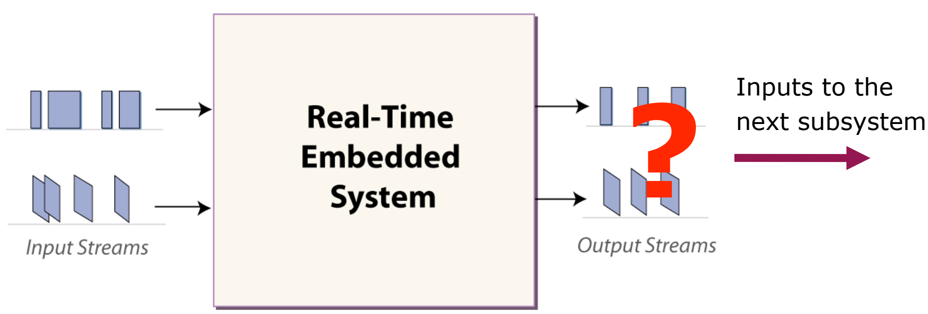
Performance Analysis Of Distributed Real-Time Embedded Systems

Linh T.X. Phan

linhphan AT cis.upenn.edu

Computer and Information Science
University of Pennsylvania

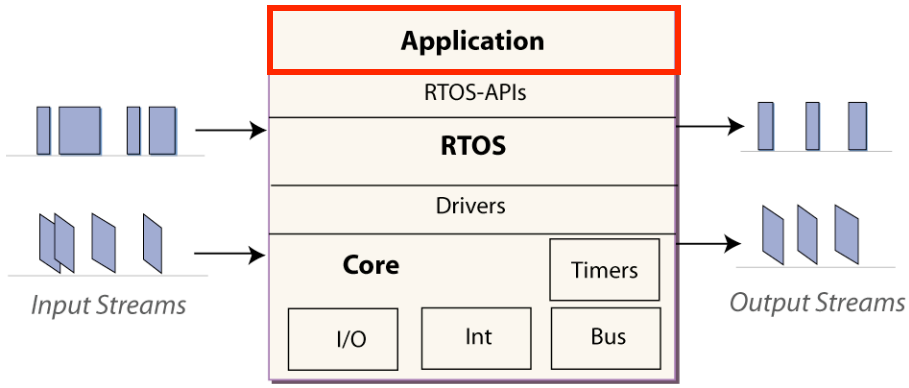
Overview of a system component



Distributed components communicate via input and output streams

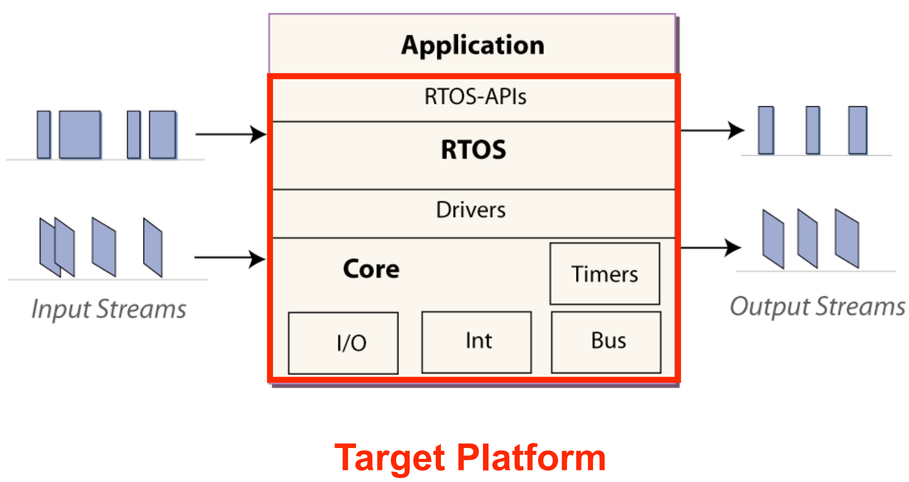
2

Inside the box…



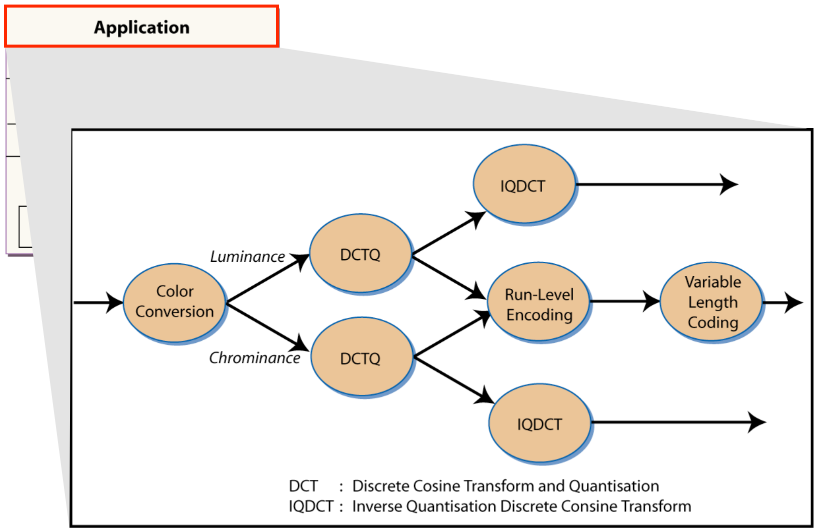
3

Inside the box…



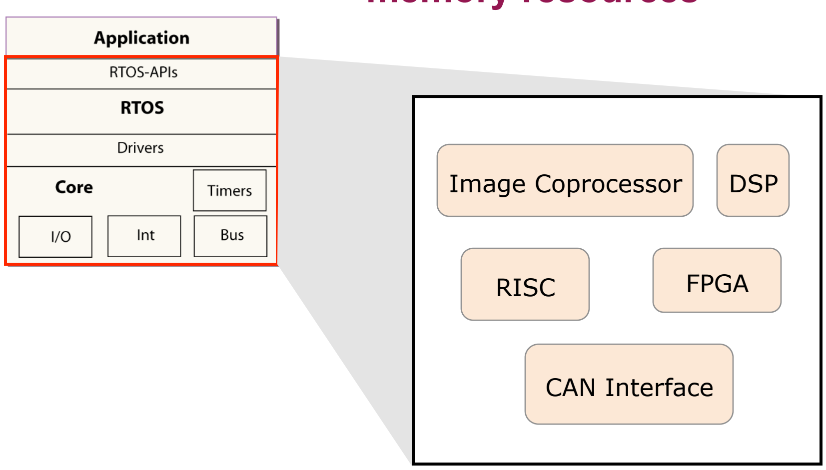
4

1. Streaming application tasks



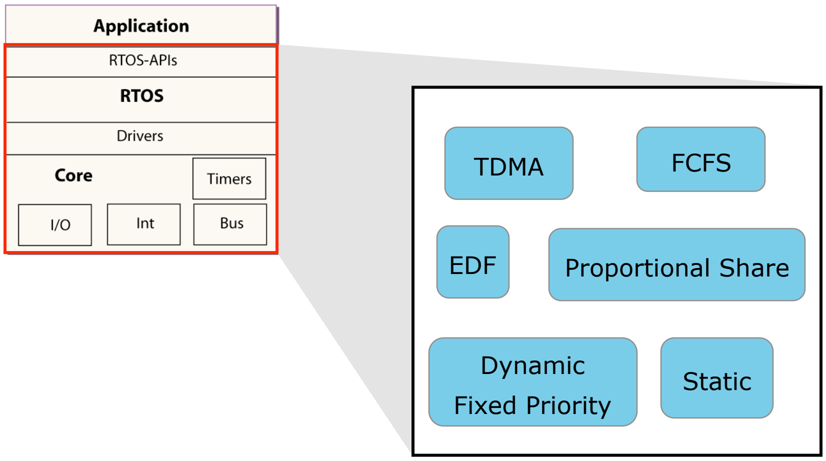
5

2. Heterogeneous computing and memory resources



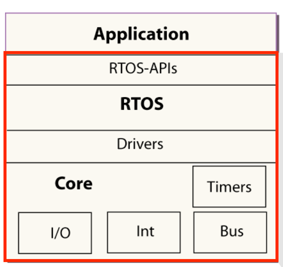
6

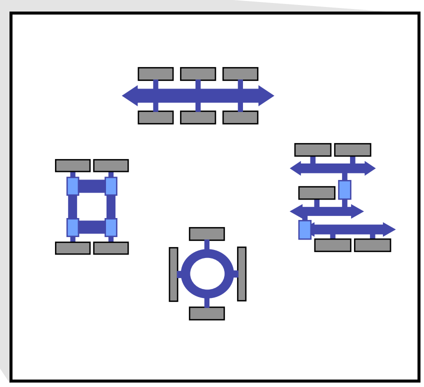
3. Heterogeneous RTOS scheduling and synchronization protocols



7

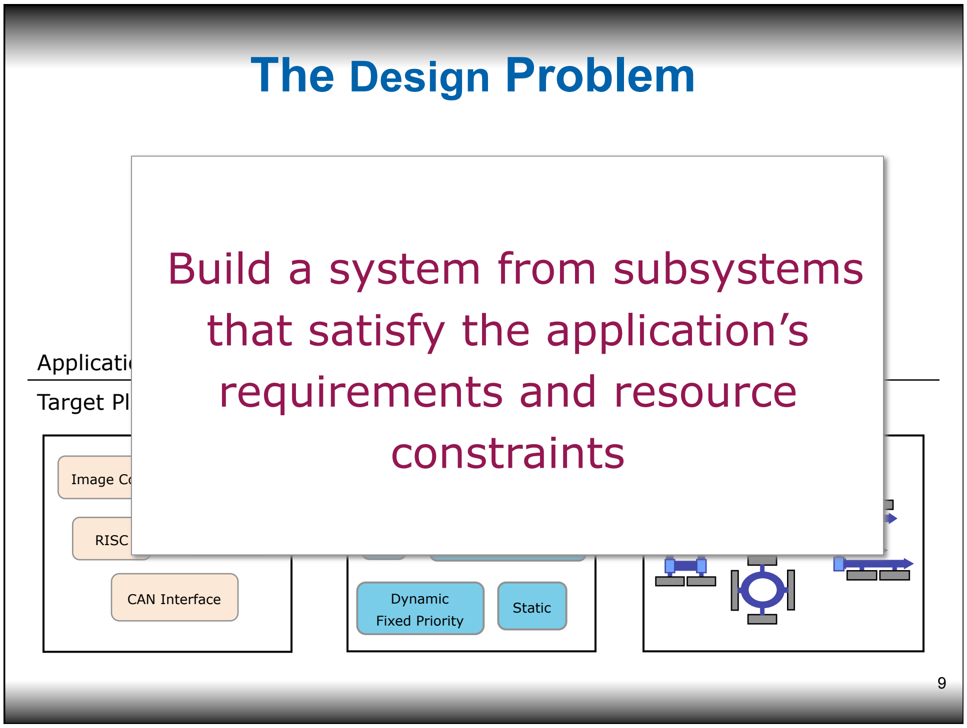
4. Heterogeneous communication resources



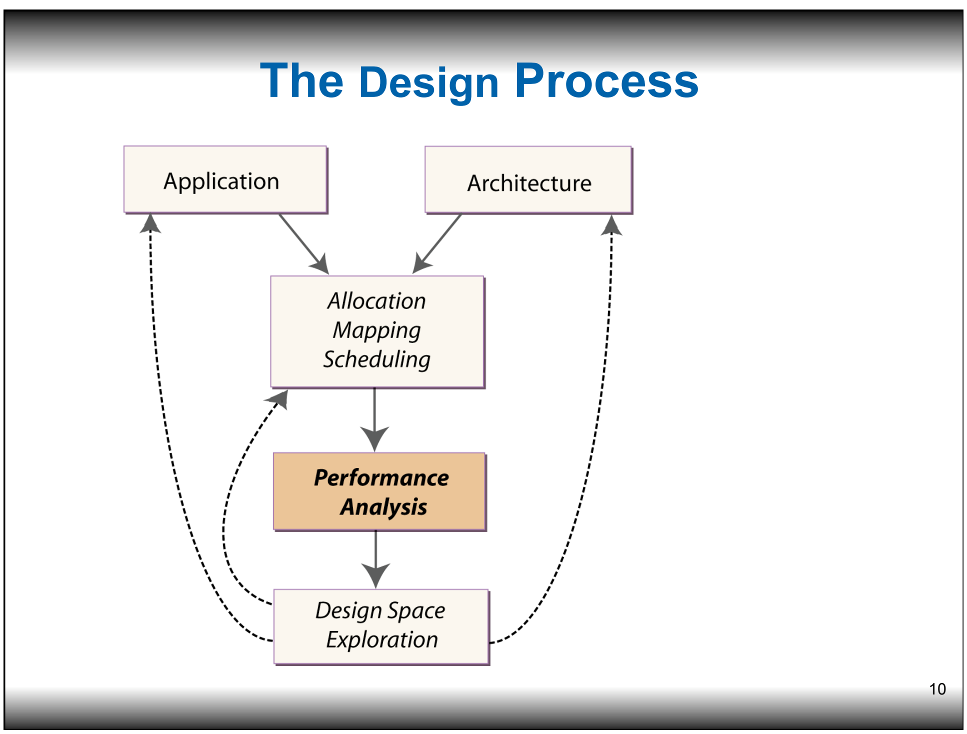


- Topology (ring, mesh, star)
- Switching strategies (packet, circuit)
- Routing strategies
  (static, dynamic, reconfigurable)
- Arbitration policies (dynamic, TDM, CDMA)

8

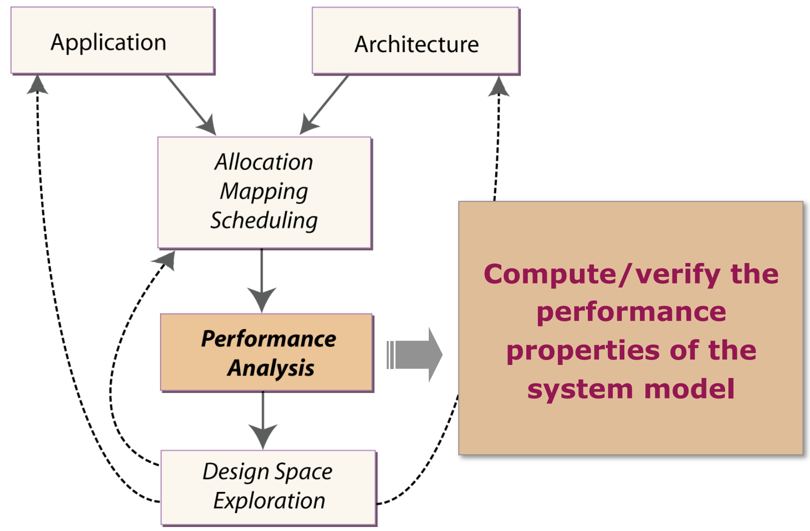


The Design Problem
Build a system from subsystems that satisfy the application's requirements and resource constraints
Applicati
Target Pl
Image C
RISC
CAN Interface
Dynamic
Fixed Priority
Static
9



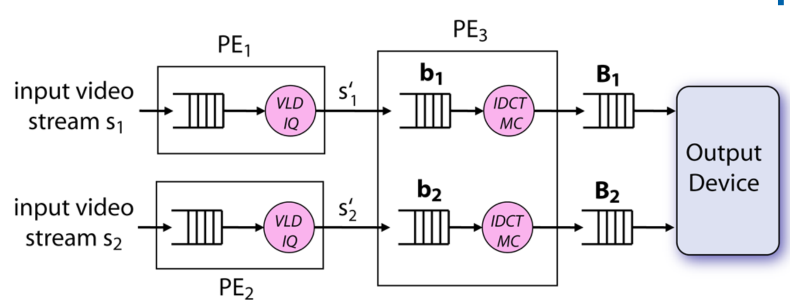
The Design Process
Application
Architecture
Allocation
Mapping
Scheduling
Performance
Analysis
Design Space
Exploration
10

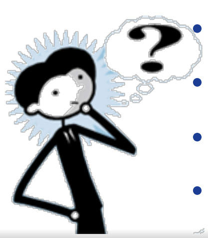
The Performance Analysis Problem



11

e.g. Architecture of a Picture-in-Picture App.



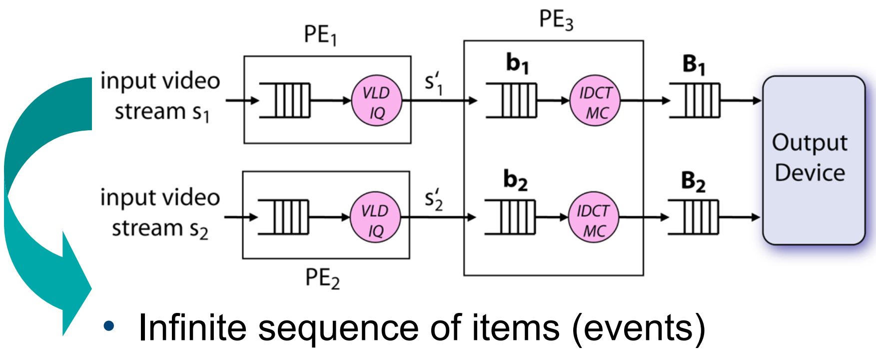


- Maximum fill-level (backlog) of the buffers?
- Maximum end-to-end delay of the stream?
- Characteristics of the output stream?
- Characteristics of the remaining resource?

. . . . . .

12

Key Challenges: Complex Event Streams



- Infinite sequence of items (events)
- Highly bursty
- Events of multiple types interleaving
- Varied memory and execution demand
- …

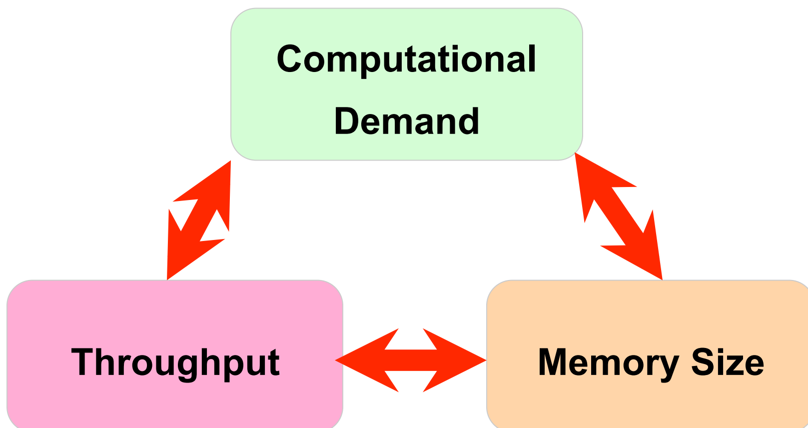
13

Key Challenges: Complex Tasks & Architectures

- Complex processing semantics
  - fill-level of the buffers
  - synchronization between different streams
- Heterogeneous computing and communication resources
- Various scheduling policies
  - EDF, Fixed-Priority, TDMA, etc.
  - complex state-dependent scheduling schemes

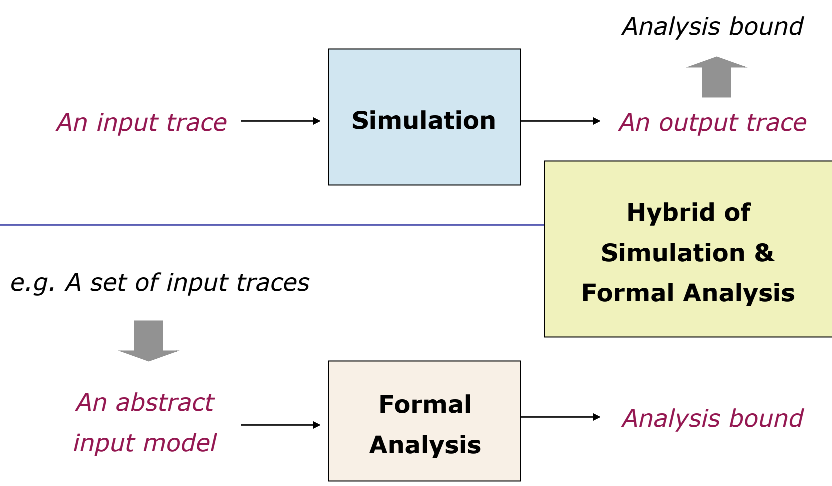
14

Complex Trade-offs



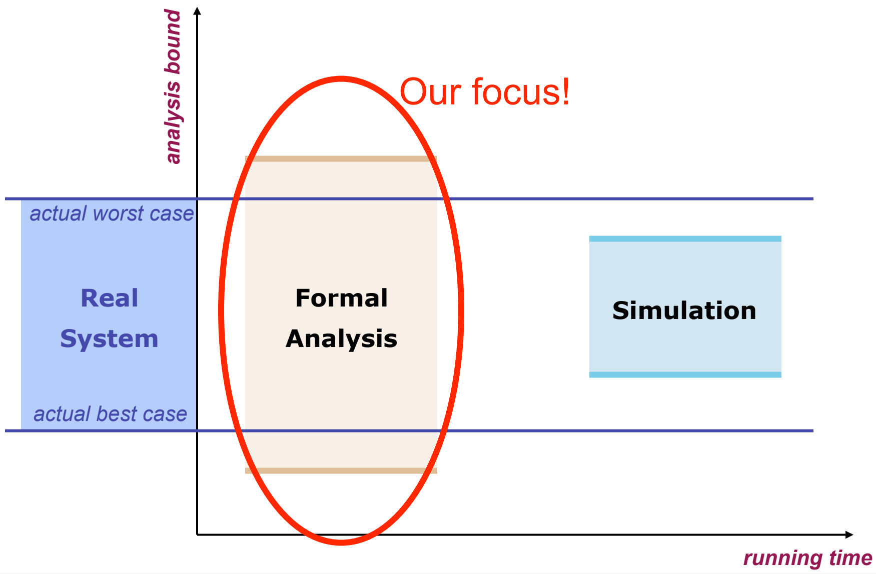
15

Two Categories of Performance Analysis



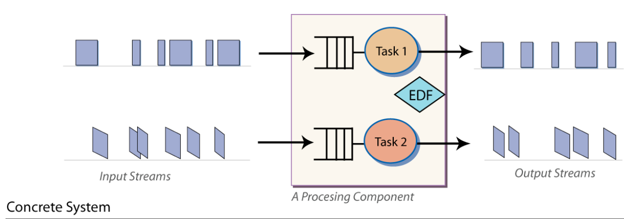
16

Simulation vs Formal Analysis



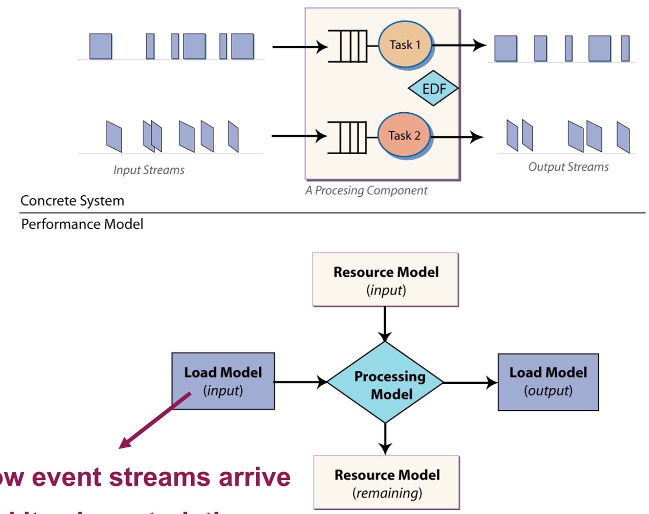
17

Formal Analysis Overview



18

Formal Analysis Overview

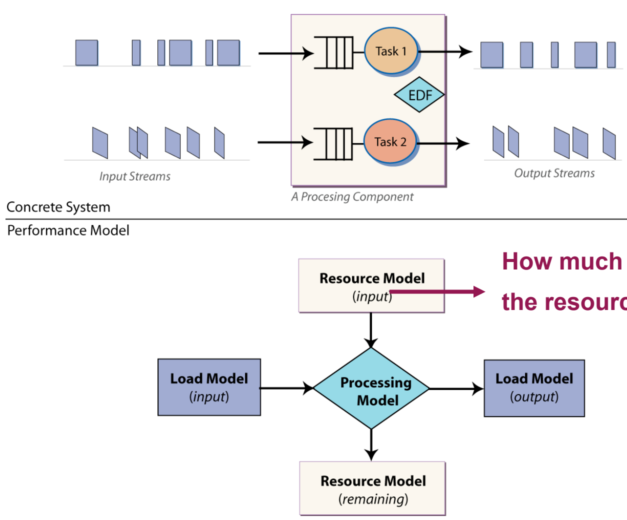


How event streams arrive

and its characteristics

19

Formal Analysis Overview

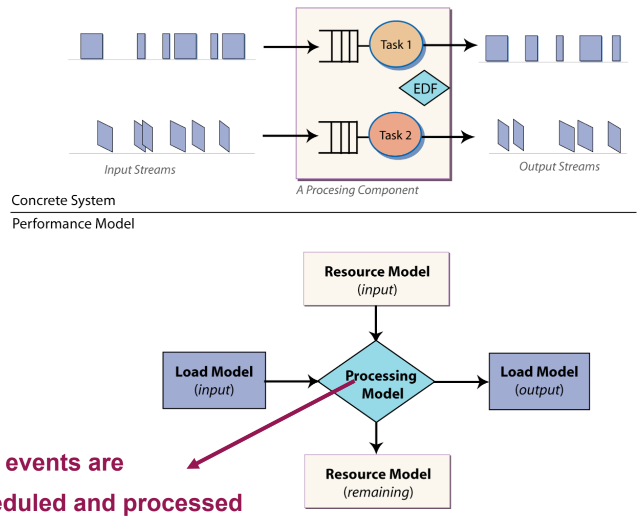


How much and when

the resource are available

20

Formal Analysis Overview



How events are scheduled and processed

21

Formal Models and Analysis Methods

1 Standard Event Models (SEM)

– periodic, periodic with jitter/burst

– based on classical scheduling theory

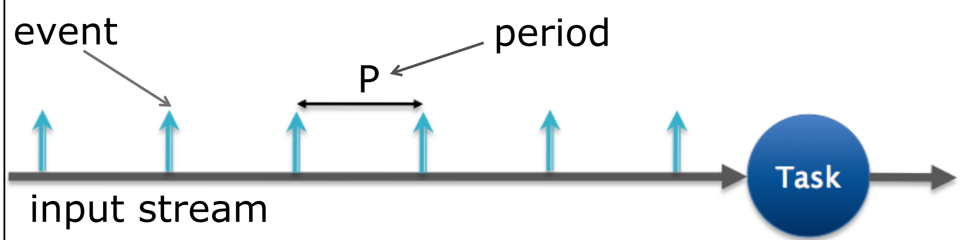
Simple, easy to analyze

Too restrictive

Pessimistic results

22

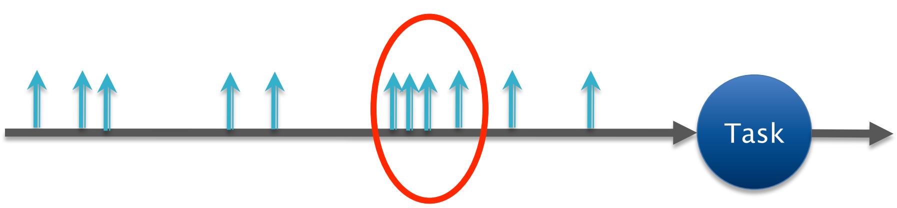
Standard Event Models



- Input data items arrive regularly, one data item (event) every P time units

23

Distributed Real-Time Systems



- Sensed data items (events) arrive irregularly, may come in burst

➠ Periodic task model: ?

- Pessimistic analysis results if assuming classical models

24

2 Queing theory and variations

- e.g., Real-Time Calculus (RTC)
- streams & resources: functions
- analysis: (min,+) and (max,+) algebra

👍 **Capture burstiness of streams & resource availability**

👍 **Highly efficient**

👎 **Cannot model state-dependencies**

25

3 Automata-based models

- Timed automata, event count automata, dataflow graphs, etc.

👍 **Models state-dependencies**

👍 **Highly accurate**

👎 **Large systems ⟶ inefficient**

26

4 Hybrid Models and Methods

- RTC + SEM
- RTC + ECA
- Multi-Mode RTC
- …

Good accuracy-efficiency trade-off

27

The rest of the talk…

Formal Analysis using

Real-Time Calculus
(RTC)

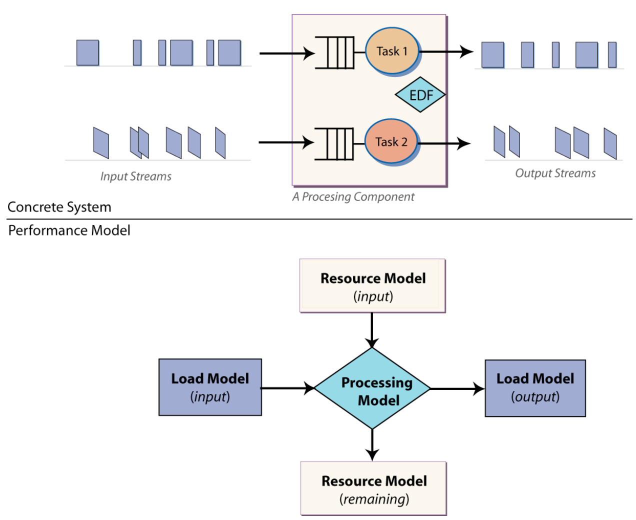
28

RTC Background

- Originated from Network Calculus in computer networks domain
  - extended for real-time embedded systems
- Worst-case deterministic formal analysis
  - variant of classical queuing theory
- Abstract models: count-based abstraction
- Analysis: min-plus / max-plus algebra

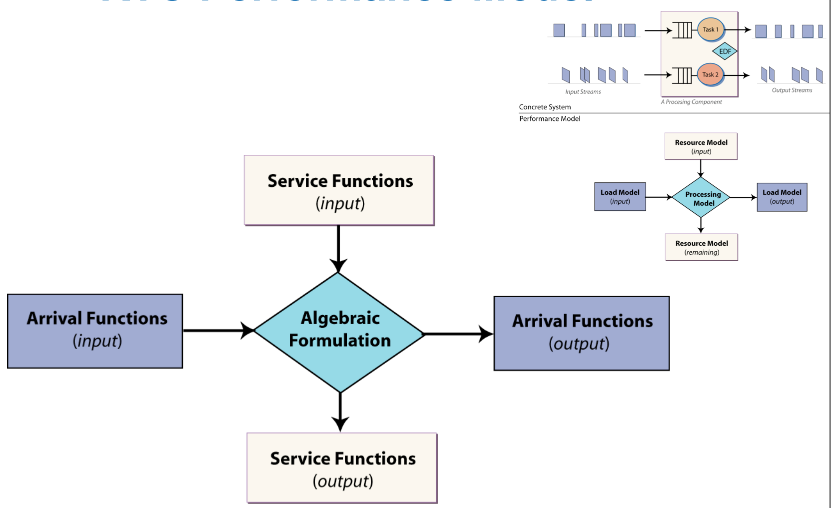
29

Recall…



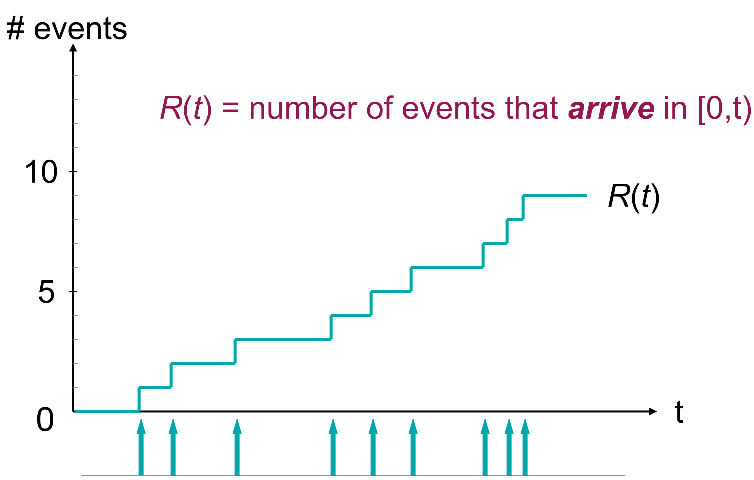
30

RTC Performance Model



31

An Arrival Pattern



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

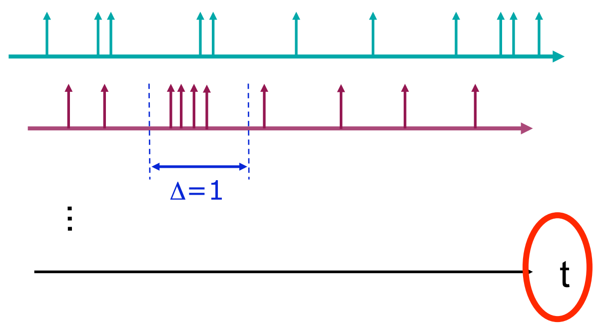
32

Count-based Abstraction

sliding window size

Δ	Lower bound	Upper bound
1	1	4
2	3	6
⋮		

A set of arrival patterns



concrete time instant

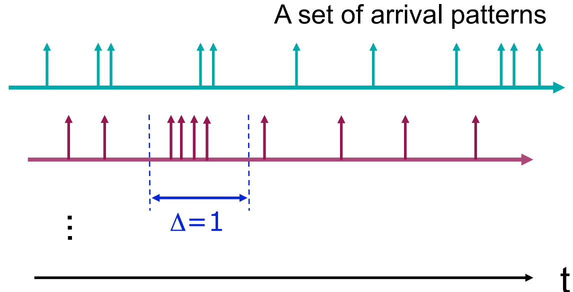
33

Load Model: Arrival Functions

$\alpha = (\alpha^l, \alpha^u)$

Δ	$\alpha^l(\Delta)$	$\alpha^u(\Delta)$
1	1	4
2	3	6
⋮		

A set of arrival patterns

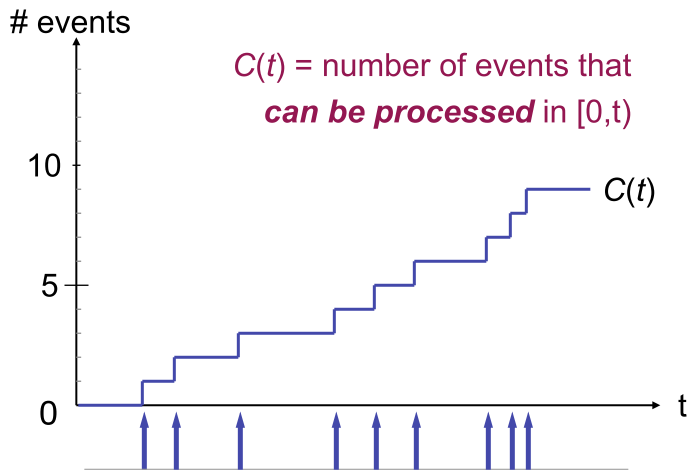


An arrival pattern $R(t)$ satisfies α iff

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

34

A Service Pattern



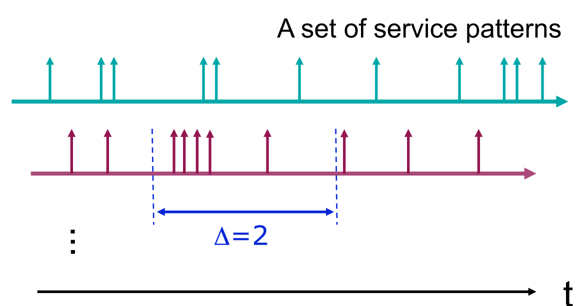
#events that can be processed in [t, t+Δ) is: $C(t+\Delta) - C(t)$

35

Service Model: Service Functions

$\beta = (\beta^l, \beta^u)$

Δ	$\beta^l(\Delta)$	$\beta^u(\Delta)$
1	0	3
2	3	5
⋮		



A service pattern $C(t)$ satisfies β iff

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

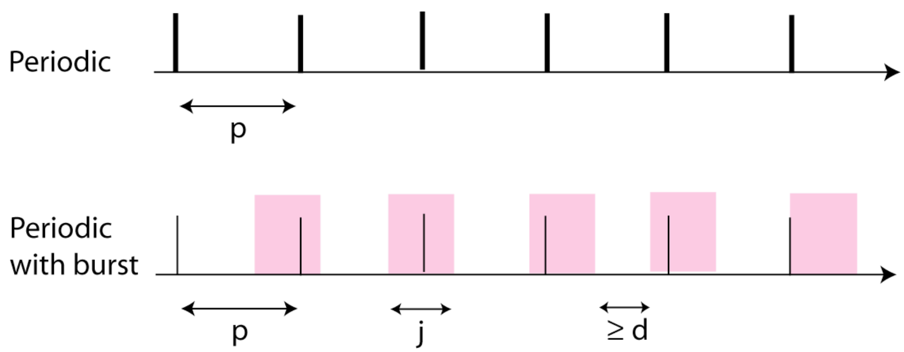
36

Units of Arrival and Service Functions

- [$R(t)$, $\alpha(\Delta)$] and [$C(t)$, $\beta(\Delta)$] can also be specified in terms of the number of resource units
  - processor cycles, transmiting bit, etc.
- Should *always* convert to the same unit before performing analysis

Examples of Arrival and Service Functions

Periodic Event Streams

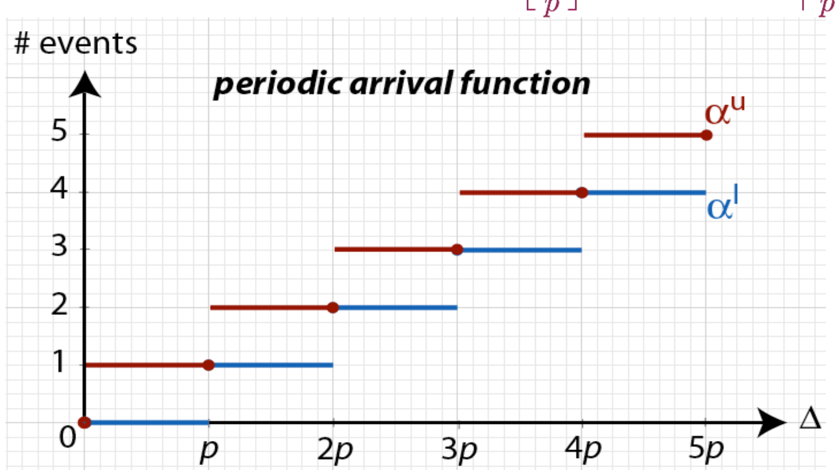


p: period *j: jitter* *d: minimum inter-arrival distance*

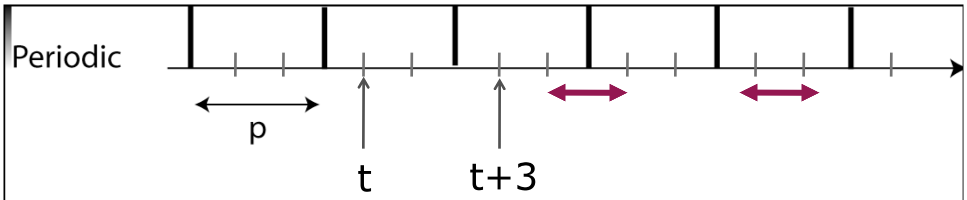
39

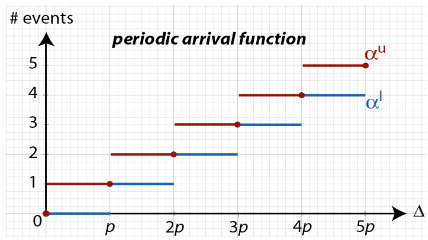
Arrival Function of Periodic Event Streams

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



40



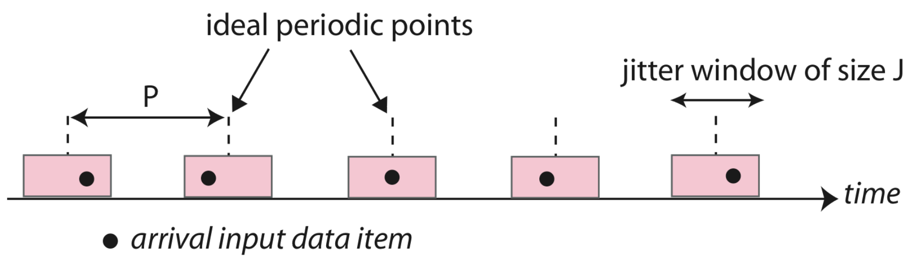


$$\alpha^u(2) = 1, \ \alpha^l(2) = 0$$
$$\alpha^u(3) = 1, \ \alpha^l(3) = 1$$

P = 3: exactly 1 event arrives every 3 time units

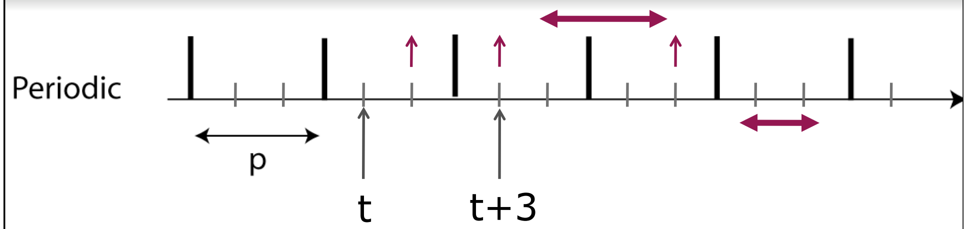
→ For any given t, exactly 1 event arrives in the interval [t, t+3)

Periodic with Jitter Event Streams



Suppose min distance between any two events is 0, then

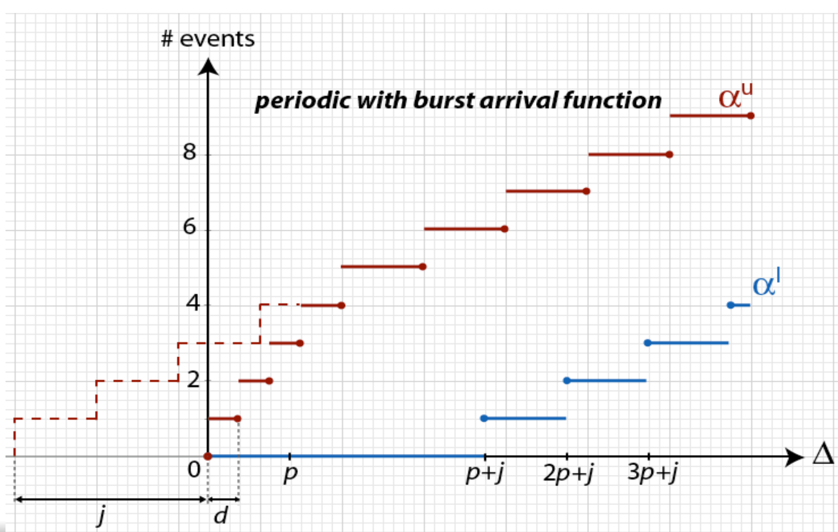
$$\text{In any interval of length } \Delta: \ \text{at most } \left\lceil \frac{\Delta + j}{p} \right\rceil \text{ events}$$
$$\text{at least } \left\lfloor \frac{\Delta - j}{p} \right\rfloor \text{ events}$$



$$\alpha^u(3) = \lceil (3+2)/3 \rceil = 2$$
$$\alpha^l(3) = \lfloor (3-2)/3 \rfloor = 0$$

P = 3, J = 2: at most 2, at least 0 in [t, t+3)

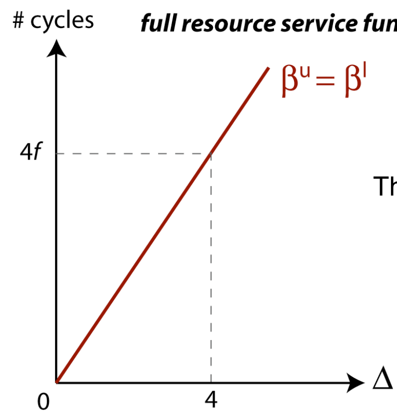
$$\alpha^l(\Delta) = \left\lfloor \frac{\Delta - j}{p} \right\rfloor \qquad \alpha^u(\Delta) = \min\left\{ \left\lceil \frac{\Delta + j}{p} \right\rceil, \left\lceil \frac{\Delta}{d} \right\rceil \right\}$$



44

Common Resources

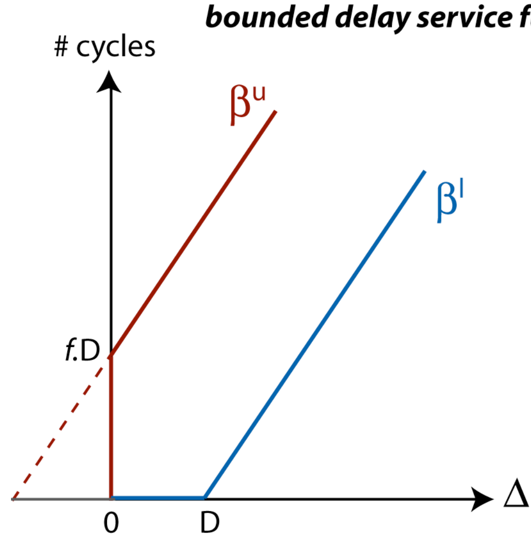
full resource service function



The processor is always available

f = processor frequency

bounded delay service function



TDMA Resource

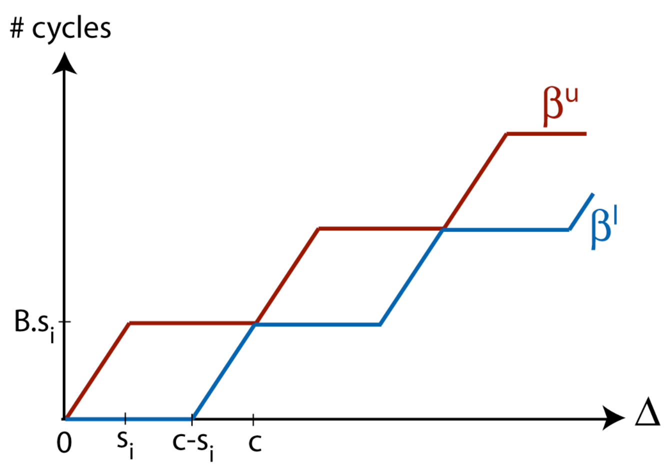
- A shared resource of bandwidth *B*
- *n* applications: App_1, …, App_n
- TDMA policy
  - a resource slot of length s_i is assigned to App_i in every cycle of length *c*
  - ➢ the resource given to App_i is bounded by

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

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

47

TDMA Resource



48

Performance Analysis Of Distributed Real-Time Embedded Systems
(cont)

Linh T.X. Phan
linhphan AT cis.upenn.edu

Computer and Information Science
University of Pennsylvania

Previous Lecture…

- General concepts of the design and performance analysis of distributed real-time embedded 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

2

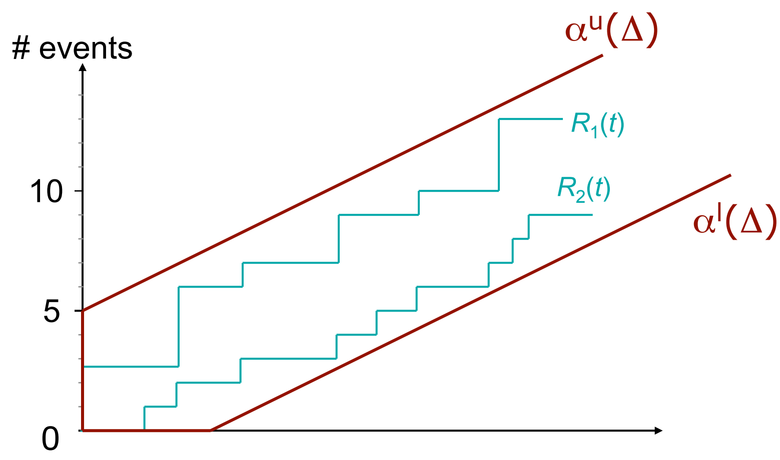
Real-Time Calculus (cont.)

- 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 $\alpha(\Delta)$

Arrival Function of A Set of Concrete Patterns



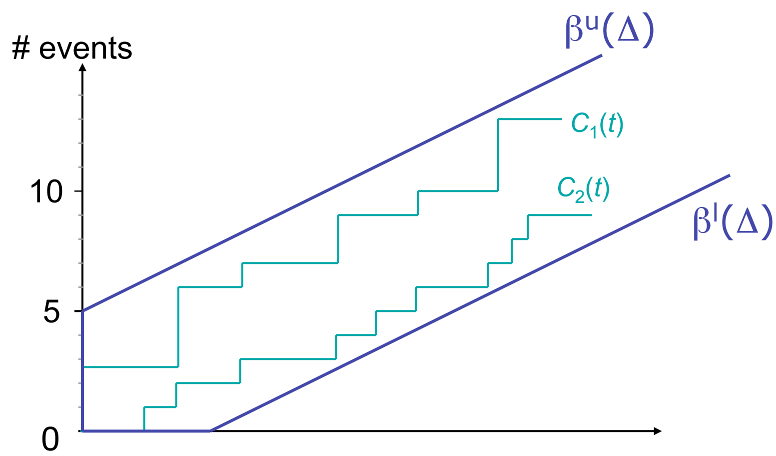
5

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 $\beta(\Delta)$

6

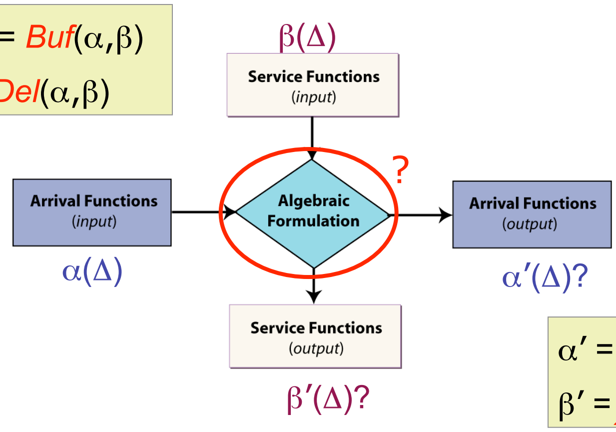
Service Function of A Set of Concrete Patterns



RTC Performance Model

backlog = *Buf*(α,β)

delay = *Del*(α,β)



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

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

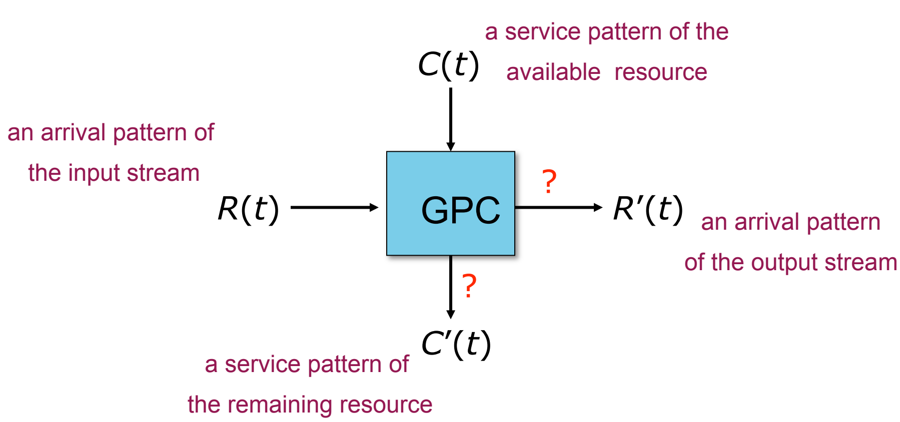
The functions f_α, f_β, *Buf*, *Del* must take into account the *scheduling policy* and the *processing semantics* of the component

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

9

A concrete system component

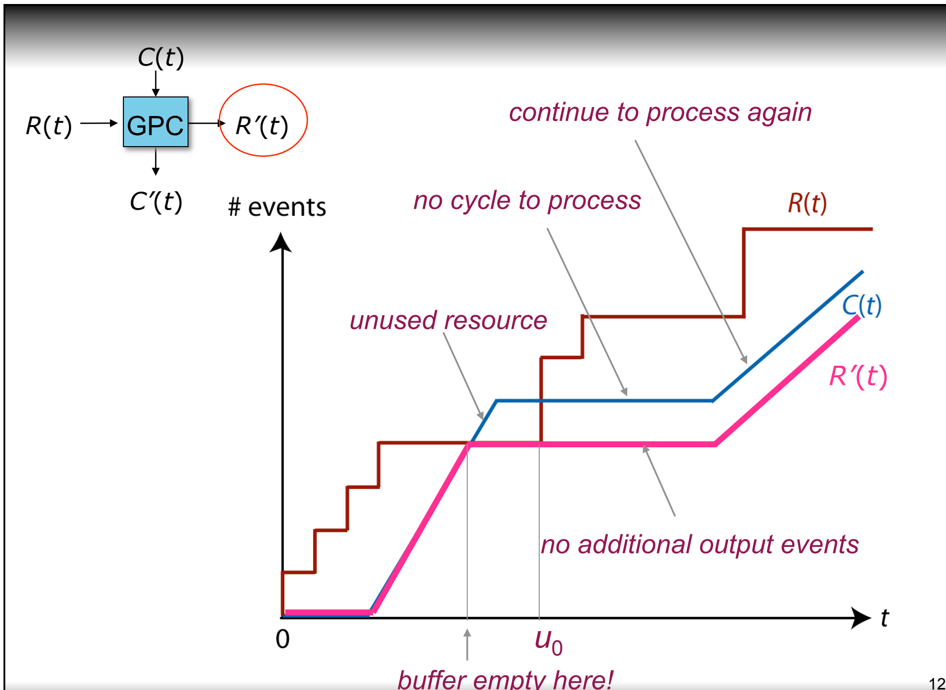


10

Greedy Processing Component

- Triggered by incoming events
- 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

11



12

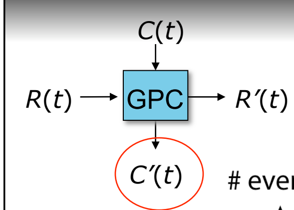
GPC: Output Stream

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

For all u≤t:

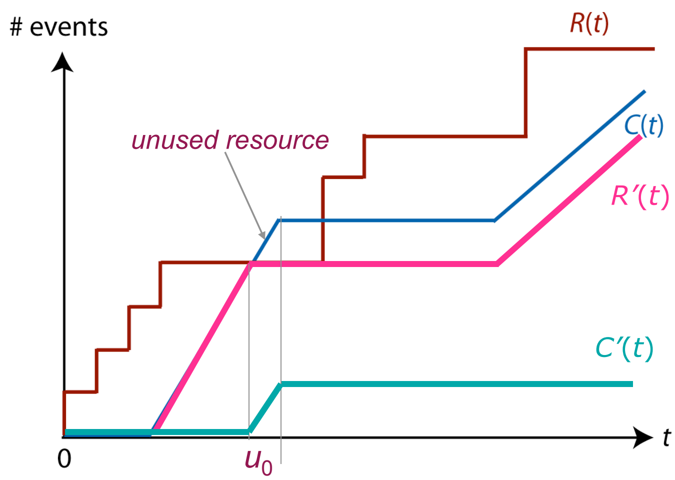
- $R'(u) \le R(u)$ and $R'(t) \le R'(u) + C(t) - C(u)$
  - #output-events in [0,u) is no more than #input-events in [0,u)
  - #output-events in [u,t) is no more than #events that can be processed in [u,t)

  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, $R'(t) = R'(u_0) + C(t) - C(u_0)$

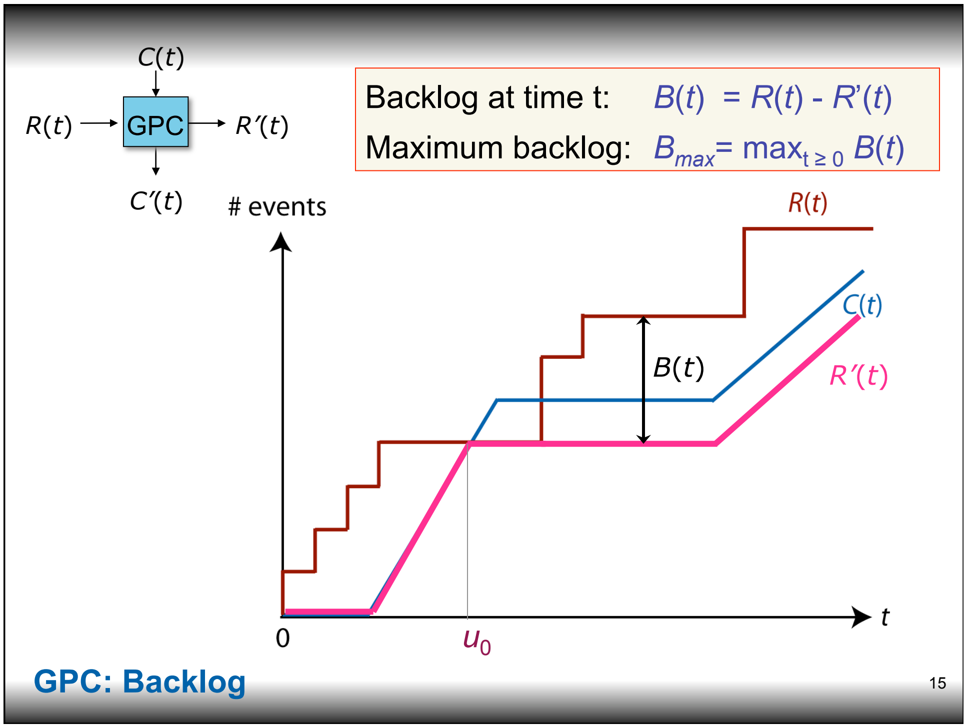


Conservative use of resource:

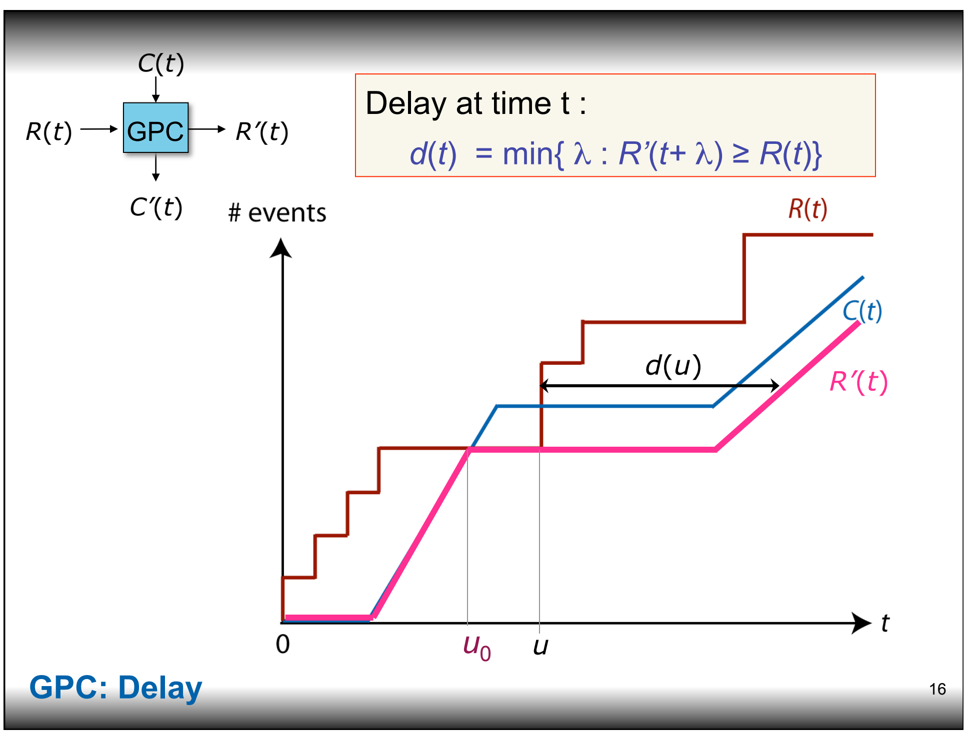
C(t) = C'(t) + R'(t)



GPC: Remaining Resource

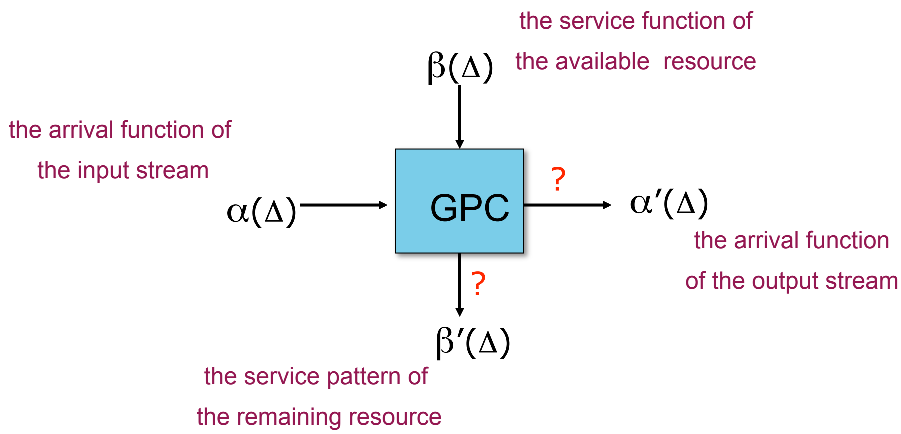


Backlog at time t: B(t) = R(t) - R'(t)
Maximum backlog: B_max = max_{t ≥ 0} B(t)
C(t)
R(t)
GPC
R'(t)
C'(t)
# events
R(t)
C(t)
B(t)
R'(t)
0
u0
t
GPC: Backlog
15



Delay at time t :
d(t) = min{ λ : R'(t+ λ) ≥ R(t)}
C(t)
R(t)
GPC
R'(t)
C'(t)
# events
R(t)
C(t)
d(u)
R'(t)
0
u0
u
t
GPC: Delay
16

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

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

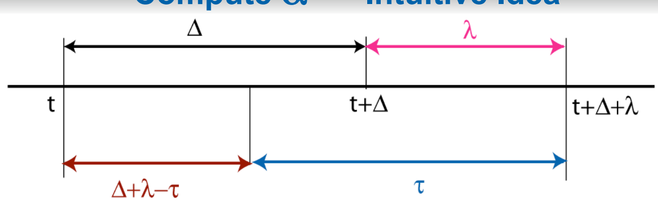
$$(f \overline{\oslash} g)(t) = \inf_{u \ge 0} \{ f(t+u) - g(u) \}$$

GPC: Output Bounds

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

19

Compute $\alpha^{u'}$ - Intuitive Idea



- *max_output*$(\Delta+\lambda) \leq$ *max_input*$(\Delta+\lambda-\tau)$ + *max_processed*(τ), $\forall\ 0 \leq \tau \leq \Delta+\lambda$
  - $\leq \sup_{0 \leq \tau \leq \Delta+\lambda} \{\alpha^{u}(\Delta+\lambda-\tau) + \beta^{u}(\tau)\} = \gamma(\Delta+\lambda)$, with $\gamma = \alpha^{u} \otimes \beta^{u}$
- *max_output*$(\Delta) \leq$ *max_output*$(\Delta+\lambda)$ – *min_processed*(λ), $\forall\ \lambda \geq 0$
  - $\leq \gamma(\Delta+\lambda) - \beta^{l}(\lambda),\ \forall\ \lambda \geq 0$
  - $\leq \inf_{\lambda \geq 0} \{\gamma(\Delta+\lambda) - \beta^{l}(\lambda)\} = (\gamma \varnothing \beta^{l})(\Delta)$
  - $\leq [(\alpha^{u} \otimes \beta^{u}) \varnothing \beta^{l}](\Delta)$
- Further, *max_output*$(\Delta) \leq$ *max_processed*$(\Delta) \leq \beta^{u}(\Delta)$
- $\Rightarrow\ \alpha^{u'} \leq \min\{(\alpha^{u} \otimes \beta^{u}) \varnothing \beta^{l},\ \beta^{u}\}$

20

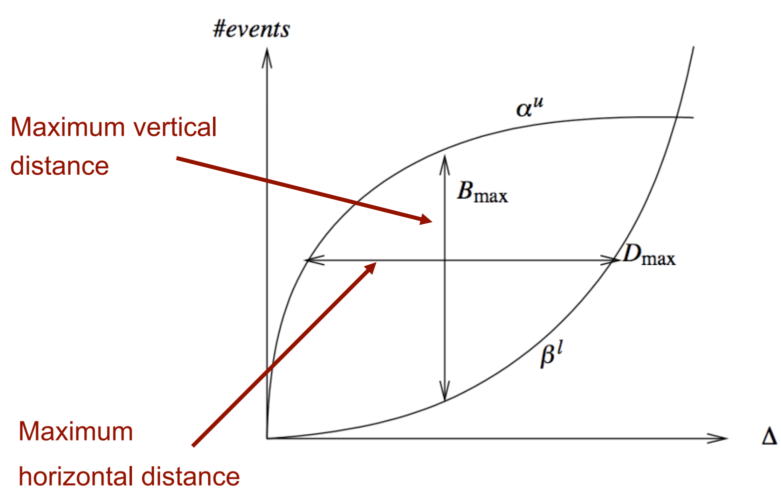
GPC: Backlog and Delay Bounds

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

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

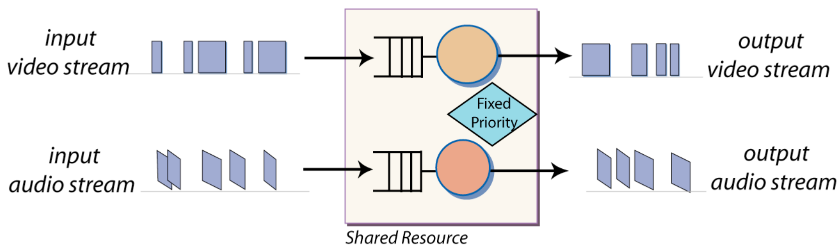
21

GPC: Backlog and Delay Bounds



22

Scheduling Multiple Event Streams

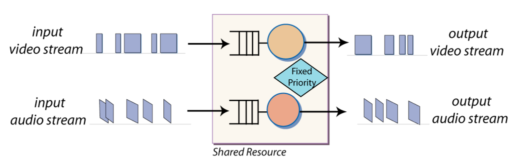


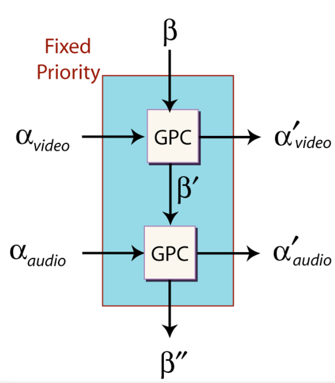
Fixed Priority:

- video stream has higher priority than audio stream
  - ➔ process the video stream first
- remaining resource is used to process the audio stream

23

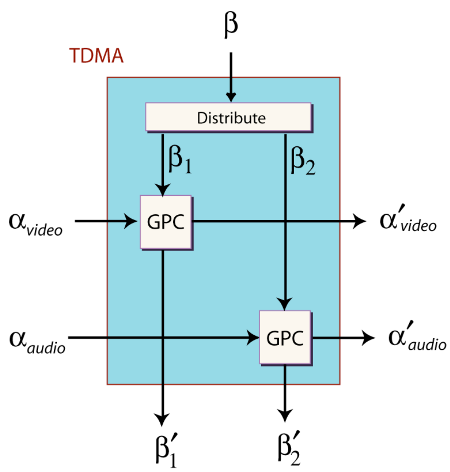
Fixed Priority Scheduling





24

TDMA Scheduling



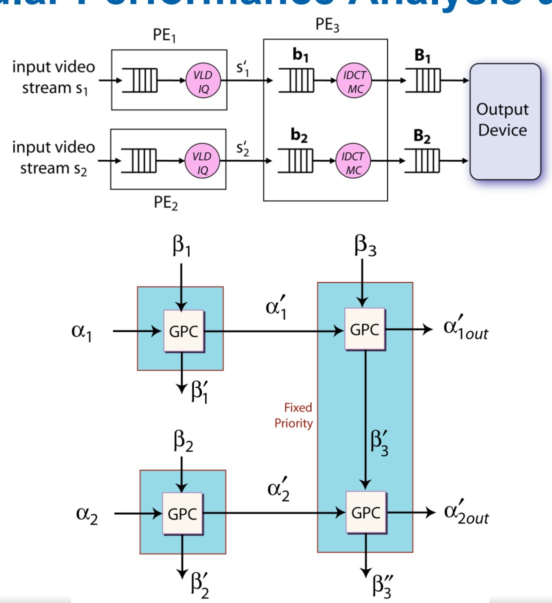
β_i : computed based on the length of the TDMA cycle c and the slot s_i

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

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

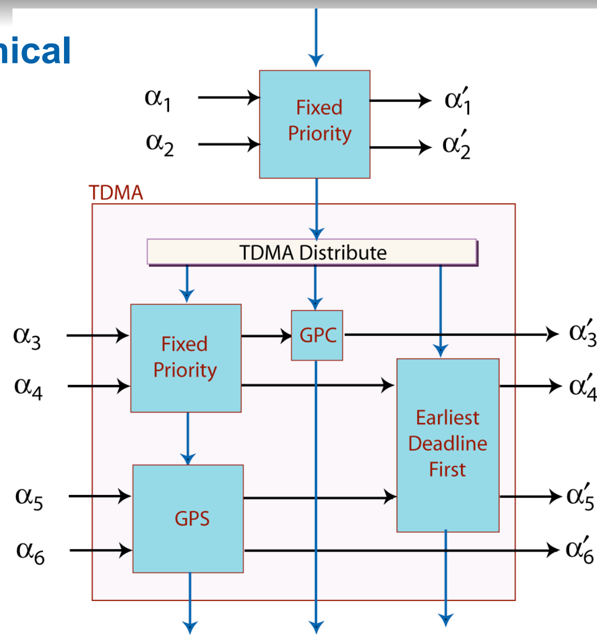
25

Modular Performance Analysis using RTC



26

Mixed Hierarchical Scheduling



27

The RTC Toolbox

www.mpa.ethz.ch/rtctoolbox

28

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

29

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. 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

30

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. 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

3. 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

4. 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]

linhphan AT cis.upenn.edu

Office: Levine 275

33