An efficient and simple class of functions to model arrival curve of packetised flows

Marc Boyer, Jörn Migge, Nicolas Navet

ONERA THE FRENCH AEROSPACE LAB

RTaW RealTime-at-Work

RTSS/WCTT Workshop
Nov. 29th, 2011

Outline

Outline

What is Network Calculus ?

- A theory designed to compute *guaranteed bounds* on delays.
- With a strong mathematical background: (min,+) algebra
  - Basic object: non-decreasing, non-negative functions

$$\mathcal{F} = \{ f : \mathbb{R}_+ \to \mathbb{R}_+ \mid x < y \implies f(x) \leq f(y) \}$$

  - Three basic operations: the convolution $*$, deconvolution $\oslash$, the sub-additive closure f^*.

$$(f * g)(t) = \inf_{0 \leq u \leq t} (f(t-u) + g(u)) \tag{1}$$

$$(f \oslash g)(t) = \sup_{0 \leq u} (f(t+u) - g(u)) \tag{2}$$

$$f^* = \delta_0 \wedge f \wedge (f * f) \wedge (f * f * f) \wedge \cdots \tag{3}$$

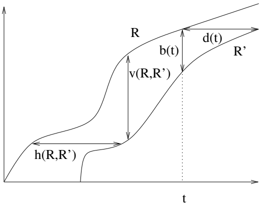
Network calculus overview

Two basic objects:

- *Flow:*
  - modelling: $R \in \mathcal{F} = \{\mathbb{R}^+ \to \mathbb{R}^+, \text{non-decreasing}\}$
  - semantics: $R(t)$, cumulative amount of data up to t
- *Server:*
  - modelling: $S \in \mathcal{F} \times \mathcal{F}$: $R \xrightarrow{S} R' \implies R' \leq R$
  - semantics: relation between some input and some output, no loss, output comes after input ($R'(t) \leq R(t)$)
- *delay:*

$$d(R, S) \quad \leq \quad \max_{R \xrightarrow{S} R'} h(R, R')$$

$h(R, R')$: horizontal deviation



Contract modelling

- Flow contract: arrival curve α

$$R \prec \alpha \iff \forall t, \Delta \in \mathbb{R}^{+} R(t+\Delta) - R(t) \leq \alpha(\Delta)$$
$$\iff R \leq R * \alpha$$

- Server contract: service curve
  - simple service of curve β
    $R \xrightarrow{S} R' \iff R' \geq R * \beta$
  - strict service of curve β
    for all backlogged period $[t, t+\Delta[$
    $(i.e. \forall x \in [t, t+\Delta[$: $R'(x) < R(x))$:
    $R'(t+\Delta) - R'(t) \geq \beta(\Delta)$
- Results: $R \xrightarrow{S} R'$, $R \prec \alpha$, S has service curve β:

$$R' \prec \alpha \oslash \beta$$
$$d(R, S) \leq h(\alpha, \beta)$$

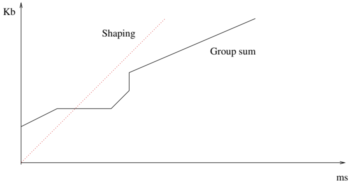
Outline

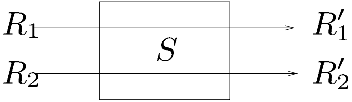
Shaping on links

A link is shared by a set of flows: what is the throughput of this set ?

- Principle: whatever the applicative throughput is, is it limited by the links capacity
  Also known has:
  - Serialisation: the frames of the different flows can not be sent at the same time
  - Grouping: computes per-group throughput, not per-flow
- Interest: considering long term rate ρ and instantaneous burst b
  - applicative flows: small ρ, big b
  - link: big ρ, null b
- Impact: up to 40% in industrial system

Shaping and network calculus





Let S be a server, with *shaping curve* σ, then, the output is constrained by σ.
If the output is constrained by α', it is by $\alpha' \wedge \sigma$.

$$R \xrightarrow{S} R' \implies R' \prec \sigma$$

$$R \prec \alpha, S \trianglerighteq \beta \implies R' \prec \sigma \wedge (\alpha \oslash \beta)$$

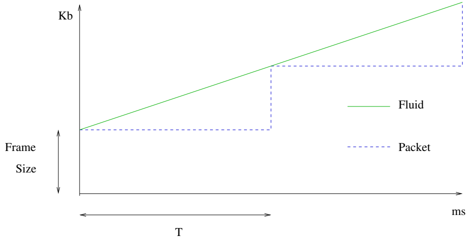
Modelling a packetized flow

Common example: sporadic flow

- inter emission "period": T
- frame size (fixed or max): b

Two modelling:

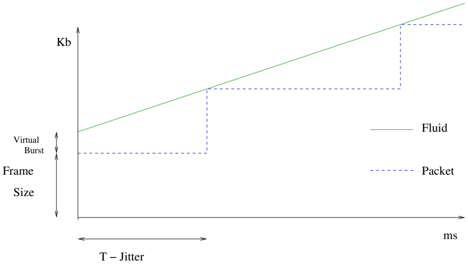
- fluid ("token bucket"): affine function, continuous
- packetized: stair-case functions, discontinuous



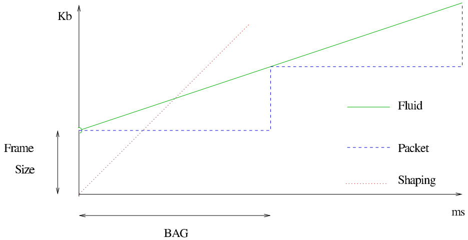
Fluid modelling: the virtual burst problem

Jitter "shifts" the arrival curve:

- if jitter $<$ period: instantaneous burst unchanged
- in fluid modelling: creation of virtual burst $\Longrightarrow$ increase bounds



Putting all together



- fluid + shaping: concave piecewise linear function (CPL)
  Efficient min, max, sum
  Implementation in floating points
- stair-case modelling: general class (UPP)
  Complex min, max, sum
  Implementation in exact rationals ($\mathbb{Q}$)

Outline

Getting the better of each class

Classes strengths/weaknesses:

- jitter effect: stair-case class
- summing ("grouping"): CPL class
- shaping: CPL class

Idea:

- keeping stair-case for individual flow constraint
- converting into CPL when summing and shaping

From stair-case to CPL

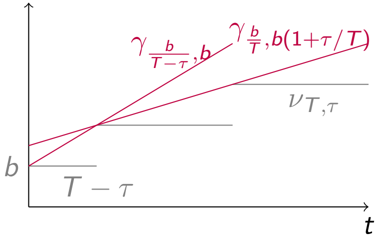


Figure: CPL overapproximation of a stair-case function

$$cpl(b\nu_{T,\tau}) = \gamma_{\frac{b}{nT-\tau},nb} \wedge \gamma_{\frac{b}{T},b(1+\tau/T)} \quad (4)$$

Algorithm adaptation

Function $GroupAC(S_j, \mathcal{F})$;
$\texttt{Assert}(\mathcal{F} \subset S_j{}^\bullet)$;
begin
  if $S_j = S_0$ **then**
    return $\sum_{F_i^k \in \mathcal{F}} \alpha_i^k$; // $\mathcal{F} \subset S_j{}^\bullet \implies k = 0$
  else if S_j *is an aggregate server* **then**
    if $d_j = null$ **then**
      $\alpha^{S_j} \leftarrow \sum_{S_p \in {}^{\bullet\bullet}S_j} \texttt{GroupAC}(S_p, S_p{}^\bullet \cap {}^\bullet S_j)$;
      $d_j = h(\alpha^{S_j}, \beta_j)$;
      foreach $F_i^k \in {}^\bullet S_j$ **do**
        $\alpha_i^{k+1} \leftarrow (\alpha_i^k \oslash \delta_{d_j})^*$;
    return $\left(\sigma_j \wedge \sum_{F_i^k \in \mathcal{F}} \alpha_i^k\right)^*$

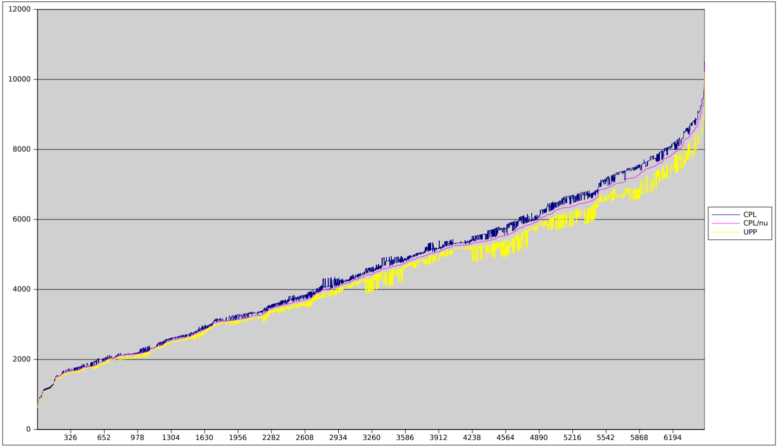
Adaptation: replace $\sum_{F_i^k \in \mathcal{F}} \alpha_i^k$ by $\sum_{F_i^k \in \mathcal{F}} cpl(\alpha_i^k)$

Outline

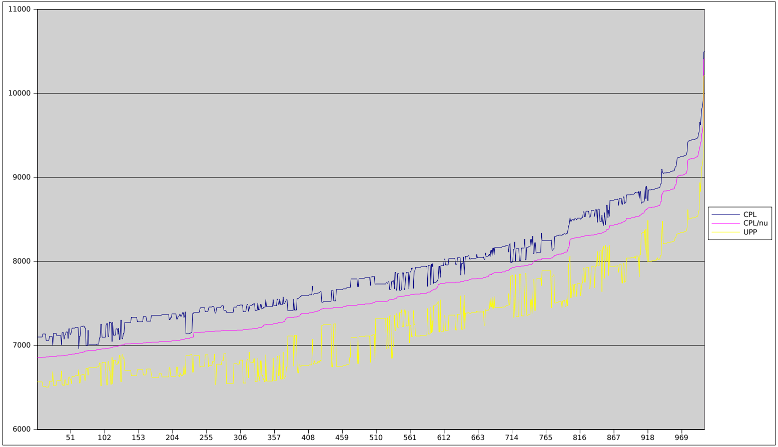
Testbed configuration

- industrial (Thales) configuration
- 104 nodes
- 8 switches
- 974 multicast flows
- 6501 end-to-end bounds

Comparing methods



Zoom on worst delays



Pessimism evaluation

- Method comparison: based on upper bound (UB_m)
- Best comparison: based on pessism

$$\text{pess}_m = UB_m - WCTT$$

- Worst case unknown ($WCTT$)
- Delay lower bound: trajectorial based approach (LB)

$$LB \leq WCTT \leq UB_m$$
$$\text{pess}_m \leq UB_m - LB$$

Uppers and lower bounds

An efficient and simple class

M. Boyer

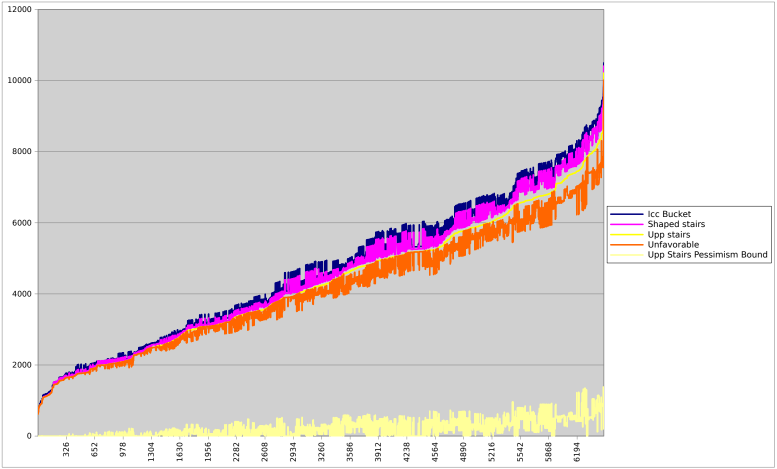
Network calculus

Shaping, packetization and computation time

Swaping between function classes

Experiment

Conclusion



Pessimism bounding, per method

An efficient and simple class

M. Boyer

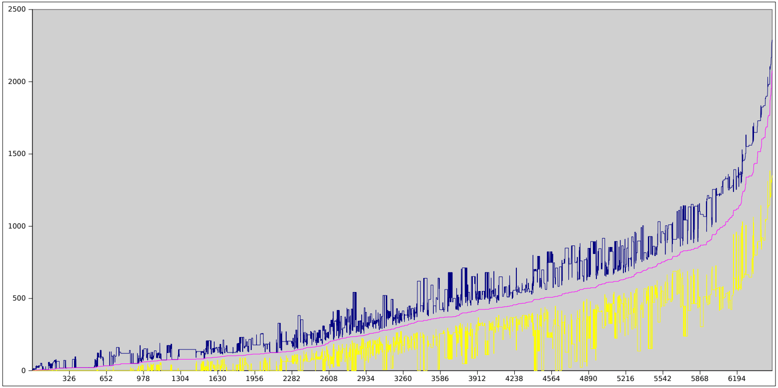
Network calculus

Shaping, packetization and computation time

Swaping between function classes

Experiment

Conclusion



Experiment values

Method	CPL (float)	CPL$/b.\nu_{T,\tau}$ (float)	UPP (rat)
Computation time	0.9 s	1.1 s	7.2 s
Min gain	-	0%	0.15%
Max gain	-	7.8%	15.2%
Av. gain	-	2.49%	5.92%
Min gain on 1000 biggest	-	0.8%	2.0%
Max gain on 1000 biggest	-	4.4%	11.9%
Av. gain on 1000 biggest	-	2.9%	8.3%

Gain correlation: 0.785

Outline

Conclusion

- Two critical aspects: shaping and packetization
- Two existing methods:
  - fluid: bad packetization, quick computation
  - stair-case: good packetization, longer computation
- Contribution: trade-off tightness/computation time
- Don't use CPL, use CPL/$b.\nu_{T,\tau}$
  - simple to implement
  - low computation time over-head
  - significant bound improvement
- Perspective: use in optimisation loop
  - quick computation in first iterations
  - longer computation to finalise