Network calculus: from theory to avionic applications

INRIA/Spades seminar

Marc Boyer

ONERA – The French Aerospace Lab

Tuesday, May 12th, 2015

THE FRENCH AEROSPACE LAB

r e t o u r s u r i n n o v a t i o n

Outline

Outline

- Network calculus: theory
  - NC: what and why?
  - System modelling in network calculus
  - The $(\min, +)$ dioid(s)
  - From reality to contracts
  - From contracts to bounds
  - Aggregated and residual services
- Links with other theory
  - Real-Time calculus
  - Event stream
  - Task scheduling
  - Comparison
- Tools
  - NC/RTC tools
- Accuracy in avionic applications
- Conclusion

ONERA THE FRENCH AEROSPACE LAB

Outline

What is network calculus?

- a theory to compute *memory* and *delay* bounds in networks
- based on the $(\min, +)$ dioid
- used to certify A380 AFDX backbone

What is network calculus?

- a theory to compute *memory* and *delay* bounds in networks
- based on the $(\min, +)$ dioid
- used to certify A380 AFDX backbone

Why use it?

Because it is elegant

What is network calculus?

- a theory to compute *memory* and *delay* bounds in networks
- based on the $(\min, +)$ dioid
- used to certify A380 AFDX backbone

When to use it

- multi-hop real-time communications
- no simple analysis exists
- no (or few) cyclic dependencies
- no (or simple) feedback flow control

What is network calculus?

- a theory to compute *memory* and *delay* bounds in networks
- based on the $(\min, +)$ dioid
- used to certify A380 AFDX backbone

When to use it

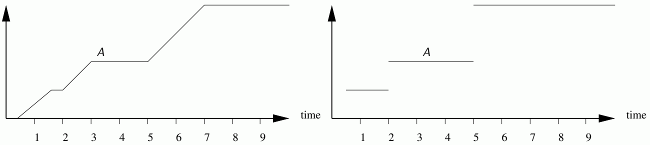
- multi-hop real-time communications
- no simple analysis exists
- no (or few) cyclic dependencies
- no (or simple) feedback flow control

Network calculus benefits

- Different accuracies
- Scalable approach

Outline

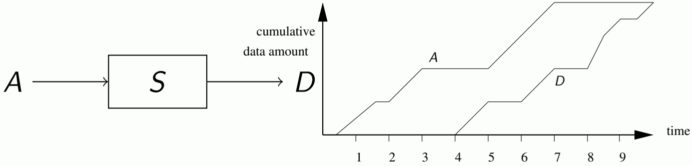
Basic object: cumulative function



- Flow: Cumulative function A
  - $A(t)$: amount of data sent up to time t
  - Properties:
    - null at 0 (and before)
    - non decreasing
  - Discrete of fluid modeling
  - Better definition than instantaneous throughput $\rho(t)$

$$A(t) = \int_0^t \rho(x)\mathrm{d}x \tag{1}$$

Basic object: server



- Simple input/output relation:

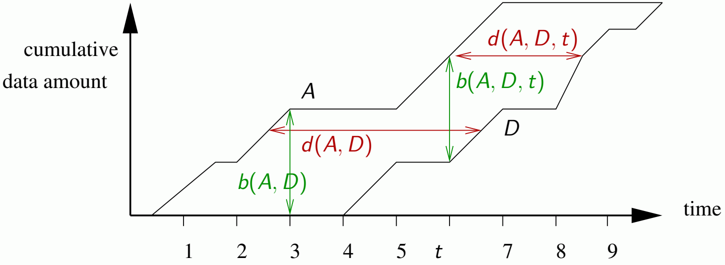
$$S \subset \mathcal{F}_0 \times \mathcal{F}_0$$

- Departure "after" arrival

$$A \xrightarrow{S} D \implies A \geq D$$

- Basic server model:
  - no loss of messages
  - infinite memory
  - no add (header, checksum, etc.)

Performance criteria: delay and backlog



$$b(A, D, t) \stackrel{\text{def}}{=} D(t) - A(t) \qquad d(A, D, t) \stackrel{\text{def}}{=} \inf\{\tau \geq 0 : A(t) \leq D(t + \tau)\}$$

$$b(A, D) \stackrel{\text{def}}{=} \max_{t \geq 0} \{b(A, D, t)\} \qquad d(A, D) \stackrel{\text{def}}{=} \max_{t \geq 0} \{d(A, D, t)\}$$

Outline

The $(\min, +, \mathbb{R})$ dioid

- The dioid on values: $(\wedge, +, \mathbb{R})$: $a \wedge b = \min(a, b)$
  - associativity, commutativity, distributivity: $a + (b \wedge c) = (a + b) \wedge (a + c)$
  - "looks like" $(+, \times, \mathbb{R})$
- The dioid on functions:

min-plus convolution $$(f * g)(t) \overset{\text{def}}{=} \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\}$$

min-plus deconvolution $$(f \oslash g)(t) \overset{\text{def}}{=} \sup_{s \geq 0} \{f(t+s) - g(s)\}$$

min-plus Kleene closure $$f^* \overset{\text{def}}{=} \delta_0 \wedge f \wedge (f * f) \wedge (f * f * f) \wedge \ldots$$

Outline

Why contracts?

- cumulative functions are *real* behaviours

Why contracts?

- cumulative functions are *real* behaviours
  - unknown at design time

Why contracts?

- cumulative functions are *real* behaviours
  - unknown at design time
  - or too complex to handle

Why contracts?

- cumulative functions are *real* behaviours
  - unknown at design time
  - or too complex to handle

$\Rightarrow$ need to handle *contracts*

Why contracts?

- cumulative functions are *real* behaviours
  - unknown at design time
  - or too complex to handle

$\Rightarrow$ need to handle *contracts*

  - arrival curves: contracts on input traffic
  - service curves: contracts on service

Arrival curve definition

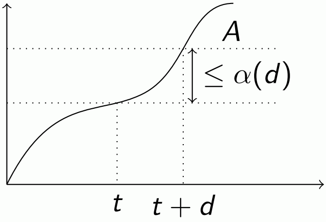
A flow A has α as (maximal) arrival curve iff

$$\forall t, \Delta \in \mathbb{R}_{\geq 0} : A(t+\Delta) - A(t) \leq \alpha(\Delta)$$

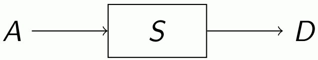
$$\iff$$

$$A \leq A * \alpha$$

- $\alpha(\Delta)$ upper bounds the amount of data send on any interval of width Δ
- minimal arrival curve also exist



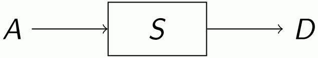
How to define a service? Engineer point of view.



- Constant service: R bits per second
  - First idea: $D(t+\Delta) - D(t) \geq R\Delta$
  - Only when there is some backlog ($\forall x \in [t, t+\Delta] : D(t) < A(t)$)
- Generalisation to non constant: any β function
  - $D(t+\Delta) - D(t) \geq \beta(\Delta)$
  - on backlogged periods

  $\Rightarrow$ minimal *strict* service

How to define a service? Mathematician point of view.



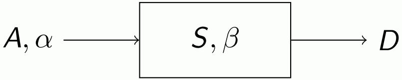
- Use $(\wedge, +)$ convolution (symmetry with arrival curve)
- Must be linked with arrival curve (more arrival, more departure, up to service capacity)

$$D \geq A * \beta$$

$\Rightarrow$ minimal *simple* service (or minimal *min-plus* service)

Outline

Basic results



Assume a server S with minimal service β and arrival flow A with arrival curve α, and departure D

- Bound on delay:

$$d(A, D) \leq d(\alpha, \beta) \tag{2}$$

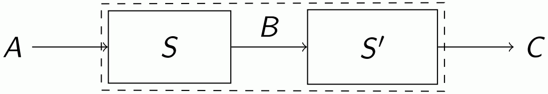
- Bound on memory usage

$$b(A, D) \leq b(\alpha, \beta) \tag{3}$$

- Arrival curve of departure D

$$(\alpha \oslash \beta) \tag{4}$$

Pay burst only once principle



Pay burst only once

The sequence S, S' can be replaced by a virtual server $S; S'$ with service curve $\beta * \beta'$.

Interest End-to-end delay is less than sum of individual delays.

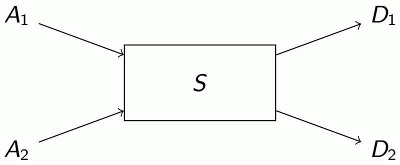
$$h(\alpha, \beta * \beta') \leq h(\alpha, \beta) + h(\alpha, \beta') \tag{5}$$

Proof: $R'' \geq R' * \beta \geq (R * \beta) * \beta' = R * (\beta * \beta')$

Rq: In [20], more than 7 pages are required to prove limited version of this result.

Outline

Shared service



$$A_1 \geq D_1$$
$$A_2 \geq D_2$$

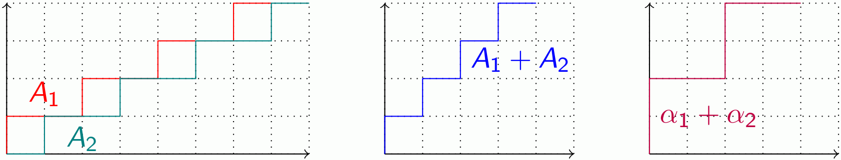
Service repartition still depends on

- server policy (FIFO, Static Priority...)
- individual flow contracts

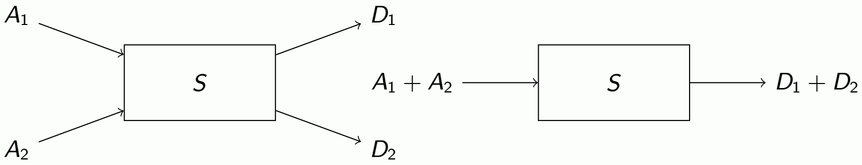
Point of view on flow aggregation

Consider two flows A_1, A_2 of arrival curves α_1, α_2.

- $A_1 + A_2$ is an arrival curve, *i.e.* a non-decreasing function $\mathbb{R}_{\geq 0} \to \{R\}$
- $(A_1 + A_2)(t)$ is the amount of data set by both flows up to time t
- $\alpha_1 + \alpha_2$ is an arrival curve for $A_1 + A_2$
- not the best one
  - $A_1 = \nu_{2,1}$, $\alpha_1 = A_1$, $A_2(t) = A_1(t-1)$, $\alpha_2 = \alpha_1$



Aggregated service

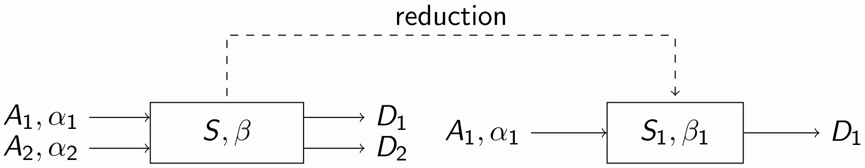


$$A_1 \geq D_1$$
$$A_2 \geq D_2$$
$$(D_1 + D_2) \geq (A_1 + A_2) * \beta$$

Service repartition depends on

- server policy (FIFO, Static Priority...)
- individual flow contracts

Challenge: residual service



$$A_1 \geq D_1 \qquad D_1 \geq A_1 * \beta_1$$
$$A_2 \geq D_2 \qquad D_2 \geq A_2 * \beta_2$$

How to compute β_1, β_2*?*

Per policy residual service

Residual service must be defined for every scheduling policy

- Static priority: $\beta^M = [\beta - \alpha_H - l_L^{\max}]_{\uparrow}^{+}$
- FIFO: $\forall \theta \in \mathbb{R}_{\geq 0} : \beta_i = [\beta - \alpha_j * \delta_\theta]_{\uparrow}^{+} \wedge \delta_\theta$
- GPS [17, 1]
- WFQ [11, 16]
- DRR [21, 6]
- AVB [18, 19]
- TDMA [10]
- EDF [15]

The result may depend on the kind of service (simple, strict)...
The function $[f]_{\uparrow}^{+}$ is the non-negative, non-decreasing closure of f.

Combining scheduling policies

Hierarchical scheduling

- A residual service is still a service
- Can be used to combine scheduling policies
  - SP/FIFO (AFDX)
  - SP/DRR/FIFO
  - DRR/EDF
  - ...
- Some restriction on the kind of service may exist

Heterogeneous network

- different scheduling policies may be used in a network/system
- from NC point of view, they all are service

Outline

Outline

Real-Time calculus

The same model, with another name [2]

- different presentation
- more focus on minimal arrival curves
- time domain: $\mathbb{R}$ vs. $\mathbb{R}_{\geq 0}$

Outline

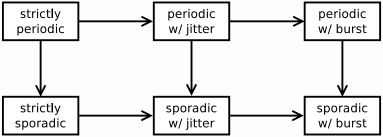
Event stream: the trade-of between NC/RTC and scheduling

Philosophy [12]

"Furthermore, the new models [ie NC/RTC] are far less intuitive than the ones known from the classical real-time systems research, e. g. the model of rate-monotonic scheduling with its periodic tasks and worst-case execution times. A system-level analysis should be simple and comprehensible, otherwise its acceptance is extremely doubtful."
"We don't necessarily need to develop new local analysis techniques if we can benefit from the host of work in real-time scheduling analysis."

Event stream model

- count events, not amount of data
  - is workload related to frame sizes?
- event stream $\langle \eta^+, \eta^-, \delta^+, \delta^- \rangle$
  - η^+, η^- maximal/minimal amount of events per interval
  - δ^+/δ^- maximal/minimal distance between events
- event models: sub-classes of event streams



- no formal model of stream transformation
  - re-use of scheduling results
  - assume periodic-based model
  - propagation: jitter propagation + ad-hoc enhancements

Outline

Task scheduling

A large research area:

- Main model: periodic task (+offsets, + dependencies, +...)
- Main problem: local schedulability

Differences with network analyses:

- propagation: output of a system is input of another
- shaping: maximal throughput $\implies$ maximal input

Outline

Comparison

Three basic objects

- system behaviour
- bounds on behaviours
- "computer friendly" sub-classes

Not all do clear distinction between these objects.

Outline

Outline

NC/RTC tools

Two main components:

- a library to handle curves (sum, minimum, convolution, etc.)
- a network analyser

- DISCO: NC analyser, in Java, Licence LGPL
  - http://disco.informatik.uni-kl.de/index.php/projects/disco-dnc
  - curves library and network analyser
- Real-Time Calculus (RTC) Toolbox
  - http://www.mpa.ethz.ch/Rtctoolbox
  - Curve Library: Java implementation (no source code) + matlab interface
  - Network analyser: Modular Performance Analysis (MPA), Matlab code
- RTaW-PEGASE
  - http://www.realtimeatwork.com/software/rtaw-pegase
  - commercial Java tool

Outline

Accuracy is "good enough"

Experiment on realistic configuration [5]:

- 8 switches
- 104 end-systems
- 6500 VL

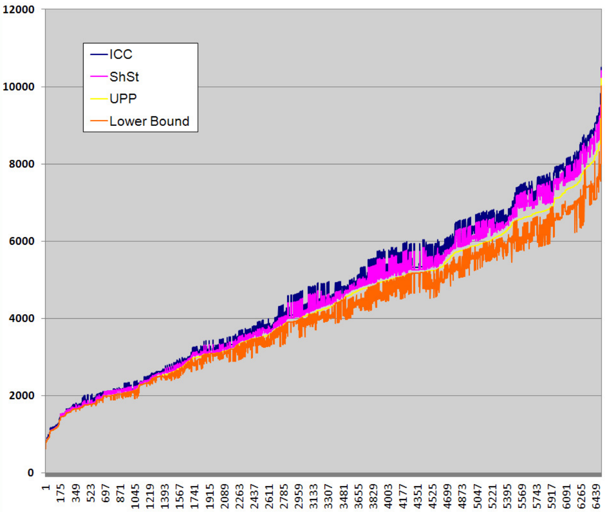
Computation time:

- Fast algorithm (ICC): 1s
- Accurate algorithm (UPP): 10s

Method accuracy:

	All Virtual Links		20% of VLs with highest WCTT	
	ICC	UPP	ICC	UPP
Min	3.74%	0%	15.2%	3.55%
Av.	31.02%	**16.44%**	42.08%	25.37%
Max.	82.4%	76.06%	81.53%	76.08%

Plotting accuracy



Outline

Conclusion

Lot of work done from [8, 9, 13], two reference books [14, 7]...

Toward smaller upper bound for avionic systems

- either more information on system
- or more complex analyse
- but exact delay is NP-Hard [3, 4]

Next research topics:

- formal correction proof (cf Stephan Merz, Oct. 2014)
- other application domains
    - network on chip
    - from network to system

Personal feeling

Main contributions of NC are

- clear distinction between
  - real behaviour
  - bounds on behaviours (arrival/service curves)
  - computation-friendly sub-classes
- formal definition of delay
- new point of view on real-time
- hierarchical scheduling

Drawbacks (improvement areas):

- too many definitions of service
- infimum based proofs, continuity problems

Bibliography I

P. Barta, F. Nemeth, R. Szabo, and J. Biro.
End-to-end delay bound calculation in generalized processor sharing networks.
In *Proc. of the sixth IEEE Symposium on Computers and Communications (ISCC 2001)*, pages 282 –287, 2001.

Anne Bouillard, Laurent Jouhet, and Eric Thierry.
Service curves in Network Calculus: dos and don'ts.
Research Report RR-7094, INRIA, 2009.

Anne Bouillard, Laurent Jouhet, and Eric Thierry.
Tight performance bounds in the worst-case analysis of feed-forward networks.
Technical Report 7012, INRIA, 2009.

Anne Bouillard, Laurent Jouhet, and Eric Thierry.
Tight performance bounds in the worst-case analysis of feed-forward networks.
In *Proceedings of the 29th Conference on Computer Communications (INFOCOM 2010)*, pages 1–9, march 2010.

Marc Boyer, Nicolas Navet, and Marc Fumey.
Experimental assessment of timing verification techniques for afdx.
In *Proc. of the 6th Int. Congress on Embedded Real Time Software and Systems*, Toulouse, France, February 2012.

Bibliography II

Marc Boyer, Giovanni Stea, and William Mangoua Sofack.
Deficit round robin with network calculus.
In *Proc. of the 6th International Conference on Performance Evaluation Methodologies and Tools (ValueTools 2012)*, Cargese, France, October, 9–12 2012.

Cheng-Shang Chang.
Performance Guarantees in communication networks.
Telecommunication Networks and Computer Systems. Springer, 2000.

Rene L. Cruz.
A calculus for network delay, part I: Network elements in isolation.
IEEE Transactions on information theory, 37(1):114–131, January 1991.

Rene L. Cruz.
A calculus for network delay, part II: Network analysis.
IEEE Transactions on information theory, 37(1):132–141, January 1991.

Bibliography III

Dinh-Khanh Dang and Ahlem Mifdaoui.
Timing analysis of tdma-based networks using network calculus and integer linear programming.
In *IEEE 22nd Int. Symp. on Modeling Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Paris, FR, 2014.

R. Guérin and V. Peris.
Quality of service in packets networks – basic mechanisms and directions.
In *Proc. of the 5th Int Work. on Network and Operating system support for Digital Audio and Video*, Durham NH, April 1995.

R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst.
System level performance analysis - the symta/s approach.
IEEE Proceedings on Computers and Digital Techniques, 152(2):148 – 166, march 2005.

Jean-yves Le Boudec.
Network calculus made easy.
Technical Report EPFL-DI 96/218, Ecole Polytechnique Fdrale de Lausanne (EPFL), December 1996.

Bibliography IV

Jean-Yves Le Boudec and Patrick Thiran.
Network Calculus, volume 2050 of *LNCS*.
Springer Verlag, 2001.
http://lrcwww.epfl.ch/PS_files/NetCal.htm.

J. Liebeherr, Y. Ghiassi-Farrokhfal, and A. Burchard.
On the impact of link scheduling on end-to-end delays in large networks.
IEEE Journal on Selected Areas in Communications, 29(5):1009–1020, May 2011.

William Mangoua Sofack and Marc Boyer.
Generalisation of gps and p-gps in network calculus.
In *Proc. of the Work in Progress session of 9th IEEE Workshop on Factory Communication Systems (WFCS'2012)*, Lemgo/Detmold, Germany, May 21–24 2012.

A. Parekh and R. Gallager.
A generalised processor sharing approach to flow control in integrated services networks: the single-node case.
IEEE transactions on networking, June 1993.

Bibliography V

R. Queck.
Analysis of ethernet AVB for automotive networks using network calculus.
In *Proc. of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 61–67, July 2012.

Joan Adrià Ruiz de Azua and Marc Boyer.
Complete modelling of AVB in network calculus framework.
In *Proc. of the 22nd Int. Conf. on Real-Time Networks and Systems (RTNS 2014)*, Versailles, France, October 8-10 2014.

Dimitrios Stiliadis and Anujan Varma.
Latency-rate servers: a general model for analysis of traffic scheduling algorithms.
Technical Report UCSC-CRL-95-38, University of California, Santa Cruz, Santa Cruz, CA, USA, July 1995.

Hua Yu and Xue Liu.
Scheduling design and analysis for end-to-end heterogeneous flows in an avionics network.
In *Proc. of the 30st IEEE Int. Conf. on Computer Communications (IEEE INFOCOM 2012)*, pages 2417 –2425, april 2011.