Analysing TSN within network calculus framework

Marc Boyer

École d'Été Temps Réel 2021
23 septembre 2021

Outline

What is TSN ?

TSN is not a technology

- TSN is the name of a IEEE task group of the IEEE 802.1 Working Group
  - TSN : Time-Sensitive Networking
  - http://ieee802.org/1/pages/tsn.html
  - https://1.ieee802.org/
- Documents : Naming : 802.1Q, 802.1ad, and 802.1Qat...
  From one up to even four letters after 802.1
  - Uppercase : standards
  - Lower-case : amendments
  - -REV : revision (more extensive changes to the existing text than can be undertaken in an amendment)
- Document Access :
  - Working documents : need to be member (≈)
  - Published standard :
    - ≈ free after 6 months : "IEEE Standards runs a Get IEEE802 program that allows anyone to download the standards for free, 6 months after publication."
    - Or buy it

TSN promises

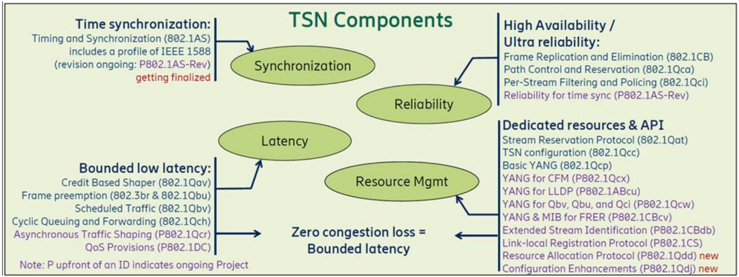


Figure – TSN Overview, J. Farkas [3]

Outline

An Ethernet network

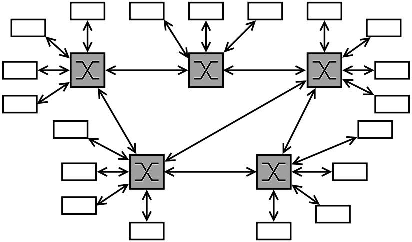


Figure – Principle of Ethernet network (switch-based)

- full duplex links
- propagation delay : signal transmission (≈ 60% light speed)
- main delay : in switches
- routing, frame format : lack of time

An Ethernet switch

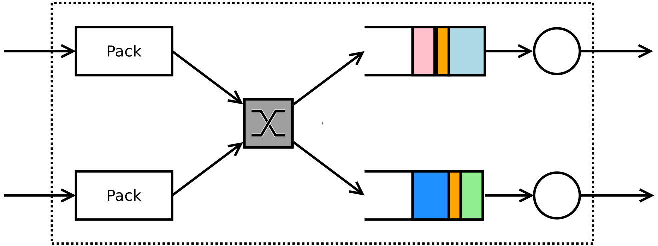


Figure – Common architecture of Ethernet switch

- input ports : frame arrivals
- switching : copy in destination port(s)
- output port : queuing + transmission

An 8 priority level Ethernet switch

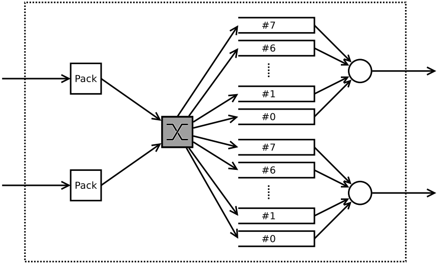


Figure – Ethernet switch with priority levels

- non-preemption : up to 1542B blocking
- preemption (802.3br, 802.1Qbu) :
  - partial blocking (up to 148 B) + overhead
  - single-level preemption

Outline

ONERA
THE FRENCH AEROSPACE LAB

Outline

Main TSN addenda

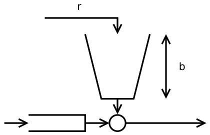
- Frame preemption (802.3br, 802.1Qbu)
- Synchronisation mechanisms (algorithms, architecture, protocols) 802.1AS-Rev
- Resource reservation, access control, configuration, signalisation, stream identification (802.1Qat, 802.1Qcc, 802.1CBdb, 802.1Qca, 802.1Qdd...)
- Safety and reliability :
  - Input port policing : 802.1Qci
  - Redondancy : 802.1CB
- Output port scheduling :
  - Credit Based Shaper, CBS (802.1Qav)
  - Scheduled Traffic (802.1Qbv)
  - Cyclic Queuing and Forwarding (802.1Qch)
  - Asynchronous Traffic Shaping, ATS (802.1Qcr)
  - ETS for bandwidth sharing (802.1Qaz, pre-TSN)

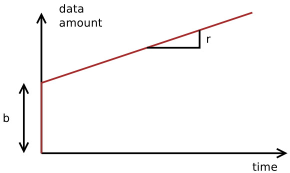
Outline

The token-bucket model

- two parameters :
  - throughput r,
  - burst b (aka capacity, depth)
- the bucket rules
  - the bucket is initially full of b tokens
  - sending a frame of size s consumes s tokens
  - the bucket fills with rate r tokens per time unit
  - can never be negative nor exceed b
- in case of insufficient tokens
  - drop the frame : policing
  - queue until enough : shaping
- property : on *any* observation interval of duration d, the data amount is less than

$$b + d \cdot r \qquad (1)$$

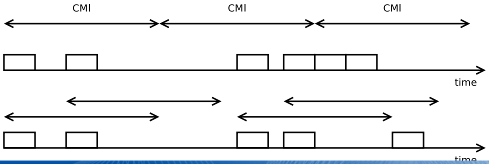




- a periodic flow with frames of size S and period T respects token-bucket $b = S$, $r = S/T$

Flows contract

- notion of *stream*
- several "traffic specification"
- the AVB stream traffic specification
  - Traffic Specification associated with a Stream [1, § 35.2.2.8.4 TSpec]
    - MaxFrameSize : the maximum frame size
    - MaxIntervalFrames : the maximum number of frames that the Talker may transmit in one "class measurement interval" (34.4).
    - Class Measurement Interval (CMI) : static, per class (in 0-7)
  - Semantics : tumbling window vs. sliding window
    TSpec as token-bucket
    - sliding $\implies r^s = r, b^s = b$
    - tumbling $\implies r^t = r, b^t = 2b$



$$b = \mathsf{MFS} \cdot \mathsf{MIF}$$

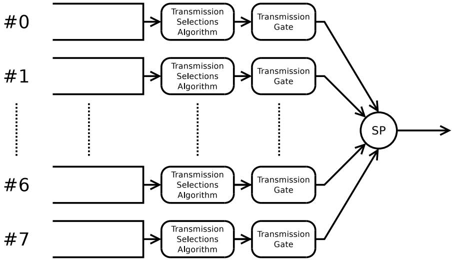
$$r = \frac{b}{\mathsf{CMI}}$$

Input port policing : 802.1Qci

- 802.1Qci : Per-Stream Filtering and Policing – PSFP
- done at input port
- associates a token-bucket to a (configurable) set of streams
- drop “out of contract” frames

Outline

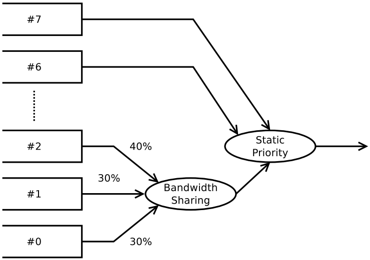
TSN output port



- Transmission Selection Algorithm :
  - per queue choice
  - one in "none, CBS, ATS, ETS"
- Transmission gate :
  - a gate is either open or closed
  - based on a static cyclic schedule

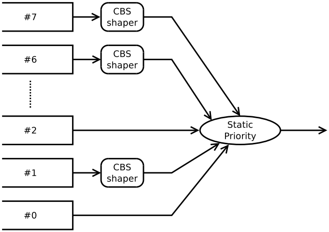
802.1Qaz : Bandwidth Sharing (SP/WRR, SP/DRR... – pre-TSN)

- Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes (aka ETS)
- 802.1Qaz, 2011 (pre-TSN)
- Simple hierarchical scheduling : Static priority + Round-Robin-like
- Introduced for data centers
- Sharing the leftover bandwidth
- Bandwidth Sharing is implementation-defined
  - WRR cited in the standard
  - DRR used in Linux
  - not able to find choice of Cisco, Juniper...

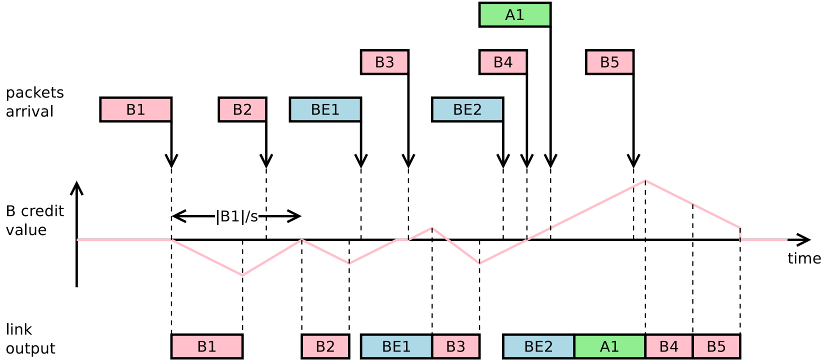


802.1Qav : Credit-Based Shaped (CBS)

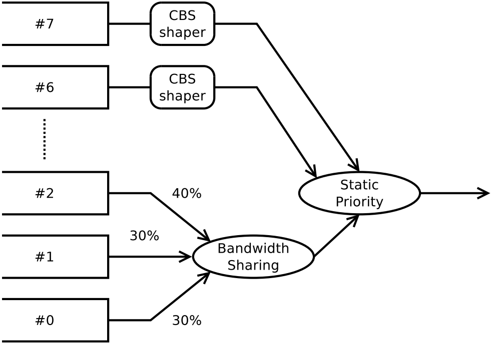
- "Forwarding and Queuing Enhancements for Time-Sensitive Stream – FQTSS"
- 802.1Qaz, 2011 (AVB, pre-TSN)
- CBS shaper is optionnal
- Each CBS shaper has a "slope" s parameter (in bit per second)
- A credit increases when the queue waits, and decreases when the queue transmits
- It limits the associated queue to throuput s
- Its shapes/spreads/smoothes the output
- Designed to
  - avoid starvation
  - limit jitter



Example of CBS credit evolution rule

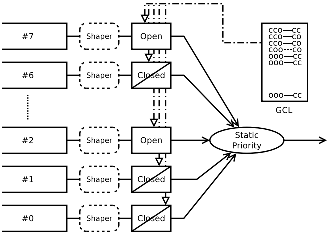


802.1Qaz + 802.1Qav : ETS+CBS

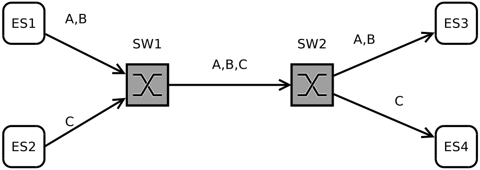


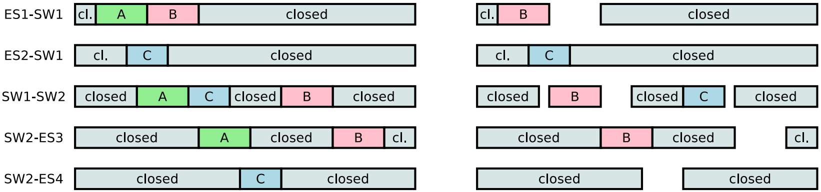
802.1Qbv : Time Aware Shaper – TAS

- “Enhancements for Scheduled Traffic”
- A *gate* is associated to each queue
- The gate is either open or closed
- A global cyclic schedule (Gate Control List – GCL), w.r.t local clock
- Building schedule is out of standard
- “Exclusive gating” ≈ one gate opened at a time
- Integration with GCL : update of credit evolution rules
- End-to-end TT schedule requires
  - global build of local schedules
  - synchronisation of local clocks (eg. 802.1AS)



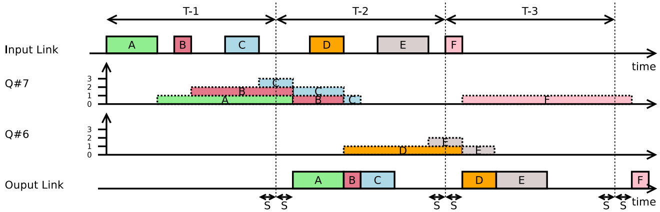
TAS : a Time-Trigerred implementation ?





802.1Qch : Cyclic Queuing and Forwarding – CQF

- Not a new “mechanism” : based on 802.1Qci (Filtering) and 802.1Qbv (Time Aware Shaper)
- Divide time into time intervals of common length T
- Frames received in one interval are forwarded in the next one

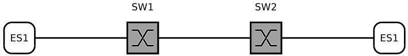


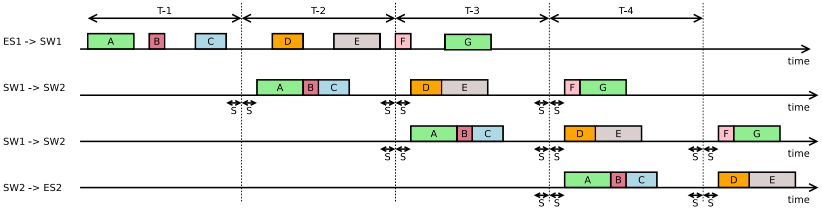
CQF performances

- Global synchronisation

  ⟹ Low jitter (2T)

  ⟹ simple delay computation ($T \times$ nb of hops)

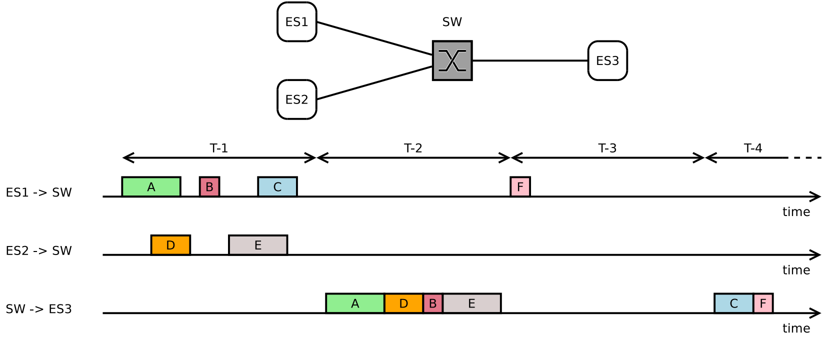




ONERA THE FRENCH AEROSPACE LAB

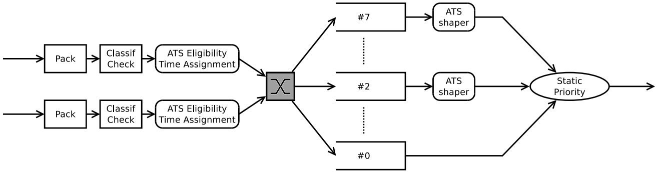
CQF configuration

- Cycle time must be “large enough” w.r.t. bursts



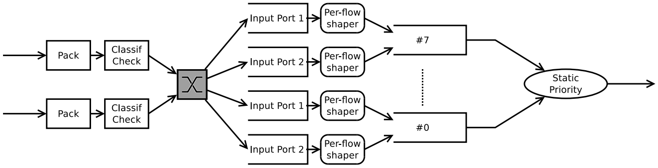
802.1Qcr : Asynchronous Traffic Shaping – ATS

- Queue waiting create bursts / jitter
- ATS introduces delay to absorb the jitter
  - computes a “Eligibility Time” per frame
    - a local value (no global synchronisation)
    - token-bucket parameters
    - use some share variables between ATS schedulers
  - head of queue can not be selected before this Eligibility Time



ATS : implementation and equivalent model

- Complexity relies in computattion of "Eligibility Time"
- Computed in order to be equivalent to group reshaping (token bucket) with aggregate queuing
- A major theoretical breakthrough
  - reshaping comes for free
  - avoid cyclic dependency problem



Outline

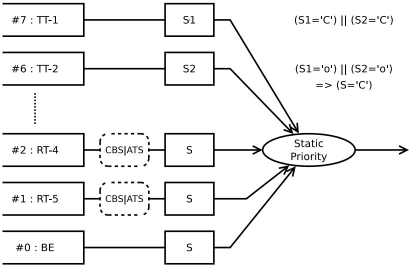
The most obvious one : TT/Shaper/BE

- TT queues : for very low latency and jitter flows
- CBS|TAS queues : for real time
- Best Effort

Principles :

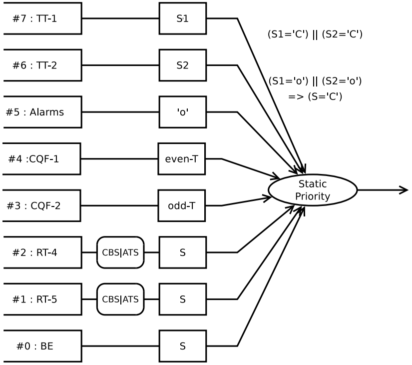
- build TT queue GCL wrt TT behaviour, no shaper for TT queues
- set other GCL queues as the opposite (exclusive gating)
- set BE at lower priority
- configure CBS or ATS wrt expected workload

Rq : exclusive gating allows TT files to use any priority level.



With alarms and CQF

- TT queues : for very low latency and jitter flows
- Static priority : for asynchronous alarms
- CQF
- CBS|TAS queues : for real time
- Best Effort



Outline

TSN conclusion

- the next real-time network
- a lot on industry involved
- able to host several kinds of flows
- offering several scheduling policies
- how to configure it ?
- how to bound buffer usage and delay ?

Outline

What is network calculus ?

- another theory for real-time (computes response-time bound)
- based on (min,plus) dioid theory
- used to certify AFDX network in A380, A400M, etc.
- several tools (e.g. RTaW-PEGASE)
- share common aspects with Event Stream theory [6]

Why not using scheduling analysis ?

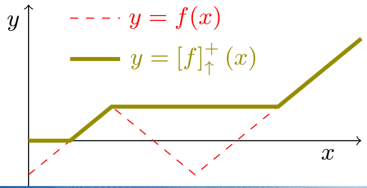
because :

- diversity of approaches is beneficial
- input models are different
  - shaping, serialization
- worst end-to-end delay is not the sum of local worst delays

Notations

- $\mathbb{R}$: the set of real numbers, $\mathbb{R}^+$ the subset of non-negative real numbers,
- $\mathbb{Z}$ the set of integers,
- $\lceil\cdot\rceil : \mathbb{R} \to \mathbb{Z}$ the ceiling function ($\lceil 1.2 \rceil = 2, \lceil 4 \rceil = 4, \lceil -1.2 \rceil = -1$)
- $a \wedge b = \min(a, b)$
- $\forall x \in \mathbb{R}, [x]^+ = \max(x, 0)$
- $\forall f : \mathbb{R}^+ \to \mathbb{R}$, its non-decreasing non-negative closure is defined by

$$[f]^+_{\uparrow}(t) = \max_{0 \le s \le t} [f(s)]^+ . \qquad (2)$$



Outline

Modeling data flows

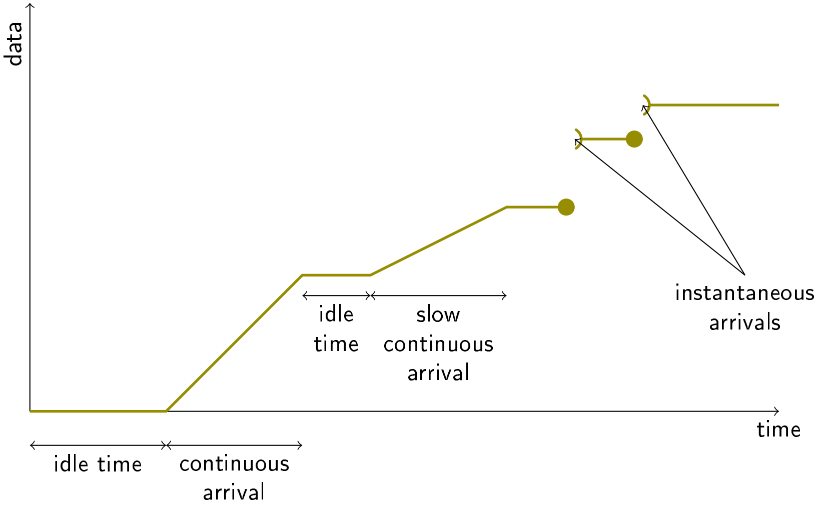
Definition : Cumulative curve

$\mathcal{F} = \{f : \mathbb{R}^+ \to \mathcal{R}\}$ $\mathcal{C} \subset \mathcal{F}$ is the subset

- non-negative
- non-decreasing
- piece-wise continuous
- left-continuous

- An element $A \in \mathcal{C}$ is used to model a data flow in the network
- A is called "cumulative curve"
- $A(t)$ represent the amount of data the amount of data from a flow observed at some point up to time t
- A lot of information lost

Example


data
time
idle time
continuous arrival
idle time
slow continuous arrival
instantaneous arrivals

Modeling server

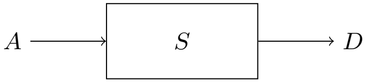
Definition : Serveur

A server is a relation $S \subseteq \mathcal{C} \times \mathcal{C}$,

- left-total ($\forall A \in \mathcal{C}, \exists D \in \mathcal{C} : (A, D) \in S$)
- $\forall (A, D) \in S : A \geq D$
- $A \xrightarrow{S} D \equiv (A, D) \in S$

Semantics

- $A(t)$: amount of data arrived into S up to t
- $D(t)$: amount of data departed from S up to t
- Departure after arrival $\implies D \leq A$
- No loss, no creation, compression, etc.



Measures : delay and backlog

Definition : Delay et backlog

Let S a server. Define

$$d(A,D,t) = \inf \left\{ d \in \mathbb{R}^+ \;\middle|\; A(t) \leq D(t+d) \right\}, \tag{3}$$

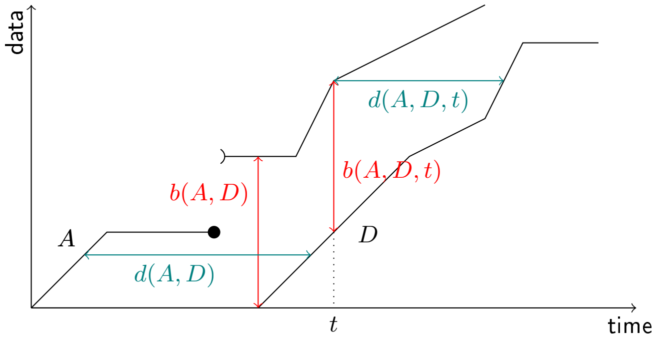
$$d(A,D) = \sup_{t \geq 0} d(A,D,t), \tag{4}$$

$$b(A,D,t) = A(t) - D(t), \tag{5}$$

$$b(A,D) = \sup_{t \geq 0} b(A,D,t). \tag{6}$$

Semantics

- $b(A,D)$: amount of memory (buffers) required to store data
- $d(A,D)$: delay, assuming FIFO service


data
time
A
D
t
d(A, D)
b(A, D)
d(A, D, t)
b(A, D, t)

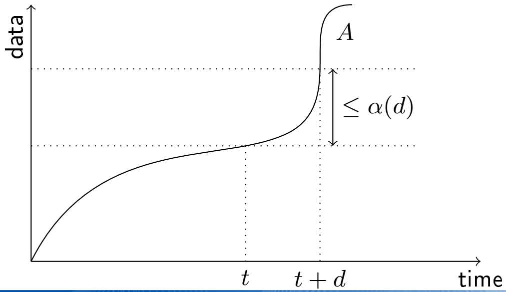
Performance contract of flow

Definition : Arrival curve

A cumulative curve $A \in \mathcal{C}$ admits $\alpha \in \mathcal{F}$ as arrival curve if

$$\forall t, d \in \mathbb{R}^+ : A(t+d) - A(t) \leq \alpha(d) \tag{7}$$

$$\mathcal{C}(\alpha) = \left\{ A \in \mathcal{C} \,\middle|\, \forall t, d \in \mathbb{R}^+ : A(t+d) - A(t) \leq \alpha(d) \right\} \tag{8}$$



ETR2021 :
arrival curve ≈ dbf

A few properties

Theorem : Arrival curve properties

Let $A \in \mathcal{C}, \alpha, \alpha' \in \mathcal{F}$.

- if α and α' are arrival curves for A, then also is $\alpha \wedge \alpha'$

$$\mathcal{C}(\alpha) \cap \mathcal{C}(\alpha) = \mathcal{C}(\alpha \wedge \alpha') \quad (9)$$

- one can always assume that $\alpha(0) = 0$
- if α is an arrival curve for A, and $\alpha' \geq \alpha$, then α' is an arrival curve for A

$$\alpha \leq \alpha' \implies \mathcal{C}(\alpha) \subseteq \mathcal{C}(\alpha') \quad (10)$$

- proofs are obvious
- properties are of practical importance (cf. class)

Performance contract of server

Definition : Backlogged period

Let S a server, $(A, D) \in S$. An interval $I \subset \mathbb{R}^+$ is a backlogged period if

$$\forall t \in I : A(t) > D(t) \tag{11}$$

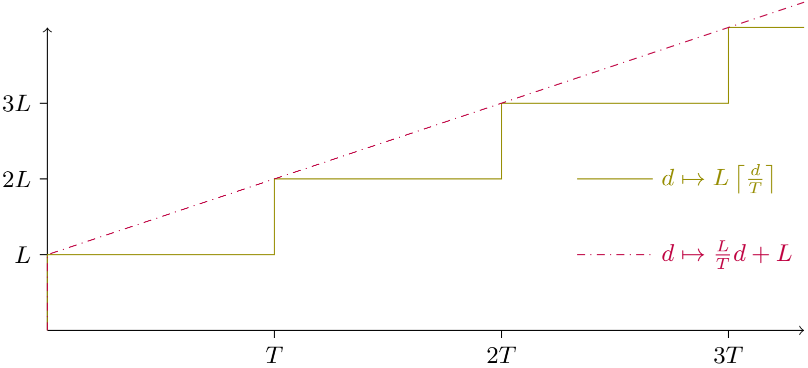
Definition : minimal strict service

A serveur S offers a strict service of function $\beta \in \mathcal{C}$ if

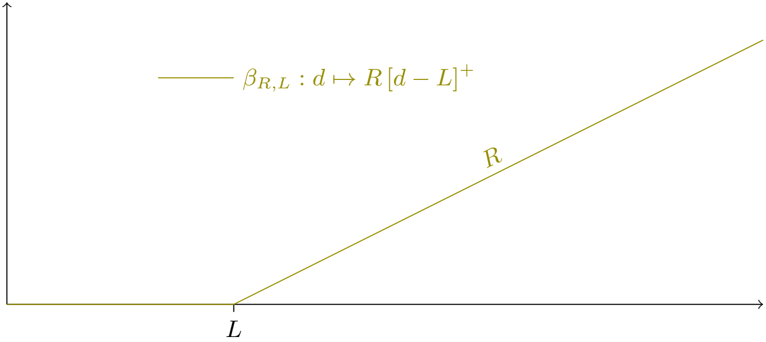
$$\forall t, d \in \mathbb{R}^+ \text{ t.q. } [t, t+d[\text{ backlogged period } D(t+d) - D(t) \geq \beta(d). \tag{12}$$

This is denoted $S \in \mathcal{S}_{\mathsf{strict}}(\beta)$.

What are these curves about ?



What are these curves about ?



What are these curves about ?

From contract to performances bounds

Theorem : Performance bounds (first version)

Let S a server, $\alpha, \beta \in \mathcal{F}$, $(A, D) \in S$. If $A \in \mathcal{A}(\alpha)$, $S \in \mathcal{S}_{\text{strict}}(\beta)$, then

$$d(A, D) \leq d(\alpha, \beta), \tag{13}$$
$$b(A, D) \leq b(\alpha, \beta). \tag{14}$$

. Moreover, D admits as arrival curve $\alpha' : d \mapsto \alpha(d + d(\alpha, \beta))$.

Outline

Dioid

Definition : Dioid

A dioide is a tuple $\langle \mathcal{D}, \oplus, \otimes, \tilde{0}, \tilde{1} \rangle$ with $\forall a, b, c \in \mathcal{D}$

- $\mathcal{D}$ is a set,
- $\oplus$ and $\otimes$ are two internal operators,
- operator $\oplus$ is
  - associative : $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$
  - commutative : $a \oplus b = b \oplus a$
  - *idempotent* : $a \oplus a = a$
  - with neutral element $\tilde{0}$: $a \oplus \tilde{0} = a$
- operator $\otimes$ is
  - associative : $(a \otimes b) \otimes c = a \otimes (b \otimes c) = a \otimes b \otimes c$
  - distributive over $\oplus$: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$
  - neutral element $\tilde{1}$: $a \otimes \tilde{1} = \tilde{1} \otimes a = a$,
  - absorbing element $\tilde{0}$: $a \otimes \tilde{0} = \tilde{0} \otimes a = \tilde{0}$

Why dioids ?

- a lot of interesting properties
- will simplify proofs

More details and properties

- commutative, complete dioid
- residuation
- ...
- out of scope of this presentation

The min-plus dioid

The min-plus dioid

Let $\wedge$ as $a \wedge b = \min(a, b)$, then $\langle \mathbb{R} \cup \{-\infty\}, \wedge, +, \infty, 0 \rangle$ is a dioid.

But this is not the one of interest.

The min-based convolution

Definition : Convolution

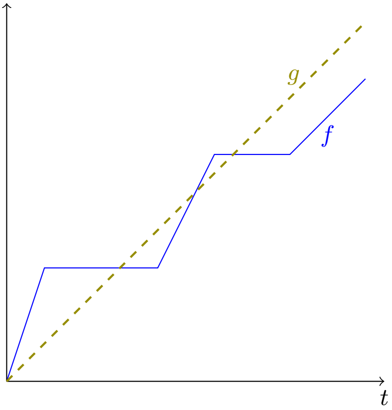
Let $f, g \in \mathcal{F}$, and define $f * g \in \mathcal{F}$ as $\forall t \in \mathbb{R}^+$

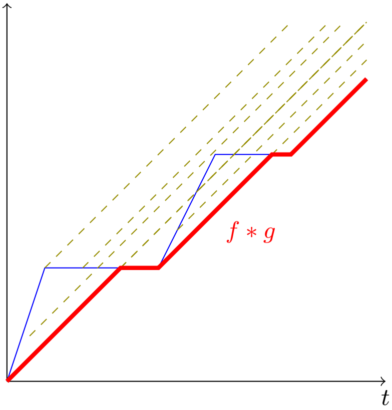
$$\begin{aligned}(f * g)(t) &= \inf_{0 \leq s \leq t} (f(t-s) + g(s)) \\ &= \inf_{u,s \geq 0,\ u+s=t} (f(u) + g(s))\end{aligned}$$

Properties $\forall f, g, h \in \mathcal{F}$

- *commutative* : $f * g = g * f$
- *associative* : $(f * g) * h = f * (g * h)$
- *distributivity over* $\wedge$: $f * (g \wedge h) = (f * g) \wedge (f * h)$
- if $f(0) = g(0) = 0$, then $f * g \leq f \wedge g$

Convolution illustration





The min-based dioid of functions

Definition : Min-based dioid of functions

$(\mathcal{F}, \wedge, *, \infty, \delta_0)$ is a (commutative complete) dioid

- $\delta_0 : 0 \mapsto 0 \ ; \ t > 0 \mapsto +\infty$

Deconvolution

Definition : Deconvolution

Let $f, g \in \mathcal{F}$ and define $f \oslash g$ as

$$\forall t \in \mathbb{R}^+ \quad (f \oslash g)(t) = \sup_{u \geq 0}\{f(t+u) - g(u)\}$$

Theorem : Link between deconvolution and residuation

$$f \oslash g = \inf \{h \in \mathcal{F} \mid h * g \geq f\} \tag{15}$$

A lot of properties

$\forall f, g, h \in \mathcal{F}$

- $h * g \geq f \Leftrightarrow h \geq f \oslash g$
- $f \geq g \Rightarrow f \oslash h \geq g \oslash h$
- $f \geq g \Rightarrow h \oslash f \leq h \oslash g$
- $(f \wedge g) \oslash h \leq (f \oslash h) \wedge (g \oslash h)$.
- $f \oslash (g * h) = (f \oslash h) \oslash g$
- $(f \oslash h) \oslash g = (f \oslash g) \oslash h$
- $(f * g) \oslash h \leq f * (g \oslash h)$

Outline

Min-based definitions

Definition : Arrival curve

A cumulative curve $A \in \mathcal{C}$ admits $\alpha \in \mathcal{F}$ as arrival curve if

$$A \leq A * \alpha \tag{16}$$

$$\mathcal{A}(\alpha) = \{A \in \mathcal{C} \mid A \leq A * \alpha\} \tag{17}$$

Definition : minimal min-plus service

A serveur S offers a minimal min-plus service of function $\beta \in \mathcal{C}$ if

$$D \geq A * \beta \tag{18}$$

This is denoted $S \in \mathcal{S}_{\text{mp}}(\beta)$.

The min-based definition generalizes the strict one.

$$\mathcal{S}_{\text{strict}}(\beta) \subset \mathcal{S}_{\text{mp}}(\beta) \tag{19}$$

Interest of min-based definitions I

Theorem : Arrival curve, min-based properties

Let $A \in \mathcal{C}, \alpha, \alpha' \in \mathcal{F}$.
If α and α' are arrival curves for A, then also is $\alpha * \alpha'$.

$$\mathcal{C}(\alpha) \cap \mathcal{C}(\alpha) = \mathcal{C}(\alpha * \alpha') \quad (20)$$

Exercise :

- $\alpha(d) = 2\left\lceil \frac{d}{4} \right\rceil$: arrival curve for periodic flow sending two frames of size 1 every 4 time unit
- $\alpha'(d) = \left\lceil \frac{d}{1} \right\rceil$: maximal link capacity, one frame per time unit
- compare $\alpha \wedge \alpha'$ and $\alpha * \alpha'$

Interest of min-based definitions II

Theorem : Performance bounds (second version)

Let S a server, $\alpha, \beta \in \mathcal{F}$, $(A, D) \in S$. If $A \in \mathcal{A}(\alpha)$, $S \in \mathcal{S}_{\text{mp}}(\beta)$, then

$$d(A, D) \leq d(\alpha, \beta), \qquad b(A, D) \leq b(\alpha, \beta). \tag{21}$$

. Moreover, D admits as arrival curve

$$\alpha' = \alpha \oslash \beta \tag{22}$$

Exercise :

- α : arrival curve for periodic flow
- β : constant rate
- compare $\alpha' : t \mapsto \alpha(t + d(\alpha, \beta)$ and $\alpha \oslash \beta$

Interest of min-based definitions III

Theorem : Pay burst only once

Let S, S' two servers. Then, the sequence $S \circ S'$ is also a server

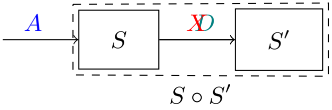
$$S \circ S' = \left\{ (A, D) \in \mathcal{C}^2 \,\middle|\, \exists X, (A, X) \in S, (X, D) \in S' \right\} \qquad (23)$$

Moreover, if S, S' offers respectively a min-plus service of curve β, β', then $S; S'$ offers a min-plus service of curve $\beta * \beta'$.

$$S \in \mathcal{S}_{\mathsf{mp}}(\beta), S' \in \mathcal{S}_{\mathsf{mp}}(\beta') \implies S; S' \in \mathcal{S}_{\mathsf{mp}}(\beta * \beta') \qquad (24)$$

Exercise :

- α : arrival curve for periodic flow
- β_1, β_2 : constant rate
- compare $hDev(\alpha, \beta_1 * \beta_2)$ and $hDev(\alpha, \beta_1) + hDev(\alpha \oslash \beta_1, \beta_2)$



Outline

Aggregate flow

Definition : Aggregate flow

Let $A_1, A_2 \in \mathcal{C}$, then $A_1 + A_2 \in \mathcal{C}$ is called cumulative curve of the aggregate flow.

Simple but useful...

MIMO server I

Definition : MIMO Server

A Multiple-Input Multiple-Output of dimension n server (aka n-MIMO) is $S \subseteq \mathcal{C}^n \times \mathcal{C}^n$ such that

$$(A_1, \ldots, A_n) \xrightarrow{S} (D_1, \ldots, D_n) \implies \forall i : A_i \geq D_i. \tag{25}$$

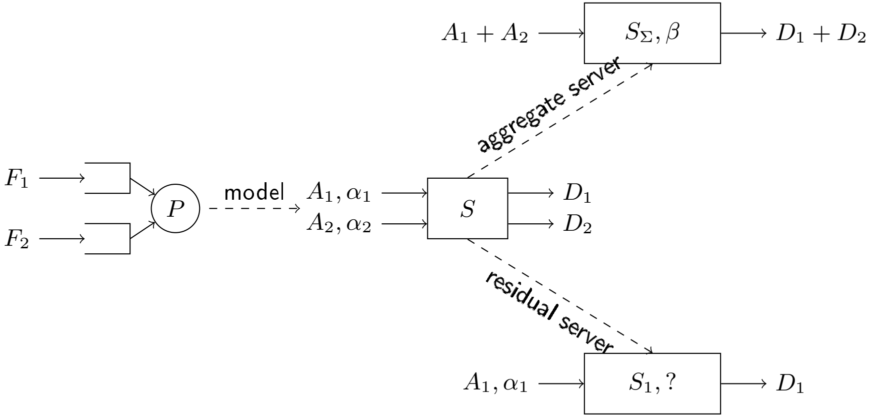
The associated aggregate server S_Σ is defined

$$\sum_{i=1}^{n} A_i \xrightarrow{S_\Sigma} \sum_{i=1}^{n} D_i \tag{26}$$

and for any $i \in [1, n]$, the residual/leftover server S_i is

$$A_i \xrightarrow{S_i} D_i. \tag{27}$$

MIMO server II



Computing a residual/leftover service curve

Individual service depends on

- the aggregate capacity
- the scheduling policy
- the competing arrival curves
- specific parameters (frame sizes, preemption cost, etc.)

Outline

Recall on TSN scheduling

- static priority (SP)
- bandwidth-sharing (ETS)
- credit-based shaper (CBS)
- asynchronous traffic shaping (ATS)
- time-triggered gates (TAS)

Static priority residual service

Theorem : Static priority residual service

Let S be a n-MIMO server, with aggregate strict service curve β ($S_\Sigma \in \mathcal{S}_{\text{strict}}(\beta)$) using non-preemptive static priority scheduling. If each input flow A_i admits arrival curve α_i and maximal frame size $L_i^{\max}$, then each residual server S_i offers the min-plus minimal service

$$\beta_i = \left[\beta - \sum_{j=1}^{j<i} \alpha_j - \max_{j>i} L_j^{\max}\right]_{\uparrow}^{+}. \tag{28}$$

ONERA THE FRENCH AEROSPACE LAB

Bandwith-sharing (ETS)

Theorem : DRR residual service

Let S be a n-MIMO server offering an aggregate strict service curve β with DRR service policy. If each flow i has a maximum packet size $L_i^{\max}$ and a quantum Q_i, then flow i is guaranteed the strict service curve β_i^{DRR} defined by

$$\beta_i^{\mathit{DRR}}(t) = \left[\frac{Q_i}{F} \beta(t) - \frac{Q_i(L - L_i^{\max}) + (F - Q_i)(Q_i + L_i^{\max})}{F} \right]^+$$

with $F = \sum_{i=1}^{n} Q_i$, $L = \sum_{i=1}^{n} L_i^{\max}$.

Other schedulers overview, [5]

TABLE I. A short overview of existing NC work. All mechanisms are listed.

Source	Mechanism	Author	Year	Pre-emption	Work Basis	Impact of CDT	Arrival α	Min. Service β	Max. Service β^{max}	Shaper σ	Max. Output α^*	Delay	Backlog
[31]	CBS	R. Queck	July 2012	-	[48]	no	leaky-bucket	min. 2 CBS & x SP			CBS & SP	CBS & SP	
[27]	CBS	J. A. Ruiz De Azua *et al.*	Oct. 2014	-	[31]	no	detailed	min. & strict for 2 CBS & strict for SP	2 CBS	2 CBS			
[49]	CBS	Lin Zhao *et al.*	Nov. 2018	-	[27]	no	detailed	min. 3 CBS					
[41]	TAS	Luxi Zhao *et al.*	July 2018	no	[29] [42]	-	leaky-bucket	min. TT			TT	TT	
[50]	TAS-CBS	F. He *et al.*	May 2017	no	[27]	yes	leaky-bucket			2 CBS	CBS incl. shaper		
[29]	TAS-CBS	Luxi Zhao *et al.*	April 2018	yes&no	[27]	yes	leaky-bucket	min. 2 CBS			CBS	CBS	
[26]	TAS-CBS	H. Daigmorte *et al.*	June 2018	yes&no	[29] [27]	yes	detailed	min. & strict for x CBS		x CBS			
[28]	TAS-CBS	Luxi Zhao *et al.*	Dec. 2018	no	[26] [29] [27]	yes	leaky-bucket	min. x CBS		x CBS & link	CBS incl. shaper	CBS	
[37] & [35]	ATS-CBS	E. Mohammad-pour *et al.*	Sep. 2018	-	[27]	yes	$=\alpha^*$	min. 2 CBS / min. x CBS [32]			CBS incl. link	CBS	CBS

What about FIFO ?

What about FIFO ?

- easy to implement
- easy to analyze per node
- hard to analyze end-to-end

Outline

Conclusion

- TSN is a revolution

Conclusion

- TSN is a revolution
- but not all revolutions keep all promises

Conclusion

- TSN is a revolution
- but not all revolutions keep all promises
- Networks calculus is a mature analysis method

Conclusion

- TSN is a revolution
- but not all revolutions keep all promises
- Networks calculus is a mature analysis method
- adapted to TSN

Conclusion

- TSN is a revolution
- but not all revolutions keep all promises
- Networks calculus is a mature analysis method
- adapted to TSN
- presenting both in 90' is challenging

Bibliography I

[1] Virtual Bridged Local Area Networks Amendment 14 : Stream Reservation Protocol (SRP).
Technical Report IEEE 802.1Qat, IEEE, 2010.

[2] Anne Bouillard, Marc Boyer, and Euriell Le Corronc.
Deterministic Network Calculus – From theory to practical implementation.
Number ISBN : 978-1-119-56341-9. Wiley, 2018.

[3] János Farkos.
Standards from ieee, ietf, 3gpp, and beyond –.
Proc. of the TSN/A Conference 2019.

[4] Jean-Yves Le Boudec and Patrick Thiran.
Network Calculus, volume 2050 of *LNCS*.
Springer Verlag, 2001.

Bibliography II

[5] Lisa Maile, Kai-Steffen Hielscher, and Reinhard German.
Network calculus results for tsn : An introduction.
In *Proc. of the Information Communication Technologies Conference (ICTC 2020)*, pages 131–140. IEEE, 2020.

[6] Jonas Rox and Rolf Ernst.
Compositional performance analysis with improved analysis techniques for obtaining viable end-to-end latencies in distributed embedded systems.
International Journal on Software Tools for Technology Transfer, 15(3) :171–187, 2013.