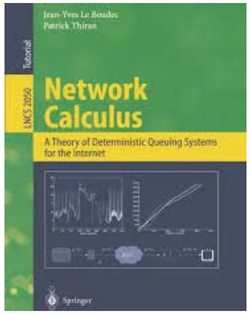
EPFL

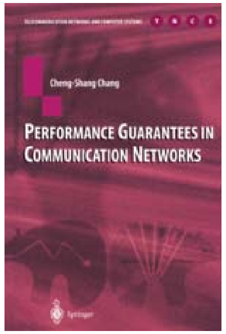
An Introduction to Network Calculus

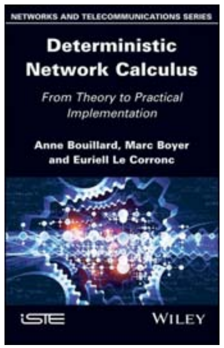
Jean-Yves Le Boudec
2019 April 5

Contents

What is Network Calculus ?







A theory and tools to compute bounds on queuing delays, buffers, burstiness of flows, etc.

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

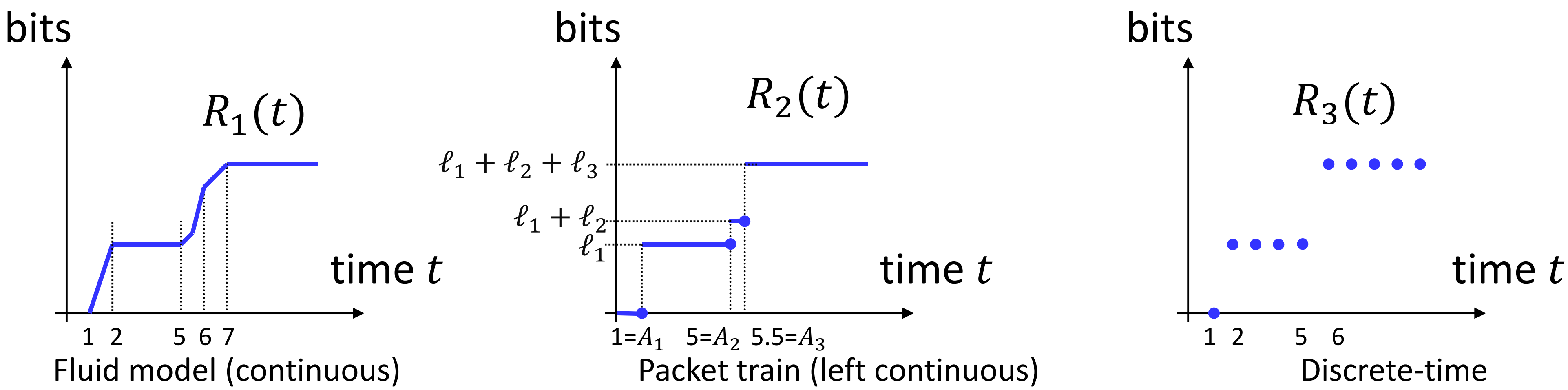
For deterministic networking, per-flow and per-class queuing

Derive system equations ⇒ formal proofs

Stochastic extensions exist (not discussed here)

1. Representation of Data Flow

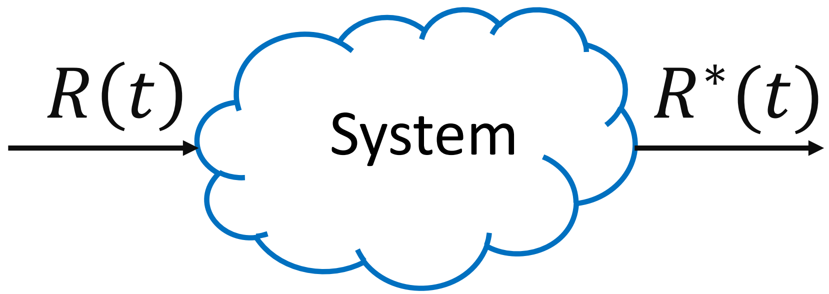
Cumulative flow: $R(t)$, non-decreasing with $R(0) = 0$

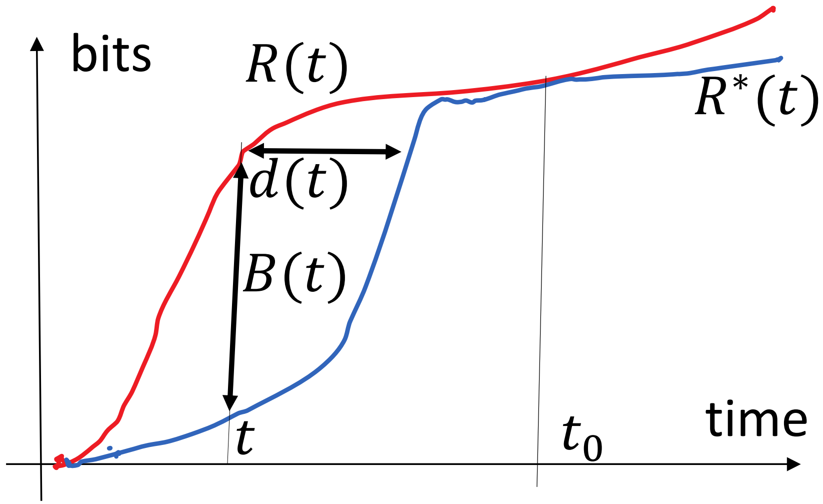


Daters: (A, L) with $A = (A_1, A_2, \dots)$ (dates) and $\ell = (\ell_1, \ell_2, \dots)$ (lengths in bits)

For a packet train: $R(t) = \sum_{n \geq 1} \ell_n 1_{\{A_n < t\}}$

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)

2. Arrival Curve

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

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

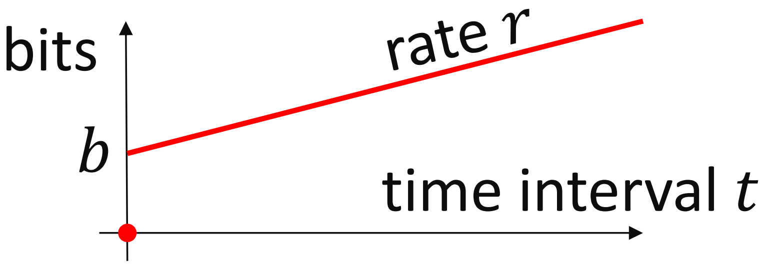
where α is a monotonic nondecreasing function $\mathbb{R}^+ \to [0, +\infty]$

token bucket constraint (r, b)

(leaky buket constraint)

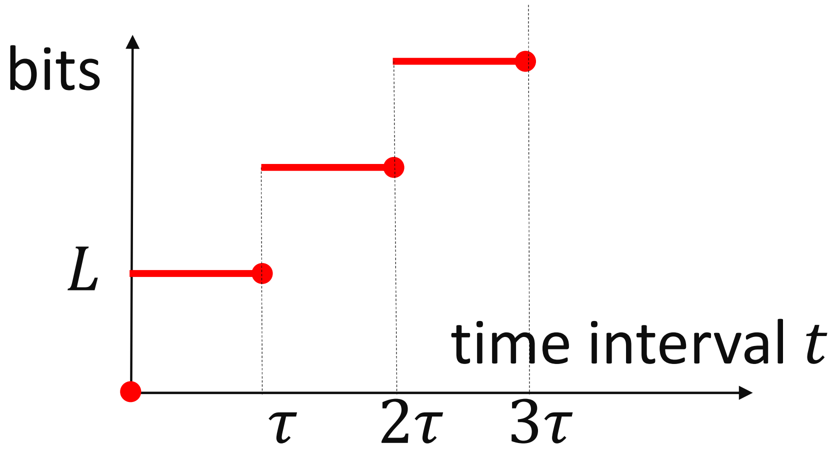
with rate r and burst b:

$\alpha(t) = rt + b$



[R. Cruz, PhD Dissertation 1987]

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



Aggregation Property

If every flow f has arrival curve α_f then the aggregation $R = \sum_f R_f$ has arrival curve $\sum_f \alpha_f$

If every flow f is token-bucket constrained (r_f, b_f) then the aggregation is token-bucket constrained $(\sum_f r_f\,,\ \sum_f b_f)$

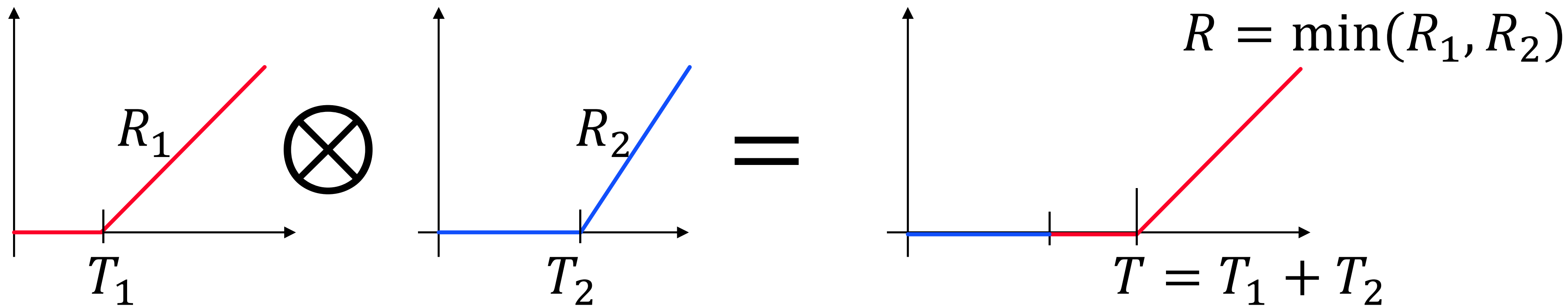
Min-Plus Convolution of $f_1, f_2 \geq 0$

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

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 easily: e.g.



Min-Plus Calculus

$\otimes$ is associative, commutative

Neutral element: $f \otimes \delta_0 = f$ where $\delta_0(0) = 0, \delta_0(t) = +\infty, t > 0$

$\otimes$ distributes w.r.t. min: $f \otimes (g \wedge h) = (f \otimes g) \wedge (f \otimes h)$

$\otimes$ is isotone: $f \leq f'$ and $g \leq g' \Rightarrow f \otimes g \leq f' \otimes g'$

Functions passing through the origin ($f(0) = g(0) = 0$):
$f \otimes g \leq f \wedge g$

Concave functions passing through the origin: $f \otimes g = f \wedge g$

Convex piecewise linear functions: $f \otimes g$ is the convex piecewise linear function obtained by putting end-to-end all linear pieces of f and g, sorted by increasing slopes

[Bouillard et al 2018, Chapters 3 and 4]

Min-Plus Convolution and Arrival Curves

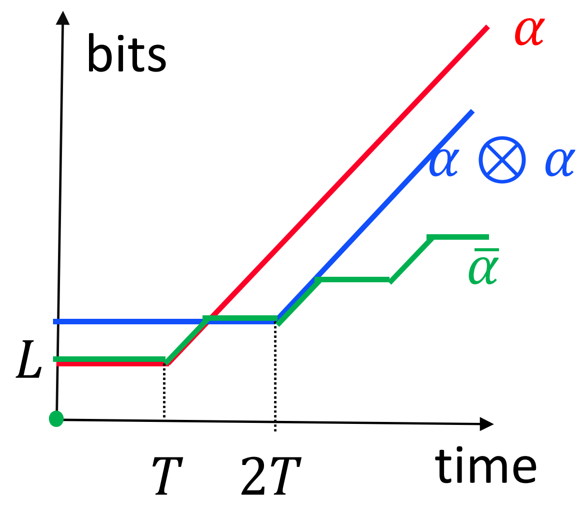
α is an arrival curve for $R \Leftrightarrow R(t) \leq R(s) + \alpha(t-s), \forall s \in [0,t]$

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

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



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

and $\bar{\alpha}(0) = 0$

when $\alpha(0) = 0$,

α is an arrival curve for $R \Leftrightarrow R = R \otimes \alpha$

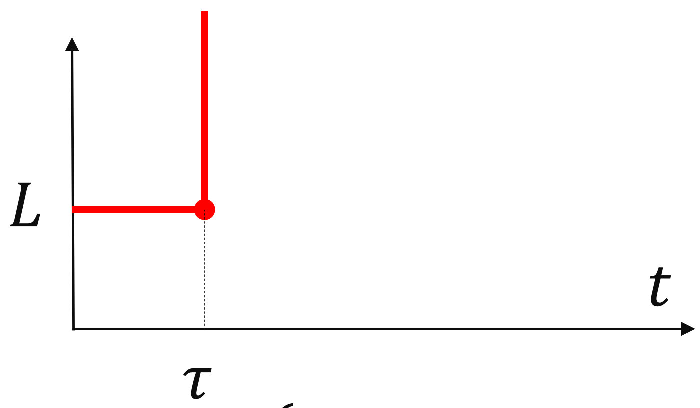
Example of Sub-Additive Closure

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

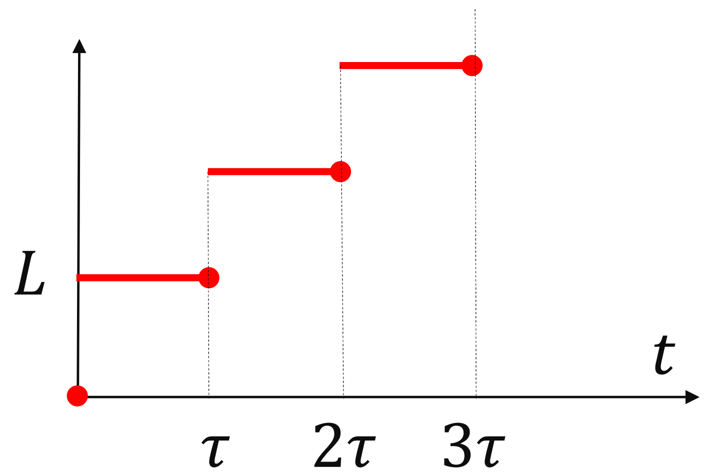
$\Leftrightarrow R(t+\tau) - R(t) \leq L$ for all t

$\Leftrightarrow$ flow has arrival curve α

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

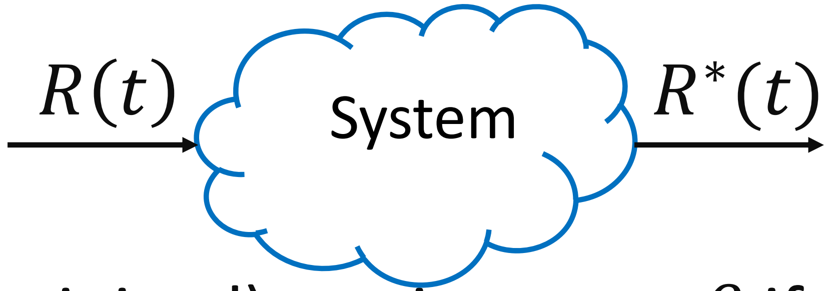


$$\alpha(t) = \begin{cases} L, t \leq \tau \\ +\infty, t > \tau \end{cases}$$



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

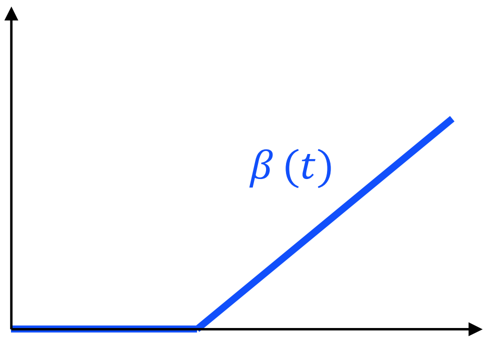
3. Service Curve

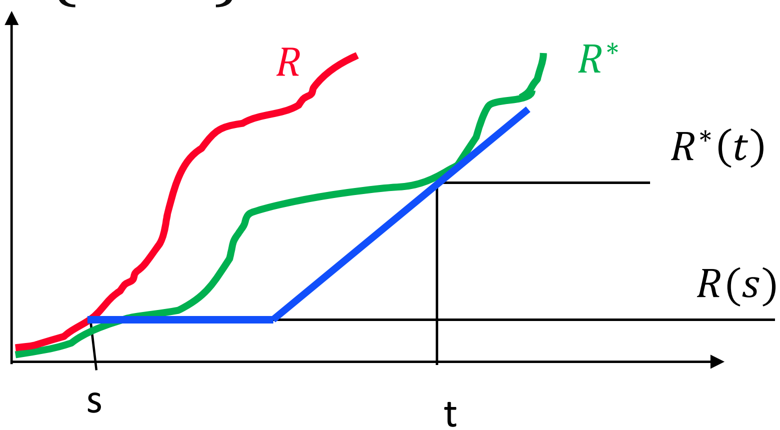


System offers to this flow a (minimal) service curve β if $R^* \geq R \otimes \beta$, i.e. :

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

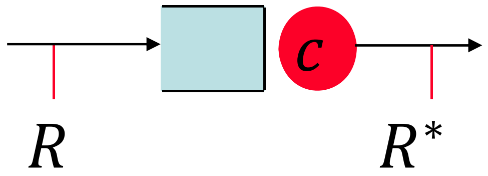
where β is a function : $\mathbb{R}^+ \to \mathbb{R} \cup \{+\infty\}$

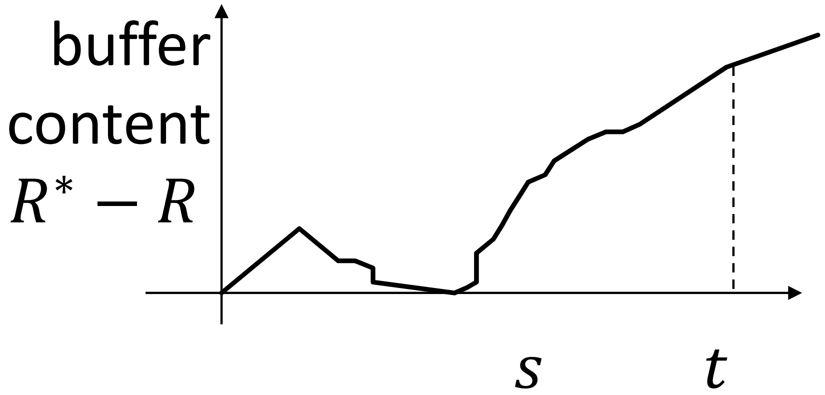


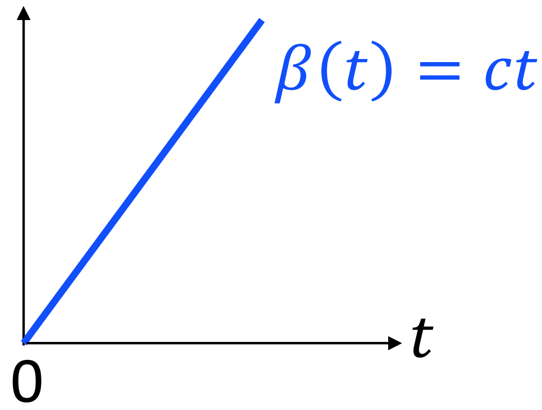


[Le Boudec 96, Chang 97, Bouillard et al 2018]

The constant rate server offers service curve $\beta(t) = ct$







Proof: take s = beginning of busy period:

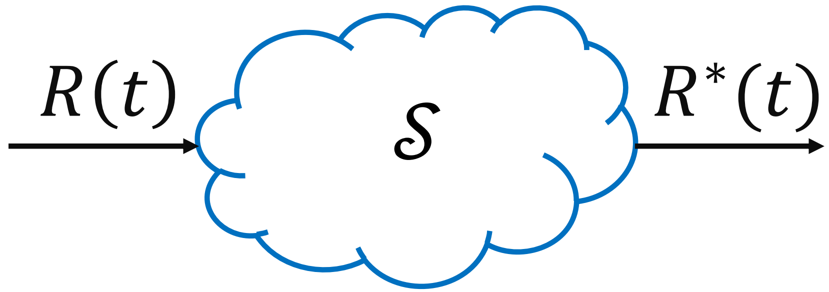
$$R^*(t) - R^*(s) = c\,(t - s) \text{ and } R^*(s) = R(s)$$

$$\Rightarrow R^*(t) - R(s) = c\,(t - s)$$

$\beta(t) = ct$ is a service curve; it is also a *strict* service curve

[Cruz 95]

Strict Service Curve

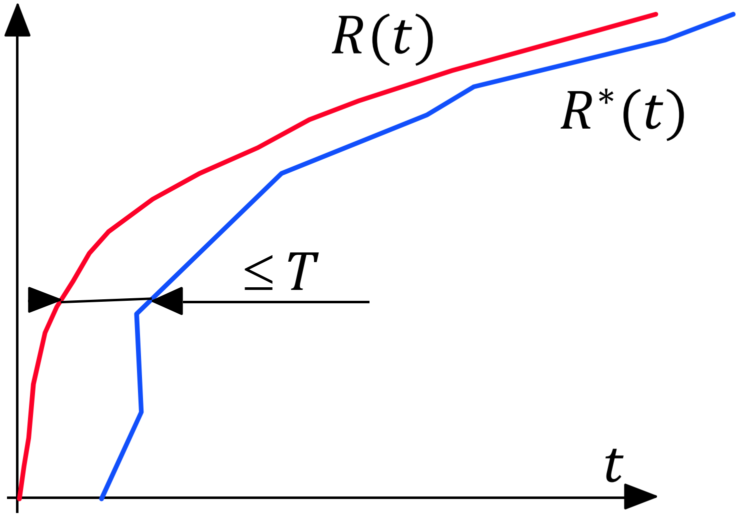


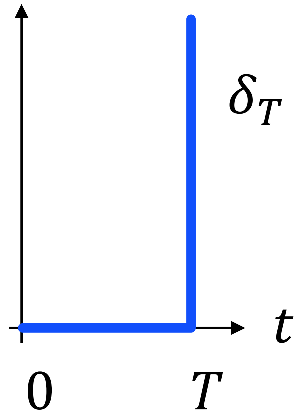
System $\mathcal{S}$ offers to a flow a strict service curve β if for any $s < t$ inside a backlogged period, i.e. such that $R^*(u) < R(u), \forall u \in (s, t]$, we have $R^*(t) - R^*(s) \geq \beta(t - s)$

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

strict service curve $\Rightarrow$ service curve

The guaranteed-delay node offers service curve δ_T





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

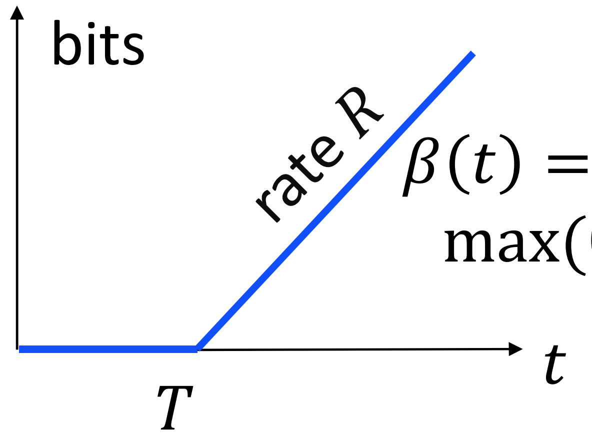
For a node that is FIFO for this flow:

delay $\leq T$ $\Leftrightarrow$ nodes offers to this flow a service curve δ_T

Service Curve Examples

Rate-latency service curve :

$$\beta(t) = R(t-T)^+$$



$\beta(t) = R(t-T)^+ =$
$\max(0, R(t-T))$

Example: Static Priority without pre-emption, fixed line rate C

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 et al 2018]

Service Curve Example: Deficit Round Robin

Popular per-flow scheduler, one queue per flow [Shreedhar and Varghese, 1995]

Visit every queue in sequence and at every visit, serve up to Q_i bits.

DRR offers to flow i a strict service curve:

$$\beta_i(t) = R_i(t - T_i)^+$$

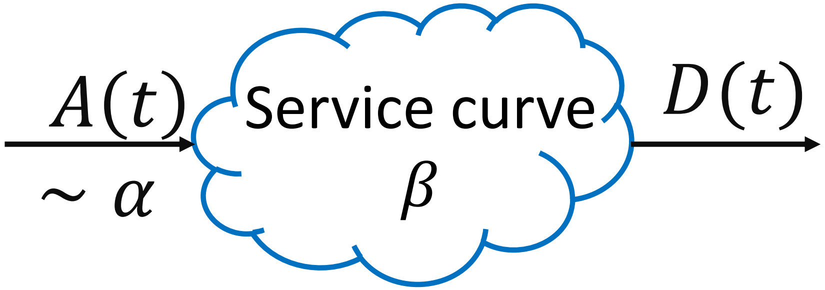
with $R_i = \frac{Q_i}{\sum_j Q_j} c, \quad T_i = \frac{\bar{Q}_i + \bar{L}_i}{c} + L_{\max,i}(\frac{1}{R_i} - \frac{1}{c}),$

$\bar{Q}_i = \sum_{j \neq i} Q_j, \ \ \bar{L}_i = \sum_{j \neq i} L_{\max,j}$ and c is the line rate.

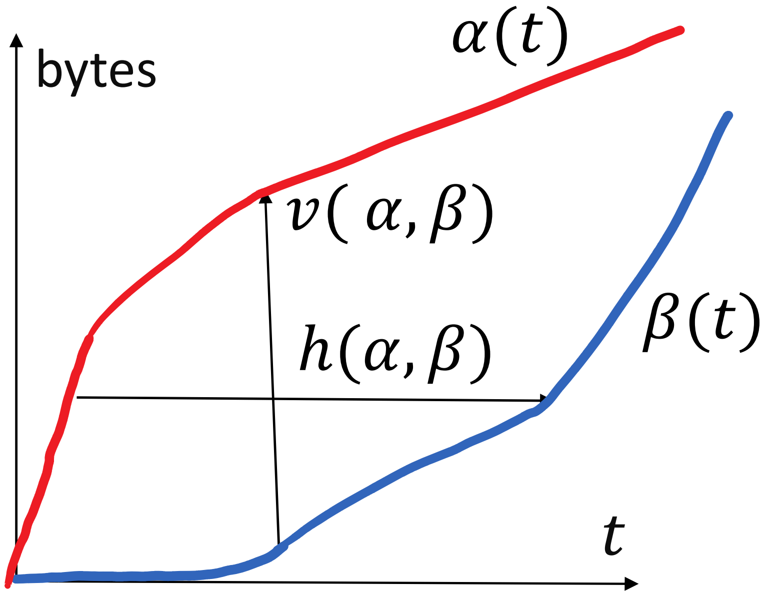
[Boyer et al 2012]

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

Three Tight Bounds

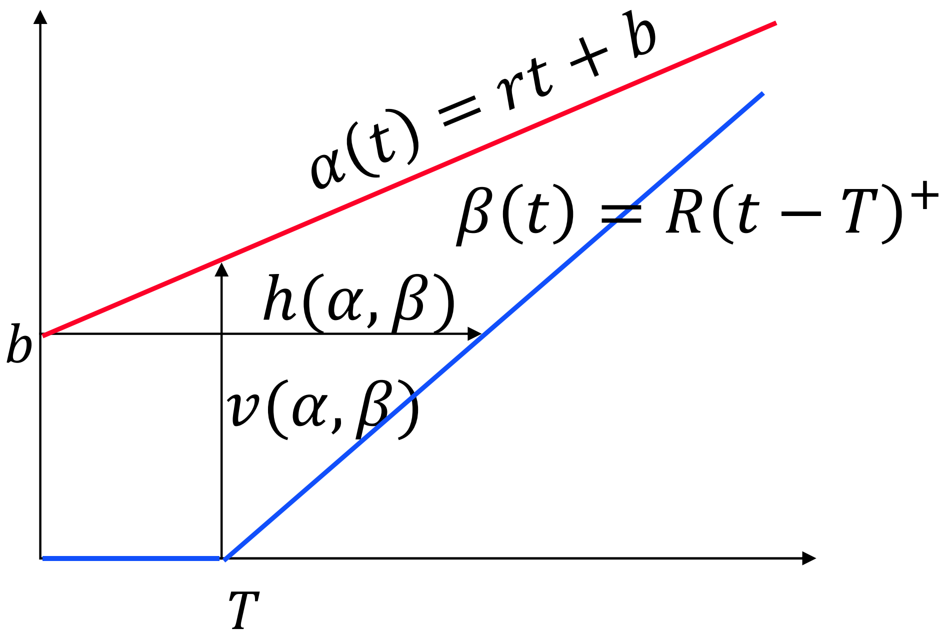


Flow is constrained by arrival curve α; served in network element with service curve β. Then



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)

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: $b + rT$

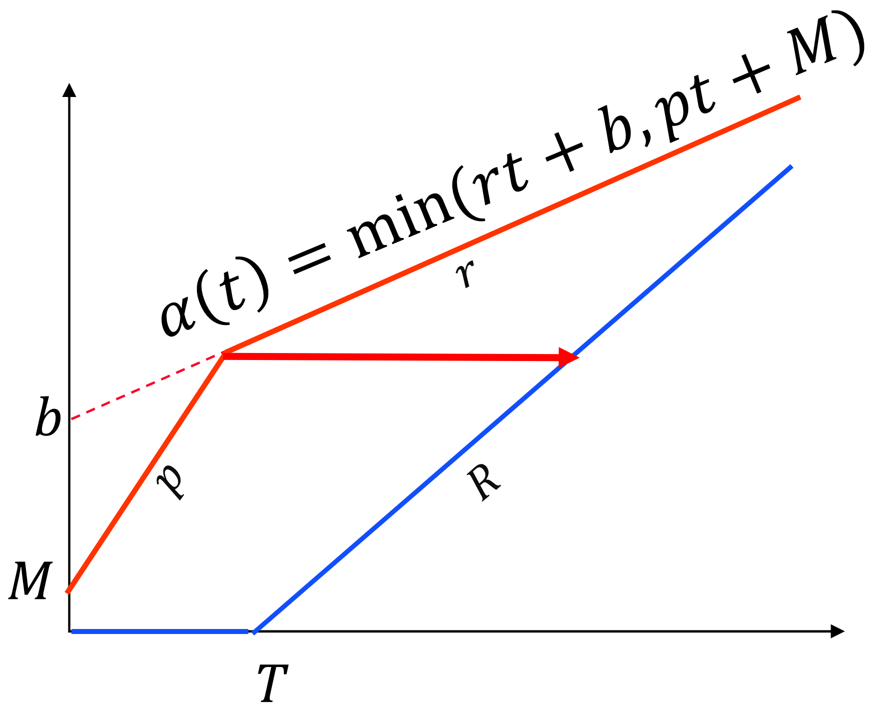
Delay bound: $\frac{b}{R} + T$

Output arrival curve:

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

with $b^* = b + rT$

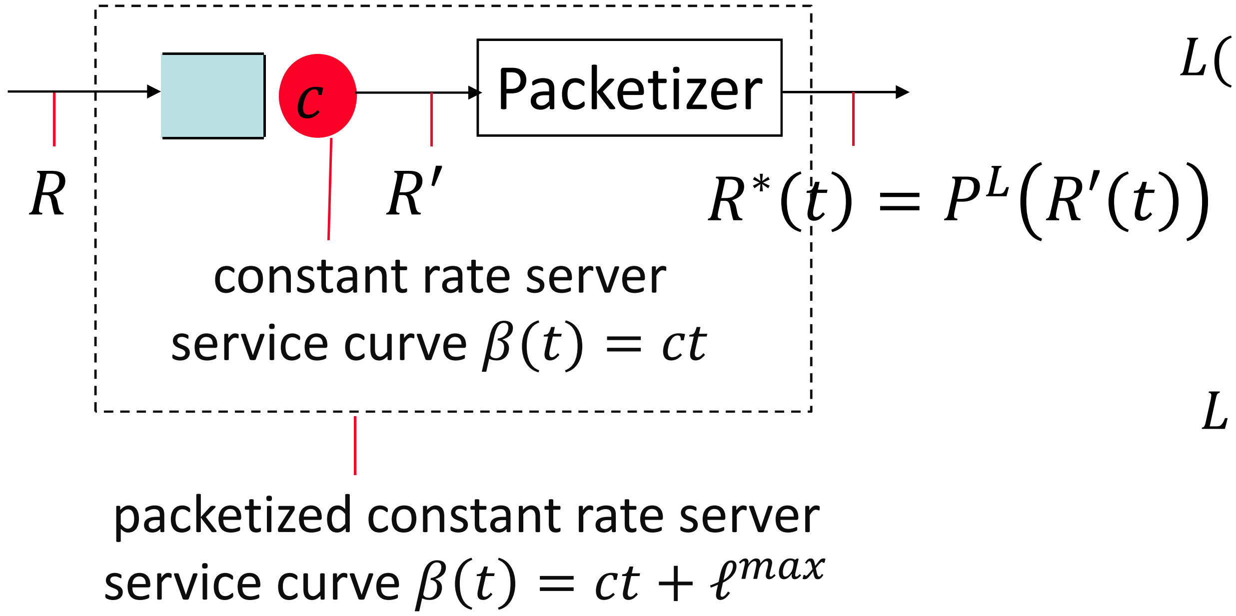
Example with peak rate limit

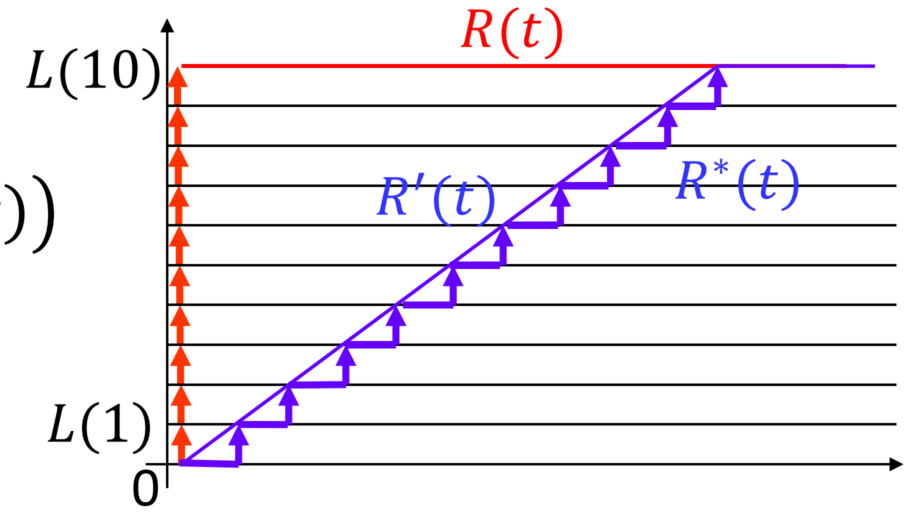


Assume $r \leq R$

Delay bound: $\dfrac{M + \frac{b-M}{p-r}(p-R)^{+}}{R} + T$

4. Packetizer





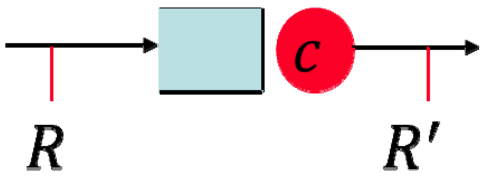
Packetization is captured by the function P^L: [Chang-Lin 1998]

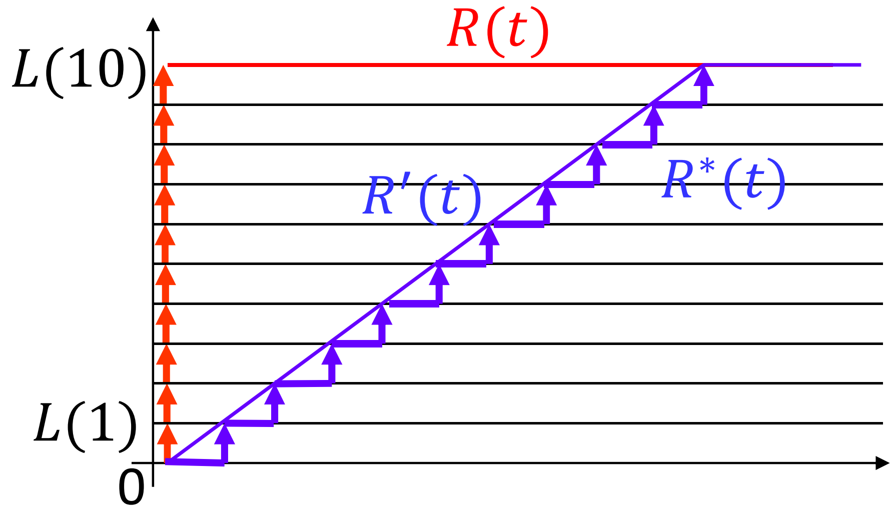
$$P^L(x) = L(n) \Leftrightarrow L(n) \leq x < L(n+1)$$

with $L(n) = \ell_1 + \ell_2 + \ldots + \ell_n$

Packetizer is the system $R \rightarrow P^L(R)$

Service Curve Definition Must Specify Packetization or Not

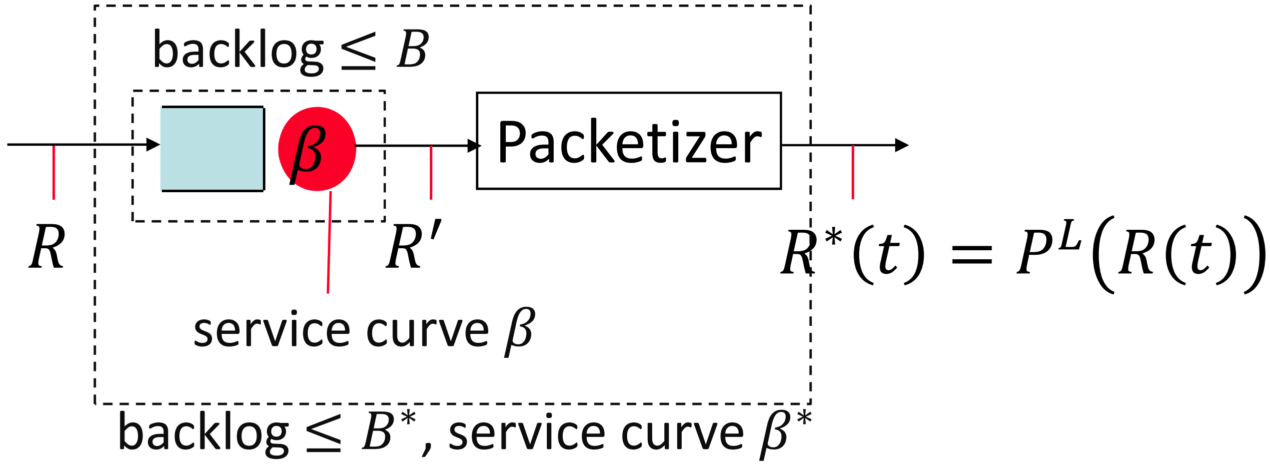




This system offers the service curve $\beta(t) = ct$

If we consider only emission times of entire packets, (i.e. the output is R^*), the system now offers the (weaker) service curve $\beta(t) = (ct - \ell^{max})^+$

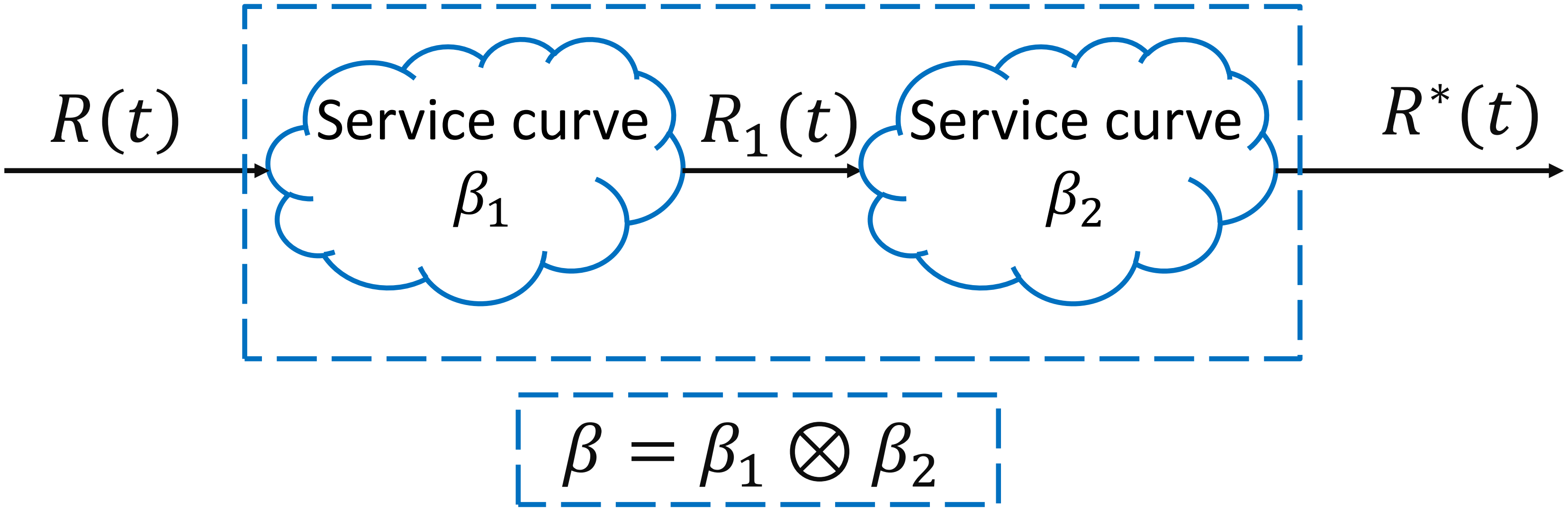
Effect of Packetization



1. Delay bounds are same for R' and R^*
2. Backlog bound $B^* = B + \ell^{max}$
3. Service curve $\beta^*(t) = [\beta(t) - \ell^{max}]^+$
4. If a fluid flow $R(t)$ has arrival curve $\alpha(t)$ then the packetized flow $P^L(R(t))$ has arrival curve $\alpha(t) + \ell^{max}$

$\ell^{max} = \max_n \ell_n$

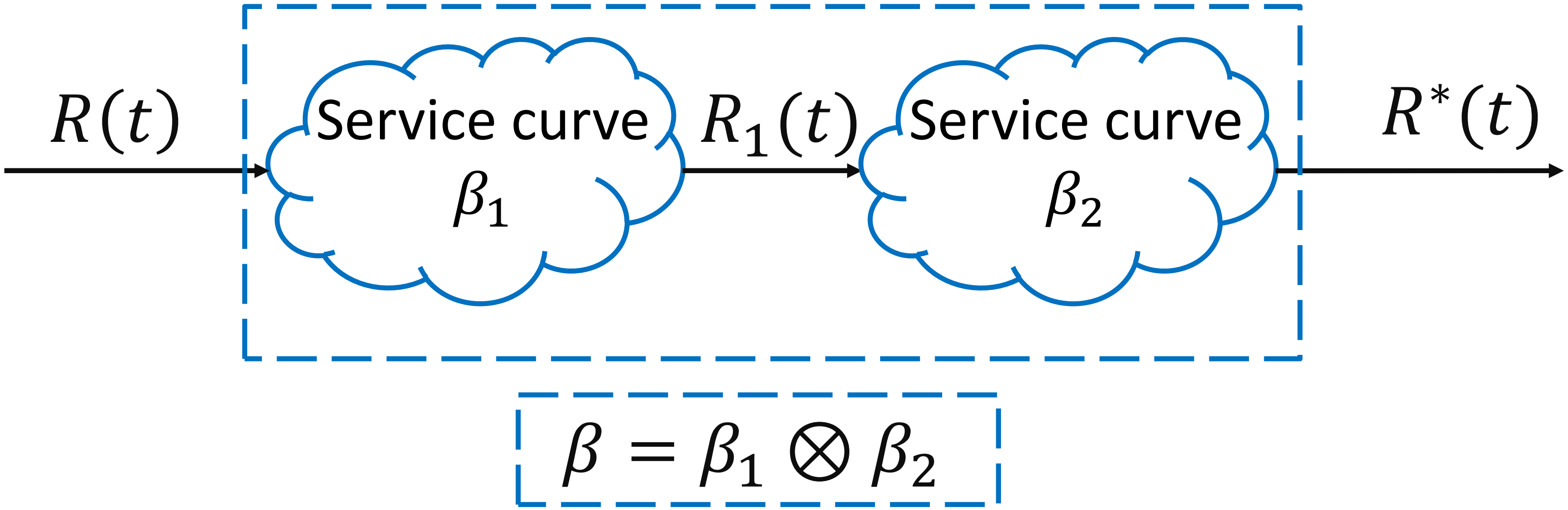
5. Concatenation



A flow is served in series, network element i offers service curve β_i.
The concatenation offers to flow the service curve $\beta = \beta_1 \otimes \beta_2$

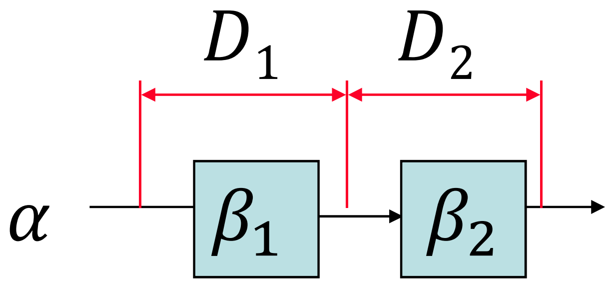
Proof: $R^* \geq R_1 \otimes \beta_2 \geq (R \otimes \beta_1) \otimes \beta_2 = R \otimes (\beta_1 \otimes \beta_2)$

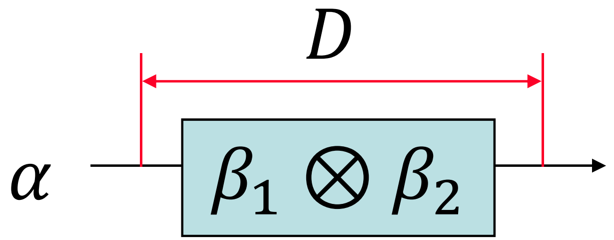
Example



If β_i is rate-latency R_i, T_i then the concatenation β is rate-latency $R = \min(R_1, R_2)$ and $T = T_1 + T_2$

Pay Bursts Only Once





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

one flow constrained *at source* by α

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

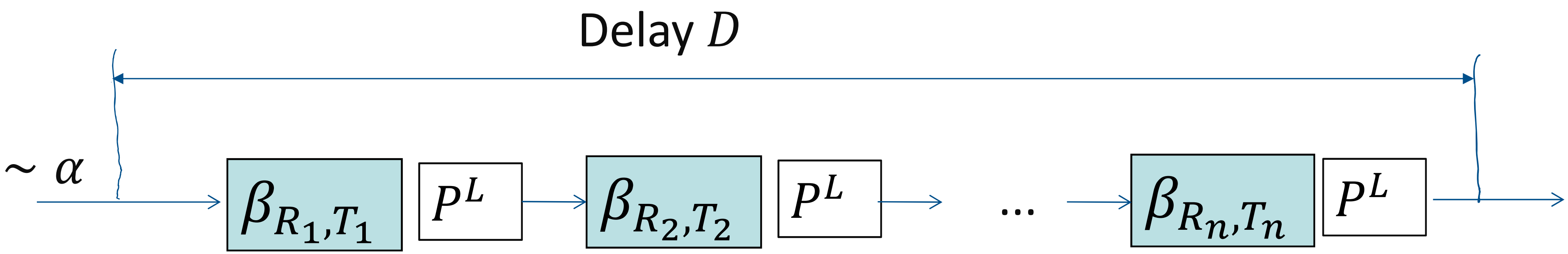
$$D_1 + D_2 = \frac{2b + rT_1}{R} + T_1 + T_2$$

computed *by concatenation:*

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

i.e. worst cases cannot happen simultaneously – concatenation captures this !

Packetization and Concatenation



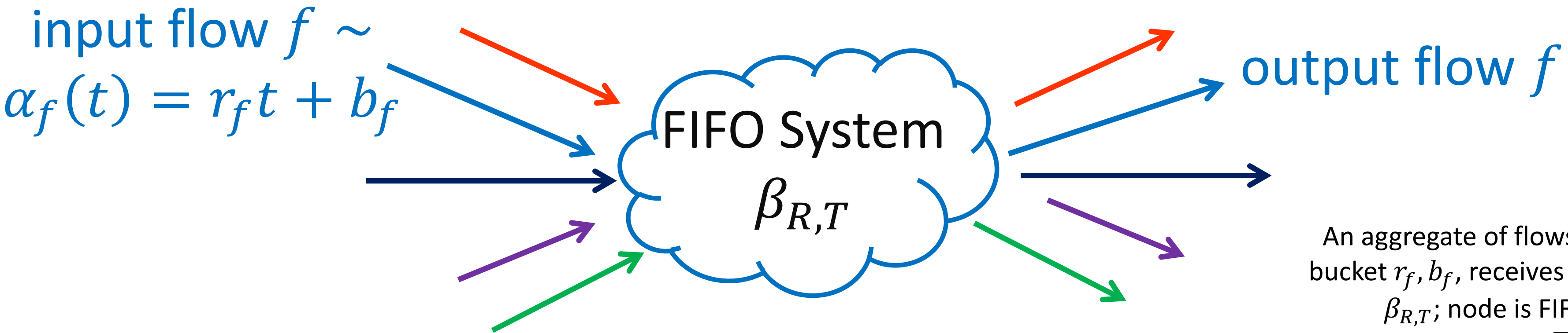
To compute delay bound D, use end-to-end service curve ignoring last packetizer

$$D \leq h\left(\alpha, \beta_{R,T}\right)$$

with $R = \min_{i=1\ldots n} R_i$ and $T = \sum_{i=1}^{n} T_i + (n-1) R_i \ell^{max}$

Cannot ignore the intermediate packetizers.

6. Per-Class FIFO Networks



An aggregate of flows, each constrained by leaky bucket r_f, b_f, receives a rate-latency service curve $\beta_{R,T}$; node is FIFO for the aggregate; $r_{tot} = \sum_{f \in S} r_f \leq R,\ b_{tot} = \sum_{f \in S} b_f$

Per-class FIFO networks use schedulers to separate classes

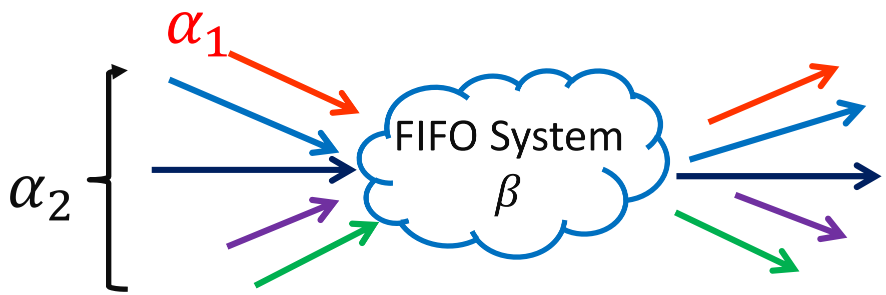
Service received by a class can be modelled by a rate-latency service curve $\beta_{R,T}$ (e.g. : DRR, AVB, TSN) [Boyer et al 2012, De Azua – Boyer 2014].

delay bound for any packet of any flow: $D = \frac{b_{tot}}{R} + T$

backlog bound for the aggregate of all flows: $B = b_{tot} + r_{tot}T$

[Le Boudec-Thiran 2001, Section 6.4] [Bouillard et al 2018, Chapters 10,11]

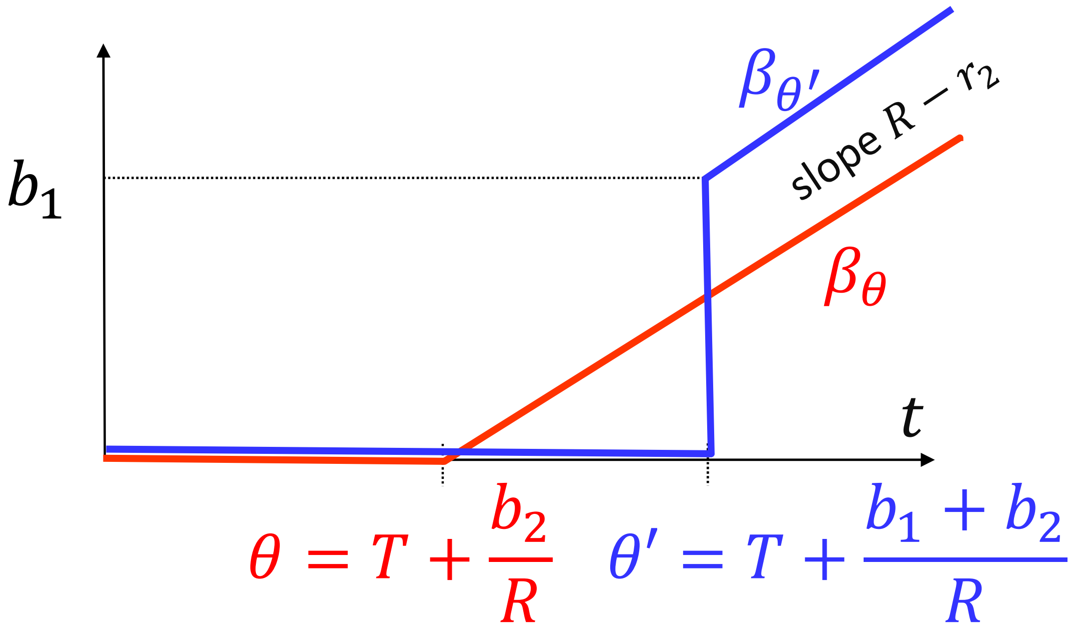
FIFO Residual Service Curve



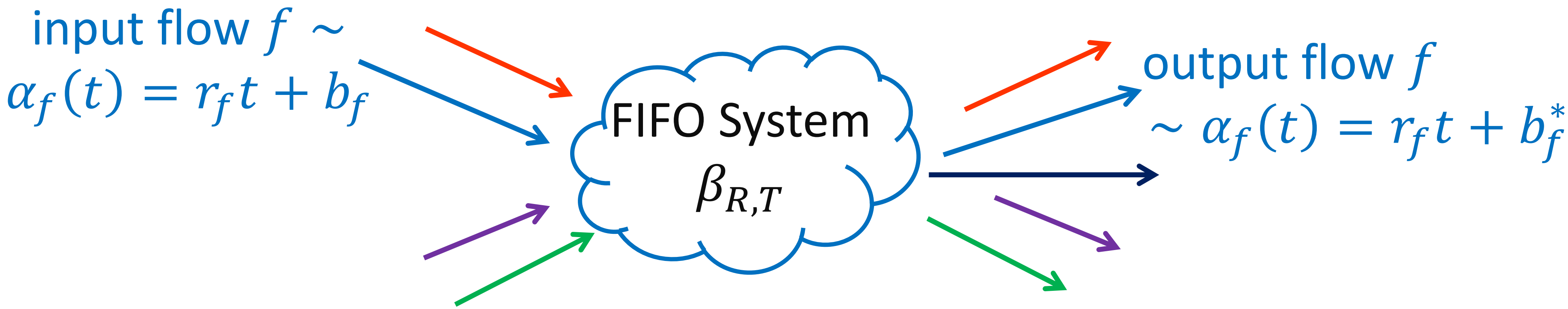
Flows 1 and 2 served FIFO + service curve β to the aggregate; Flow 2 is constrained by arrival curve α_2
$\Rightarrow$ for every θ, β_θ is a service curve for flow 1, with

$$\beta_\theta(t) = [\beta(t) - \alpha_2(t-\theta)]^+ 1_{\{t>\theta\}}$$

Example: $\alpha_i(t) = r_i t + b_i$,
$\beta =$ rate-latency (R, T)
β_θ and $\beta_{\theta'}$ are
service curves for flow 1



Analyzing Per-Class Networks



arrival curve for output f is leaky bucket r_f, b_f^* with

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

(obtained using service curve β_θ with $\theta = T + \frac{b_{tot} - b_f}{R}$)

Burstiness of every flow inside network increases along its path as a function of other flows' burstiness (cascade).

Solved recursively in feed-forward networks, otherwise by fix-point.

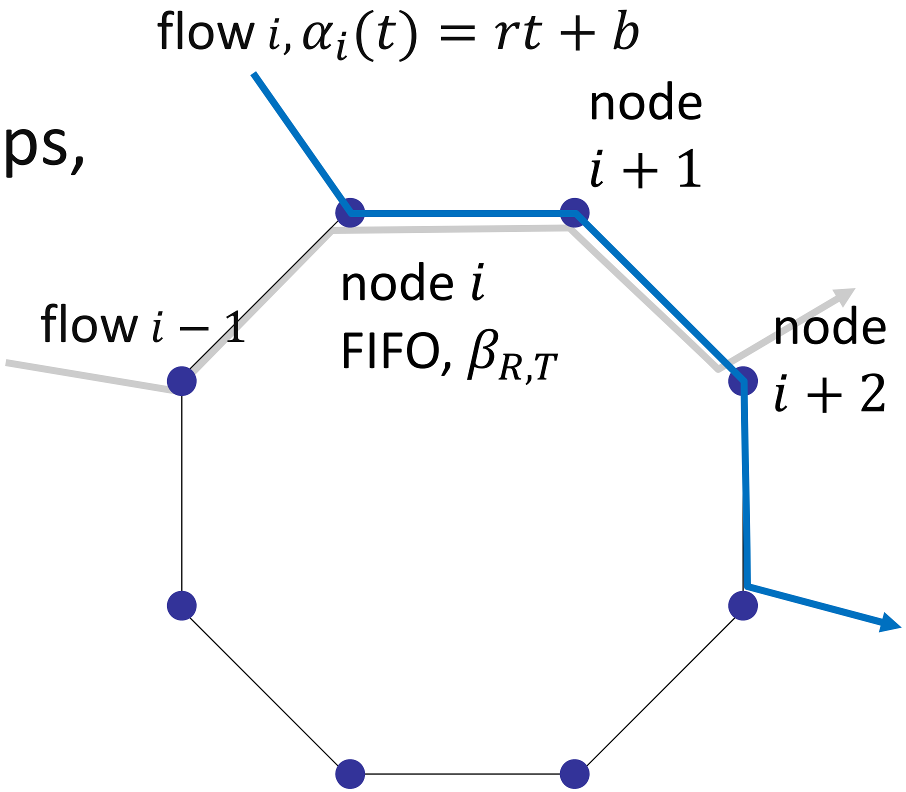
Example

b_j= burstiness of every flow i after j hops, with $b_0 = b$ and :

$$b_j = b_{j-1} + r\left(T + \frac{b_{tot} - b_{j-1}}{R}\right)$$

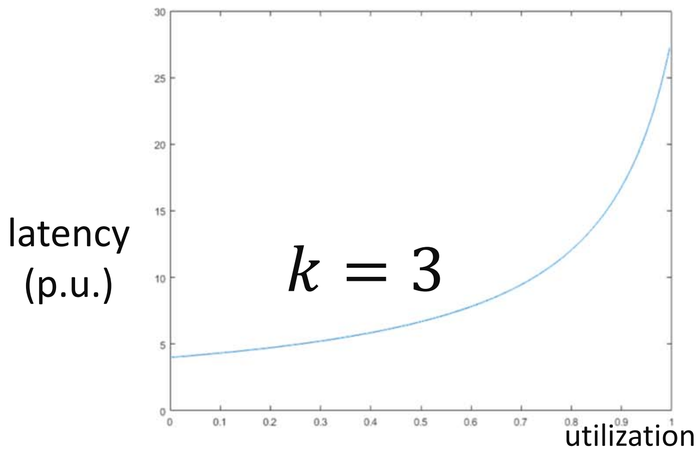
$$b_{tot} = b + b_1 + \cdots + b_{k-1}$$

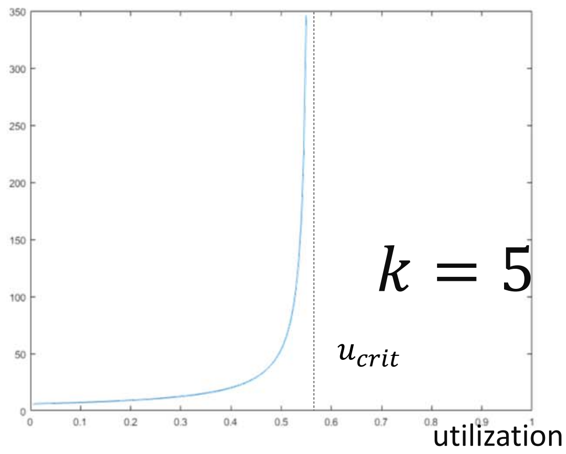
Fixed point in $(b_{tot}, b_1, \ldots, b_{k-1})$ has a positive solution when $(k-1)r + (1-r)^k < 1$, which occurs for $r < r_{crit}$

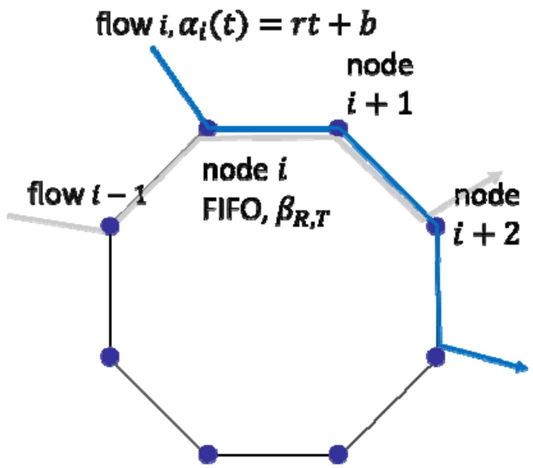


one flow sourced at every node, uses k hops
$k = 3$ on the figure
all flows have same arrival curve at source
I nodes, I flows in total
$kr \leq R$

Upper Bound on Latency at any Node





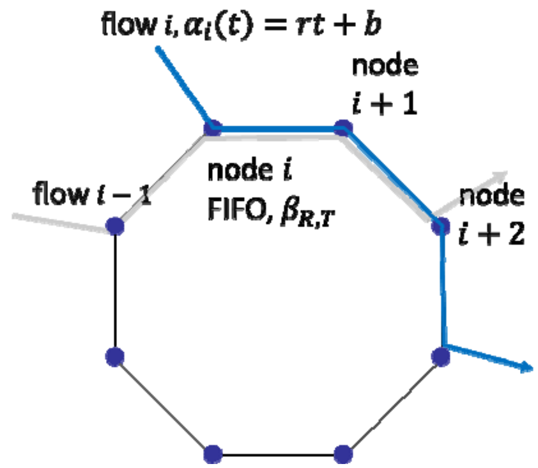


$T = 1, R = 1$ p.u.
$b = 1$ p.u.

k	1	2	3	4	5	6	7	8
$u_{crit} = k\, r_{crit}$	1	1	1	0.7579	0.5591	0.4409	0.3632	0.3084

If positive fixed point exists, then it gives a valid bound (time-stopping method)

Delay Based Method



If a system imposes a delay $\leq D$ to a flow f with input arrival curve α_f, an output arrival curve is $\alpha_f(t + D)$.

Often used in feed-forward networks [Boyer et al 2011]

For this ring example: after j hops, delay $\leq jD$, hence the fix-point:

$$b_j = b + jrD, \qquad D = \frac{b + b_1 + \cdots + b_{k-1}}{R} + T$$

Delay bound and $u_{crit} = \min\left(1, \frac{2}{k-1}\right)$ are less good than with previous method.

Other Methods

Several techniques improve delay bounds, tightness and u_{crit}: PMOO, PMOC, linear programming, etc.

[Bouillard et al 2018][Amari et al 2016] [Boyer et al 2012] [Bouillard-Stea 2015][Bondorf et al 2017] [Rizzo-Le Boudec 2008] [Tassiulas-Georgiadis 1996] [Chlamtac et al 1998]

There cannot be a bound that depends only on max aggregate burstiness, number of hops h and link utilization u when $u \geq \frac{1}{h-1}$

[Charny – Le Boudec 2000]

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.

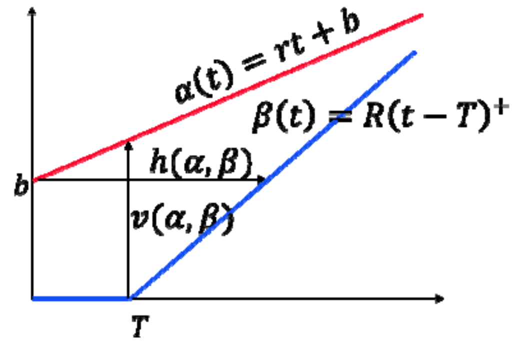
(existence of a bound on all delays ⇔ existence of a bound on all backlogs)

Which FIFO Networks are Stable ?

An overloaded FIFO network is not stable.

A single-node network that is underloaded or critical is stable.

A feed-forward network that is underloaded or critical is stable.



Question: Is an arbitrary underloaded network stable ?

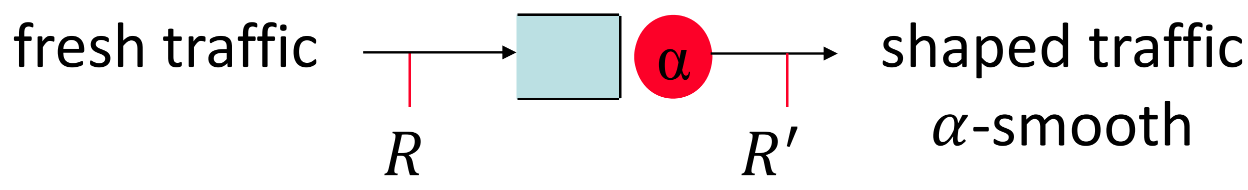
Is an Arbitrary Underloaded FIFO Network Stable ?

For any $\varepsilon > 0$ there is an unstable underloaded FIFO network with load factor $u < \varepsilon$ [Andrews 2009]

Every underloaded ring is stable [Tassiulas and Georgiadis 1996, Le Boudec and Thiran 2001]

If the interference condition [(4) in Rizzo-Le Boudec 2007] holds and service curves are strict then the FIFO network is stable; in the special case $R_i = R, \forall i,$ the condition is $r_f \leq R/(1 + RIN_f)$ where RIN_f is the number of flows that interfere with f.

7. Fluid Shapers



(Fluid) Shaper forces output to be constrained by α

(Fluid) Greedy Shaper stores data in a buffer only if needed

Examples:

constant bit rate link is fluid greedy shaper for $\alpha(t) = ct$

Shaper constraints are

$$R'(t) \leq R(t)$$
$$R'(t) \leq (R' \otimes \alpha)(t)$$

Min-Plus Residuation Theory [Baccelli et al. 1992]

$\mathcal{G}$ is the set of wide-sense increasing functions $[0, +\infty) \to [0, +\infty]$ and $\Pi\colon \mathcal{G} \to \mathcal{G}$. Consider the set of functions $x \in \mathcal{G}$ that satisfy

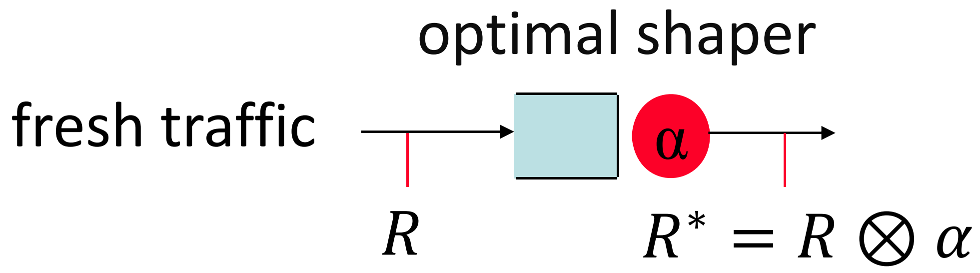
(P) $\quad x(t) \le [\Pi(x)](t)$ and $x(t) \le g(t)$ for some $g \in \mathcal{G}$

If Π is isotone and upper-semi-continuous, there is one maximal solution to (P). It is given by

$$x^* = \inf\{x^0, x^1, x^2, \dots\} \text{ with } x^0 = g \text{ and } x^{i+1} = \Pi(x^i).$$

"Π isotone" means that $x \le y \Rightarrow \Pi(x) \le \Pi(y)$; "$\Pi$ upper semi-continuous" means that $\Pi\left(\inf_n x_n\right) = \inf_n \Pi(x_n)$

I/O Characterization of Fluid Greedy Shaper



The problem
(where the unknown is
the function R')

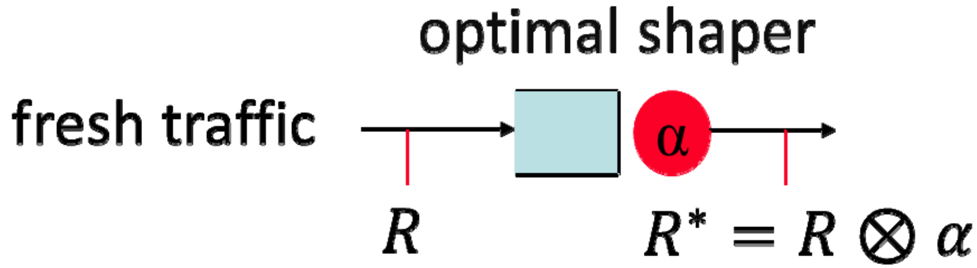
$$R'(t) \leq R(t)$$
$$R'(t) \leq (R' \otimes \alpha)(t)$$

has one maximal solution R^*, given by $R^* = R \otimes \bar{\alpha}$

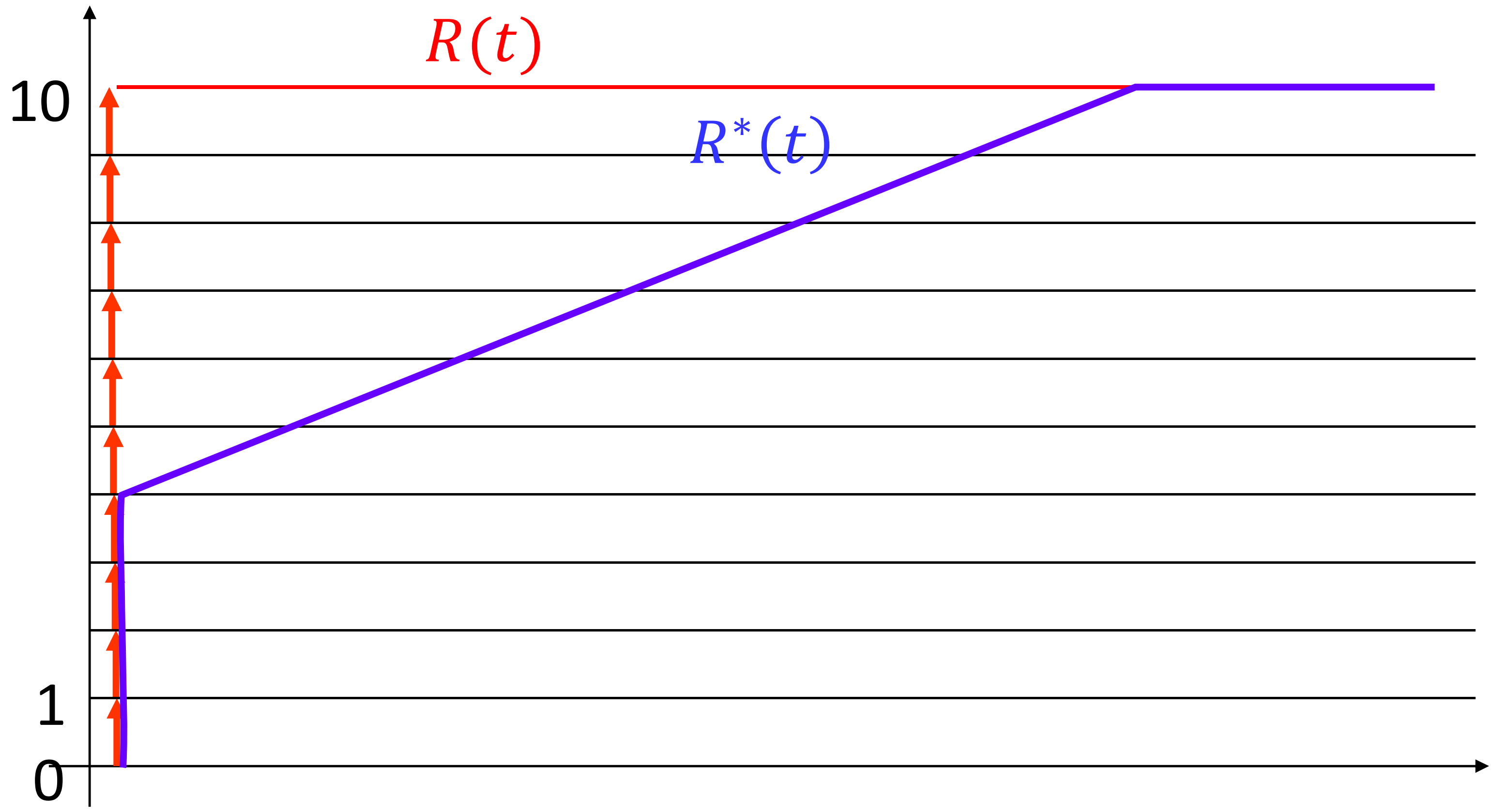
We can always assume that α is sub-additive and $\alpha(0) = 0$ so that
$R^* = R \otimes \alpha$

$\Rightarrow$ greedy shaper has service curve α

Example



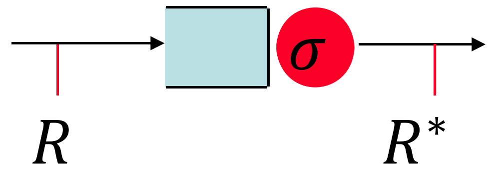
$\alpha(t) = rt + b$



$b = 4\ p.u.$

Fluid Greedy Shaper Keeps Arrival Constraints

fresh traffic
constrained by α



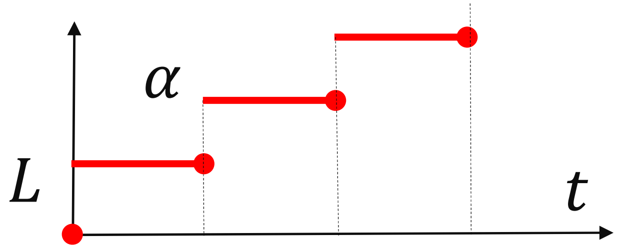
re-shaped traffic

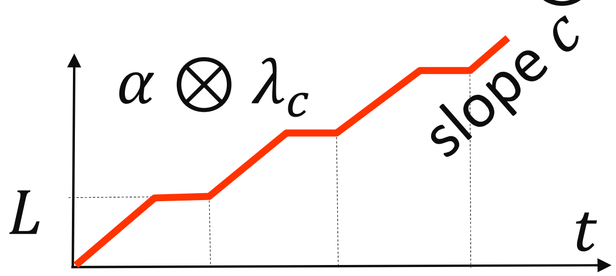
The output of the fluid greedy shaper is still constrained by α

Proof:

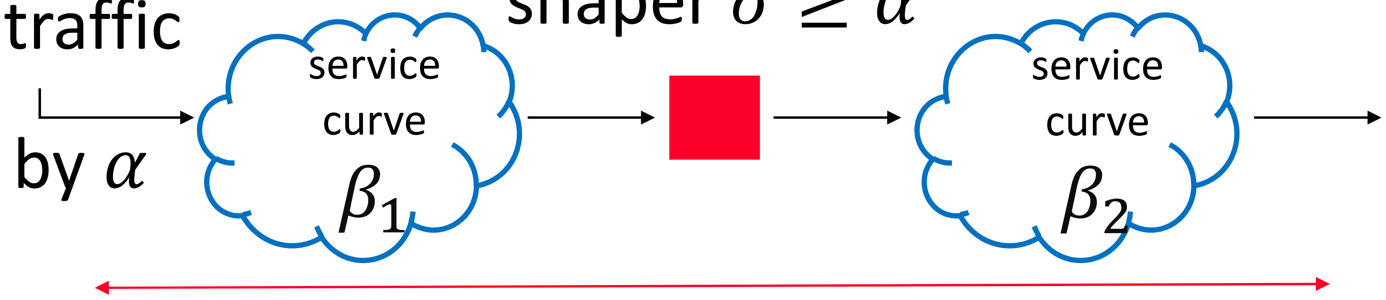
$$R^* = R \otimes \sigma = (R \otimes \alpha) \otimes \sigma = (R \otimes \sigma) \otimes \alpha = R^* \otimes \alpha$$

Application: if flow, with arrival curve α, is transmitted over a constant rate line of rate c, the output has arrival curve $\alpha \otimes \lambda_c$
$(\lambda_c(t) = ct)$





Fluid Greedy Shapers Do Not Increase End-to-end Delay Bound

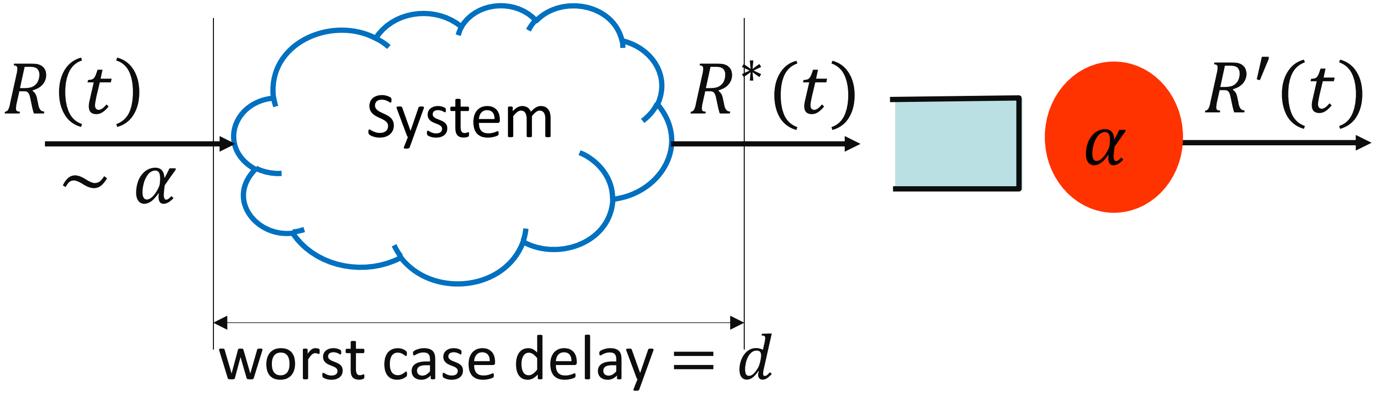


same end-to-end delay bound with or without shaper

Proof: $h(\alpha, \beta_1 \otimes \sigma \otimes \beta_2) = h(\alpha, \sigma \otimes \beta_1 \otimes \beta_2) = h(\alpha, \beta_1 \otimes \beta_2)$

Re-shaping traffic is for free (in terms of delay), but reduces downstream buffering.

Fluid Greedy Re-Shaper does not Increase Worst-Case Delay



If R has arrival curve α and if $h(R, R^*) \leq d$ then $h(R, R') \leq d$.

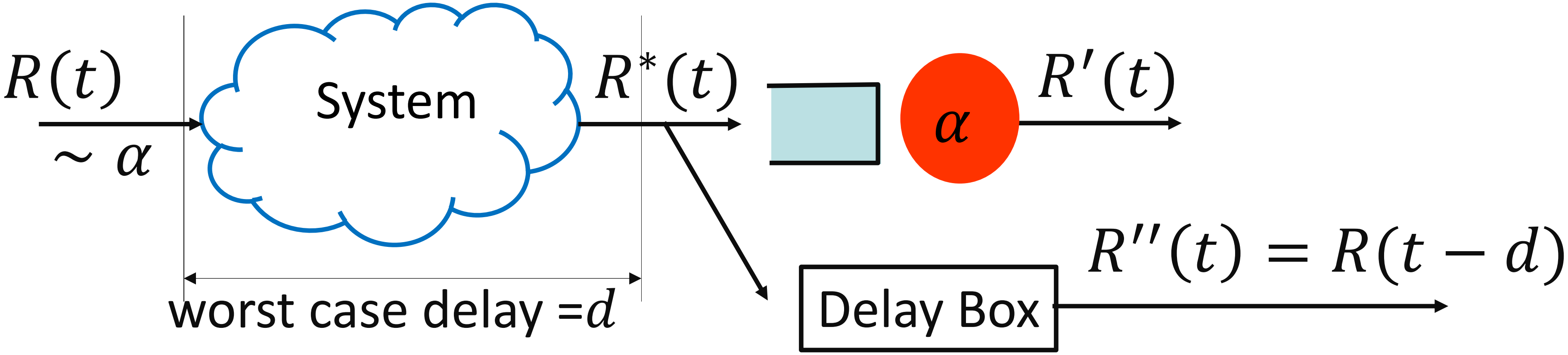
Proof: Note that $h(R, R^*) \leq d \Leftrightarrow R^* \geq R \otimes \delta_d$

By hypothesis: $R = R \otimes \alpha$ and $R^* \geq R \otimes \delta_d$

Now $R' = R^* \otimes \alpha$

$\Rightarrow R' \geq (R \otimes \delta_d) \otimes \alpha = (R \otimes \alpha) \otimes \delta_d = R \otimes \delta_d$

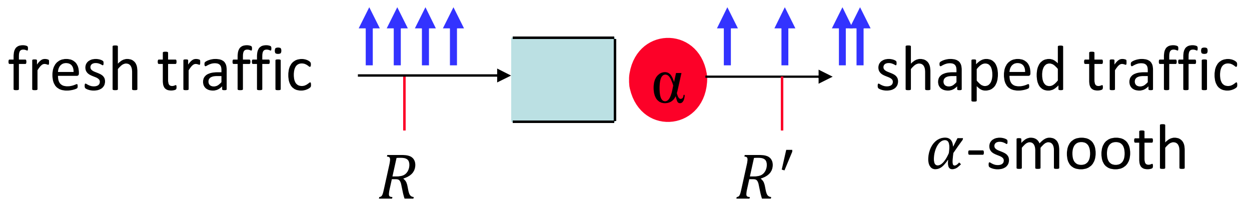
An Alternative Proof



If R has arrival curve α and if $h(R, R^*) \leq d$ then $h(R, R') \leq d$.

Proof: Note that $h(R, R^*) \leq d \Leftrightarrow R^*(t) \geq R(t-d)$
The delay box is a (non-greedy) shaper for R^*;
therefore (optimality of greedy shaper):
$R''(t) \leq R'(t)$ i.e. $R'(t) \geq R(t-d)$

8. Packetized Shapers

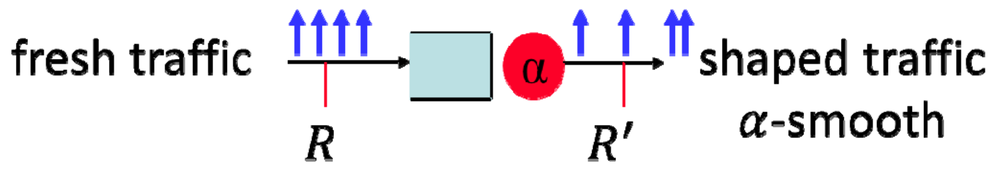


(Packetized) Shaper

forces output to be packetized and constrained by α

(Packetized) Greedy Shaper stores packets in a buffer only if needed

Packetized Shaper



Packetized Shaper constraints are

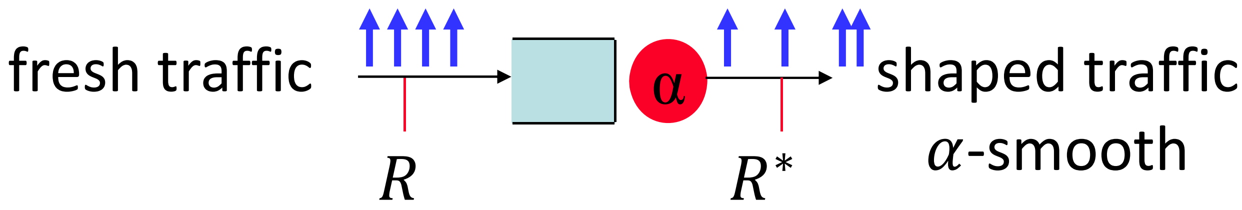
$$R'(t) \leq R(t)$$
$$R'(t) \leq (R' \otimes \alpha)(t)$$
$$R'(t) \leq P^L(R'(t))$$

There is a maximal solution (Packetized Greedy Shaper)

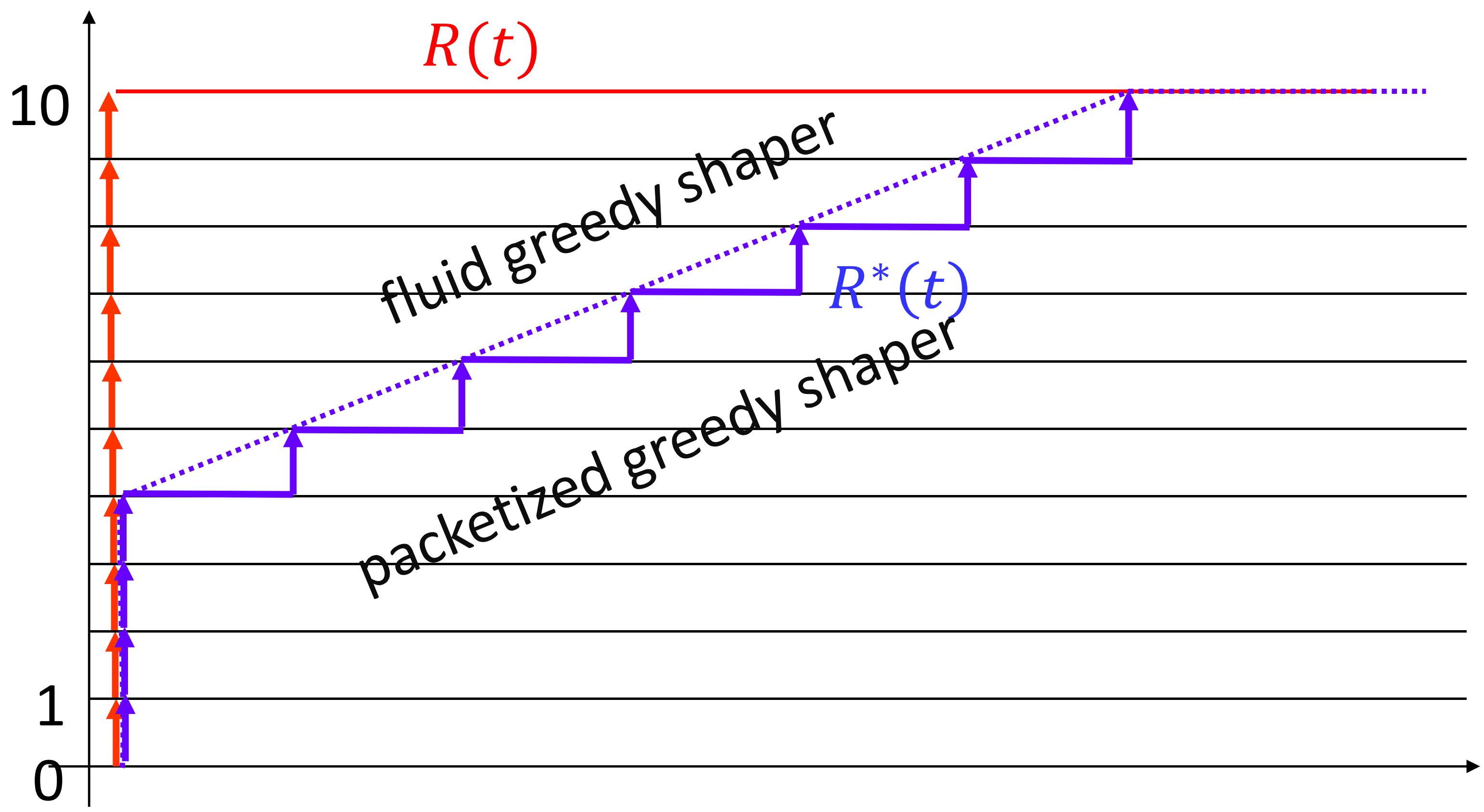
$$R^* = \inf\{R^{(1)}, R^{(2)}, \dots\}$$

with $R^{(0)} = R$ and $R^{(i)} = P^L\left(\alpha \otimes R^{(i-1)}\right)$

Example



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

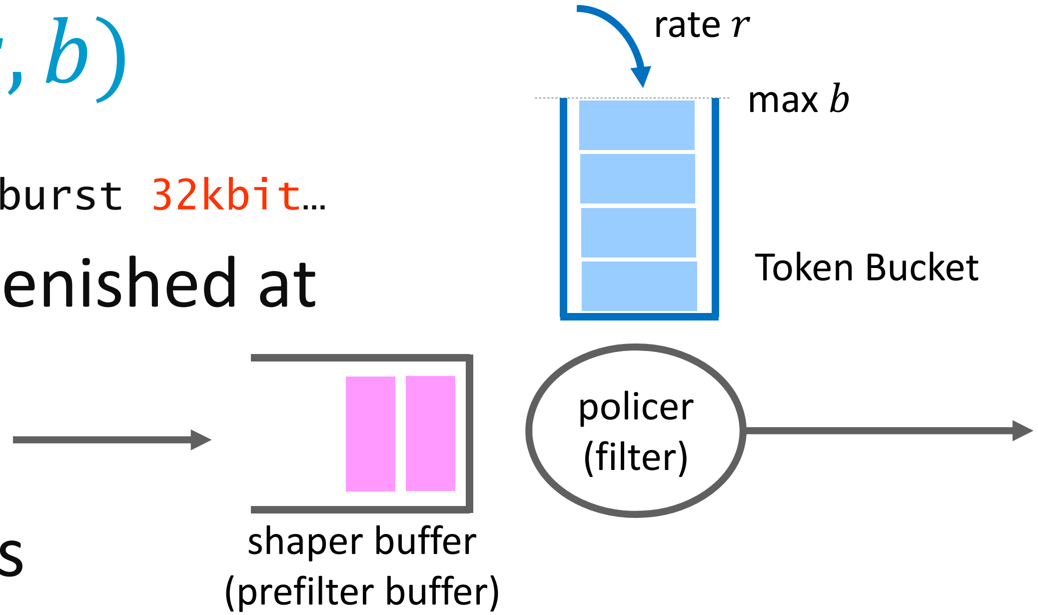


$r = 4\ p.u.$

Linux's Token Bucket Filter (r, b)

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

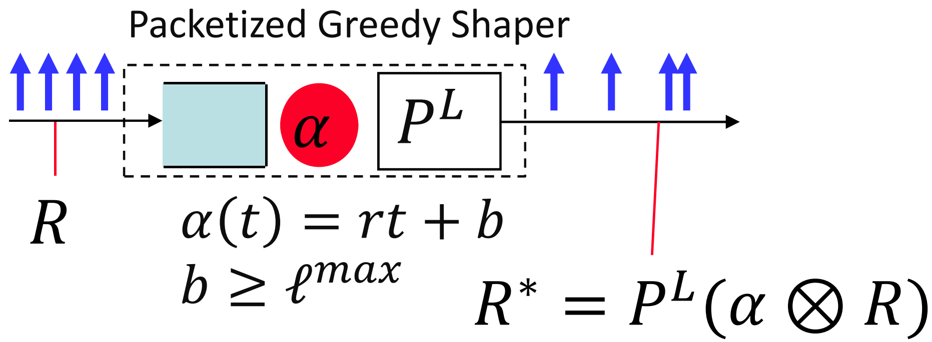
Token bucket is spontaneously replenished at rate r up to some maximum b.
In order to be released, a packet must consume an amount of tokens equal to its size.



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

TBF(r, b) is the packetized greedy shaper for the arrival curve $\alpha(t) = rt + b$.

Packetized Greedy Shaper



When

(C) α is piecewise-linear, concave and $\alpha(0^+) \geq \ell^{max}$

then $R^* = P^L(\alpha \otimes R)$
i.e. packetized greedy shaper = fluid greedy shaper + packetizer.

This is why TBF(r, b) = packetized greedy shaper for $\alpha(t) = rt + b$.

Such packetized greedy shapers keep arrival constraints.

[Le Boudec 2002]

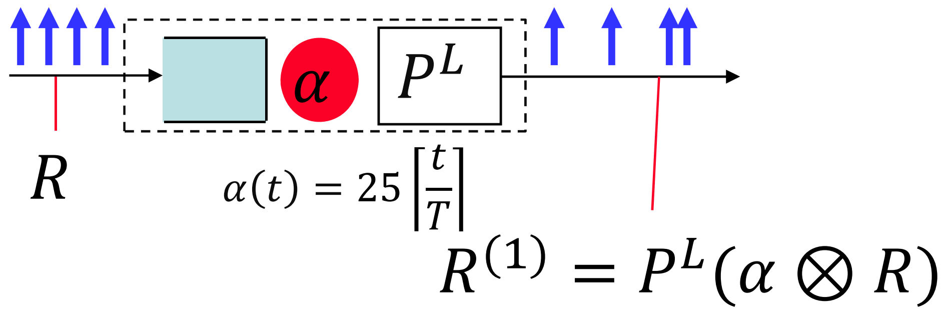
Counter-Example

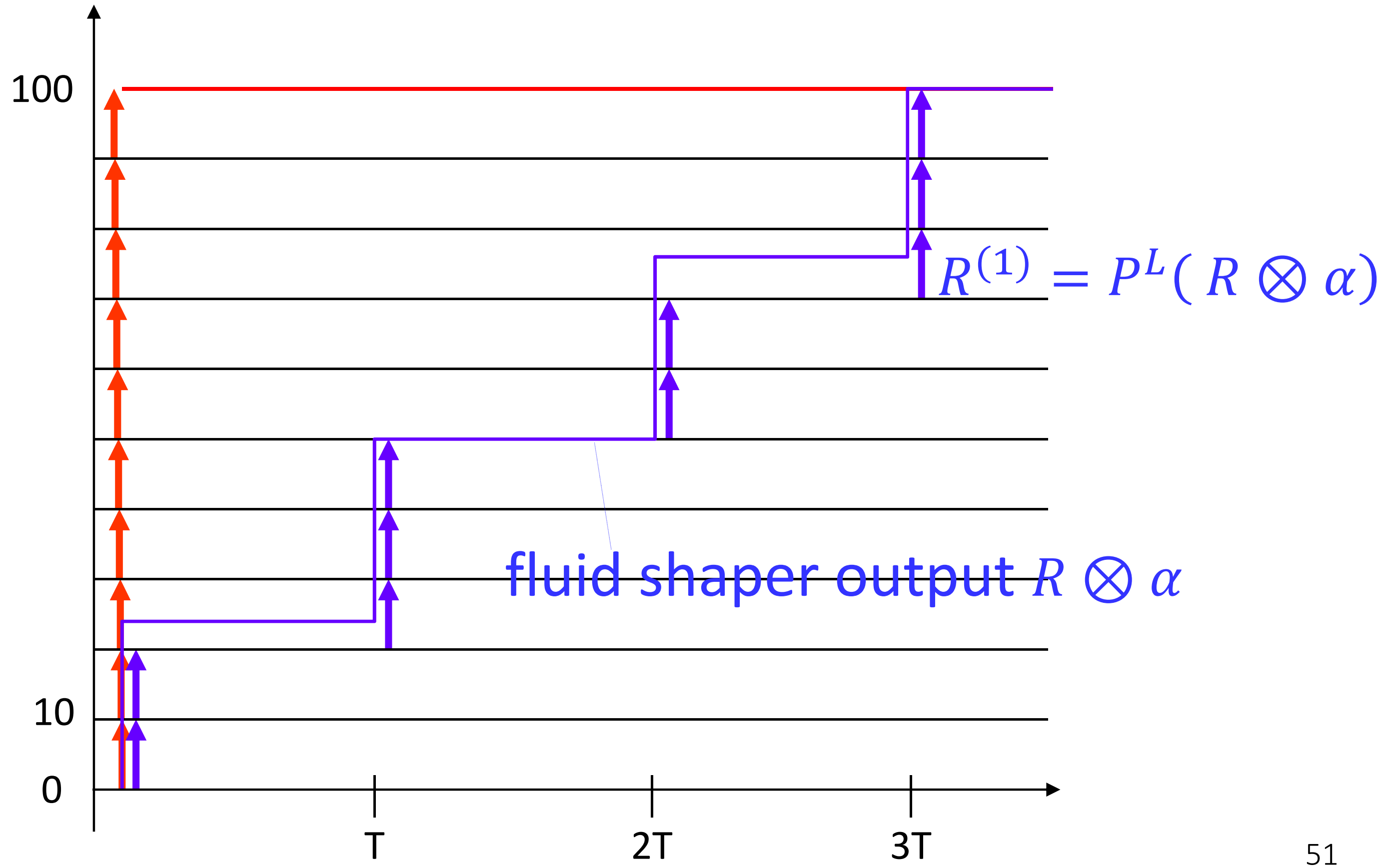
$$\alpha(t) = 25\left\lceil \frac{t}{T} \right\rceil$$

(at most 25 data units every T time units)

$R(t) =$ 10 packets of size 10 at time $t = 0$

$R^{(1)} = \alpha \otimes R$ is not constrained by arrival curve α

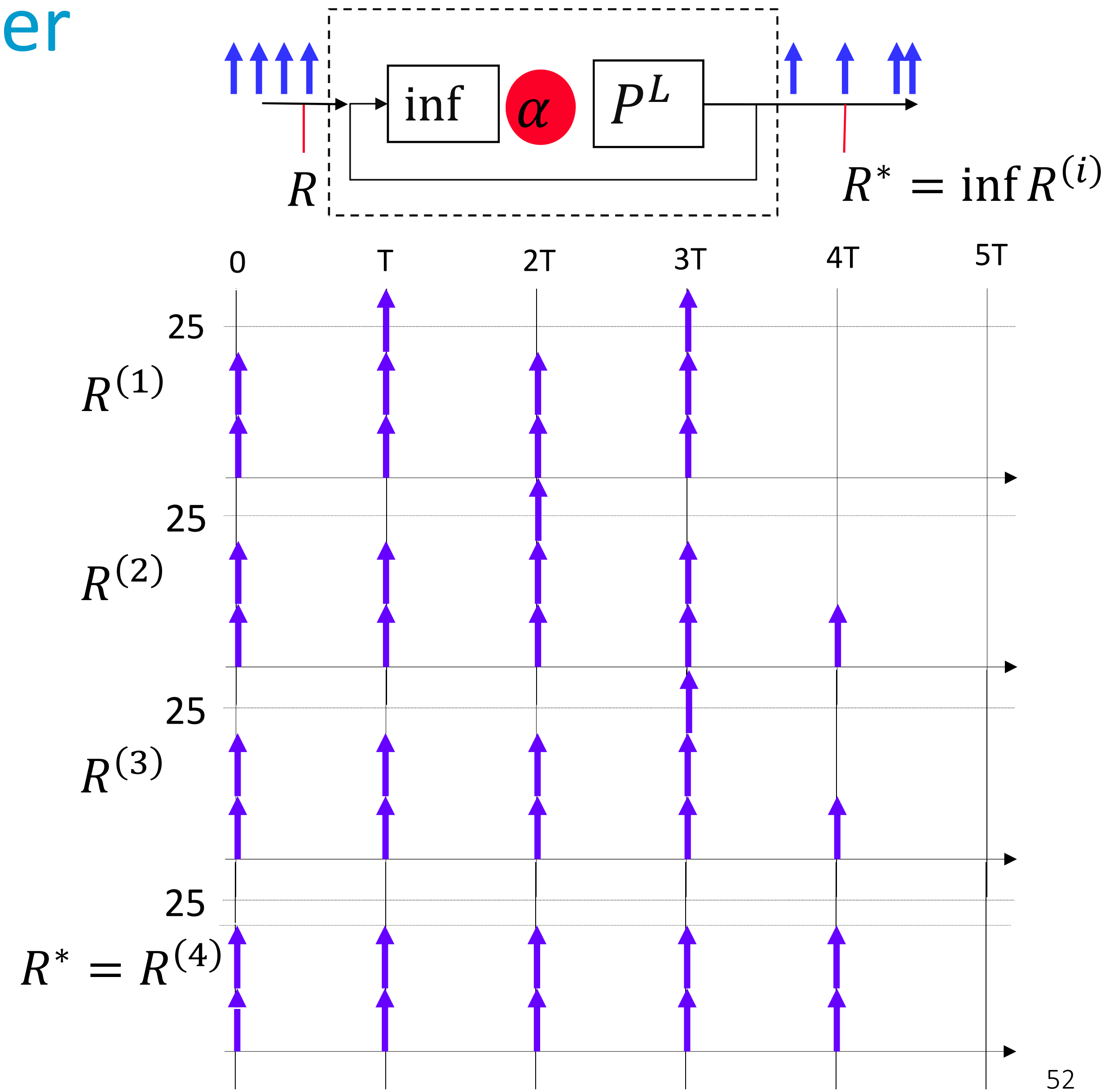




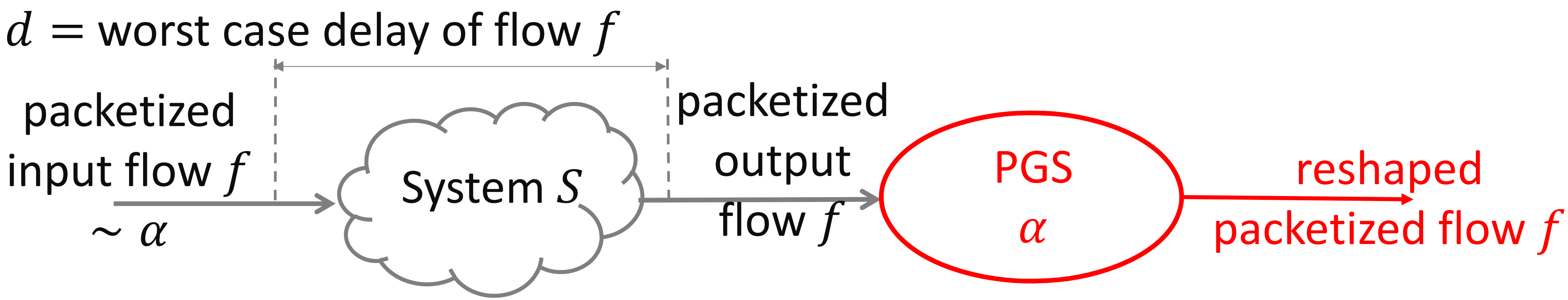
Packetized Greedy Shaper Output

When (C) does not hold (e.g. for staircase arrival curves), $R^* \neq P^L(\alpha \otimes R)$ the iteration does not stop at step 1

(and packetized greedy shaper may not keep arrival constraints)



Delay Property of Packetized Greedy Shapers

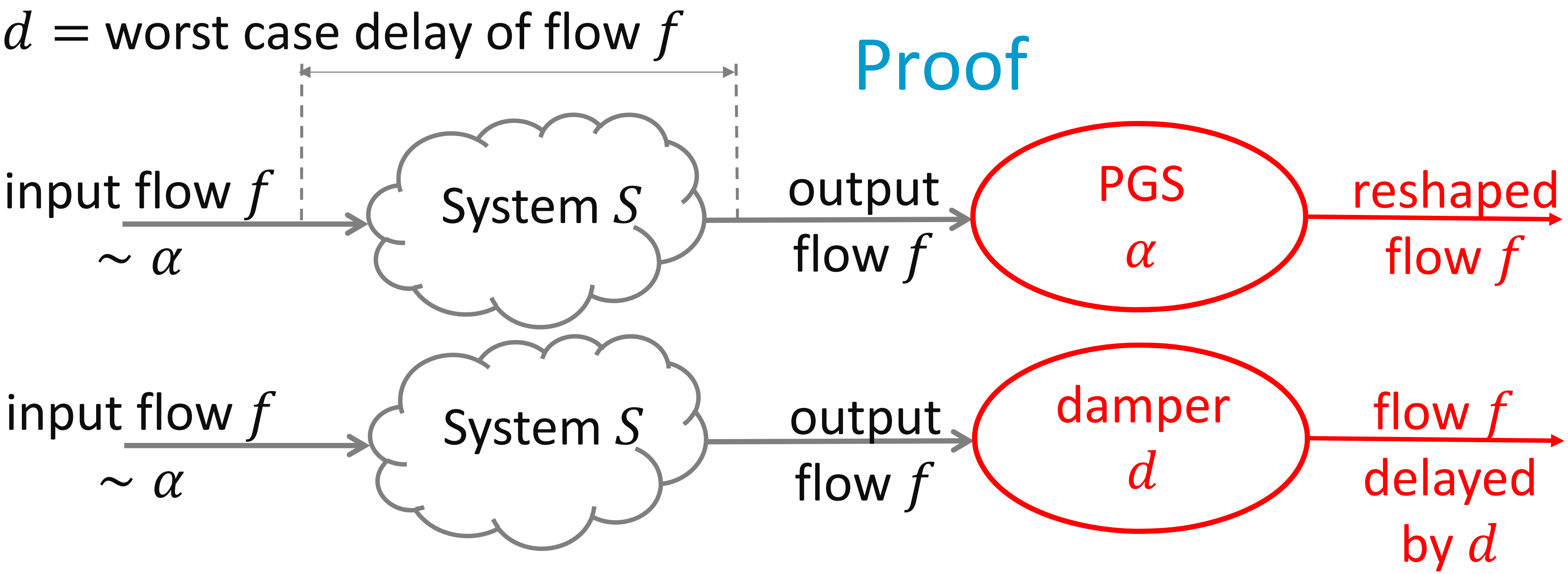


One packetized flow f goes through a system S; system S is FIFO for flow f; flow f is constrained by arrival curve α at input to S; output flow f is reshaped at output through a packetized greedy shaper for same arrival curve α (shaper is FIFO)

Theorem: The worst case delay of flow f is not increased:

Re-shaping is for free ! [Le Boudec 2018] -- True whether (C) holds or not.

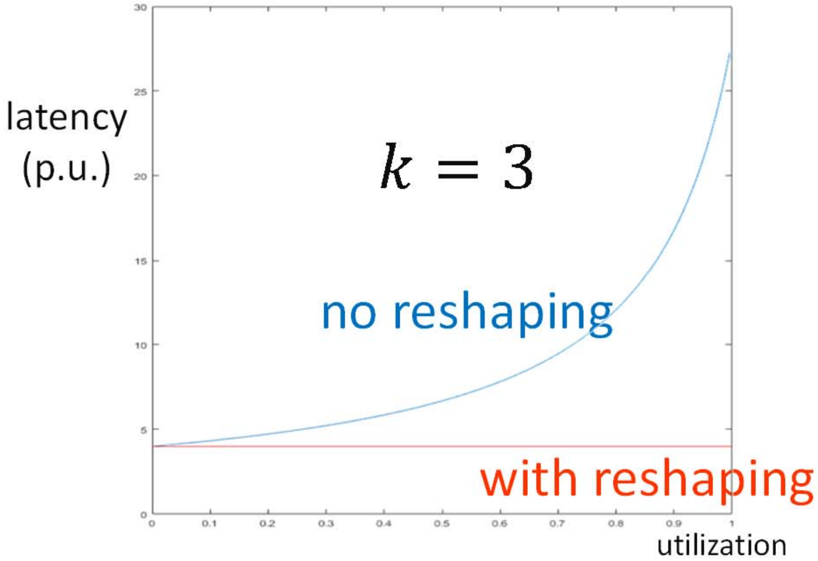
Proof

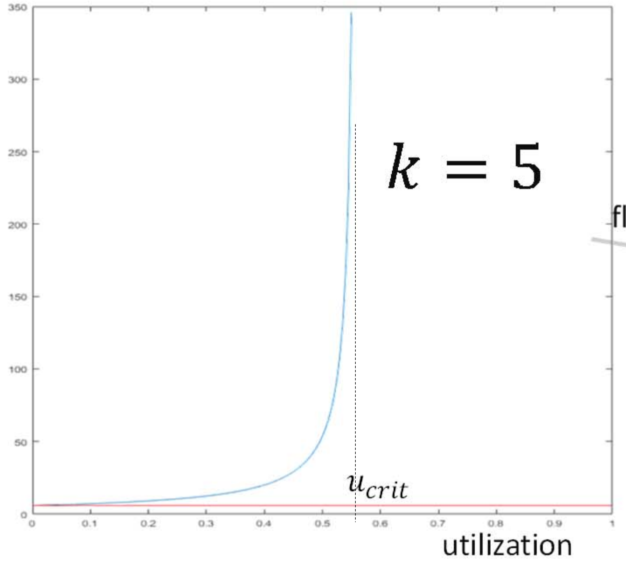


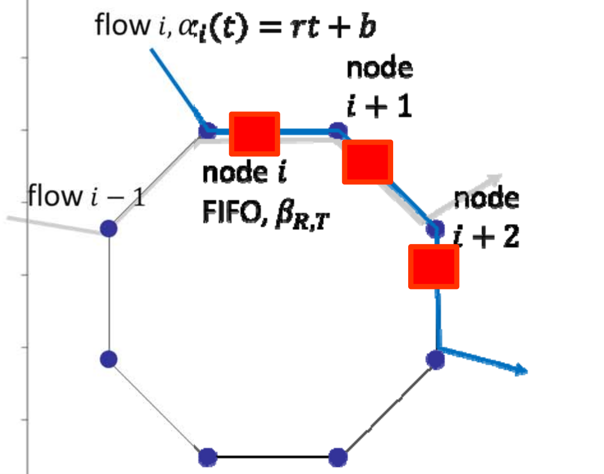
Replace packetized greedy shaper by **damper** [Verma et al 1991]:
Damper forces total delay of flow f to be exactly d; Damper is causal if d is $\geq$ worst-case delay through S.

Output of damper is input flow f, time-shifted by $d \Rightarrow$ is α −smooth $\Rightarrow$ Damper is a packetized shaper $\Rightarrow$ (maximal property of packetized greedy shaper) flow f delayed by d is no earlier than reshaped flow f

Re-Shaping Avoids Issue of Non-Feedforward Networks







$T = 1, R = 1$ p.u.
$b = 1$ p.u.

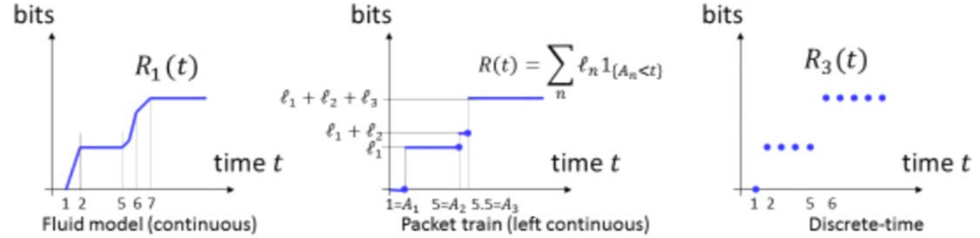
Per-flow (re-) shaper at every node (e.-g. TBF) avoids burstiness cascade

Implementation requires per-flow FIFO ⇒ see Interleaved Regulator for a solution [Le Boudec 2018]

9. Min-Plus or Max-Plus ?

Network calculus uses mainly cumulative arrival or departure functions ($R(t) =$ number of bits observed in t time units) → min-plus algebra

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



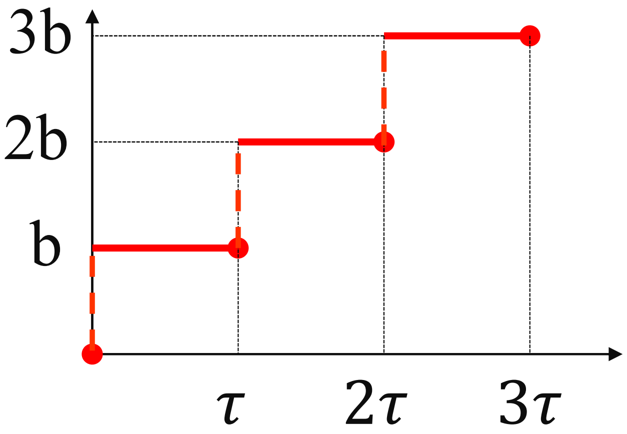
Daters: (A, L) with $A = (A_1, A_2, \ldots)$ (dates) and $\ell = (\ell_1, \ell_2, \ldots)$ (lengths in bits)

For a packet train: $R(t) = \sum_{n\geq 1} \ell_n 1_{\{A_n<t\}}$

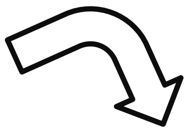
Liebeherr uses cumulative arrival time functions ($T(x) =$ time taken to observe x bits); leads to a dual, quasi-equivalent approach, that uses max-plus algebra. Duality uses pseudo-inverses $R = T^{\downarrow}, T = R^{\uparrow}$. [Liebeherr 2017]

Pseudo-Inverses [Liebeherr 2017]

$$f^{\downarrow}(x) = \inf\{t, f(t) \geq x\}$$
$$f^{\uparrow}(x) = \sup\{t, f(x) \leq x\}$$

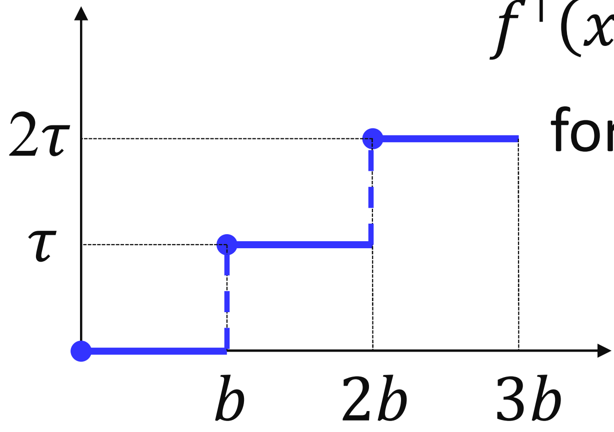


$$f(t) = b\left\lceil \frac{t}{\tau} \right\rceil$$



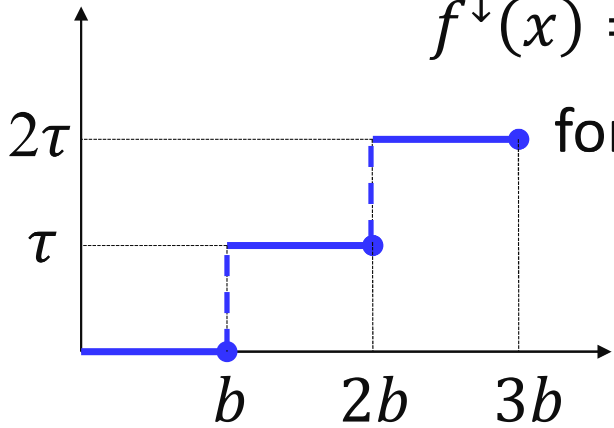
$$f^{\uparrow}(x) = \tau\left\lfloor \frac{x}{b} \right\rfloor$$

for $x \geq 0$



$$f^{\downarrow}(x) = \tau\left\lceil \frac{x}{b} - 1 \right\rceil$$

for $x > 0$



Chang's Max-Plus Calculus

Yet a different approach, uses Daters (also called "Marked Point Process"). Leads to a different max-plus algebra where "convolution" is

$$(A \odot_L g)(x) = \max_{1 \le m \le n} \left(A_m + g(\ell_m + .. + \ell_{n-1})\right)$$

[Chang 2002]

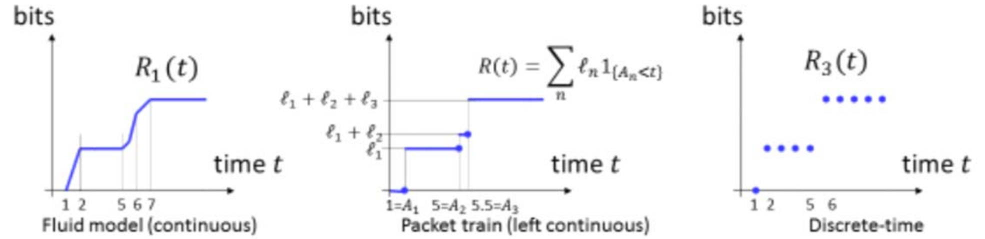
Not associative nor commutative

E.g. A flow is g −regular if $A \ge A \odot_L g$

E.g. $g(x) = \frac{x}{r}$, flow is g −regular $\Leftrightarrow A_n - A_{n-1} \ge \frac{\ell_{n-1}}{r}$

called LRQ-rule in [Specht-Samii 2016]

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



Daters: (A, L) with $A = (A_1, A_2, \ldots)$ (dates) and $\ell = (\ell_1, \ell_2, \ldots)$ (lengths in bits)

For a packet train: $R(t) = \sum_{n \ge 1} \ell_n 1_{\{A_n < t\}}$

Chang's g-regularity

g-regularity is an alternative to arrival curve, but is not equivalent. Does not work well with aggregation.

g-regularity can be mapped approximately to arrival curve: e.g.

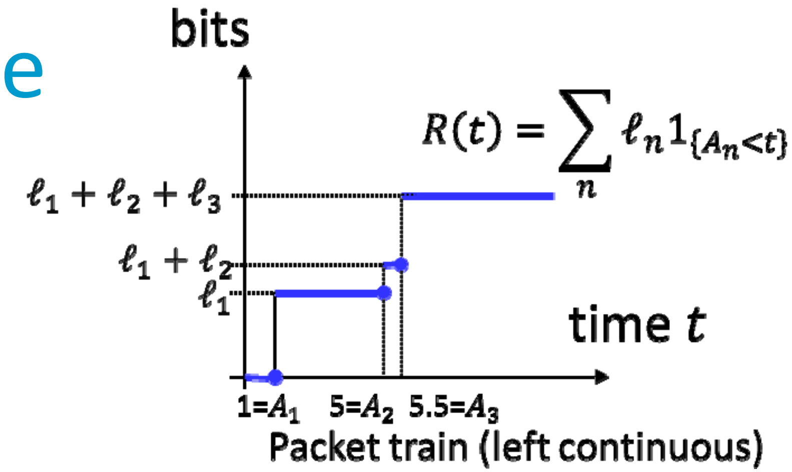
g −regular with $g(\ell) = \frac{\ell}{r} \Rightarrow$ arrival curve $\alpha(t) = rt + \ell^{MAX}$

arrival curve $\alpha(t) = rt + \ell^{MAX} \Rightarrow g$ −regular with $g(\ell) = \frac{\left(\ell - \ell^{MAX}\right)^{+}}{r}$

ℓ^{MAX} = max packet size

Instead, we discuss next the use of Daters / Max-Plus with standard network calculus concepts of arrival curves (in the future, with service curves).

Max-Plus Expression of Arrival Curve



Theorem: Arrival curve constraint α is equivalent to

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

[Le Boudec 2018]

E.g. for $\alpha(t) = rt + b,\ \alpha^{\downarrow}(x) = \max(0, \frac{x-b}{r})$

Equivalent Formulations

	Original Definition	Equivalent Max-Plus Formulation
affine arrival curve (leaky bucket)	$R(t) - R(s) \leq r(t-s) + b$	$A_n - A_m \geq \frac{L_m + \cdots + L_n - b}{r}$
staircase arrival curve (at most b bits in τ seconds)	$R(t) - R(s) \leq b \left\lceil \frac{t}{\tau} \right\rceil$	$A_n - A_m \geq \tau \left\lceil \frac{L_m + \cdots + L_n - b}{b} \right\rceil$

$R(t) =$ number of bits seen in $[0, t]$; $A_m =$ arrival time for packet n

Max-Plus Representation of Rate-Latency Service Curve

A FIFO network element offers a rate-latency service curve $\beta_{R,T}$

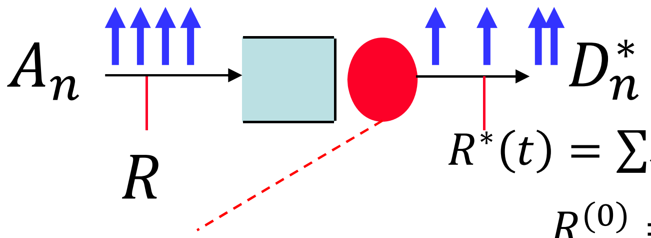
Input flow is packetized

Then: for every n there exists some $m \leq n$ s.t.

$$D_n \leq A_m + \frac{\ell_m + \cdots + \ell_n}{R} + T$$

And conversely, if this holds for some network element S, then S is the concatenation of a rate-latency service curve element and a packetizer (= Guaranteed Rate node)

Max-Plus Representation of Packetized Greedy Shaper



Packetized Greedy Shaper α

$$R^*(t) = \sum_n \ell_n 1_{\{D_n^* < t\}} = \inf_i R^{(i)}(t),$$

$$R^{(0)} = R, \qquad R^{(i)} = P^L(R^{(i-1)} \otimes \alpha)$$

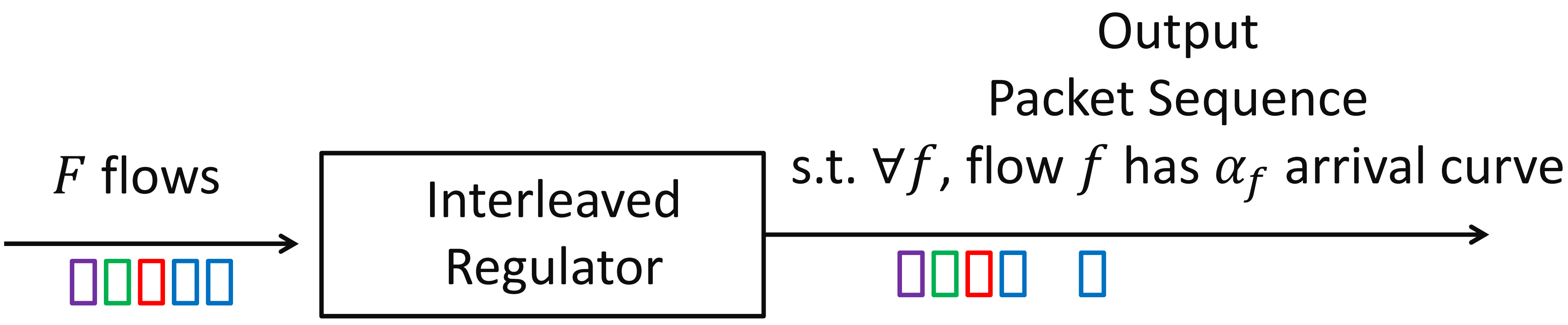
The output times D_n^* of packetized greedy shaper are given by $D_1^* = A_1$ and

$$D_n^* = D_{n-1}^* \vee A_n \vee \bigvee_{m=1}^{n-1} \left(A_m + \alpha^{\downarrow}(\ell_m + \cdots + \ell_n)\right)$$

(time domain representation of PGS) [Le Boudec 2018]

(optimality) For any other (non greedy) packetized shaper, the output times D'_n satisfy $D'_n \geq D_n^*$

Application: Interleaved Regulator



An Interleaved Regulator (= interleaved packetized shaper) is a system such that is globally FIFO + every output flow f has α_f arrival curve.

Theorem: there is one minimal interleaved regulator, i.e. one that ouput packets no later than any other interleaved regulator.

[Le Boudec 2018]

Implementation of Minimal Interleaved Regulator

$$D_n = \max\left\{A_n, D_{n-1}, \vee_{m=1}^{i-1}\left(A_m^f + \alpha^{\downarrow}\left(\ell_m^f + \cdots + \ell_i^f\right)\right)\right\}$$

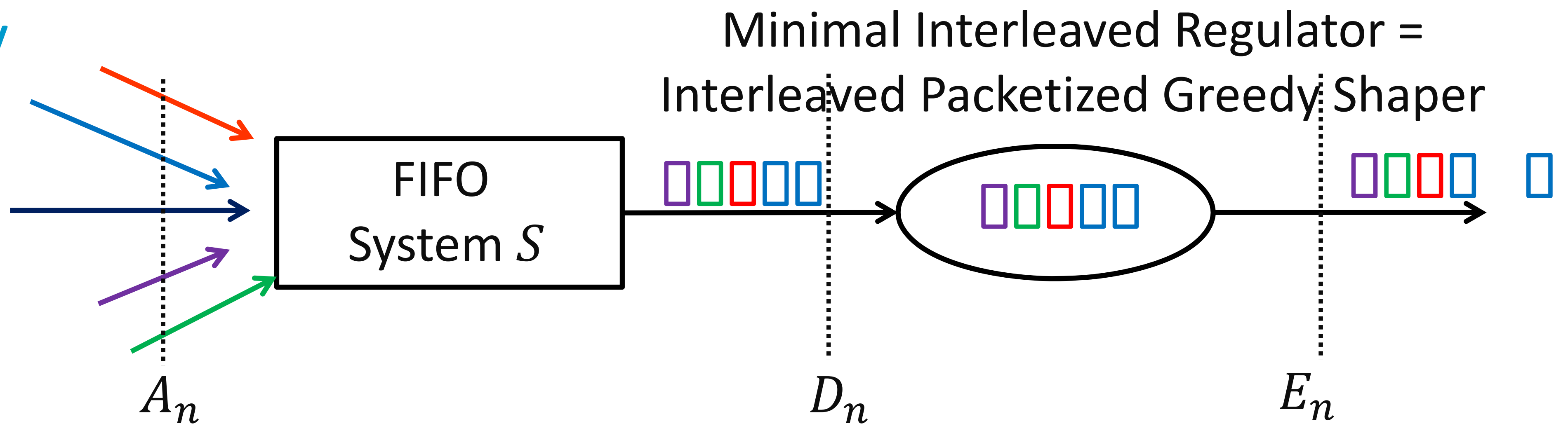
Eligibility Time of packet at head of queue

with f = flow of packet n, i =index of packet n in its flow, A_m^f = arrival time of mth packet of flow f, ℓ_m^f = length of mth packet of flow f

- One FIFO queue for all packets of all flows.
- Packet at head of queue is examined and delayed until it can be released while satisfying the arrival curve of its flow.
- Other packets wait until their turn comes.

[Specht-Samii 2016], called “Urgency Based Scheduler”. In IEEE TSN, called “Asynchronous Traffic Shaping”.

Minimal Interleaved Regulator Does Not Increase Worst Case Delay



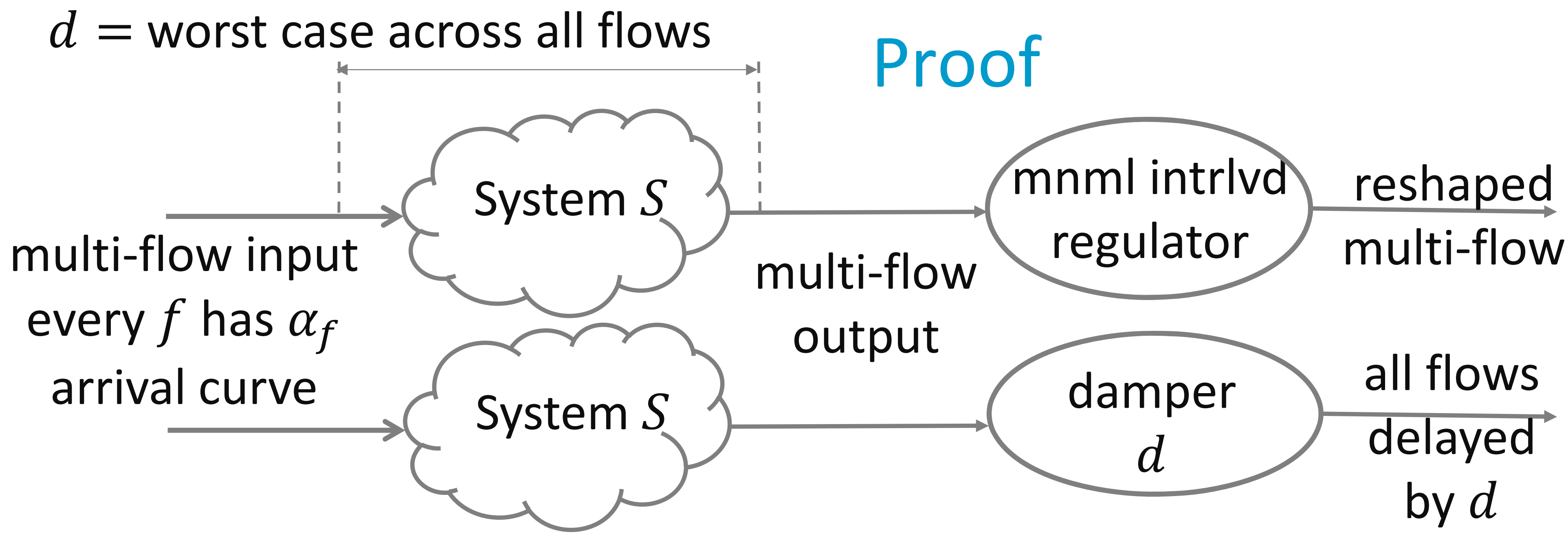
Every flow f has α_f arrival curve before input to S

Output of S is fed to minimal interleaved regulator with arrival curve α_f for flow f

Theorem: $\sup_n (D_n - A_n) = \sup_n (E_n - A_n)$

Worst-case delay of S across all flows is not increased by minimal interleaved regulator [Le Boudec 2018] [Le Boudec 2018 – slides]

Proof



- Replace minimal regulator by **damper** [Verma et al 1991]:
  Damper forces total delay of input to be exactly d; Damper is causal if d is $\geq$ worst-case delay through S
- Damper is an interleaved regulator $\Rightarrow$ multi-flow output delayed by d is no earlier than reshaped multi-flow

Other Regulation Constraints

Example: Packet Burstiness $PB(\rho, K)$ defined by:

number of packets seen in interval of duration t is $\leq \rho t + K \Leftrightarrow$

$A_n - A_m \geq \frac{n-m+1-K}{\rho}$ for $m \leq n$ [Le Boudec 2018, Jiang 2018]

Regulator concept replaces the concept of shaper; under mild conditions (``Pi-regulation"), any regulation rule has a minimal regulator. Any combination of regulation rules has a minimal interleaved regulator.

Minimal regulator does not increase per-flow delay.
Minimal Interleaved regulator does not increase FIFO system delay.

[Le Boudec 2018] [Le Boudec 2018 – slides]

Packet Count, Event Stream

Arrival function $R(t)$ = number of bits up to time t

Packet count function $P(d)$ = number of packets in d bits

Event count function $E(t)$ = number of packets up to time t

$E(t) = P\big(R(t)\big)$ (event stream)

$P^L\big(R(t)\big) = \; P^{\downarrow}\Big(P\big(R(t)\big)\Big)$ (packetizer) [Boyer-Roux 2016]

Theorem: The conditions are equivalent

1. $E(t) - E(s) \;\le\; f(t-s)$ for any $0 \le s \le t$
2. $A_n - A_m \ge f^{\downarrow}(n-m+1)$ for all $1 \le m \le n$ where A_n is arrival date of packet n [Le Boudec-slides 2018]

Viewpoint

Cumulative functions and daters / marked point process are complementary viewpoints. One naturally leads to min-plus, the other to max-plus.

We should be opportunistic and use both (and perhaps other) representations.

We should strive for results that are independent of the representation (e.g. delay properties of optimal regulators).

Tools

- The DiscoDNC is an academic Java implementation of the network calculus framework.[6]
- The RTC Toolbox is an academic Java/MATLAB implementation of the Real-Time calculus framework, a theory quasi equivalent to network calculus.[4]
- The CyNC [7] 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.[8]
- The Network calculus interpreter is an on-line (min,+) interpreter.
- The WOPANets is an academic tool combining network calculus based analysis and optimization analysis.[9]
- The DelayLyzer is an industrial tool designed to compute bounds for Profinet networks.[10]
- DEBORAH is an academic tool devoted to FIFO networks.[11]
- NetCalBounds is an academic tool devoted to blind & FIFO tandem networks.[12][13]
- The Siemens Network Planner (SINETPLAN) uses network calculus (among other methods) to help the design of a PROFINET network.[14]

copied on 2019 Feb 28 from https://en.wikipedia.org/wiki/Network_calculus

Conclusion

Network calculus main concepts:

arrival curve, minimal service curve and universal bounds
shapers, concatenation
for packet-based systems and for fluid systems.

Some key results are based on existence of minimal or maximal solutions to functional problems.

Challenges exist for per-class systems.

Fundamental results use min-plus convolution; other techniques (dater-based) can be useful.

References

[Amari et al 2016] Amari, Ahmed and Mifdaoui, Ahlem and Frances, Fabrice and Lacan, Jérôme. Worst-Case Timing Analysis of AeroRing- A Full Duplex Ethernet Ring for Safety-critical Avionics. In: 12th IEEE World Conference on Factory Communication Systems (WFCS), 3 May 2016 - 6 May 2016 (Aveiro, Portugal).

[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), p.33.

[Baccelli et al 1992] Baccelli, F., Cohen, G., Olsder, G.J. and Quadrat, J.P., 1992. Synchronization and linearity: an algebra for discrete event systems.

[Bennett et al 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 (TON)*, *10*(4), pp.529-540.

[Bondorf et al 2017], Bondorf, Steffen, Paul Nikolaus, and Jens B. Schmitt. "Quality and Cost of Deterministic Network Calculus: Design and Evaluation of an Accurate and Fast Analysis." *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1.1 (2017) and arXiv:1603.02094v3

[Bouillard et al 2018] Bouillard, A. Boyer, M. and Le Coronc, E. *Deterministic Network Calculus: from Theory to Practical Implementation,* ISTE Editions, 2018

[Bouillard-Stea 2015] Bouillard, A. and Stea, G., 2015. Exact worst-case delay in FIFO-multiplexing feed-forward networks. *IEEE/ACM Transactions on Networking (TON)*, *23*(5), pp.1387-1400.

[Boyer at al 2011] Boyer, M., Migge, J. and Fumey, M., 2011. *PEGASE-a robust and efficient tool for worst-case network traversal time evaluation on AFDX* (No. 2011-01-2711). SAE Technical Paper.

[Boyer et al 2012] Boyer, Marc, Nicolas Navet, and Marc Fumey. "Experimental assessment of timing verification techniques for AFDX." *6th European Congress on Embedded Real Time Software and Systems*. 2012.

[Boyer et al 2012] Boyer, M., Stea, G. and Sofack, W.M., 2012, October. Deficit Round Robin with network calculus. In *6th International ICST Conference on Performance Evaluation Methodologies and Tools* (pp. 138-147). IEEE.

[Boyer – Roux 2016] Boyer, M. and Roux, P., 2016. A common framework embedding network calculus and event stream theory. Proc. of the 21th IEEE Int. Conf. on Emerging Technologies and Factory Automation (EFTA 2016).

[Chang 1997] Chang, C.S., 1997, April. A filtering theory for deterministic traffic regulation. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE* (Vol. 2, pp. 436-443). IEEE.

[Chang-Lin 1998] Chang, C.S. and Lin, Y.H., 1998, May. A general framework for deterministic service guarantees in telecommunication networks with variable length packets. In *Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop on* (pp. 49-58). IEEE.

[Chang-Lin] Chang, C.S. and Lin, Y.H., 1998, May. A general framework for deterministic service guarantees in telecommunication networks with variable length packets. In *Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop on* (pp. 49-58). IEEE.

[Chang 2002] Chang, C.S., 2012. *Performance guarantees in communication networks*. Springer Science & Business Media.

[Charny-Le Boudec 2000] Charny, A. and Le Boudec, J.Y., 2000, September. Delay bounds in a network with aggregate scheduling. In *International Workshop on Quality of Future Internet Services*(pp. 1-13). Springer, Berlin, Heidelberg.

[Chlamtac et al 1998] Chlamtac, I., Faragó, A., Zhang, H. and Fumagalli, A., 1998. A deterministic approach to the end-to-end analysis of packet flows in connection-oriented networks. *IEEE/ACM transactions on networking*, *6*(4), pp.422-431.

[Cruz 1995] Cruz, R.L., 1995. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected areas in Communications*, *13*(6), pp.1048-1056.

[De Azua – Boyer 2014] De Azua, J.A.R. and Boyer, M., 2014, October. Complete modelling of AVB in Network Calculus Framework. In *RTNS*(p. 55).

[Gamarnik 1998] Gamarnik, D., 1998, November. Stability of adversarial queues via fluid models. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)* (pp. 60-70). IEEE.

[Guillemin et al 1992] Fabrice P. Guillemin, Pierre E. Boyer, Michel J. Servel. The spacer-controller: an efficient UPC/NPC for ATM networks. ISS '92, Session A9.3, volume 2, Oct 1992.

[Jiang 2018] Jiang, Yuming, "A basic result on the superposition of arrival processes in deterministic networks", arXiv preprint arXiv:1804.10973

[Le Boudec 1996] Le Boudec, Jean-Yves, “Network Calculus Made Easy”, EPFL-DI Research Report 96/218, Dec 1996.

[Le Boudec 2002] Le Boudec, J.Y., 2002. Some properties of variable length packet shapers. *IEEE/ACM Transactions on Networking (TON)*, *10*(3), pp.329-337.

[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 (TON)*, *26*(6), pp. 2721-2733.

[Le Boudec 2018-slides] "Network Calculus -- Regulators from Integrated Services to Deterministic Networking“, Keynote Lecture - International Workshop on Network Calculus and Applications (NetCal 2018), Vienna, Sept 7, 2018, online at http://icawww1.epfl.ch/PS_files/KeyNoteITC2018LEB.pdf.

[Le Boudec-Thiran 2001] Le Boudec, Jean-Yves, and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Vol. 2050. Springer Science & Business Media, 2001, online at http://ica1www.epfl.ch/PS_files/NetCal.htm

[Liebeherr 2017] Liebeherr, J., 2017. Duality of the Max-Plus and Min-Plus Network Calculus. *Foundations and Trends® in Networking*, *11*(3-4), pp.139-282.

[Rizzo-Le Boudec 2007] Rizzo, G. and Le Boudec, J.Y., 2007, November. Generalization of the RIN result to heterogeneous networks of aggregate schedulers and leaky bucket constrained flows. In *Networks, 2007. ICON 2007. 15th IEEE International Conference on* (pp. 388-393). IEEE.

[Rizzo-Le Boudec 2008] Rizzo, G. and Le Boudec, J.Y., 2008, April. Stability and delay bounds in heterogeneous networks of aggregate schedulers. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (pp. 1490-1498). IEEE.

[Specht-Samii 2016] Specht, J. and Samii, S., 2016, July. Urgency-based scheduler for time-sensitive switched ethernet networks. In *Real-Time Systems (ECRTS), 2016 28th Euromicro Conference on* (pp. 75-85). IEEE.

[Tassiulas-Georgiadis 1996] Tassiulas, L. and Georgiadis, L., 1996. Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACM Transactions on Networking (TON)*, *4*(2), pp.205-208.

[Verma et al 1991] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. Tricomm ’91, Chapel Hill, pages 35–46, April 1991.