Modeling Earliest Deadline First (EDF) Scheduling with Real Time Calculus

TEC Group meeting

11. December 2007

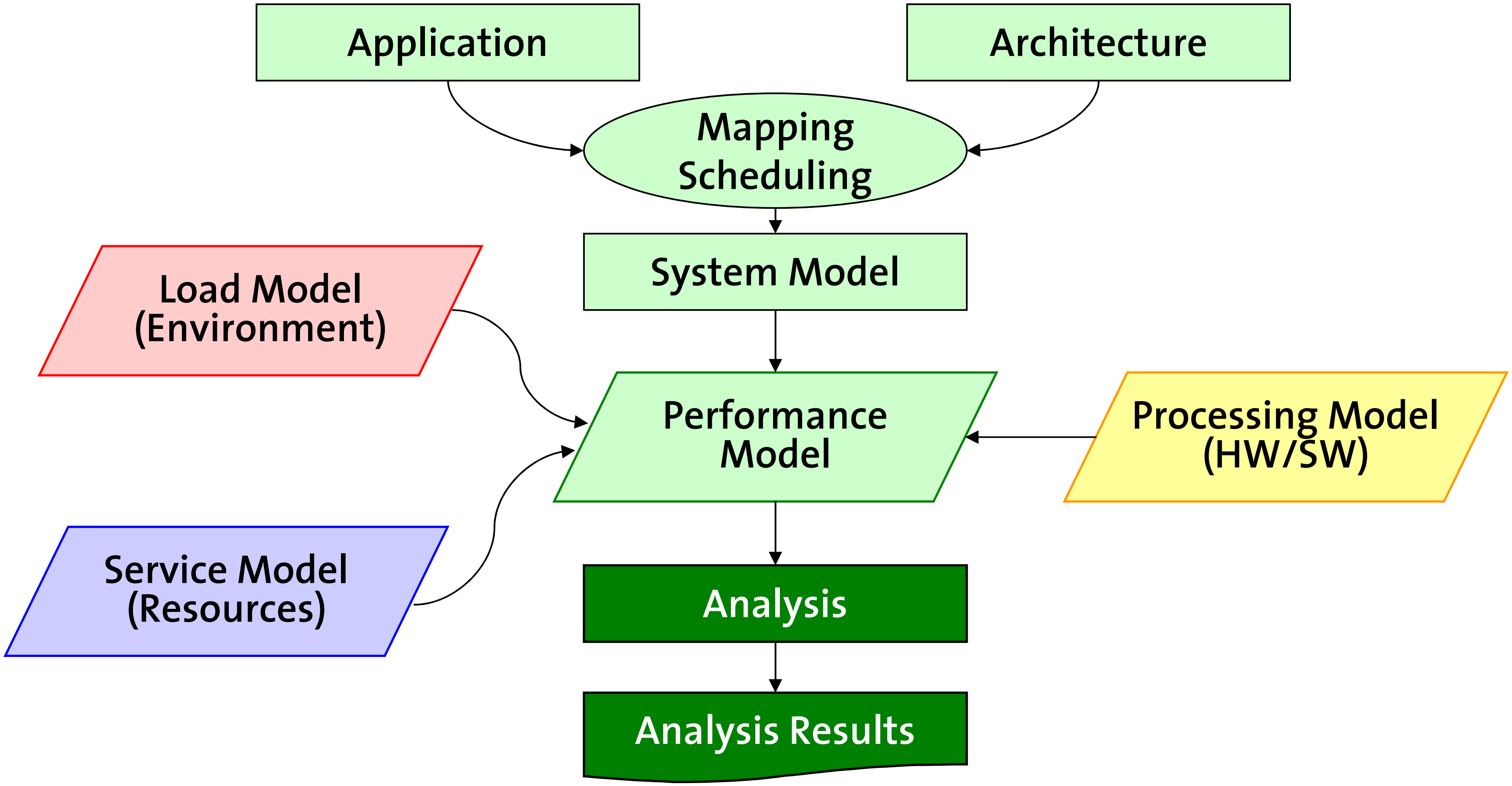
Simon Perathoner

Computer Engineering and Networks Laboratory
ETH Zürich, Switzerland

Outline

- Modular Performance Analysis
- Earliest Deadline First Scheduling
- Modeling of EDF component with RTC
  - Feasibility test
  - Remaining service
  - Outgoing event streams

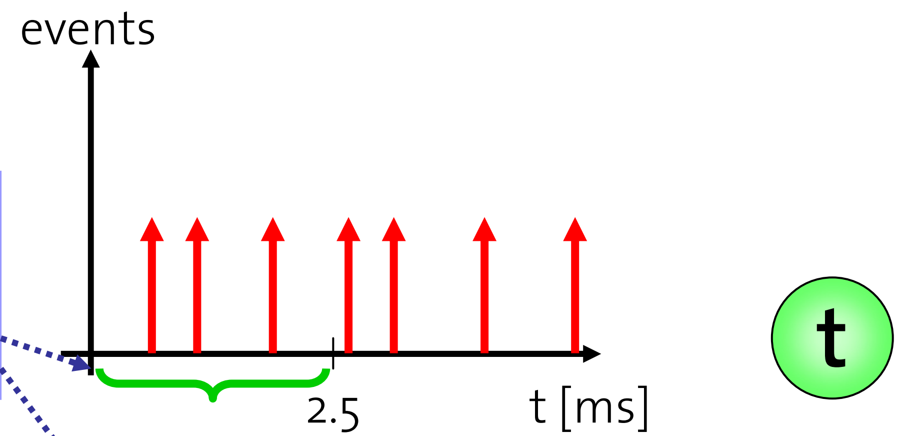
Modular Performance Analysis



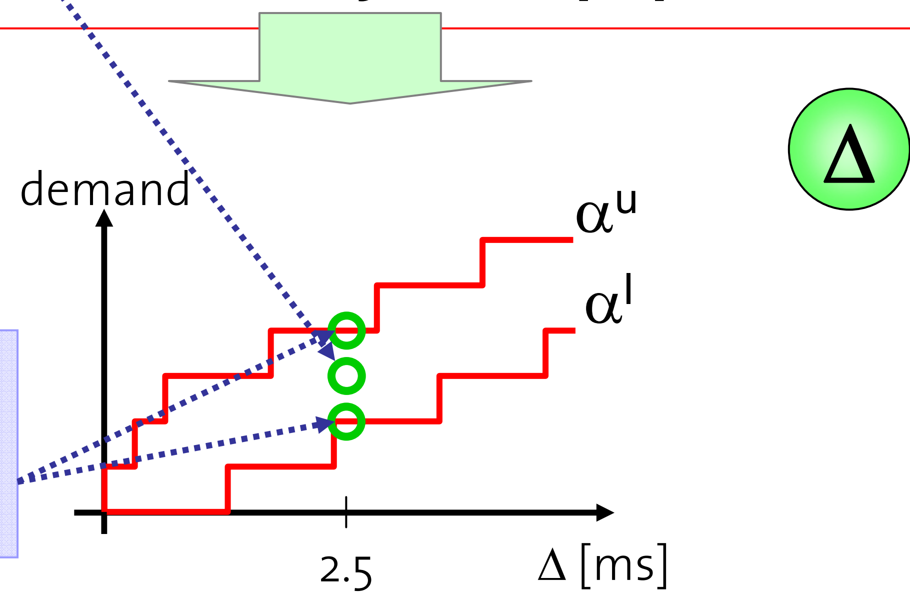
Load Model (Environment)

Event Stream

number of events in
in t=[0 .. 2.5] ms



Arrival Curve α



maximum / minimum
arriving demand in *any*
interval of length 2.5 ms

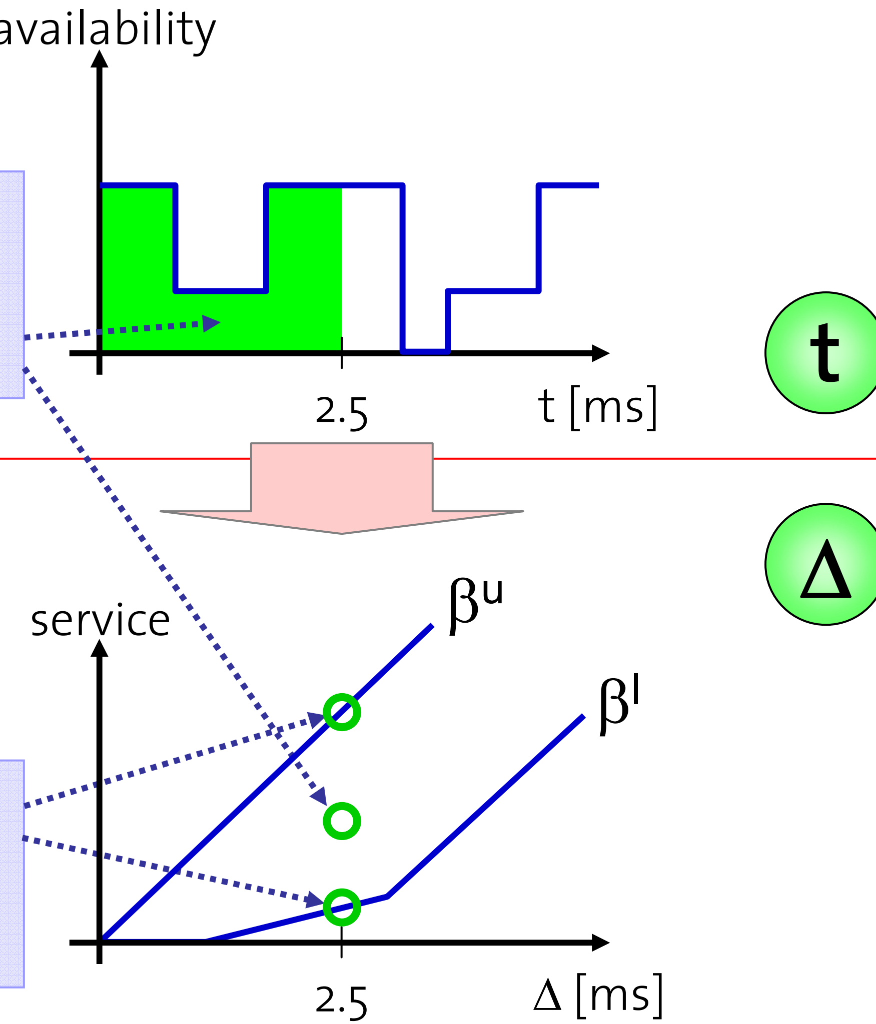
Service Model (Resources)

Resource Availability

available service in t=[0 .. 2.5] ms

Service Curves [β^l, β^u]

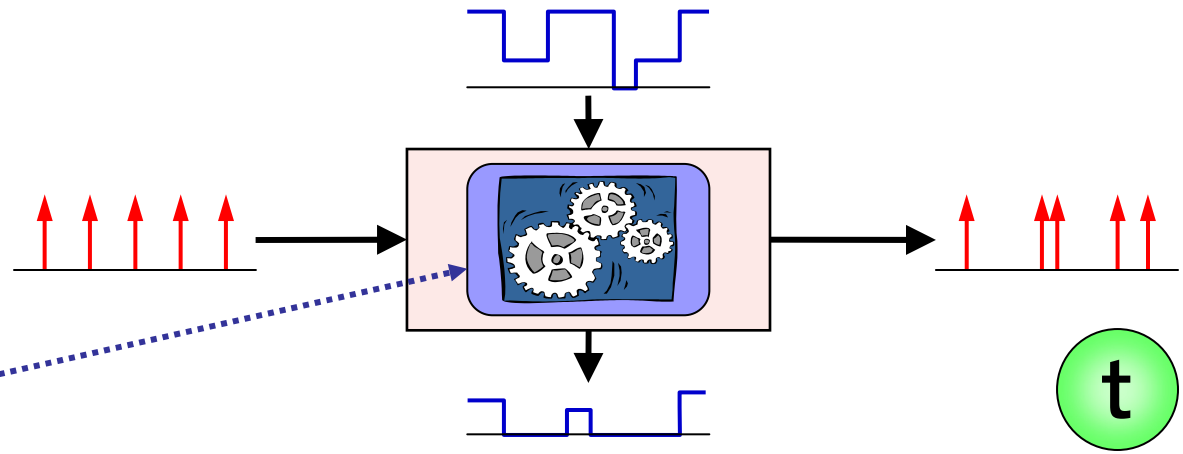
maximum/minimum available service in *any interval* of length 2.5 ms



Processing Model (HW/SW)

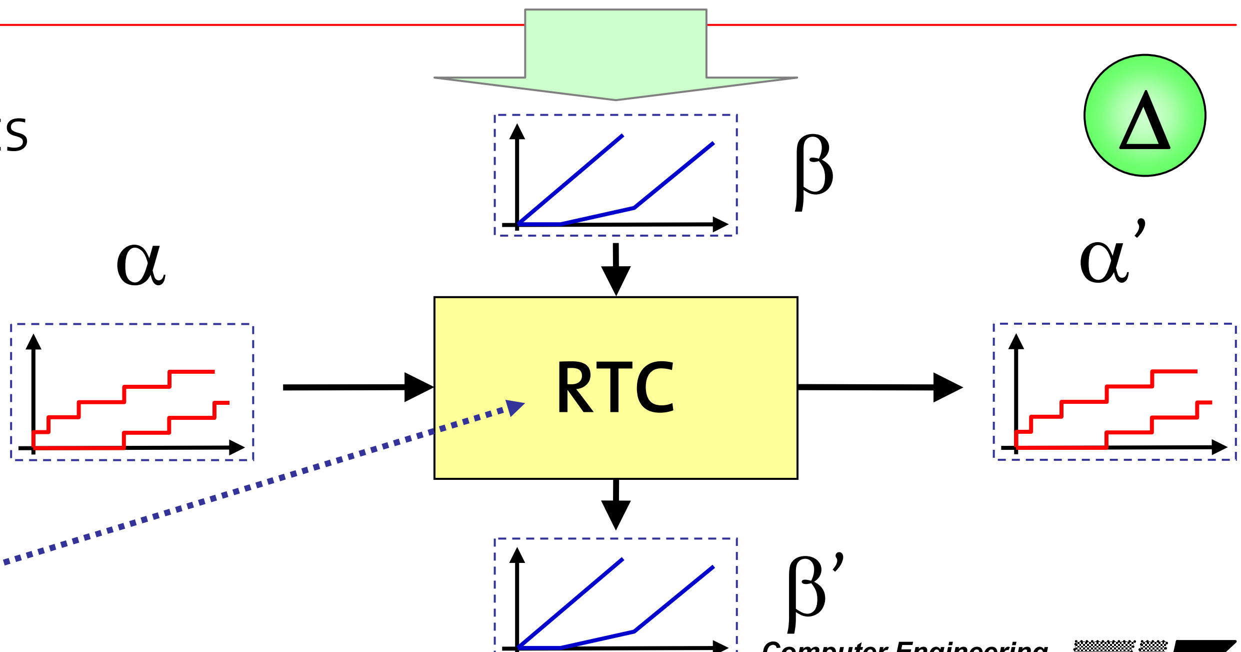
HW/SW Components

Processing semantics and functionality of hardware or software tasks



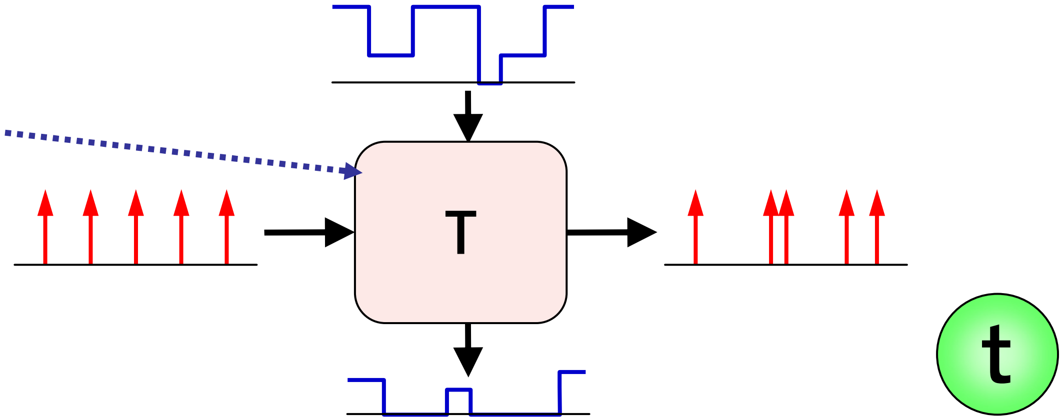
Abstract Components

$$\alpha'(\Delta) = f_\alpha(\alpha, \beta)$$
$$\beta'(\Delta) = f_\beta(\alpha, \beta)$$



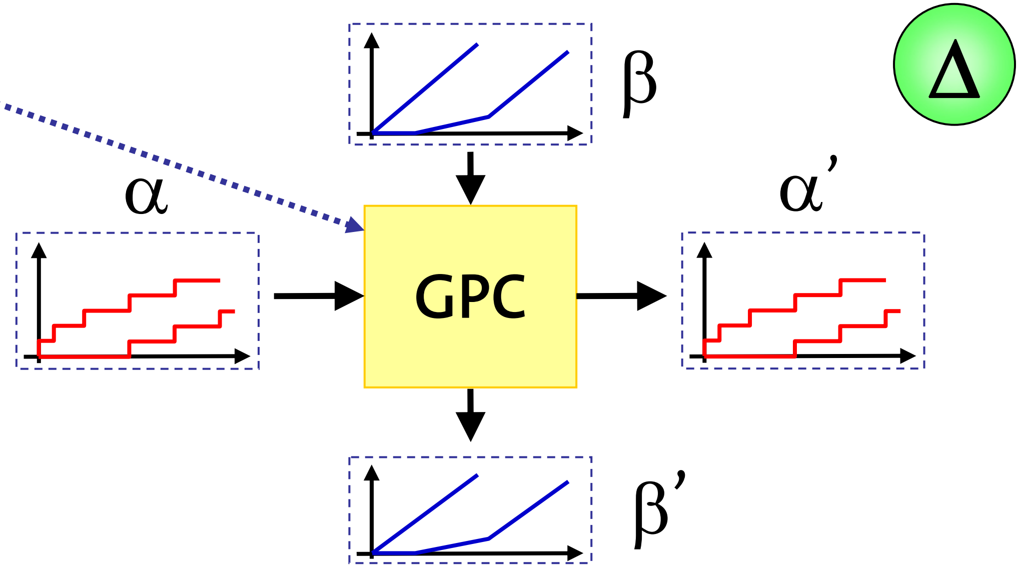
Greedy Processing component

Preemptive
Fixed Priority Task

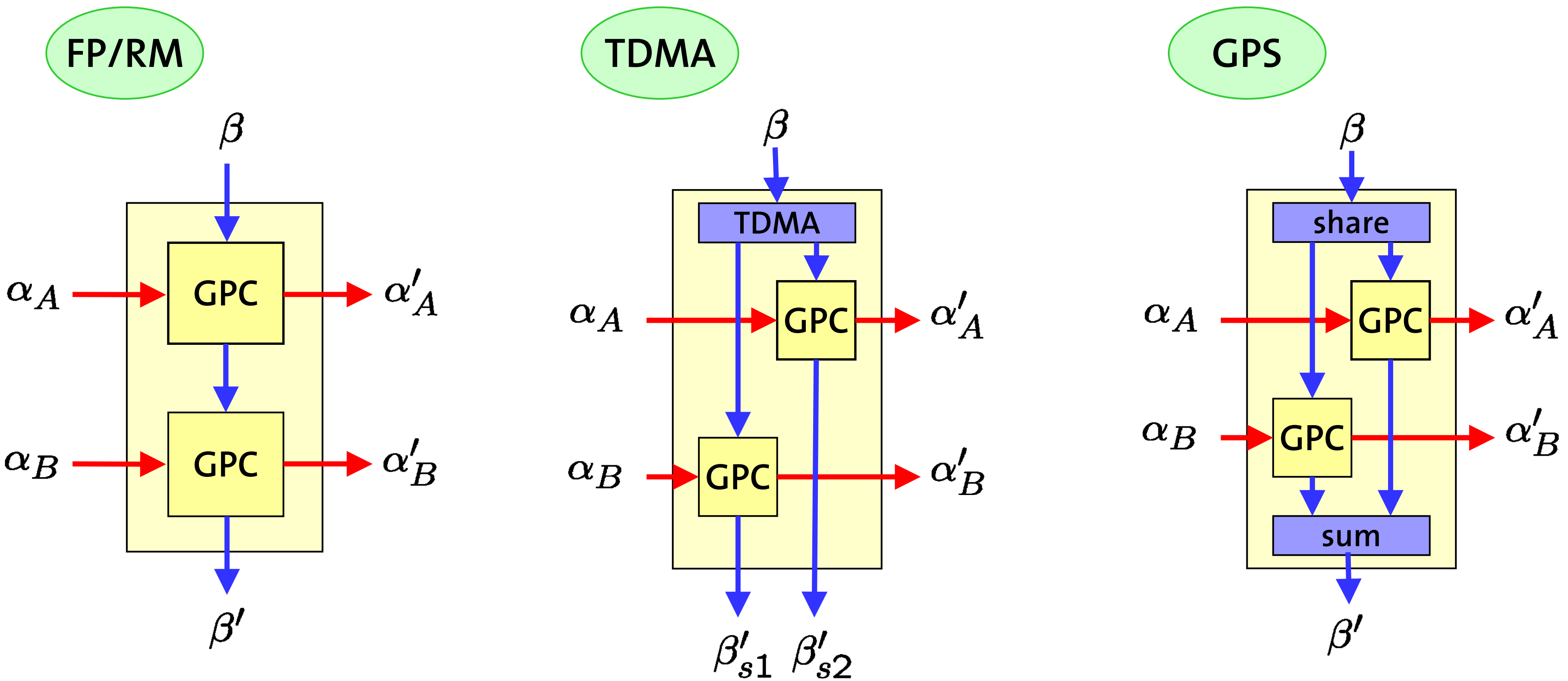


Real-Time Calculus

$$\begin{aligned}
\alpha'^{u} &= \min\{(\alpha^{u} \otimes \beta^{u}) \oslash \beta^{l}, \beta^{u}\} \\
\alpha'^{l} &= \min\{(\alpha^{l} \oslash \beta^{u}) \otimes \beta^{l}, \beta^{l}\} \\
\beta'^{u} &= (\beta^{u} - \alpha^{l}) \,\overline{\oslash}\, 0 \\
\beta'^{l} &= (\beta^{l} - \alpha^{u}) \,\overline{\otimes}\, 0
\end{aligned}$$



Static Scheduling Policies



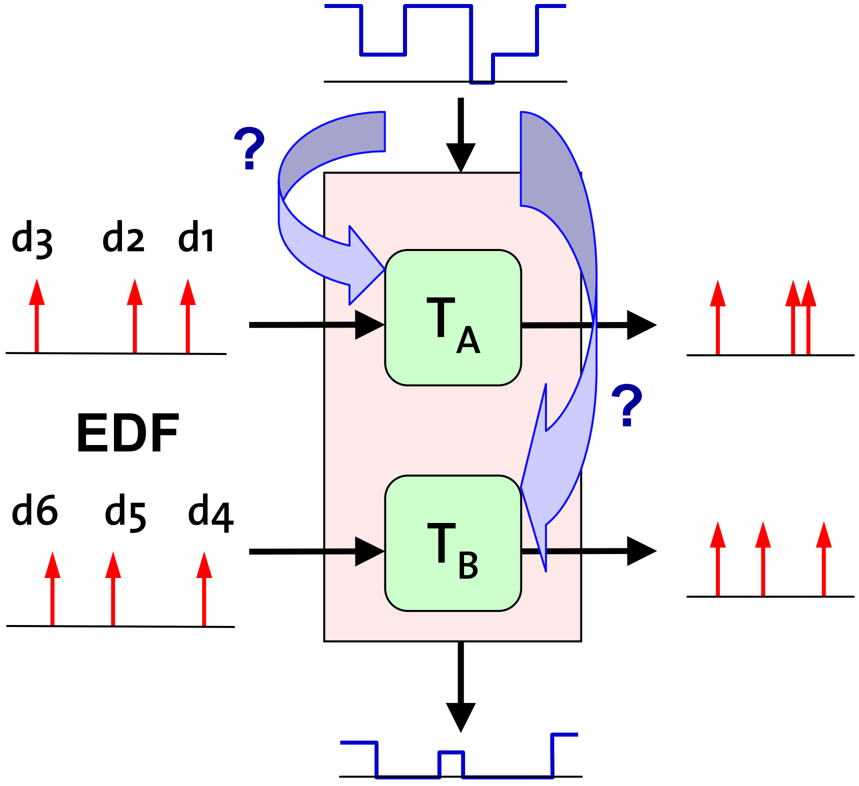
Several static scheduling policies can be modeled by composition of GPC components

Dynamic Scheduling Policies

Dynamic scheduling policies require different models

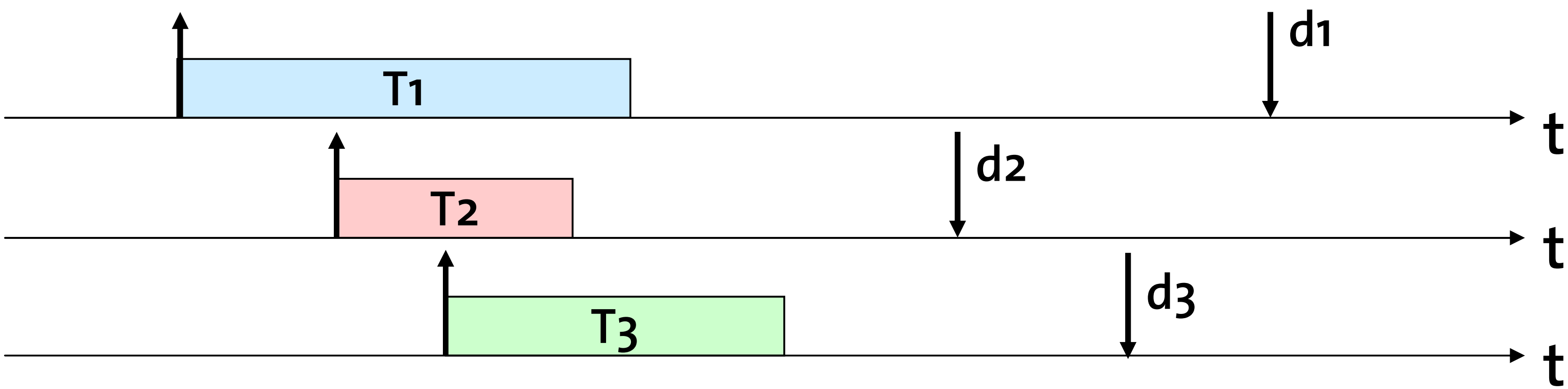
Example:

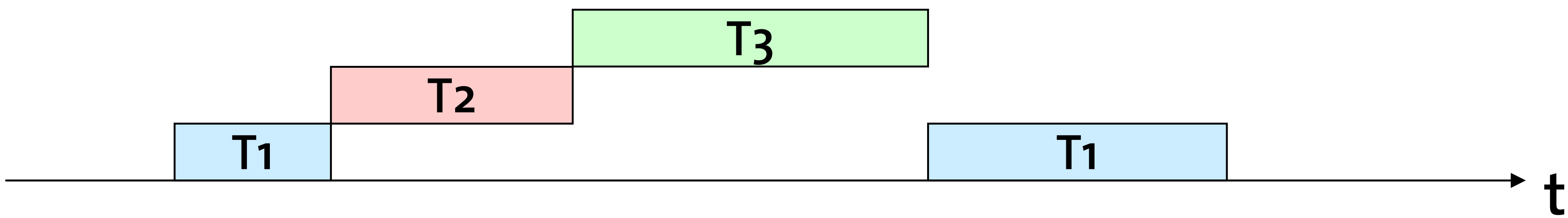
How to model **EDF** scheduling with RTC ?



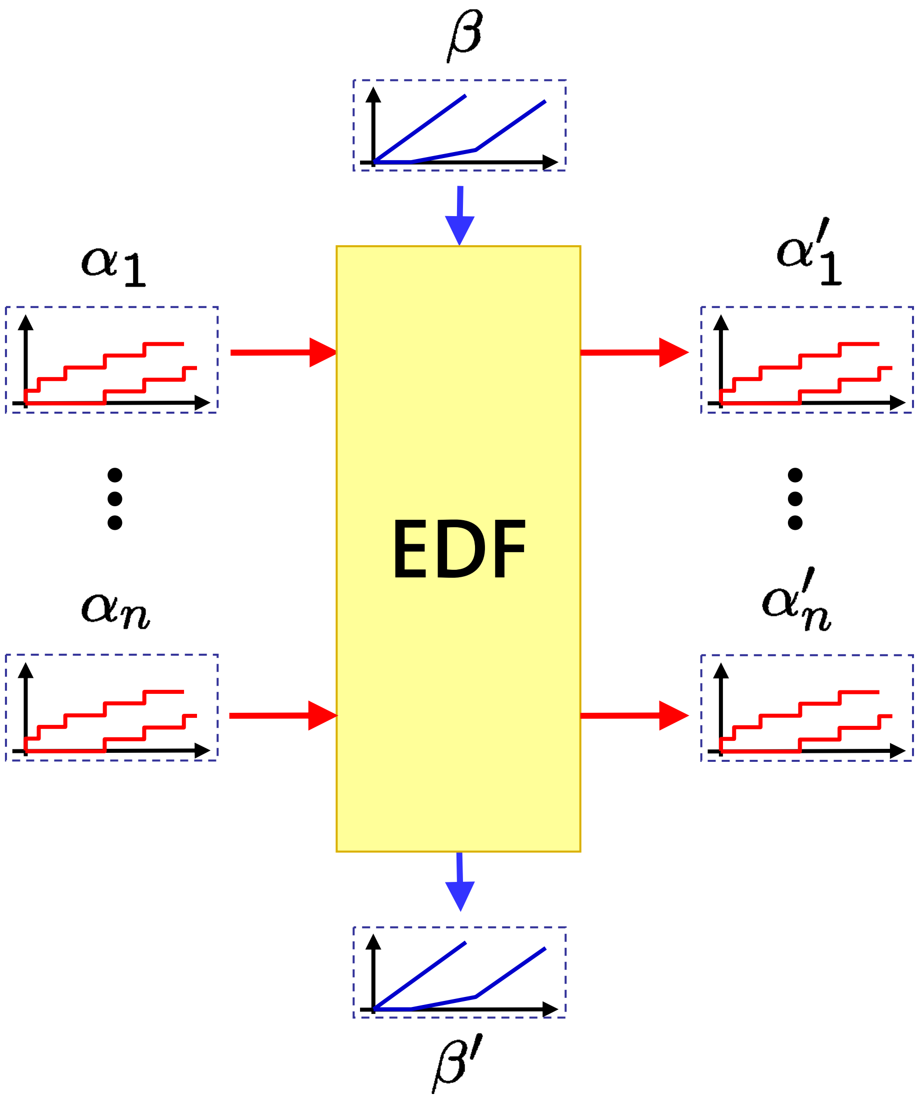
Earliest Deadline First (EDF) Scheduling

At any instant execute the task with the earliest absolute deadline





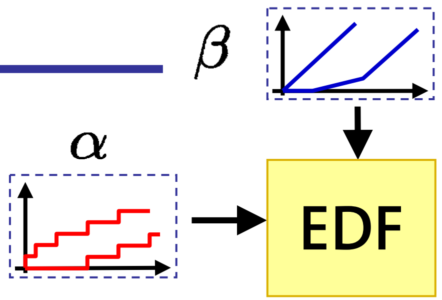
EDF component



We need RTC equations for:

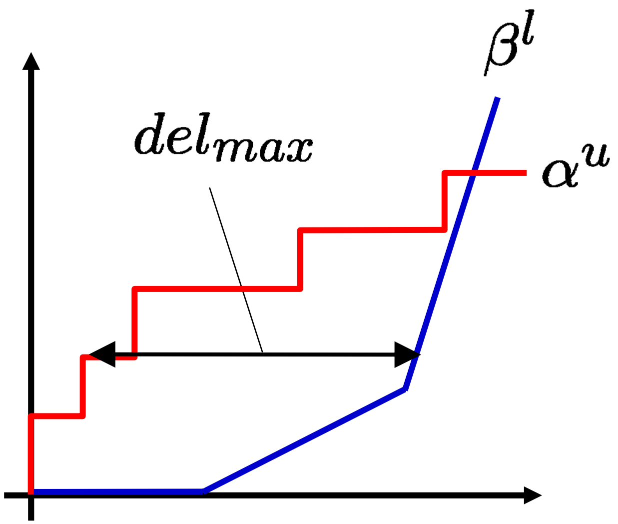
1. Feasibility test
2. Remaining service β'
3. Output event streams
   α'_1 ... α'_n

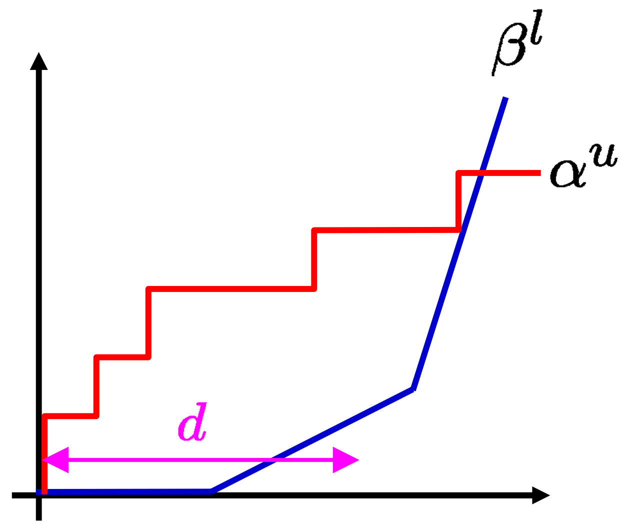
Feasibility test



- Simple case: only 1 event stream with deadline d

Feasibility test: $del_{max} \leq d$





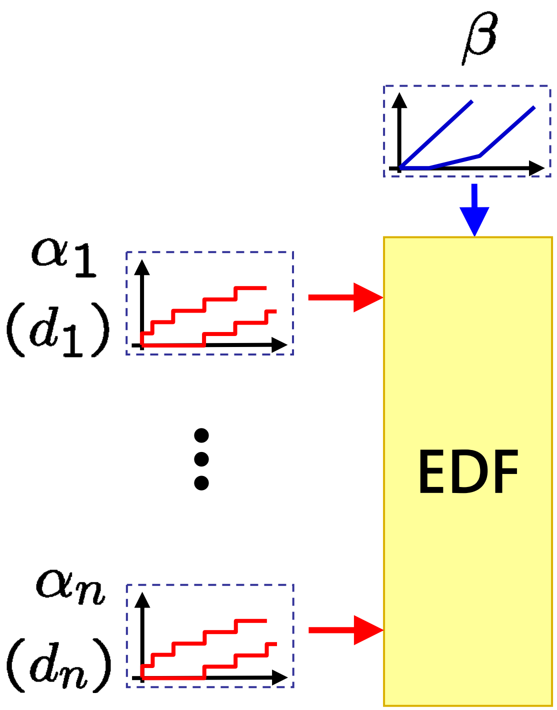
$$hdist(w \cdot \alpha^u(\Delta), \beta^l(\Delta)) \leq d$$

$$hdist(w \cdot \alpha^u(\Delta - d), \beta^l(\Delta)) \leq 0$$

$$w \cdot \alpha^u(\Delta - d) \leq \beta^l(\Delta)$$

Feasibility test

- n event streams with deadlines $d_1, \ldots, d_n$



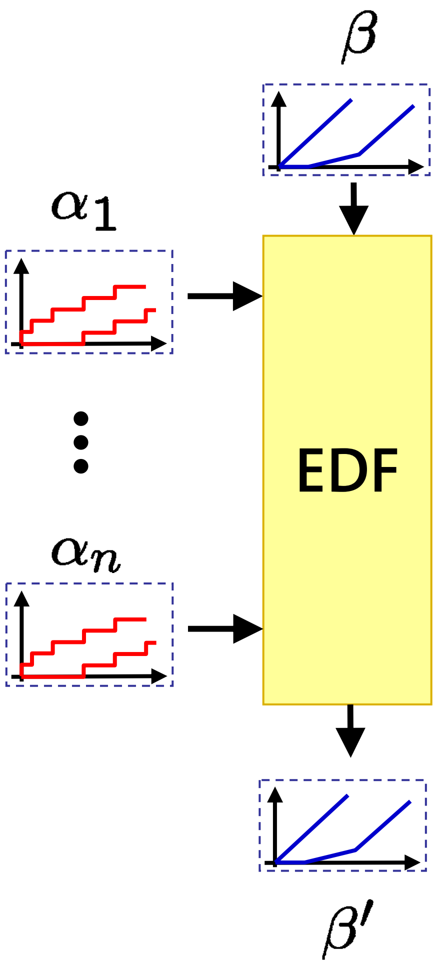
1 **Test**

$$\sum_i w_i \, \alpha_i^u(\Delta - d_i) \le \beta^l(\Delta)$$

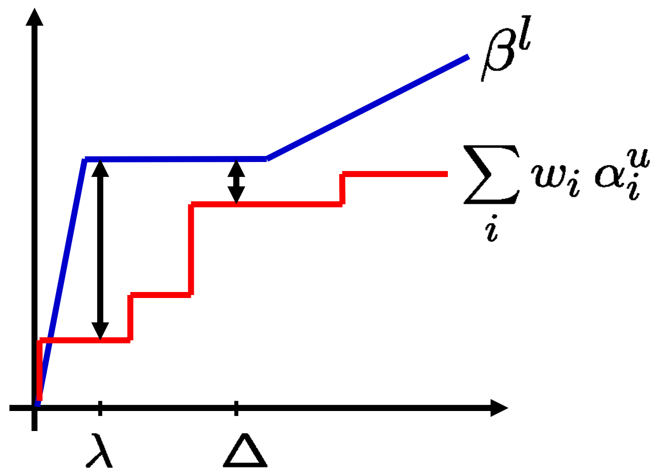
true → Taskset schedulable

false → Taskset not schedulable

Remaining service



$$\beta^l(\Delta) - \sum_i w_i\, \alpha_i^u(\Delta)$$

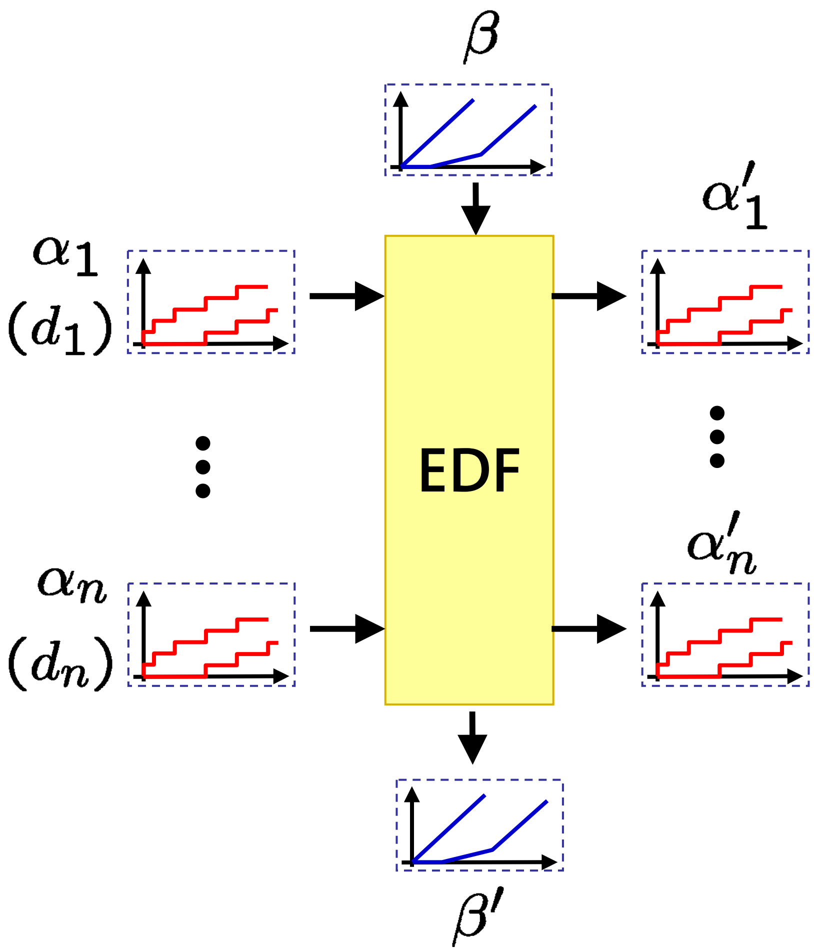


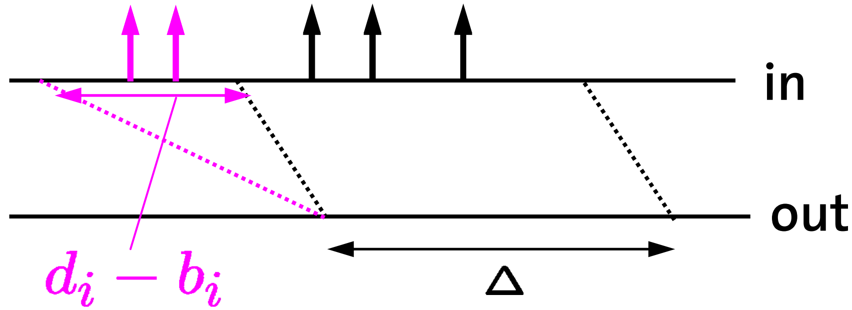
2

$$\beta'^{\,l}(\Delta) = \sup_{0 \le \lambda \le \Delta} \left\{ \beta^l(\lambda) - \sum_i w_i\, \alpha_i^u(\lambda) \right\}$$

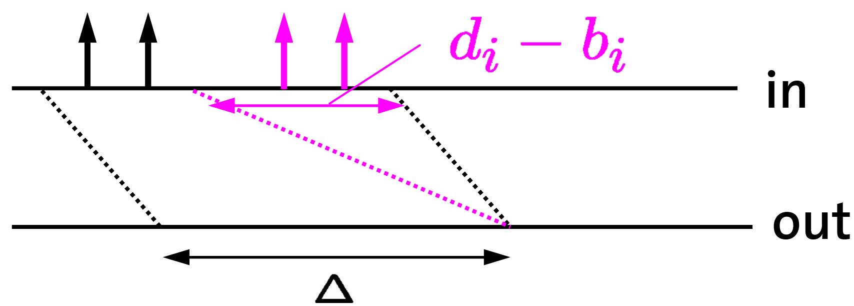
$$\beta'^{\,u}(\Delta) = \inf_{\Delta \le \lambda} \left\{ \beta^u(\lambda) - \sum_i b_i\, \alpha_i^l(\lambda) \right\}$$

Output event streams – Simple bounds



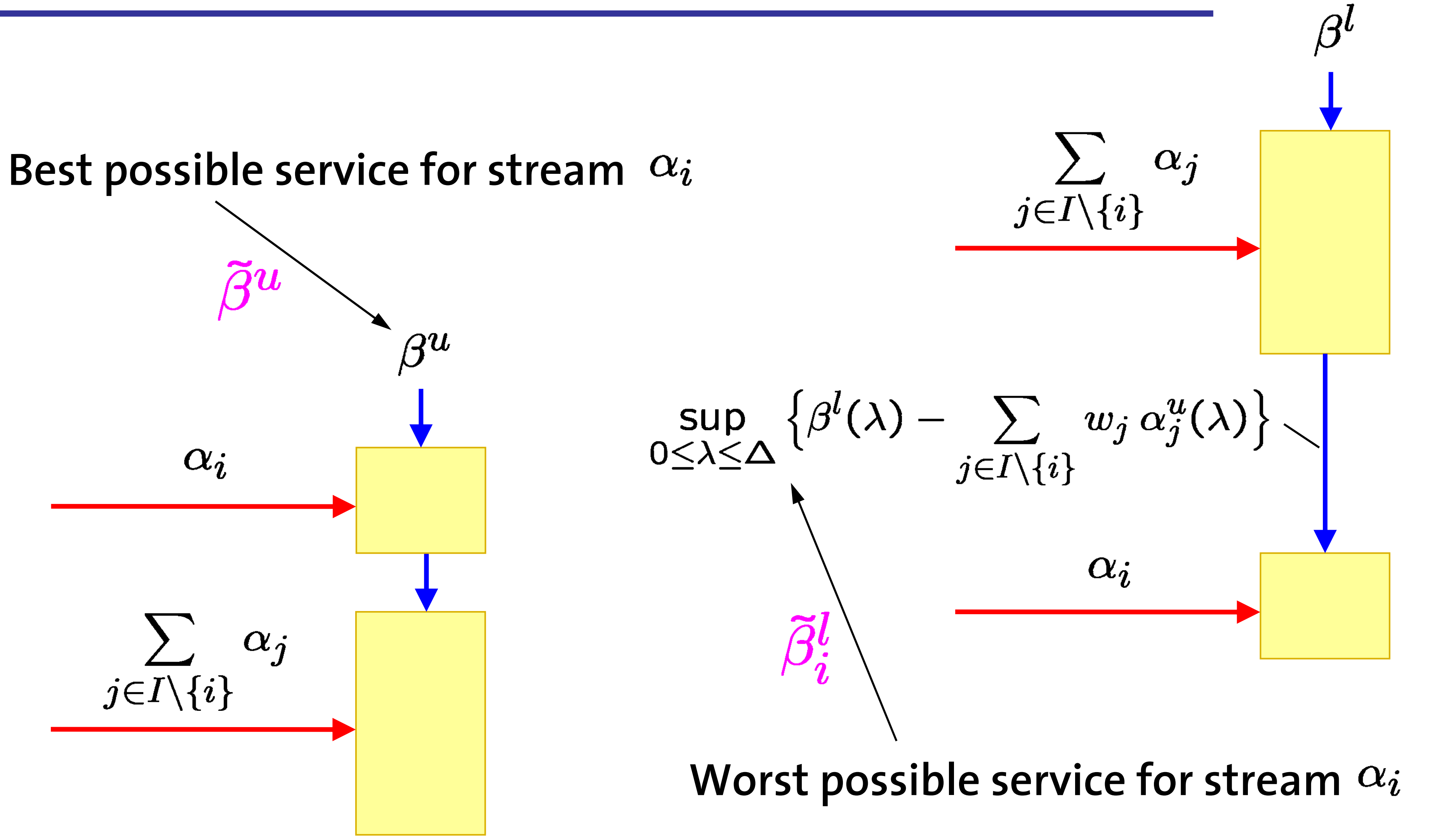


$$\hat{\alpha}'^{\,u}_i(\Delta) = \alpha^u_i\big(\Delta + (d_i - b_i)\big)$$



$$\hat{\alpha}'^{\,l}_i(\Delta) = \alpha^l_i\big(\Delta - (d_i - b_i)\big)$$

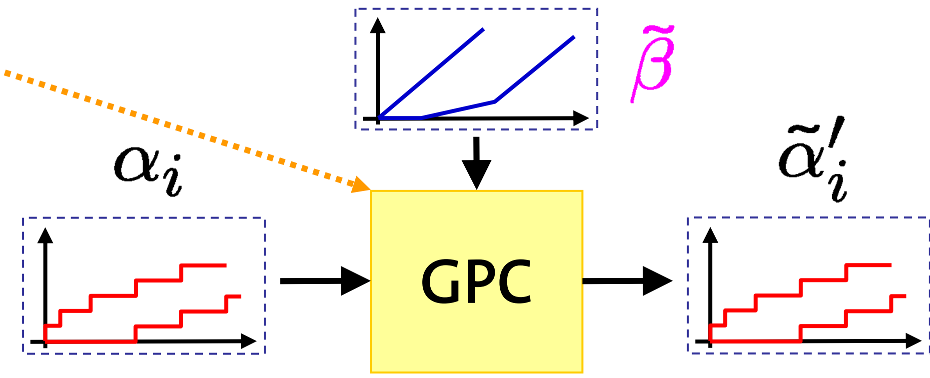
Output event streams – Improved bounds

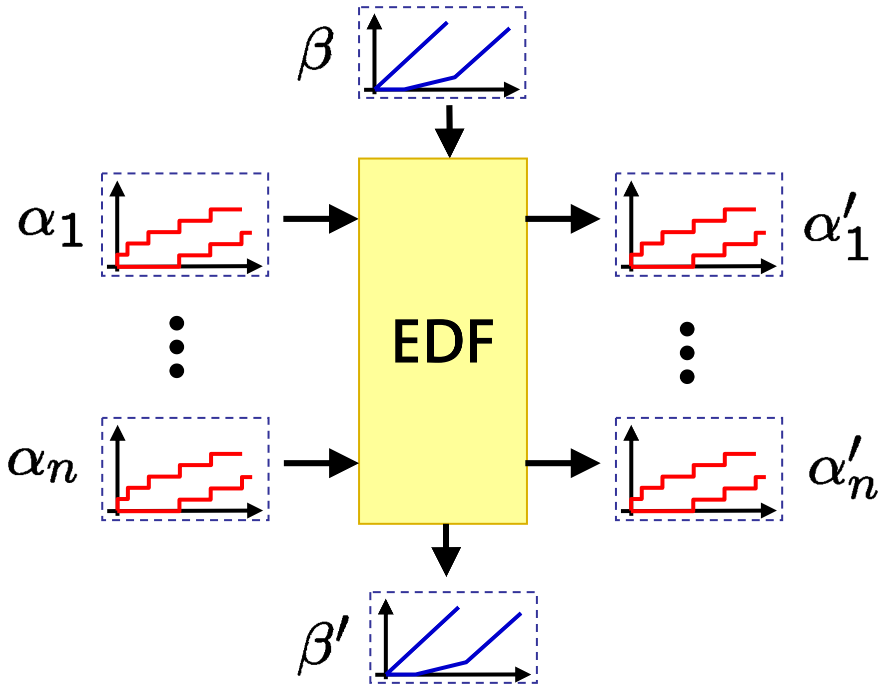


Output event streams – Improved bounds

$$\tilde{\alpha}_i'^u = \min\left\{(\alpha_i^u \otimes \tilde{\beta}^u) \oslash \tilde{\beta}_i^l,\ \tilde{\beta}^u\right\}$$

$$\tilde{\alpha}_i'^l = \min\left\{(\alpha_i^l \oslash \tilde{\beta}^u) \otimes \tilde{\beta}_i^l,\ \tilde{\beta}_i^l\right\}$$





3

$$\alpha_i'^u = \min\left\{\hat{\alpha}_i'^u,\ \left\lceil \frac{\tilde{\alpha}_i'^u}{b_i} \right\rceil \right\}$$

$$\alpha_i'^l = \max\left\{\hat{\alpha}_i'^u,\ \left\lfloor \frac{\tilde{\alpha}_i'^l}{w_i} \right\rfloor \right\}$$

Thank you!

Simon Perathoner

perathoner@tik.ee.ethz.ch