Network Calculus

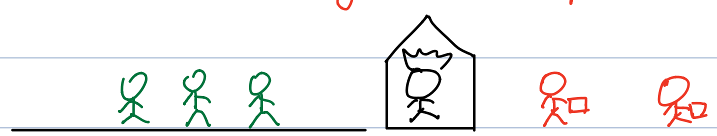
An example for Network Calculus :

Suppose the scenario of Holy Cow! at EPFL

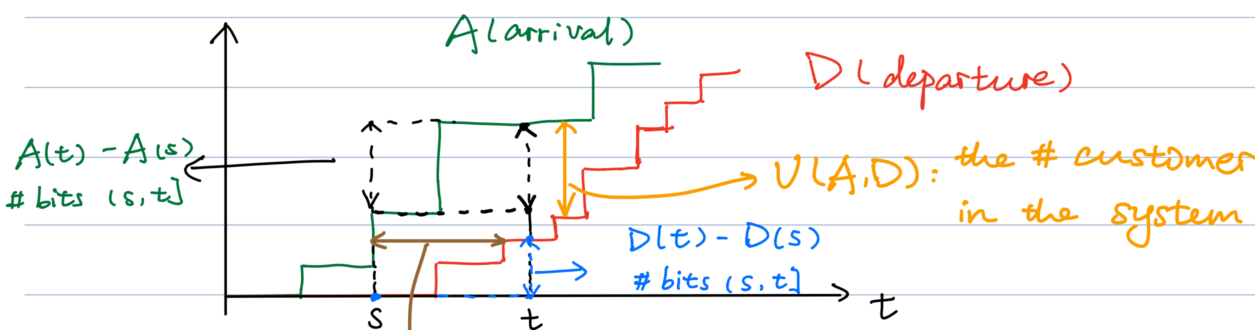
: customers waiting to order (input packets)

: chef (servers)

: customers who g- their food (output packets)



The diagram is shown below :



h(A, D) : the waiting time for a customer (delay)

Arrival curve α

$(s, t]$ $\qquad A(t) - A(s) \le \alpha(t-s)$

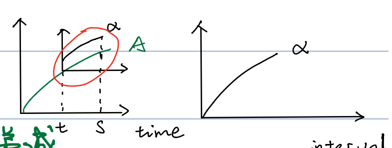
bits (s, t] $\qquad$ duration of interval

Def. Flow with cumulative function

$R(t)$ has α as (maximal) arrival curve curve if

$R(t) - R(s) \le \alpha(t-s)$

for any $t \geq s \geq 0$



单调非递减

a monotonic nondecreasing function $R^{+} \rightarrow [0, +\infty]$

Aggregation Property

If every flow f has arrival curve α_f, then the aggregation $R = \sum_f R_f$ has arrival curve $\sum_f \alpha_f$

Min-Plus Convolution of $f_1, f_2 \geq 0$

卷积

$$f(t) = \inf_{s \geq 0} \left(f_1(s) + f_2(t-s) \right)$$

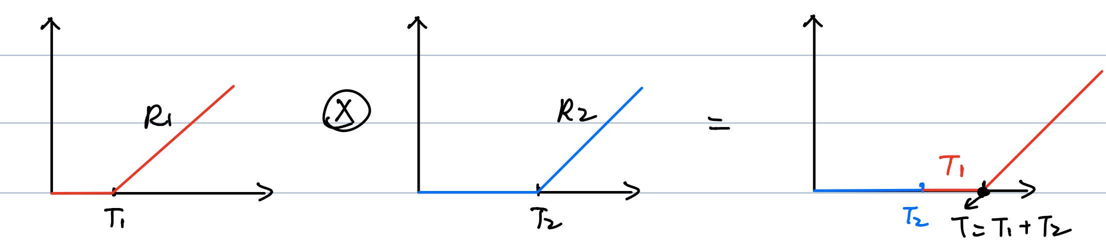
$$f = f_1 \otimes f_2$$

the max lower bound

properties:

1. $(f_1 \otimes f_2) \otimes f_3 = f_1 \otimes (f_2 \otimes f_3)$
2. $f_1 \otimes f_2 = f_2 \otimes f_1$

eg.



Min-Plus Convolution and Arrival Curves

α is an arrival curve for $R \Leftrightarrow R(t) \leq R(s) + \alpha(t-s) \quad \forall s \in [0, t]$

$\Leftrightarrow R(t) \leq R \otimes \alpha$

Any arrival curve α can be replaced by its sub-additive closure

$\bar{\alpha} = \inf \{ \delta_0, \alpha, \alpha \otimes \alpha, \alpha \otimes \alpha \otimes \alpha, \cdots \}$

with $\delta_0(0) = 0$, $\delta_0(t) = +\infty$ for $t > 0$

$\bar{\alpha}$ is sub-additive, i.e. $\bar{\alpha}(s+t) \leq \bar{\alpha}(s) + \bar{\alpha}(t)$

and $\bar{\alpha}(0) = 0$

when $\alpha(0) = 0$, α is a curve for $R \Leftrightarrow R \otimes \alpha$

e.g. Flow has at most L bits in any interval of duration τ

$\Leftrightarrow R(t+\tau) - R(t) \leq L$ for all t

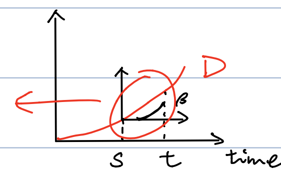
(Strict) service curve β

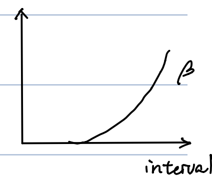
$(s,t]$ a flow is backlogged (busy)

$\Rightarrow \quad \forall \tau \in (s,t], \; A(\tau) > D(\tau)$

$\underbrace{D(t) - D(s)}_{\text{\# bits departured}} \geq \beta(\underbrace{t-s}_{\text{Duration}})$

$D(t) - D(s) \geq \beta(t-s)$





$R(t) \longrightarrow \text{System} \xrightarrow{R^*(t)}$

System offers to this flow a (minimal) service curve β if

$R^* \geq R \otimes \beta$

$\forall t \geq 0, \; \exists s \in [0,t] : \; R^*(t) \geq R(s) + \beta(t-s)$

where β is a function : $\mathbb{R}^+ \longrightarrow \mathbb{R} \cup \{+\infty\}$

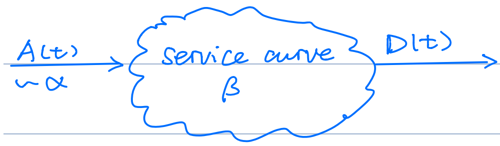
System S offers to a flow a strict service curve β if for any $s < t$ inside a backlogged period, i.e. such that $R^*(u) < R(u)$, $\forall u \in (s,t]$, we have $R^*(t) - R^*(s) \geq \beta(t-s)$

S is typically a single queuing point.

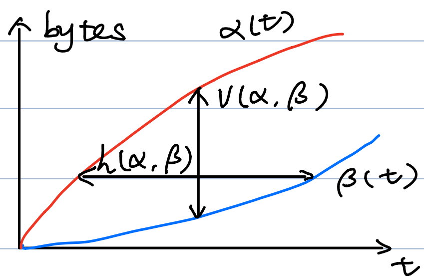
strict service curve $\Rightarrow$ service curve

but service curve may not be a strict sc

Three Tight Bounds



Flow is constrained by arrival curve α; served in network element with service curve β. Then

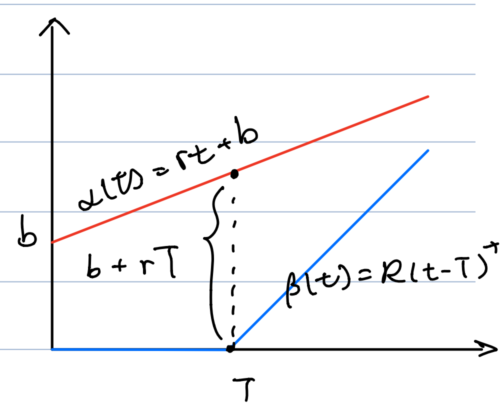


1. backlog $\le V(\alpha, \beta) = \sup_t (\alpha(t) - \beta(t))$
2. If FIFO for this flow,
   delay $\le h(\alpha, \beta)$
3. output is constrained by arrival curve $\alpha^*(t) = \sup_{u \ge 0} (\alpha(t+u) - \beta(u))$
   i.e. $\alpha^* = \alpha \oslash \beta$ (deconvolution) 去卷积

Example

One flow, constrained by one token bucket is served in a network element that offers a rate latency service curve

Assume $r \le R$



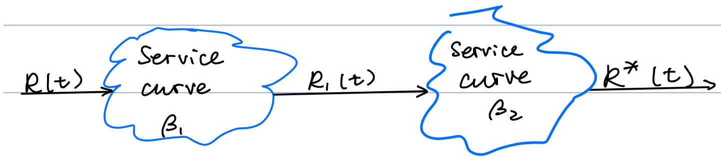
Backlog bound : $b + rT$

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

Output arrival curve : $\alpha^*(t) = rt + b^*$

with $b^* = b + rT$

Concatenation —系列相互关联的事



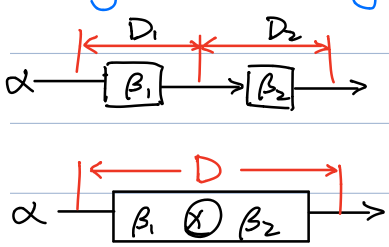
$$\boxed{\beta = \beta_1 \otimes \beta_2}$$

A flow is served in series, network element i offers service curve β_i. The concatenation offers to flow the service curve $\beta = \beta_1 \otimes \beta_2$

Proof: $R^* \geq R_1 \otimes \beta_2 \geq (R \otimes \beta_1) \otimes \beta_2 = R \otimes (\beta_1 \otimes \beta_2)$

If β_i is rate-latency R_i, T_i, then the concatenation β is rate-latency $R = \min(R_1, R_2)$ and $T = T_1 + T_2$

Pay Bursts Only Once



one flow constrained at source by α

end-to-end delay bound computed node-by-node (also accounting for increased burstiness at node 2):

$$D_1 + D_2 = \frac{2b + rT_1}{R} + T_1 + T_2$$

$\alpha(t) = rt + b$

$\beta_1(t) = R(t - T_1)^+$

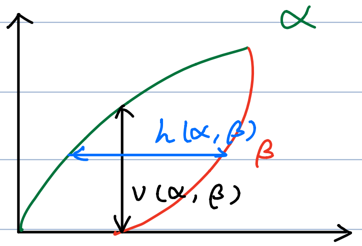
$\beta_2(t) = R(t - T_2)^+$

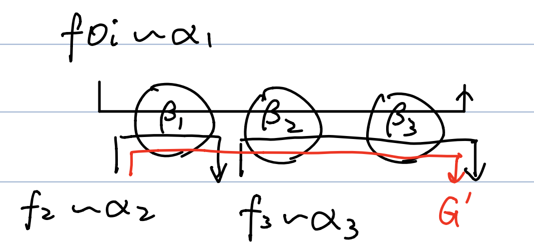
$r \leq R$

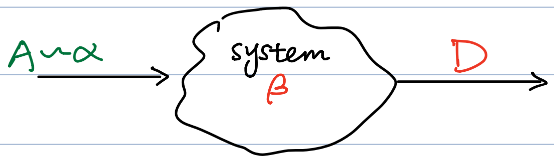
computed by concatenation:

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

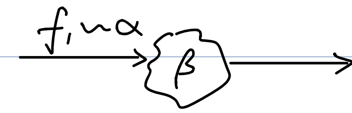
i.e. worst cases cannot happen simultaneously — concatenation captures this!





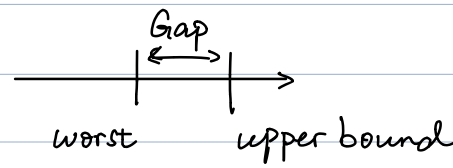


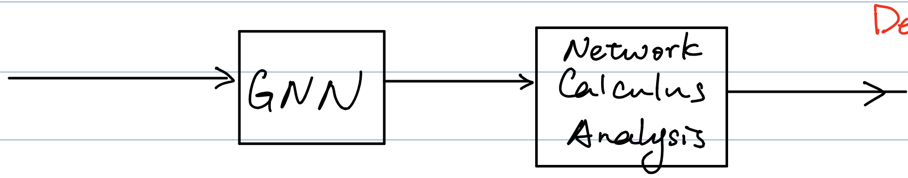
$\Downarrow$ D(G, foi)



Properties of the system

1) FIFO
2) Lossless
3) Causal

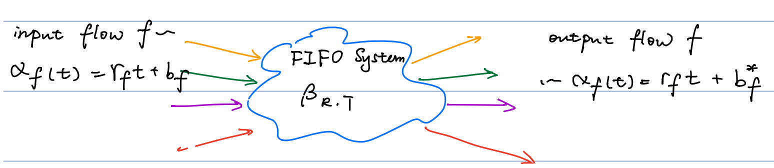




Delay bound for foi

"Delay" and "Backlog" are the two things that we are interested about.

Analyzing Per-class Networks

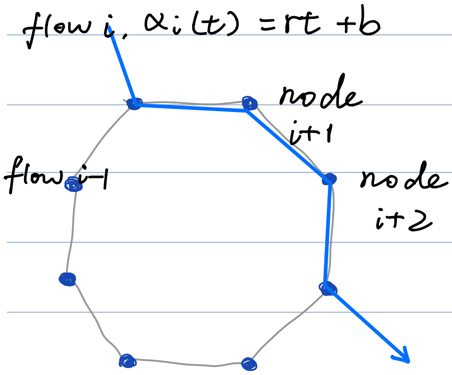


arrival curve for output f is leaky bucket r_f, b_f^* with

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

Burstiness of every flow inside network increases along its path as a function of other flows' burstiness (cascade)

e.g.



b_j = burstiness of every flow i after j hops with $b_0 = b$:

$$b_j = b_{j-1} + r\left(T + \frac{b_{tot} - b_{j-1}}{R}\right)$$

$$b_{tot} = b + b_1 + \cdots + b_{k-1}$$

Fixed point in $(b_{tot}, b_1, \cdots, b_{k-1})$ has a positive solution when

$(k-1)r + (1-r)^k < 1$, which occurs for $r < r_{crit}$

Other Methods

- Several techniques improve delay bounds, tightness and u_{crit}: PMOO, PMOC, linear programming, etc.
- There cannot be a bound that depends only on max aggregate burstiness, number of hops h and link utilization u when $u \geq \frac{1}{h-1}$

Stability of a FIFO Network

A network instance is stable if there is a bound on all delays (or backlogs), that is valid for any execution trace of the network. (existence of a bound on all delays $\Leftrightarrow$ existence of a bound on all backlogs)

- An overloaded FIFO network is NOT stable
- A single-node network that is underloaded or critical is stable
- A feed-forward network that is underloaded or critical is stable ↳ [Boyer et al 2011]
- For any $\varepsilon > 0$ there is an unstable underloaded FIFO network with load factor $u < \varepsilon$
- Every underloaded ring is stable
- If the interference condition holds and service curves are strict then the FIFO network is stable; in the special case $R_i = R$, $\forall i$, the condition is $r_f \le R/(1+RIN_f)$ where RIN_f is the number of flows that interfere with f.

Conclusion:

- Network calculus main concepts:
  ① arrival curve,
  ② minimal service curve and universal bounds shapers
  ③ concatenation for packet-based systems and for fluid systems
- Some key results are based on existence of minimal or maximal solutions to functional problems.
- Challenges exist for per-class systems
- Fundamental results use min-plus convolution; other techniques (dater-based) can be useful.

[illegible]