

Application of Network Calculus to the TSN Problem Space

Jean-Yves Le Boudec^{1,2,3}

EPFL

IEEE 802.1 Interim Meeting

22-27 January 2018

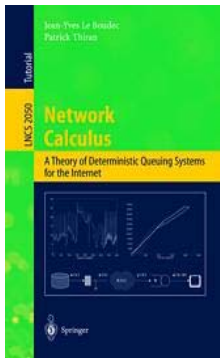
¹ <https://people.epfl.ch/105633/research>

² <http://smartgrid.epfl.ch>

³ IP Parallel Redundancy Protocol <https://github.com/LCA2-EPFL/iprp>

What is Network Calculus ?

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



C.S. Chang, R. Cruz, JY Le Boudec, P. Thiran, ...

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

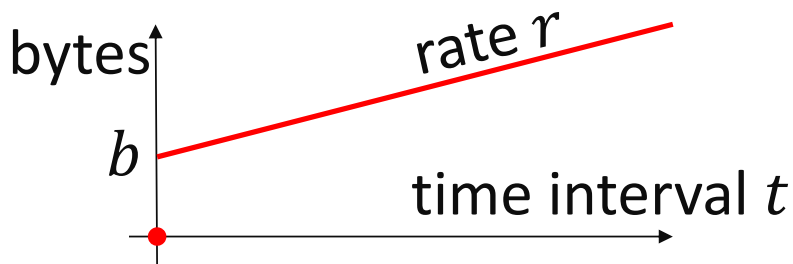
Derive system equations $\Rightarrow$ formal proofs

Arrival Curve

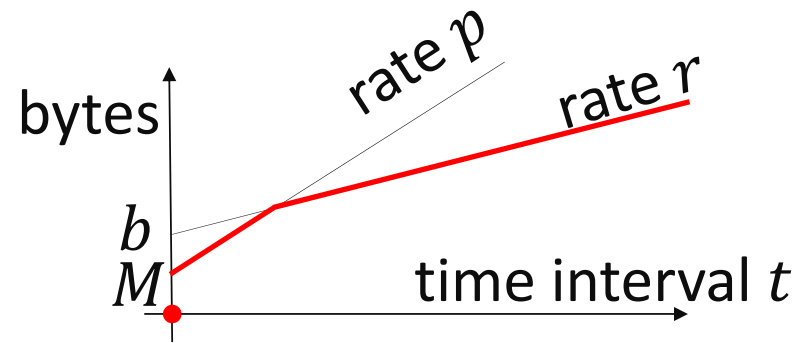
For a flow, at an observation point

Flow is constrained by arrival curve $\alpha()$ iff the amount of basic data units (e.g. bytes) observed in *any interval* of duration t is $\leq \alpha(t)$

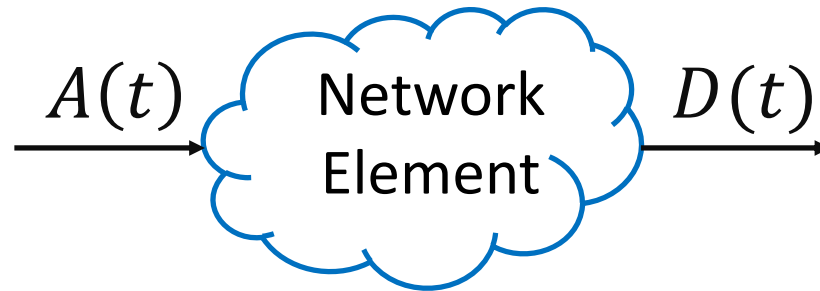
token bucket with rate r and burst b : $\alpha(t) = rt + b$



token bucket + peak rate p and MTU M : $\alpha(t) = \min(pt + M, rt + b)$



Service Curve



$A(t), D(t)$: amount of basic data units observed in $[0, t]$

Network element offers to this flow a service curve $\beta()$ if

$$\forall t \geq 0, \exists s \in [0, t]: D(t) \geq A(s) + \beta(t - s)$$

Service Curve Example

Rate-latency service curve :

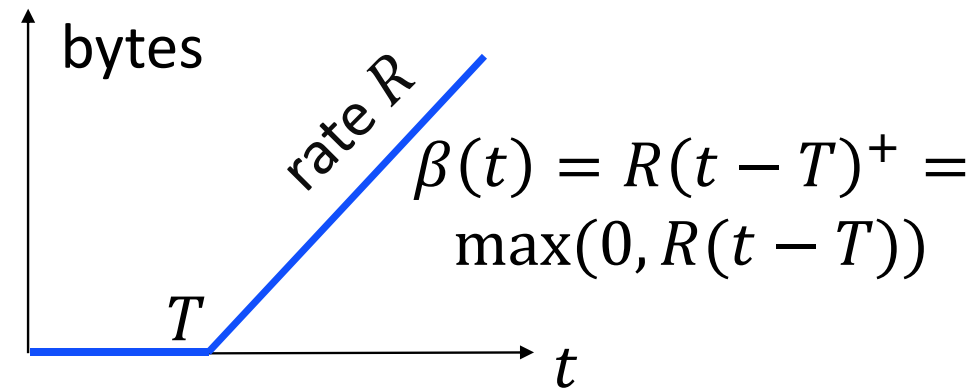
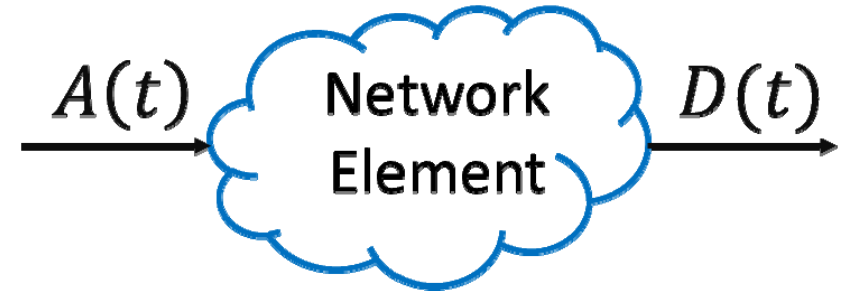
$$\beta(t) = R(t - T)^+$$

Models many schedulers: DRR, PGPS, RFC 2212, etc.

Example: service received by a high priority flow (no pre-emption):

R = line rate

$T = \frac{1}{R} \times MTU$ of low priority packets



$$D(t) = A(s) + \beta(t - s)$$

for some s
(e.g. s = beginning of busy period)

Service Curve Example: Non Pre-emptive Priority

One server of rate C , two priorities

Every high priority flow f is leaky bucket controlled (r_f, b_f)

Let $r_H = \sum_{f \in H} r_f$, $b_H = \sum_{f \in H} b_f$

Then if $r_H < C$ the aggregate of all **low priority** flows receives a rate-latency service curve $\beta(t) = R(t - T)^+$ with

$$R = C - r_H, T = \frac{b_H}{C - r_H}$$

[Le Boudec and Thiran 2001, Section 1.3]

Service Curve Example: AVB / CBS

The aggregate of all flows (streams) served in one server as AVB **class A** receives a rate-latency service curve $\beta(t) = R(t - T)^+$ with

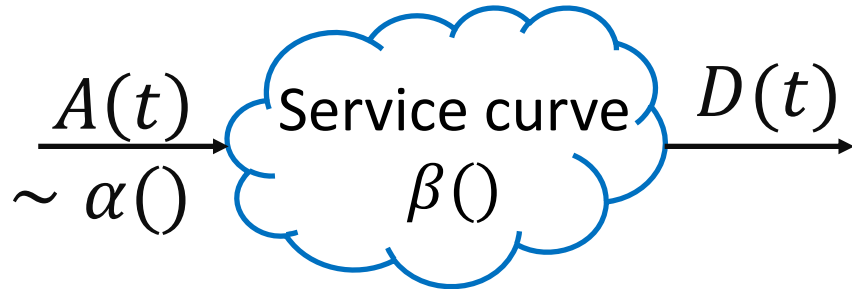
$$R = \frac{id_A C}{id_A - sd_A}, T = \frac{l_{\max}^n}{C}$$

Similar results for class B

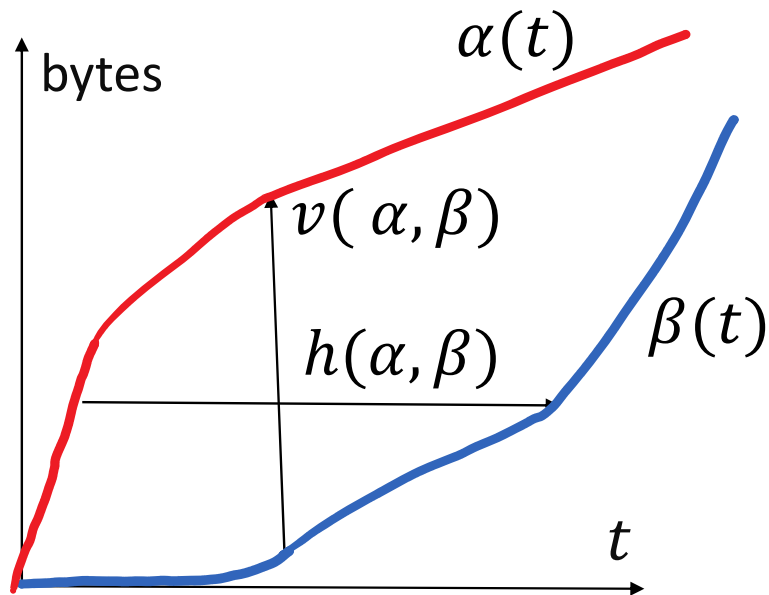
[Ruiz-Boyer 2014]

id_A = idle slope, sd_A = send slope, $l_{\max}^n$ = max packet size other than class A

Basic Results: 3 Tight Bounds

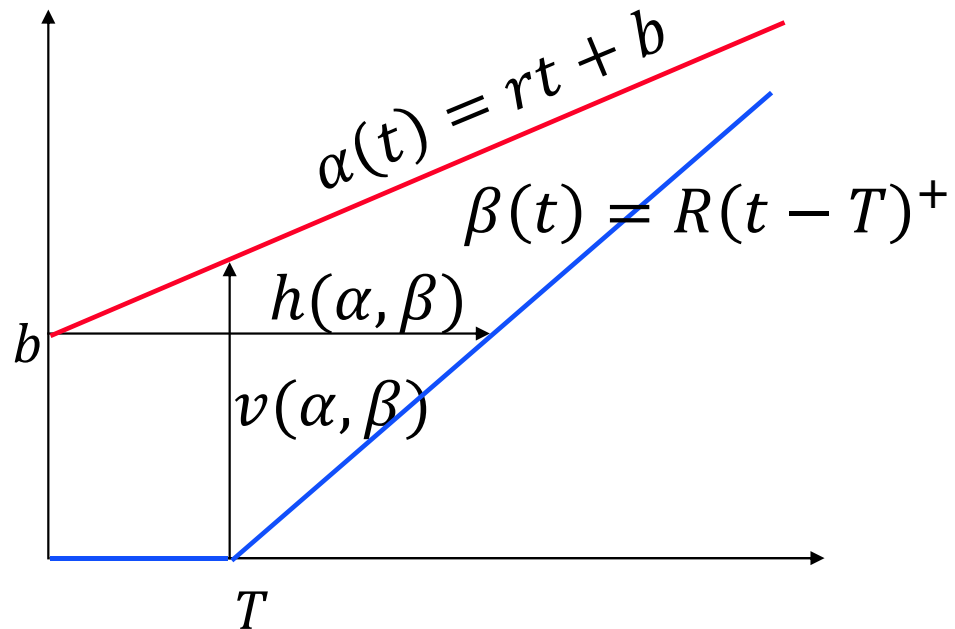


Flow is constrained by arrival curve $\alpha()$; served in network element with service curve $\beta()$. Then



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

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

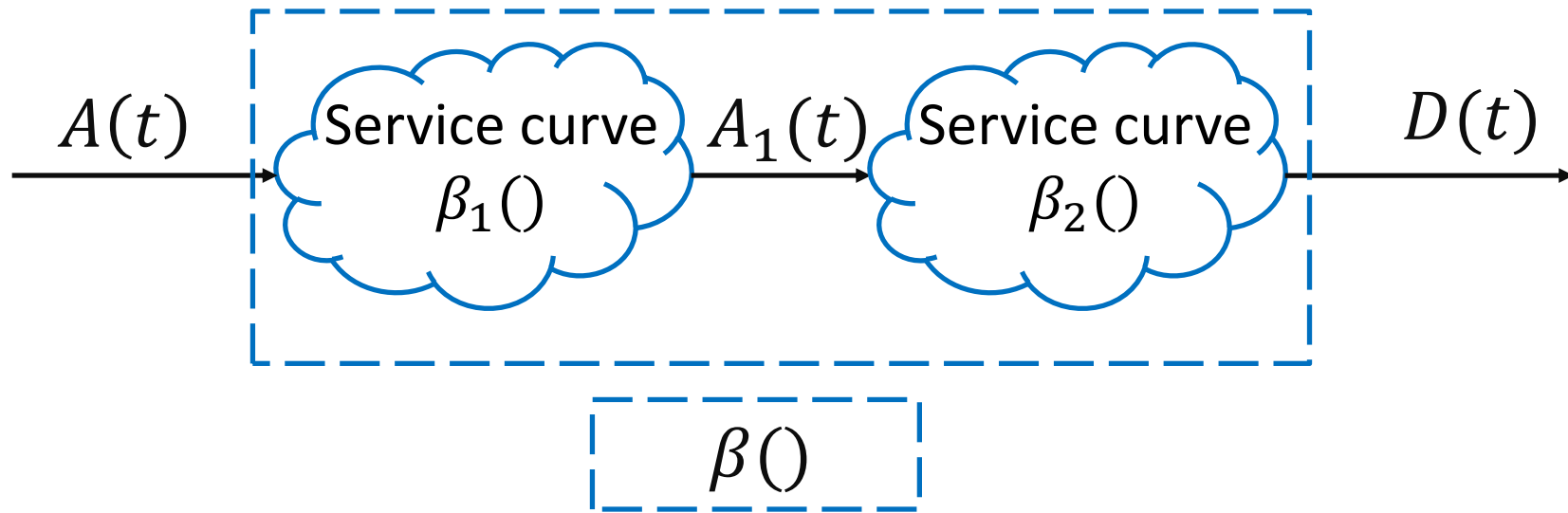
Output arrival curve:

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

$$\text{with } b^* = b + rT$$

(burstiness b is increased by rT)

Concatenation



A flow is served in series, network element i offers service curve $\beta_i()$. The **concatenation** offers the service curve $\beta()$ defined by

$$\beta(t) = \inf_{s \geq 0} (\beta_1(s) + \beta_2(t - s))$$

Min-Plus Convolution

$$\beta(t) = \inf_{s \geq 0} (\beta_1(s) + \beta_2(t - s))$$

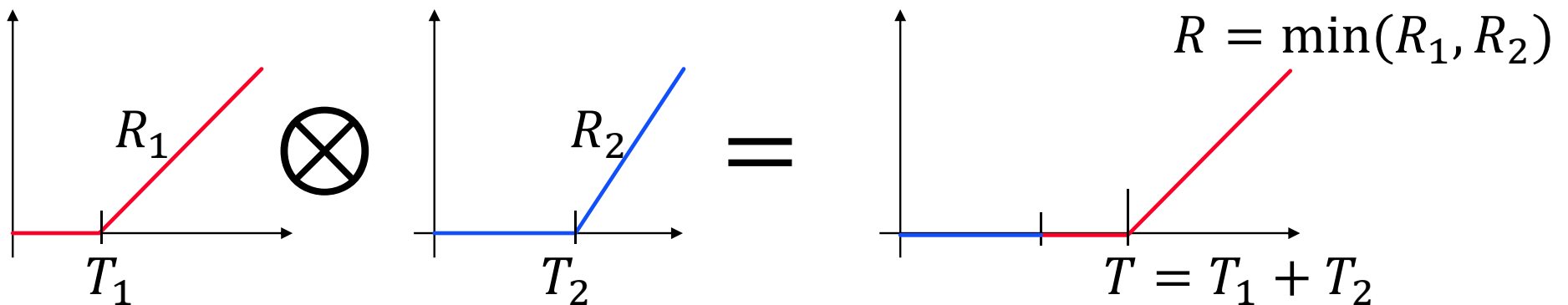
$$\beta = \beta_1 \otimes \beta_2$$

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

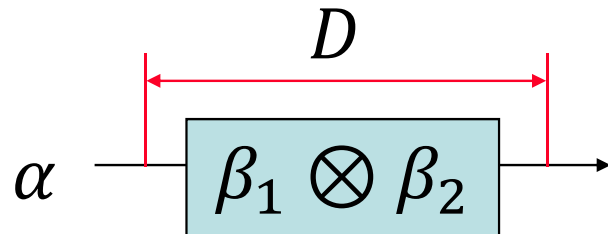
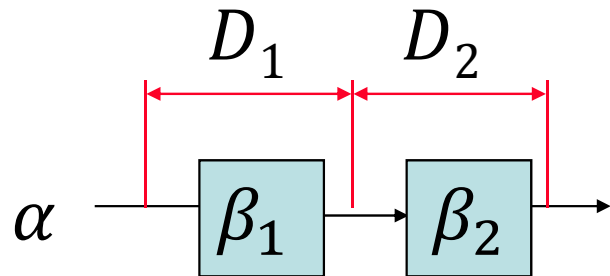
$$(\beta_1 \otimes \beta_2) \otimes \beta_3 = \beta_1 \otimes (\beta_2 \otimes \beta_3)$$

$$\beta_1 \otimes \beta_2 = \beta_2 \otimes \beta_1$$

It can be computed easily: e.g.



Pay Bursts Only Once



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

one flow constrained *at source* by $\alpha()$
end-to-end delay bound computed
node-by-node (also accounting for
 increased burstiness at node 2):

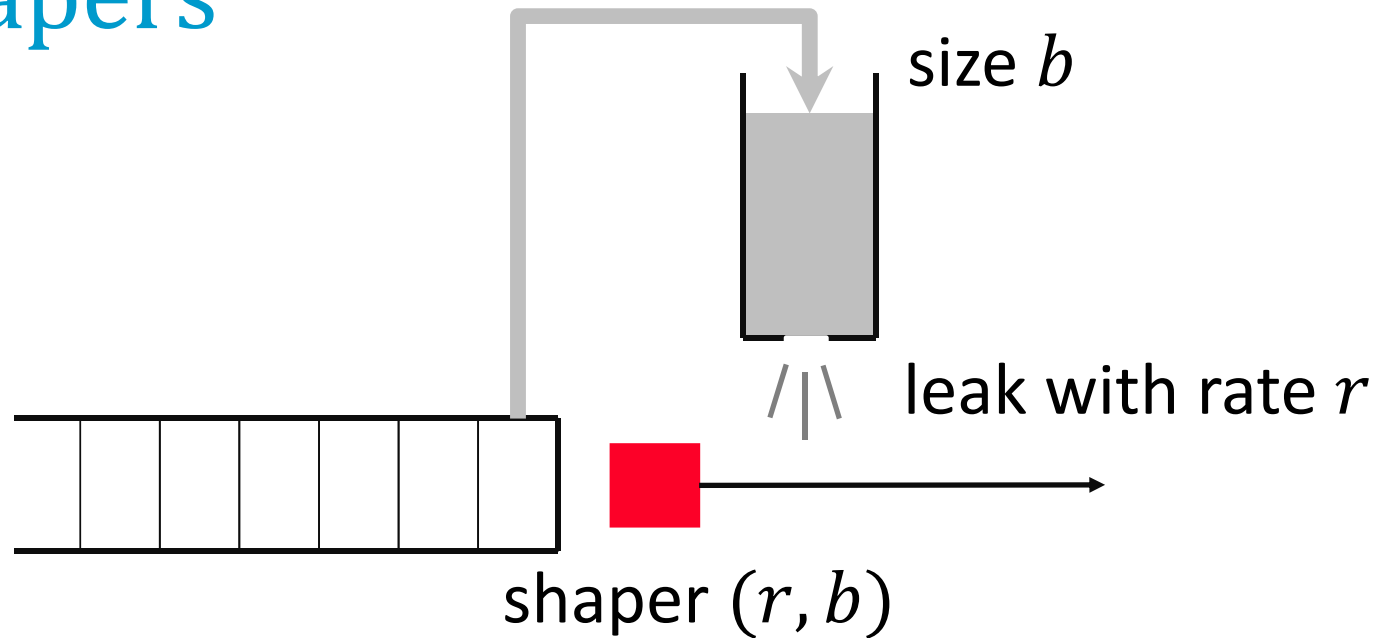
$$D_1 + D_2 = \frac{2b + RT_1}{R} + T_1 + T_2$$

computed *by concatenation*:

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

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

Shapers

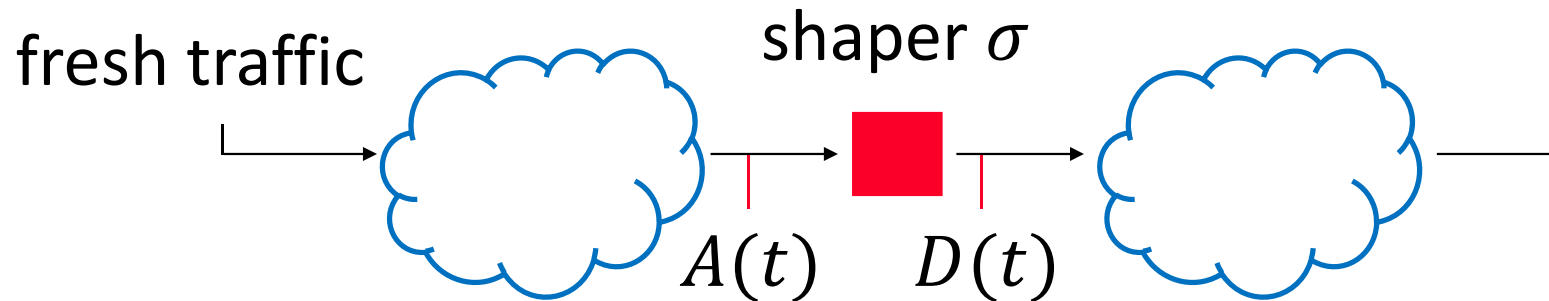


Burstiness increases as flows traverse network elements

Shapers are used to reduce burstiness

Example: **leaky bucket** shaper (r, b) releases a packet only if there is space to put an equivalent amount of fluid into bucket

The Mathematics of Per-Flow Shapers



For leaky bucket flow shaper

min-plus equation:

$$D(t) = \min_{0 \leq s \leq t} (A(t-s) + rs + b) \quad (1)$$

max-plus equation

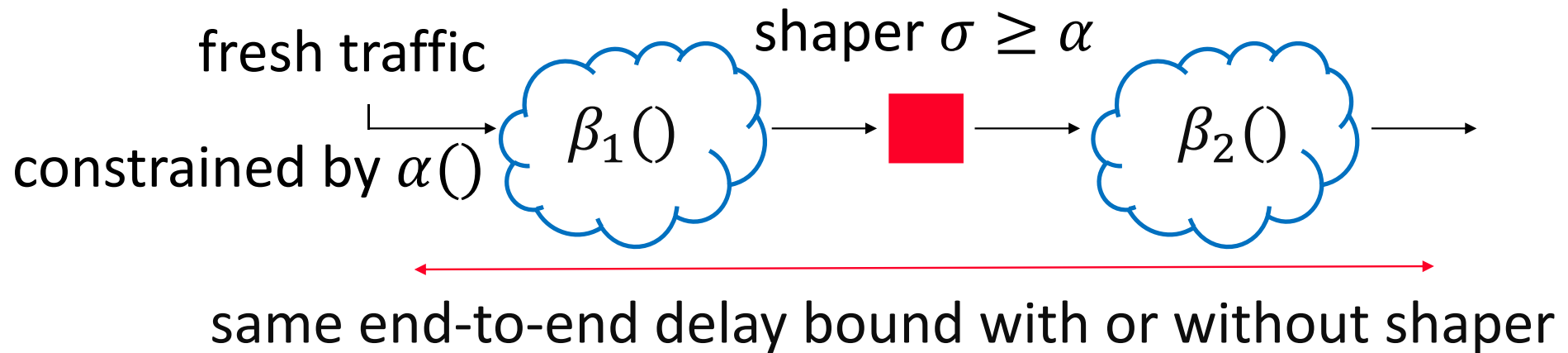
$$D_n = \max_{1 \leq m \leq n-1} \left(A_m + \frac{L_m + \dots + L_n - b}{r} \right) \quad (2)$$

arrival time of n^{th} packet, D_n departure time (at shaper)

$A(t)$ = total nb of bytes seen on arrival at shaper in $[0, t]$, $D(t)$ seen on departure

Per-Flow Re-Shaping is For Free

Per-flow re-shaping does not increase worst-case end-to-end delay
(per flow = (TSN) per stream)



Per Class Networks

A set S of flows, each constrained by leaky bucket r_f, b_f are aggregated into one class; $r_{tot} = \sum_{f \in S} r_f, b_{tot} = \sum_{f \in S} b_f$

At one node, this class receives a rate-latency service curve R, T (e.g. priority node, DRR, AVB, CBS). FIFO inside the class; $r_{tot} \leq R$

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

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

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

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

[Le Boudec-Thiran 2001, Section 6.4]

Cascading Burstiness

Unlike in a per-flow network, in a per-class network with FIFO inside every class, 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**).

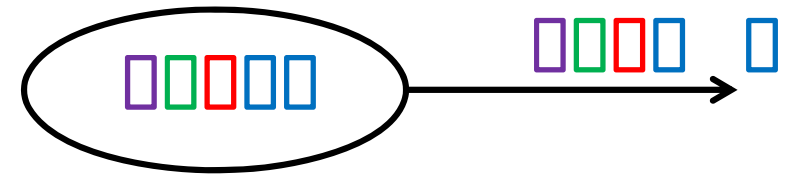
Good delay bounds depend on the topology and on the number of hops. In non-tree topologies, delay bounds are generally bad even at low utilizations and small numbers of hop.

[Bennett et al 2002]

Avoiding Cascading Burstiness in Per-Class Networks

Solution 1: re-shape every flow at every hop (per-flow shaping)
Solves the problem but defeats the purpose of per-class network.

Solution 2: **Interleaved shaper**

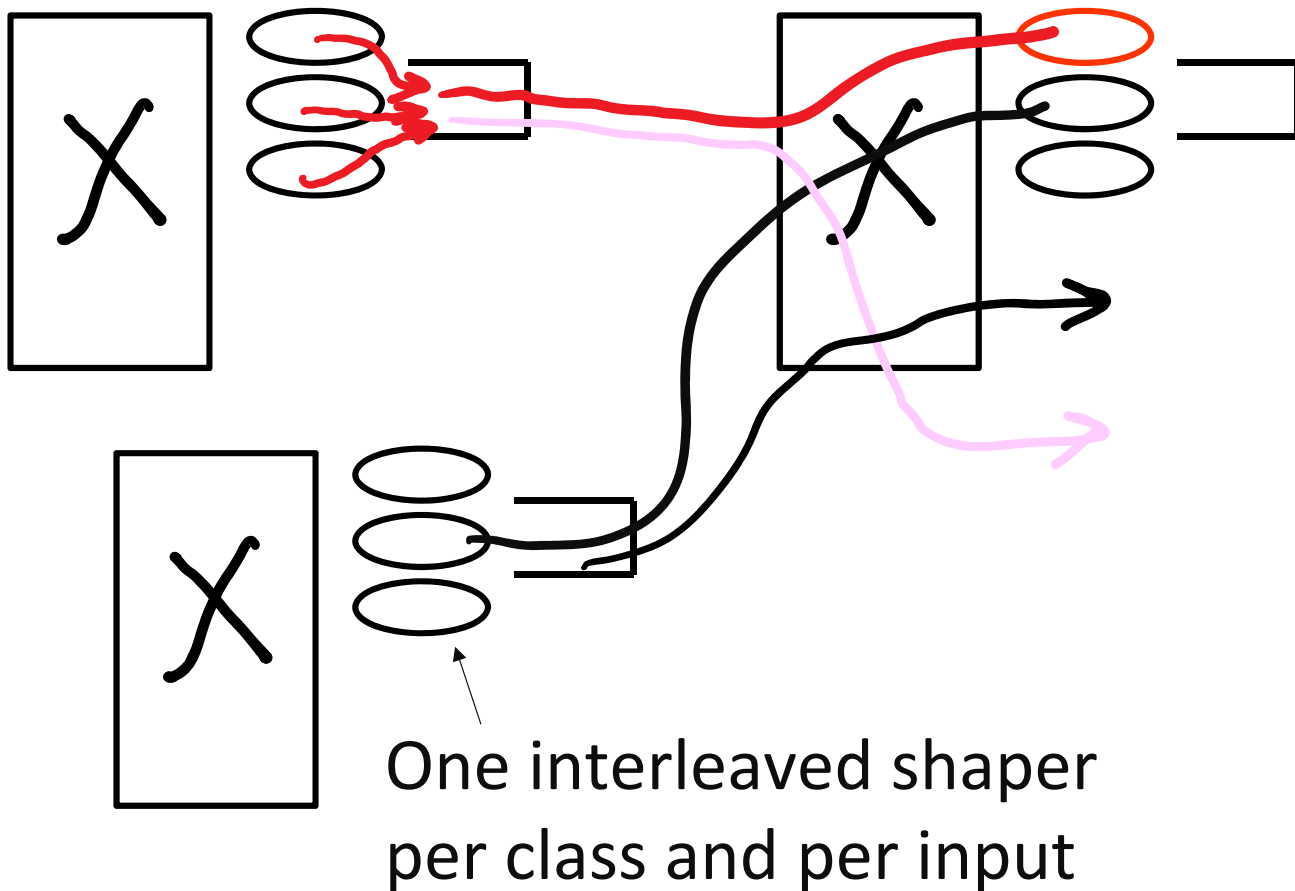


- FIFO queue of all packets of all flows in class
- packet at head of queue is examined versus arrival curve of its flow; this packet is delayed if it came too early
- packets not at head of queue wait for their turn to come

[Fecht-Samii 2016]

Network With Interleaved Shaper [Fecht-Samii 2016]

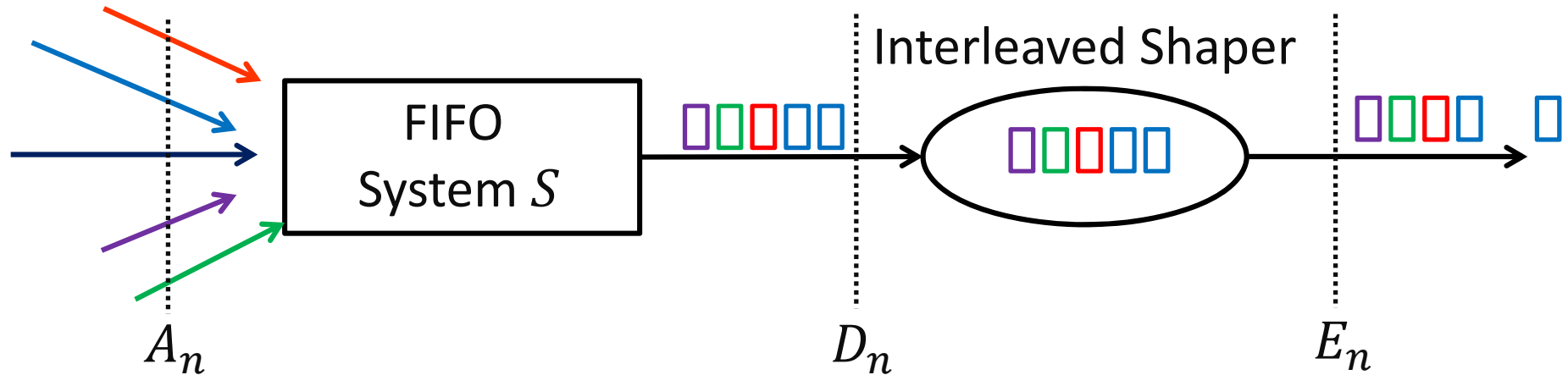
Interleaved Shaper



Output of interleaved shaper is shaped per flow
⇒ delay bound at next server is controlled – no cascading

Question:
what is the **delay**
due to interleaved shaper ?

Interleaved Shaping Does Not Increase Worst Case Delay



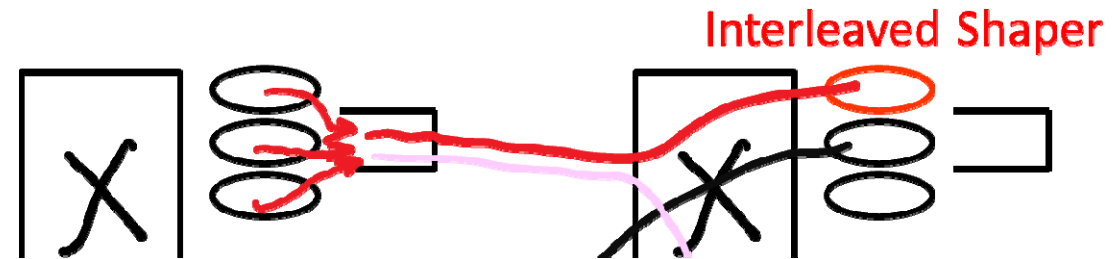
A_n = arrival time of n^{th} packet (numbered in arrival order)

Every flow f is shaped before input to S with parameters (r_f, b_f)

Output of S is fed to interleaved shaper with parameters (r_f, b_f) for flow f

Theorem: $\sup_n (D_n - A_n) = \sup_n (E_n - A_n)$ [Proof in appendix]

Implications for TSN



- Worst case **end-to-end** queuing delay can ignore interleaved shapers. Delay bound at one interleaved shaper is absorbed by delay at previous hop
- Queuing delay at **every hop** (without shaper) can be computed with e.g. by $D = \frac{b_{tot}}{R} + T$ where R, T is rate-latency service curve allocated to the class and b_{tot} is the sum of burstiness enforced by the interleaved shaper at this node.
- Worst case delay at **one node** cannot ignore interleaved shaper.
⇒ Worst case end-to-end delay is generally less than sum of per-hop delays.

Conclusions

Network Calculus can

- ▶ help understand some **physical properties** of deterministic networks (e.g. pay bursts only once, per flow reshaping does not increase end-to-end delay bound, interleaved shaper can delay can be ignored)
- ▶ provide **formal** guarantees on extreme delays that are hard to reach by simulation or by ad-hoc analysis,
- ▶ provide a simple language to **abstract** a node without prescribing an implementation.

Future Work ?

Obtain service curve characterization of TSN/other schedulers and shapers.

Formally prove end-to-end bounds.

Quantify of improvement to end-to-end delay-bounds by exporting service curves instead of per-node delay-bounds.

Explore implications for path computation and setup (distributed, centralized).

Propose and test abstract node models.

References

Textbook: [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, legally and freely available online at http://ica1www.epfl.ch/PS_files/NetCal.htm

Ultra-short tutorial: [Le Boudec-Thiran 2000] Le Boudec, J-Y., and Patrick Thiran. "A short tutorial on network calculus. I. Fundamental bounds in communication networks." *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*. Vol. 4. IEEE, 2000, also at http://www.cse.cuhk.edu.hk/~cslui/CSC6480/nc_intro1.pdf

Path Computation [Frangioni et al 2014] Frangioni, A., Galli, L., & Stea, G. (2014). Optimal joint path computation and rate allocation for real-time traffic. *The Computer Journal*, 58(6), 1416-1430.

Per-flow networks [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.

References

Automotive: [Queck 2012] Queck, Rene. "Analysis of ethernet avb for automotive networks using network calculus." *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*. IEEE, 2012.

Worst case bounds for class based queuing: [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

AVB [Ruiz-Boyer 2014] Ruiz, J.A. and Boyer, M., 2014, October. Complete modelling of AVB in network calculus framework. In *Proceedings of the 22nd International Conference on Real-Time Networks and Systems* (p. 55).

TSN [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.

Proof of Theorem 1

The interleaved shaper releases packet n at time $E_n = \max \{D_n, E_{n-1}, F_n\}$, where F_n is the earliest possible time allowed by the leaky bucket constraints (see Eq. (2)):

$$F_n = \max_{1 \leq j \leq k-1} \left(E_{i_j} + \frac{L_{i_j} + \dots + L_{i_k} - b_f}{r_f} \right)$$

with: $i_1, i_2, \dots, i_k = n$ indices of packets of flow f in the global packet sequence and f = flow of packet n . Using the operator notation $\vee$ for maximum, this gives

$$E_n = D_n \vee E_{n-1} \vee \left(E_{i_1} + \frac{L_{i_1} + \dots + L_{i_k} - b_f}{r_f} \right) \vee \dots \vee \left(E_{i_{k-1}} + \frac{L_{i_{k-1}} + L_{i_k} - b_f}{r_f} \right) \quad (3)$$

Call d the worst case delay at FIFO system S and prove by induction on n that $E_n \leq d + A_n$. Obviously this holds for $n = 1$ (first packet is not delayed by shaper). Now assume $E_m \leq d + A_m$ for $m < n$. We will prove that every terms in right-handside of (3) is $\leq d + A_n$.

- the first term is D_n ; $D_n \leq d + A_n$ by definition of d
- the second term is E_{n-1} ; $E_{n-1} \leq d + A_{n-1}$ by induction hypothesis; furthermore, $A_{n-1} \leq A_n$ by definition of the arrival times A_n ; thus $E_{n-1} \leq d + A_n$
- the third and following terms are of the form $E_{i_j} + \frac{L_{i_j} + \dots + L_{i_k} - b_f}{r_f}$; now $E_{i_j} \leq d + A_{i_j}$ by induction hypothesis. Furthermore, the input to system S is leaky-bucket constrained per flow. Thus (by Eq (2)):

$$A_{i_j} + \frac{L_{i_j} + \dots + L_{i_k} - b_f}{r_f} \leq A_{i_k} = A_n$$

It follows that the third term is $\leq d + A_{i_j} + \frac{L_{i_j} + \dots + L_{i_k} - b_f}{r_f} \leq d + A_n$

QED