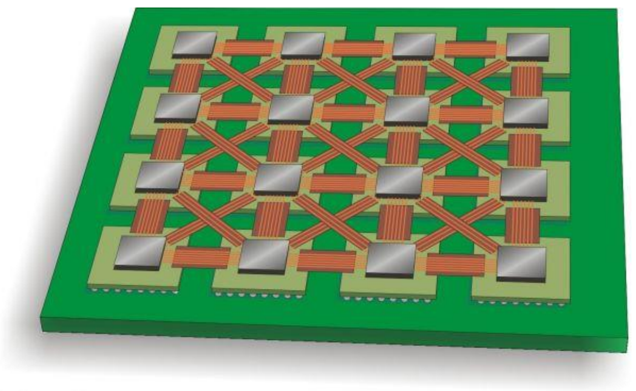
Response Time Analysis for G-EDF and G-DM Scheduling of Sporadic DAG-Tasks with Arbitrary Deadline

Andrea Parri, **Alessandro Biondi** and Mauro Marinoni

Scuola Superiore Sant'Anna – Pisa, Italy

Introduction

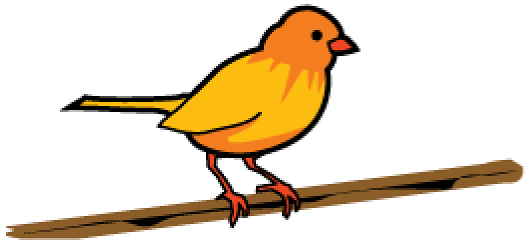
Multicore revolution



- New parallel programming models for expressing parallel computational activities

OpenMP

OpenCL



Intel TBB

Introduction

Big Data

- Novel programming models based on the Map-Reduce paradigm that relies on parallel processing

Introduction

JUNIPER EU Project – supported this work

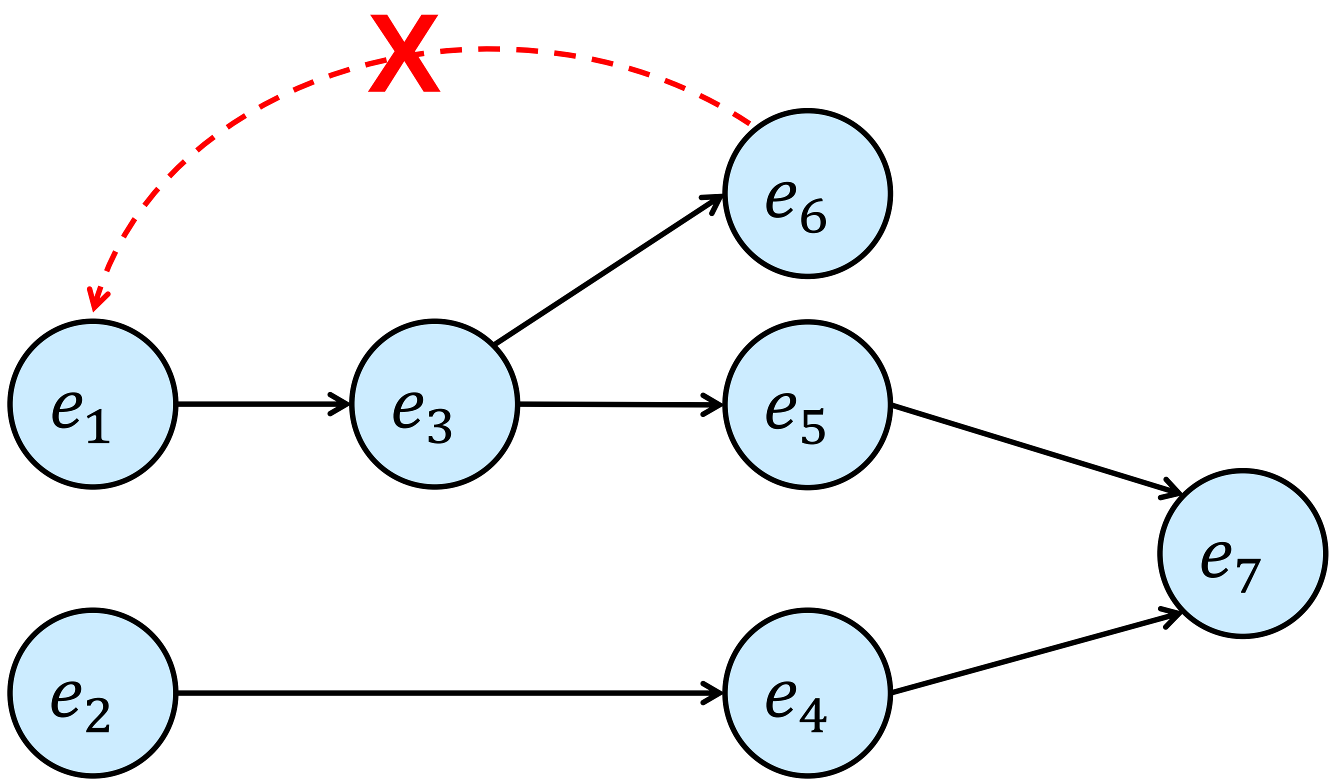
- **Goal**: enable application development with performance guarantees required for real-time exploitation of large streaming data sources and stored data;

- Case-study: applications for credit cards.

JAVA PLATFORM FOR HIGH-PERFORMANCE AND REAL-TIME LARGE SCALE DATA

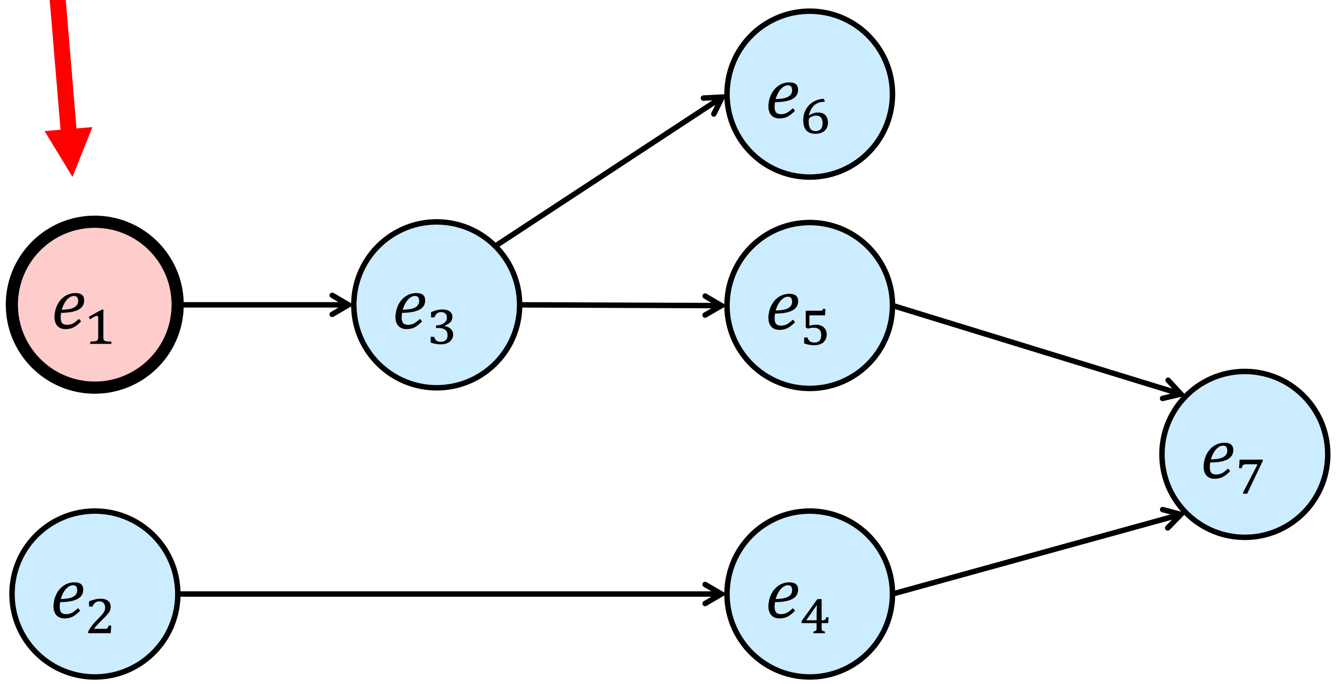
DAG-Task

- Task model for expressing parallel computations with precedence constraints
- A task is described with a Directed Acyclic Graph (DAG)



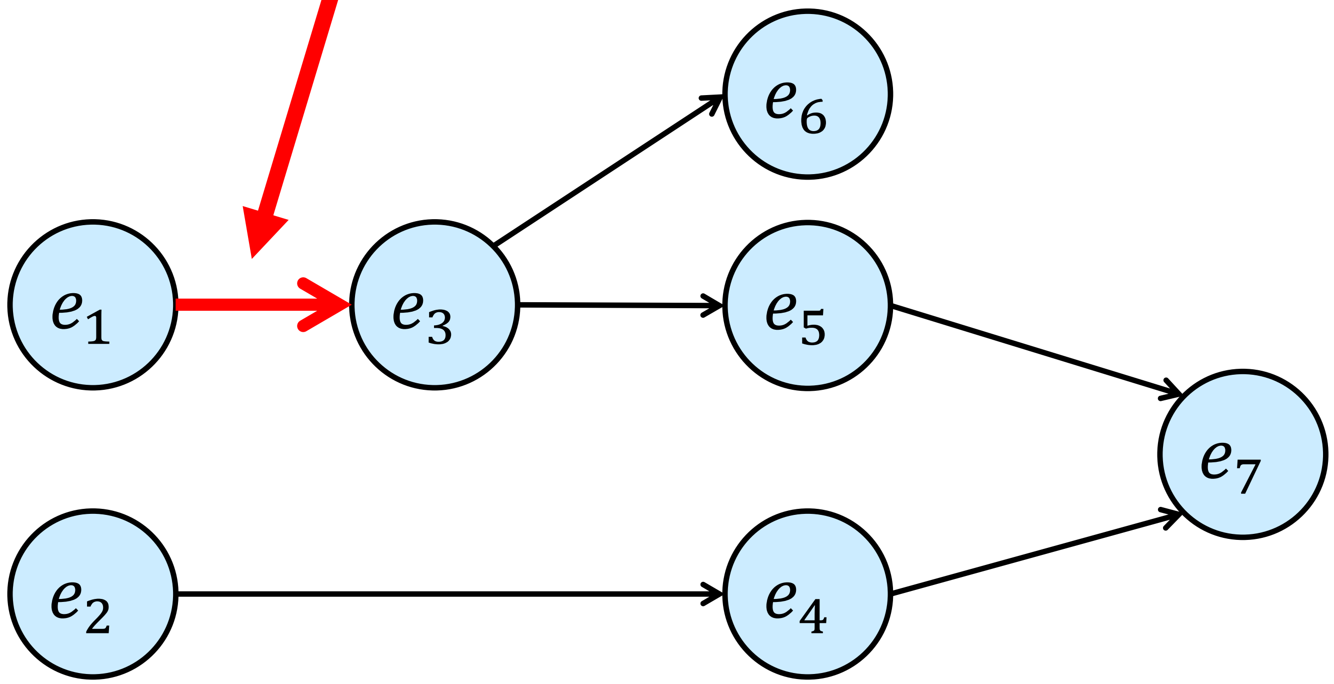
DAG-Task

Vertex – sequential computation with WCET e_i



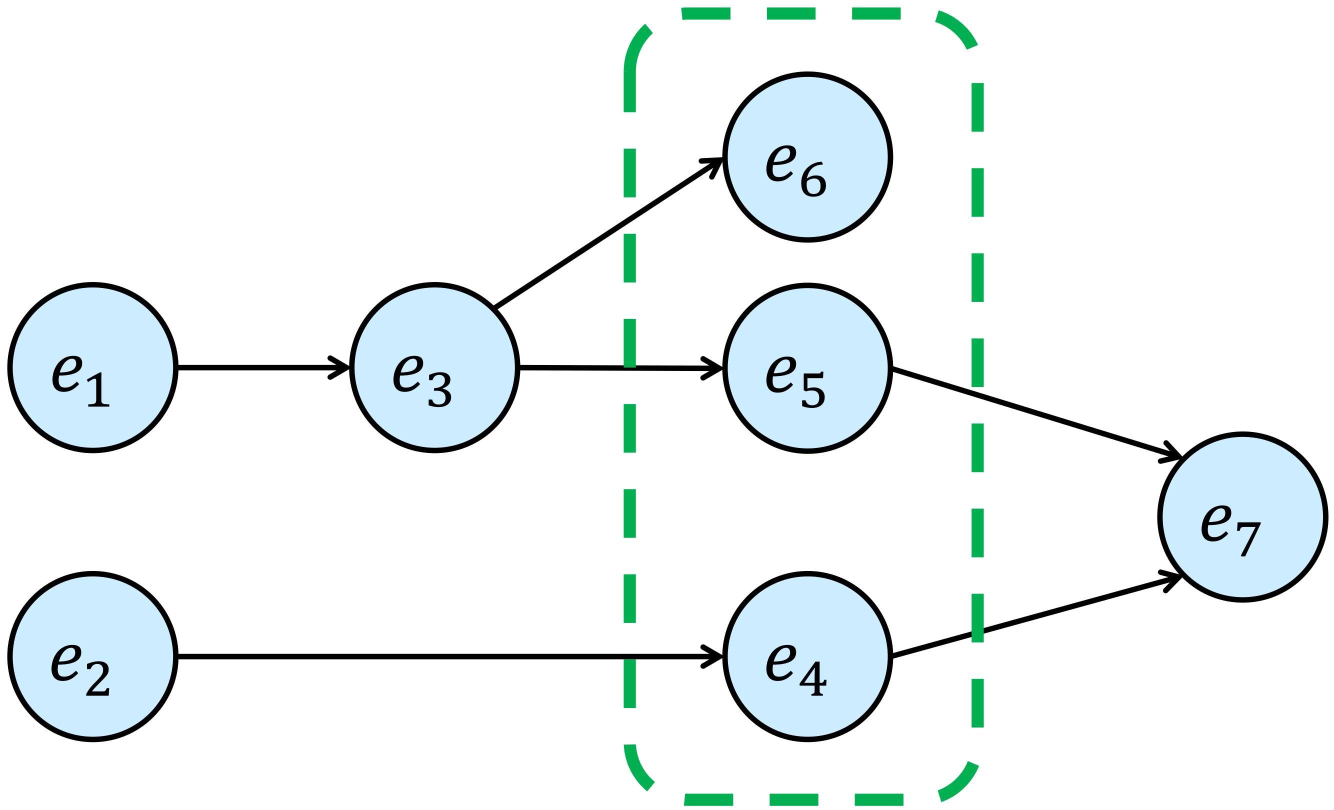
DAG-Task

Edge – *precedence constraint* among two computational activities



DAG-Task

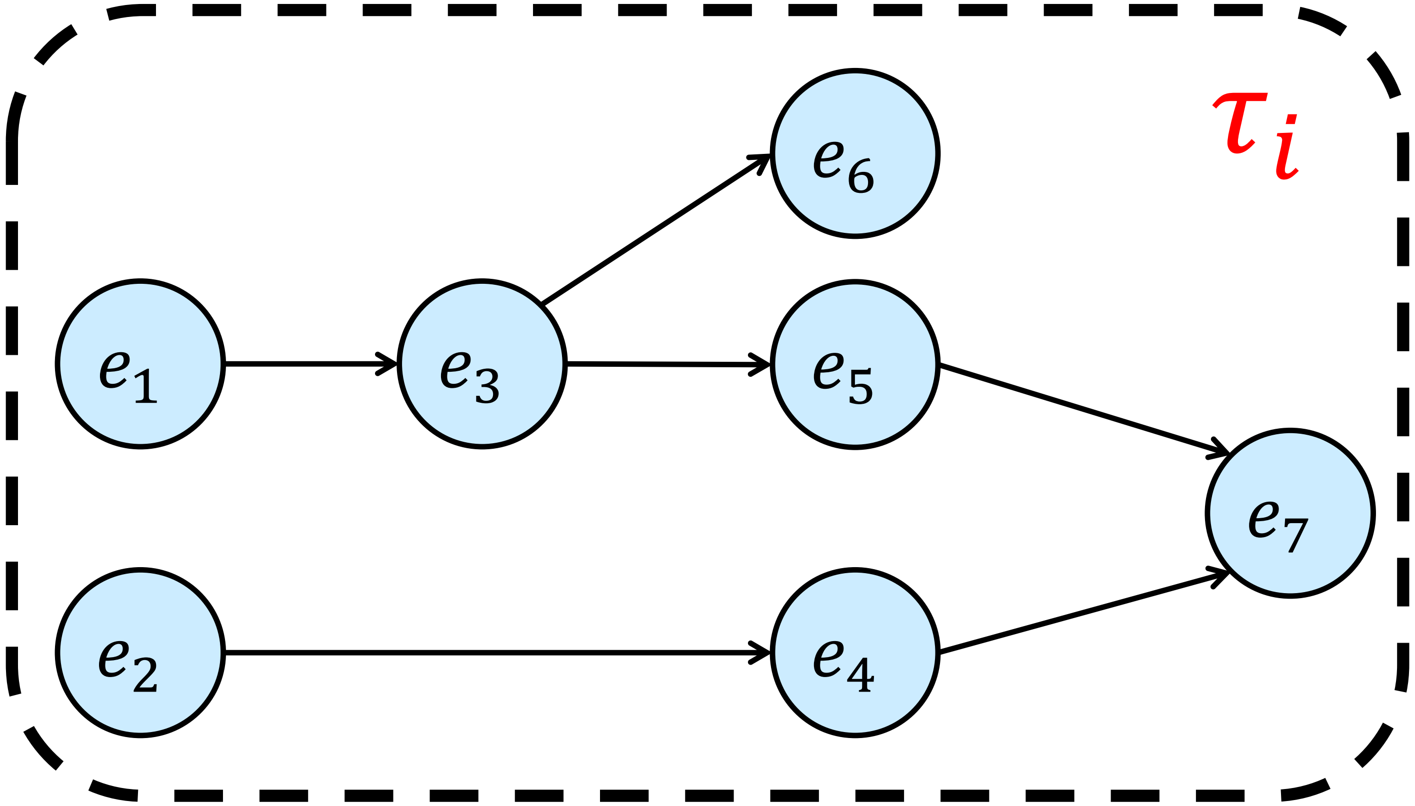
Note: this model allows to express *parallelism*



DAG-Task

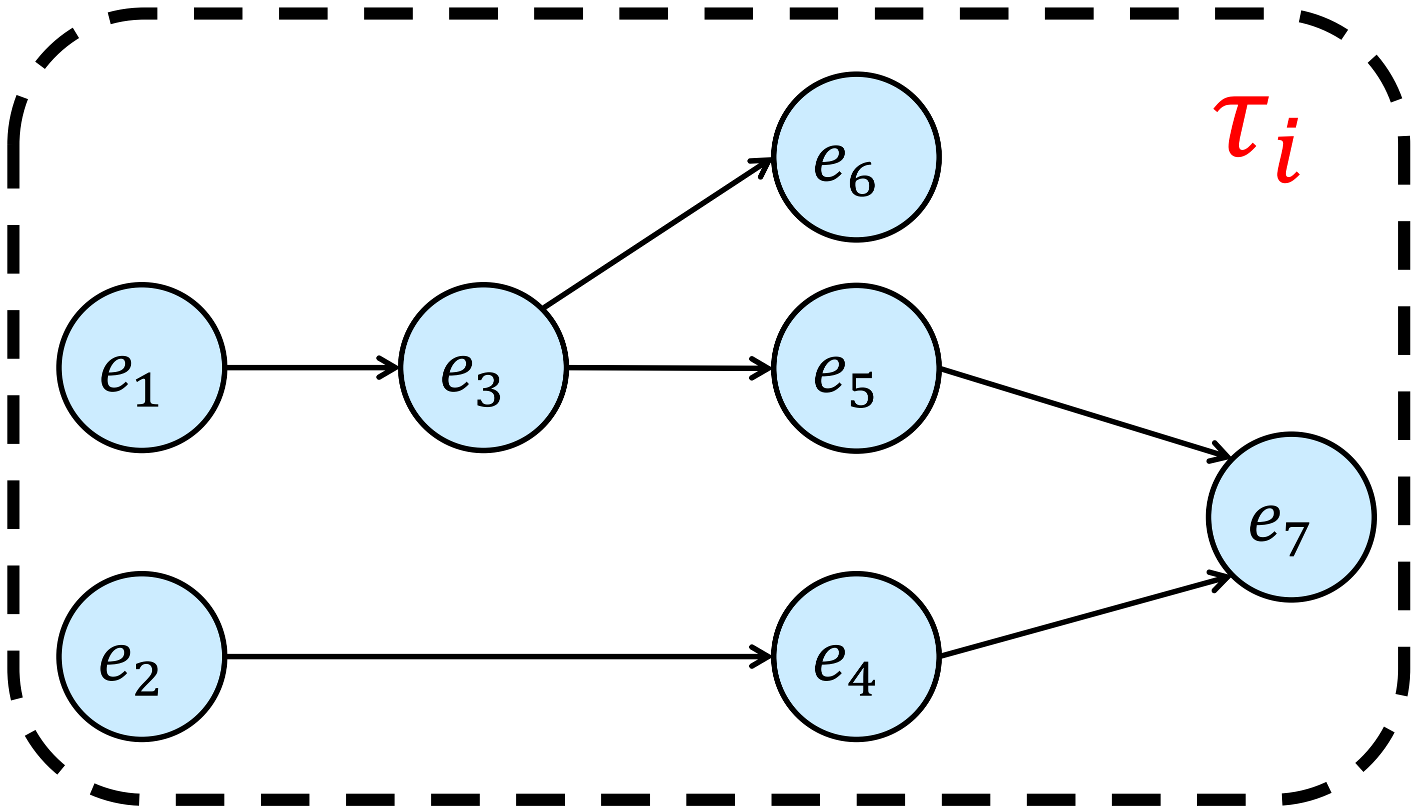
Release of a DAG-Task τ_i

- All the vertices are released simultaneously but it can be that not all of them are *enabled* due to precedence constrains

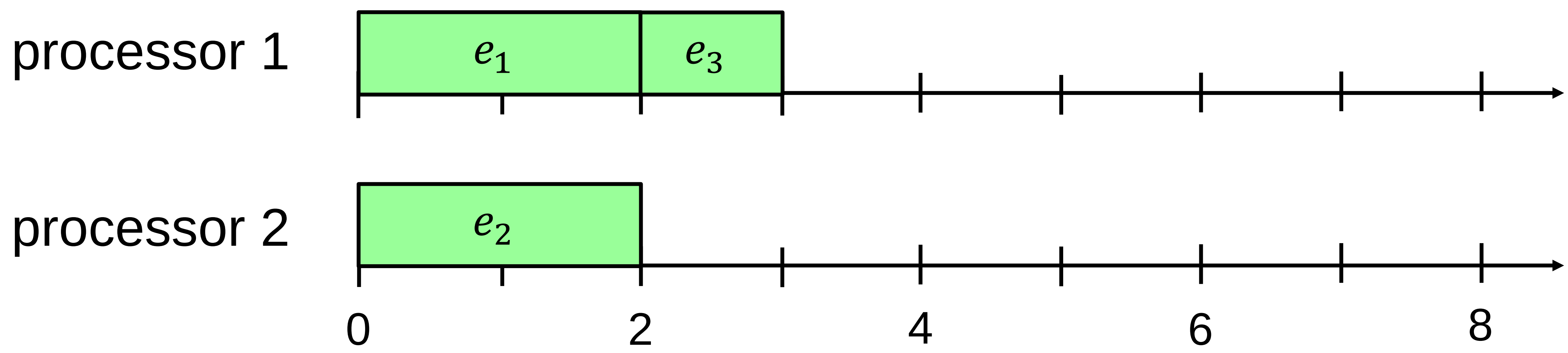


Sporadic DAG-Task

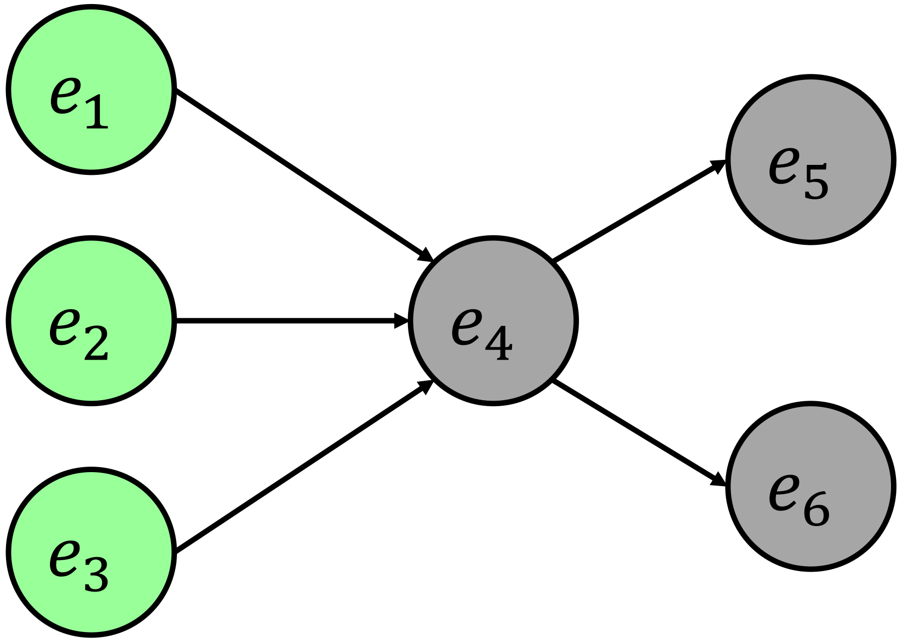
- DAG-Task τ_i
  - Released with a minimum inter-arrival time $\boldsymbol{T_i}$
  - Each vertex must complete within a deadline $\boldsymbol{D_i}$



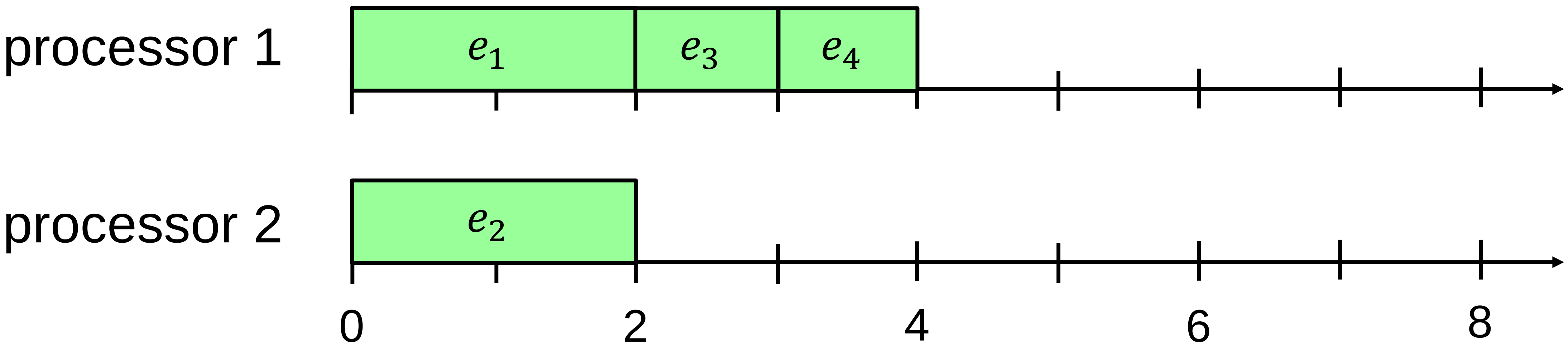
Example



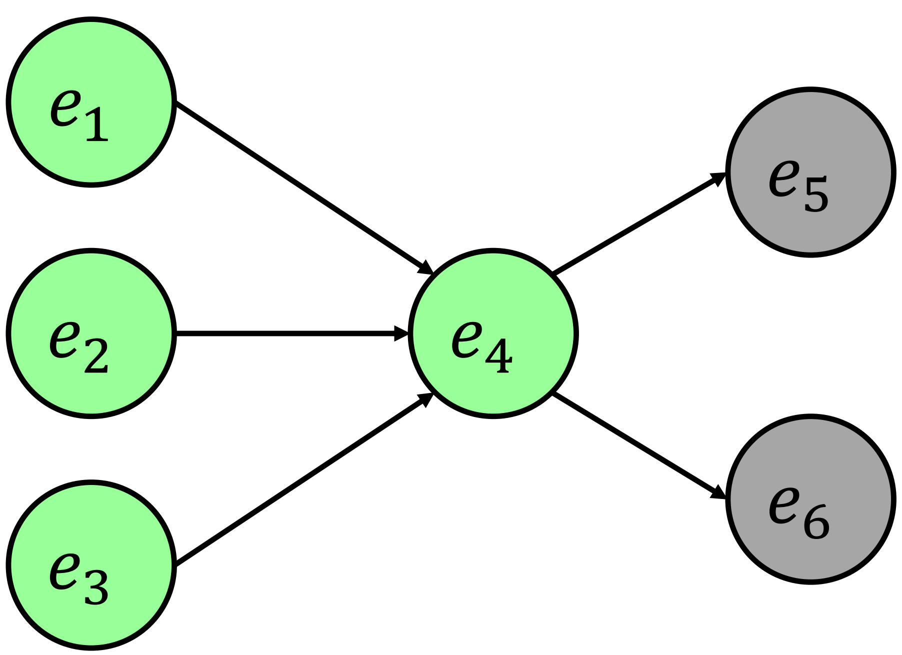
e_1	2
e_2	2
e_3	1
e_4	1
e_5	2
e_6	3



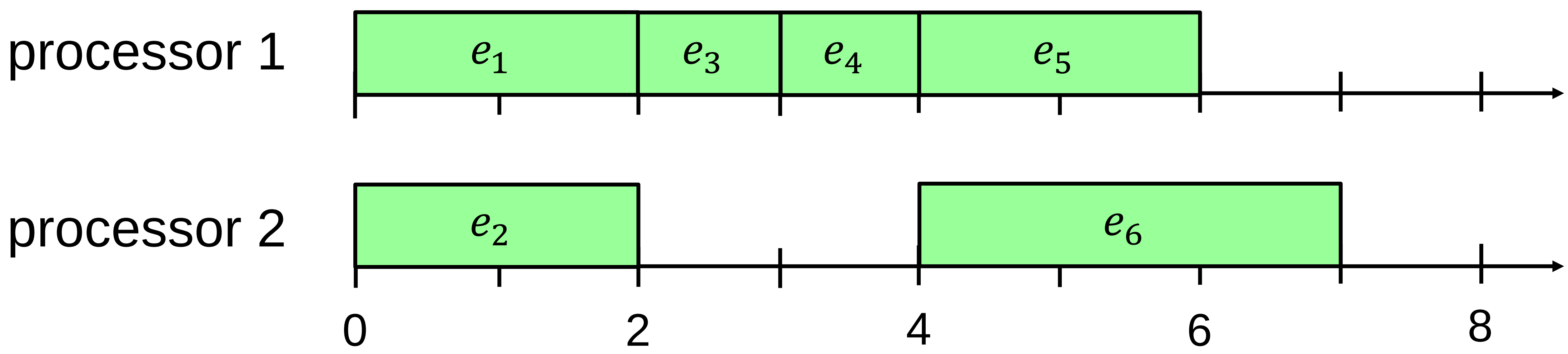
Example



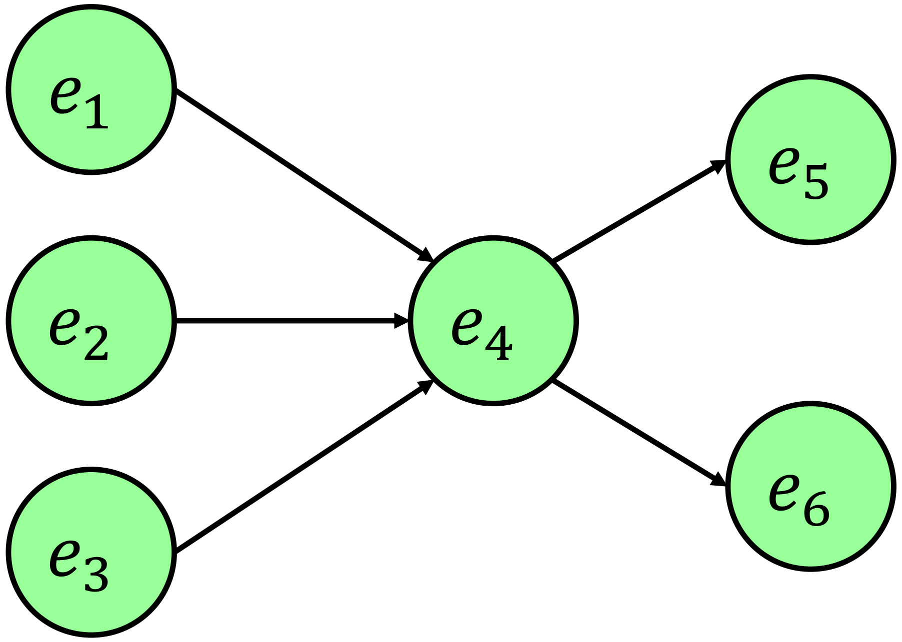
e_1	2
e_2	2
e_3	1
e_4	1
e_5	2
e_6	3



Example



e_1	2
e_2	2
e_3	1
e_4	1
e_5	2
e_6	3



Scheduling Problem

Given

- a set of N sporadic DAG-Tasks;
- A scheduling algorithm (G-EDF or G-DM);
- A platform with m identical processors;

verify if all deadlines are guaranteed.

State of The Art

Existing schedulability analysis can be split in 3 categories:

- Based on resource augmentation (speed-up);

  (Baruah et al., Bonifaci et al., Nilissen et al.,...)

- Based on capacity augmentation;

  (Kim et al., Li et al., Lakshmanan et al., ...)

- Based on Response-Time Analysis.

  (Maia et al., Chwa et al., Melani et al., ...)

This Work

- **Response-Time Analysis** of Sporadic DAG-Tasks under both **G-EDF** and **G-DM**

Contribution w.r.t. the state of the art:

- Vertices-oriented analysis;
- Tasks can have *arbitrary deadlines*;
- Vertices can have *arbitrary utilization*;
- Augmentation bounds proved for N=1.

Response-Time Analysis

- For each DAG-Task τ_i,
  - For each vertex v of τ_i,
    - Each job of vertex v must complete within a deadline $\boldsymbol{D_i}$

$$e_v + I_v = \boldsymbol{R_v} \leq D_i$$

Vertex WCET

Worst-case scheduling interference

Response-Time Analysis

- For each DAG-Task τ_i,
  - For each vertex v of τ_i,
    - Each job of vertex v must complete within a deadline $\boldsymbol{D_i}$

$$e_v + I_v = \boldsymbol{R_v} \leq D_i$$

Not easy to compute
for multiprocessor systems!

Response-Time Analysis

- **Our approach**: compute an upper-bound $\overline{I_v}$ of the interference I_v specific for each vertex v, so obtaining a response-time upper-bound $\overline{\boldsymbol{R}_v}$

$$e_v + I_v = \boldsymbol{R}_v \le$$

$$e_v + \overline{I_v} \Rightarrow \boxed{R_v \le \overline{\boldsymbol{R}_v}}$$

Response-Time Analysis

- **Main result** of this work: we proved that

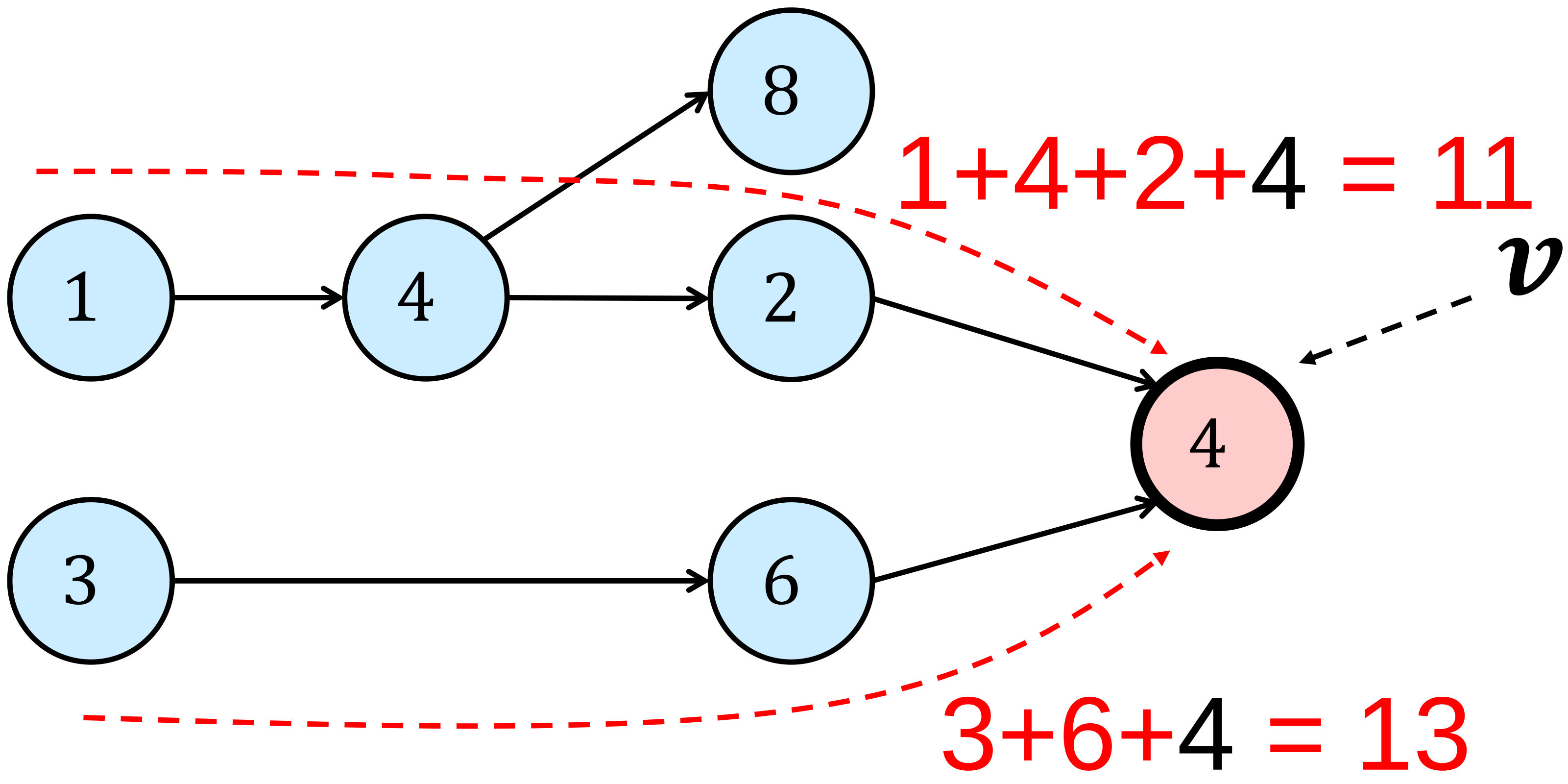
$$R_v \leq \overline{\boldsymbol{R_v}}$$

$$\boxed{\overline{\boldsymbol{R_v}} = \ell_v^{+} + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{v,v'}(\overline{\boldsymbol{R_v}}, Y_{v'}) - \ell_v^{+} \right) \right\rfloor}$$

Critical path length:
maximum sum of WCETs in a path ending with v

Critical Path

- **Critical path length**: maximum sum of WCETs in a path ending with $\boldsymbol{v}$



$$\ell_v^+ = 13$$

Response-Time Analysis

- **Main result**: we proved that

$$R_v \;\leq\; \overline{\boldsymbol{R_v}}$$

$$\boxed{\overline{\boldsymbol{R_v}} = \ell_v^+ + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{v,v'}(\overline{\boldsymbol{R_v}}, Y_{v'}) \; - \ell_v^+ \right) \right\rfloor}$$

Sum on all vertices $\boldsymbol{v'}$ in the task-set

Response-Time Analysis

- **Main result**: we proved that

$$R_v \le \overline{\boldsymbol{R_v}}$$

$$\boxed{\overline{\boldsymbol{R_v}} = \ell_v^{+} + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{v,v'}\left(\overline{\boldsymbol{R_v}}, Y_{v'}\right) - \ell_v^{+} \right) \right\rfloor}$$

Upper-bound on the worst-case workload generated by $\boldsymbol{v}'$ interfering with $\boldsymbol{v}$

Worst-Case Workload

- Upper-bound on the worst-case workload generated by $\boldsymbol{v'}$ interfering with $\boldsymbol{v}$

$$W_{v,v'}(\overline{R_v}, Y_{v'})$$

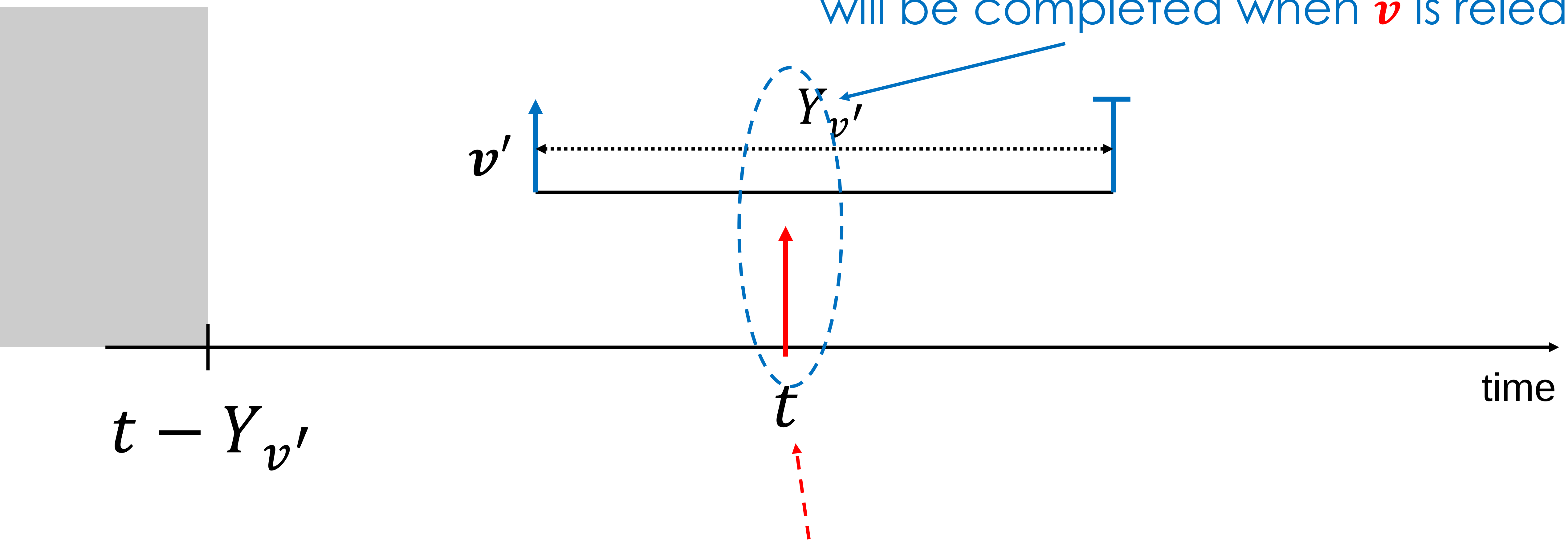
Tentative response-time of vertex v, used in the fixed-point iteration starting with $\overline{R_v} = e_v$

Response-time upper-bound Must be always greater than the response-time ($Y_{v'} = D_v + 1$ in the limit case)

Worst-Case Workload

- A generic vertex v' interferes with v released at t

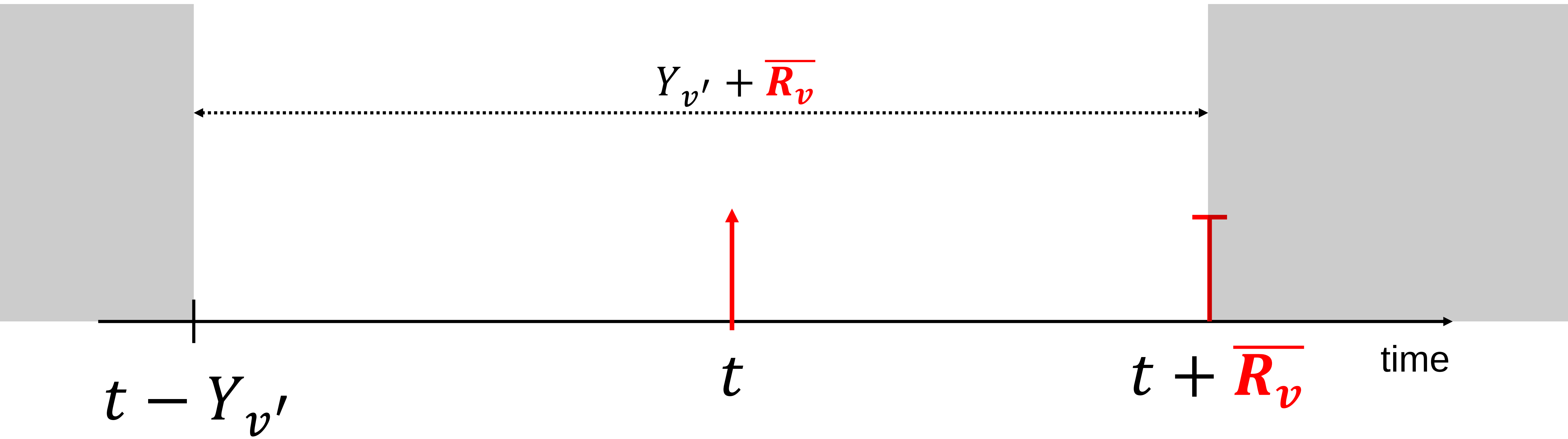
If shifted more on the left the job of v' will be completed when v is released



Release of a job of v

Worst-Case Workload

- A generic vertex v' interferes with v released at t



Interfering workload

$$\left\lceil \frac{Y_{v'} + \overline{R_v}}{T_{v'}} \right\rceil e_{v'}$$

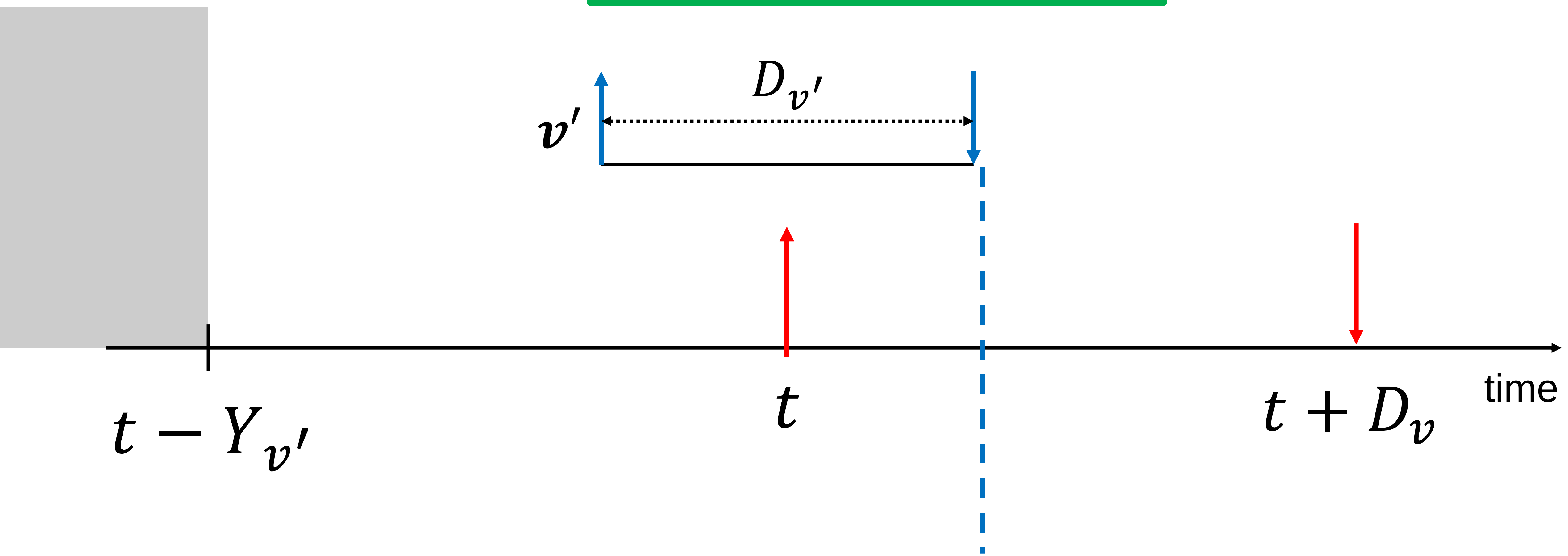
In case of **G-DM**

Null for vertices of lower-priority tasks

Worst-Case Workload

- A generic vertex v' interferes with v released at t

In case of **G-EDF**

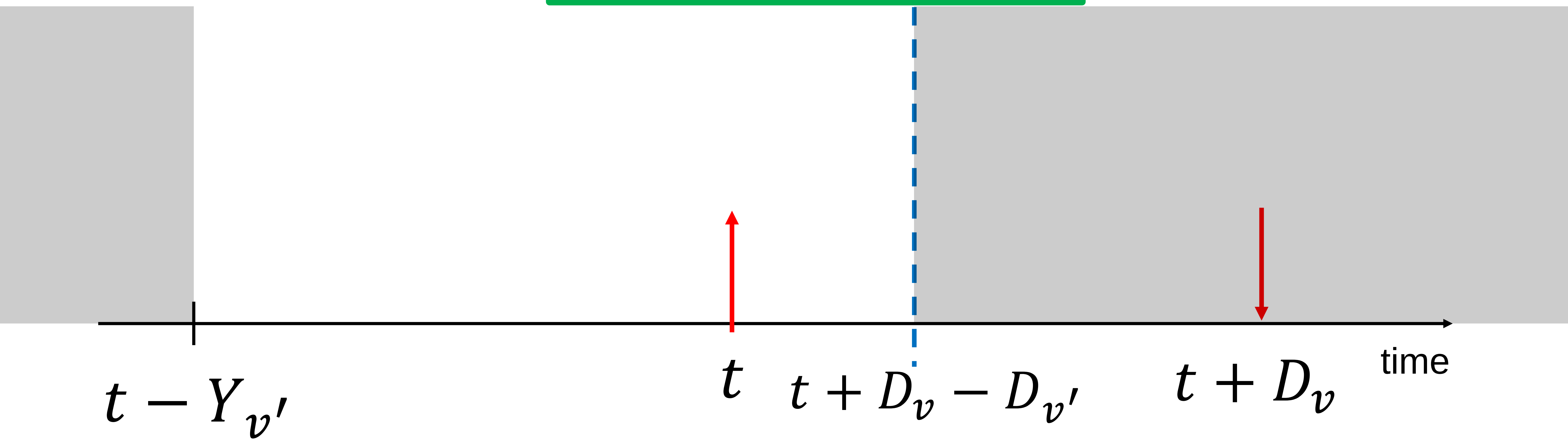


Jobs of v' released after $t + D_v - D_{v'}$ will not interfere with v

Worst-Case Workload

- A generic vertex $\boldsymbol{v'}$ interferes with $\boldsymbol{v}$ released at t

In case of **G-EDF**

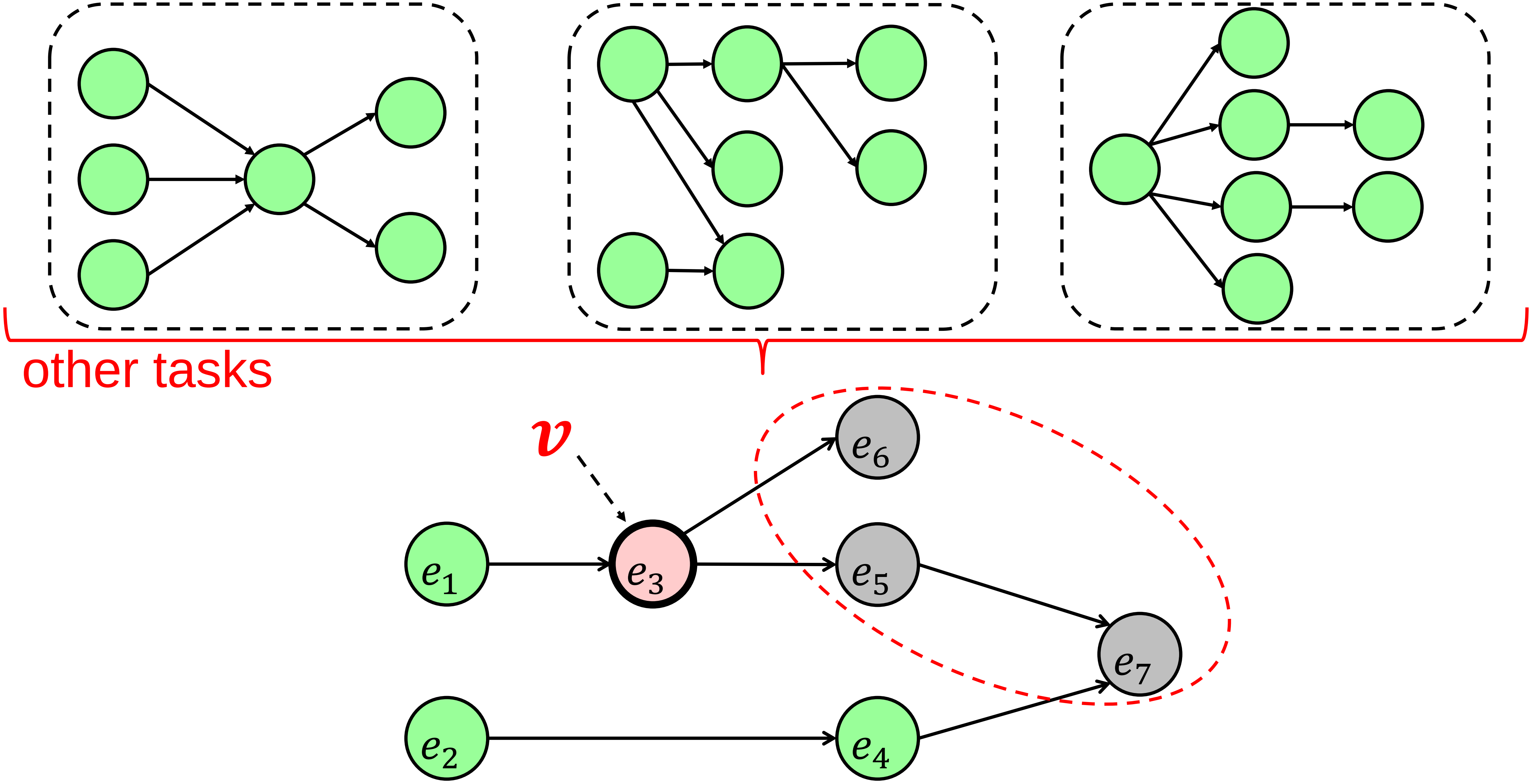


Interfering workload

$$\left\lceil \frac{Y_{v'} + min\{\overline{\boldsymbol{R_v}}, D_v - D_{v'}\}}{T_{v'}} \right\rceil e_{v'}$$

Response-Time Analysis

- **Successors** in the same job of a DAG-task cannot interfere



Response-Time Analysis

- **Main result**: we proved that

$$R_v \le \overline{\boldsymbol{R}_v}$$

$$\boxed{\overline{\boldsymbol{R}_v} = \ell_v^+ + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{v,v'}(\overline{\boldsymbol{R}_v}, Y_{v'}) - \ell_v^+ \right) \right\rfloor}$$

Schedulability Test

Algorithm RTA(N)

Maximum number of iterations

Schedulability Test

Algorithm RTA(N)

1. We start with $Y_v = D_v + 1, \forall v, \mathrm{i} = 1$

2. Compute the **fixed-point** of
$$\overline{\boldsymbol{R_v}} = \ell_v^+ + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{v,v'}(\overline{\boldsymbol{R_v}}, Y_{v'}) - \ell_v^+ \right) \right\rfloor$$

3. If $\overline{\boldsymbol{R_v}} \leq \boldsymbol{D_v}$ return SCHEDULABLE

4. If $Y_v == \overline{\boldsymbol{R_v}}, \forall v$ **OR** i==N return NOT SCHEDULABLE

5. Else, update response-times as $Y_v = \overline{\boldsymbol{R_v}}, \forall v$ and **go to** step 2

$i++$

Pseudo-Polynomial Complexity

Polynomial-Time Schedulability Test

- If we set $Y_v = D_v + 1$ and $\overline{R_v} = D_v$ it is possible to obtain a simple polynomial-time schedulability test without involving any iteration

$$\overline{\boldsymbol{R}_{\boldsymbol{v}}} = \ell_v^+ + \left\lfloor \frac{1}{m} \left(\sum_{v'} \boldsymbol{W}_{\boldsymbol{v},\boldsymbol{v}'}(D_v, D_{v'} + 1) - \ell_v^+ \right) \right\rfloor$$

Polynomial Complexity

Augmentation Bound

In case of a task-set composed of a **single DAG-Task** (N=1) we proved that

- Our test based on response-time analysis has
  - Augmentation bound **< 3** for G-EDF;
  - Augmentation bound **< 5** for G-DM.

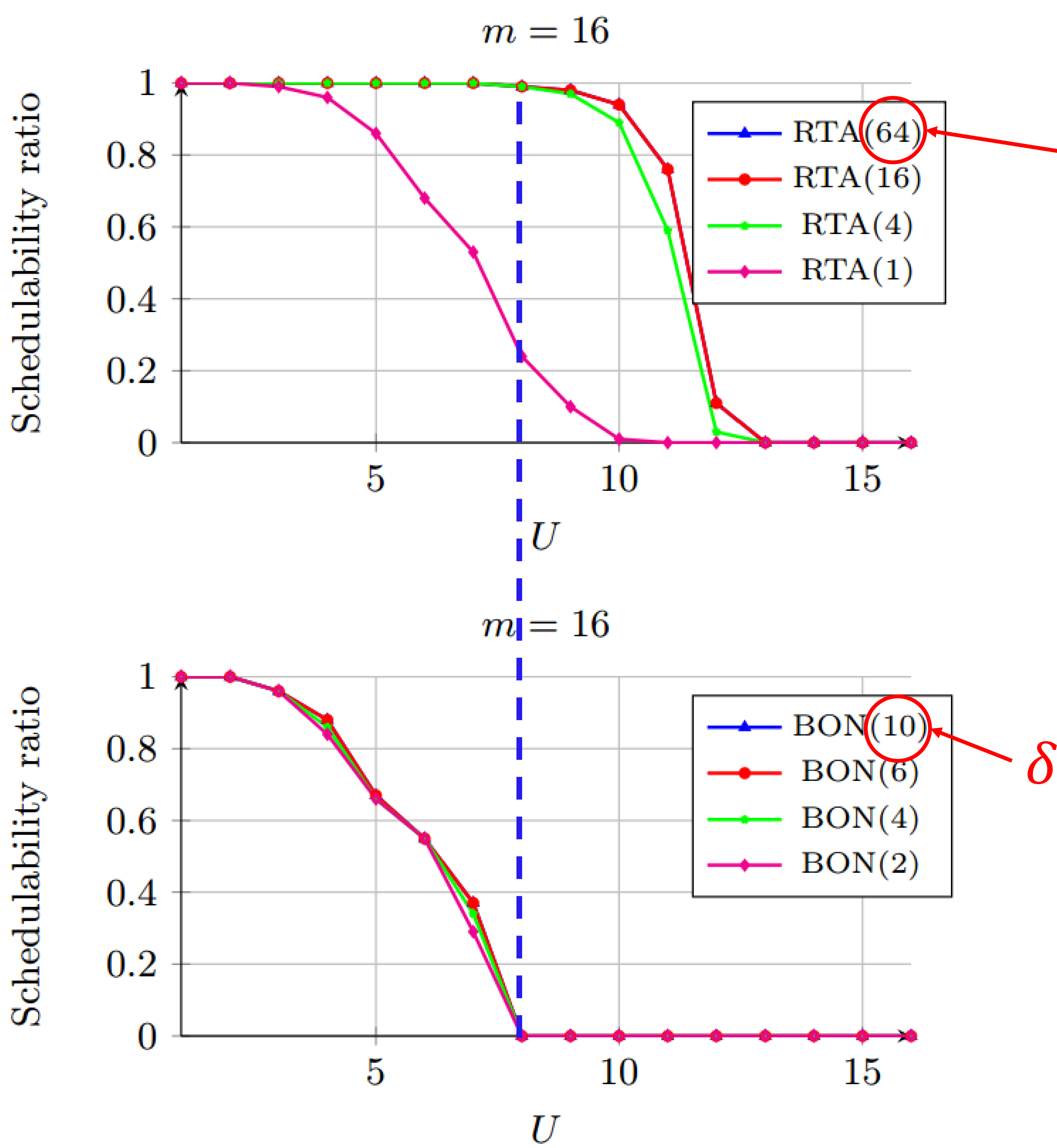
Experimental Results

- The proposed schedulability tests have been evaluated by using *synthetic workload*
- `libdag` – DAG-Tasks generator and schedulability test. Soon publicly available online!
- Comparison against the test based on augmentation bound proposed in

V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, and A. Wiese. "*Feasibility analysis in the sporadic DAG task model*", In proc. of ECRTS 2013

To the best of our knowledge it is the only test dealing with arbitrary deadlines

Experimental Results



Number N of external iterations in our algorithm

The test of Bonifaci et al. is based on a workload approximation up to an ϵ-error with $2^{-\delta}$

Experimental Results

- Running times of the schedulability tests

	Min (s)	Max (s)	Avg (s)
RTA(64)	0.001	0.397	0.014
RTA(16)	0.001	0.225	0.014
RTA(4)	0.001	0.050	0.009
RTA(1)	0.001	0.014	0.005

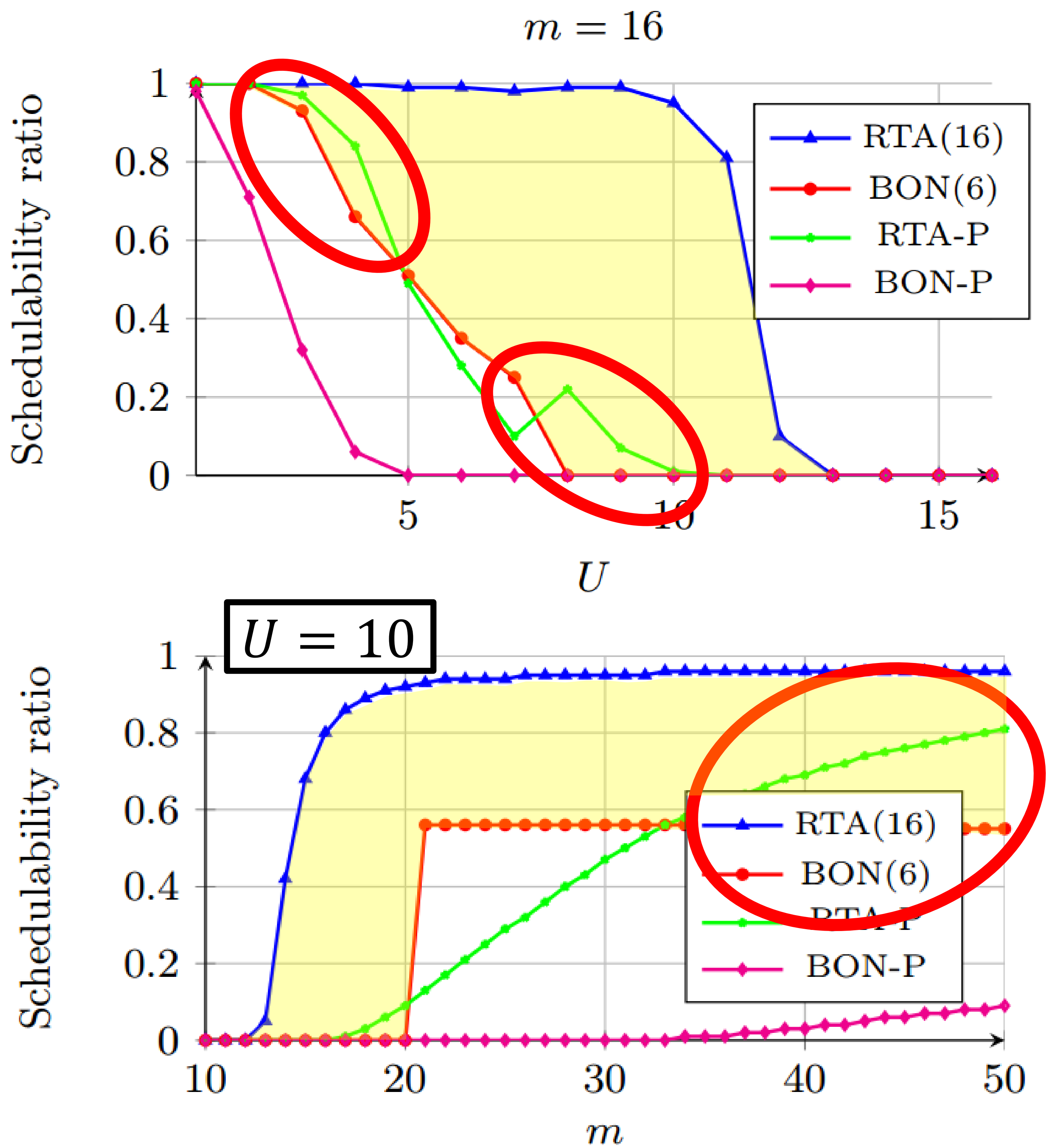
RTA test has running time lower of two orders of magnitude

	Min (s)	Max (s)	Avg (s)
BON(10)	0.000	17.855	3.357
BON(6)	0.000	1.160	0.214
BON(4)	0.000	0.292	0.051
BON(2)	0.000	0.142	0.012

Exponential increase of the running time as the test precision increases

(Intel Xeon @ 3.5 Ghz)

Experimental Results



Take-away messages

- RTA test outperforms the speed-up based test in all the tested configurations;
- In some cases our polynomial-time test performs better than the speed-up based test that has pseudo-polynomial complexity

Conclusions

- We proposed a new Response-Time Analysis for the sporadic DAG-Task model under both G-EDF and G-DM scheduling;
- The analysis handles DAG-Tasks with arbitrary deadline and arbitrary utilization;
- Two schedulability tests have been derived (pseudo-polynomial and polynomial complexity);
- Extensive set of experimental results confirmed the effectiveness of the test.

Future Work

- More accurate characterization of the interfering workload;
- Support for conditional statements in the DAG-Task;
- Integration of locking protocols in the analysis;
- Handle distributed computations.

Thank you!

Alessandro Biondi
alessandro.biondi@sssup.it