Formal Modeling and Analysis of Stream Processing Systems

(cont.)

Linh T.X. Phan

March 2009

Penn University of Pennsylvania
Computer and Information Science
University of Pennsylvania

Previous Lecture…

- General concepts of the performance analysis and design of stream processing systems
- Simulation vs formal analysis
- Existing formal analysis methods: pros and cons
- **Real-Time Calculus (RTC)**
  - High-level overview
  - Count-based abstraction
  - Definition of arrival and service functions

Real-Time Calculus (cont.)

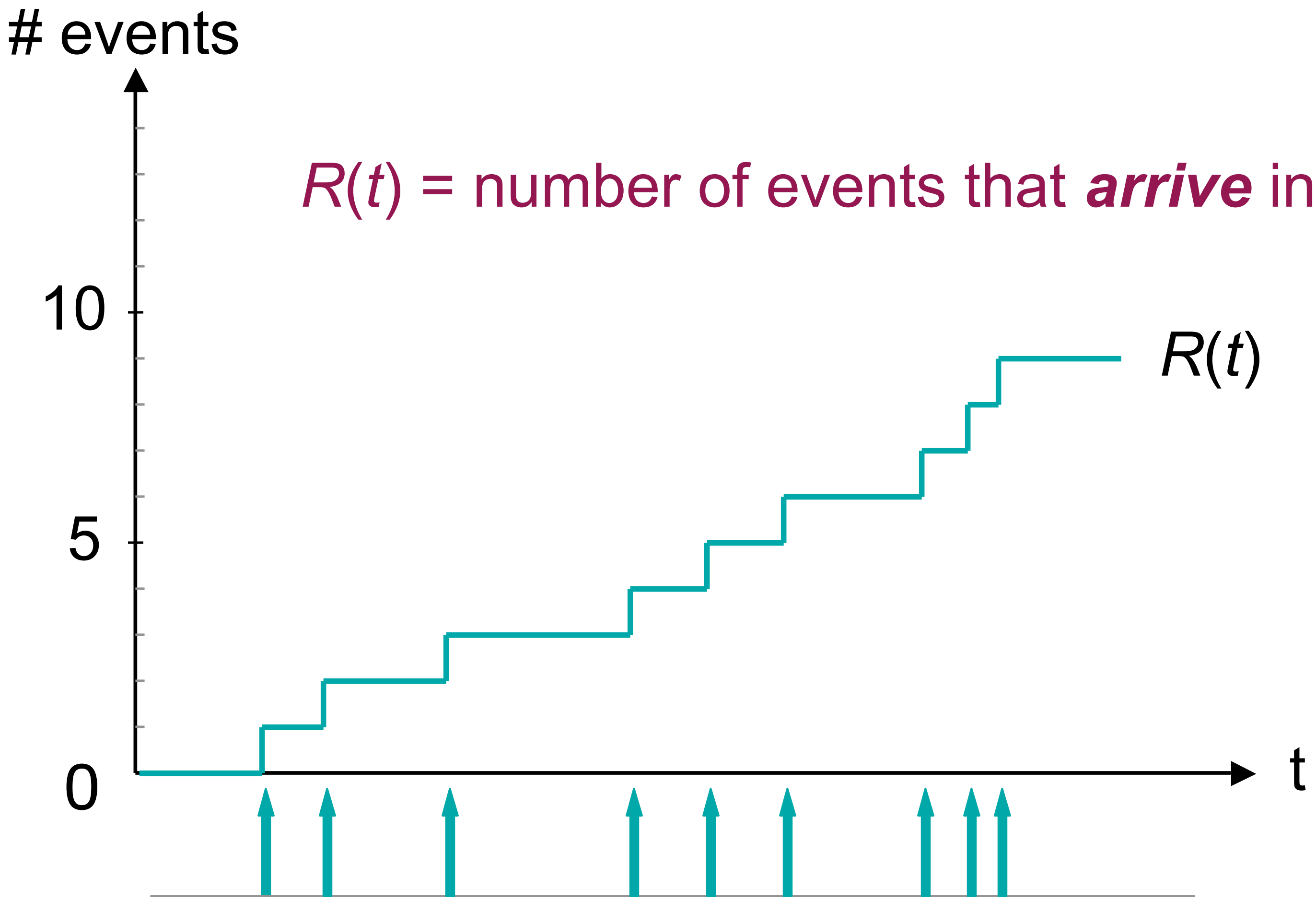
- A brief introduction to RTC
  - Refer to reading list for more!

- Materials are based on
  - Le Boudec and Thiran's book on Network Calculus
  - The MPA framework

Recall… Event Streams

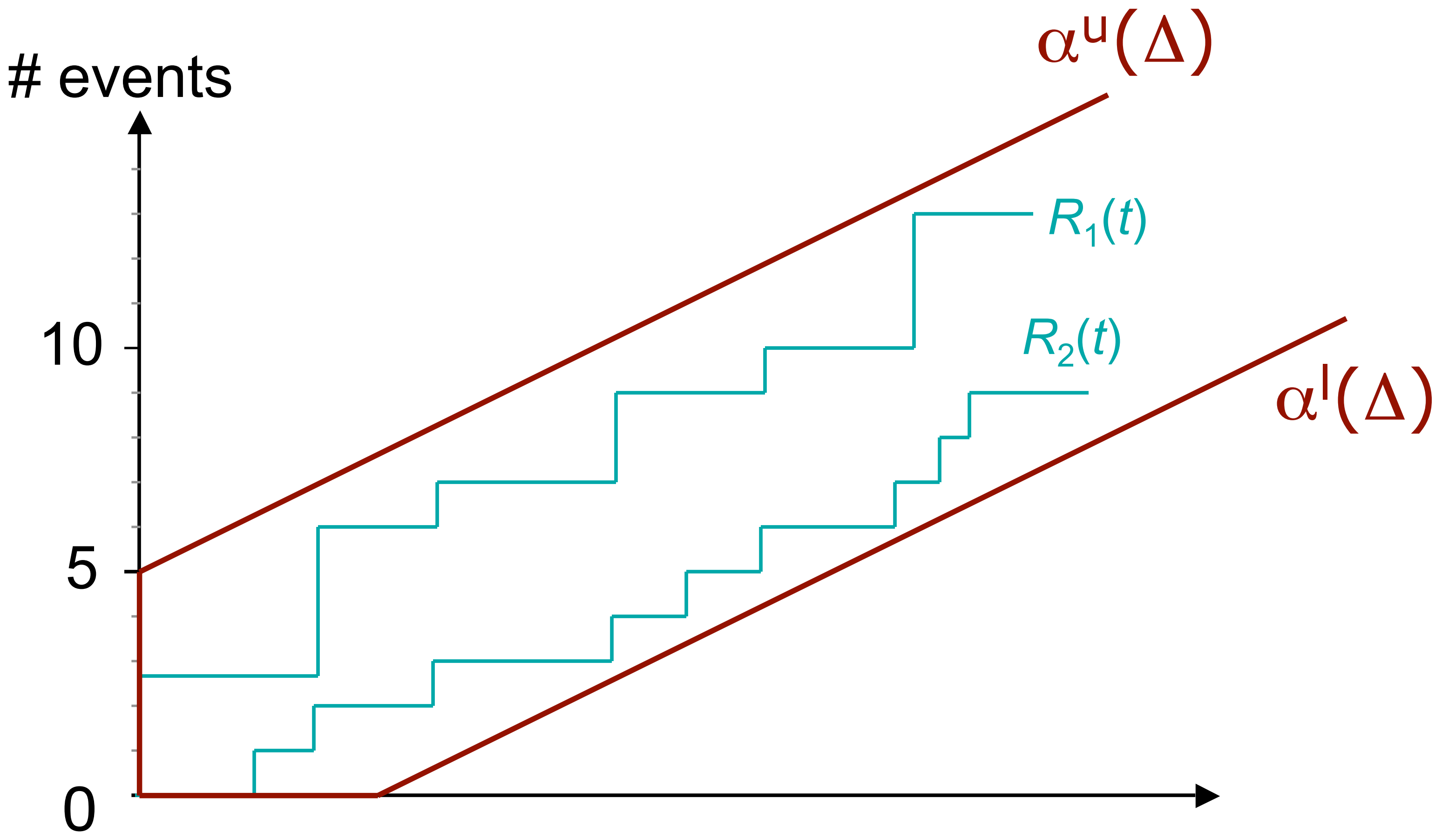
- Infinite sequences of data items (events)
- A concrete arrival pattern can be described as a cumulative function $R(t)$
  - $R(t)$ = #items arrive in the time interval $[0,t)$
- All possible arrival patterns of an event stream is abstracted as an arrival function $\alpha(\Delta)$
  - $[\alpha^l(\Delta), \alpha^u(\Delta)]$: the min. and max. number of events that arrive in *any* time interval of length Δ

An Arrival Pattern



#events that arrive in [t, t+Δ) is: $R(t+\Delta) - R(t)$

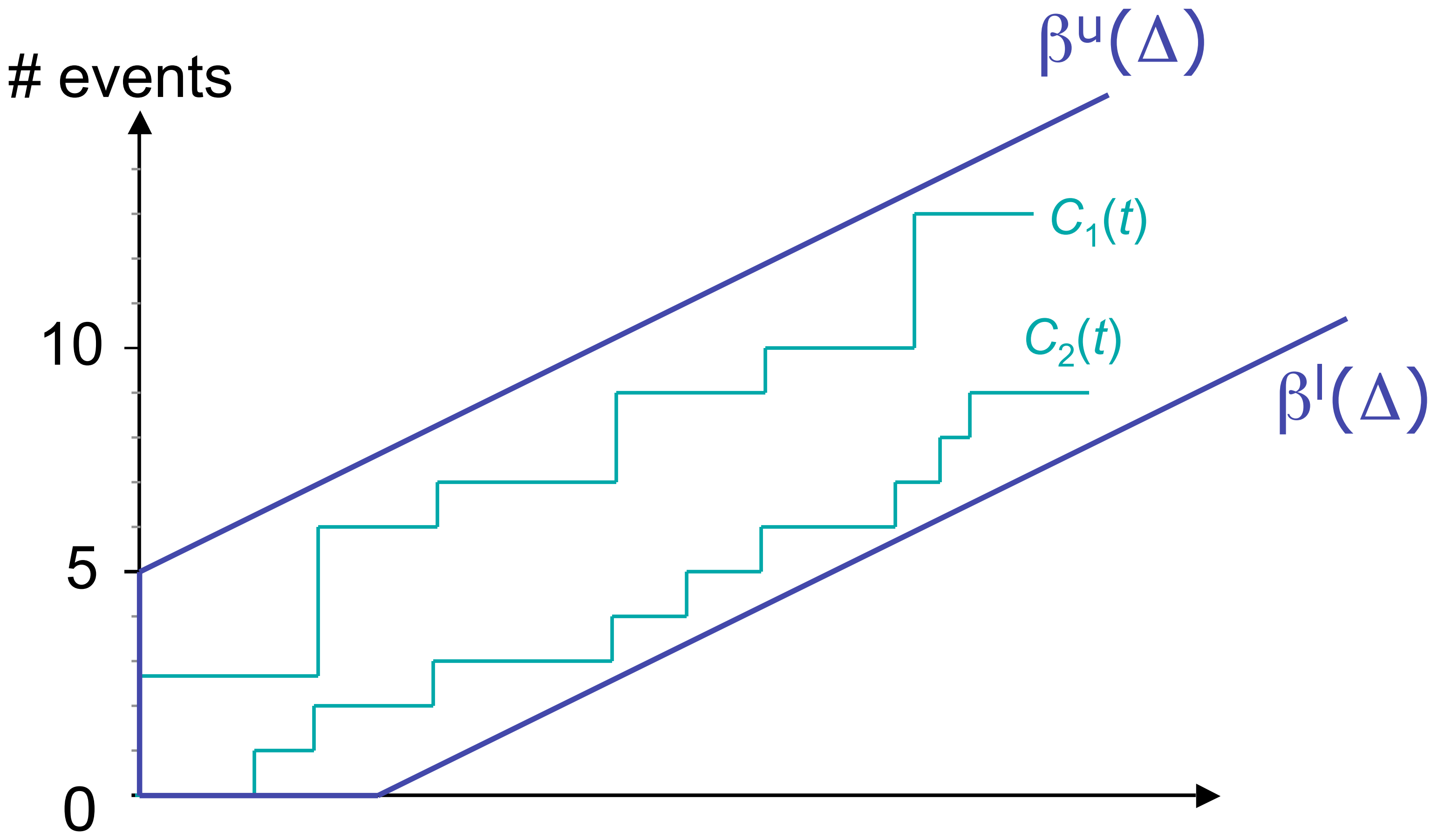
Arrival Function of A Set of Concrete Patterns



Recall… Resources

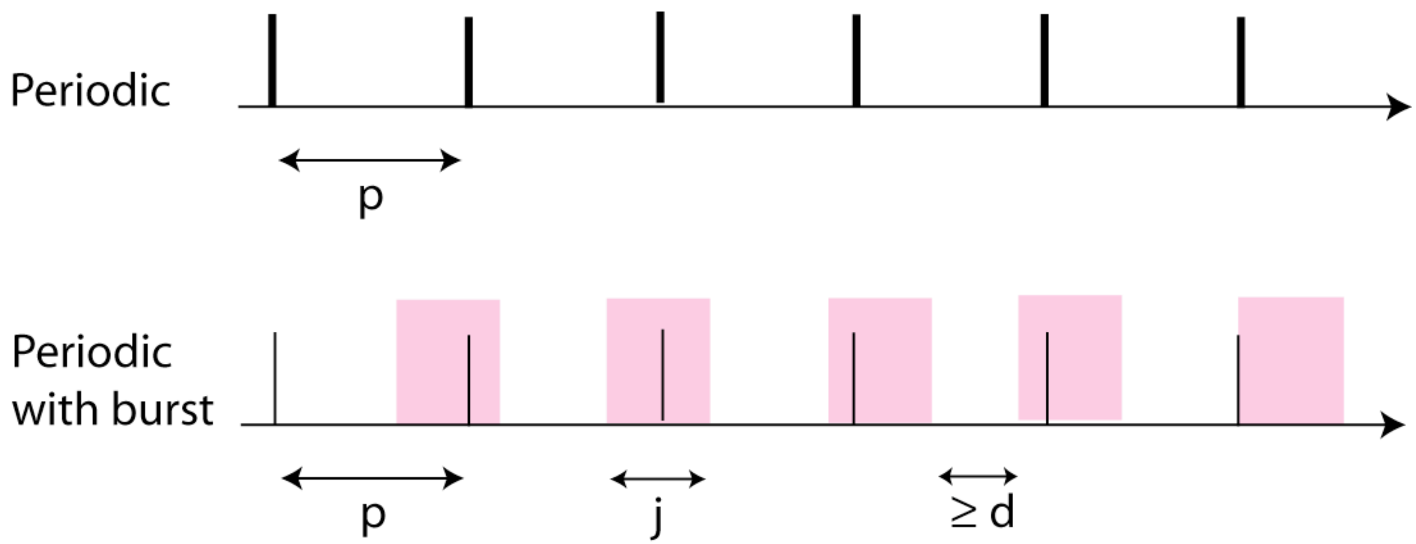
- A concrete service pattern
  - how much and when the resource is available
  - captured as a cumulative function $C(t)$ which gives the amount of resource units available in time interval $[0,t)$
- All possible service patterns of a resource is abstracted as a service function $\beta(\Delta)$
  - $[\beta^l(\Delta), \beta^u(\Delta)]$: the min. and max. number of resource units available (or the number of events that can be processed) in *any* time interval of length Δ

Service Function of A Set of Concrete Patterns



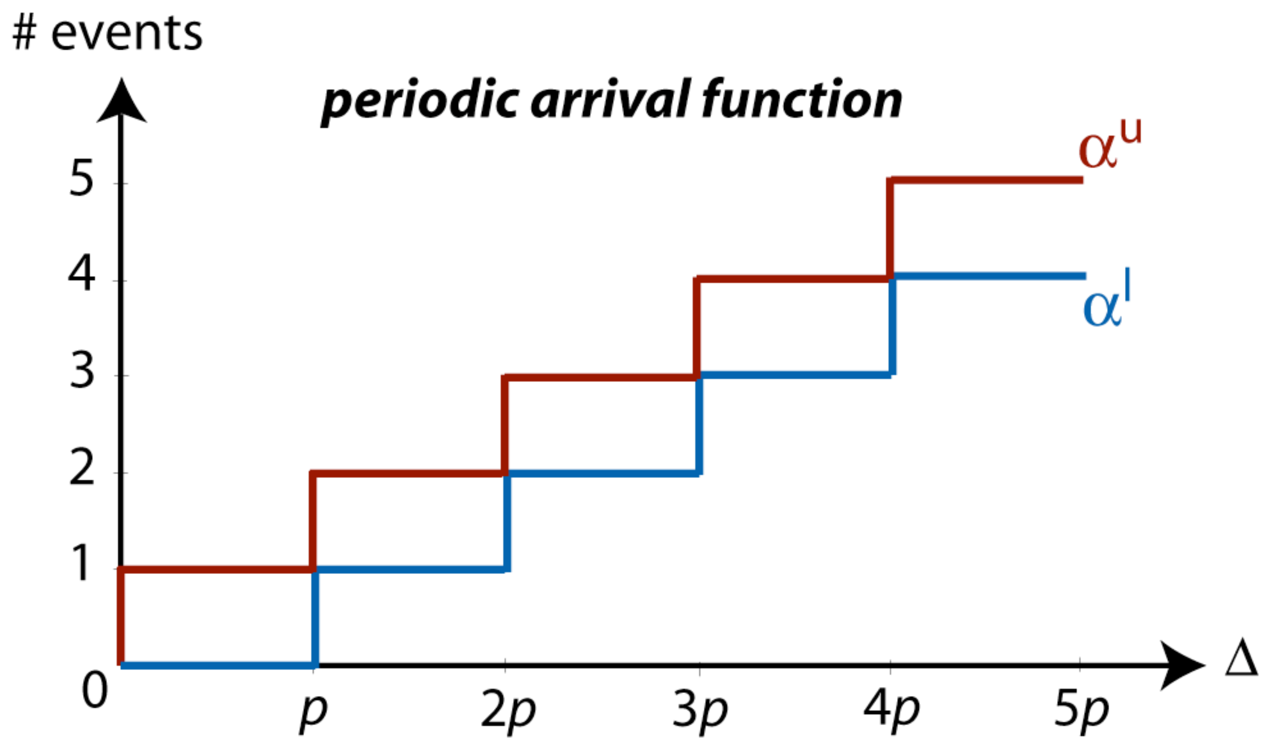
Examples of Arrival and Service Functions

Standard Event Streams

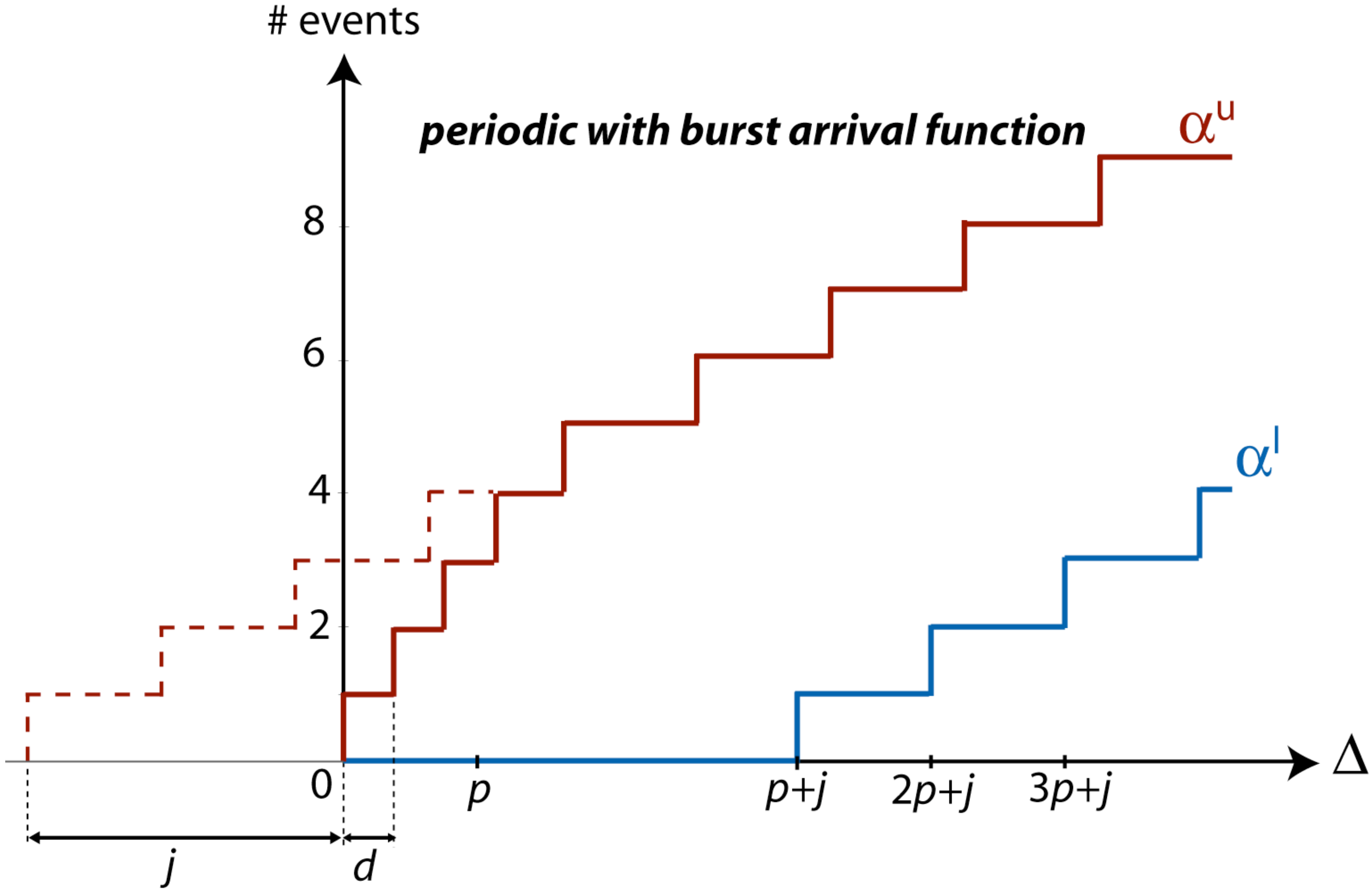


p: period *j: jitter* *d: minimum inter-arrival distance*

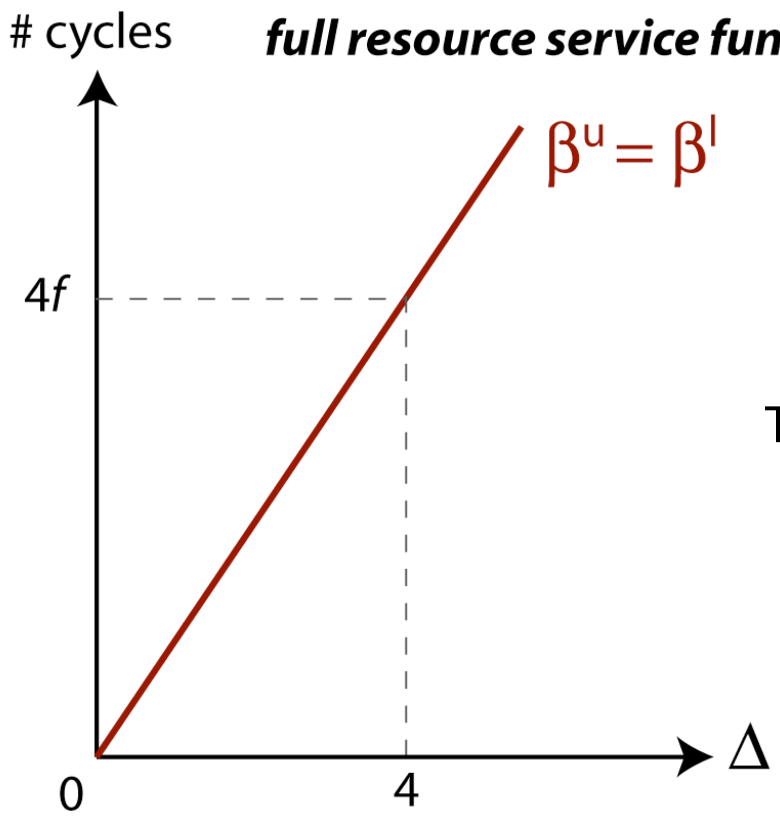
$$\alpha^l(\Delta) = \left\lfloor \frac{\Delta}{p} \right\rfloor \qquad \alpha^u(\Delta) = \left\lceil \frac{\Delta}{p} \right\rceil$$



$$\alpha^l(\Delta) = \left\lfloor \frac{\Delta - j}{p} \right\rfloor \qquad \alpha^u(\Delta) = \min\left\{ \left\lceil \frac{\Delta + j}{p} \right\rceil, \left\lceil \frac{\Delta}{d} \right\rceil \right\}$$



Common Resources

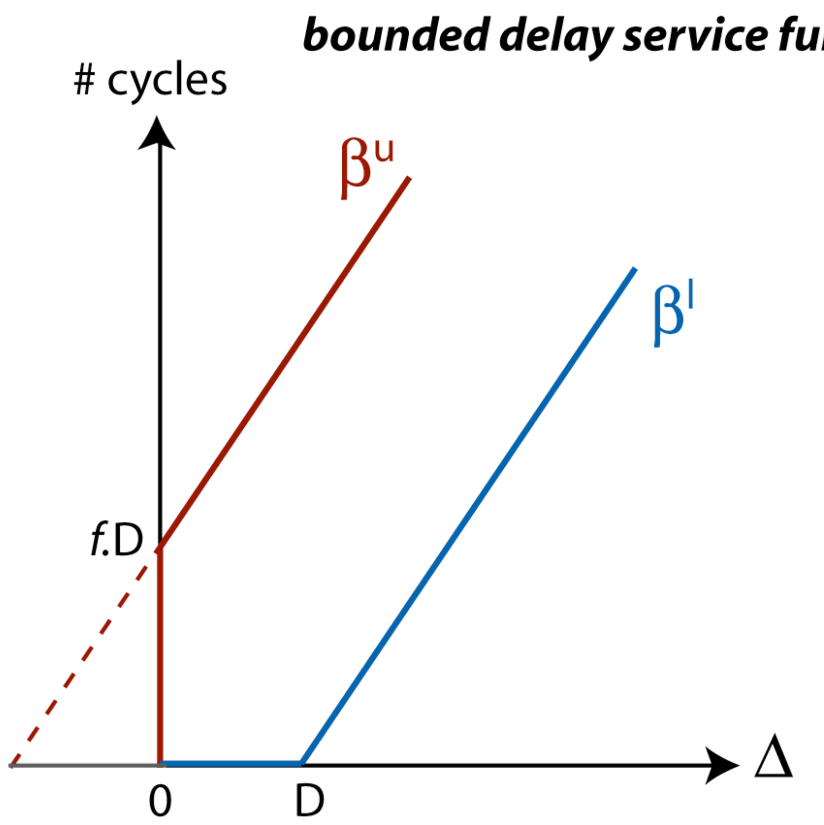


The processor is always available

f = processor frequency

Bounded Delay

$$\forall t, \forall t' \geq t: \ (t' - t - D)f \leq C(t') - C(t) \leq (t' - t + D)f$$



f = frequency

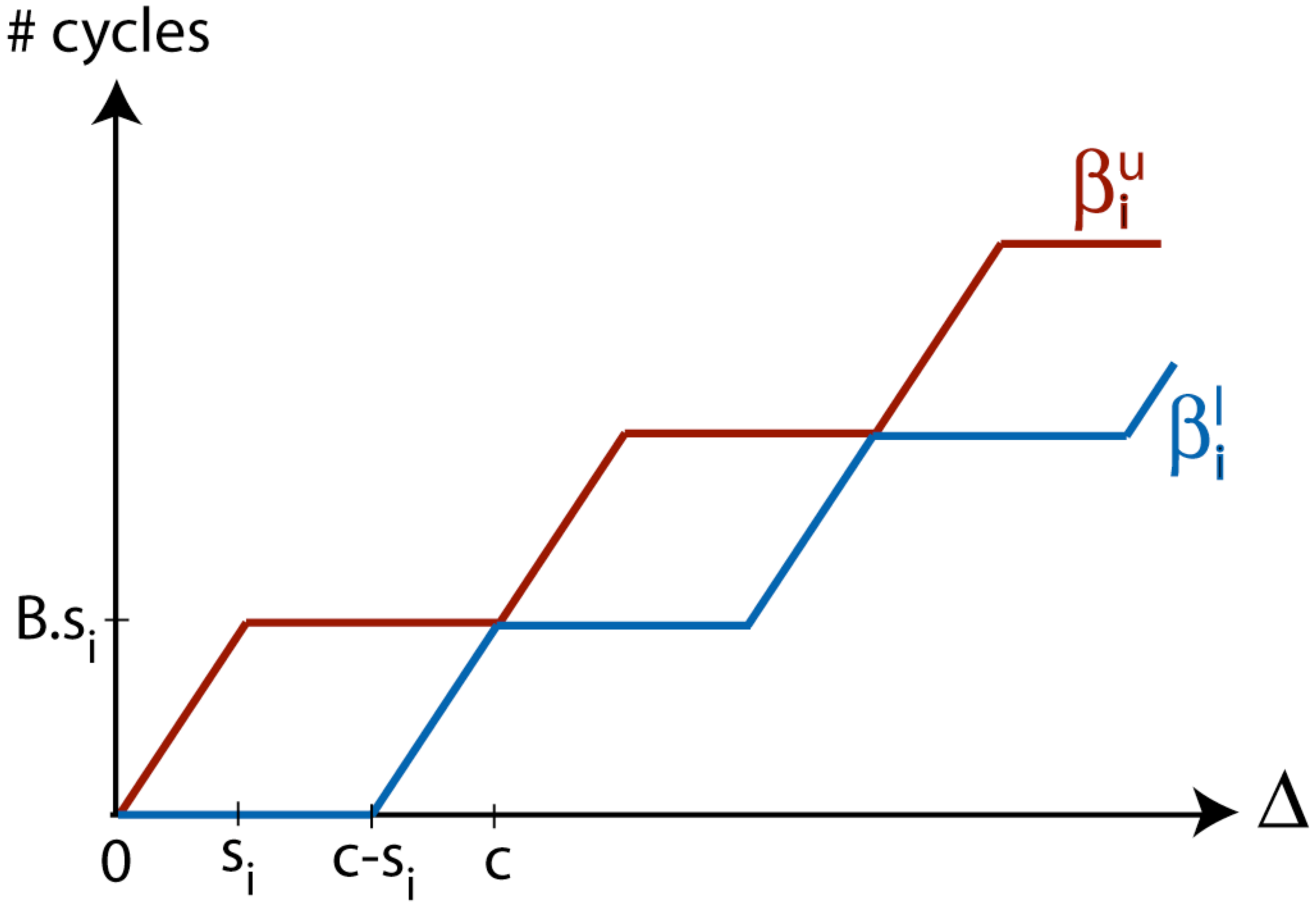
D = bounded delay

TDMA Resource

- A shared resource of bandwidth *B*
- *n* applications: $App_1, \ldots, App_n$
- TDMA policy
  - a resource slot of length s_i is assigned to App_i in every cycle of length *c*
  - ➢ the resource given to App_i is bounded by

$$\beta_i^l(\Delta) = B \max\left\{ \left\lfloor \frac{\Delta}{c} \right\rfloor s_i,\ \Delta - \left\lceil \frac{\Delta}{c} \right\rceil (c - s_i) \right\}$$
$$\beta_i^u(\Delta) = B \min\left\{ \left\lceil \frac{\Delta}{c} \right\rceil s_i,\ \Delta - \left\lfloor \frac{\Delta}{c} \right\rfloor (c - s_i) \right\}$$

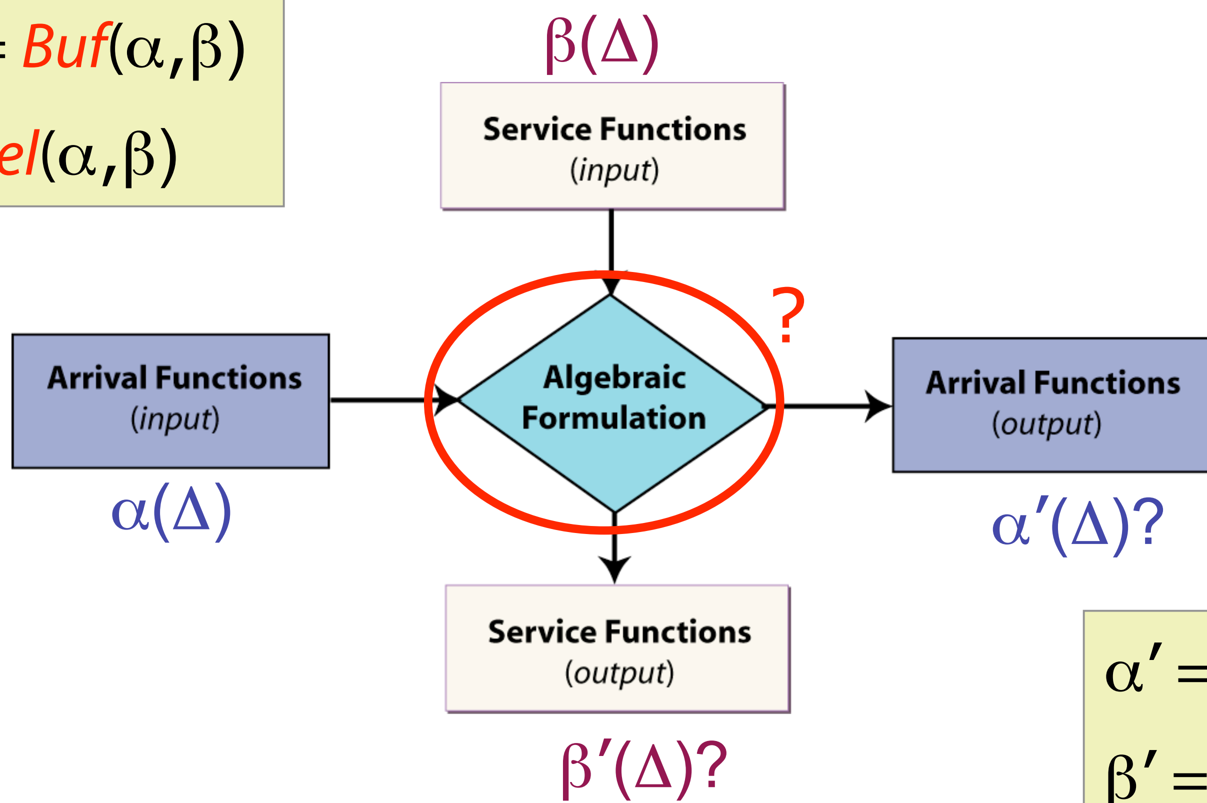
TDMA Resource



RTC Performance Model

backlog = $Buf(\alpha,\beta)$

delay = $Del(\alpha,\beta)$



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

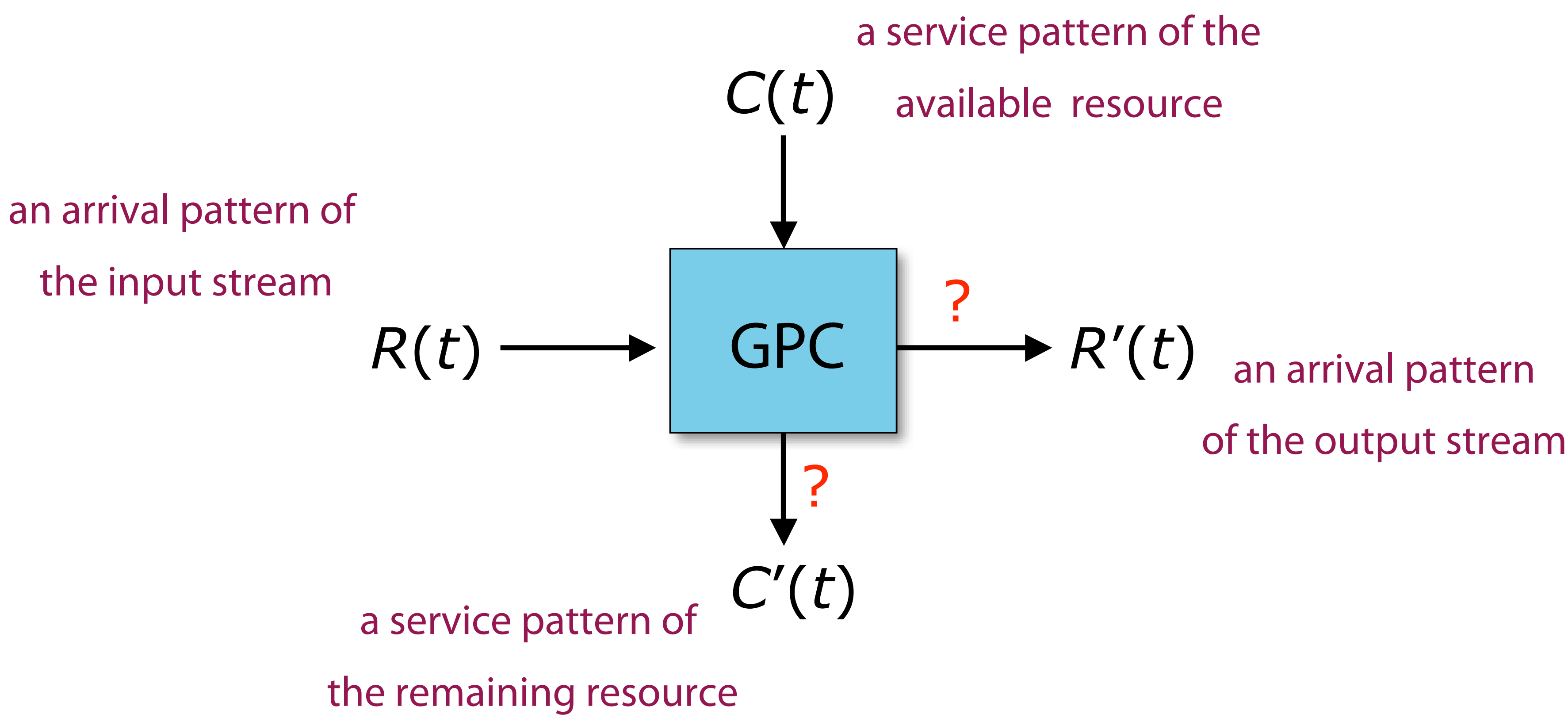
$\beta' = f_\beta(\alpha,\beta)$

The functions f_α, f_β, *Buf*, *Del* must take into account the *scheduling policy* and the *processing semantics* of the component

Processing Model: Abstract Component

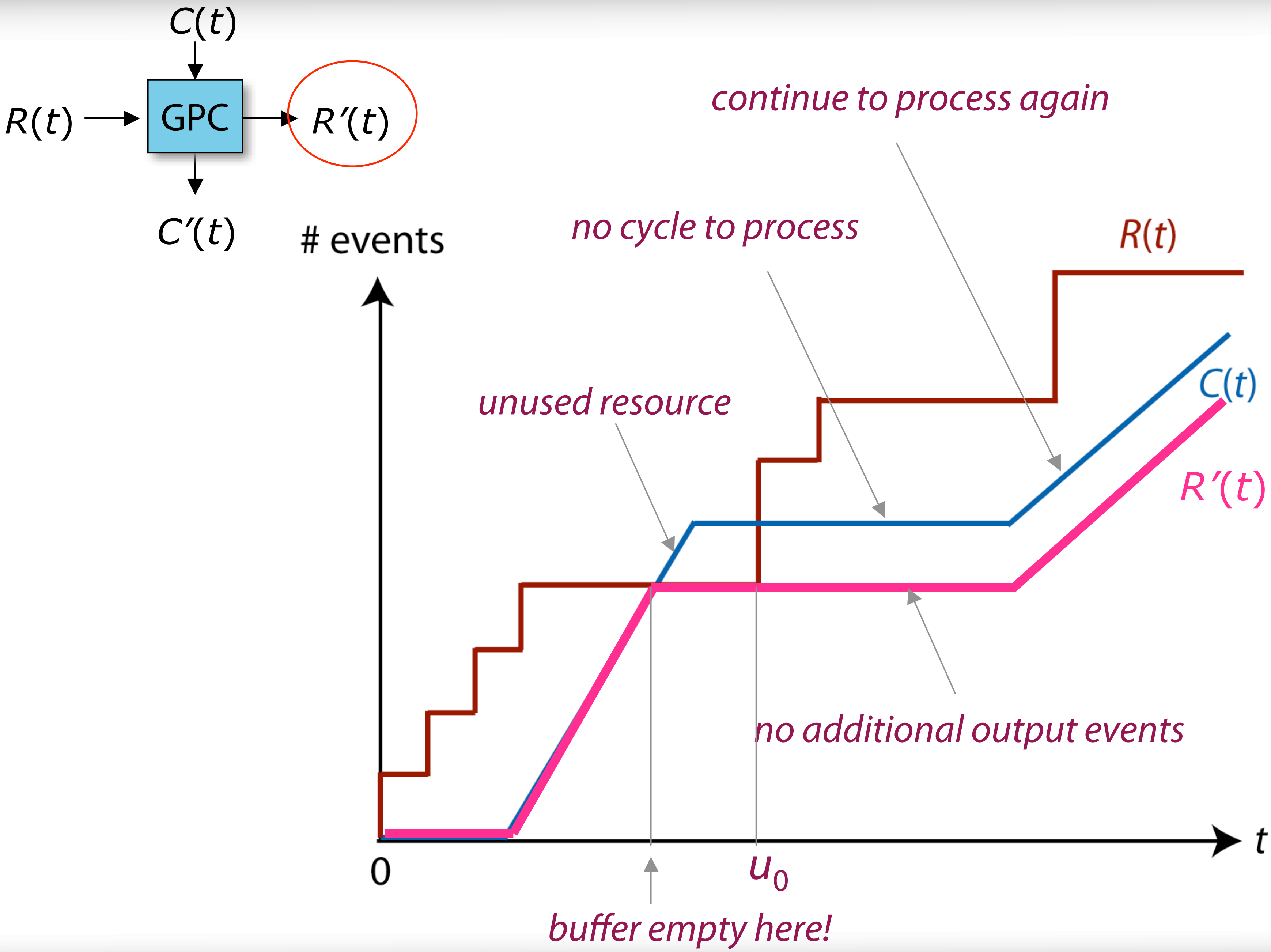
- Relate input arrival/service functions and
  - output arrival and service functions
  - maximum backlog
  - maximum delay
- The computation must capture the way input event streams are processed by the resource
- Vary depending on the scheduling policy and processing semantics, but always deterministic
- Based on min-plus and max-plus algebra

A concrete system component



Greedy Processing Component

- Triggered by incoming events
  - a preemptive task is instantiated to process each arrival event
- Events are processed in a greedy fashion and FIFO order
  - subjected to resource availability
  - waiting events are stored in the input buffer
- Backlog at time *t*
  - $B(t)$ = #events in the buffer at time *t*
- Delay at time t
  - d(t) = the maximum processing time (including waiting time) of an event arriving before t



C(t)
R(t)
GPC
R′(t)
C′(t)
events
continue to process again
no cycle to process
R(t)
C(t)
unused resource
R′(t)
no additional output events
0
t
u0
buffer empty here!

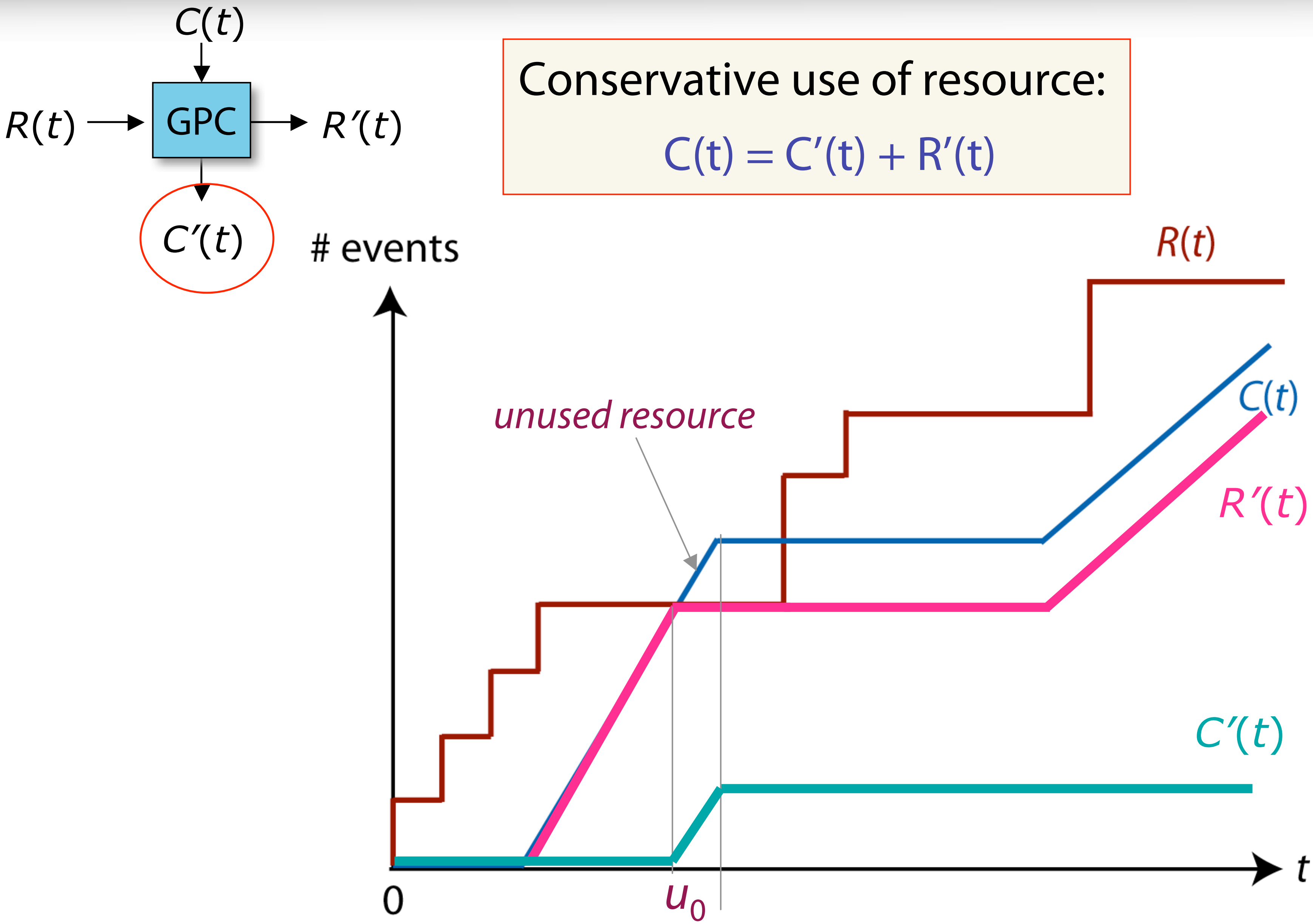
GPC: Output Stream

$$R'(t) = \inf_{0 \le u \le t} \{R(u) + C(t) - C(u)\}$$

For all u≤t:

- $R'(u) \le R(u)$ and $R'(t) \le R'(u) + C(t) - C(u)$
  - #output-events in [0,u) is no more than #input-events in [0,u)
  - #output-events in [u,t) is no more than #events that can be processed in [u,t)

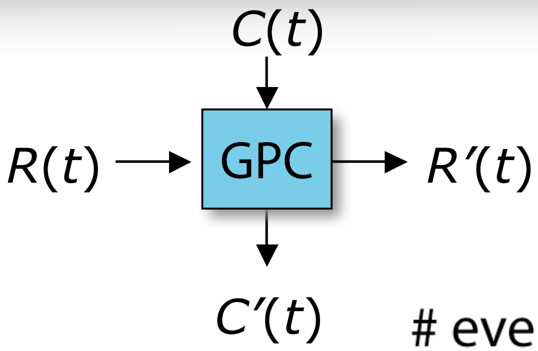
  Hence, $R'(t) \le R(u) + C(t) - C(u)$
- Let u_0 be the last instant before t at which $B(u_0) = 0$
  - $R'(u_0) = R(u_0)$; $R'(t) = R(u_0) + C(t) - C(u_0)$
  - Thus, $R'(\mathrm{t}) = R'(u_0) + C(t) - C(u_0)$



Conservative use of resource:

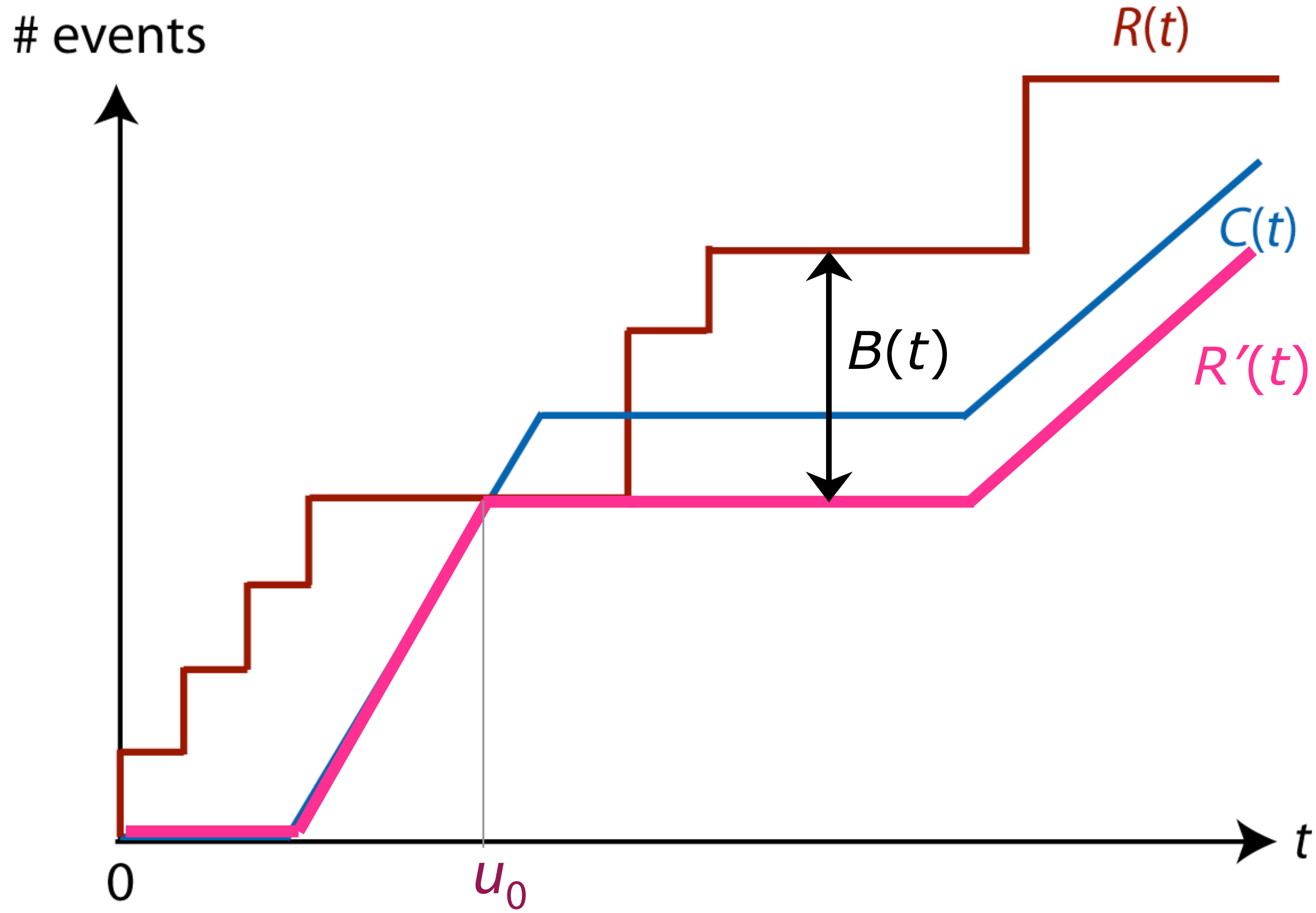
$$C(t) = C'(t) + R'(t)$$

GPC: Remaining Resource

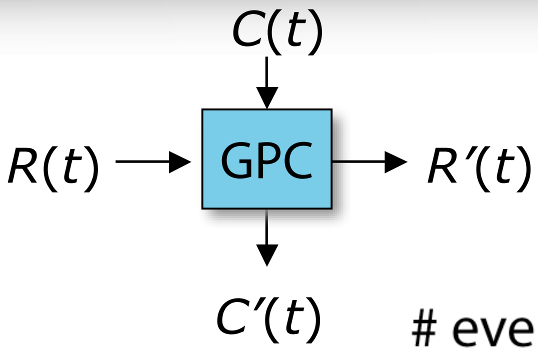


Backlog at time t: $B(t) = R(t) - R'(t)$

Maximum backlog: $B_{max} = \max_{t \geq 0} B(t)$

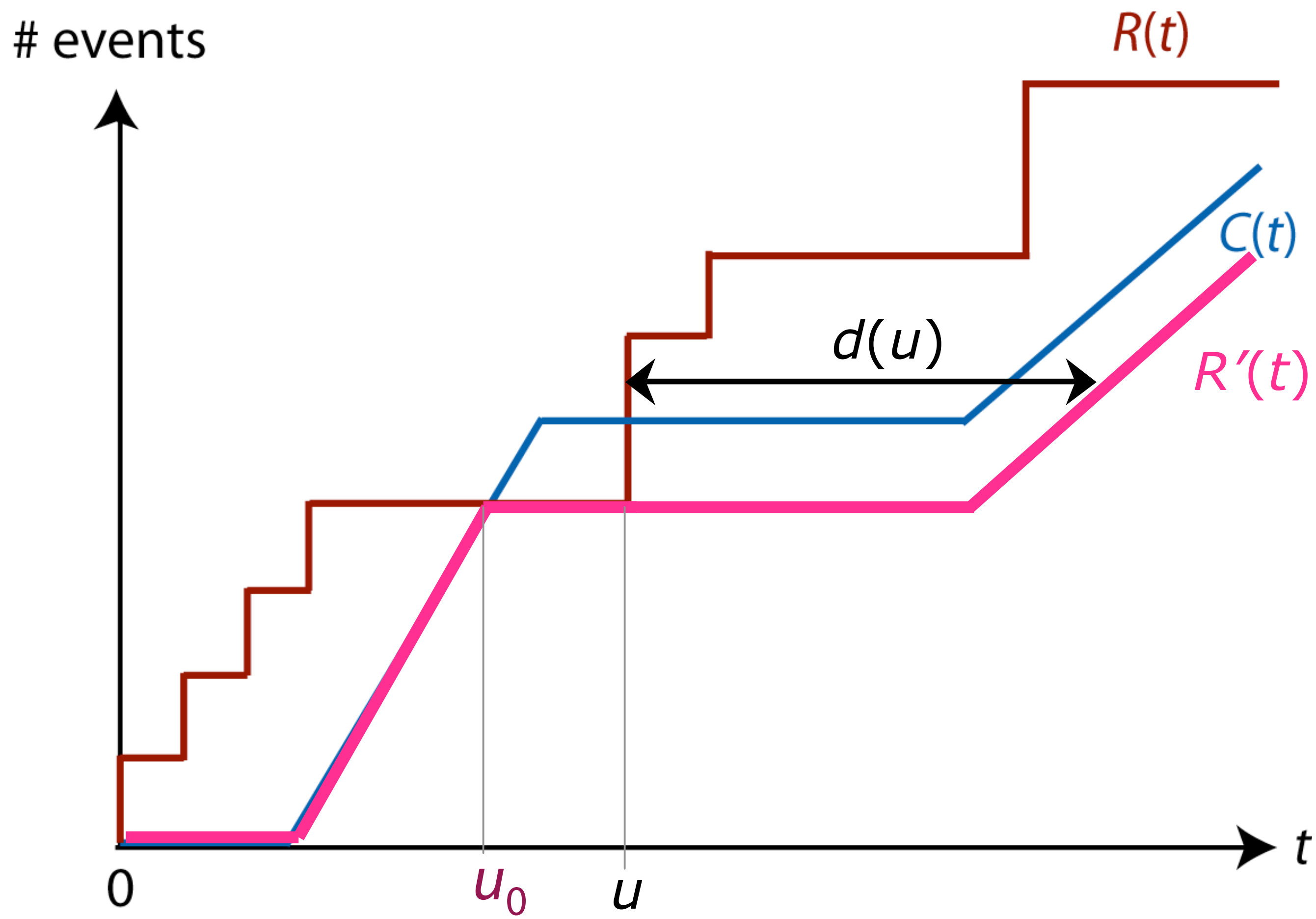


GPC: Backlog



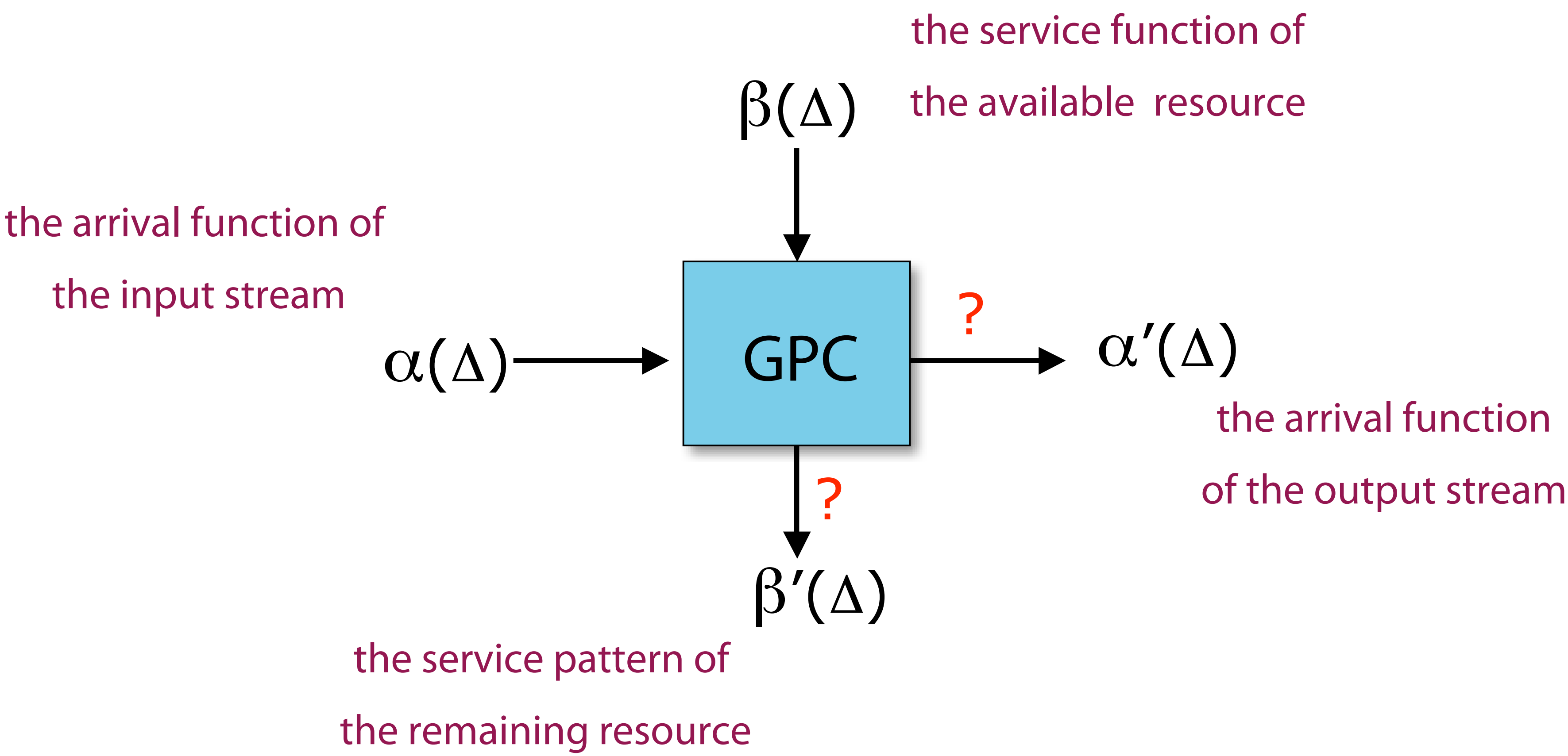
Delay at time t :

$$d(t) = \min\{ \lambda : R'(t+\lambda) \geq R(t)\}$$



GPC: Delay

An abstract system component



Basic Min-plus/Max-plus Operators

- Min-plus convolution and de-convolution

$$(f \otimes g)(t) = \inf_{0 \le u \le t} \{ f(t-u) + g(u) \}$$

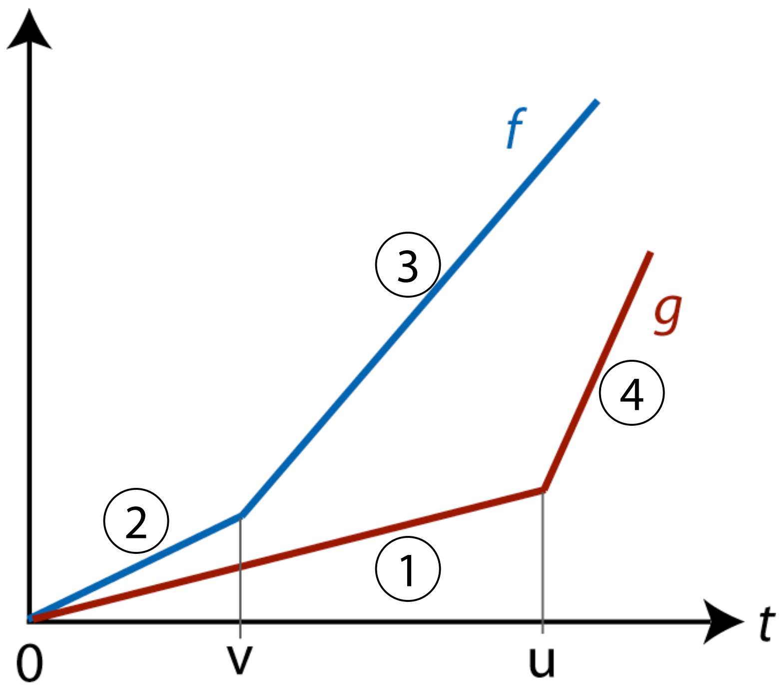
$$(f \oslash g)(t) = \sup_{u \ge 0} \{ f(t+u) - g(u) \}$$

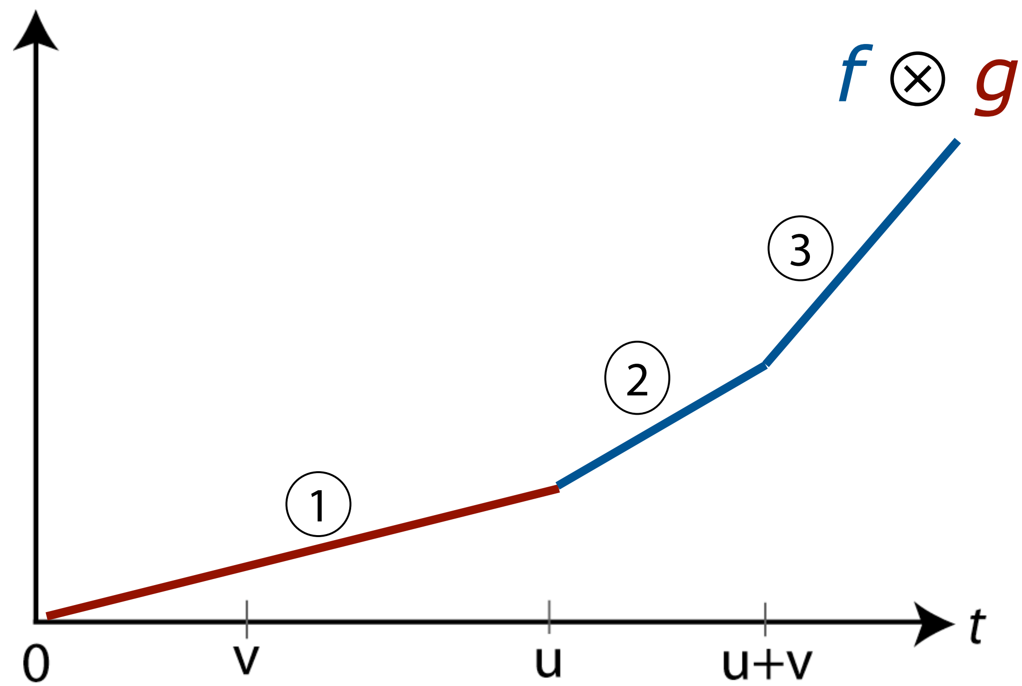
- Max-plus convolution and de-convolution

$$(f \overline{\otimes} g)(t) = \sup_{0 \le u \le t} \{ f(t-u) + g(u) \}$$

$$(f \overline{\oslash} g)(t) = \inf_{u \ge 0} \{ f(t+u) - g(u) \}$$

e.g. Min-Conv of Convex Piecewise Linear Curves





- Label segments in increasing slope order
- Connect segments end to end with increasing slope

Basic Min-plus/Max-plus Operators

- Recall for all t, Δ ≥ 0

$$\alpha^l(\Delta) \leq R(t+\Delta) - R(t) \leq \alpha^u(\Delta)$$

$$\beta^l(\Delta) \leq C(t+\Delta) - C(t) \leq \beta^u(\Delta)$$

- Valid arrival and service functions for a given *R*(*t*) and *C*(*t*)

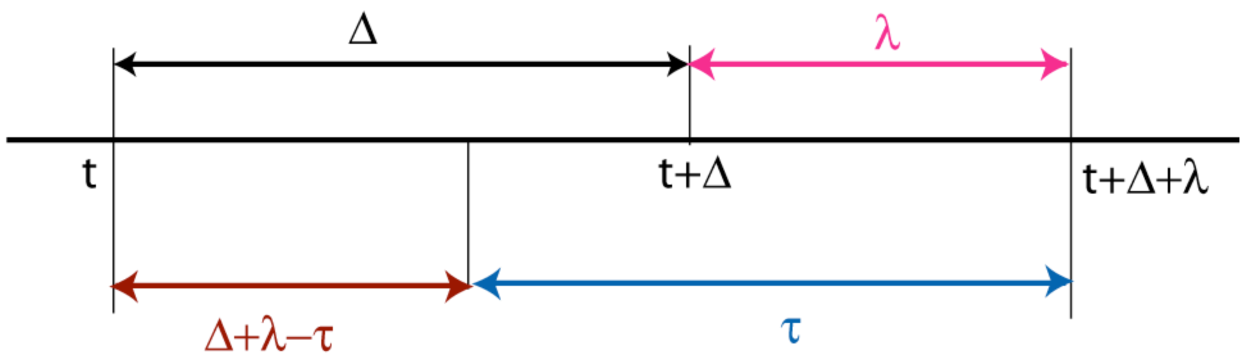
$$\alpha^l = R \,\overline{\oslash}\, R \qquad \alpha^u = R \oslash R$$

$$\beta^l = C \,\overline{\oslash}\, C \qquad \beta^u = C \oslash C$$

GPC: Output Bounds

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

Compute α^u - Intuitive Idea



- $max\_output(\Delta+\lambda) \le max\_input(\Delta+\lambda-\tau) + max\_processed(\tau), \forall\, 0 \le \tau \le \Delta+\lambda$
  - $\le \sup_{0 \le \tau \le \Delta+\lambda} \{\alpha^u(\Delta+\lambda-\tau) + \beta^u(\tau)\} = \gamma(\Delta+\lambda)$, with $\gamma = \alpha^u \otimes \beta^u$
- $max\_output(\Delta) \le max\_output(\Delta+\lambda) - min\_processed(\lambda), \forall\, \lambda \ge 0$
  - $\le \gamma(\Delta+\lambda) - \beta^l(\lambda), \forall\, \lambda \ge 0$
  - $\le \inf_{\lambda \ge 0} \{\gamma(\Delta+\lambda) - \beta^l(\lambda)\} = (\gamma \oslash \beta^l)(\Delta)$
  - $\le [\alpha^u \otimes \beta^u) \oslash \beta^l](\Delta)$
- Further, $max\_output(\Delta) \le max\_processed(\Delta) \le \beta^u(\Delta)$
- $\Rightarrow \quad \alpha^{u'} \le \min\{\alpha^u \otimes \beta^u) \oslash \beta^l, \beta^u\}$

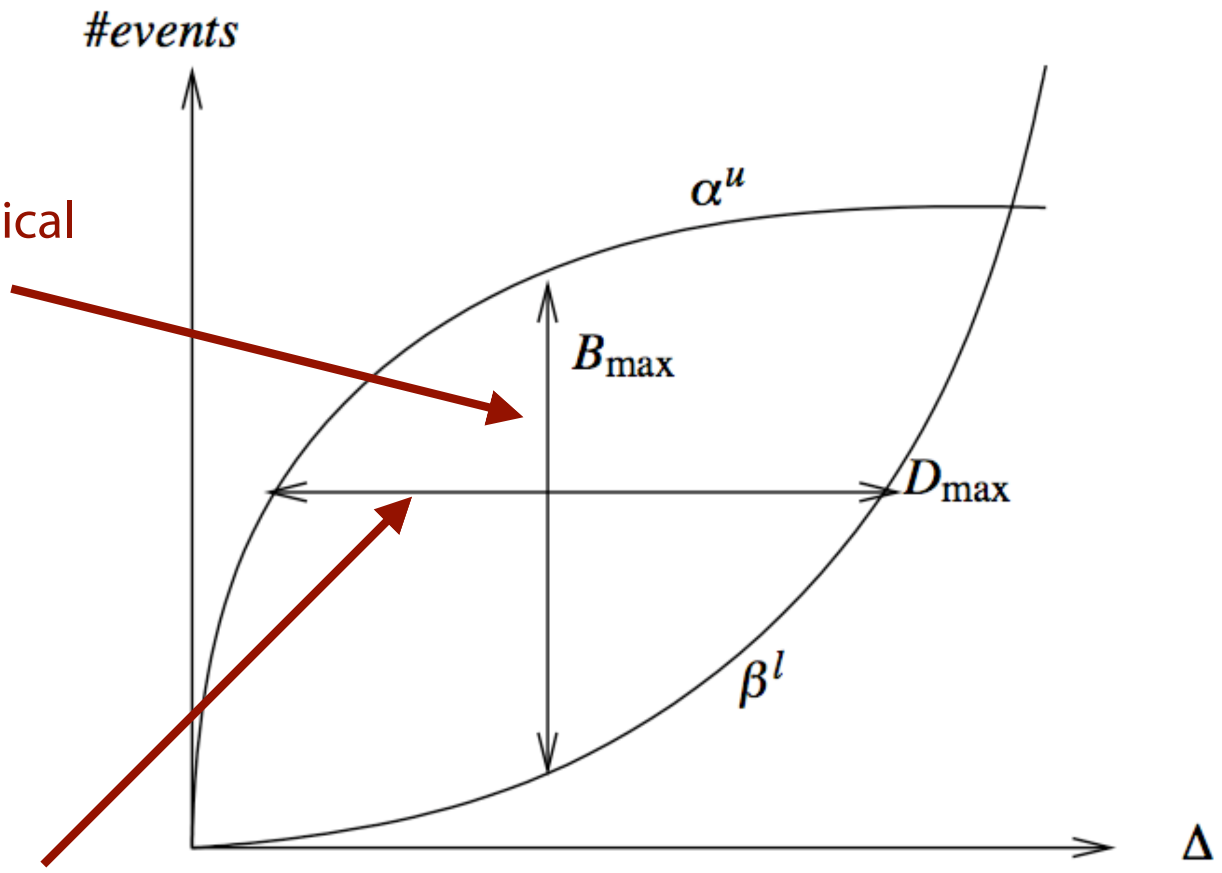
GPC: Backlog and Delay Bounds

$$B_{\max} = \sup_{t \geq 0}\{R(t) - R'(t)\}$$
$$\leq \sup_{\Delta \geq 0}\{\alpha^u(\Delta) - \beta^l(\Delta)\}$$

$$D_{\max} = \sup_{t \geq 0}\left\{\inf\{u \geq 0 : R(t) \leq R'(t+u)\}\right\}$$
$$= \sup_{\Delta \geq 0}\left\{\inf\{u \geq 0 : \alpha^u(\Delta) \leq \beta^l(\Delta + u)\}\right\}$$

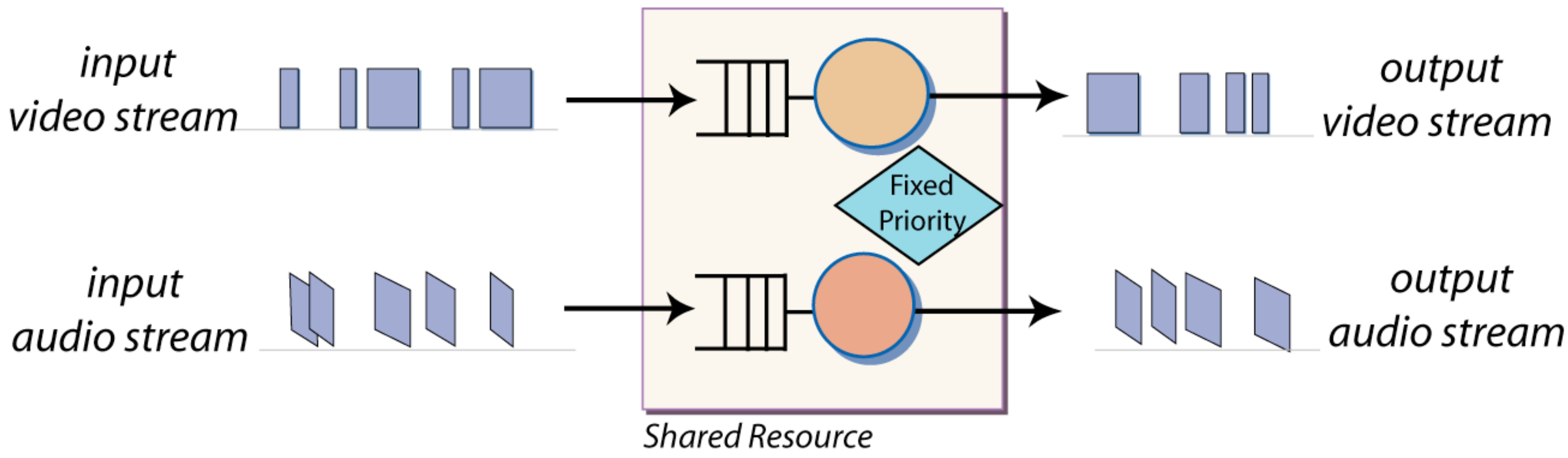
GPC: Backlog and Delay Bounds

Maximum vertical distance



Maximum horizontal distance

Scheduling Multiple Event Streams



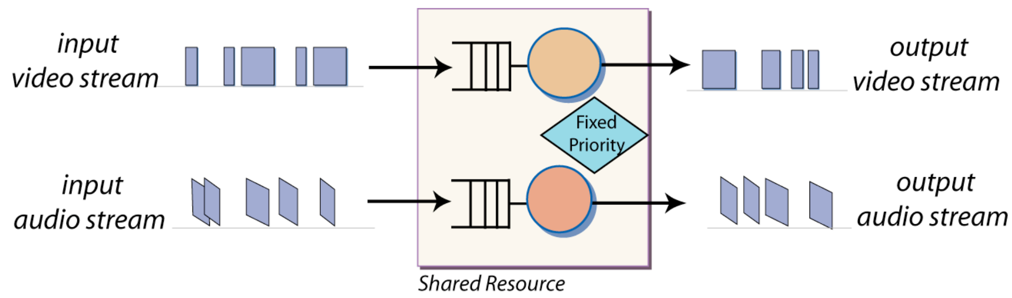
Fixed Priority:

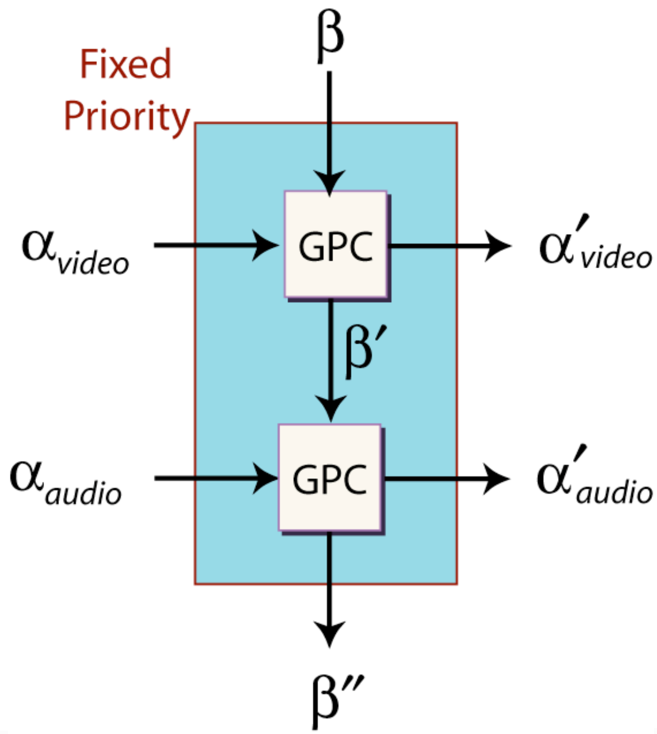
- video stream has higher priority than audio stream

  ➔ process the video stream first

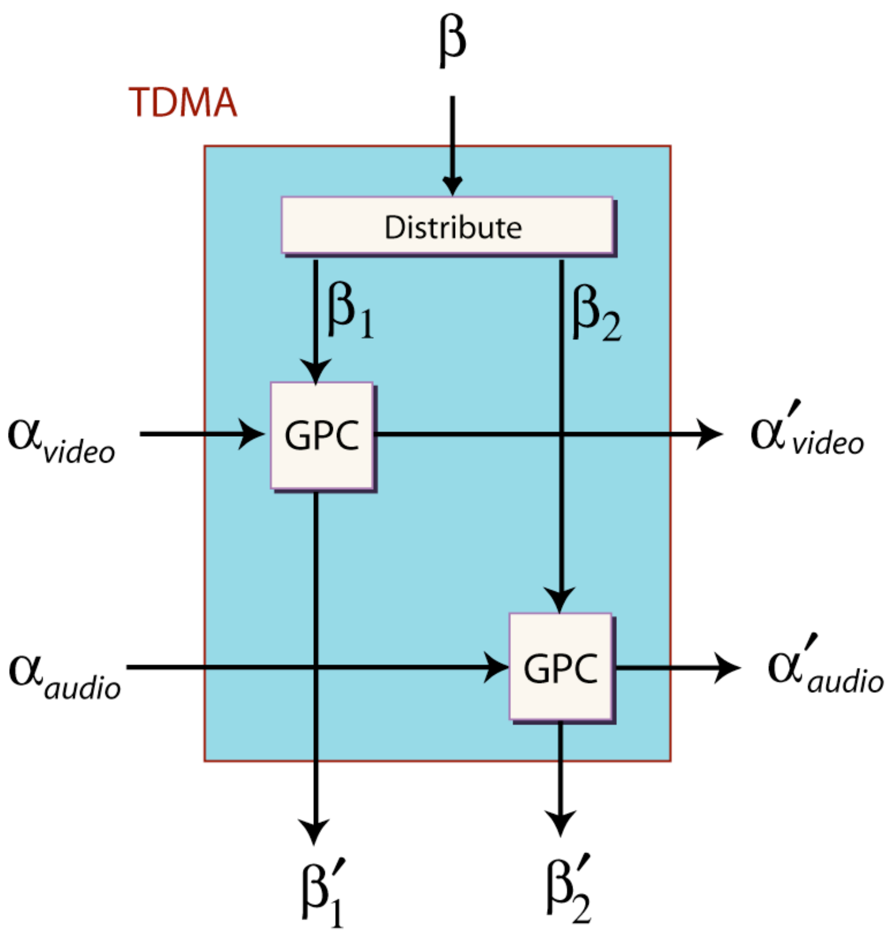
- remaining resource is used to process the audio stream

Fixed Priority Scheduling





TDMA Scheduling

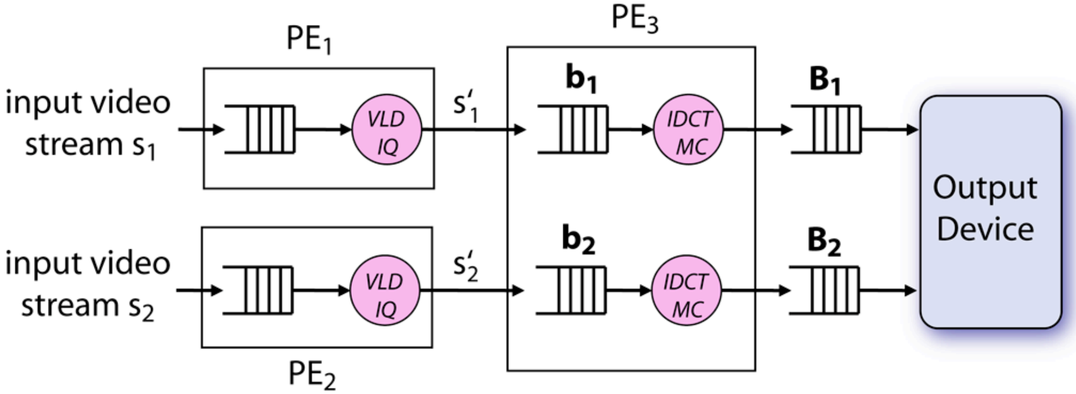


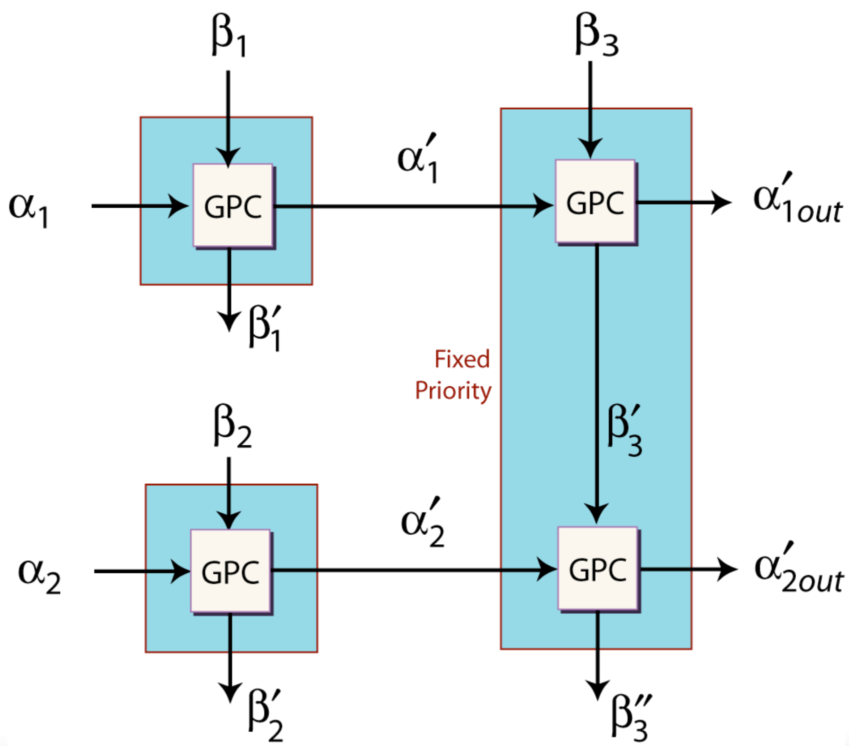
b_i : computed based on the length of the TDMA cycle c and the slot s_i

$$\beta_i^l(\Delta) = B\max\left\{\left\lfloor\frac{\Delta}{c}\right\rfloor s_i,\ \Delta-\left\lceil\frac{\Delta}{c}\right\rceil(c-s_i)\right\}$$

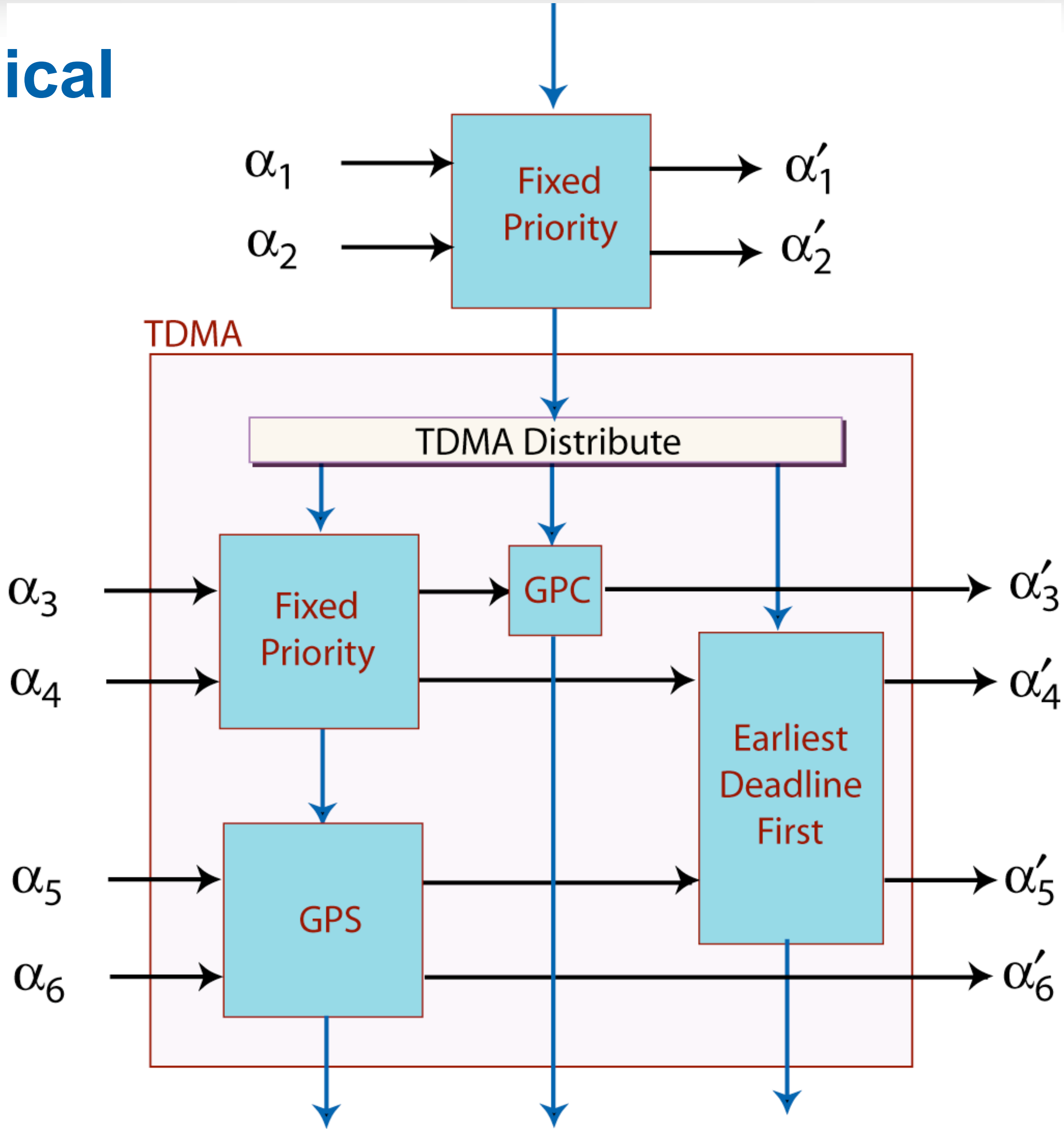
$$\beta_i^u(\Delta) = B\min\left\{\left\lceil\frac{\Delta}{c}\right\rceil s_i,\ \Delta-\left\lfloor\frac{\Delta}{c}\right\rfloor(c-s_i)\right\}$$

Modular Performance Analysis using RTC





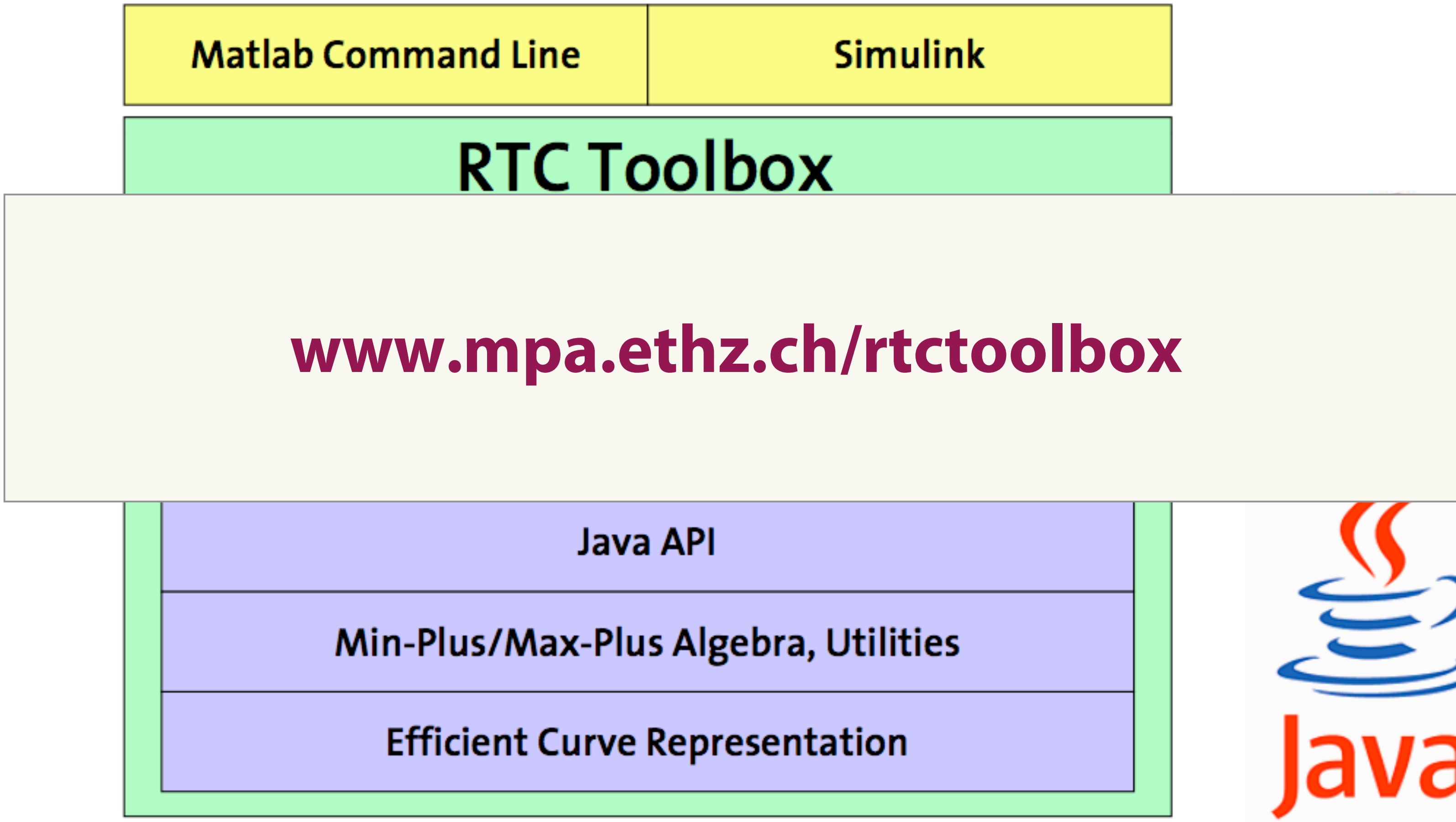
Mixed Hierarchical Scheduling



The RTC Toolbox

www.mpa.ethz.ch/rtctoolbox

The RTC Toolbox



RTC - Summary

- **Modeling: count-based abstraction**
  - captures burstiness of event streams and variability of the resources as functions
- **Analysis: min-plus and max-plus algebra**
  - can be computed efficiently with tool support
- **Modular and compositional**
  - possible combination with other methods, e.g. standard event models, ECA, simulation
- **Modeling of state-dependencies is difficult**
  - extension of RTC: an active area of study
  - various work combines concepts in RTC with automata

References and Readings

Real-Time Calculus:

1. Jean-Yves Le Boudec and Patrick Thiran: "Network Calculus", Lecture Notes in Computer Science 2050, Springer Verlag, January 2004 (Chapter 1 & 3)

2. Francois Baccelli, Guy Cohen, Geert Jan Olsder and Jean-Pierre Quadrat: "Synchronization and Linearity: An Algebra for Discrete Event Systems", Wiley, 1992

3. Samarjit Chakraborty, Simon Künzli, Lothar Thiele: "A General Framework for Analysing System Properties in Platform-Based Embedded System Designs", IEEE Design Automation & Test in Europe (DATE), 2003

4. Alexander Maxiaguine, Samarjit Chakraborty, Simon Künzli and Lothar Thiele: "Evaluating Schedulers for Multimedia Processing on Buffer-Constrained SoC Platforms", IEEE Design & Test of Computers, special issue on Embedded Systems for Real-Time Multimedia, Sep-Oct 2004

5. Many other papers can be found in the RTC Toolbox website

References and Readings

Other Formal Analysis Methods

1. Kai Richter and Rolf Ernst: "Event Model Interfaces for Heterogeneous System Analysis", Design Automation and Test in Europe Conference (DATE), 2002

2. Kai Richter, Dirk Ziegenbein, Marek Jersak, and Rolf Ernst: "Model Composition for Scheduling Analysis in Platform Design", 39th Design Automation Conference (DAC), 2002

3. Insik Shin, Insup Lee: "Compositional Real-Time Scheduling Framework," Proceedings of the 25th IEEE Real-Time Systems Symposium, RTSS 2004, pp. 57-67, Lisbon, Portugal, December 2004

4. Samarjit Chakraborty, Linh T.X. Phan and P.S. Thiagarajan: "Event Count Automata: A State-based Model for Stream Processing Systems", 26th IEEE Real-Time Systems Symposium (RTSS), Miami, Florida, Dec 2005

5. C. Norstrom, A. Wall and W. Yi: "Timed automata as task models for event-driven systems", 6th International Workshop on Real-Time Computing and Applications Symposium (RTCSA), Hong Kong, China, 1999

References and Readings

Hybrids of RTC and others:

1. Samarjit Chakraborty, Simon Künzli, Lothar Thiele, Andreas Herkersdorf and Patricia Sagmeister: "Performance Evaluation of Network Processor Architectures: Combining Simulation with Analytical Estimation", Computer Networks, 41(5):641-665, 2003 [RTC+Simulation]

2. Linh T.X. Phan, Samarjit Chakraborty, P. S. Thiagarajan, and Lothar Thiele: "Composing Functional and State-based Performance Models for Analyzing Heterogeneous Real-Time Systems", 28th IEEE Real-Time Systems Symposium (RTSS), Tucson, Arizona, Dec 2007 [RTC+ECA]

3. S. Kunzli, A. Hamann, R. Ernst and L. Thiele: "Combined approach to system level performance analysis of embedded systems", 13th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)}, Salzburg, Austria, 2007 [RTC+SEM]

4. Linh T.X. Phan, Samarjit Chakraborty, and P. S. Thiagarajan: "A Multi-Mode Real-Time Calculus", 29th IEEE Real-Time Systems Symposium (RTSS), Barcelona, Spain, Dec 2008 [Multi-mode version of RTC]

Reference and Readings

Simulation and Trace-based Analysis

1. Kanishka Lahiri, Anand Raghunathan and Sujit Dey: "System-Level Performance Analysis for Designing On-Chip Communication Architectures", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, June 2001

2. Kanishka Lahiri, Anand Raghunathan and Sujit Dey: "Efficient Exploration of the SoC Communication Architecutre Design Space", International Conference on Computer-Aided Design (ICCAD), November 2000

3. http://www.simplescalar.com/

4. http://www.systemc.org/

linhphan AT seas.upenn.edu

Office: Room 279 South