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

EPFL

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

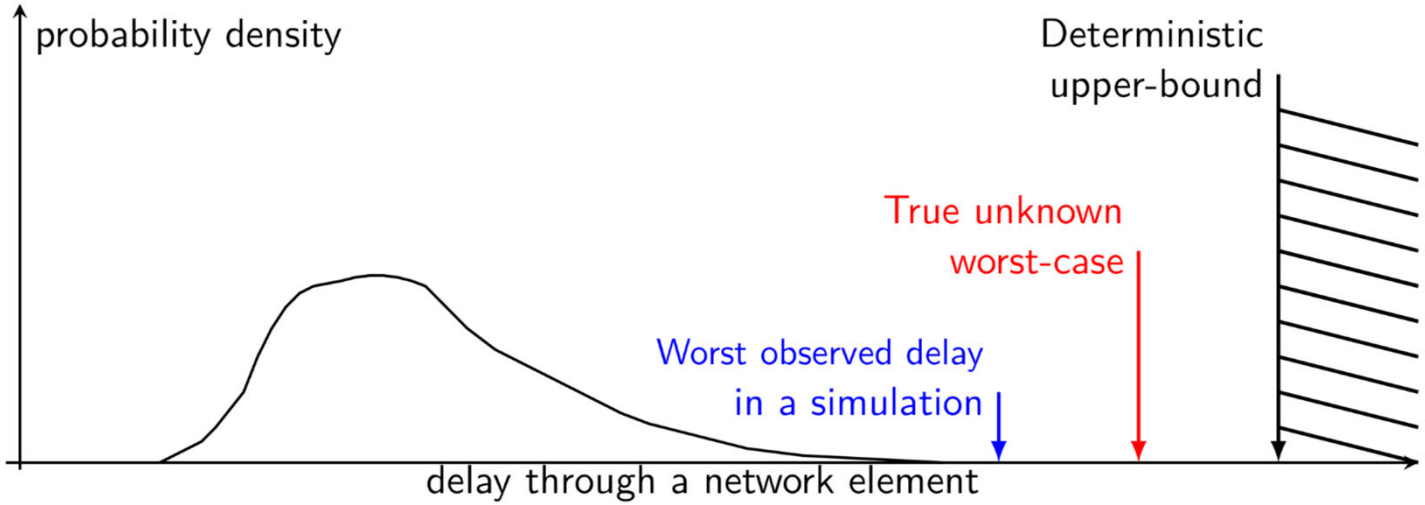
1. Time Sensitive Networks (IEEE TSN, IETF DetNet)

Time Sensitive networks = deterministic service:

upper bounds on end-to-end delay – not average
upper bound on delay jitter (= worst case – best case delay)
buffer sizing to achieve zero congestion loss.

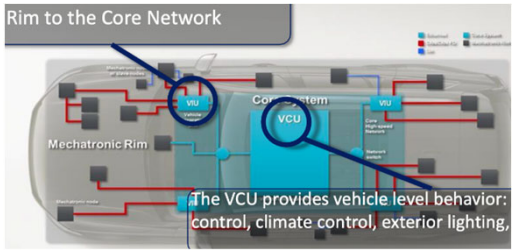
Congestion control with feedback is not an option here.

Proven bounds are required, simulation is not a solution.

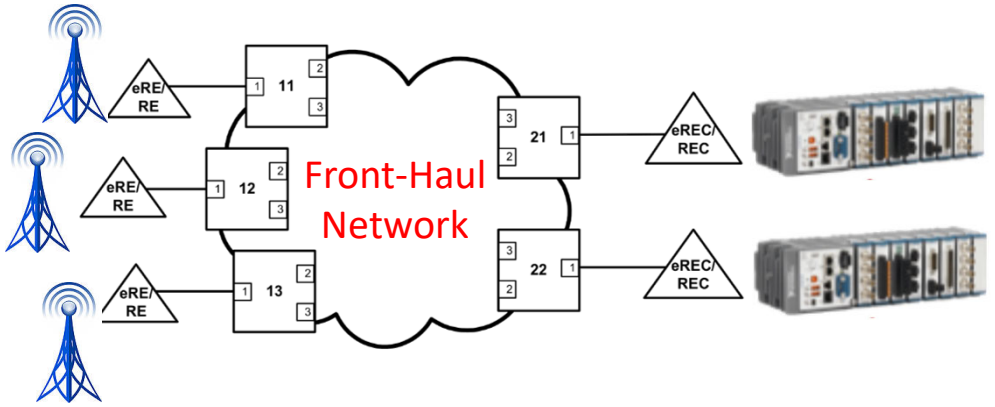


Time Sensitive Networks: Use Cases

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



From [Navet et al,2020]



Radio Equipment

Standardization:

MAC-layer networks: IEEE TSN (Time Sensitive Networking)
IP and MPLS networks: IETF Detnet (Deterministic Networking)

How can a Network Offer a Deterministic Service ?

1. Every flow is constrained at source

   e.g. source is periodic

   e.g. source is limited by a token bucket filter with rate r and burstiness b

   → number of bits sent over any interval of any duration t is $\leq rt+b$

   (*arrival curve* constraint) (T-SPEC)

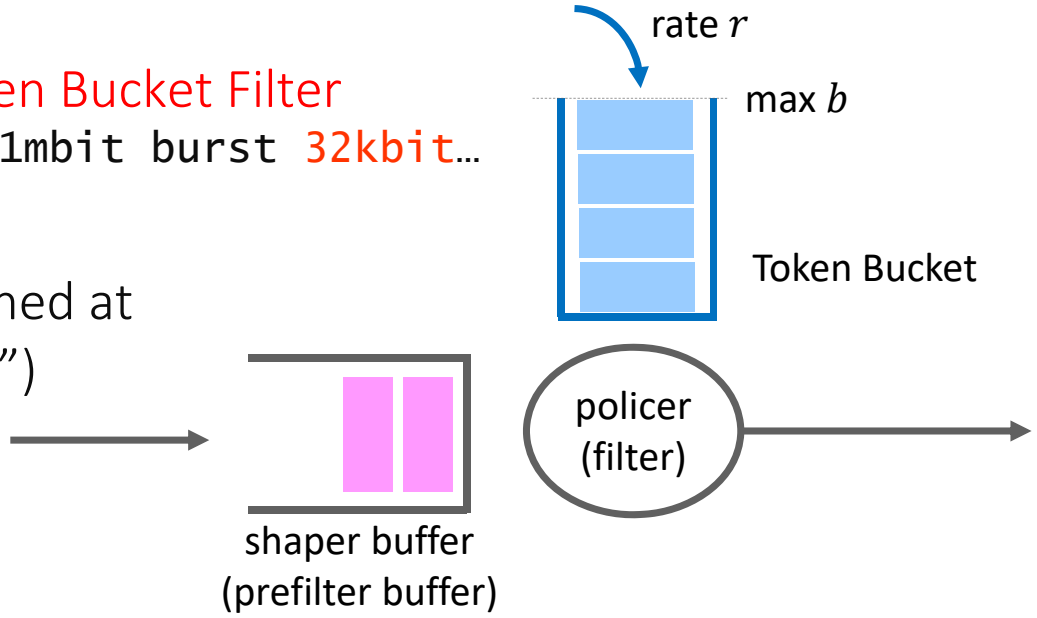
Token Bucket Filter (r, b)

Linux implements flow shaping by means of Token Bucket Filter

```
tc qdisc add dev eth0 root tbf rate 1mbit burst 32kbit…
```

Imagine a token bucket, spontaneously replenished at rate r up to some maximum b (called the "burst")

In order to be released, a packet must consume same amount of tokens as its size.

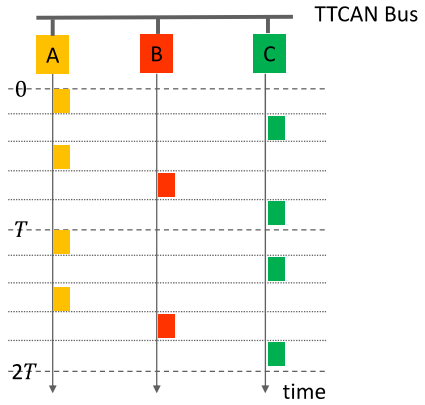


If there are not enough tokens, packet must wait. As soon as there are enough tokens, packet is released.

This forces the output such that the number of bits sent over any interval of any duration t is $\leq rt+b$ (arrival curve constraint).

How can a Network Offer a Deterministic Service ?

1. Every flow is constrained at source



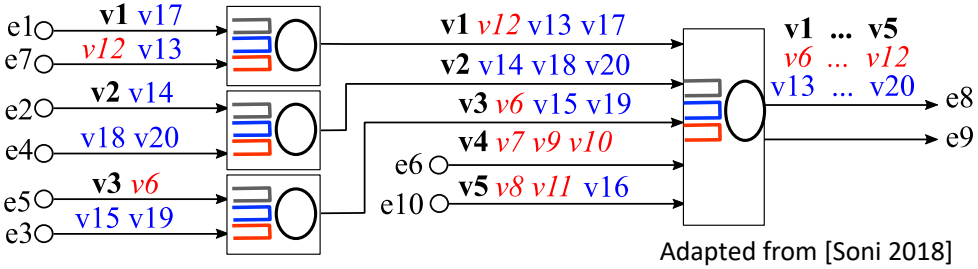
2. The network nodes offer a guaranteed service to flows or classes of flows

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

   asynchronous: e.g. switch/router network

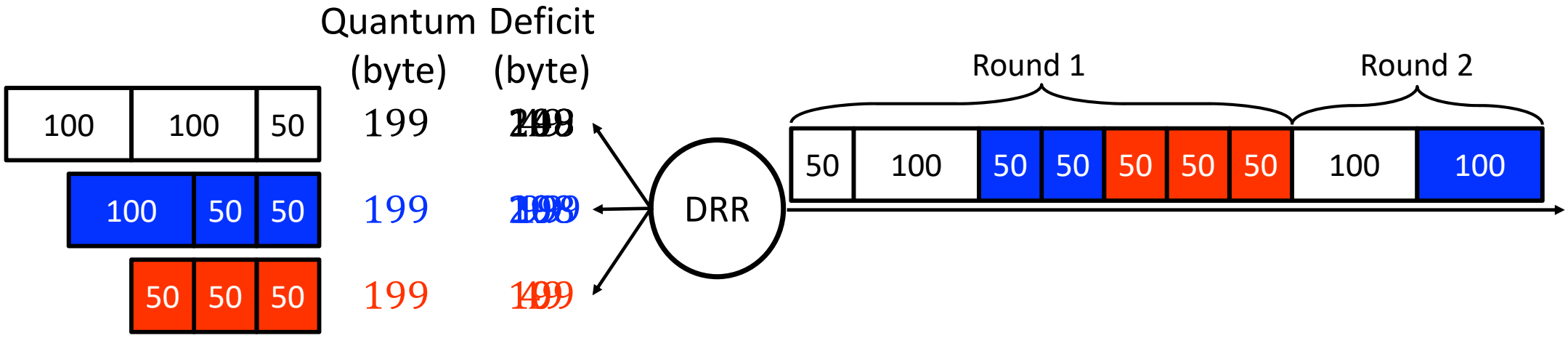
   a) Flows are assigned to a small number of classes with different quality of service requirements

   b) At every node, traffic of a given class is FIFO; a scheduler shares bandwidth and buffer between classes



Adapted from [Soni 2018]

Example of Scheduler: Deficit Round Robin (DRR) [Shreedhar 1996]



Implemented in Linux class based queuing **tc qdisc ... add drr [quantum** bytes]

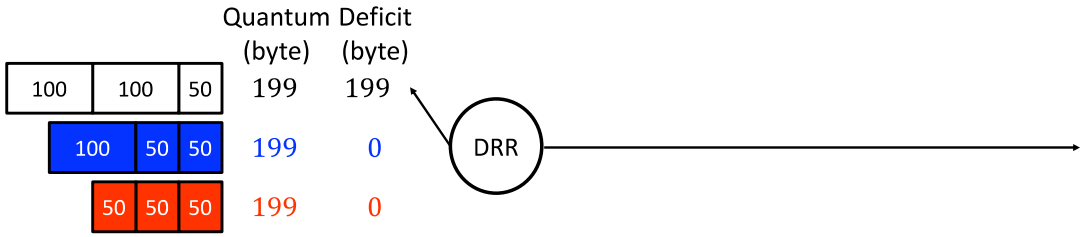
Operation: Each queue (= each class) is given a quantum.

An infinite loop of rounds visits queues.

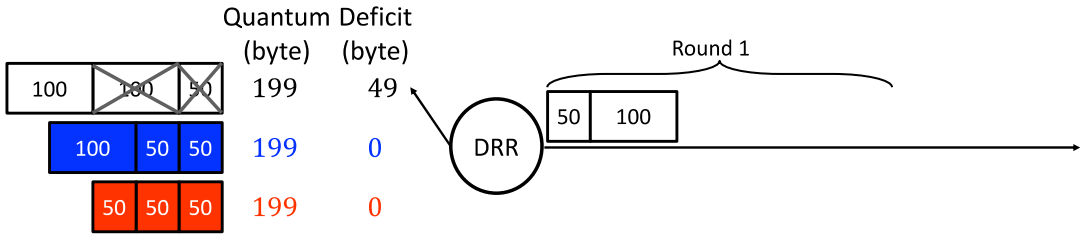
When a queue is visited its deficit is increased by the quantum.

Service for this queue stops if 1) deficit is smaller than head-of-line packet or 2) queue becomes empty (in which case deficit is reset).

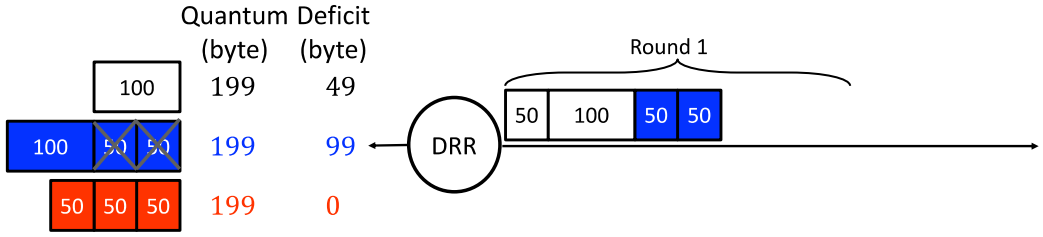
⇒ ≈ Bandwidth is allocated to every class in proportion of the quantum.



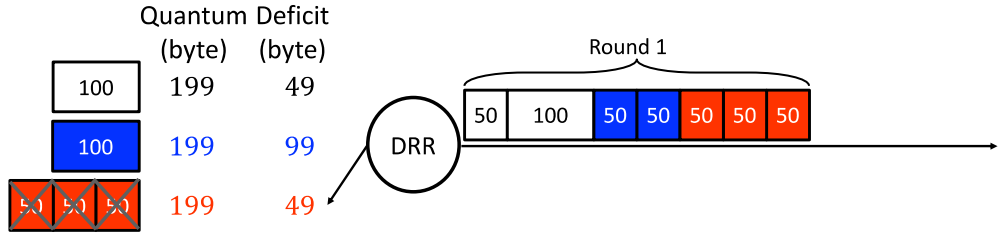
1. Black queue is visited, deficit is set to quantum value.



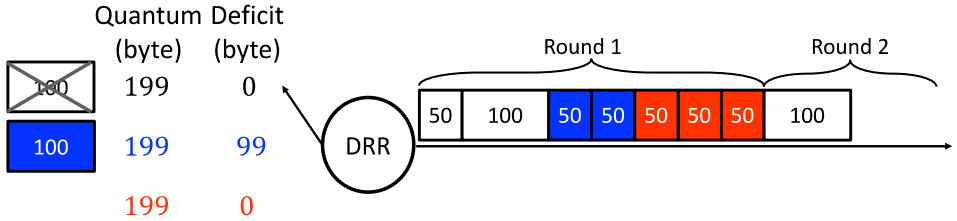
2. At end of visit to 1st queue.



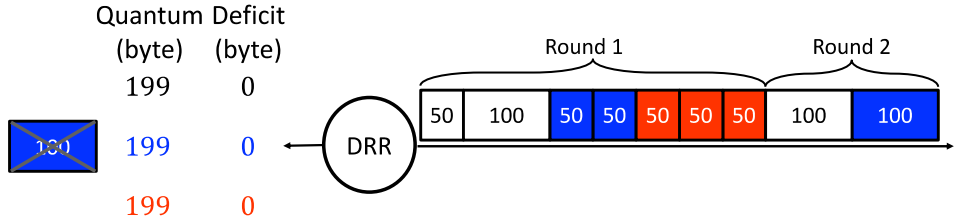
3. At end of visit to 2nd queue.



4. At end of visit to 3rd queue.



5. At end of visit to 1st queue, second round. Deficit is reset because queue is empty.



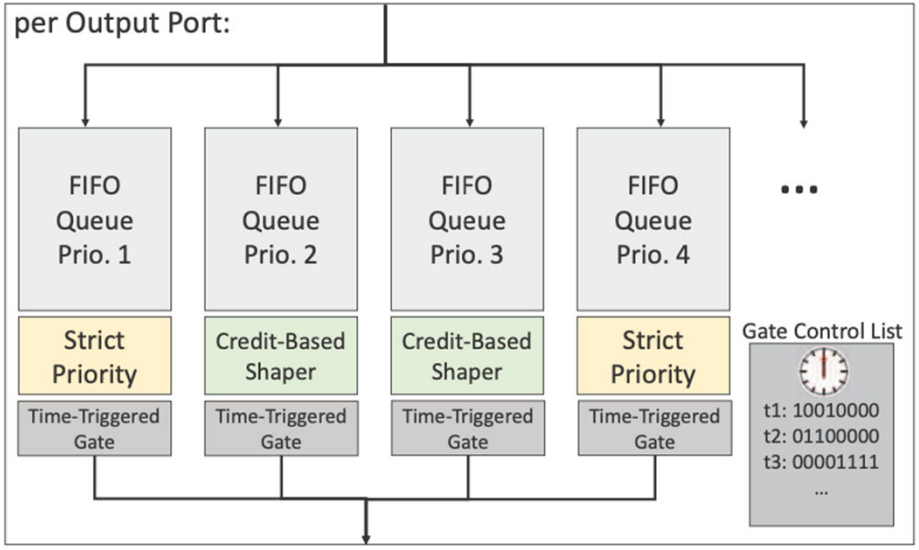
6. At end of visit to 2nd queue, second round.

Other Schedulers

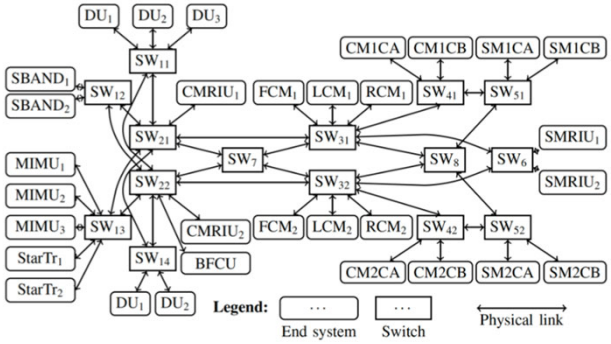
- Weighted Fair Queuing and all variants of Generalized Processor Sharing (such as DRR)
- Audio Visual Bridging (AVB) / Credit Based Shaper (CBS)
- Burst Limiting Shaper
- Time Aware Shaper
- Static Priority

Etc.

They can be combined.



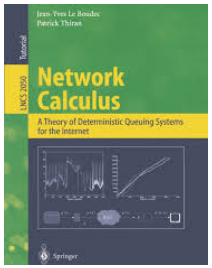
Typical IEEE TSN scheduler. From [Maile 2020]

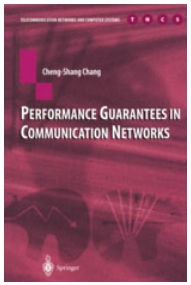


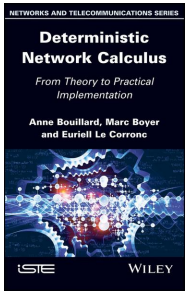
From [Zhao 2018]

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

2. Network Calculus: Single Node Analysis







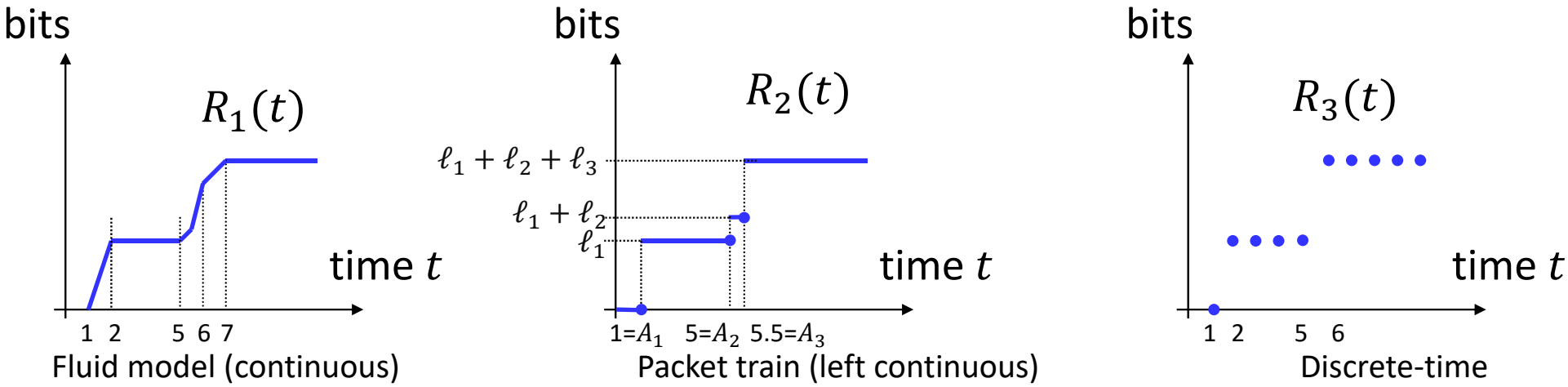
Finds bounds on delay, jitter and backlog that can be formally proven.

R Cruz, CS Chang, JY Le Boudec, P Thiran, …

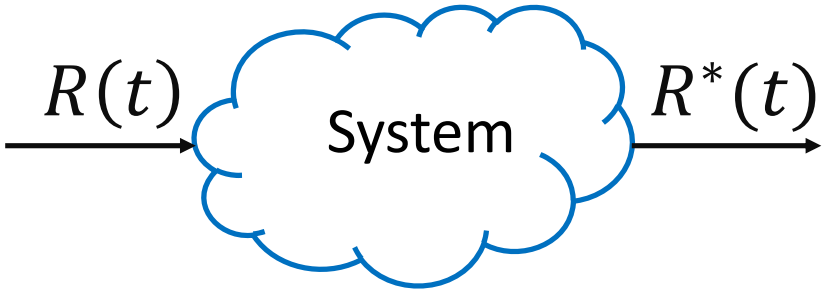
Stochastic extensions exist (not discussed here)

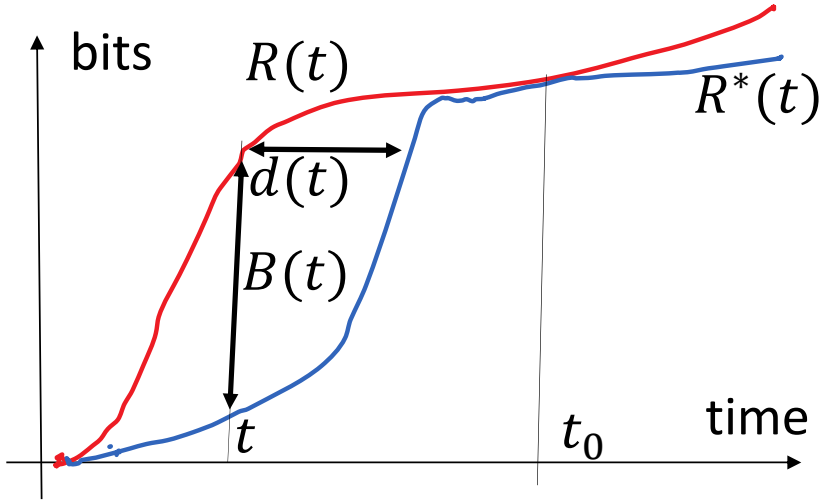
Representation of Data Flow

Cumulative flow: $R(t)$, non-decreasing with $R(0) = 0$



Delay and Backlog





Backlog at time t = $R(t) - R^*(t)$

If System preserves order for this flow: Delay $\leq h(R, R^*)$
with $h(R, R^*) = \sup_t d(t)$
and $d(t) = \inf\{d \text{ s.t. } R(t) \leq R^*(t+d)\}$
(horizontal deviation)

Arrival Curve

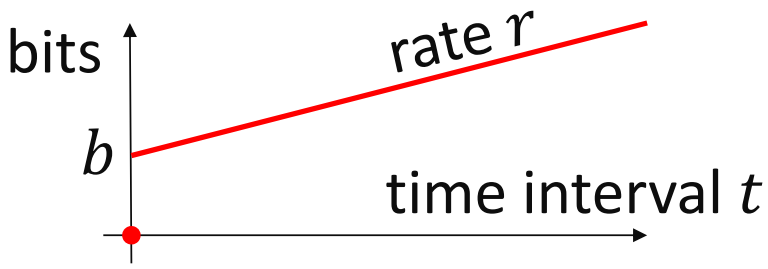
Flow with cumulative function $R(t)$ has α as (maximal) arrival curve if

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

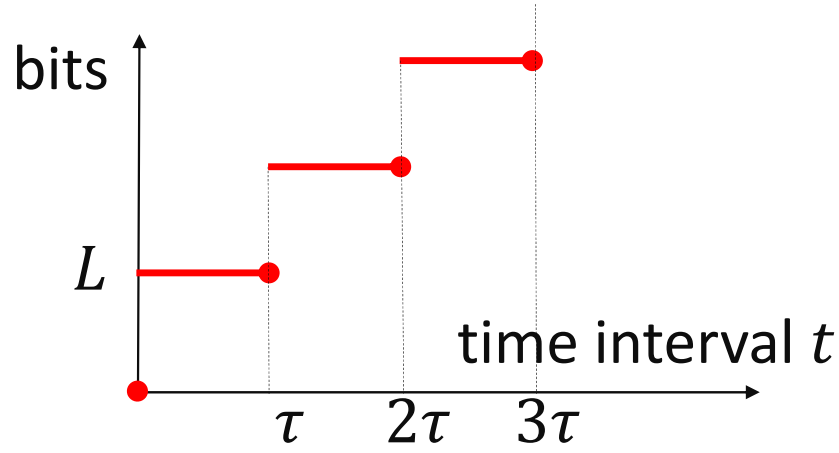
where α is a monotonic nondecreasing function $\mathbb{R}^+ \to [0, +\infty]$

token bucket constraint (r, b) with rate r and burst b:

$\alpha(t) = rt + b$



periodic stream of packets of size $\leq L$: $\alpha(t) = L \left\lceil \frac{t}{\tau} \right\rceil$



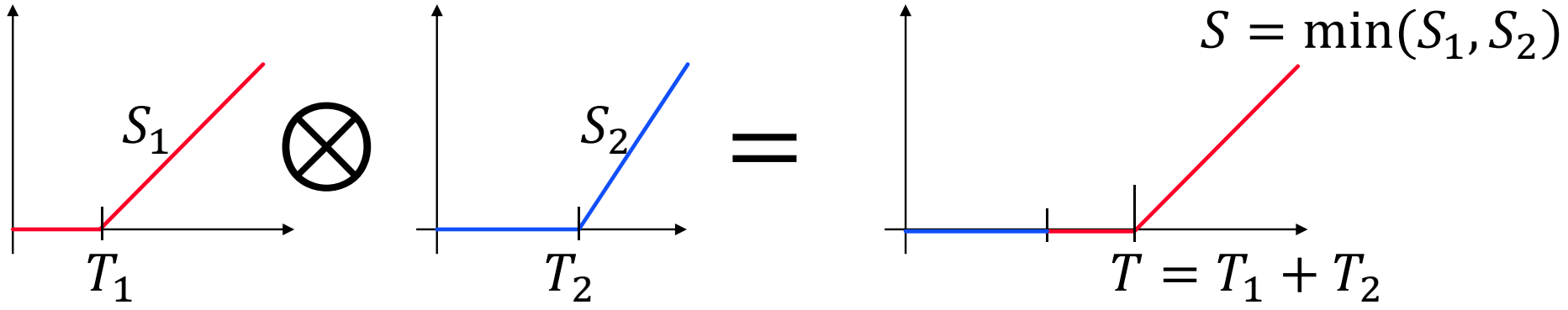
Min-Plus Convolution of wide-sense increasing functions $[0; +\infty) \to [0; +\infty]$

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

This operation is called *min-plus convolution*. It has the same nice properties as usual convolution; e.g.

$$(f_1 \otimes f_2) \otimes f_3 = f_1 \otimes (f_2 \otimes f_3)$$
$$f_1 \otimes f_2 = f_2 \otimes f_1$$

It can be computed directly or with tools [Zhou 2020]



Min-Plus Convolution and Arrival Curves

α is an arrival curve for $R \Leftrightarrow R(t) \leq R(s) + \alpha(t-s), \forall s \in [0,t]$

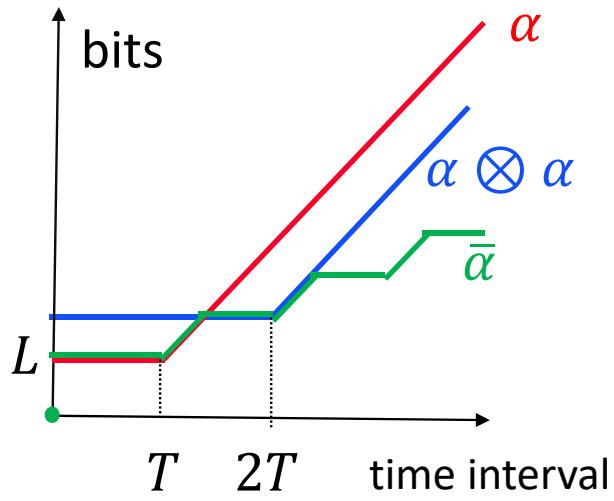
$\Leftrightarrow R \leq R \otimes \alpha$

Any arrival curve α can be replaced by its sub-additive closure

$\bar{\alpha} = \inf\{\delta_0, \alpha,\ \alpha \otimes \alpha,\ \alpha \otimes \alpha \otimes \alpha, \ldots\}$

with $\delta_0(0) = 0,\ \ \delta_0(t) = +\infty$ for $t > 0$

$\bar{\alpha}$ is sub-additive, i.e. $\bar{\alpha}(s+t) \leq \bar{\alpha}(s) + \bar{\alpha}(t)$
and $\bar{\alpha}(0) = 0$



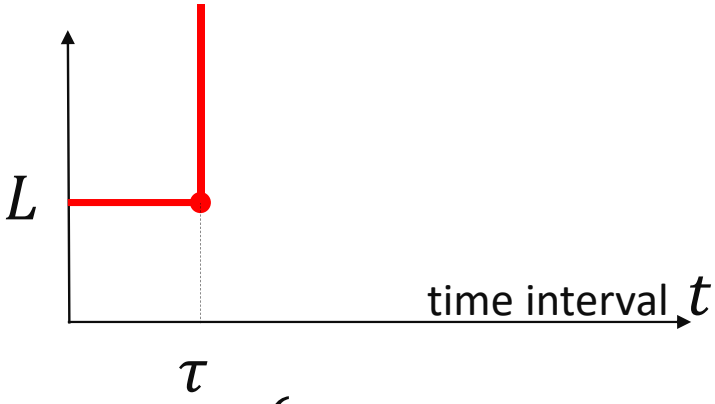
Example of Sub-Additive Closure

Flow has at most L bits in any interval of fixed duration τ

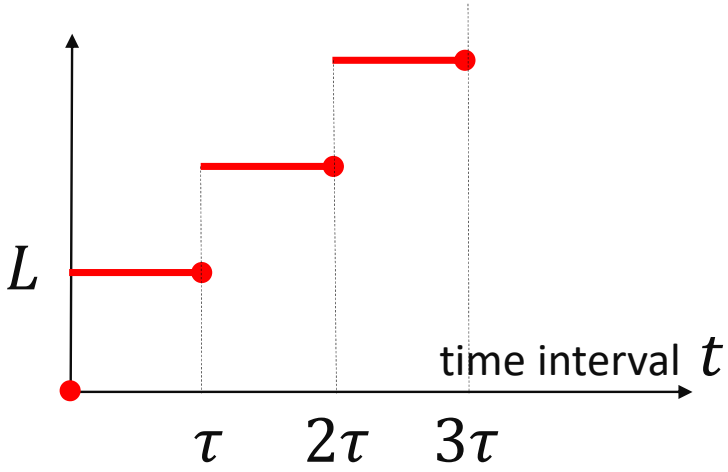
$\Leftrightarrow R(t+\tau) - R(t) \leq L$ for all t

$\Leftrightarrow$ flow has arrival curve α

$\Leftrightarrow$ flow has arrival curve $\bar{\alpha}$

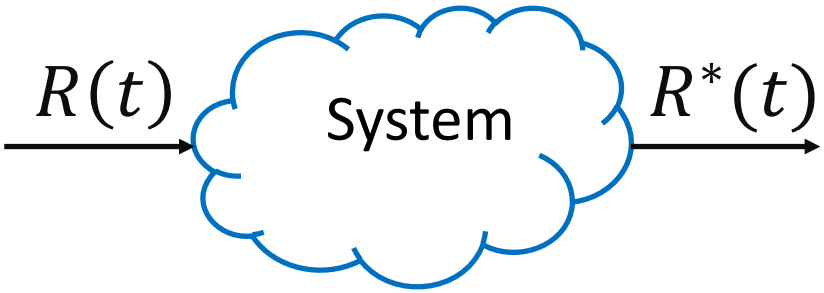


$$\alpha(t) = \begin{cases} L, t \leq \tau \\ +\infty, t > \tau \end{cases}$$



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

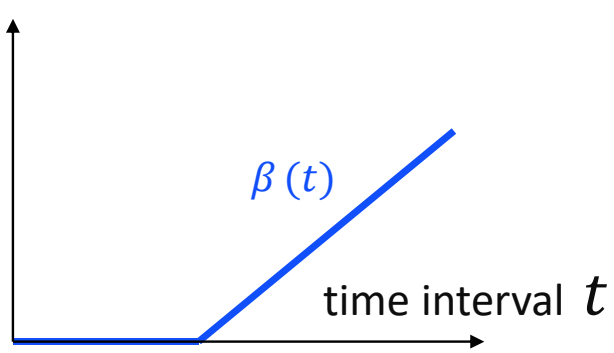
Service Curve

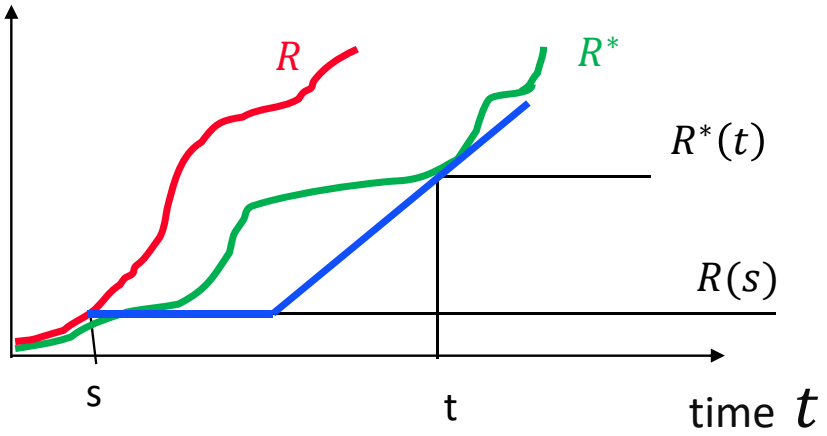


System offers to this flow a (minimal) service curve β if $R^* \geq R \otimes \beta$, i.e. :

$$\forall t \geq 0, \exists s \in [0,t]: R^*(t) \geq R(s) + \beta(t-s)$$

where β is a function : $\mathbb{R}^+ \to \mathbb{R} \cup \{+\infty\}$



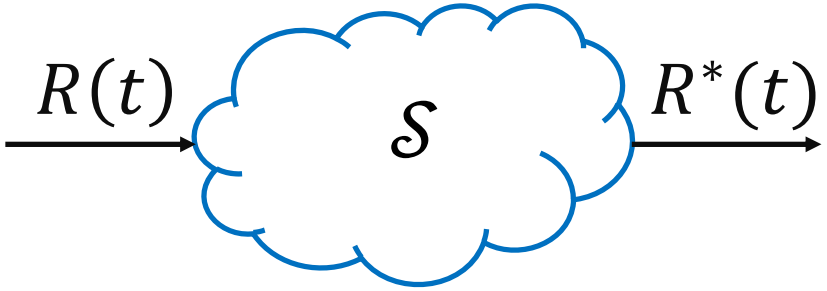


[Le Boudec 1996, Chang 1997, Bouillard 2018]

Strict Service Curve

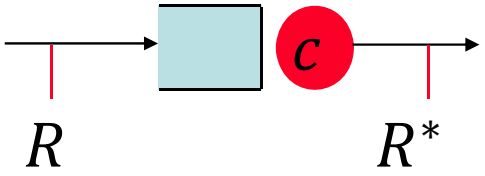
System $\mathcal{S}$ offers to a flow a strict service curve β if for any $s < t$ inside a backlogged period, i.e. such that $R^*(u) < R(u), \forall u \in (s, t]$, we have $R^*(t) - R^*(s) \geq \beta(t - s)$

$\mathcal{S}$ is typically a single queuing point



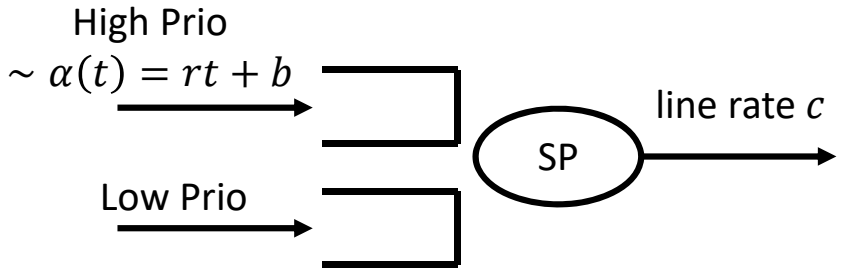
β is a strict service curve $\Rightarrow$ β is a service curve but converse is not true.

Example: constant rate server with line rate c has strict service curve $\beta(t) = ct$



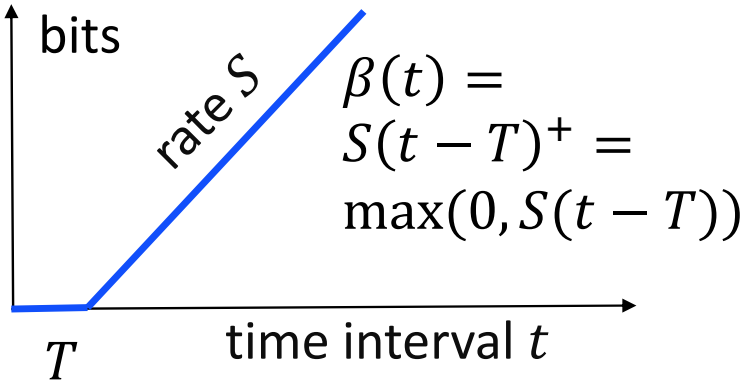
Example: Non-preemptive Static Priority

High prio: $\beta_H(t) = (ct - MTU_L)^+$
(strict service curve)
(MTU_L = max packet size, low prio)



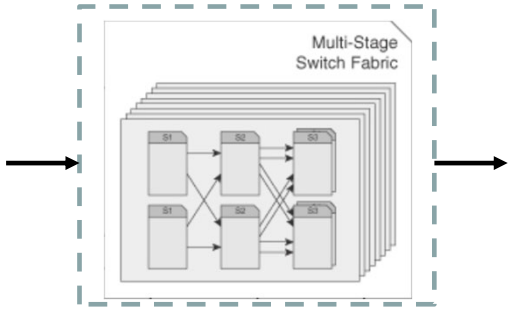
Low prio: when high priority constrained by $\alpha(t) = rt + b, r < c$:
$\beta_L(t) = ((c - r)t - b)^+$ (not a strict service curve)
$\beta'_L(t) = \left((c - r)t - b - MTU_L\right)^+$ (strict service curve)
[Bouillard 2018]

A function of the form $\beta(t) = S(t - T)^+$ is called rate-latency, with rate S and latency T



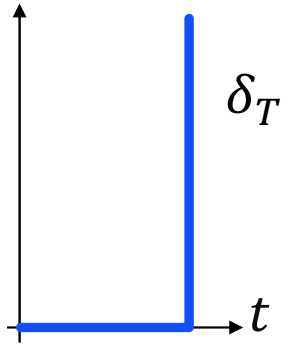
Bounded Delay Element

Sometimes it is convenient to model a system as a black box with known delay upper bound T.



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

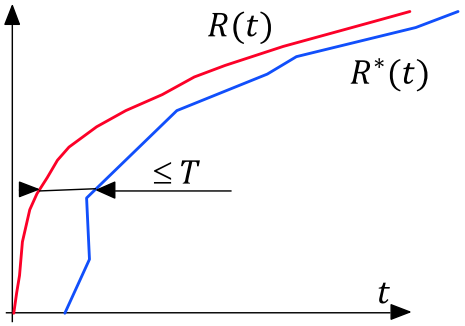
For a node that is FIFO for this flow: delay $\leq T \Leftrightarrow$ nodes offers to this flow a service curve δ_T



$\delta_T(t) = 0$ if $t \leq T$

$\delta_T(t) = +\infty$ if $t > T$

Not a strict service curve



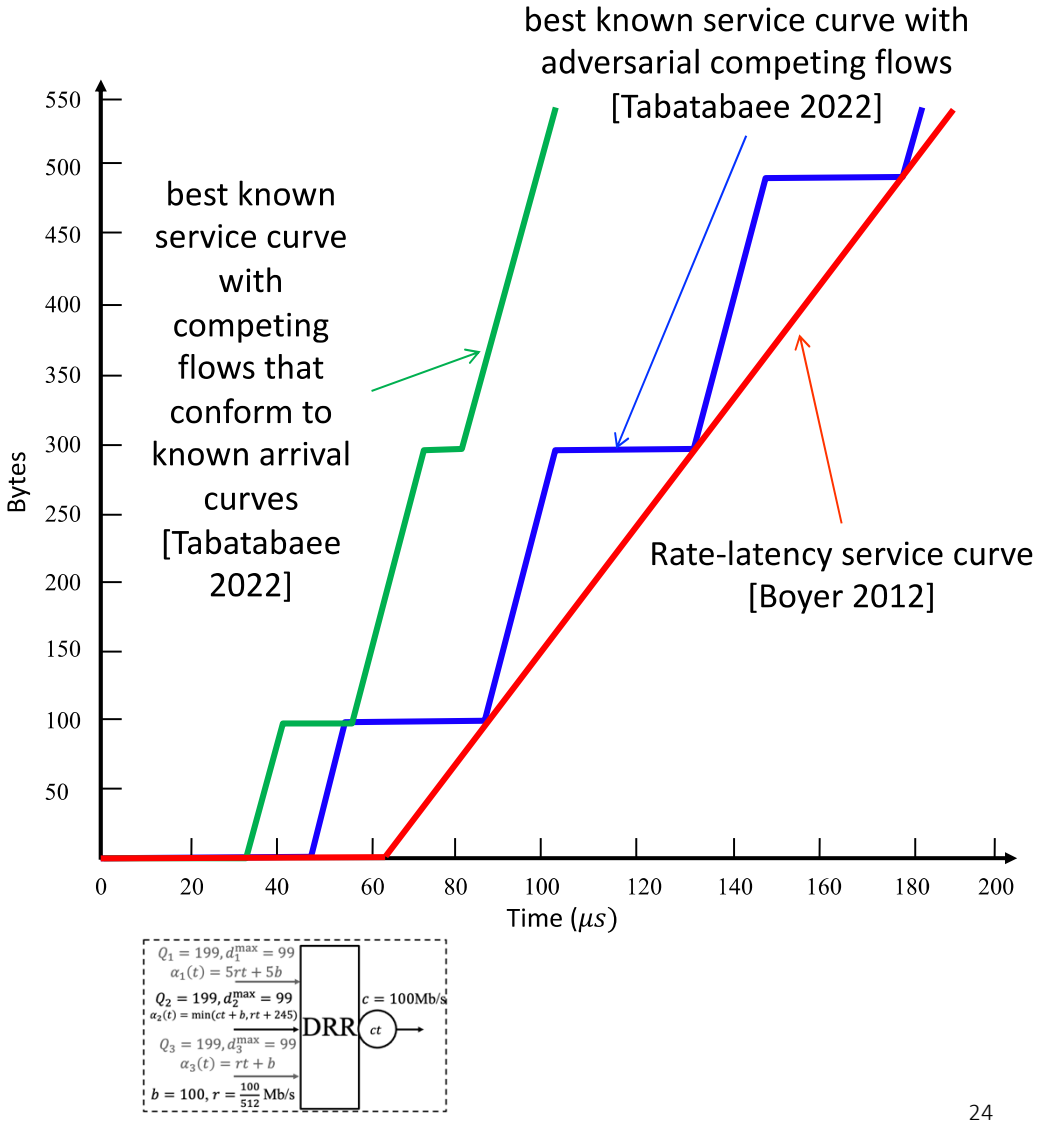
Example: Deficit Round Robin

- DRR offers to flow i a rate-latency strict service curve $\beta_i(t) = R_i(t - T_i)^+$

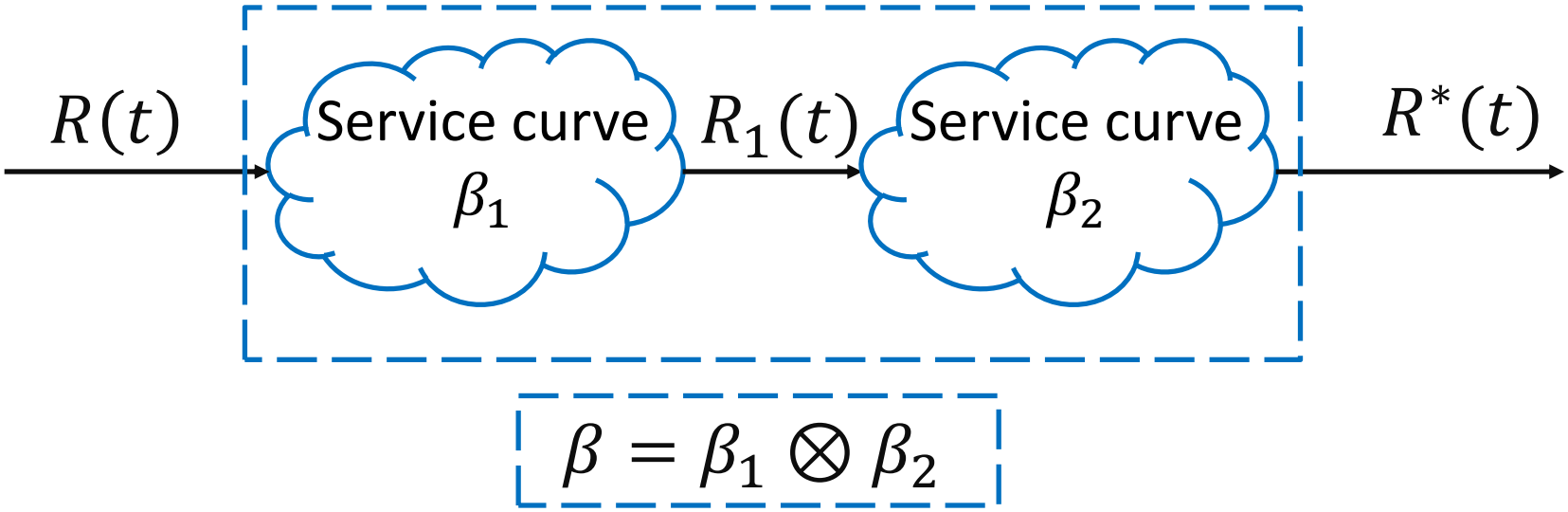
with $R_i = \frac{Q_i}{\sum_j Q_j} c, \quad T_i = \frac{\bar{Q}_i + \bar{L}_i}{c} + L_{\max,i}(\frac{1}{R_i} - \frac{1}{c}), \bar{Q}_i = \sum_{j \neq i} Q_j, \quad \bar{L}_i = \sum_{j \neq i} L_{\max,j}$ and c is the line rate [Boyer 2012].

Can be improved to more accurate service curves [Tabatabaee 2022]

- Other examples: Packetized Generalized Processor Sharing, RFC 2212, IEEE AVB, IEEE TSN, etc. [De Azua 2014] [Bouillard 2018]



Concatenation of Service Curves

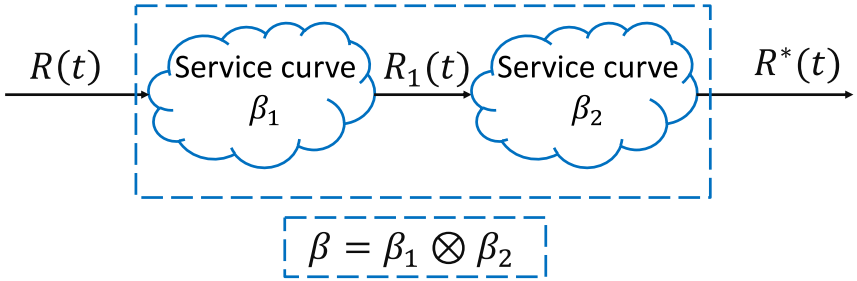


A flow is served in series, network element i offers service curve β_i.

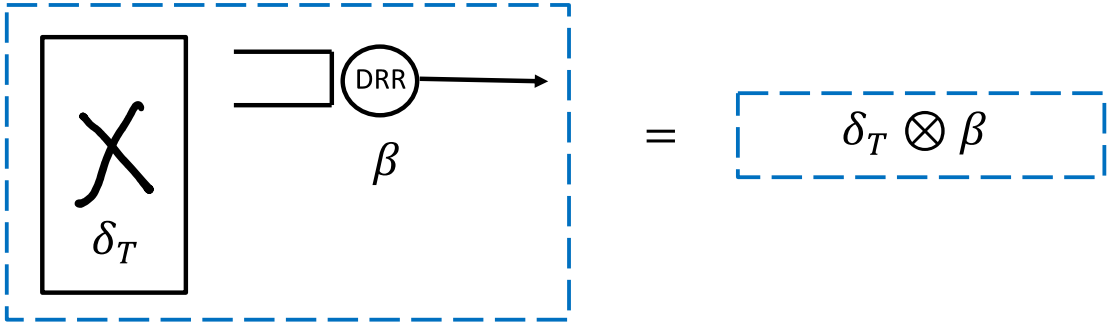
The concatenation offers to flow the service curve $\beta = \beta_1 \otimes \beta_2$

Proof: $R^* \geq R_1 \otimes \beta_2 \geq (R \otimes \beta_1) \otimes \beta_2 = R \otimes (\beta_1 \otimes \beta_2)$

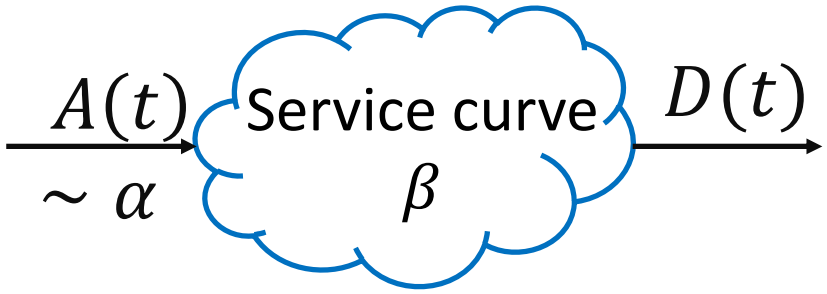
Example

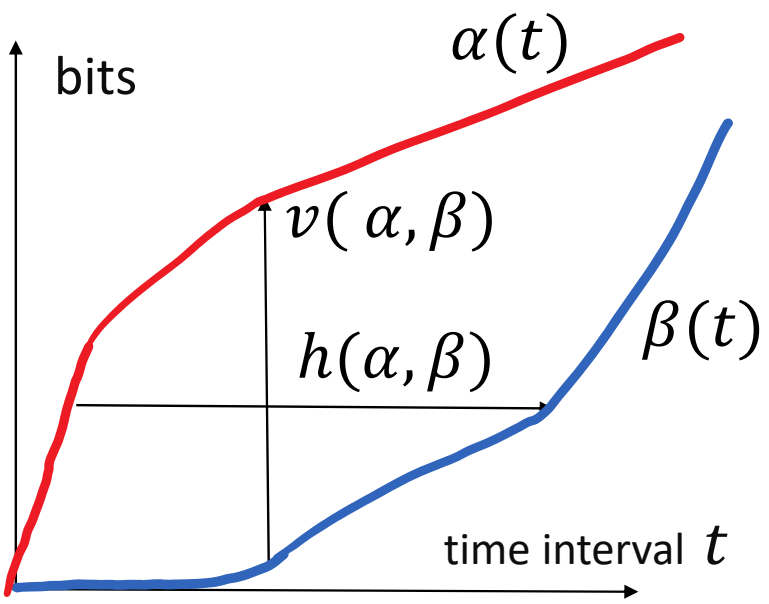


- If β_i is rate-latency R_i, T_i then the concatenation $\beta = \beta_1 \otimes \beta_2$ is rate-latency $R = \min(R_1, R_2)$ and $T = T_1 + T_2$
- a scheduler with service curve β combined with a bounded delay element with delay bound T can be modelled with service curve $\beta \otimes \delta_T$



Three Tight Bounds





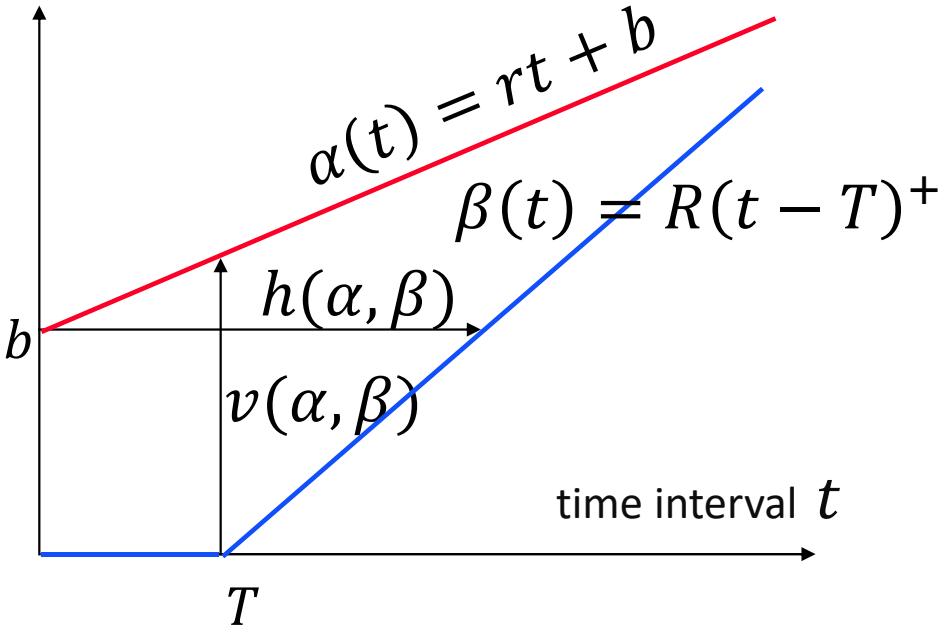
Flow is constrained by arrival curve α; served in network element with service curve β. Then

1. backlog $\leq v(\alpha,\beta) = \sup_t \big(\alpha(t) - \beta(t)\big)$
2. if FIFO for this flow, delay $\leq h(\alpha,\beta)$
3. output is constrained by arrival curve
   $\alpha^*(t) = \sup_{u \geq 0} \big(\alpha(t+u) - \beta(u)\big)$
   i.e. $\alpha^* = \alpha \oslash \beta$ (deconvolution)

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

Delay bound can be improved if we know line rate of server [Mohammadpour 2019]

Example



One flow, constrained by one token bucket is served in a network element that offers a rate latency service curve

Assume $r \leq R$

Backlog bound: $b + rT$

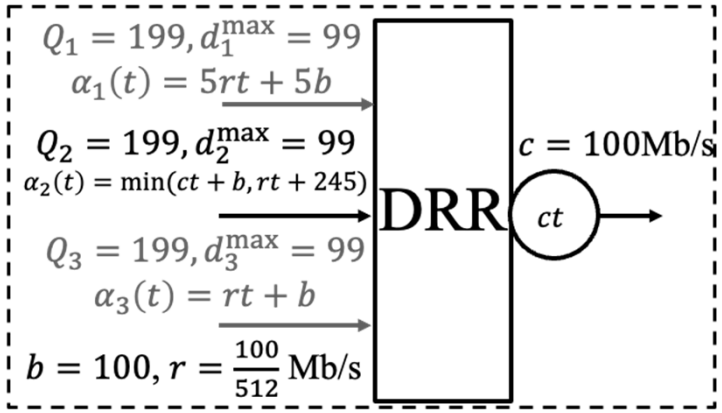
Delay bound: $\frac{b}{R} + T$

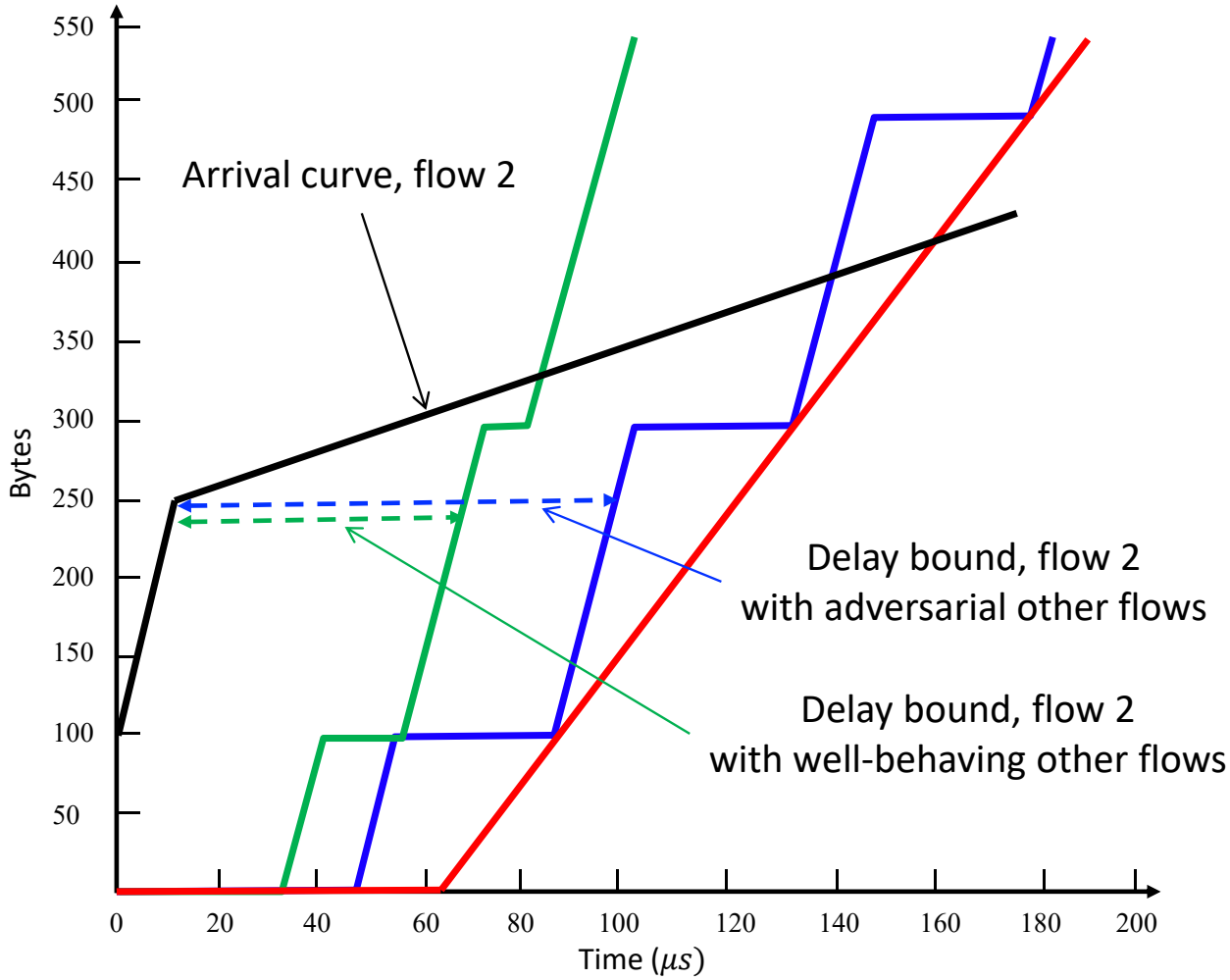
Output arrival curve:

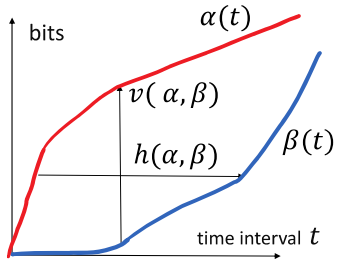
$$\alpha^*(t) = rt + b^*$$

with $b^* = b + rT$

Example: DRR


$Q_1 = 199, d_1^{\max} = 99$
$\alpha_1(t) = 5rt + 5b$
$Q_2 = 199, d_2^{\max} = 99$
$\alpha_2(t) = \min(ct + b, rt + 245)$
$Q_3 = 199, d_3^{\max} = 99$
$\alpha_3(t) = rt + b$
$b = 100, r = \frac{100}{512}$ Mb/s
DRR
ct
$c = 100$Mb/s


Arrival curve, flow 2
Delay bound, flow 2
with adversarial other flows
Delay bound, flow 2
with well-behaving other flows
Bytes
Time (μs)
550
500
450
400
350
300
250
200
150
100
50
0
20
40
60
80
100
120
140
160
180
200



Network calculus uses arrival curves and service curves to derive delay and backlog bounds.

Single node analysis follows immediately.

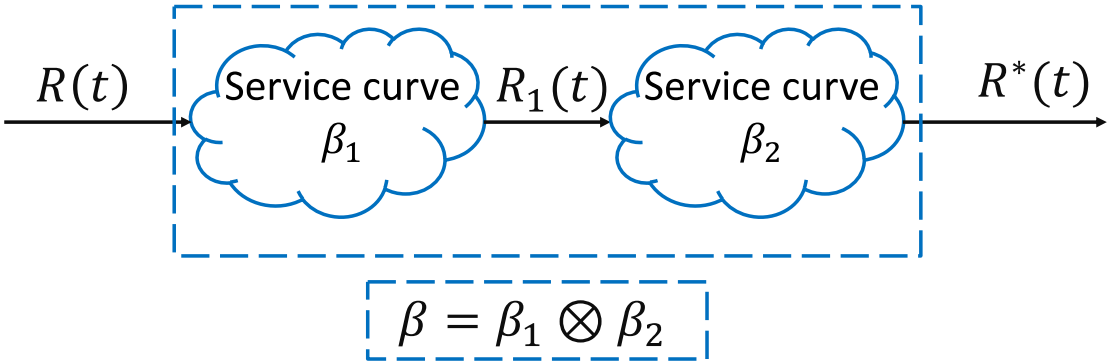
How about network analysis ?

3. Network Analysis

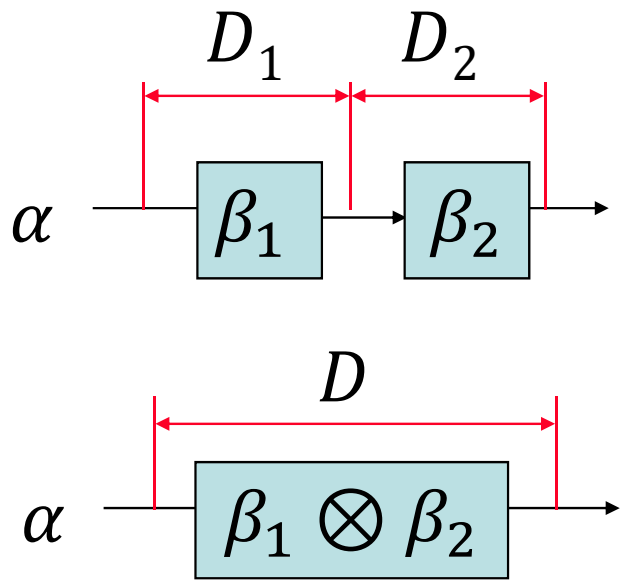
Per-flow network:

network nodes offer guarantees to individual flows
e.g. IETF IntServ

Solution: apply concatenation result



Pay Bursts Only Once



$$\alpha(t) = rt + b$$
$$\beta_1(t) = R(t - T_1)^+$$
$$\beta_2(t) = R(t - T_2)^+$$
$$r \leq R$$

In per-flow Network:
one flow constrained *at source* by α

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

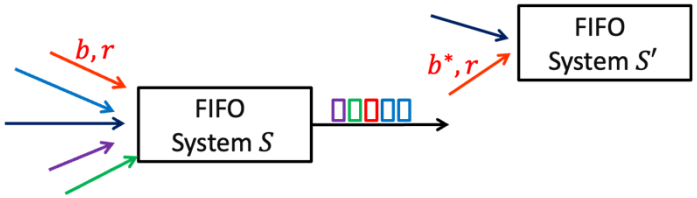
$$D_1 + D_2 = \frac{2b + rT_1}{R} + T_1 + T_2$$

computed *by concatenation:*

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

i.e. worst cases cannot happen simultaneously

FIFO Per-Class Networks



Most time sensitive networks are FIFO per-class:

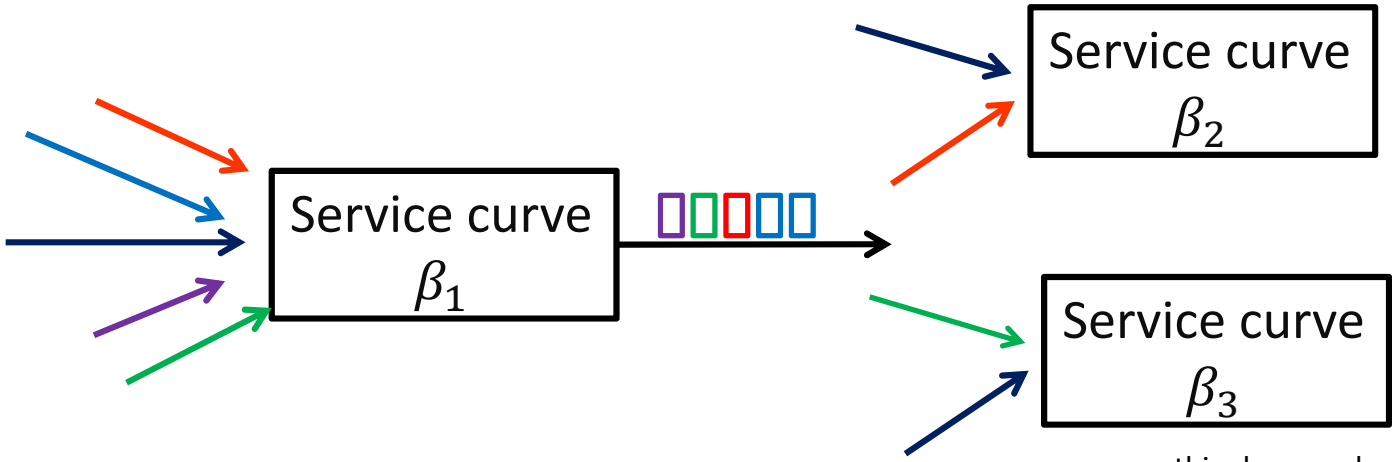
- flows are assigned to classes
- schedulers (such as DRR) separate classes and provide service guarantee to the aggregate of all flows of this class
- Inside a class, service is FIFO

- flows are constrained at sources by arrival curves

Using service curves, such a network can be analyzed per-class

→ one separate FIFO network model per class

FIFO Networks



this shows only one class;

the service curves are offered to aggregate of all flows.

Flows merge and split, no simple result as in per-flow networks.

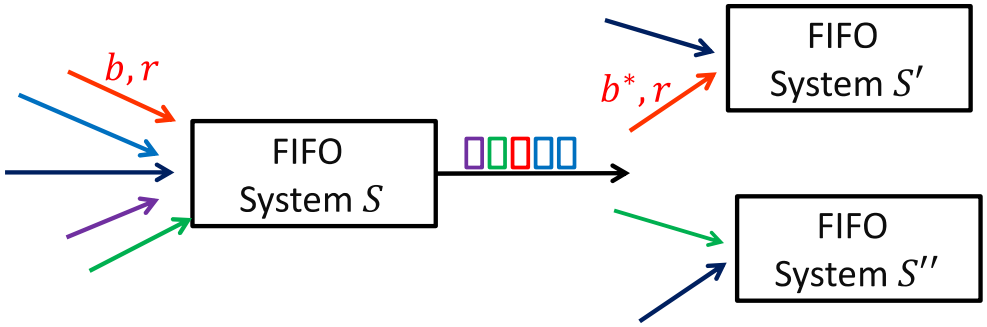
Feedforward networks: obtaining worst-case delay is NP-hard [Bouillard 2010]

Can be computed with ELP (Exponential Linear Programming) [Bouillard 2014]

- service curve, arrival curve and FIFO are expressed as constraints in a linear program
- super-exponential complexity

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

Simple, commonly used method to analyze a generic deterministic network



- Sources are constrained by token buckets

a) Propagated burstiness of flow at point inside the network is computed by
$b^* = b + r \times (\text{delay bound between source and here})$

b) Delay at every node is computed using single node network calculus, using the propagated burstinesses.

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

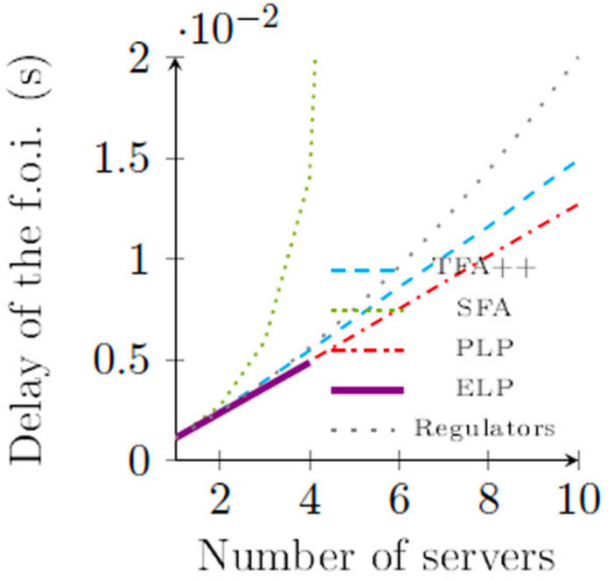
- In a feedforward network of depth d, start at edge nodes and stop in d iterations
- In a generic network, iterate a) and b) at all nodes until convergence to a fixpoint or move to infinity.
  If convergence, the bounds are valid. If divergence, we don't know. [Thomas 2019, Plassart 2022]

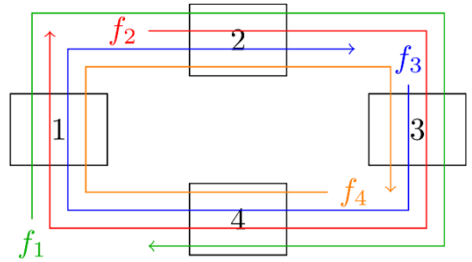
PLP and Other Methods

- PLP (Polynomial Linear Programming) [Bouillard 2022]: relaxation of ELP, with polynomial complexity, uses TFA (and other) bounds as constraints, applies to generic topologies

- Many other methods exist: SFA [Schmitt 2006, Grieu 2004], PMOO, LUDB [Fidler 2003, Lenzini 2006, Bondorf 2017,Geyer 2022] but do not apply to generic topologies;

- Tools: DISCO [Schmitt 2006], WoPaNets [Mifdaoui 2010], Pegase [Boyer 2010]





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 rate-latency service curve β_{R_i,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: $u = 1$;

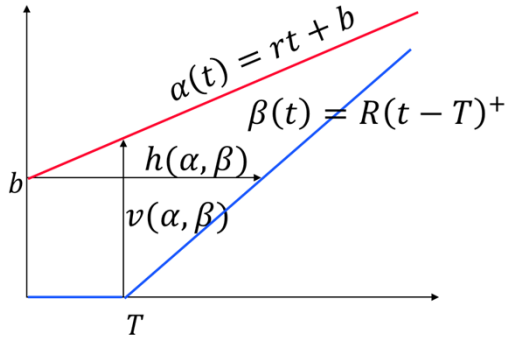
One network instance $= (\mathcal{F}, r, b, F, \mathcal{I}, R, T)$

A network instance is stable if there *is* a bound on all delays (or backlogs), that is valid for any execution trace of the network.

(existence of a bound on all delays ⇔ existence of a bound on all backlogs)

Which FIFO Networks are Stable ?

- 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.

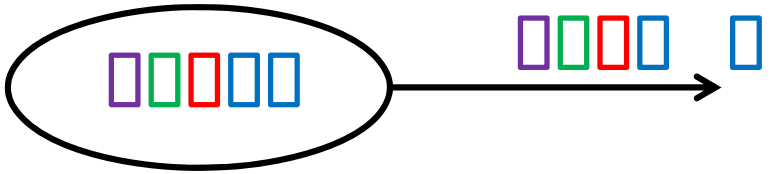
In per-flow networks, deterministic Network analysis is as simple as single node.

In per-class networks and arbitrary topologies, requires finding fixpoints (with e.g. TFA or PLP).

Underloaded networks may be unstable.

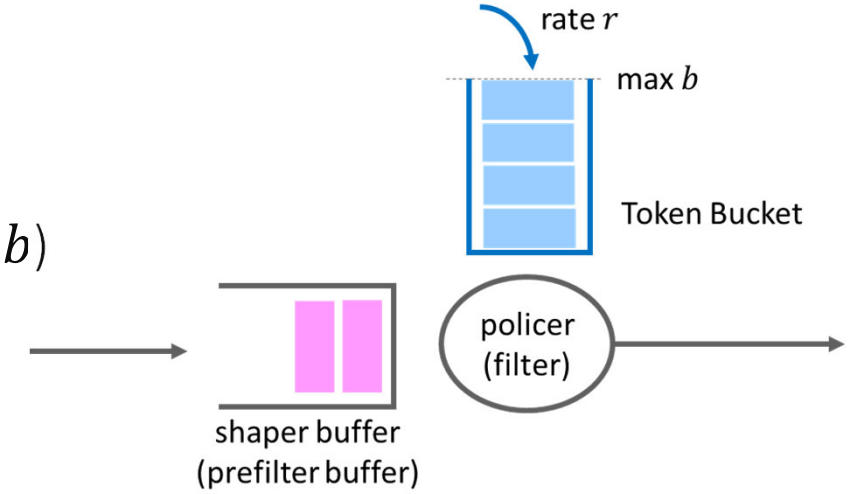
4. Regulators

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



Non work-conserving.

Example: Token Bucket regulator (regulator for the arrival curve constraint $\alpha(t) = rt + b$)



Typically placed at source / network edge to protect deterministic network from misbehaving sources

Can also be used inside the network

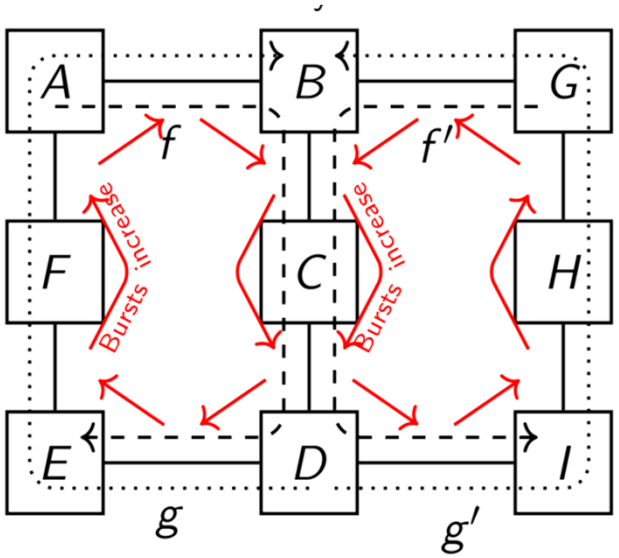
Cascading Burstiness

In a per-flow network, burstiness of a flow increases linearly with number of hops, but pay-bursts-only allows to still have good delay bounds.

In per-class networks, burstiness of every flow increases at every hop as a function of other flows' burstiness:

$$b_f^* = b_f + r\left(T + \frac{b_{tot} - b_f}{R}\right)$$

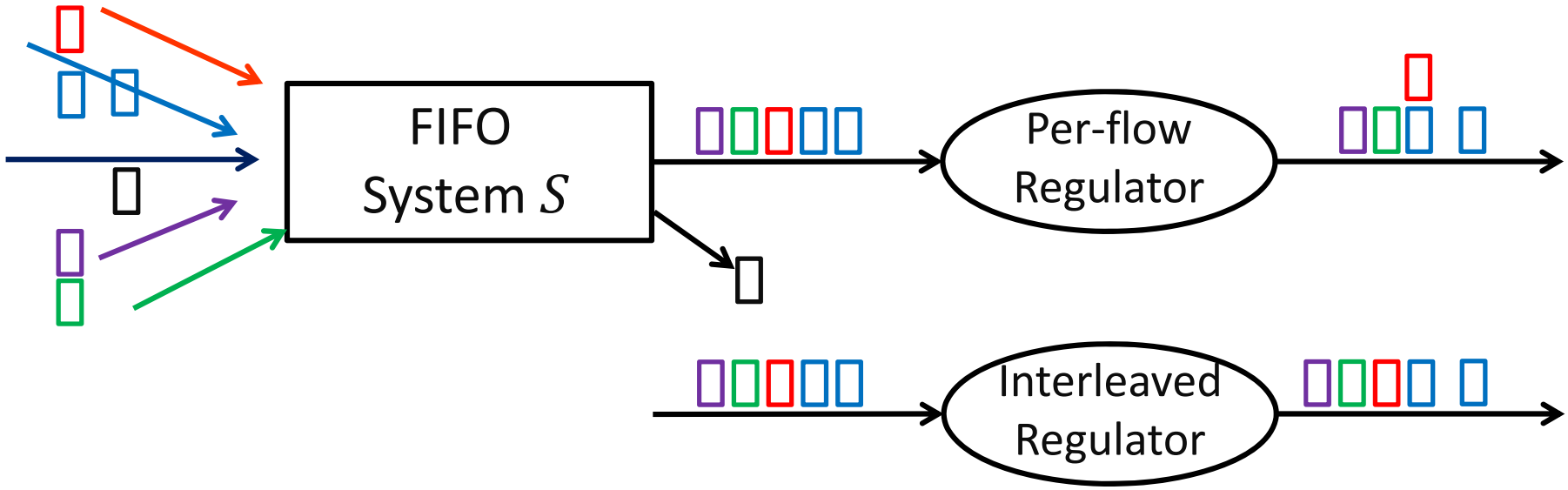
Increased burstiness causes increased burstiness (cascade).



Propagated burstiness is computed by PLP / TFA as solution to a fixpoint problem.

Cyclic dependencies are root cause for bad worst-case delays.

Regulators Avoid Cascading Burstiness in Per-Class Networks

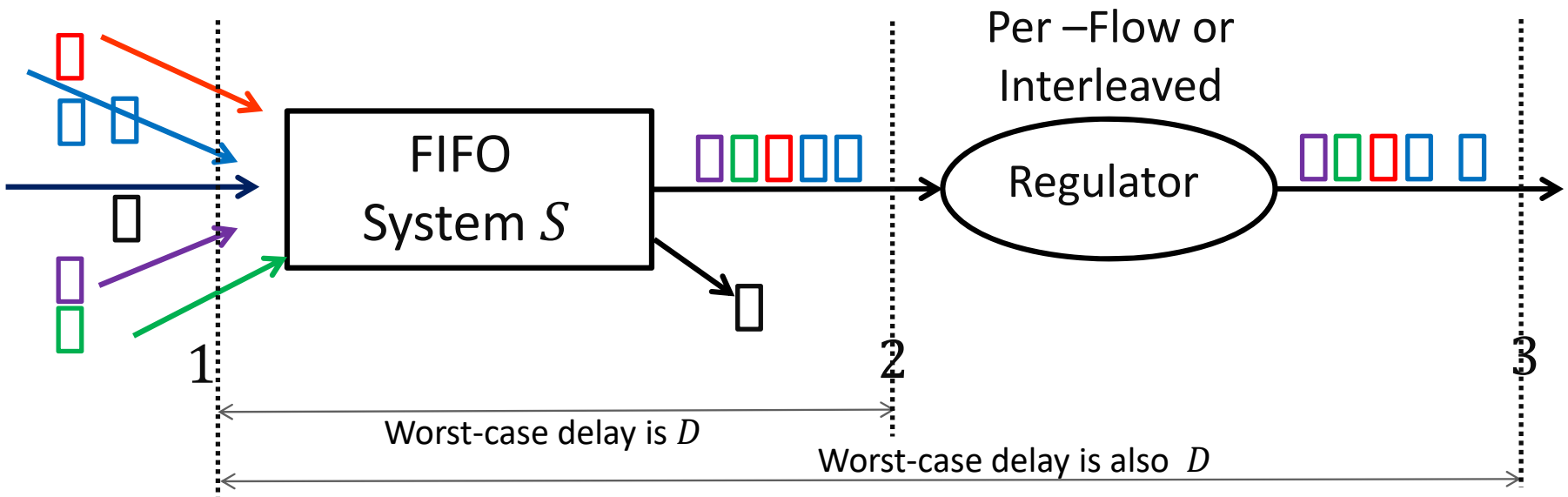


Per flow regulator: one state + one queue per flow.

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).

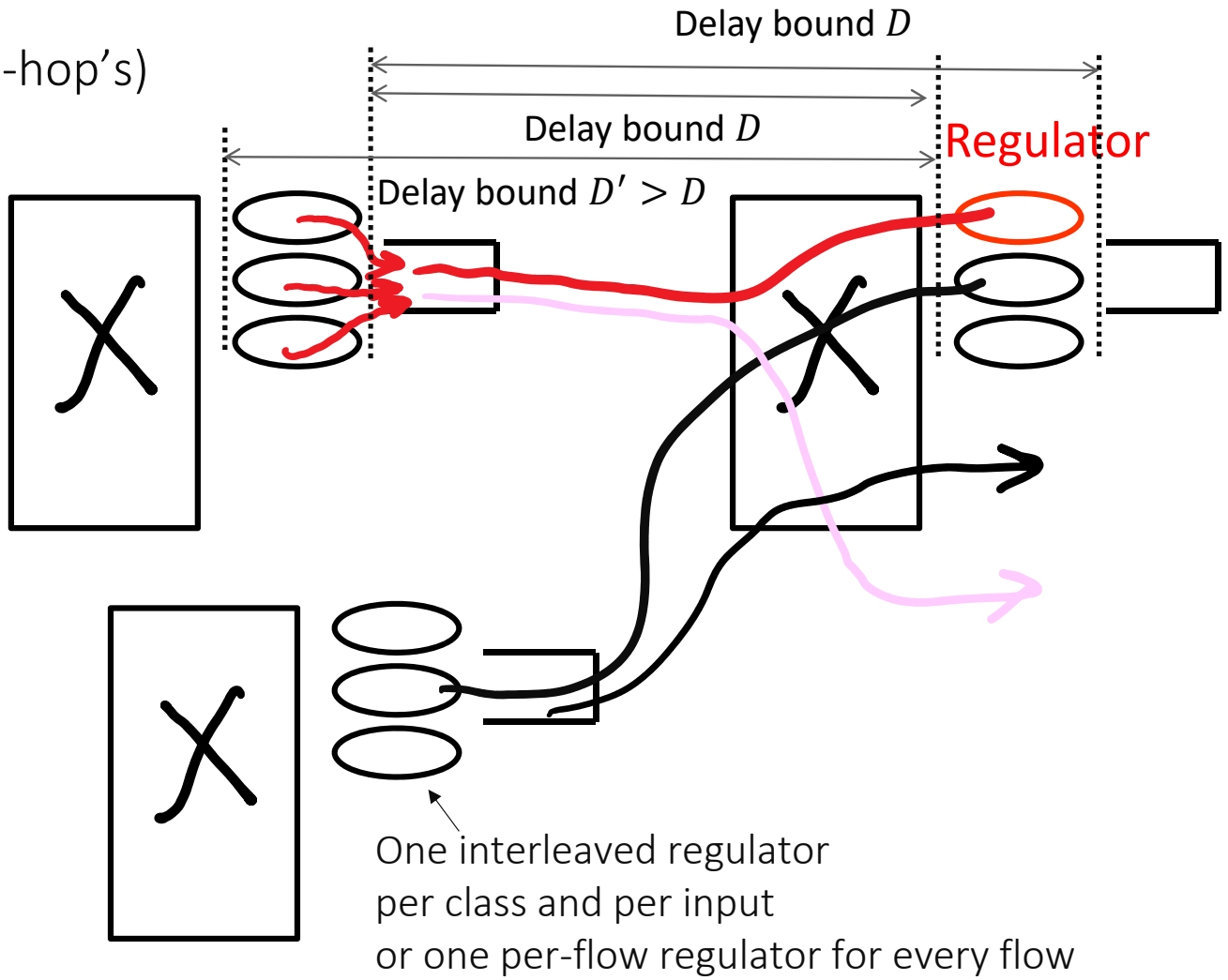
Assume every flow satisfies some arrival constraint at 1 (e.g. rate and burstiness) and regulators enforces same constraint at 3.

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

Network With Regulators [IEEE TSN ATS]

- Regulators are integrated in (next-hop's) queuing system.
- Worst case end-to-end queuing delay can ignore regulators. Worst-case delay at one regulator is absorbed by delay bound at previous hop.
- Queuing delay and backlog at every hop can be computed using single node analysis.
- Underloaded network is always stable.

[Mohammadpour 2018]

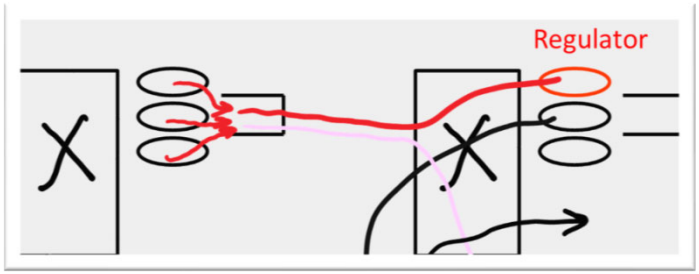


Deterministic networks use regulators at edge to protect determinism

Can also be deployed internally to avoid burstiness increase / to simplify network analysis

Re-shaping is for free (w.r. to worst-case delay)

5. Clock Non Idealities

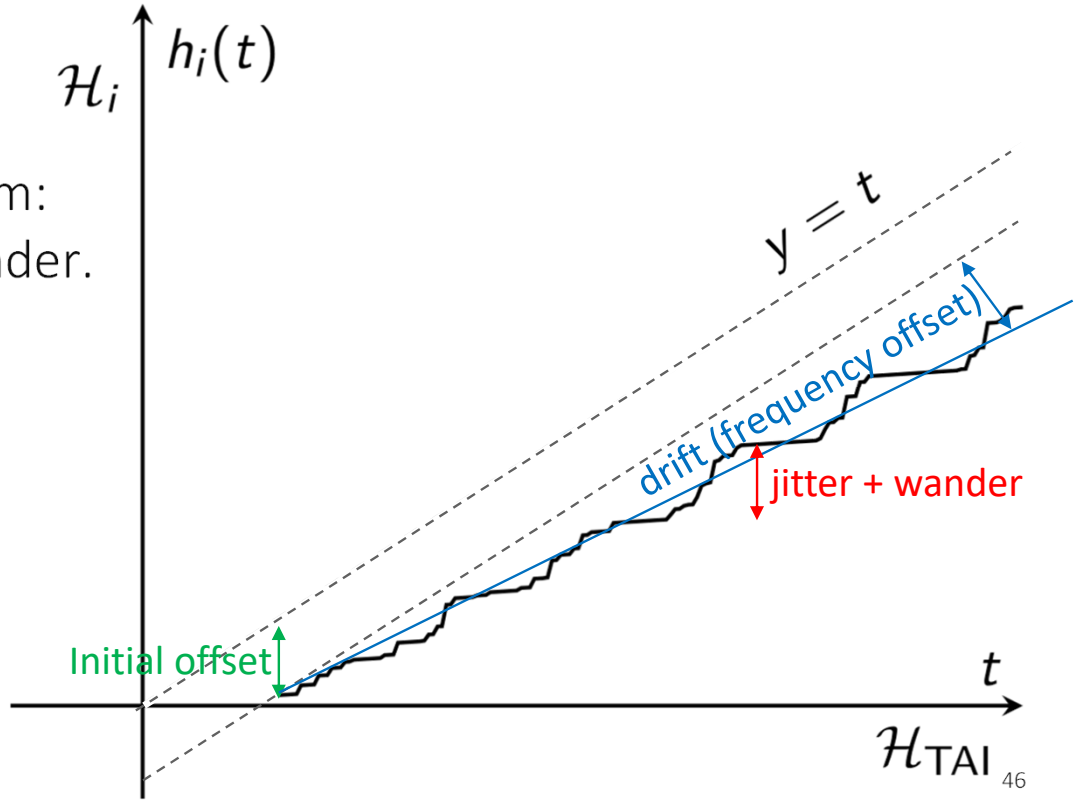


Previous theory assumes perfect time everywhere.

In reality, nodes use local clocks that are not ideal.

- tight sync (PTP, White Rabbit, GPS) : timestamping error $\leq \omega \approx$ 10ns–1μs
- loose sync (NTP): $\omega \approx$ 1ms – 1s
- no sync: timestamping error ω 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 → d and with another clock → 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

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

ω = time error bound
= 1μs in TSN with PTP;
= $+\infty$ if no synchronization

ρ = clock-stability bound
=1.0001 (e.g. in TSN)

η = timing-jitter bound
= 2ns (e.g. in TSN)

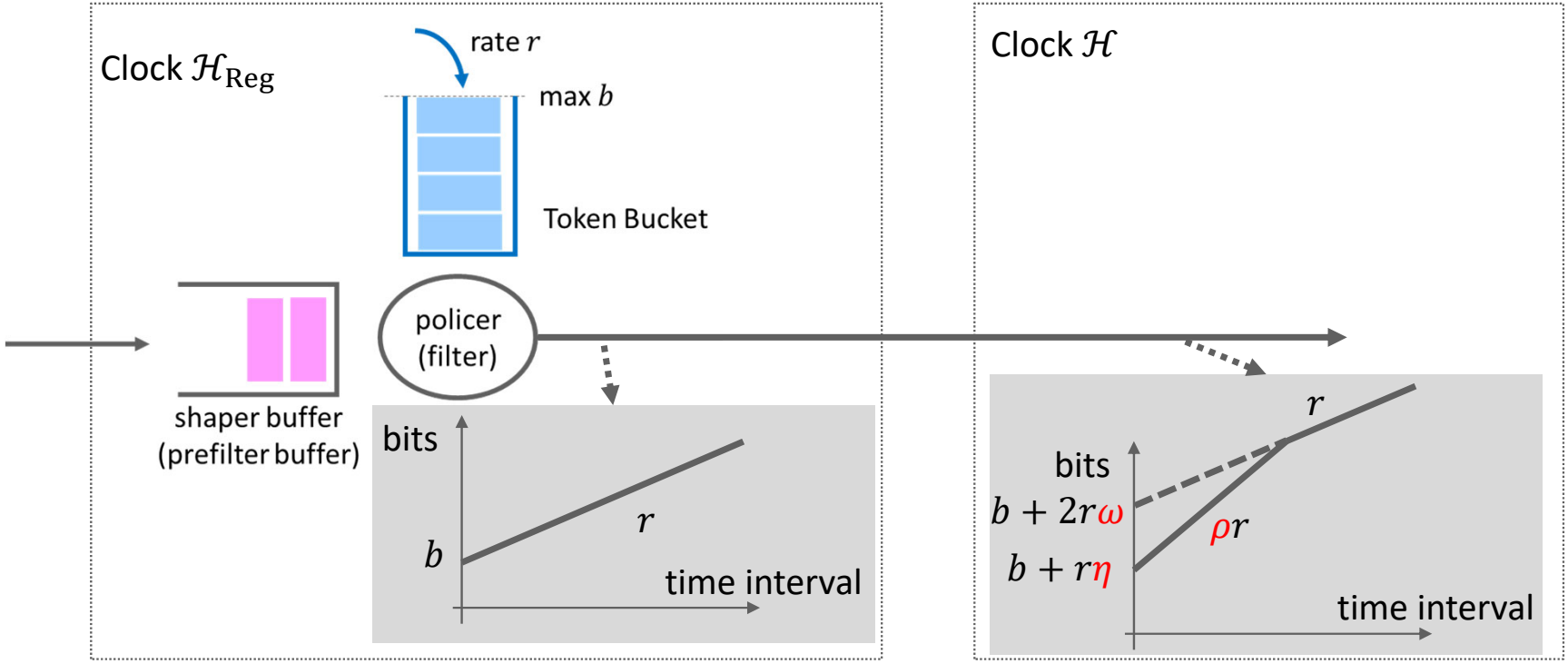
Model is symmetric, i.e. same inequalities if we exchange $d' \leftrightarrow d$

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

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



Consequences for Non-Adapted Regulators

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

Perfect clocks:

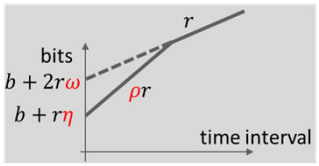
- Regulator does not increase worst-case delay

Non-synchronized network:

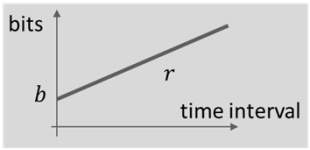
- Per-flow and interleaved regulator unstable (unbounded delay).

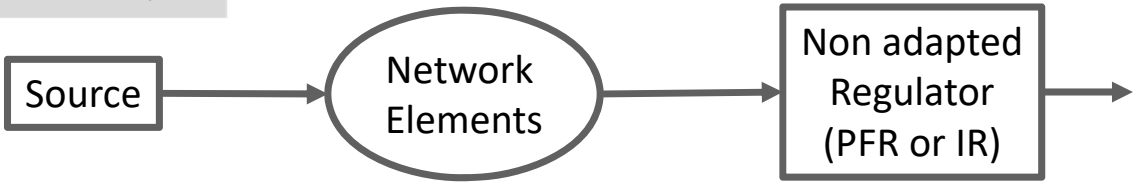
Synchronized network:

- Per-flow regulator incurs delay penalty up to 4ω;
- Interleaved regulator is unstable.



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



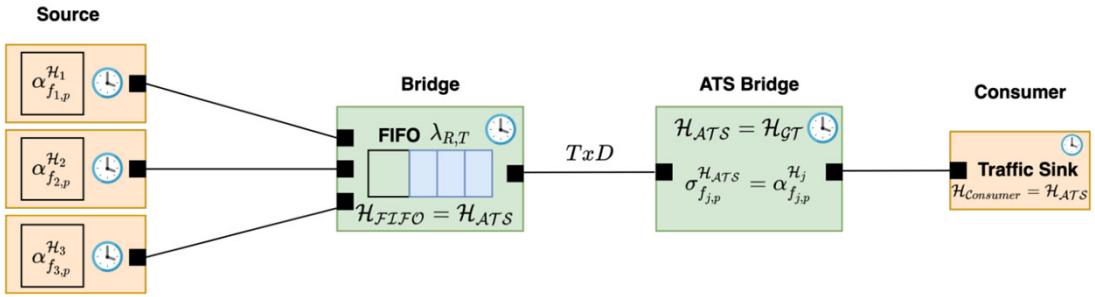


Flow constrained by $\alpha(t) = rt + b$ in local clock

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

Synchronized clocks,
Unstable non-adapted Interleaved Regulator (= IEEE TSN ATS)

ns-3 simulations



3 sources @ 147 kb/s

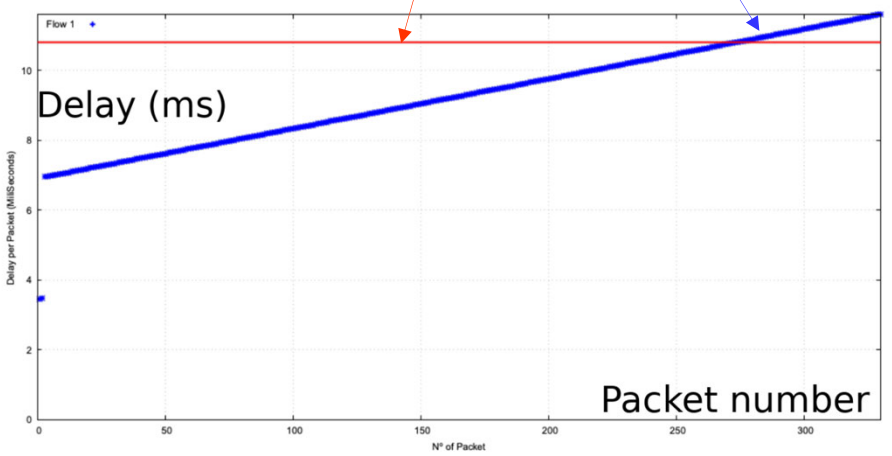
$\omega = 1\mu s, \rho = 1.0001$

Delay increases by up to 100μs per second of operation.

[Thomas 2020]

Delay bound ignoring clock non ideality

Actual packet delay

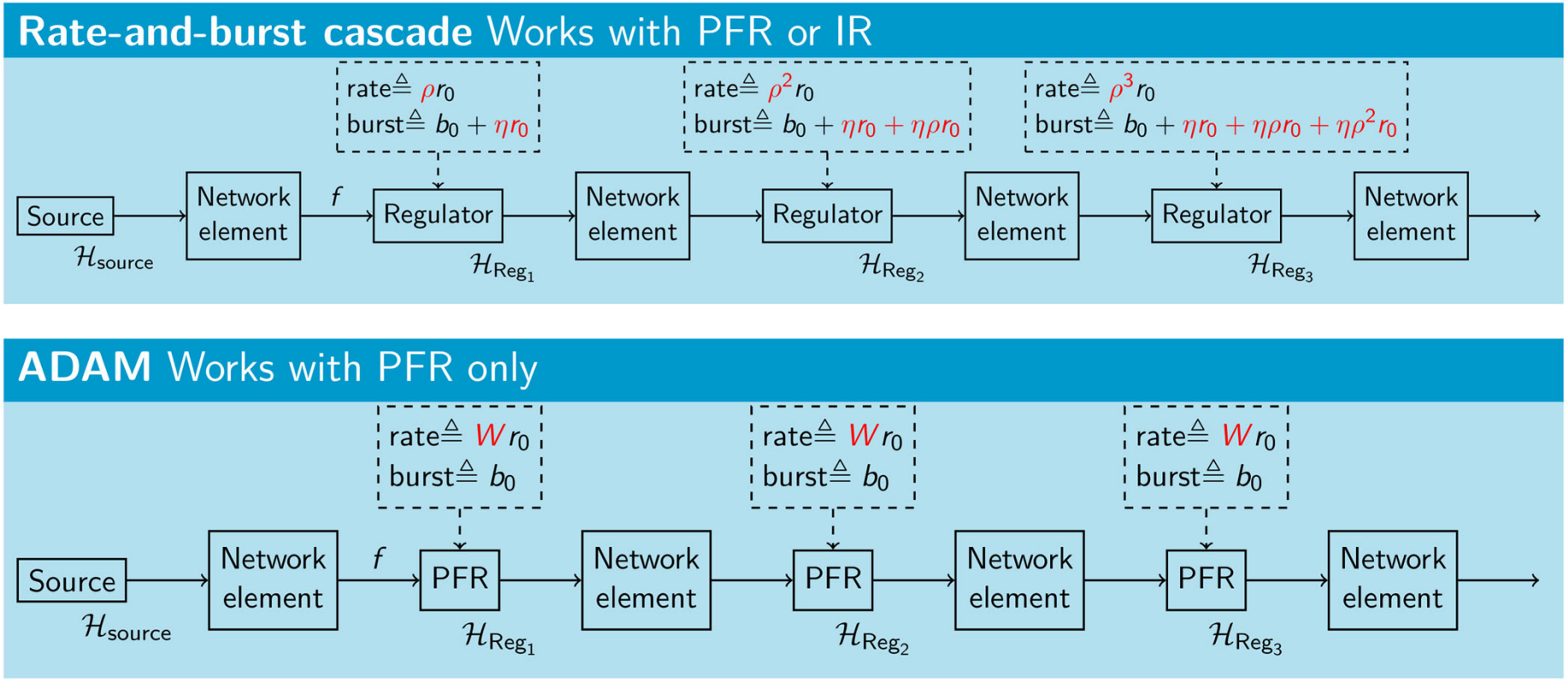


Work by Guillermo Aguirre

Regulators are sensitive to clock inaccuracies

In tighty synchronized networks, IR must be adapted otherwise is unstable; PFR need not be adapted but increased delay due to clock inaccuracy must be accounted for. In loosely synchronized or non synchronized networks, both PFR and IR must be adapted.

Examples of adaptation methods

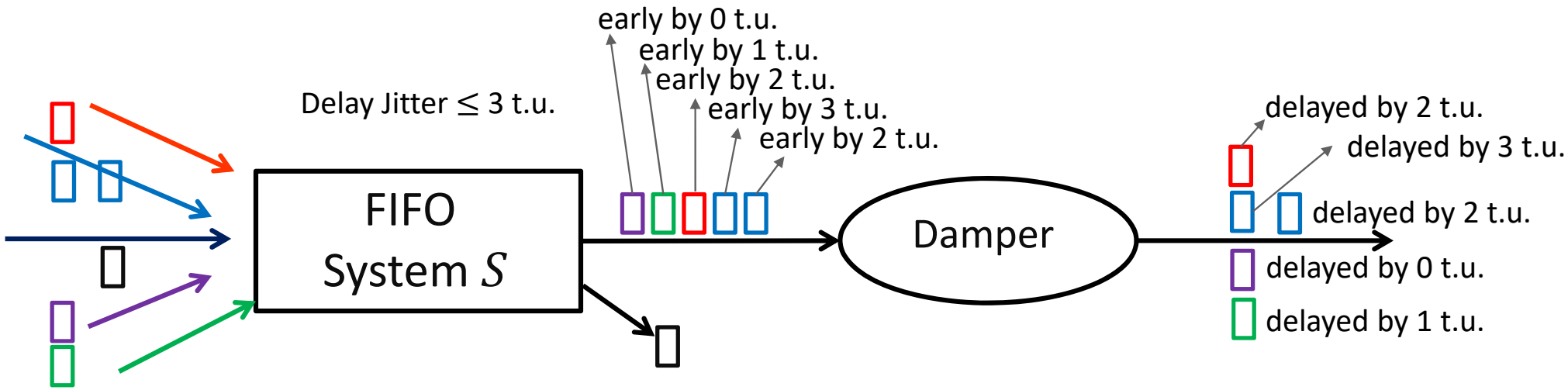


Clock non idealities can easily be accounted for in a network calculus analysis

Both for synchronized and non-synchronized networks

Arrival curves and delay bounds are (very slightly) affected, but regulators are dramatically affected and need to provision safety margins

6. Dampers



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

Removes (almost all) jitter.

Non work-conserving

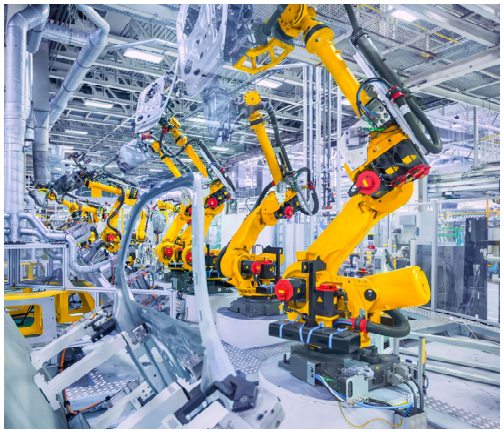
Like a regulator, does not exist in isolation, is combined with queue at next hop.

Unlike regulator, is stateless.

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

Low end-to-end jitter

- Many time sensitive applications require low latency only:
  e.g. 50ms for AVB (video processing);
  e.g. $\leq$1 ms Tactile Internet.
- Some applications may require low latency + very low jitter:
  e.g. remote process control requires latency bound 1ms, delay variation bound 1μs; latency bound alone is not sufficient for some machine control applications.
  [ITU-T 2020]



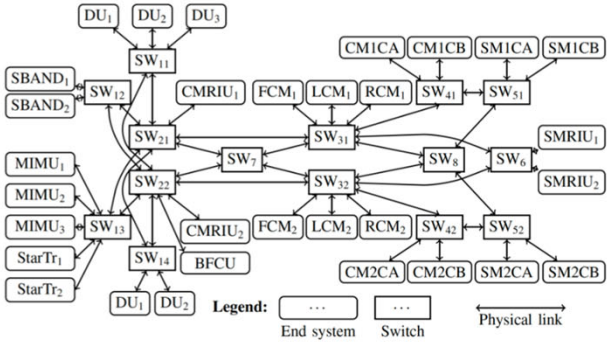
- Dampers can provide very low end-to-end jitter.

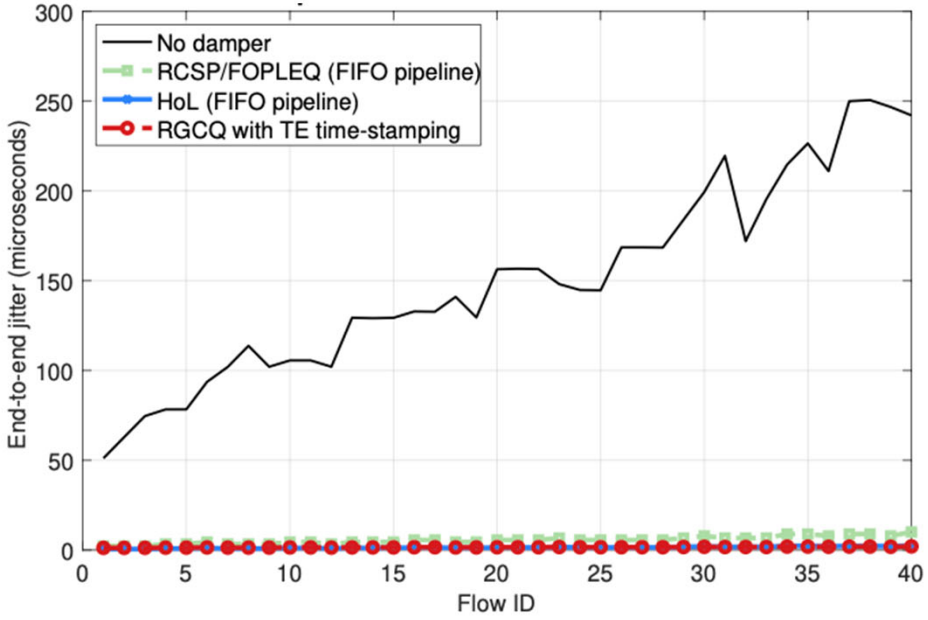
- Modelling clock accuracy [Thomas 2020] matters for very low jitter.

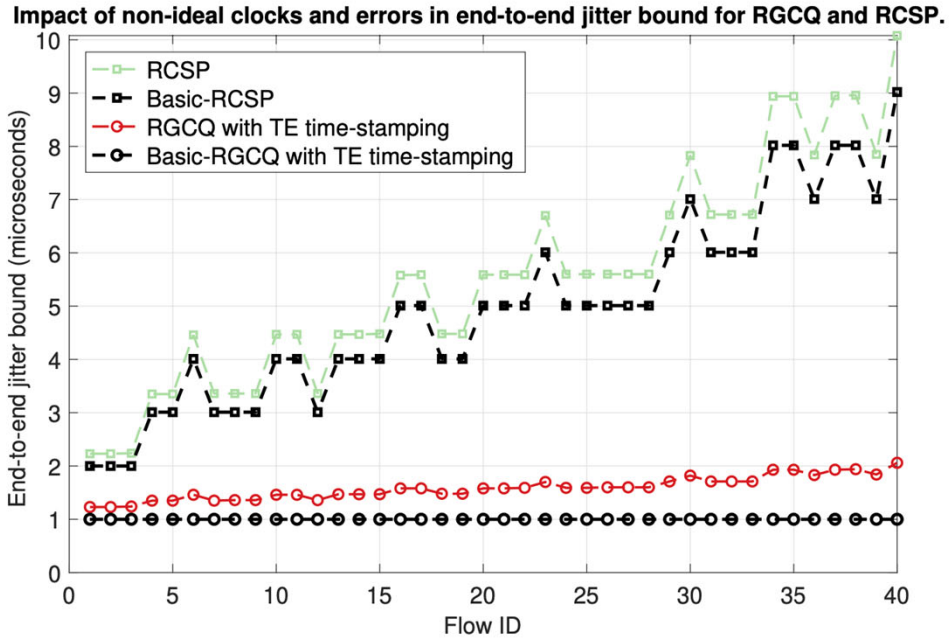
Analysis of Dampers with Realistic Clock Model

[Mohammadpour 2022]

incorporates measurement and clock errors
using the network calculus methods shown above





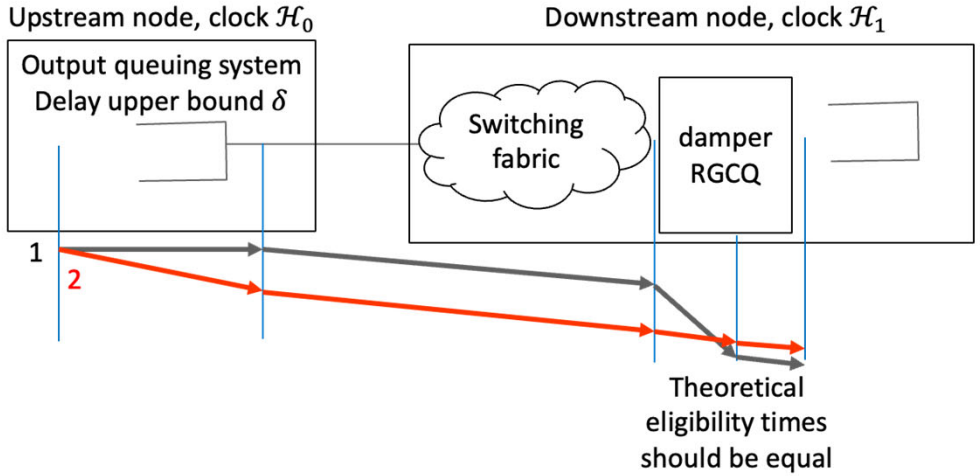


Dampers may cause re-ordering

[Mohammadpour 2022]

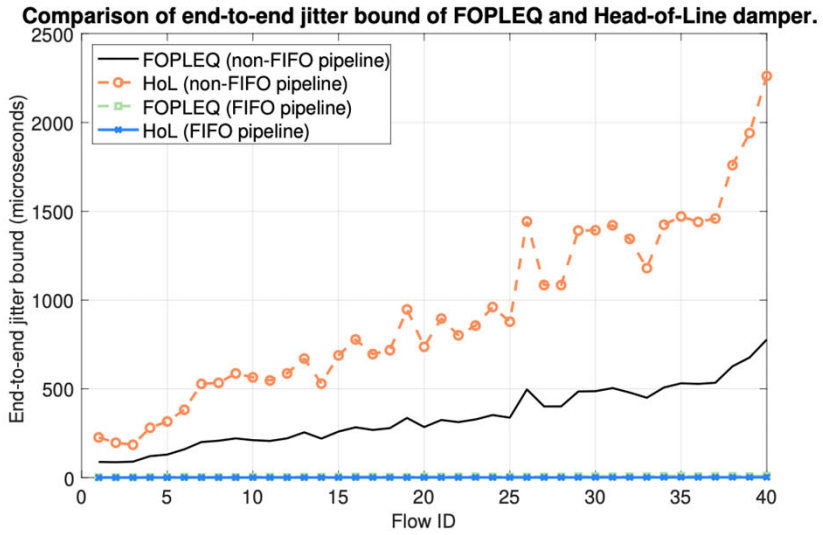
Two consecutive packets should be released by damper at ca. the same time.

Timing inaccuracies may lead to mis-ordering.

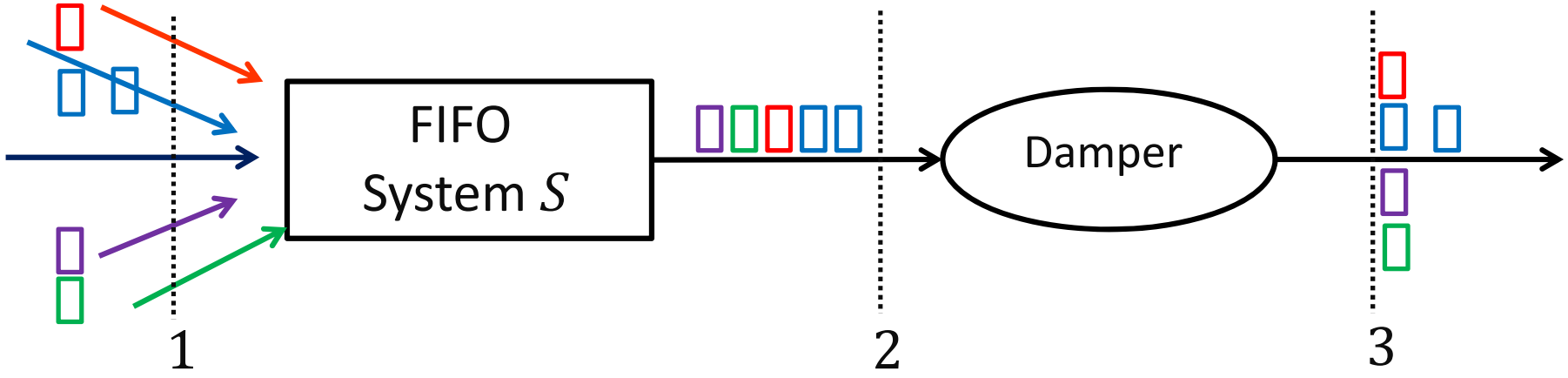


⇒ Some dampers enforce per-flow packet order (e.g. FOPLEQ, ATS with Jitter Control [Grigorjew 2020]).

- Avoids misordering
- Counterproductive if some network elements are non-FIFO



Dampers solve Burstiness Cascade Problem



Delay jitter $1 \rightarrow 3$ is 0 in theory; in practice, a small residual jitter $\leq \Delta$ (in true time):

At 1: assume a flow is constrained by token bucket with rate r and burstiness b (in true time);

$\Rightarrow$ At 3, same flow is constrained by token bucket with rate r and burstiness $b + r\Delta$ (in true time).

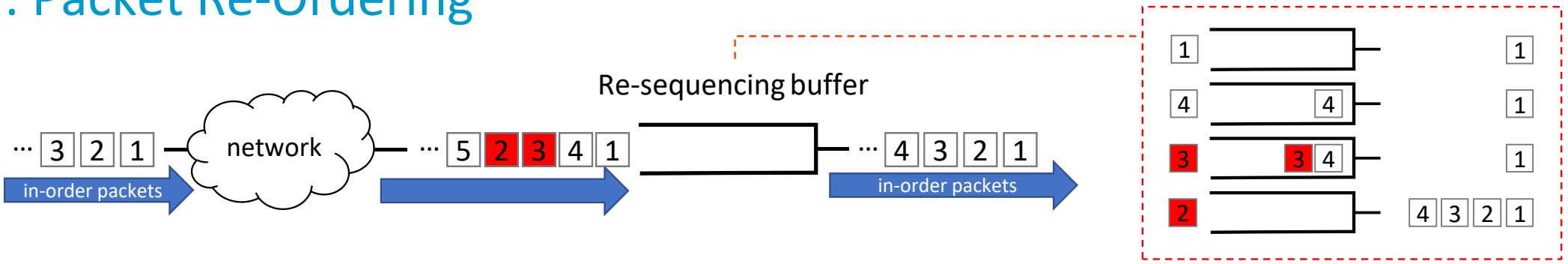
Tolerance Δ depends on jitter implementation and not on traffic
$\Rightarrow$ no burstiness cascade.

Dampers are non work-conserving devices that can dramatically reduce delay jitter

Stateless, unlike regulators

Like regulators, solve burstiness cascade and simplify network analysis

7. Packet Re-Ordering



Time Sensitive Networks may cause packet re-ordering due to e.g. parallel paths in switching fabrics.

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]

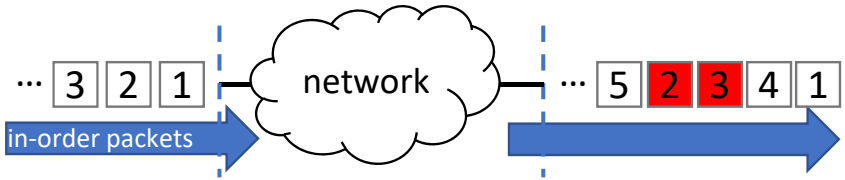
Defined between two observation points, and for a flow of interest

First observation point defines the reference order of packets

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

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

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



	E_n (μs)	λ_n (μs)
1	5	0
4	10	0
3	12	2
2	18	8
5	20	0

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

$A_j \leq A_n \iff j < n$

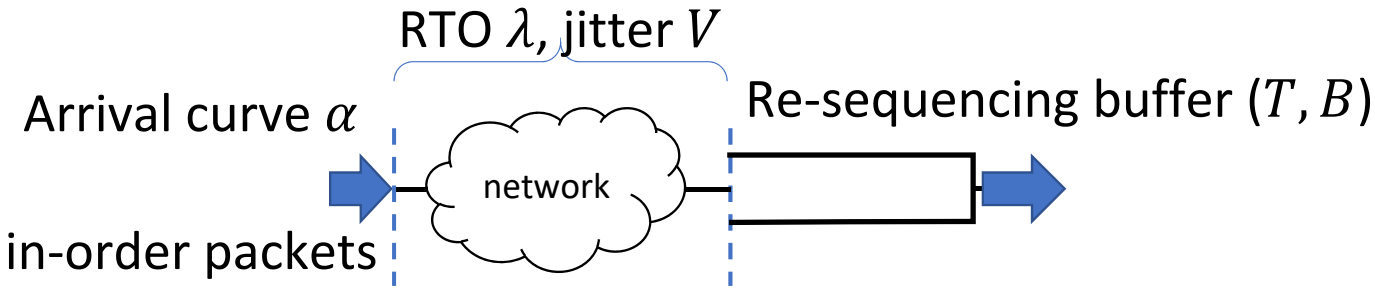
A_n: arrival time of packet n

E_n: exit time of packet n

$E_n = +\infty$, if packet n is lost

RTO $= 0$ means no re-ordering

Resequencing Buffer Calculus



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

[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 α is $\lambda = \left[V - \alpha^{\downarrow}\left(2L^{\min}\right)\right]^{+}$.

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

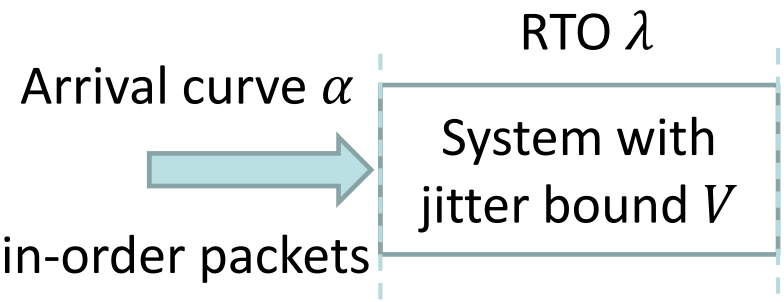
If $b < 2L^{\min}$ and $V \leq \frac{2L^{\min} - b}{r}$ then $\lambda = 0$ (no reorder)

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

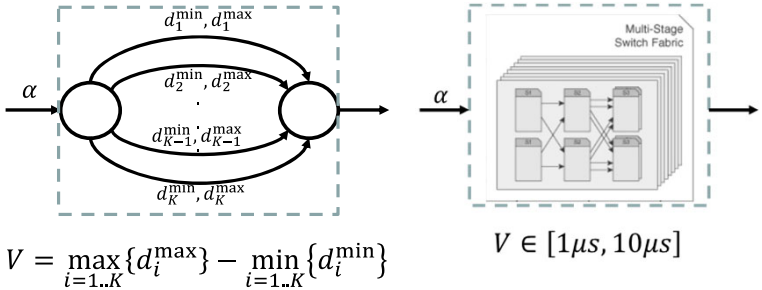
Other bounds exist for flow constraints at packet level.

[Mohammadpour 2021]

All bounds in TAI (temps atomique international)



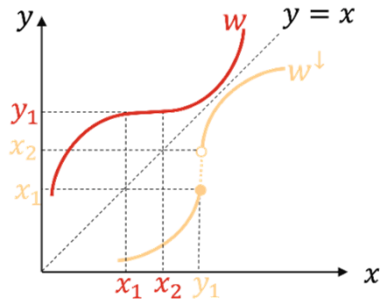
Examples of systems



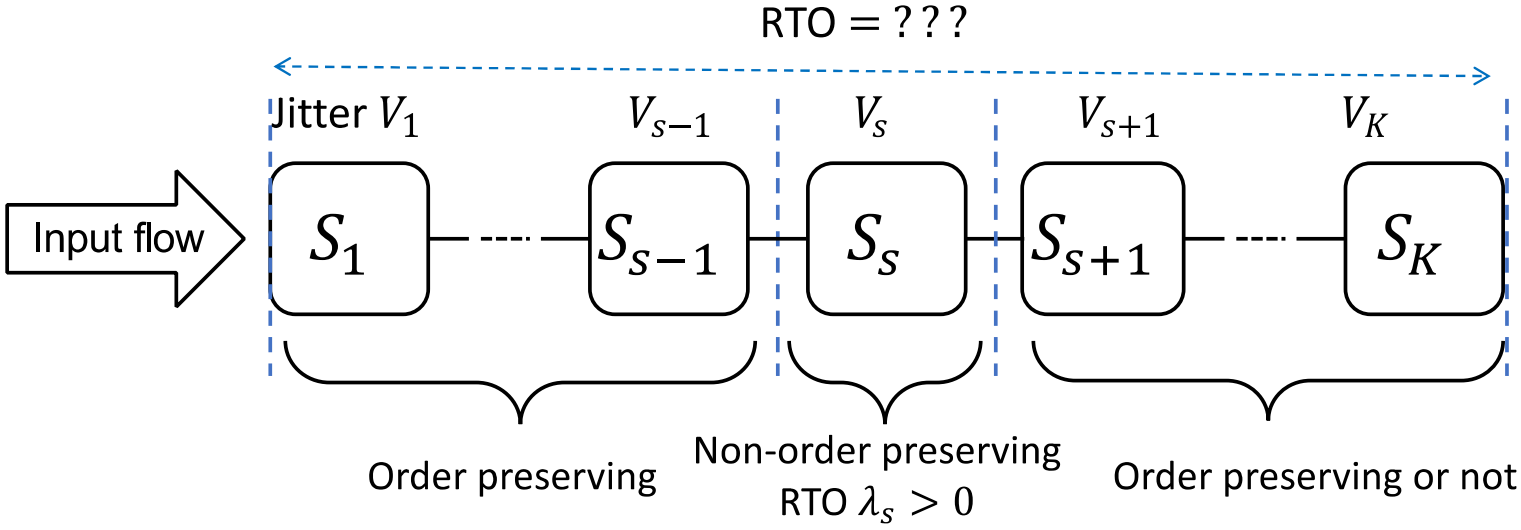
$V = \max_{i=1..K}\{d_i^{\max}\} - \min_{i=1..K}\{d_i^{\min}\}$

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

Lower pseudo-inverse



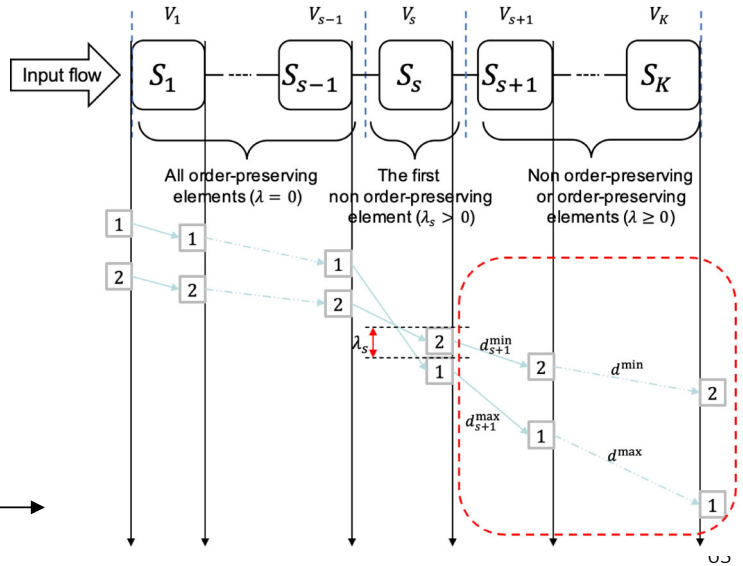
Calculus of RTO (2): Concatenation



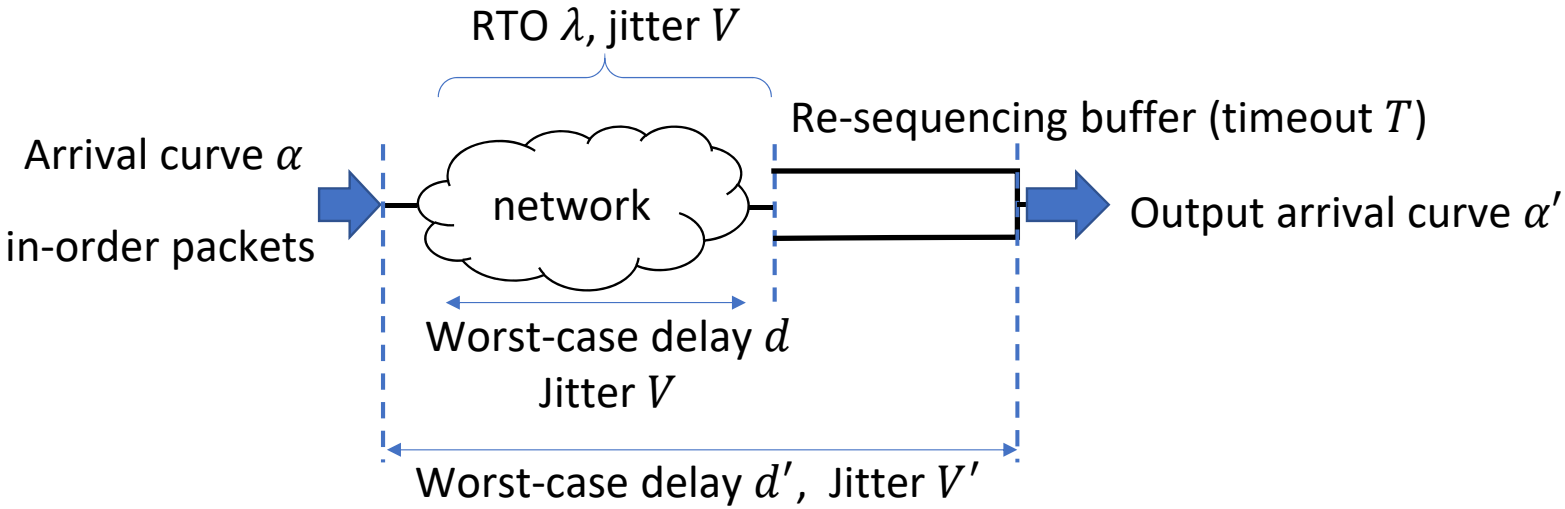
Best RTO bound for concatenation is

$$\Lambda(K) = \lambda_s + \sum_{i=s+1}^{K} V_i$$

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)

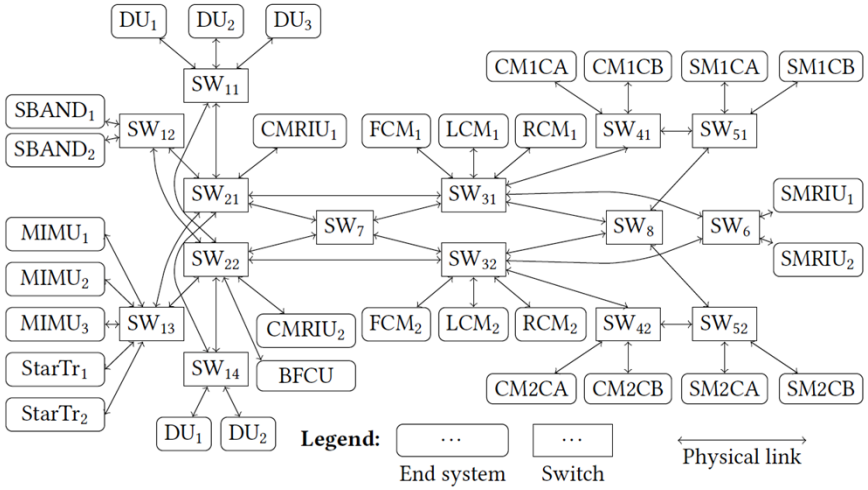
Lossy network:

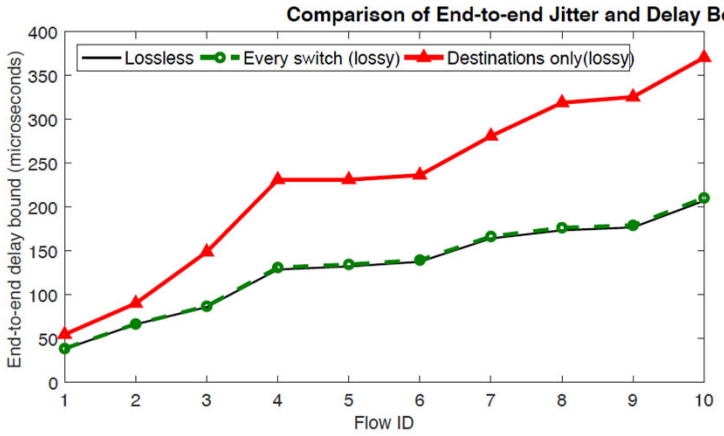
$d' = d + T, V' = V + T$ and $\alpha'(t) = \alpha(t + V + T)$.

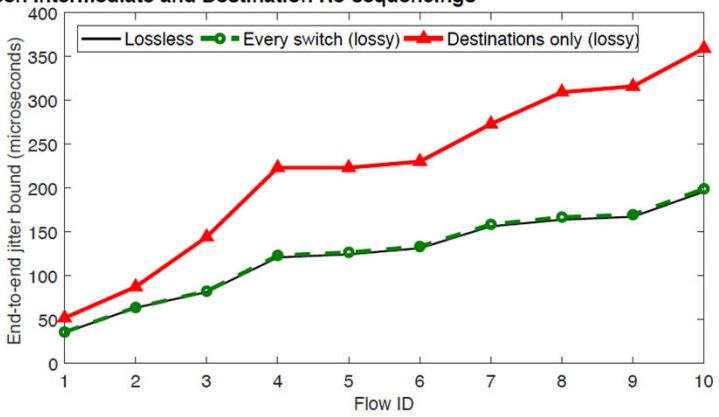
Example

Re-sequencing at destination vs at every switch increases end-to-end worst-case delay and jitter

[Mohammadpour 2021]







Packet reordering metric of interest is the RTO

Resequencing is for free in lossless networks

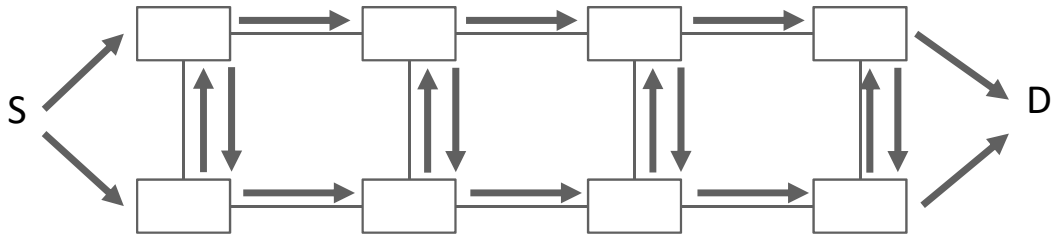
In lossy networks, timeout value (hence RTO) affects delay, jitter and propagated burstiness

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

8. Packet Replication

Deterministic networks guarantee 0 congestion loss, but other losses may occur (transient failures, reboots, transmission errors).

This is mitigated with packet replication and duplicate removal.



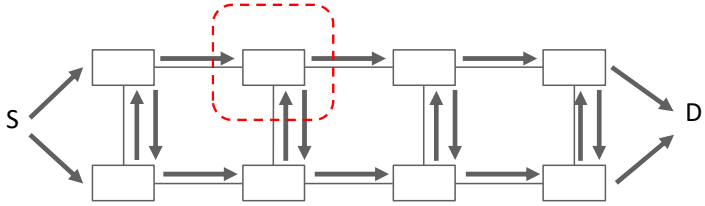
Any combination of failures that leaves at least one path up is masked (“0 msec repair”) [IEEE 802.1CB]

FRER: Frame Replication and Elimination for Reliability (IEEE TSN)

PREOF: Packet Replication Elimination and Ordering Function (IETF Detnet)

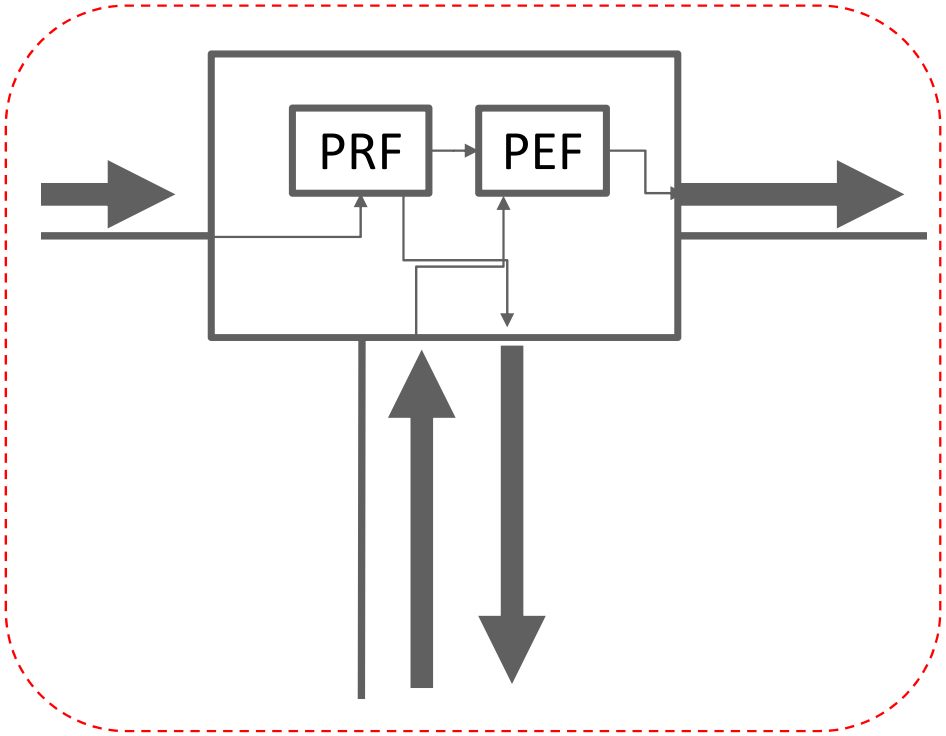
Packets are duplicated at sources and at intermediate points. Packet duplicates are removed at intermediate points and at destination.

Packet Elimination Function



Packet duplicates are eliminated (Packet Elimination Function, PEF).

Packet replication function (PRF) multicasts n copies towards destination (here $n = 2$).

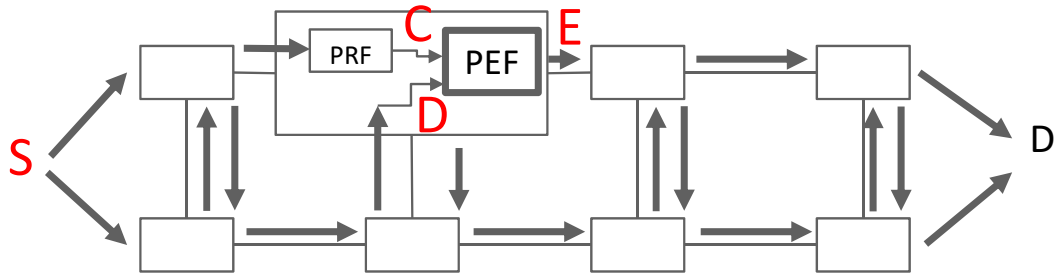


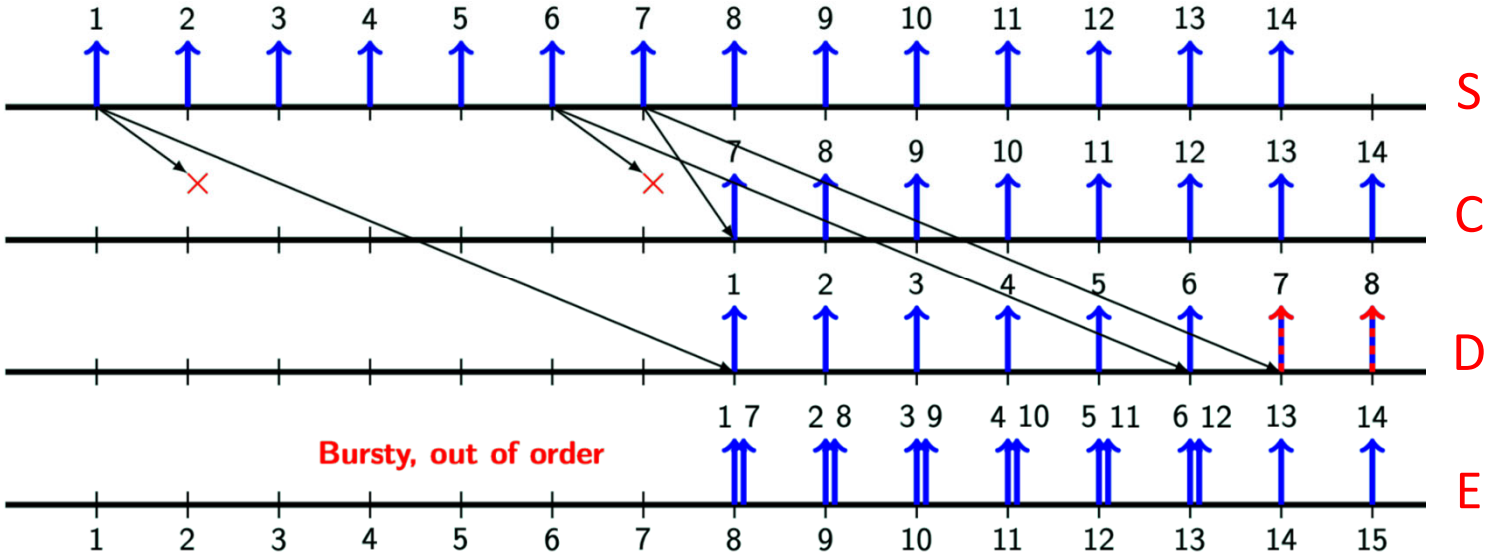
Question: effect on delay analysis ?

PRF simply generates cloned flows – business as usual

PEF requires special analysis

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.

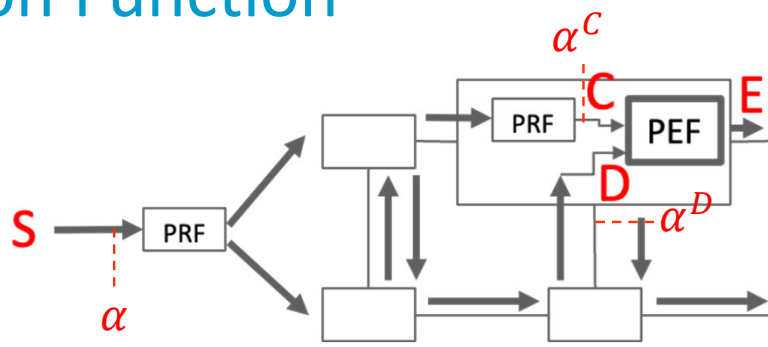
At exit of Packet Elimination Function:

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

Network Calculus Analysis of Packet Elimination Function

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

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



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

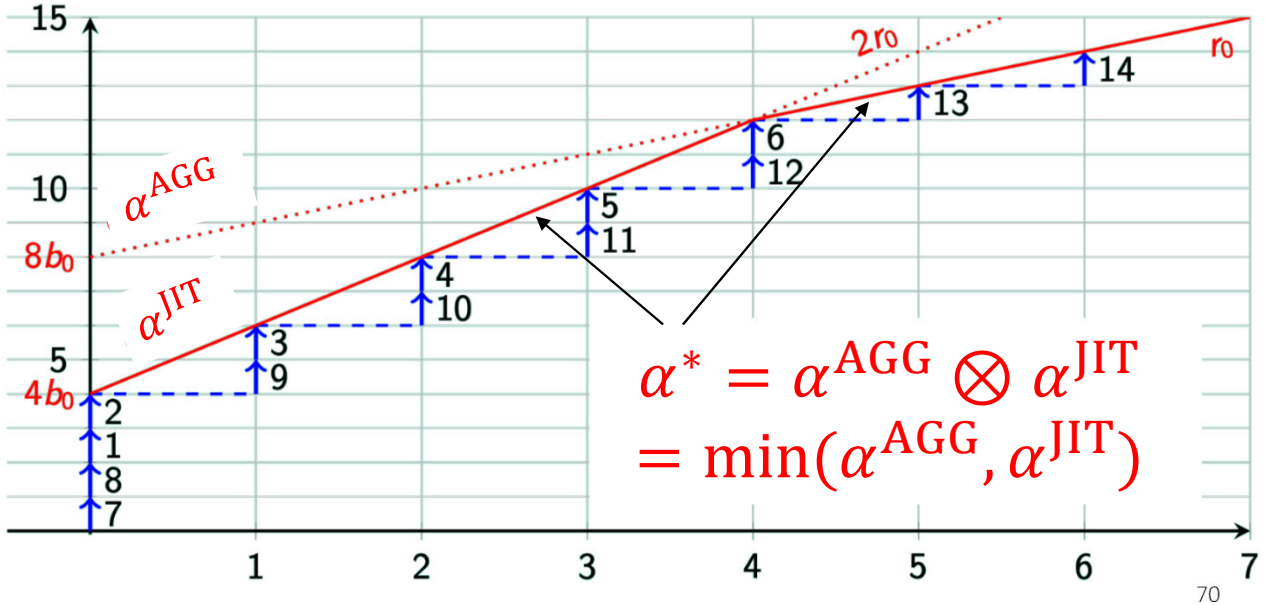
Delay from S to (C or D) $\in [D^{\min}, D^{\max}]$

Bound on RTO (amount of re-ordering)

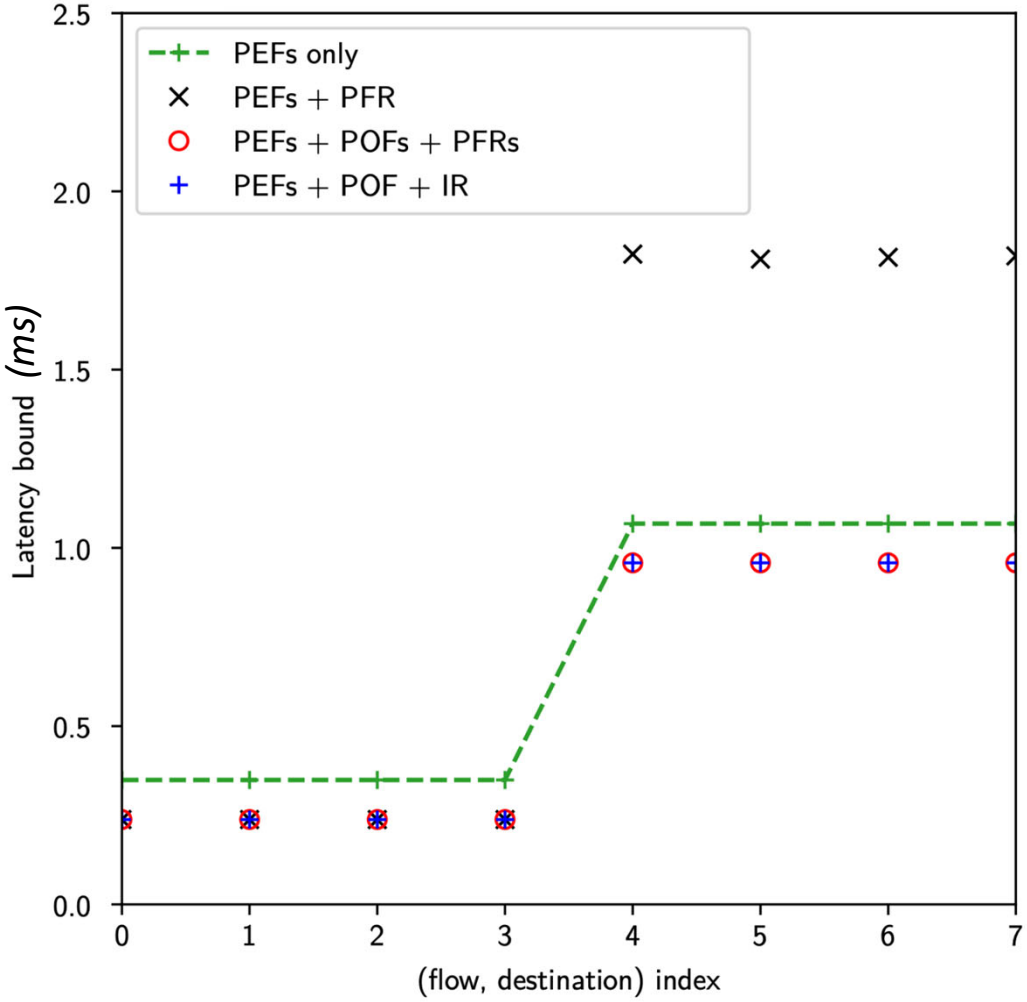
$$\lambda = \left[D^{max} - D^{\min} - \alpha^{\downarrow}(2L^{\min})\right]^+$$

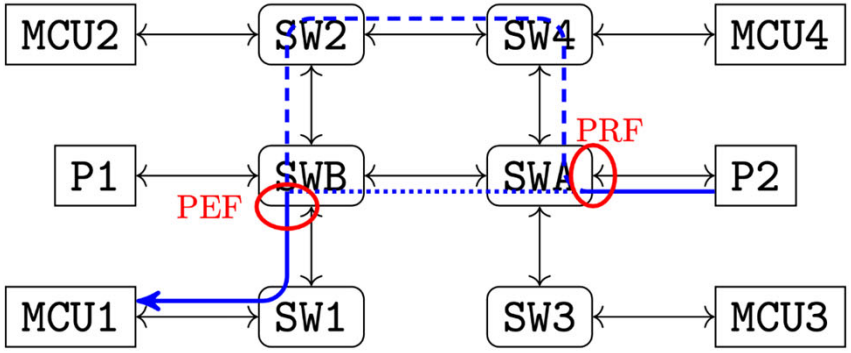
Network Analysis implemented in extension of TFA (xTFA).

[Thomas 2022]



Example





- PEF can be complemented with a re-sequencing buffer and a regulator to mitigate its negative impact
- PEF and regulator without resequencing buffer is worse / unstable

[Thomas 2022]

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

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

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.
- [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 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://orbilu.uni.lu/bitstream/10993/46282/1/AEC2020-UL-Volvo-RTaW-web.pdf
- [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.
- [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*.
- [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.