Half-modelling of shaping in FIFO net with network calculus

Marc Boyer

ONERA
THE FRENCH AEROSPACE LAB

RTNS 2010 – nov. 4th 2010

Outline

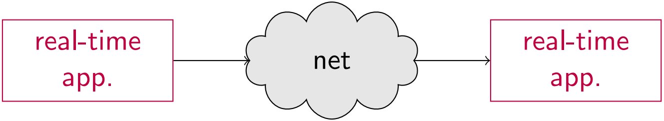
Outline

Worst Case Traversal Time: What and Why

Net in real-time systems

Embedded systems are:

- real-time ($\Longrightarrow$ real-time scheduling)
- communicating:

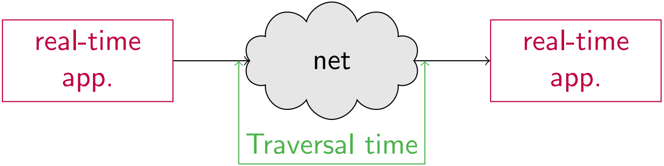


Worst Case Traversal Time: What and Why

Net in real-time systems

Embedded systems are:

- real-time ($\Longrightarrow$ real-time scheduling)
- communicating: network delay (traversal time)



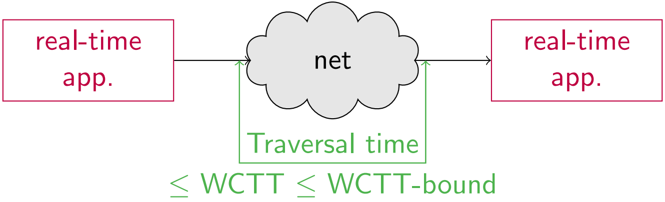
Worst Case Traversal Time: What and Why

Net in real-time systems

Embedded systems are:

- real-time ($\implies$ real-time scheduling)
- communicating: network delay (traversal time)

$\implies$ need of end-to-end delay bound (WCTT)



Worst Case Traversal Time: What and Why

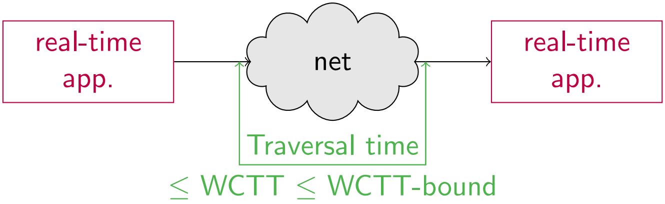
Net in real-time systems

Embedded systems are:

- real-time ($\Longrightarrow$ real-time scheduling)
- communicating: network delay (traversal time)

$\Longrightarrow$ need of end-to-end delay bound (WCTT)

$\Longrightarrow$ *traffic contract* and *service guarantee*



Outline

Basic ideas

- Theory designed to compute WCTT bounds
- Used to certify A380
- Strong mathematical background: $(\min, +)$ dioid

$$\mathcal{F} = \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; \begin{array}{lcl} x < y & \implies & f(x) \leq f(y) \\ x < 0 & \implies & f(x) = 0 \end{array} \right\}$$

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

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

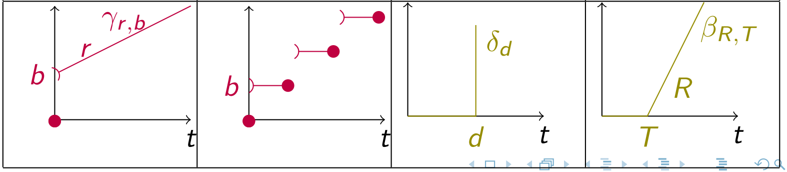
Reality modelling

- Data flow: $R(t)$ amount of data up to time t (cumulative curve)
- Server: transforms input into output $R \xrightarrow{S} R'$
- Arrival curve: α

$$\forall t, d \geq 0 : R(t+d) - R(t) \leq \alpha(d) \iff R \leq R * \alpha$$

- Service curve: β iff $R' \geq R * \beta$

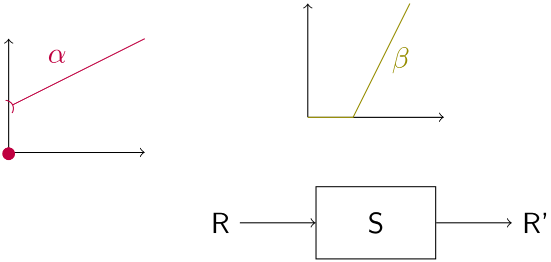
Traffic contract		Service guarantee	
Token bucket	Periodic	Delay	Rate-latency



First results

Given:

- an arrival traffic contract
- a service guarantee



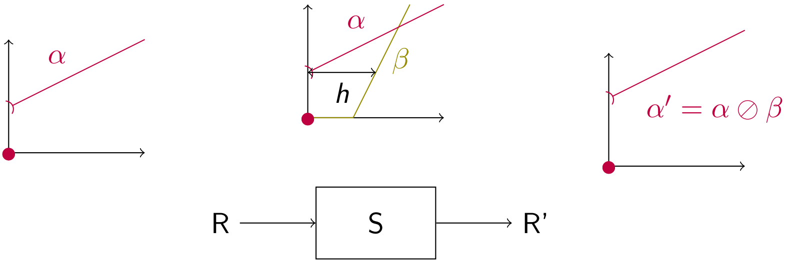
First results

Given:

- an arrival traffic contract
- a service guarantee

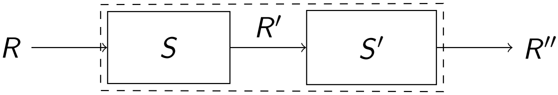
it can compute

- a delay bound (h)
- output traffic contract



Outline

Pay burst only once principle



Pay burst only once

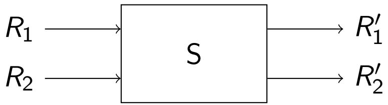
The sequence S, S' can be replaced by a virtual server $S;S'$ with service curve $\beta * \beta'$.

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

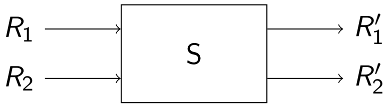
$$h(\alpha, \beta * \beta') \leq h(\alpha, \beta) + h(\alpha, \beta') \quad (3)$$

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

FIFO Aggregate scheduling



FIFO Aggregate scheduling

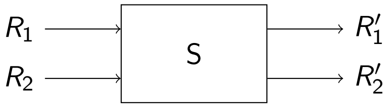


First FIFO result: aggregated delay (Th. 1)

If $d = h(\alpha_1 + \alpha_2, \beta)$ is the delay for the aggregated flow, then δ_d is a service curve for each flow.

$$\alpha'_i(t) = (\alpha'_i \oslash \delta_d)(t) = \alpha(t + d)$$

FIFO Aggregate scheduling



First FIFO result: aggregated delay (Th. 1)

If $d = h(\alpha_1 + \alpha_2, \beta)$ is the delay for the aggregated flow, then δ_d is a service curve for each flow.

$$\alpha'_i(t) = (\alpha'_i \oslash \delta_d)(t) = \alpha(t + d)$$

Second FIFO result: residual service (Th. 2)

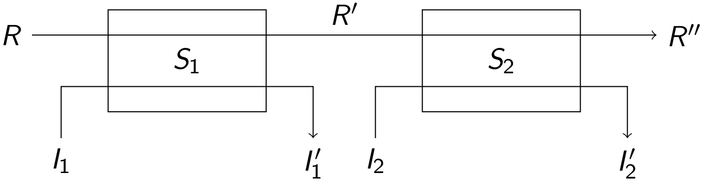
Let be $\theta \geq 0$ then, β_i^θ is a service curve for flow R_i

$$\beta_i^\theta = [\beta - \alpha_j \oslash \delta_\theta]^+ \mathbb{1}_{\{>\theta\}} \qquad \alpha'_i = \alpha_i \oslash \beta_i^\theta$$

with $\mathbb{1}_{\{>\theta\}}(x) = 1$ if $x > \theta$, 0 otherwise.

Outline

Considered topologies



- Tandem topology
- One flow of interest R
- One interfering flow I_i per server S_i

Two approaches

Local delay and shaping

- Global delay as sum of local delays
- Use of Th 1 (FIFO: aggregate result)
- University of Toulouse (IRIT, Networks and Telecommunication group)

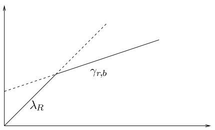
PBOO without shaping

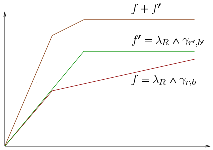
- End to end delay with Pay Burst Only Once result
- Use of Th 2 (FIFO: residual service)
- University of Pisa (Computing Networking Group)

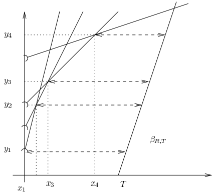
Shaping modelling

Applicative traffic is shaped by link capacity.

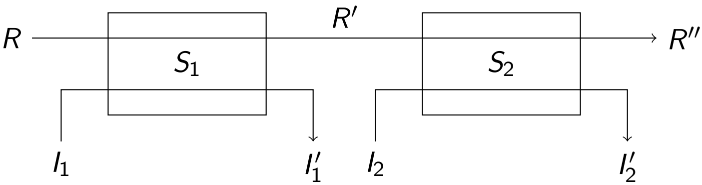
- new kind of curve (CPL)
- aggregate delay simple to compute





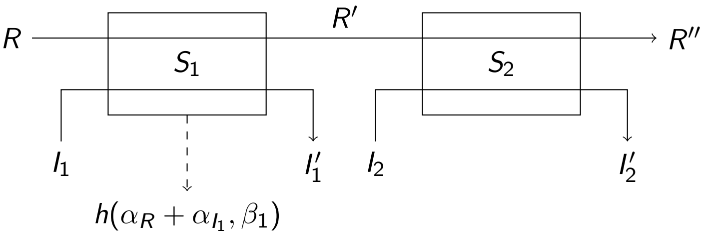


Computation by sum of local delays



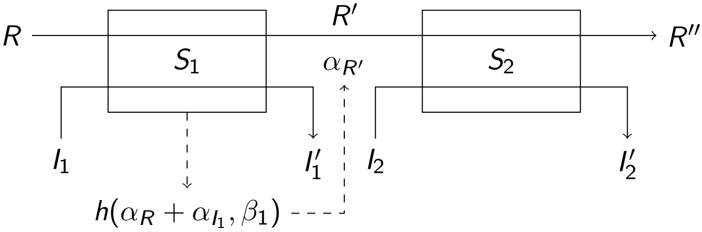
- modelling: CPL arrival curve for R and I_i (full shaping)
- propagation of result: aggregate delay
- end-to-end delay: sum of individual delays

Computation by sum of local delays



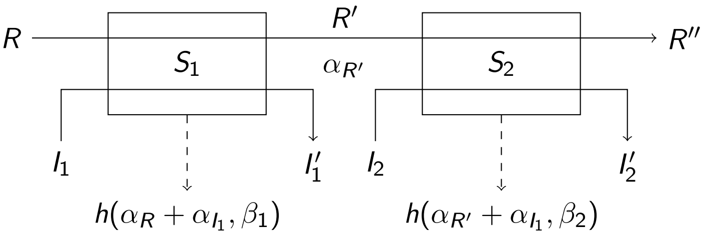
- modelling: CPL arrival curve for R and I_i (full shaping)
- propagation of result: aggregate delay
- end-to-end delay: sum of individual delays

Computation by sum of local delays



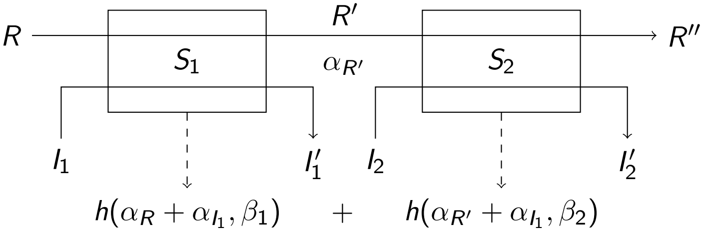
- modelling: CPL arrival curve for R and I_i (full shaping)
- propagation of result: aggregate delay
- end-to-end delay: sum of individual delays

Computation by sum of local delays



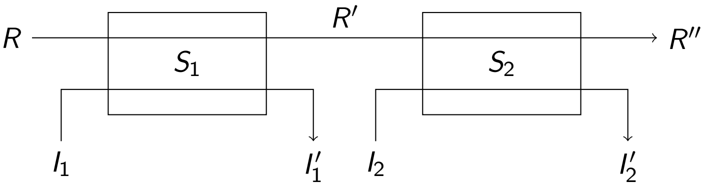
- modelling: CPL arrival curve for R and I_i (full shaping)
- propagation of result: aggregate delay
- end-to-end delay: sum of individual delays

Computation by sum of local delays



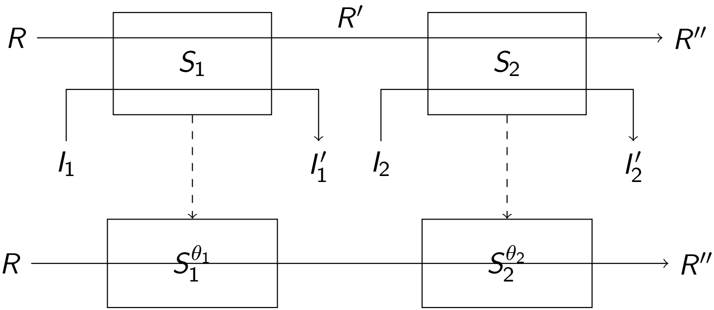
- modelling: CPL arrival curve for R and I_i (full shaping)
- propagation of result: aggregate delay
- end-to-end delay: sum of individual delays

PBOO optimisation (LUB)



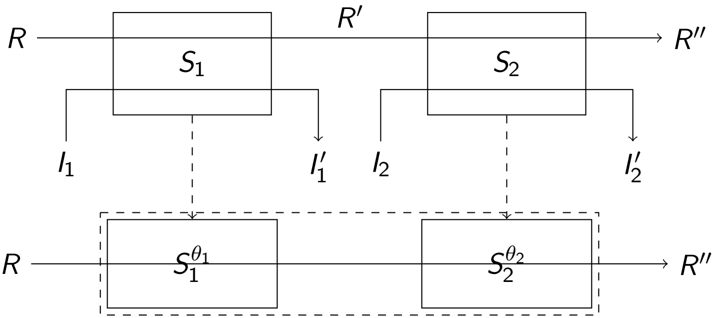
- modelling: token bucket curve for R and I_i (no shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

PBOO optimisation (LUB)



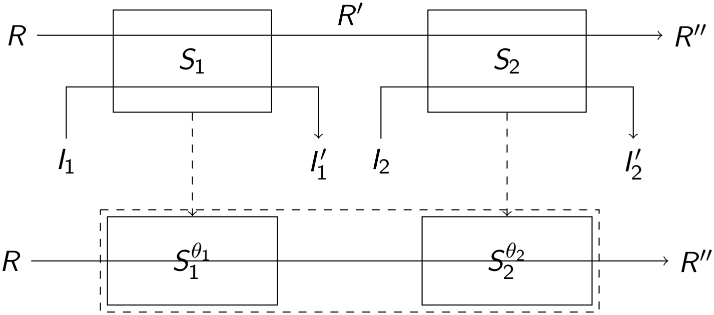
- modelling: token bucket curve for R and I_i (no shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

PBOO optimisation (LUB)



- modelling: token bucket curve for R and I_i (no shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

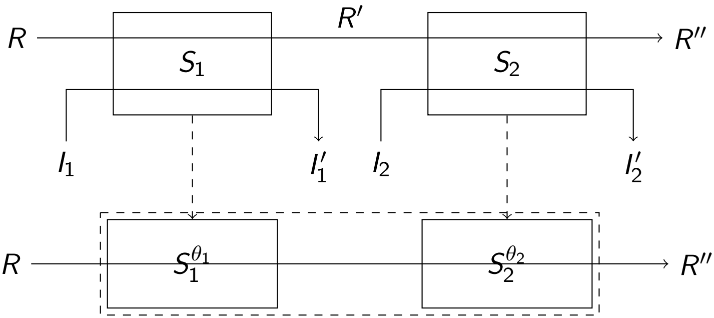
PBOO optimisation (LUB)



$h(\alpha, \beta_1^{\theta_1} * \beta_2^{\theta_2})$

- modelling: token bucket curve for R and I_i (no shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

PBOO optimisation (LUB)

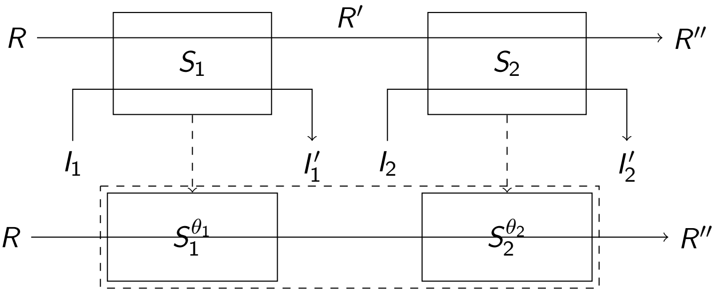


$$\inf_{\theta_1,\theta_2} \left\{ h(\alpha, \beta_1^{\theta_1} * \beta_2^{\theta_2}) \right\}$$

- modelling: token bucket curve for R and I_i (no shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

Outline

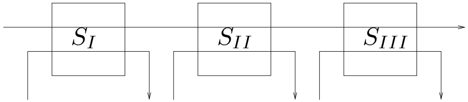
Contribution: half-modelling of shaping



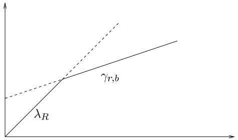
$$\inf_{\theta_1,\theta_2} \left\{ h(\alpha, \beta_1^{\theta_1} * \beta_1^{\theta_1}) \right\}$$

- modelling: CPL curve for R and token bucket for I_i (half modelling of shaping)
- propagation of result: equivalent server
- end-to-end delay: PBOO
- hard point: choice of θ_i

Experiment



- Three identical servers
- Identical interfering flows
- Two rates CPL, approximated by token bucket if needed



Experimental results ;-)

Configurations

Conf	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	1
T	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
r	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
b	1	1	5	$\frac{1}{5}$	1	1	5	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{5}{10}$	$\frac{1}{50}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{5}{10}$	$\frac{1}{50}$
r'	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
b'	1	5	1	$\frac{1}{5}$	1	5	1	$\frac{1}{5}$	$\frac{1}{10}$	$\frac{5}{10}$	$\frac{1}{10}$	$\frac{1}{50}$	$\frac{1}{10}$	$\frac{5}{10}$	$\frac{1}{10}$	$\frac{1}{50}$
$\rho = \frac{r+r'}{R}$	67%	67%	67%	67%	13%	13%	13%	13%	67%	67%	67%	67%	33%	33%	33%	33%
Delai R crossing $S_I; S_{II}$																
LUB	5.50	13.50	11.50	2.70	2.61	4.21	3.47	2.12	2.35	3.15	2.95	2.07	2.32	3.12	2.80	2.06
Loc. Shap.	5.41	10.50	9.75	2.81	2.43	2.62	2.54	2.09	2.49	3.12	2.92	2.23	2.27	2.60	2.44	2.08
Half. Shap.	4.75	12.75	7.75	2.55	2.41	4.01	2.47	2.08	2.27	3.07	2.57	2.05	2.22	3.02	2.32	2.04
Delai R crossing $S_I; S_{II}; S_{III}$																
LUB	7.50	19.50	13.50	3.90	3.81	6.21	4.67	3.16	3.45	4.65	4.05	3.09	3.42	4.62	3.90	3.08
Loc. Shap.	8.81	18.50	15.87	4.58	3.66	4.07	3.83	3.14	4.05	5.19	4.76	3.63	3.47	4.20	3.72	3.17
Half. Shap.	6.75	18.75	9.75	3.75	3.61	6.01	3.67	3.12	3.37	4.57	3.67	3.07	3.32	4.52	3.42	3.06
Gain on the new method for R crossing $S_I; S_{II}$																
vs. LUB	13.63%	5.55%	32.60%	5.55%	7.61%	4.72%	28.65%	1.87%	3.19%	2.38%	12.71%	0.72%	4.13%	3.07%	17.14%	0.93%
vs. Loc. Shap.	12.30%	-21.42%	20.51%	9.46%	0.78%	-52.81%	2.83%	0.36%	8.69%	1.60%	11.96%	7.91%	2.34%	-16.30%	4.91%	1.79%
Gain on the new method for R crossing $S_I; S_{II}; S_{III}$																
vs. LUB	10.00%	3.84%	27.77%	3.84%	5.21%	3.20%	21.29%	1.25%	2.17%	1.61%	9.25%	0.48%	2.80%	2.07%	12.30%	0.62%
vs. Loc. Shap.	23.46%	-1.35%	38.58%	18.23%	1.22%	-47.64%	4.06%	0.64%	16.80%	11.94%	22.83%	15.37%	4.29%	-7.54%	8.09%	3.48%

Interpretation

- new method always better than LUB (direct generalisation)
  - gain depends on burst sizes
  - gain independent on path lenght
- new method vs “shaping+local delays”
  - depends on interfering burst size (not shaped)
  - gain increases with path length (PBOO)

Outline

Conclusion

To have better bounds, two aspects must be modelled:

- shaping
- pay burst only once

FIFO in network calculus:

- local delay and shaping
- PBOO without shaping

Our contribution:

- Half modelling of shaping + PBOO
- $O(n \log(n))$ complexity (sorting and sums)

Future works: full modelling of shaping