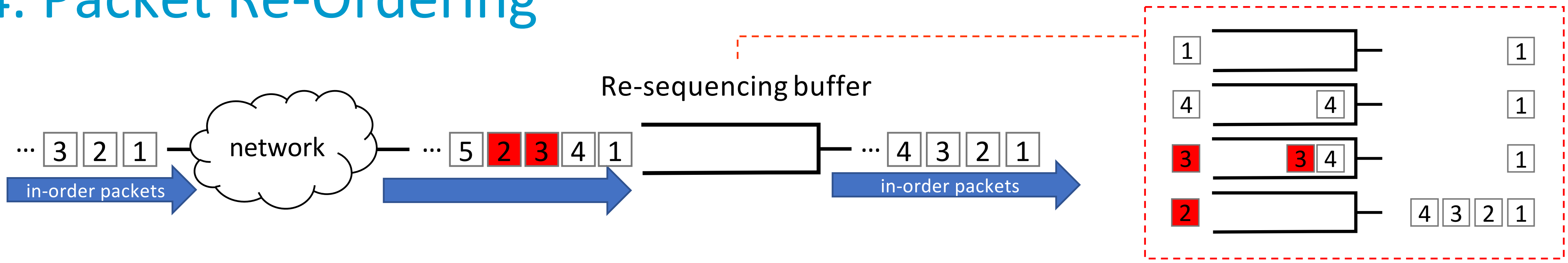



Two consecutive packets should be released by damper at ca. the same time, timing inaccuracies may lead to inversion of order

⇒ Some dampers enforce per-flow packet order (e.g. FOPLEQ, ATS with Jitter Control [Grigorjew 2020]) - work properly only if all network elements are FIFO per flow

[Mohammadpour 2022]

# 4. Packet Re-Ordering




Time Sensitive Networks may cause packet mis-ordering due to e.g. parallel paths in switching fabrics, dampers, packet replication and elimination etc.

Re-sequencing buffer may be needed before delivery:

stores packets until in-sequence delivery or timeout

**Questions:** Buffer size ? Minimal timer value ? Effect on worst-case delay bounds ?

# Reordering late Time Offset (RTO)

[RFC 4737, Mohammadpour 2021]

Defined between two observation points, and for a flow of interest

First observation point defines the reference order of packets

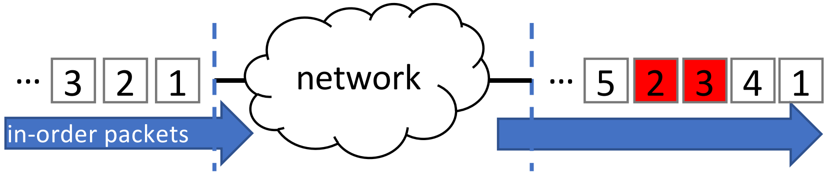
RTO = largest time by which a mis-ordered packet is late

$$\text{RTO} = \max_{n \mid E_n < +\infty} \lambda_n$$

$$\lambda_n = E_n - \min_{j \geq n, E_j \leq E_n} \{E_j\}$$

| | $E_n$ $(\mu s)$ | $\lambda_n$ $(\mu s)$ |
|---|---|---|
| 1 | 5 | 0 |
| 4 | 10 | 0 |
| 3 | 12 | 2 |
| 2 | 18 | 8 |
| 5 | 20 | 0 |

$\text{RTO} = 8\mu s$

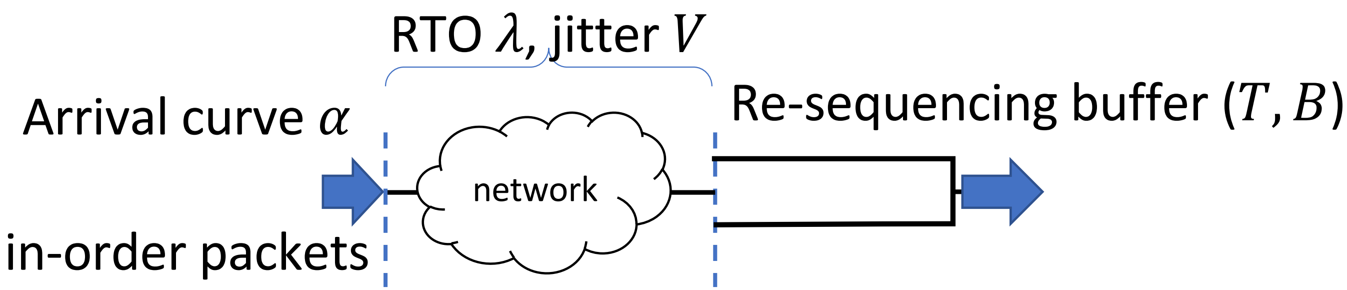



$A_j \leq A_n \iff j < n$ $\quad A_n$ $\quad E_n$ $\quad E_n = +\infty$, if packet $n$ is lost

arrival time of packet $n$ $\quad$ exit time of packet $n$

RTO $= 0$ means no re-ordering

# Resequencing Buffer Calculus




1. Re-sequencing buffer timeout $T$ minimum value is $T_{\min} = \lambda = \text{RTO}$
2. Required size of resequencing buffer is $B_{\min} = \alpha(V + T)$

[Mohammadpour 2021]

# Calculus of RTO (1)

For a system that may re-order packets and has known delay jitter $V$, the best RTO bound for a flow with arrival curve $\alpha$ is $\lambda = \left[V - \alpha^{\downarrow}(2L^{\min})\right]^{+}$.

**Example**: (token bucket) $\alpha(t) = rt + b$

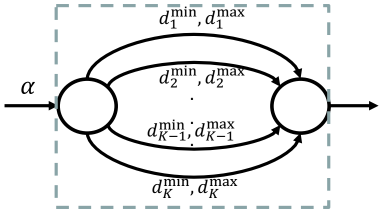
If $b < 2L^{\min}$ and $V \leq \frac{2L^{\min} - b}{r}$ then $\lambda = 0$ (no reorder)

If $b > 2L^{\min}$ then $\lambda = V$ (reordering is possible)

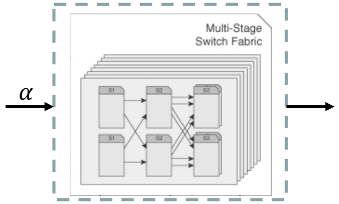
Other bounds exist for flow constraints at packet level. [Mohammadpour 2021]

All bounds in TAI (temps atomique international)

Examples of systems

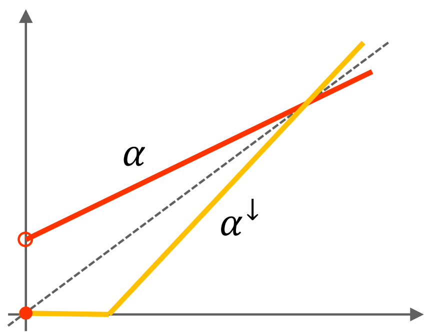



$V = \max_{i=1..K}\{d_i^{\max}\} - \min_{i=1..K}\{d_i^{\min}\}$

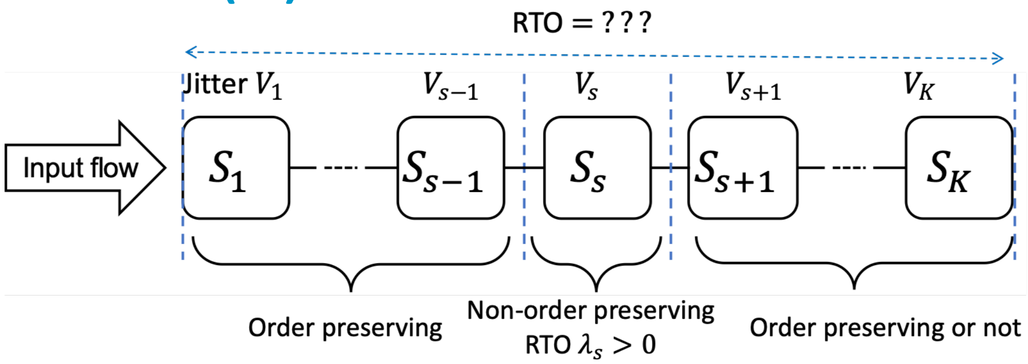



$V \in [1\mu s, 10\mu s]$

Lower pseudo-inverse

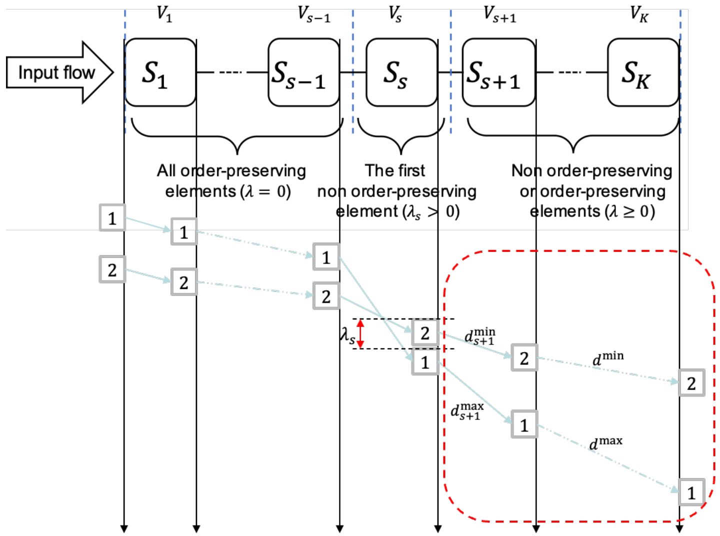


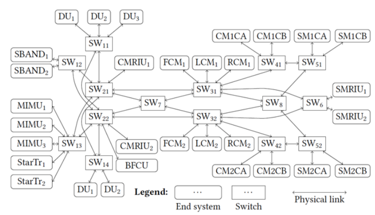
# Calculus of RTO (2): Concatenation

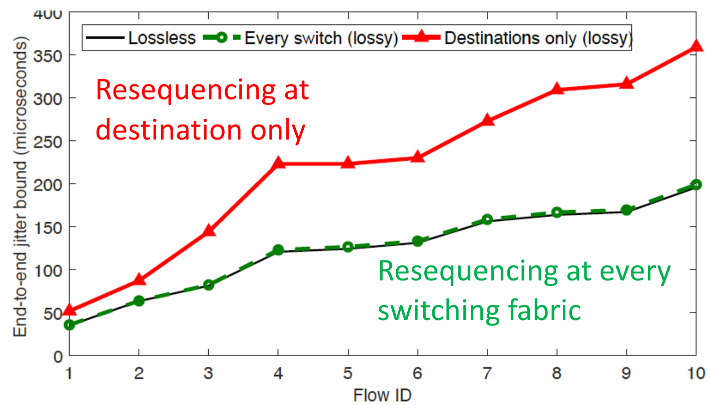



Best RTO bound for concatenation is $\lambda_s + \sum_{i=s+1}^{K} V_i$ [Mohammadpour 2021]

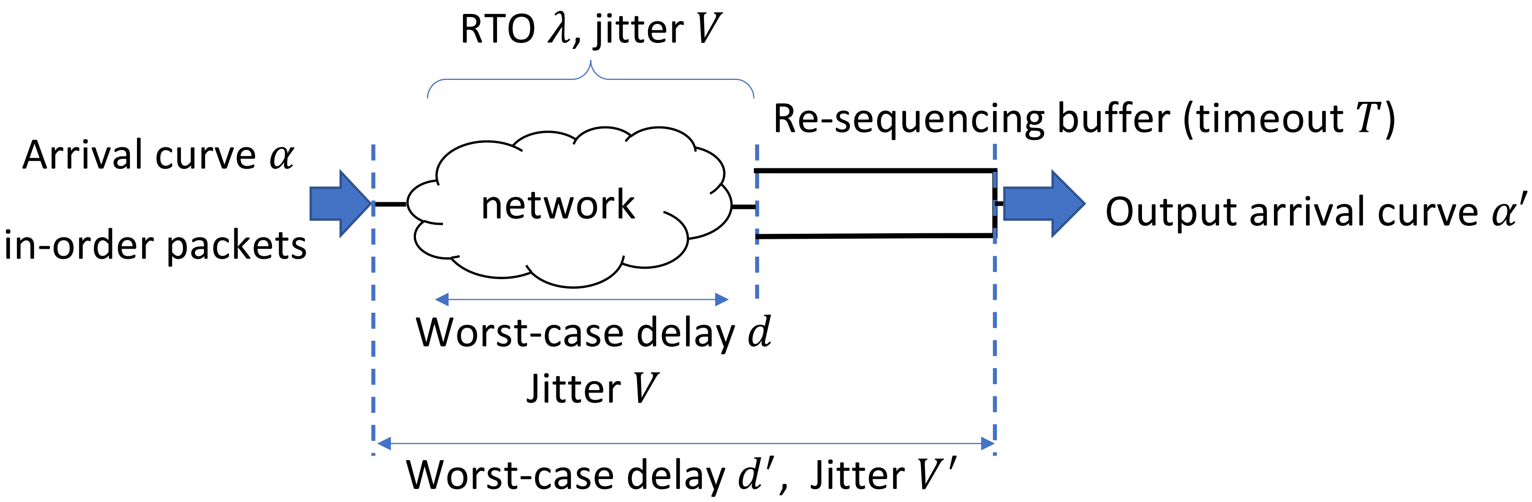
RTO amplification by downstream jitter





# Network Calculus with Re-sequencing Buffers




Lossless network:

$d' = d, V' = V$ and $\alpha'(t) = \alpha(t + V)$ (re-sequencing is for free)

Lossy network:

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

Packet reordering metric of interest is the RTO

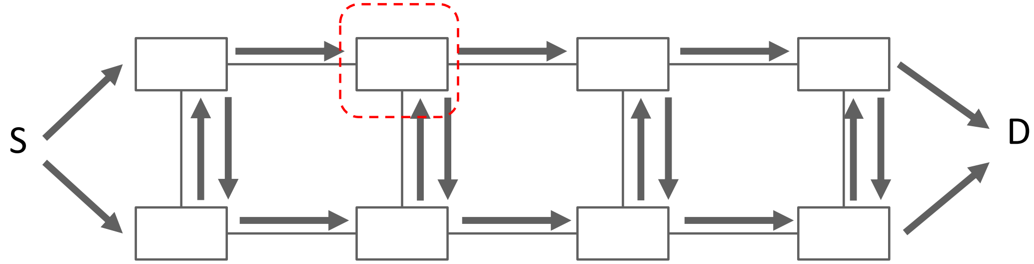
Resequencing is for free in lossless networks

In lossy networks, timeout value (hence RTO) affects delay, jitter and propagated burstiness

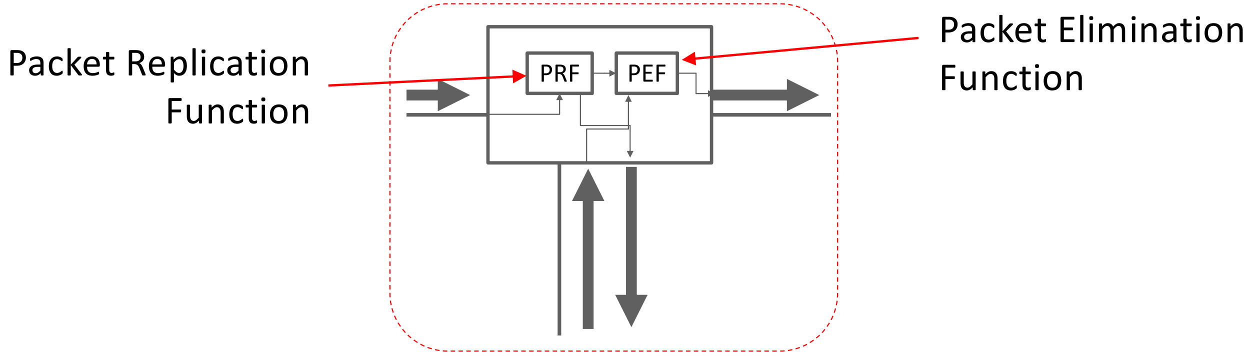
Resequencing-at-end only may cause large re-sequencing delay due to RTO amplification

# 5. Packet Replication and Elimination

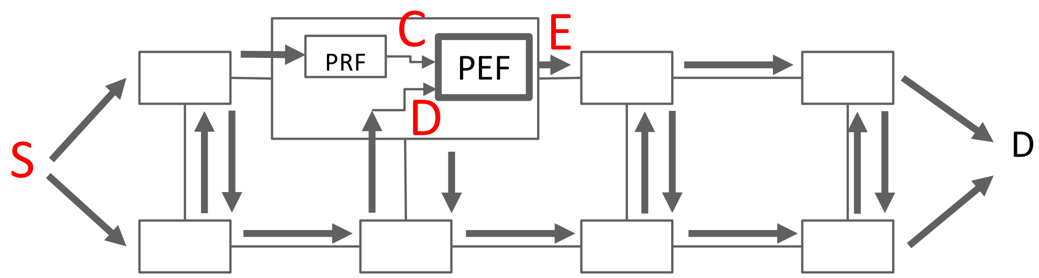
Deterministic networks guarantee 0 congestion loss, but other losses may occur (transient failures, reboots, transmission errors). This is mitigated by packet replication and duplicate elimination (FRER, PREOF).

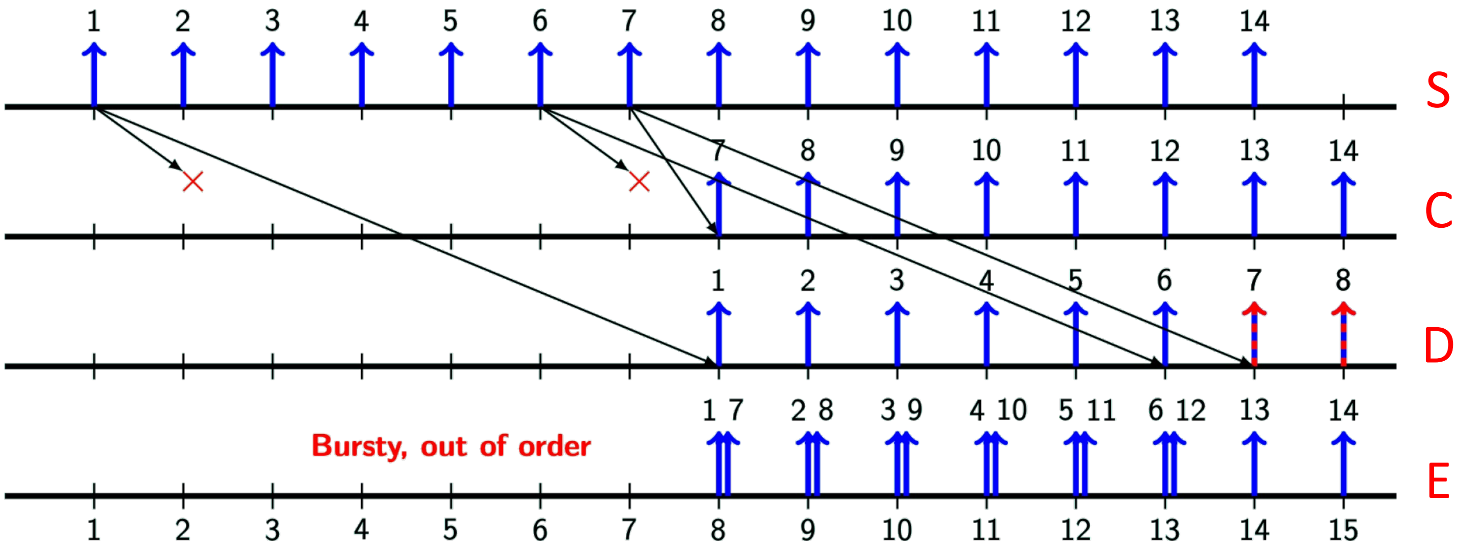



Any combination of failures that leaves at least one path up is masked (“0 msec repair”) [IEEE 802.1CB]



# Packet Elimination Function Causes Mis-Ordering + Increased Burstiness [Thomas 2022]




Source sends one packet every time unit. Packets 1 to 6 are lost between S and C due to transient failure of north-west node.

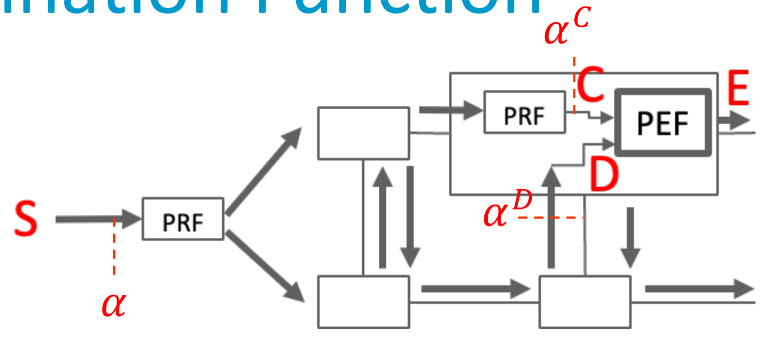
At exit of Packet Elimination Function:

- Up to two packets per time unit (more bursty)
- Mis-ordering

# Network Calculus Analysis of Packet Elimination Function

Arrival curve at output of PEF: $\alpha^* = \alpha^{\text{AGG}} \otimes \alpha^{\text{JIT}}$ where:

- $\alpha^{\text{AGG}} = \sum$ propagated arrival curves at input of PEF;
- $\alpha^{\text{JIT}}(t) = \alpha(t + D^{\max} - D^{\min})$ where $D^{\max}$ [resp. $D^{min}$] is an upper [resp. lower] bound on delay between common ancestor and input of PEF and $\alpha$ is arrival curve at output of common ancestor on any path.




$\alpha^{\text{AGG}} = \alpha^C + \alpha^D$

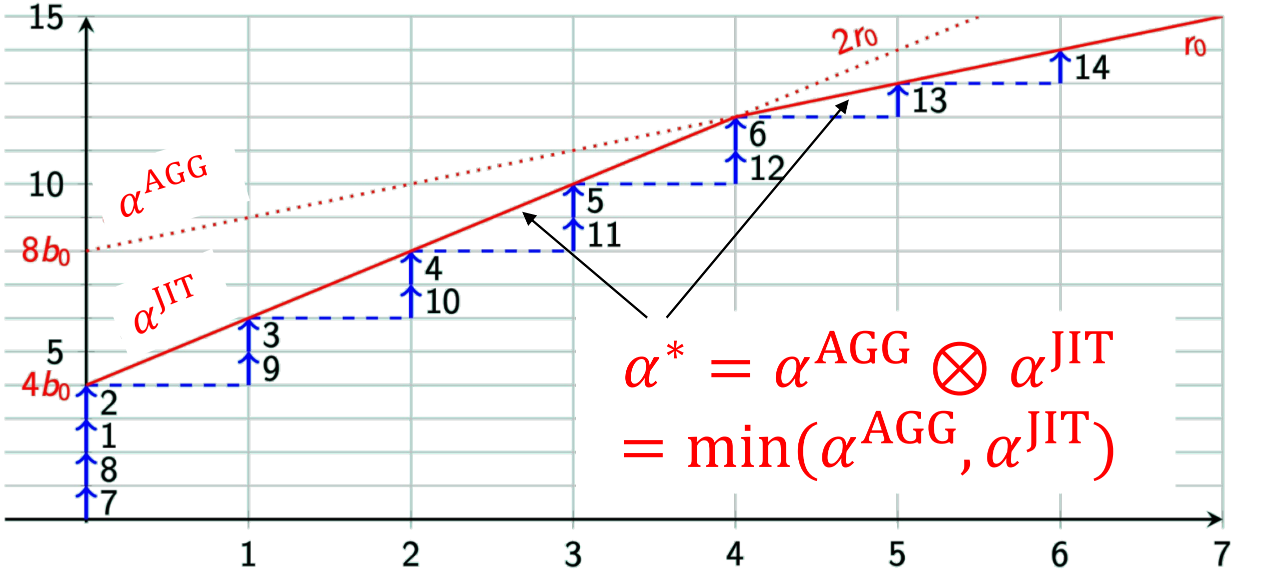
Delay from S to (C or D) ∈ $[D^{\min}, D^{\max}]$

Bound on RTO (amount of re-ordering)

$$\lambda = \left[D^{\max} - D^{\min} - \alpha^{\downarrow}(2L^{\min})\right]^+$$

Network Analysis implemented in extension of TFA (xTFA).

[Thomas 2022]



In-network packet replication and elimination is present in time-sensitive networks

Packet elimination negatively affects the deterministic delay bounds and can be taken into account

# Other Fascinating Topics...

Network analysis with TFA, PLP, PMOO...

Service curves for bandwidth sharing schedulers (WRR, DRR, ...)

CQF

Quasi-deterministic bounds

# Tools

- The DiscoDNC is an academic Java implementation of the network calculus framework.[10]
- The RTC Toolbox is an academic Java/MATLAB implementation of the Real-Time calculus framework, a theory quasi equivalent to network calculus.[4][11]
- The CyNC[12] 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.[13]
- The WOPANets is an academic tool combining network calculus based analysis and optimization analysis.[14]
- The DelayLyzer is an industrial tool designed to compute bounds for Profinet networks.[15]
- DEBORAH is an academic tool devoted to FIFO networks.[16]
- NetCalBounds is an academic tool devoted to blind & FIFO tandem networks.[17][18]
- 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.[19]
- The Siemens Network Planner (SINETPLAN) uses network calculus (among other methods) to help the design of a PROFINET network. [20]
- experimental modular TFA (xTFA) is a Python code, support of the PhD thesis of Ludovic Thomas[21]
- Panco is a Python code that computes network calculus bounds with linear programming methods.
- Saihu is a Python interface that integrates three worst-case network analysis tools: xTFA, DiscoDNC, and Panco.
- CCAC is an SMT-solver based tool to verify the performance properties of congestion control algorithms (CCAs) using a network-calculus-like model

copied on 2024 March 20 from https://en.wikipedia.org/wiki/Network_calculus

# 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.
- [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.
- [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 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*.
- [Mohammadpour 2023] Mohammadpour, E., Stai, E. and Le Boudec, J.Y., 2023. Improved Network-Calculus Nodal Delay-Bounds in Time-Sensitive Networks. *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://orbilu.uni.lu/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 methodolgies 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.
- [Zhang 2022] Zhang, Y., Jiang, Y. and Fu, S., 2022. Service Modeling and Delay Analysis of Packet Delivery over a Wireless Link. arXiv:2207.01730.
- [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.