

# Performance Analysis of Cyclic Dataflow Graphs

Third International Workshop on  
Foundations and Applications of Component-based Design

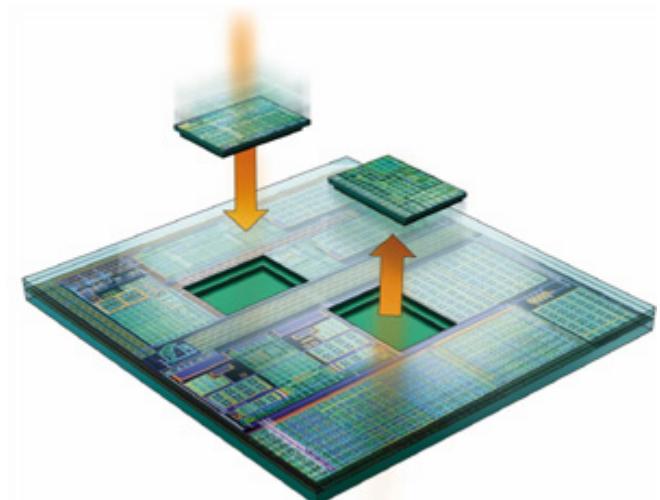
Lothar Thiele

# Analysis and Design

Embedded System =  
Computation + Communication + Resource Interaction

## Analysis:

Infer system properties from  
subsystem properties.

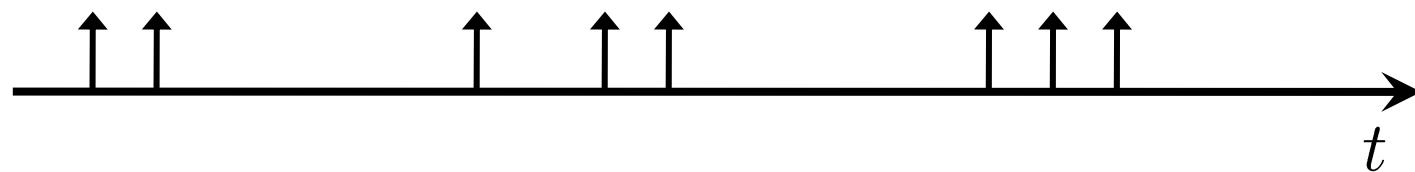


## Design:

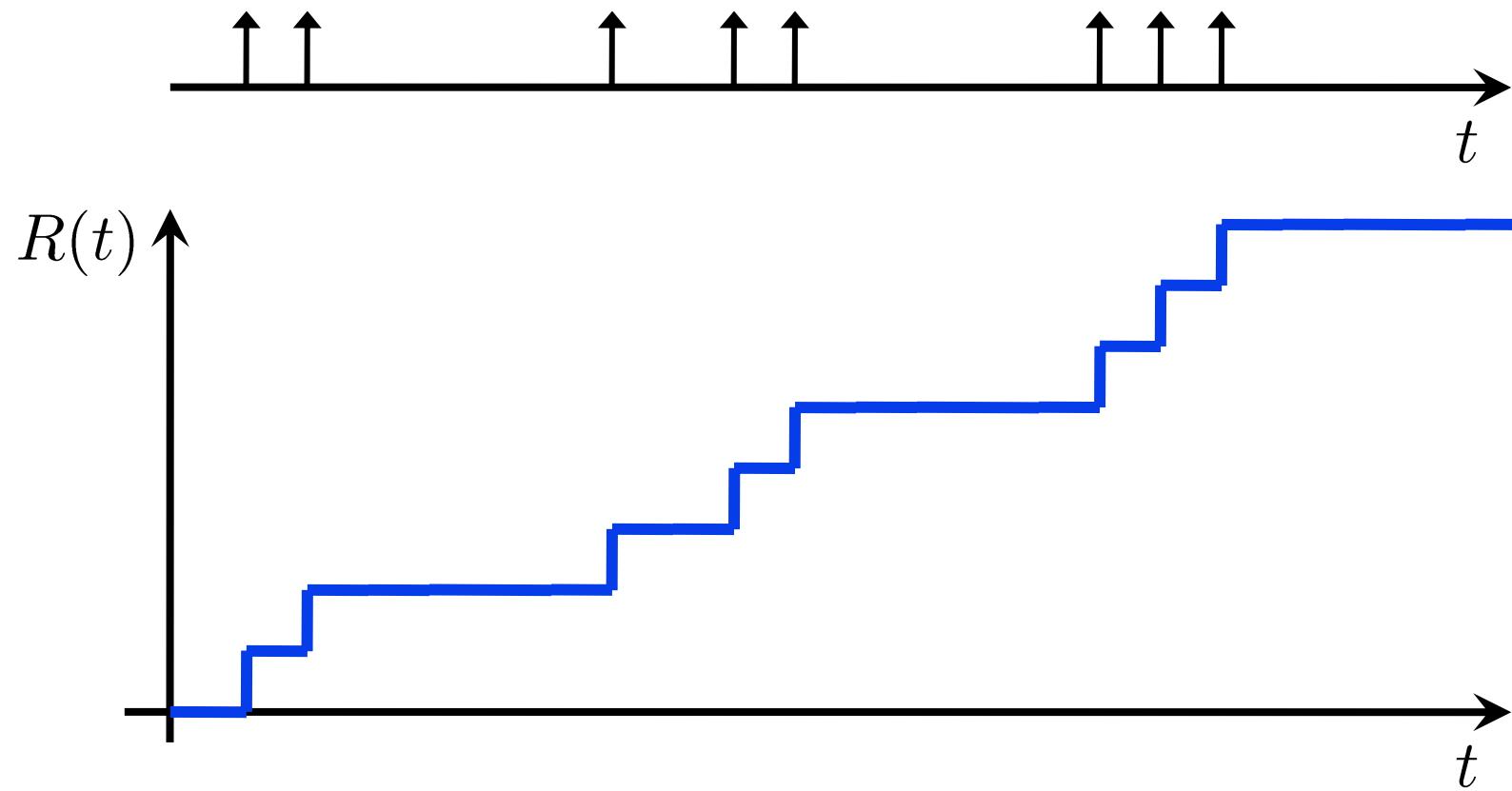
Build a system from subsystems  
while meeting requirements.

**extend the scope of analytic methods**

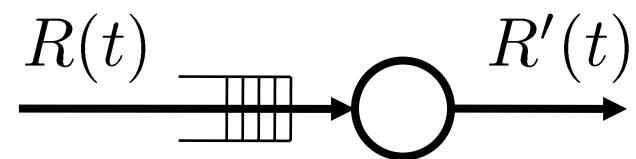
stream



stream

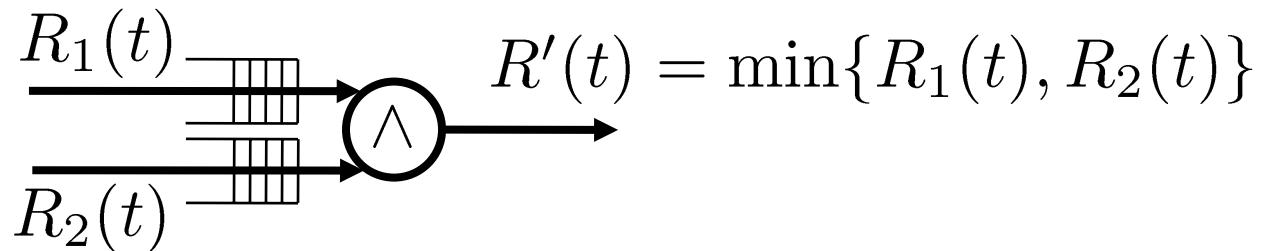
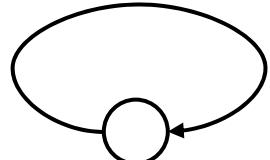


# process

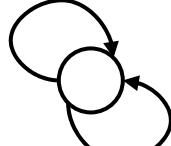


# process examples

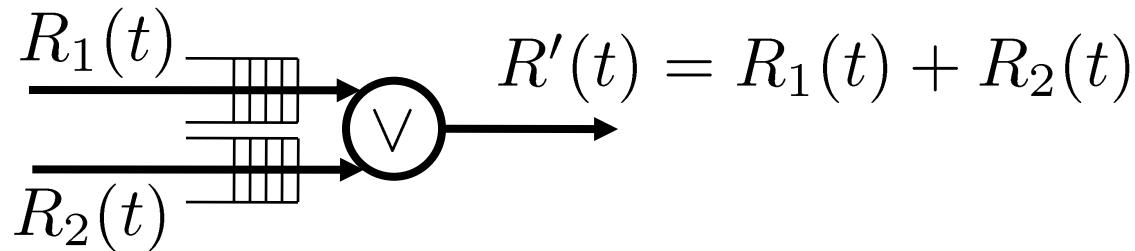
$$r_1 \wedge r_2 \rightarrow r'$$



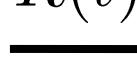
$$r_1 \rightarrow r'$$



$$r_2 \rightarrow r'$$



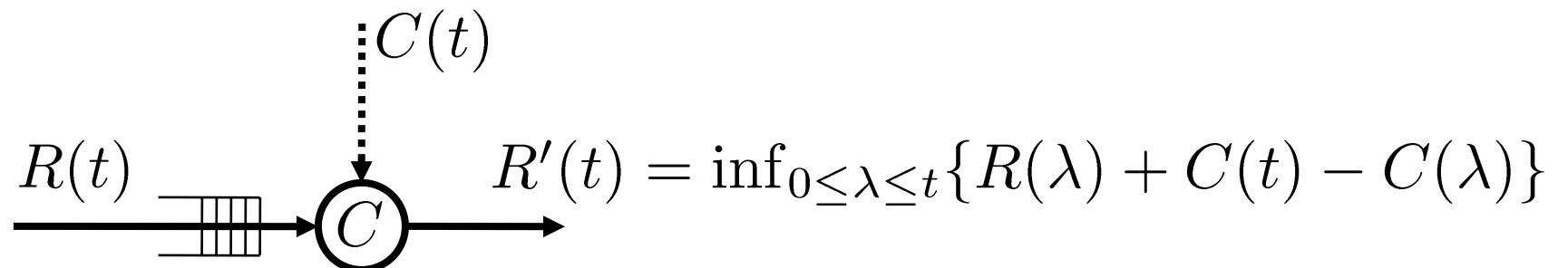
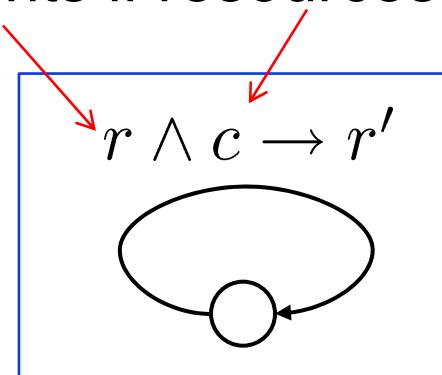
$$R(t)$$



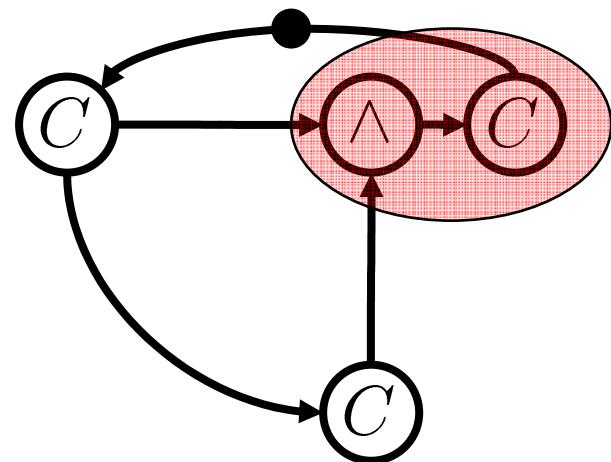
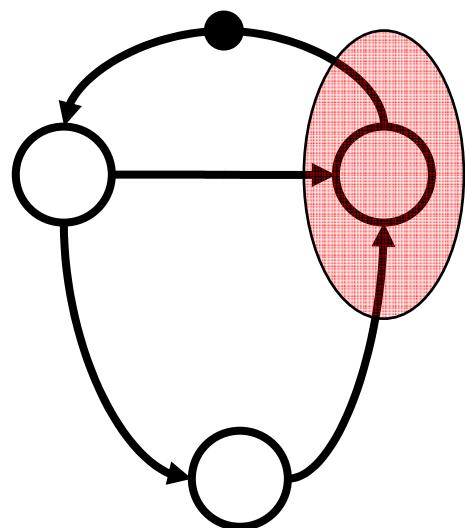
$$R'(t) = \lfloor \sigma \cdot R(t) \rfloor$$

# resource interaction

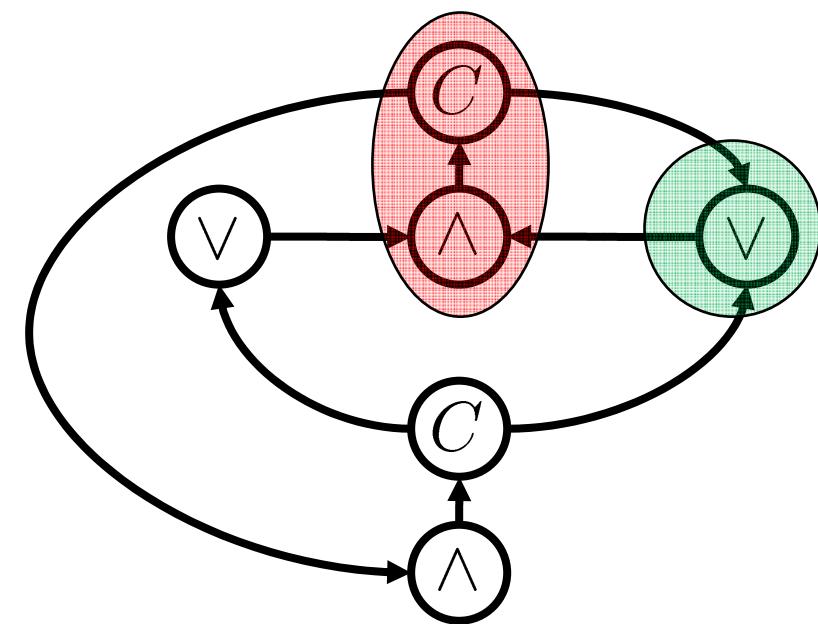
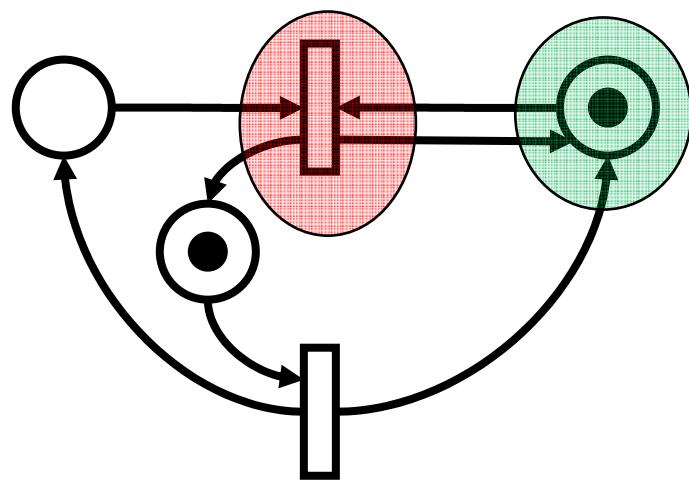
transfer events if resources are available



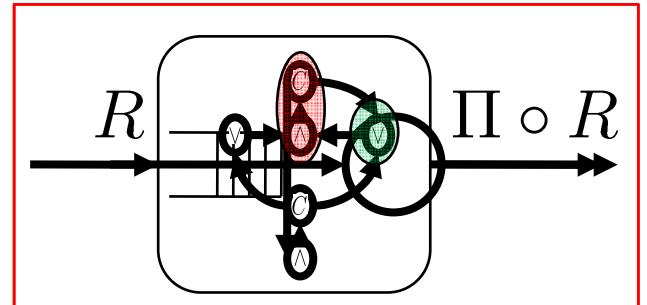
# example marked graph



# example petri net



# properties



monotone components:  $R_1 \geq R_2 \longrightarrow \Pi \circ R_1 \geq \Pi \circ R_2$

system simulation:  $R_i = \Pi \circ R_{i-1}$

unique fixed point:  $R = \Pi \circ R$

# abstraction

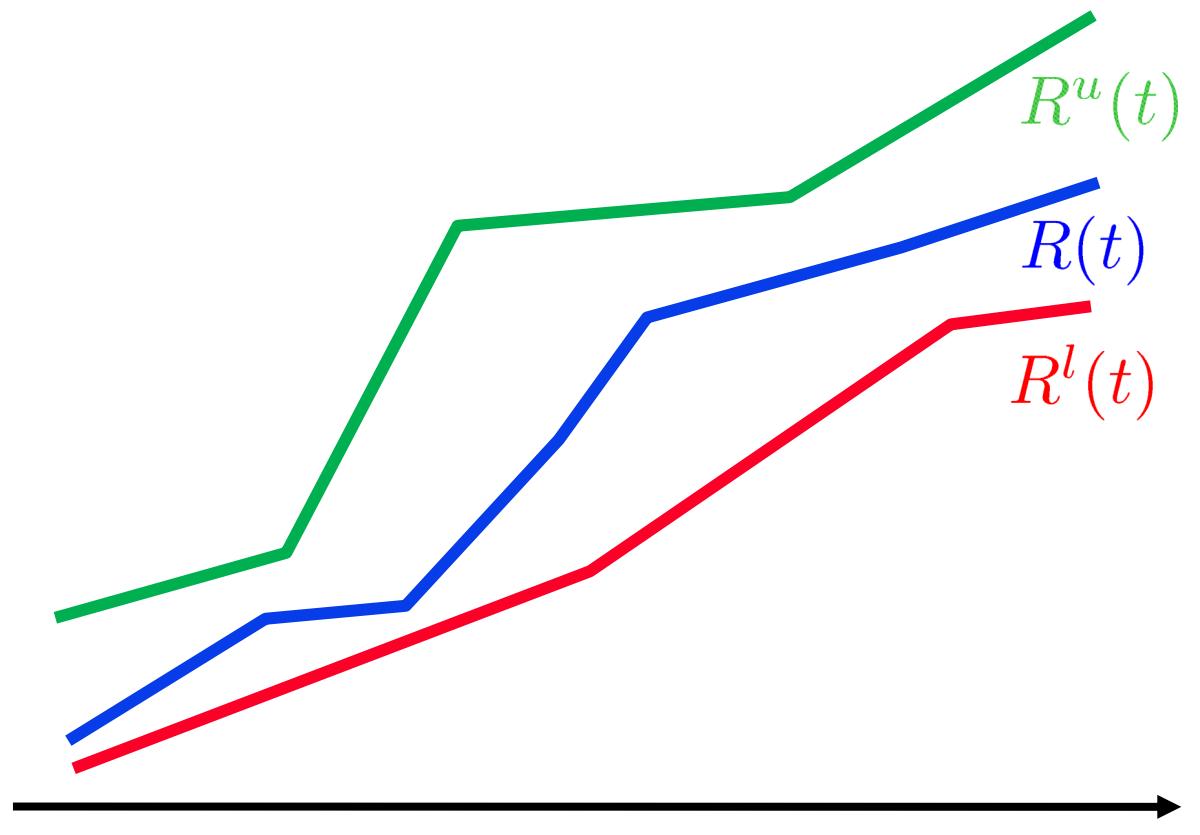
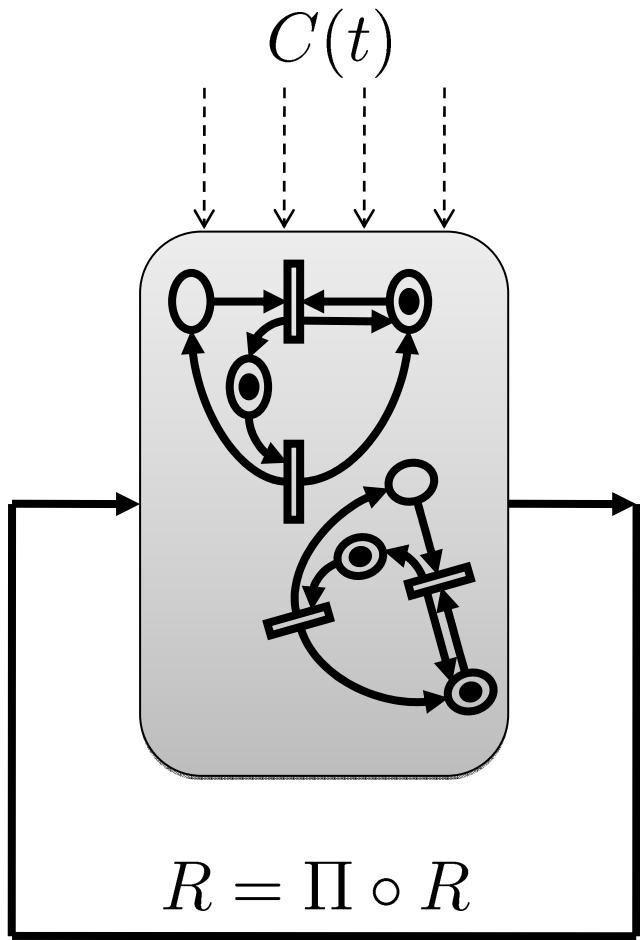
system abstraction:  $\Pi^u \geq \Pi \geq \Pi^l$

fixpoints:  $R^u = \Pi^u \circ R^u$        $R^l = \Pi^l \circ R^l$

stream bounds:  $R^u \geq R \geq R^l$

**introduce non-determinism**

use



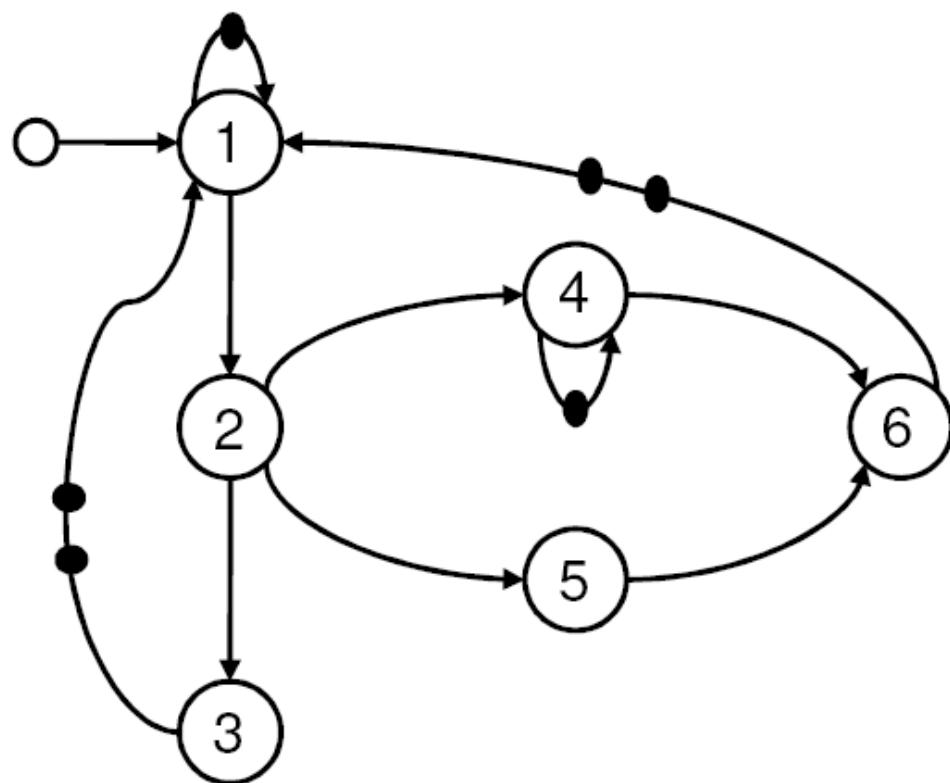
$$R^u = \Pi^u \circ R^u$$

$$R = \Pi \circ R$$

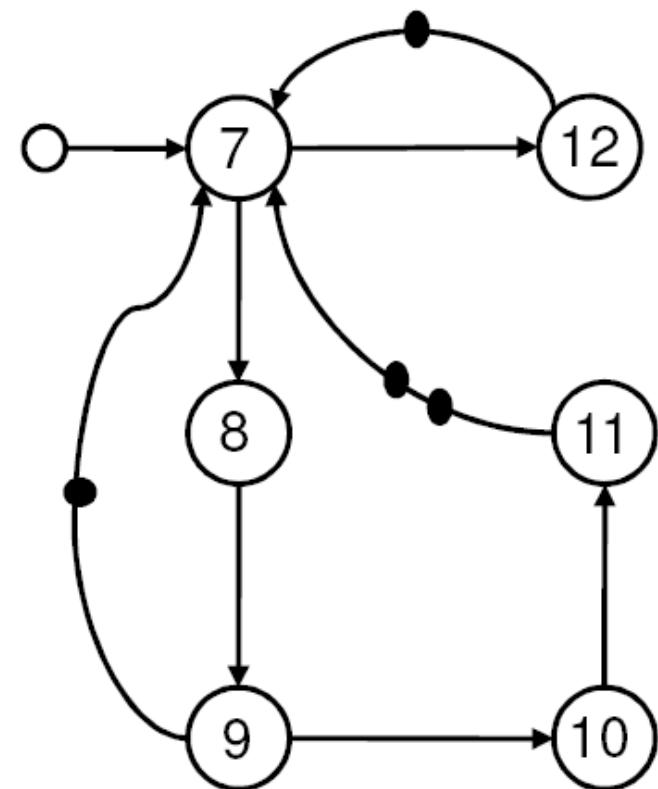
$$R^l = \Pi^l \circ R^l$$

# **analysis of marked graphs**

# examples



TD-SCDMA



WLAN

# questions

performance analysis of marked graphs has a long history,  
some results are related to  $(\min, +)$  algebra

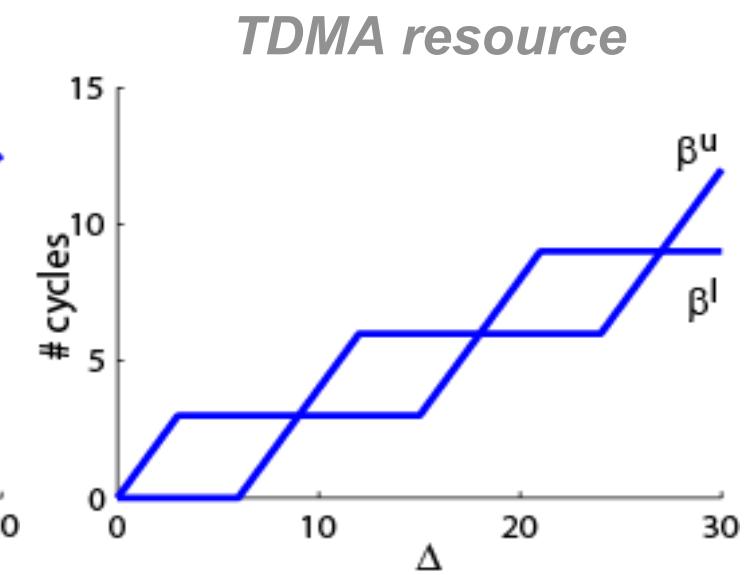
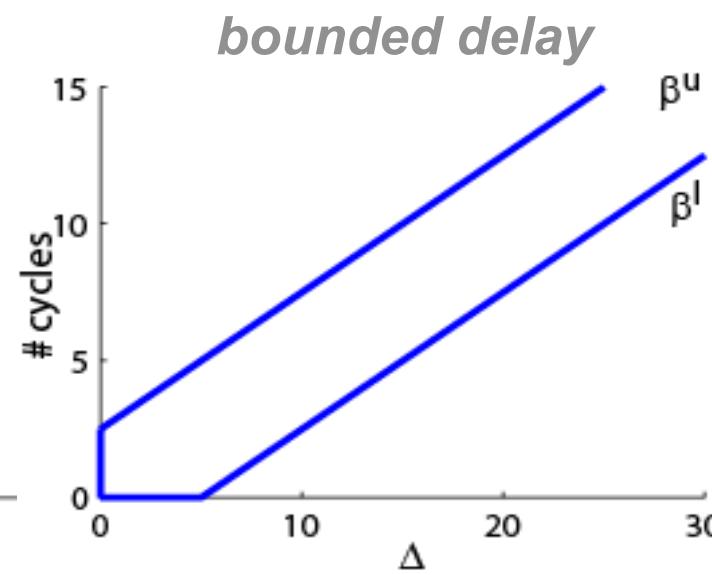
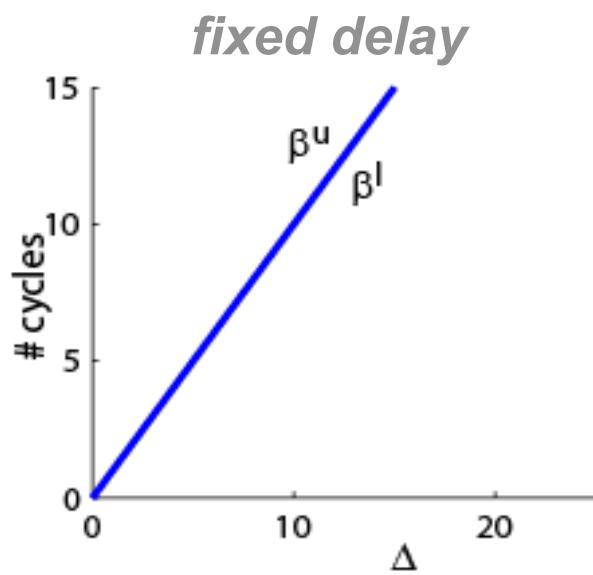
what is the performance in case of complex resource  
resource sharing (TDMA scheduling, fixed priority)?

how can we analyze properties related to jitter, bursts, non-  
periodic input streams, end-to-end delays?

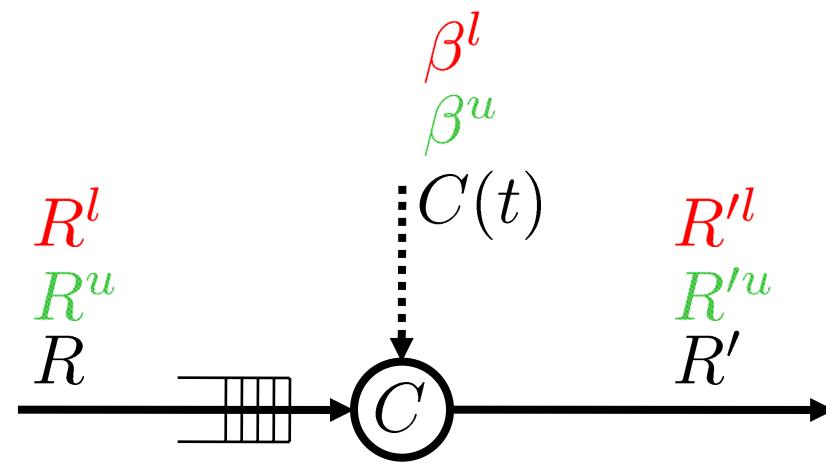
how do appropriate interfaces/components look like?

# resource abstraction

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



# resource abstraction

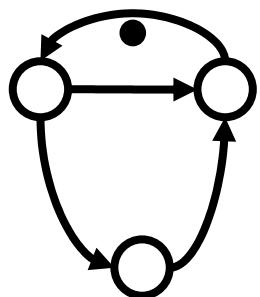


$$R'(t) = \inf_{0 \leq \lambda \leq t} \{R(\lambda) + C(t) - C(\lambda)\}$$

$$R'^u(t) = \inf_{0 \leq \lambda \leq t} \{R^u(\lambda) + \beta^u(t - \lambda)\} = R^u \otimes \beta^u$$

$$R'^l(t) = \inf_{0 \leq \lambda \leq t} \{R^l(\lambda) + \beta^l(t - \lambda)\} = R^l \otimes \beta^l$$

# system equations



smallest upper bound

$R^u = \beta^u \wedge S^u \otimes R^u$

$R^l = \beta^l \wedge S^l \otimes R^l$

largest lower bound

$R^u = (S^u)^* \otimes \beta^u$

$R^l = (S^l)^* \otimes \beta^l$

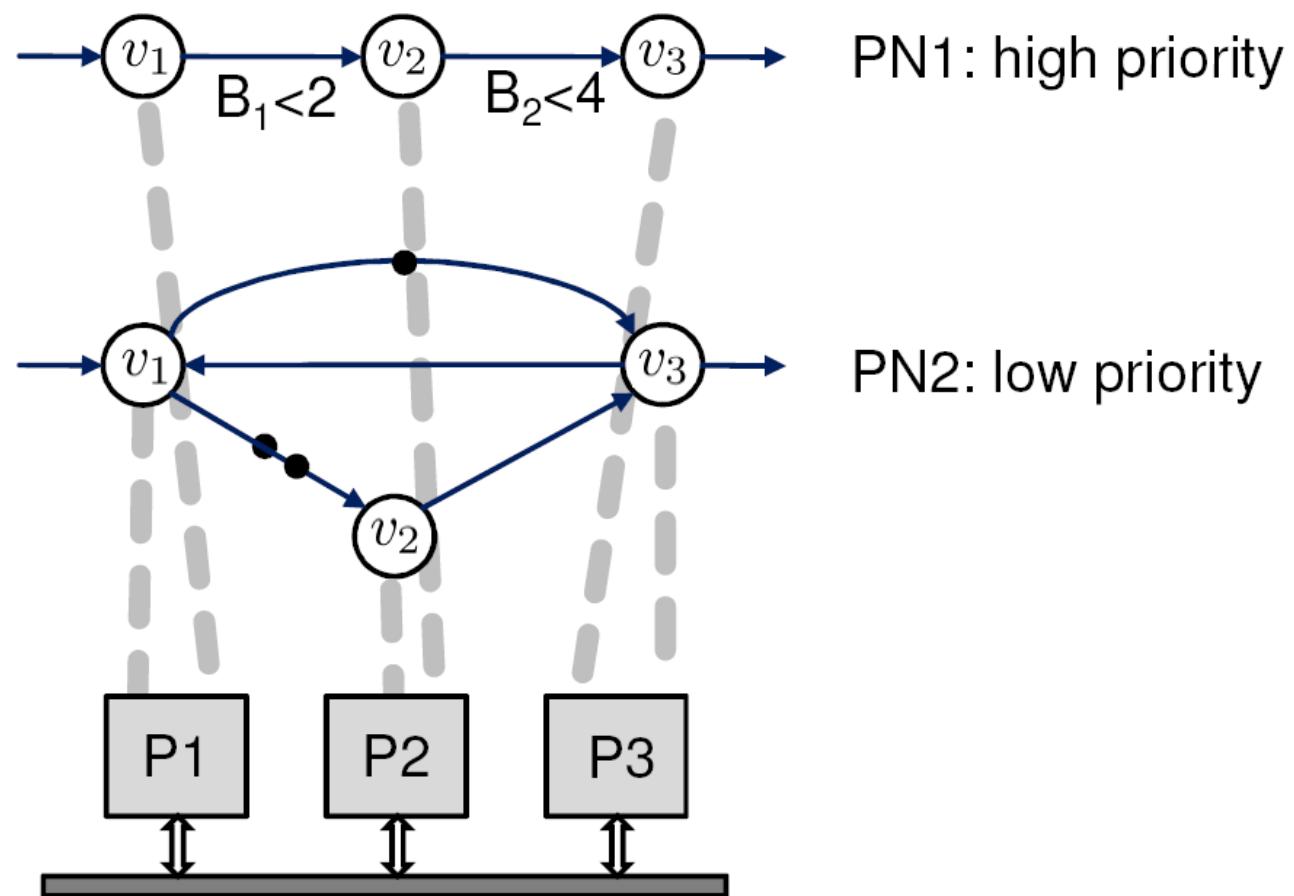
$\beta_j$

$m_{ij}$

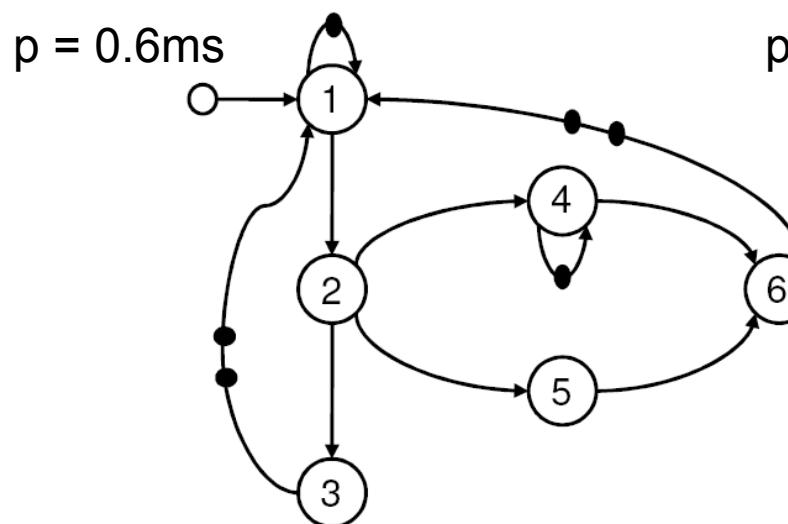
$S_{ji} = m_{ij} + \beta_j$

The diagram illustrates the derivation of system equations from a state transition graph. It starts with a state transition diagram showing three states and their transitions. This leads to two intermediate equations:  $R^u = \beta^u \wedge S^u \otimes R^u$  and  $R^l = \beta^l \wedge S^l \otimes R^l$ . A red oval highlights the term  $S^u$  in the first equation. Red arrows point from the text "smallest upper bound" to the first equation and from the text "largest lower bound" to the second equation. Further simplification leads to the final equations  $R^u = (S^u)^* \otimes \beta^u$  and  $R^l = (S^l)^* \otimes \beta^l$ . Below these, a red box contains a simplified model with a single state transition labeled  $\beta_j$  and  $m_{ij}$ , with the equation  $S_{ji} = m_{ij} + \beta_j$ .

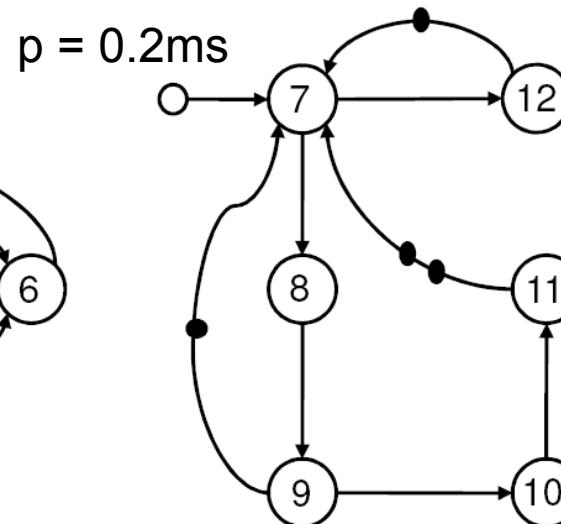
# abstract example



# SDR example



TD-SCDMA



WLAN

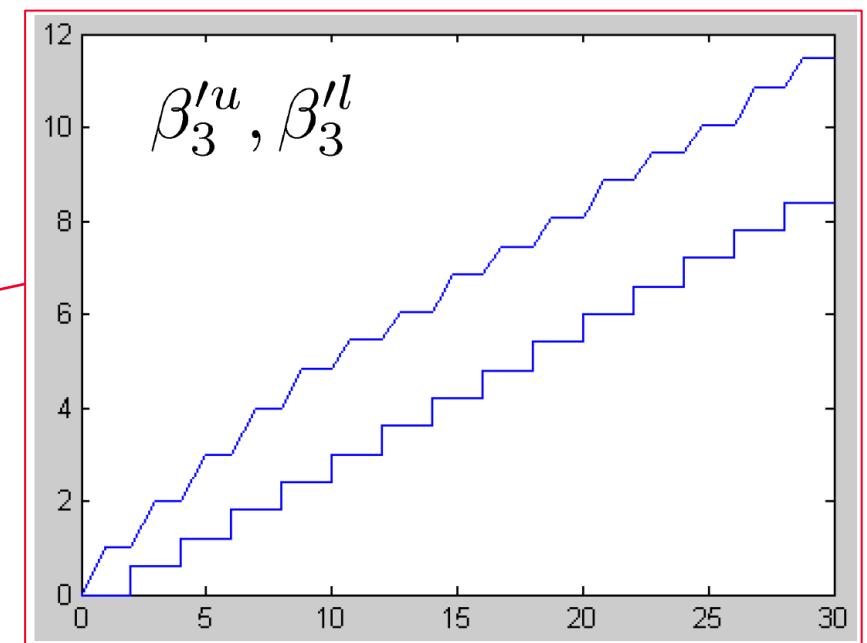
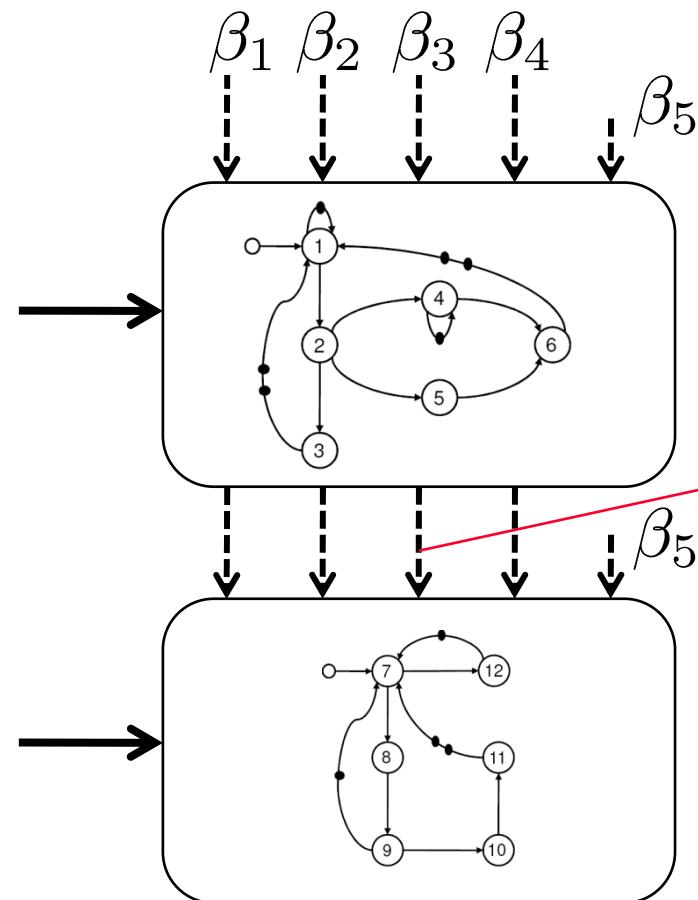
Fixed Priority

	1	2	3	4	5	6
cycl.	50k	12.5k	20k	3.3k	0.25k	50k
proc.	1	2	5.1	3	4	5.2

TDMA

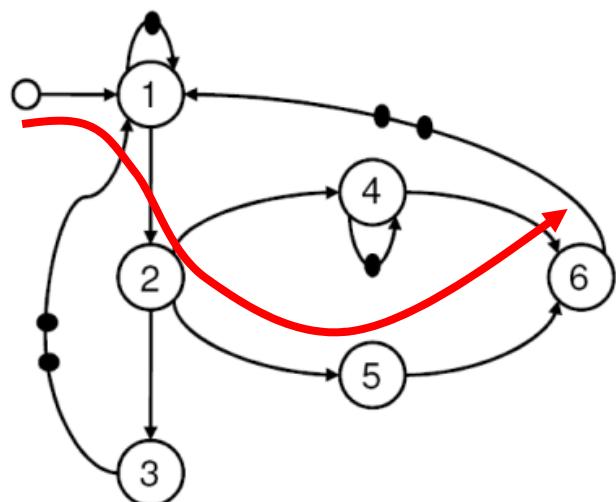
	7	8	9	10	11	12
cycl.	2k	0.31k	0.33k	0.42k	4k	2k
proc.	1	2	4	3	5.1	5.2

# example



# example

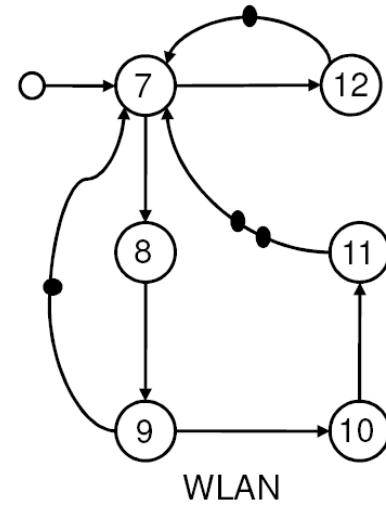
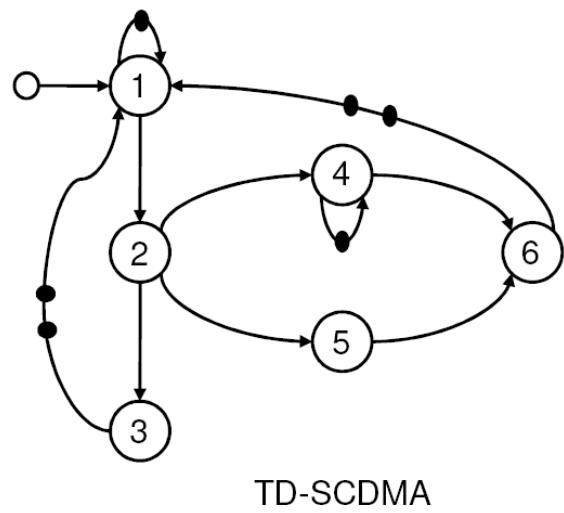
path delays



TD-SCDMA

node	delay [ms]
1	0.57
2	0.68
3	0.84
4	0.72
5	0.69
6	1.27

# example



node	delay [ms]
7	0.02
8	0.023
9	0.026
10	0.03
11	0.15
12	0.13

high priority

node	delay [ms]
7	24.5
8	31
9	34.5
10	41.3
11	44.7
12	48.2

low priority