Requirements for a Real-Time Mapping Flow based on Convex Programming

Maarten Wiggers
University of Twente, The Netherlands
Visiting scholar, UC Berkeley

Joint work with Marco Bekooij

Overview

- System organization
  - All synchronization on FIFO buffers
  - All arbitration through budget schedulers
- Convex program to find buffers and budgets
  - Satisfy real-time constraints
  - Polynomial complexity
- Approach based on
  - Model with only upper bounds on arrival times

Dataflow history

- Untimed dataflow – static scheduling
  - Edward Lee and David Messerschmitt
- Timed dataflow – static scheduling
  - S. Sriram and S. Bhattacharyya
- Timed dataflow – run-time scheduling
  - Subject current talk

Trends

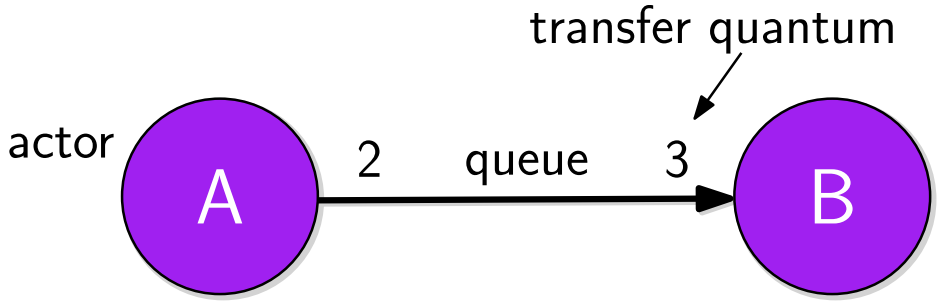
- Integration : multi-stream systems
  - User starts and stops streams
- Adaptation : data-dependent processing
  - Infrequent; e.g. modes in digital radios
  - Frequent; e.g. entropy encoding in MPEG

Trends

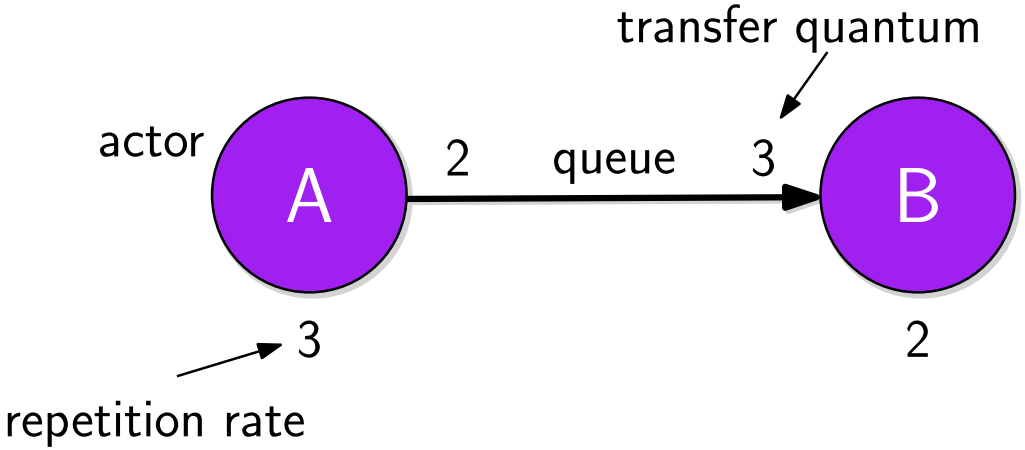
- Integration : multi-stream systems
  - User starts and s
- Adaptation : dat
  - Infrequent; e.g.
  - Frequent; e.g. e

Run-time scheduling increasingly attractive

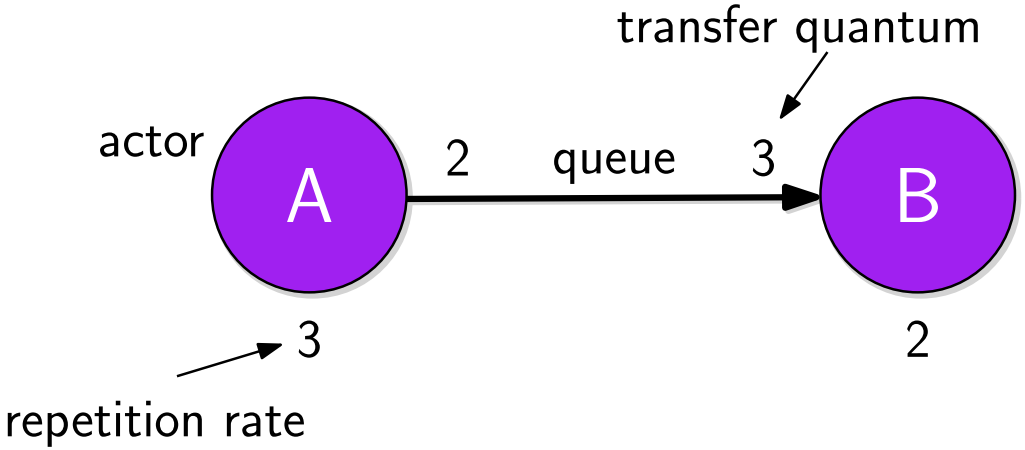
Untimed multi-rate dataflow



Untimed multi-rate dataflow

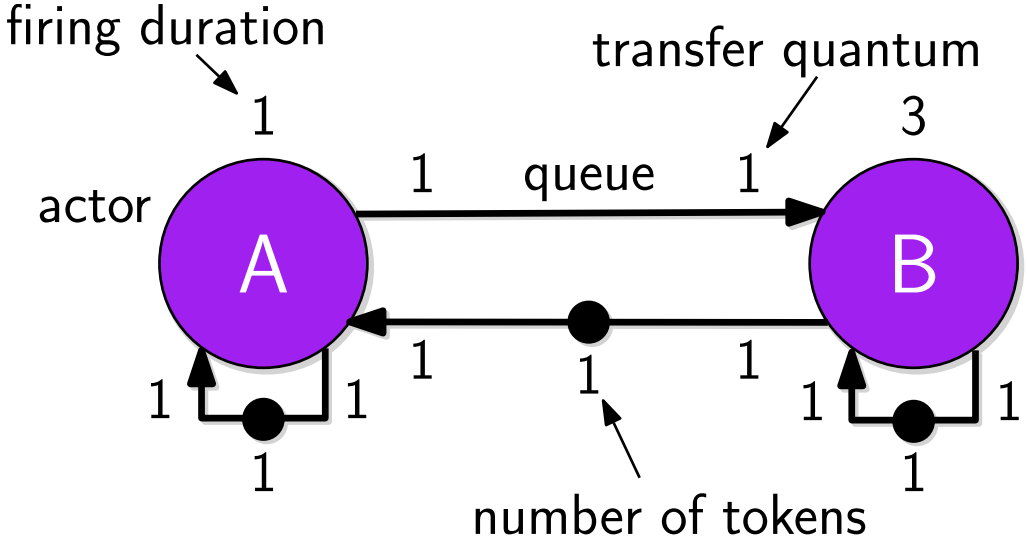


Untimed multi-rate dataflow



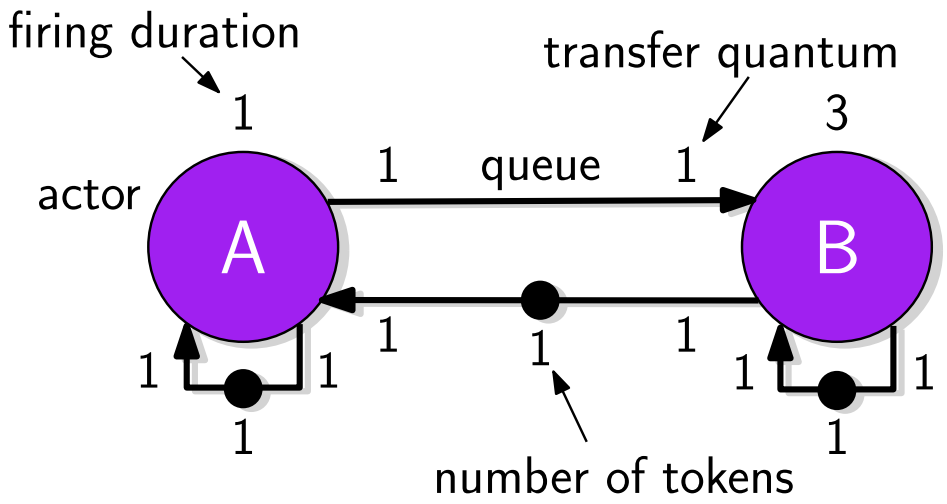
Example schedules: (AAABB)* or (AABAB)*

Timed single-rate dataflow



- Timed dataflow actor
  - Two atomic events: start and finish
  - Constant firing duration from start to finish
  - Consumes at start and produces at finish

Throughput - 1 location buffer



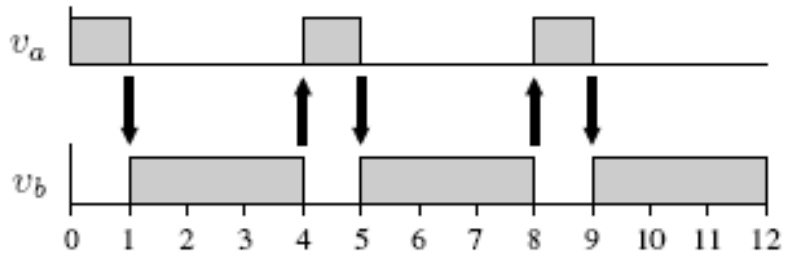
$$\frac{1}{\text{throughput}} = \text{MCR} = \max(\tfrac{1}{1}, \tfrac{3}{1}, \tfrac{1+3}{1}) = 4$$

$$\text{Maximum Cyle Ratio (MCR)} := \max_{c \in C}(\frac{\Sigma_c \text{ firing durations}}{\Sigma_c \text{ tokens}})$$

Periodic schedule

Periodic schedule
of dataflow graph

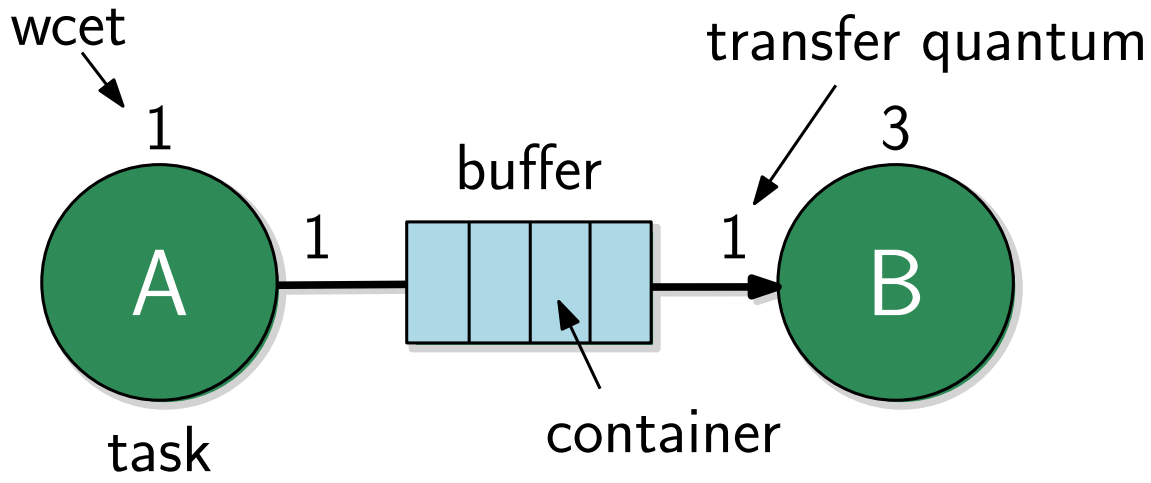
Period equal to MCR = 4



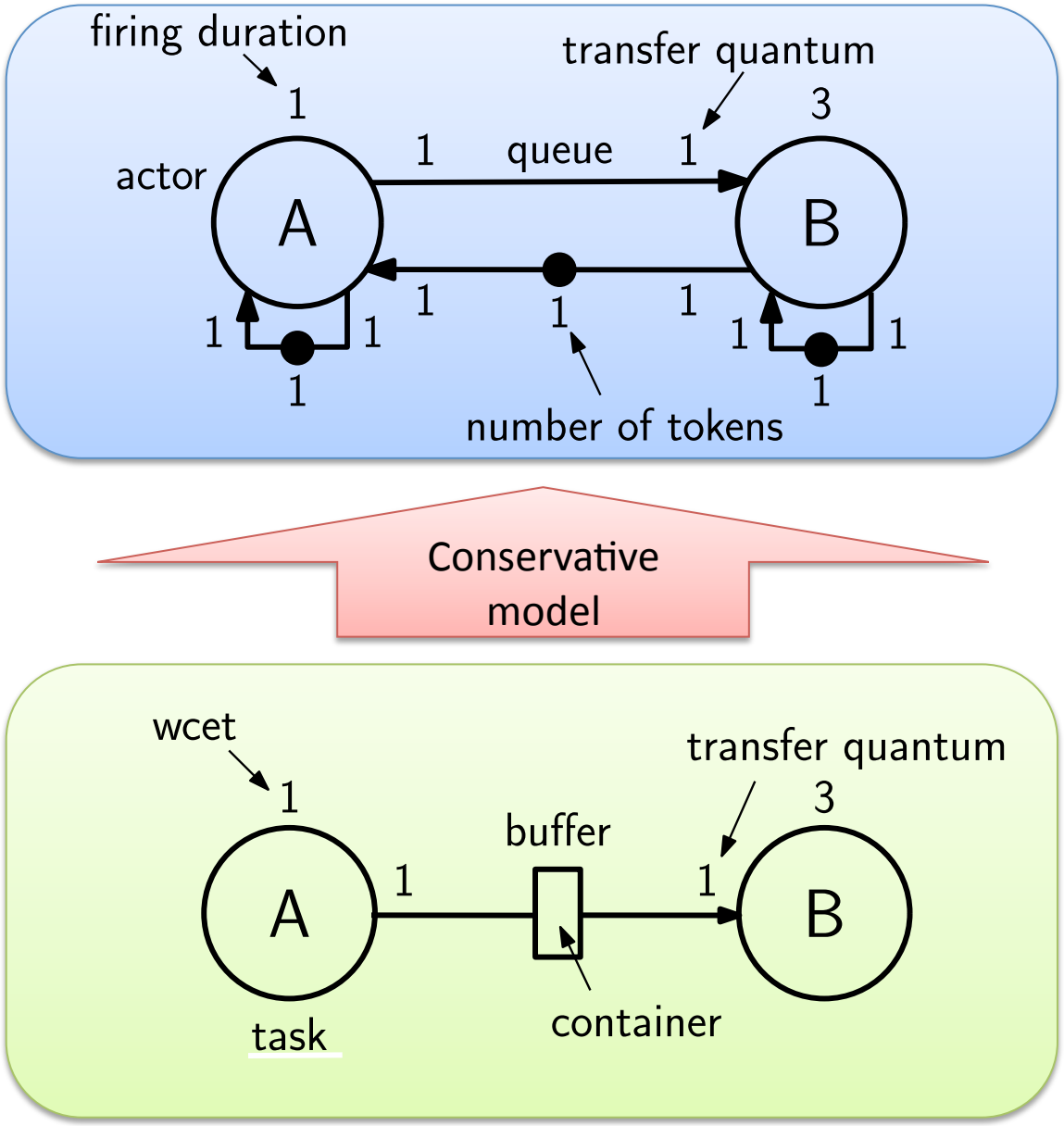
Next slides …

- Timed dataflow has constant firing durations
- Next slides will introduce task graphs
  - Task graphs are closer to final implementation
  - Timed dataflow conservatively models task graph

Task graph



- Task
  - Variable execution time (upper bound = WCET)
  - Produces and consumes during execution
- Buffer is a fixed capacity circular buffer


firing duration
1
transfer quantum
3
actor
A
1
queue
1
B
1
1
1
1
1
1
1
1
1
number of tokens
Conservative model
wcet
1
transfer quantum
3
buffer
A
1
1
B
task
container

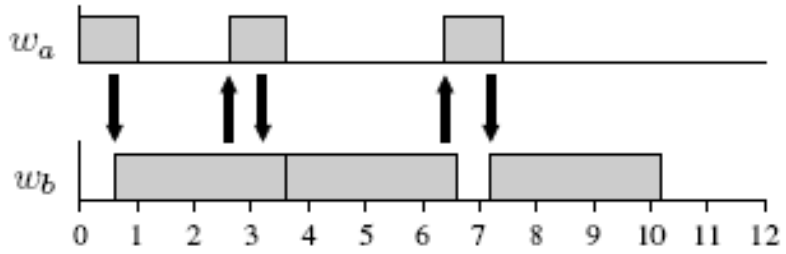
Conservative dataflow model

- 1-1 function: task graph to dataflow graph
  - From buffers to unordered pairs of queues
  - From containers to tokens

- Conservative iff container productions not later than corresponding token productions

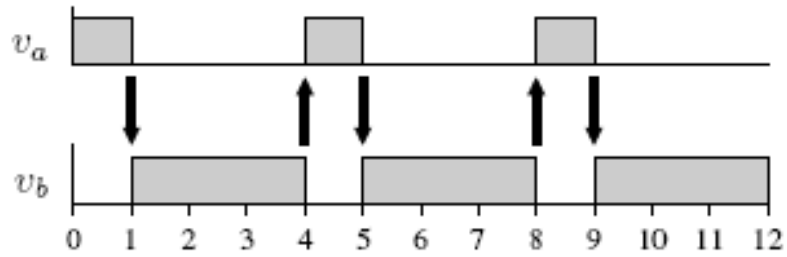
Conservative production times

Data-driven schedule
of task graph
(constant execution time)



Periodic schedule
of dataflow graph

Period equal to MCR = 4



Monotone production times

- Def.: Earlier production times cannot result in later production times
- Timed SRDF has monotone production times
  - Assumed every actor on private resource
- Preserves conservatism over compositions
- Explained in:
  - Wiggers, Bekooij, Smit. EMSOFT, 2009

Modeling schedulers in dataflow

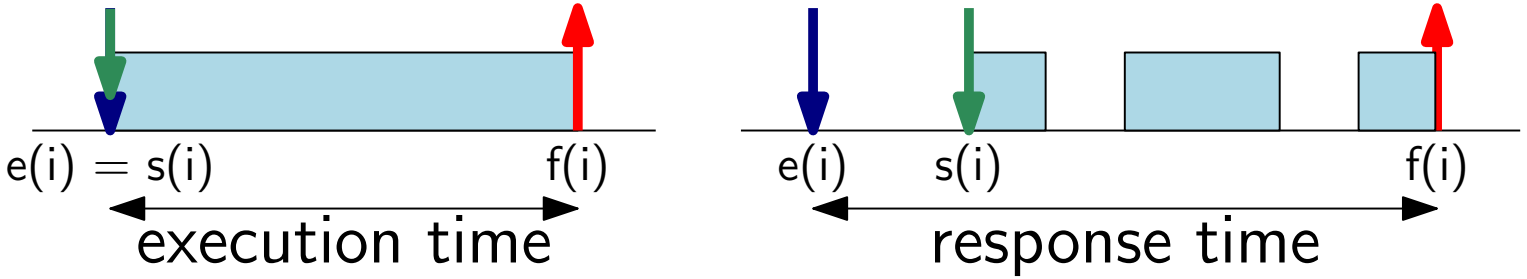
- Timed SRDF cannot express run-time schedulers
  - Run-time schedulers make decisions based on arrival time of data
- Timed non-determinate merge : no monotone productions
  - Earlier arrival -> re-order -> later production times

- Attractive : temporally monotone determinate model
  - Abstract from scheduling algorithm
  - Model worst-case relative finish time

- Resulting model has monotone arrival times
  - Bounds multiprocessor scheduling anomalies

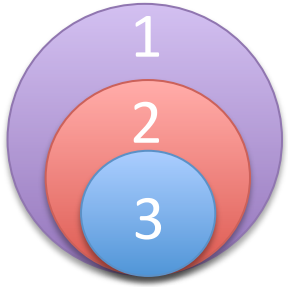
Response time

- Task execution i has
  - e(i) := enabling time: sufficient data and space present
  - s(i) := start time
  - f(i) := finish time
- Execution time := f(i) – e(i), in isolation
- Response time := f(i) – e(i), on shared resource



Three classes of schedulers

- Response time depends on
  - Own execution time
  - Interference : activation rate + execution time
- Classification of interference dependencies
  1. Activation rate & execution times
     - Static priority pre-emptive
  2. Execution times (starvation-free schedulers)
     - Round robin, weighted round robin
  3. Bounded by construction (budget schedulers)
     - Time division multiplex, priority-budget scheduling



Budget schedulers

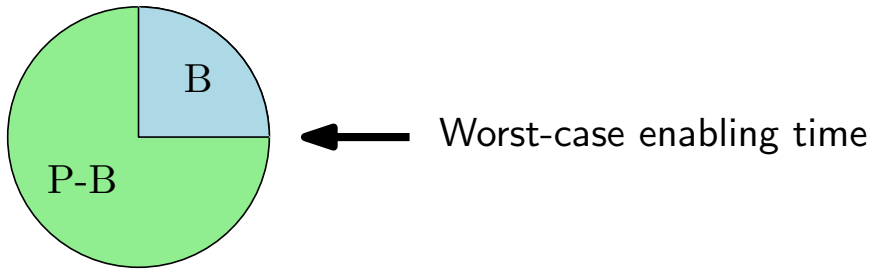
- Guarantees a task a minimum budget B in every time interval of length P
  - Subclass of the aperiodic servers
  - e.g., time-division multiplex, priority-based budget scheduler, polling server, constant-bandwidth server

Response time analysis

- Worst case response time for budget scheduler
  - B := budget
  - P := interval in which budget is guaranteed
  - x(i) := execution time of execution i

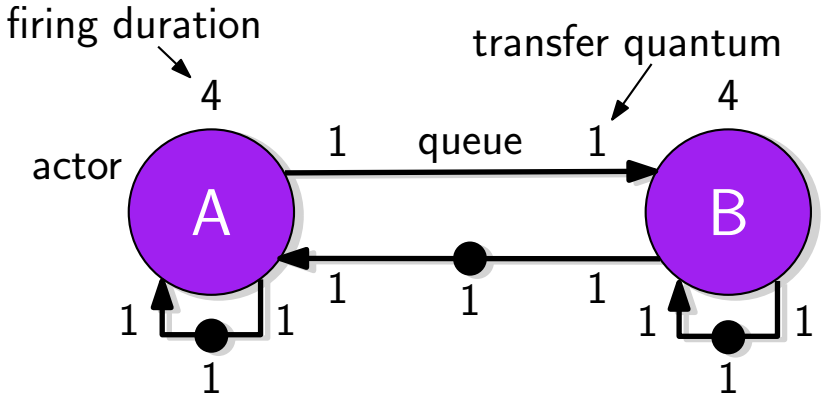
$$\max(f(i) - e(i)) = x(i) + (P - B) \cdot \lceil \frac{x(i)}{B} \rceil$$

Time-division multiplex



Example

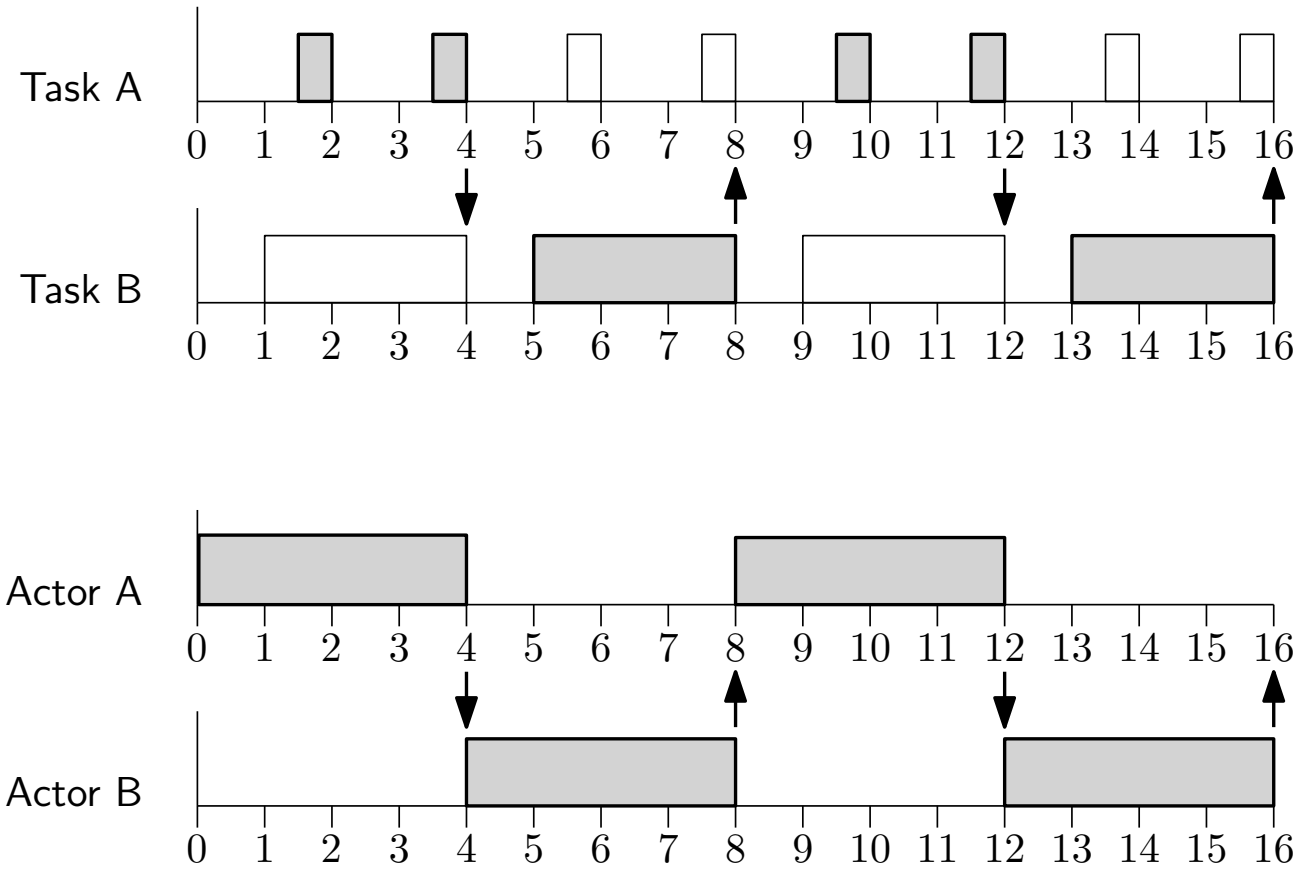
	Task A	Task B
Budget	1/2	3
Period	2	4
WCET	1	3
Worst-case response time	1 + (2-1/2) * 2 = 4	3 + (4-3) * 1 = 4



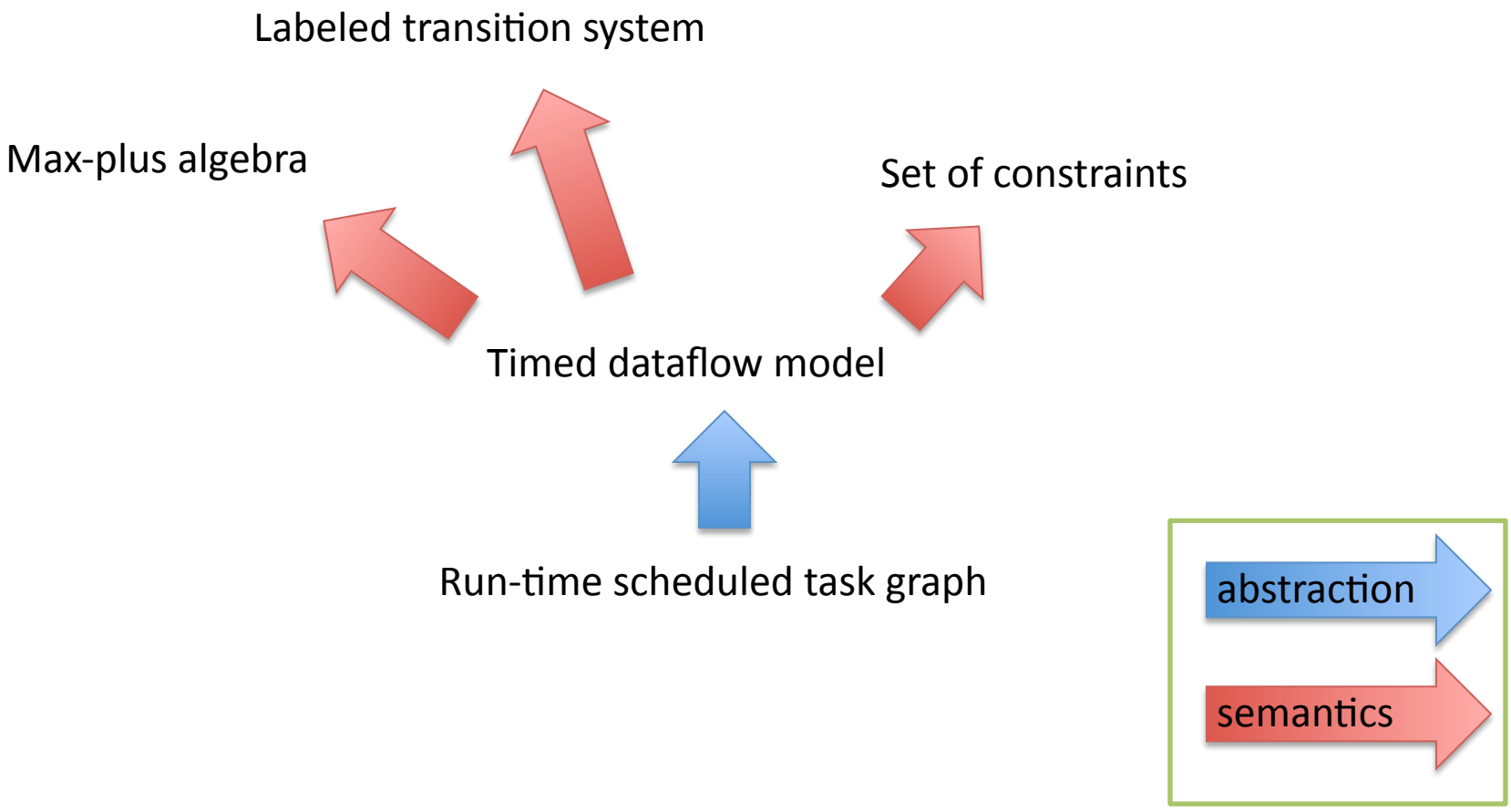
$$\frac{1}{\text{throughput}} = \text{MCR} = \max\left(\frac{4}{1}, \frac{4}{1}, \frac{4+4}{1}\right) = 8$$

MCR = Maximum Cycle Ratio

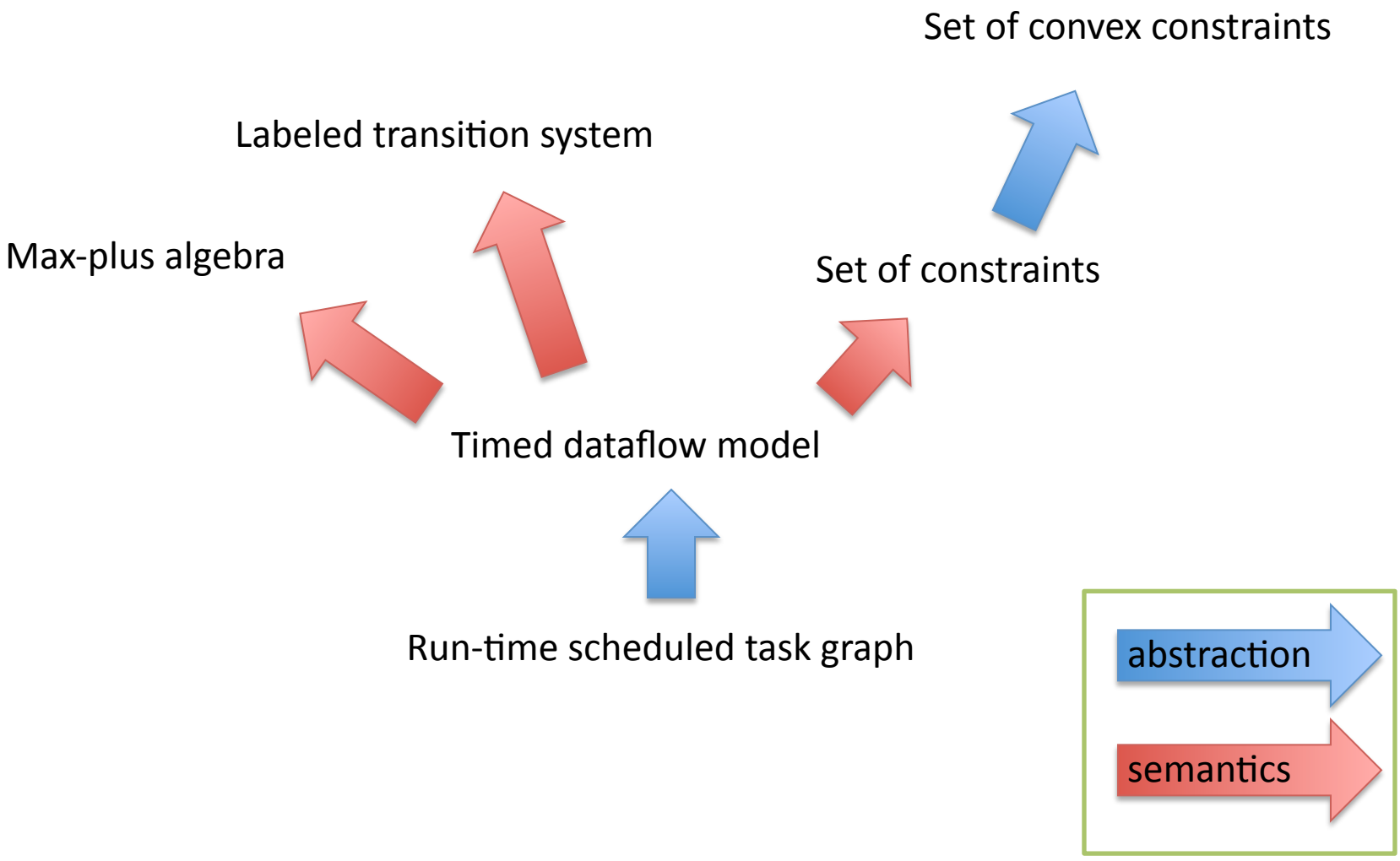
Schedules


Task A
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Task B
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Actor A
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Actor B
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

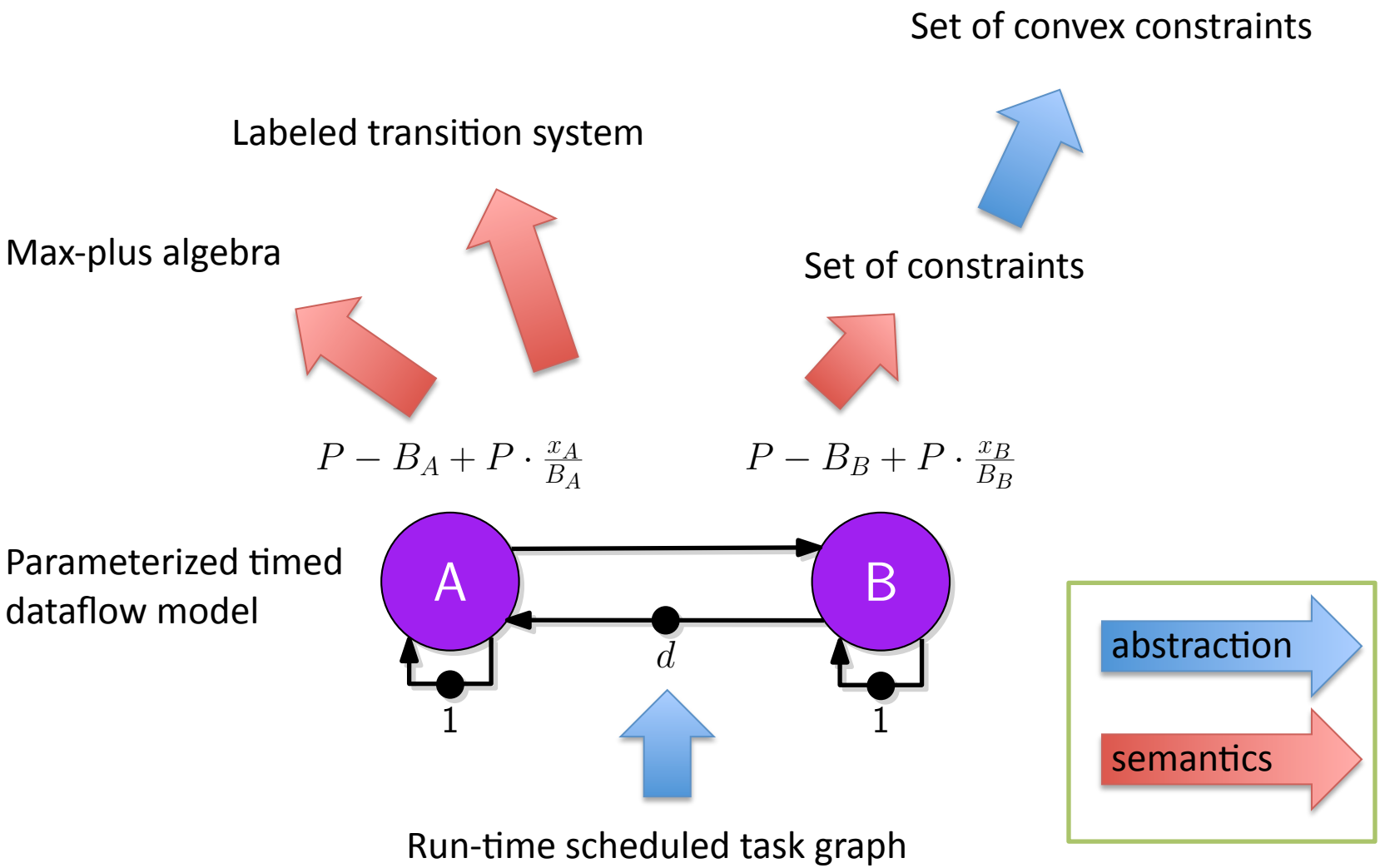
Timed dataflow modeling



Timed dataflow modeling



Timed dataflow modeling



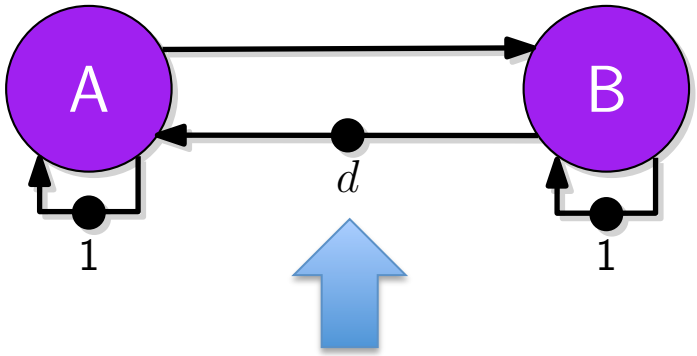
Timed dataflow modeling

Set of convex constraints

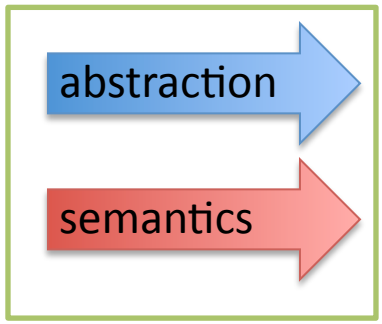
Set of constraints

$P - B_A + P \cdot \frac{x_A}{B_A}$ $\quad$ $P - B_B + P \cdot \frac{x_B}{B_B}$

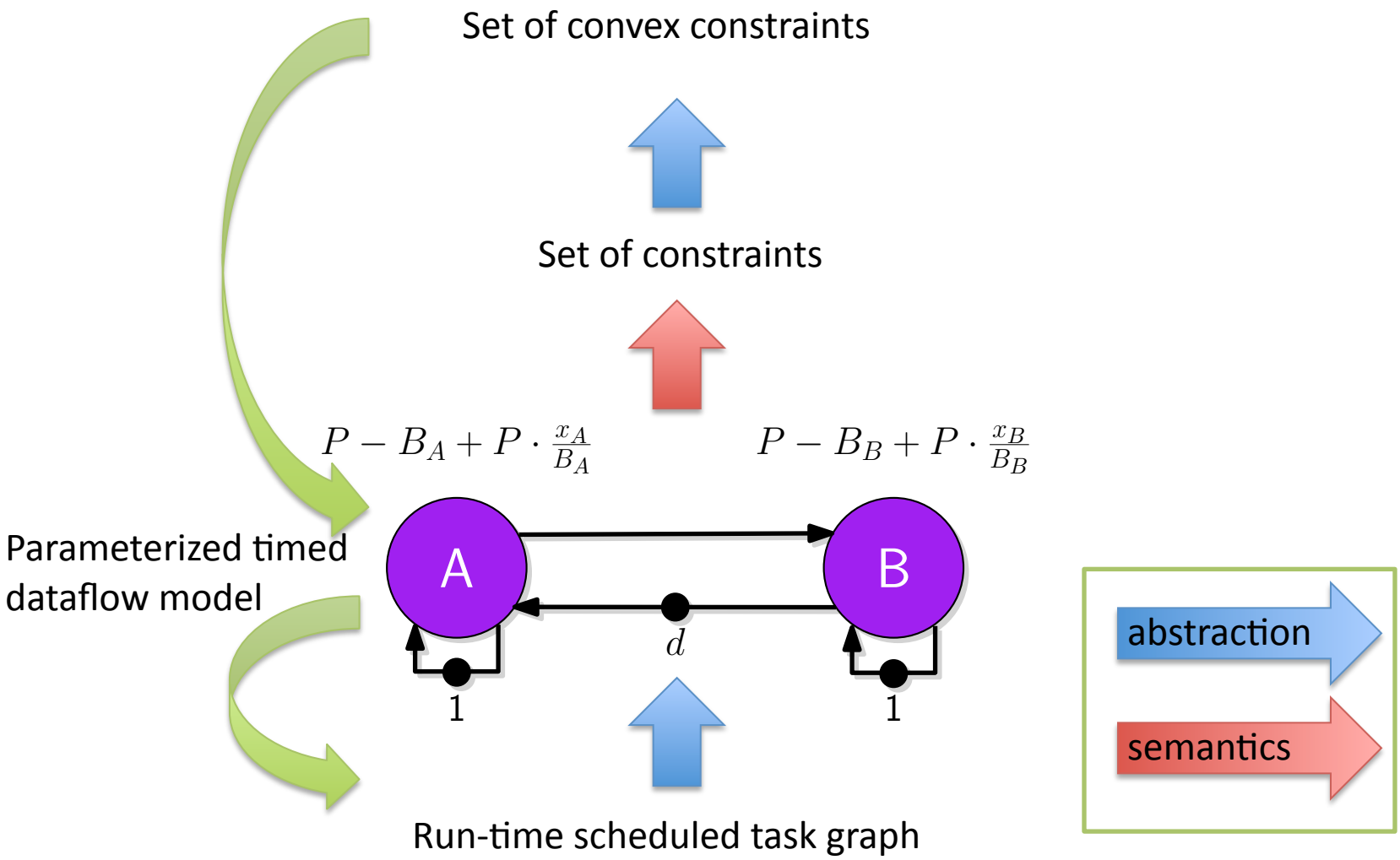
Parameterized timed dataflow model



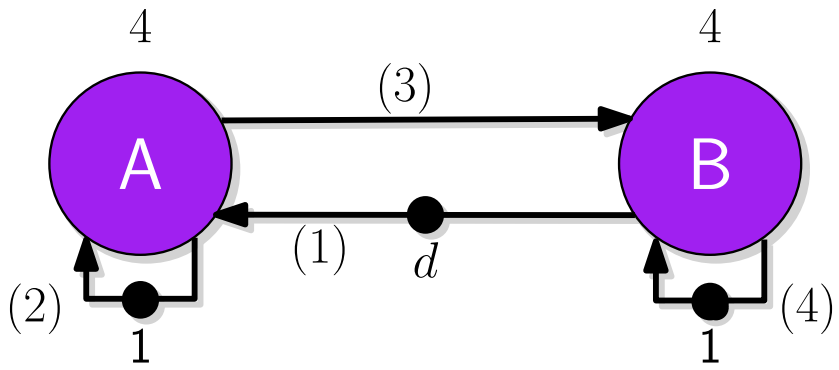
Run-time scheduled task graph



Timed dataflow synthesis



Linear program: find buffer size



$$(1):\ s(A) \geq s(B) + 4 - d \cdot \mu$$

$$(2):\ s(A) \geq s(A) + 4 - \mu$$

$$(3):\ s(B) \geq s(A) + 4$$

$$(4):\ s(B) \geq s(B) + 4 - \mu$$

Objective: $\min d$

μ is desired MCR (throughput)

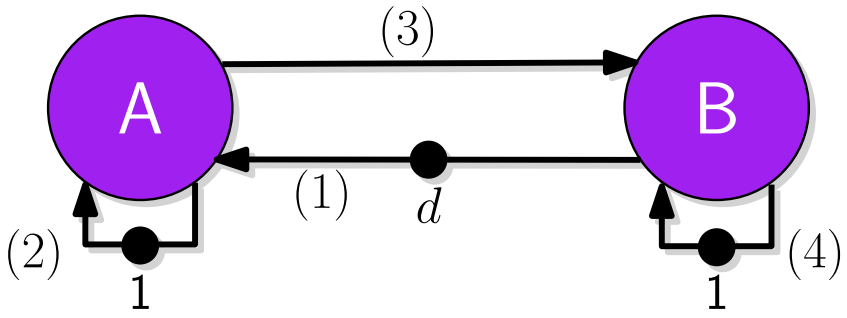
Monotonicity of model tells rounding to next integer is allowed

Convex programming

- Generalization of linear programming
- Constraints specify a convex space instead of polyhedron
- Efficiently solvable (polynomial complexity)

Convex program: buffers & budgets

$P - B_A + P \cdot \frac{x_A}{B_A}$ $\qquad$ $P - B_B + P \cdot \frac{x_B}{B_B}$



$$(1): \; s(A) \geq s(B) + P - B_B + P \cdot \frac{x_B}{B_B} - d \cdot \mu$$

$$(2): \; s(A) \geq s(A) + P - B_A + P \cdot \frac{x_A}{B_A} - \mu$$

$$(3): \; s(B) \geq s(A) + P - B_A + P \cdot \frac{x_A}{B_A}$$

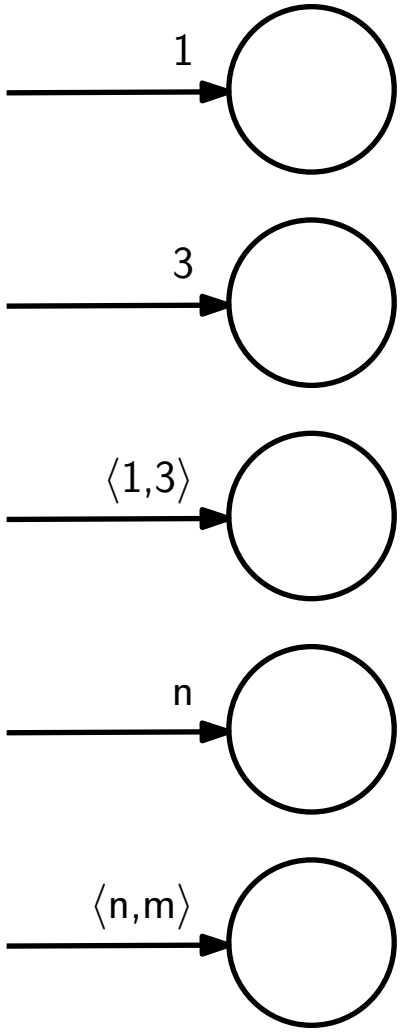
$$(4): \; s(B) \geq s(B) + P - B_B + P \cdot \frac{x_B}{B_B} - \mu$$

$$(5): \; P \geq B_A + B_B$$

Objective: $\min \alpha_A B_A + \alpha_B B_B + \beta d$

Monotonicity of model tells rounding to next integer is allowed

More expressive models



Single-rate dataflow

Multi-rate dataflow, Synchronous Dataflow

Cyclo-static dataflow

Variable-rate dataflow

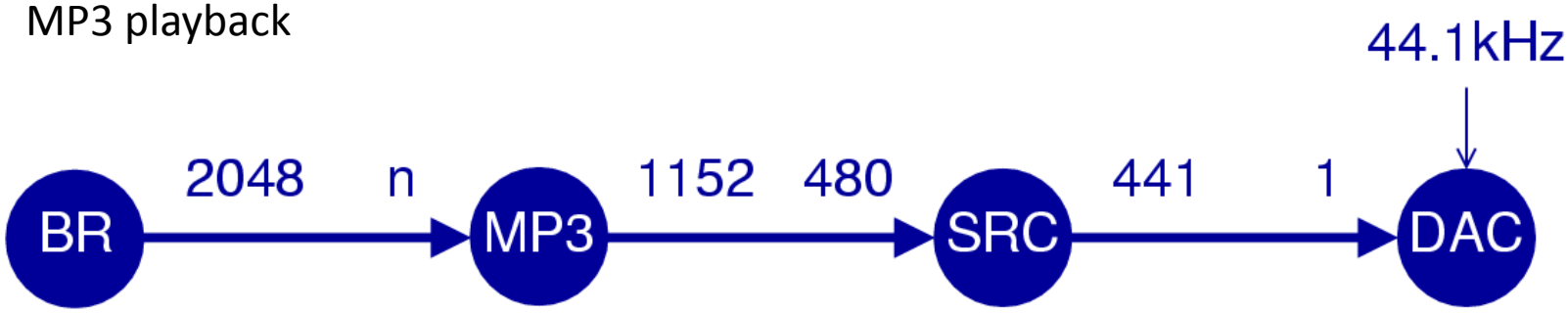
Variable-rate phased dataflow

More expressive models

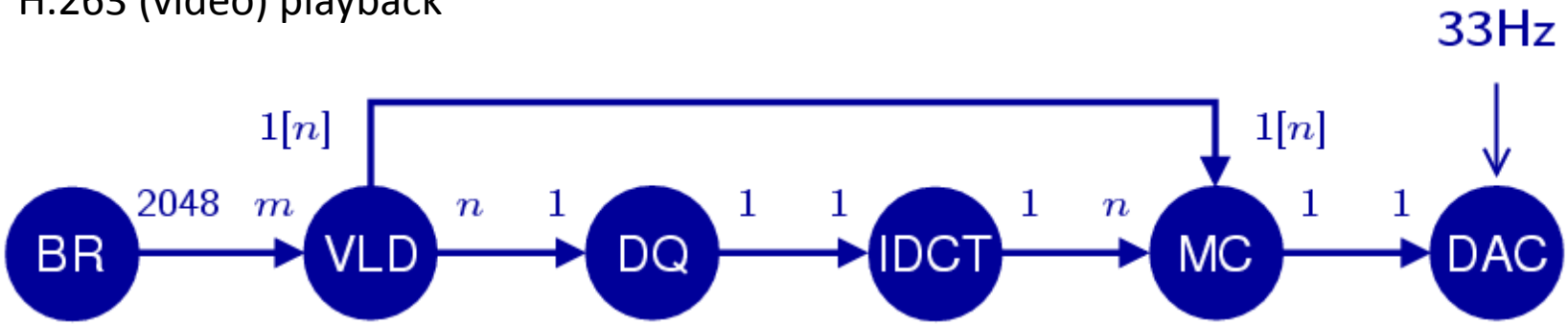
- More expressive dependency relations
  - Not every producer firing causes consumer firing
  - Dependency relation is no longer a convex expression
- Conservative linear abstractions of dependencies
  - Linear bounds on token transfer times
  - Allows efficient computation of budgets and buffers

Example applications

MP3 playback



H.263 (video) playback



Conclusion

- Run-time scheduling attractive
  - Multi-application systems
  - Data-dependent processing
- Modeling run-time scheduling with timed dataflow : restrict to budget schedulers
- Results in temporally monotone dataflow model
  - LP/SOCP formulations for resource allocations
- Budget schedulers allow for efficient resource allocation: can this class be increased?

Questions?

m.h.wiggers@utwente.nl