Improving Multiprocessor Real-Time Systems with Bursty Inputs under Global EDF using Shapers

Yue Tang[1], Yuming Jiang[2], Xu Jiang[1], Nan Guan[1]

[1] The Hong Kong Polytechnic University [2]Norwegian University of Science and Technology

Outline

- **Motivation**

- **Framework**

- **Analysis Details**

- **Evaluation**

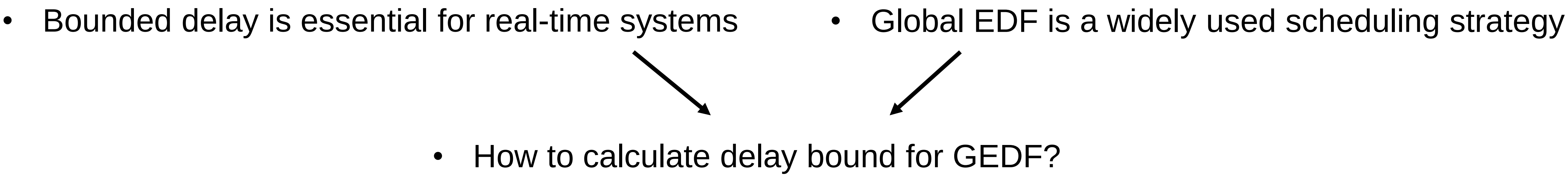
Outline

- **Motivation**
- **Framework**
- **Analysis Details**
- **Evaluation**

Motivation

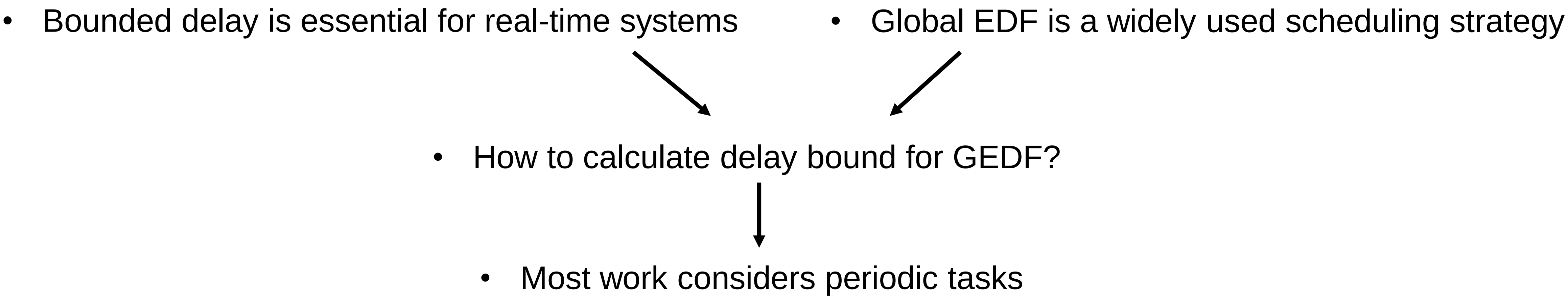
- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy

Motivation



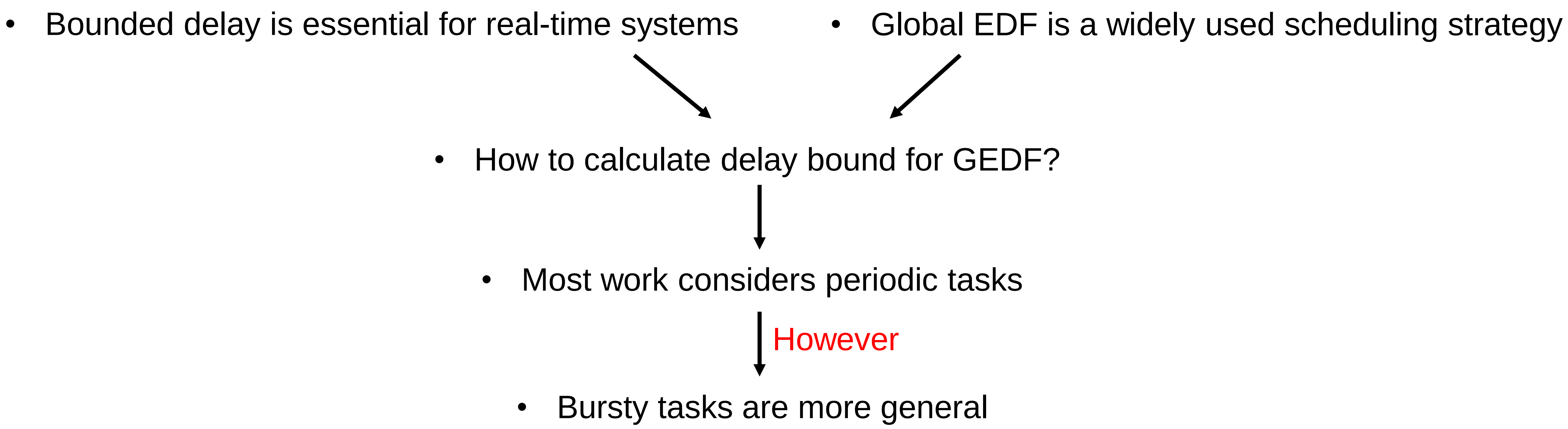
- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy
- How to calculate delay bound for GEDF?

Motivation



- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy
- How to calculate delay bound for GEDF?
- Most work considers periodic tasks

Motivation



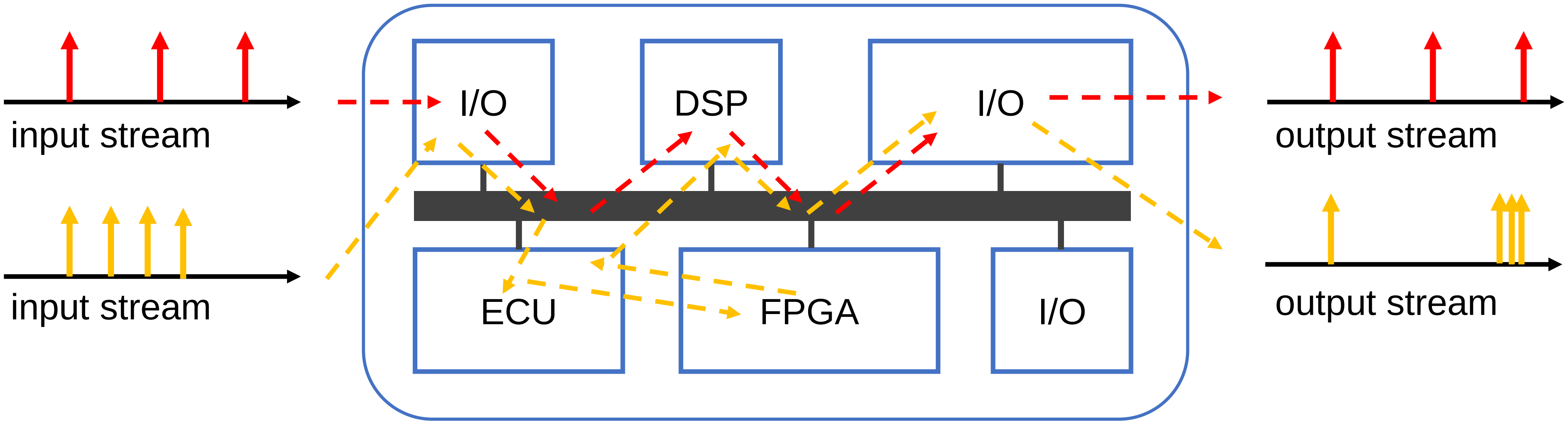
- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy
- How to calculate delay bound for GEDF?
- Most work considers periodic tasks

However

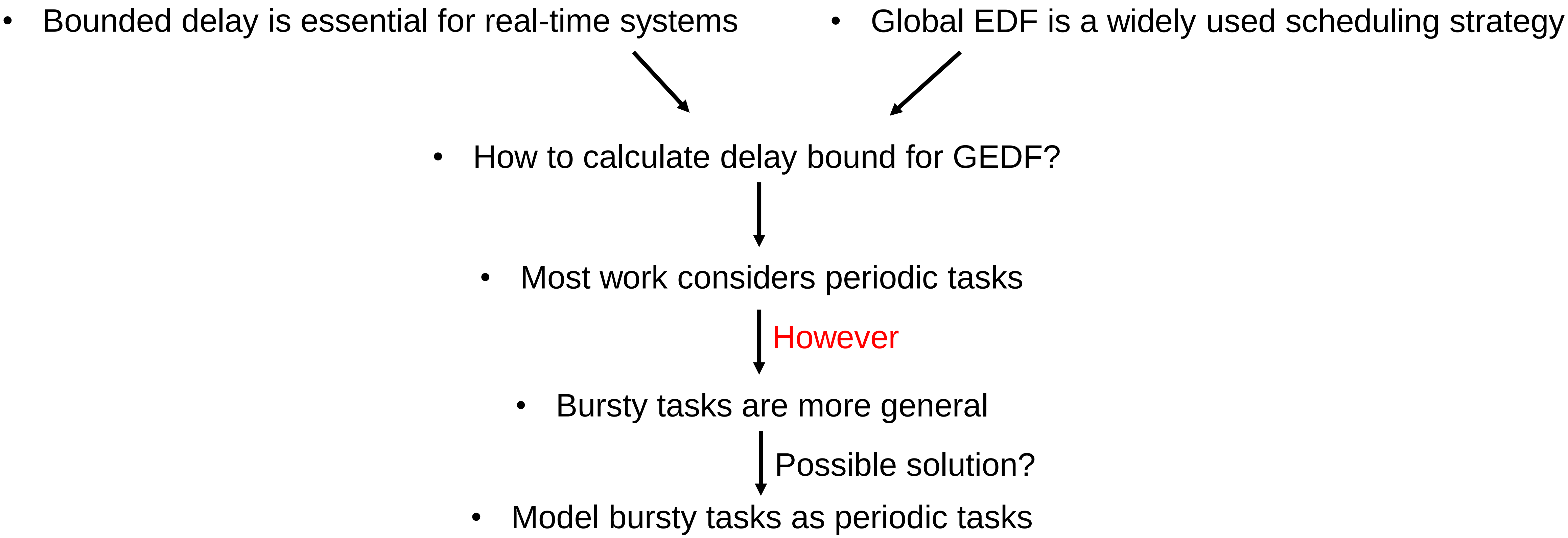
- Bursty tasks are more general

Motivation

- Bursty tasks are more general in real applications



Motivation



- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy
- How to calculate delay bound for GEDF?
- Most work considers periodic tasks

However

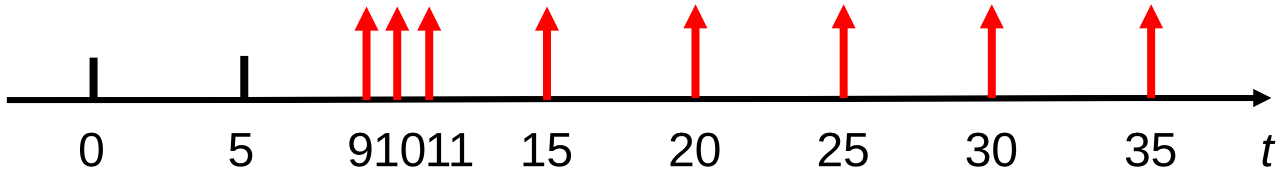
- Bursty tasks are more general

Possible solution?

- Model bursty tasks as periodic tasks

Motivation

- Does modeling bursty tasks as periodic tasks work for calculating delay bound? *No !*

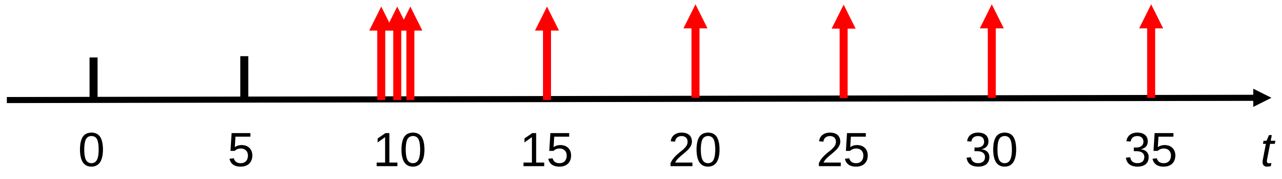


Choose the period of periodic tasks

T = 1 ? Overestimate workload → too pessimistic

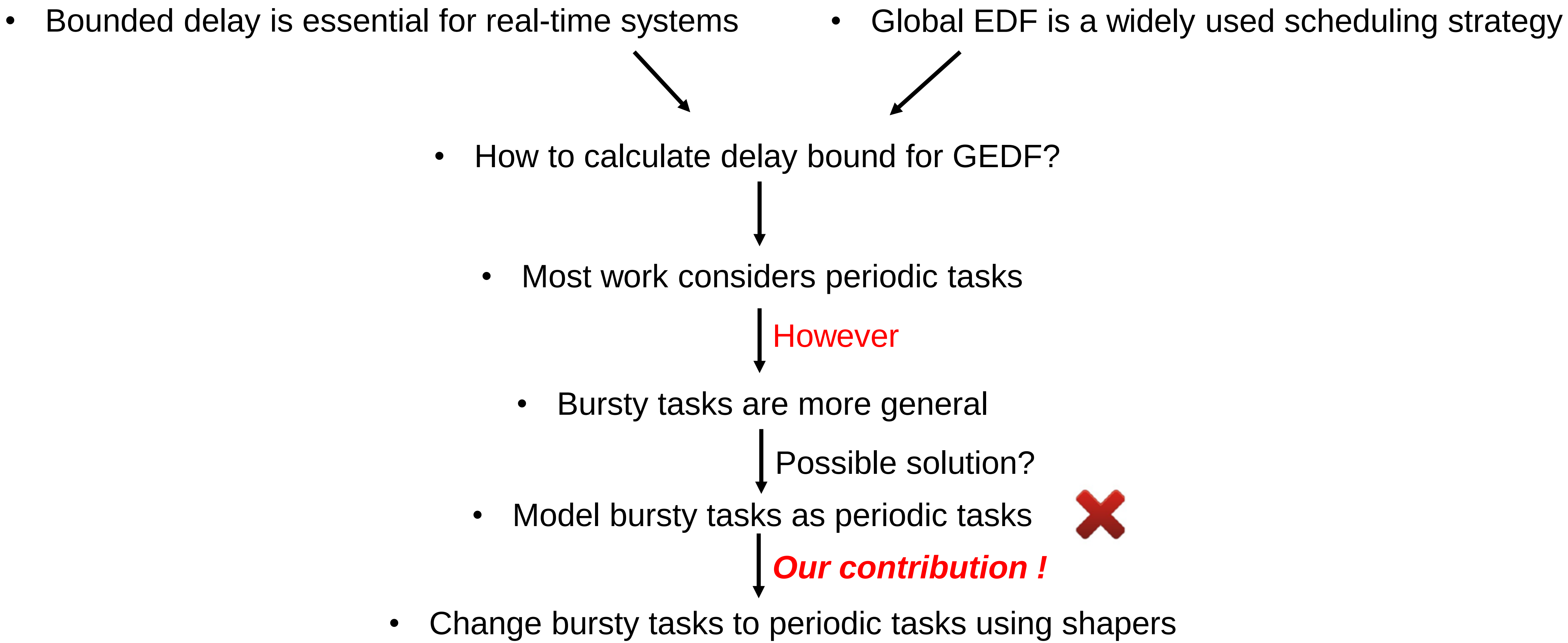
T = 5 ? Underestimate workload → wrong results (Assume D = 5, one task on uniprocessor)

In the extreme case, more than one job arrives simultaneously (e.g., 3 events at time 10)



Can not set the period !

Motivation



- Bounded delay is essential for real-time systems
- Global EDF is a widely used scheduling strategy

- How to calculate delay bound for GEDF?

- Most work considers periodic tasks

However

- Bursty tasks are more general

Possible solution?

- Model bursty tasks as periodic tasks ✗

Our contribution !

- Change bursty tasks to periodic tasks using shapers

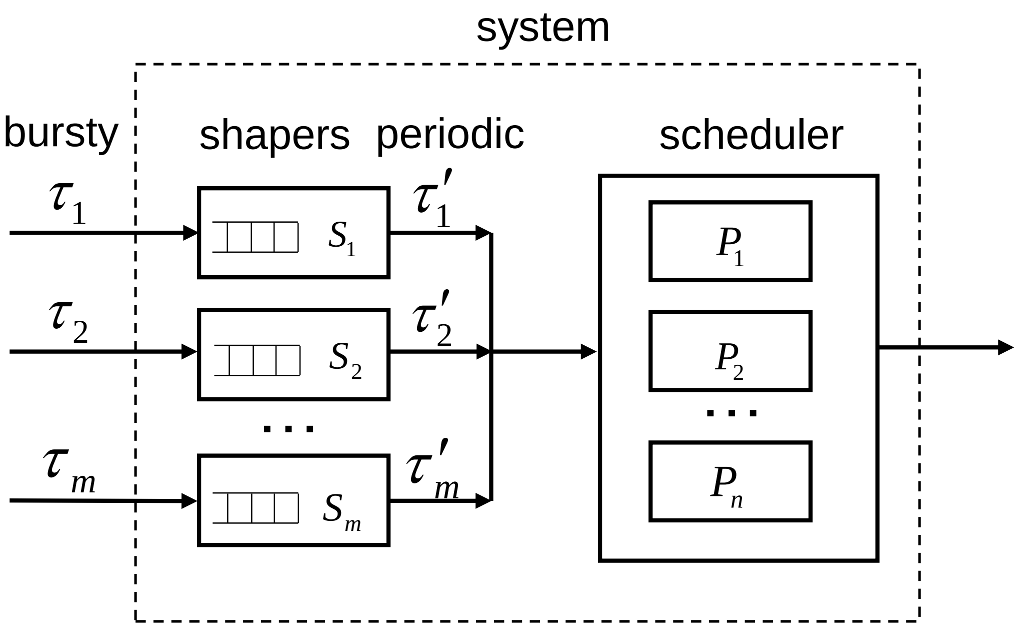
Our Contribution

- Use shapers to calculate delay bound for multiprocessors under global EDF
- Design a heuristic algorithm to adjust the parameters of shapers for higher acceptance ratio

Outline

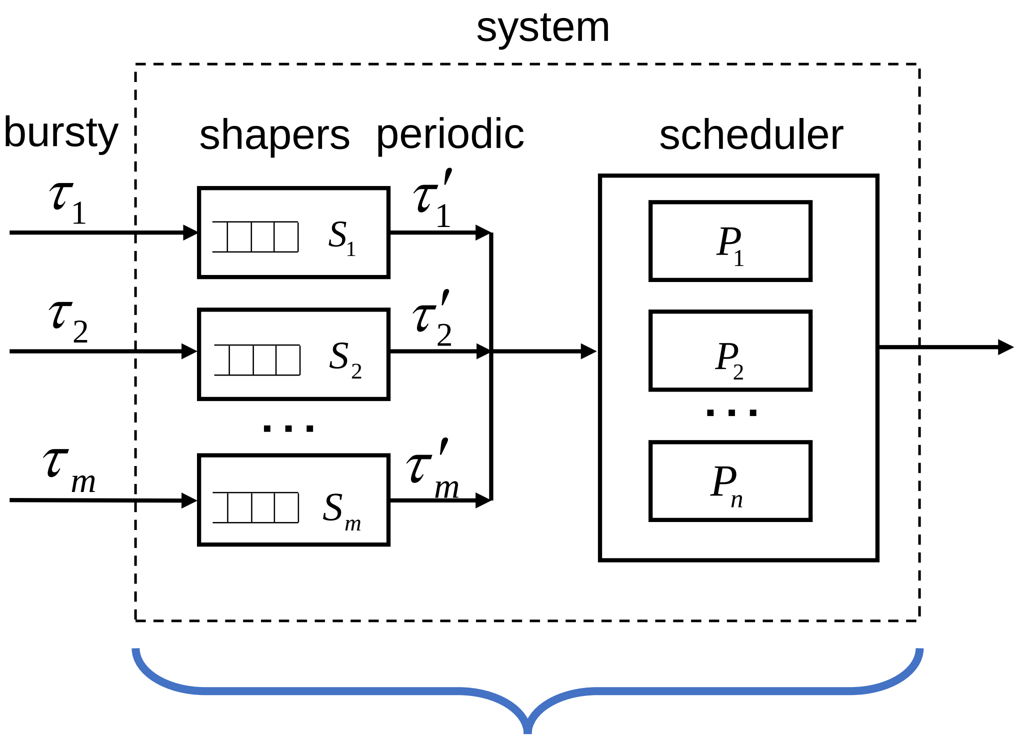
- **Motivation**
- **Framework**
- **Analysis Details**
- **Evaluation**

Framework



Why our approach is correct?

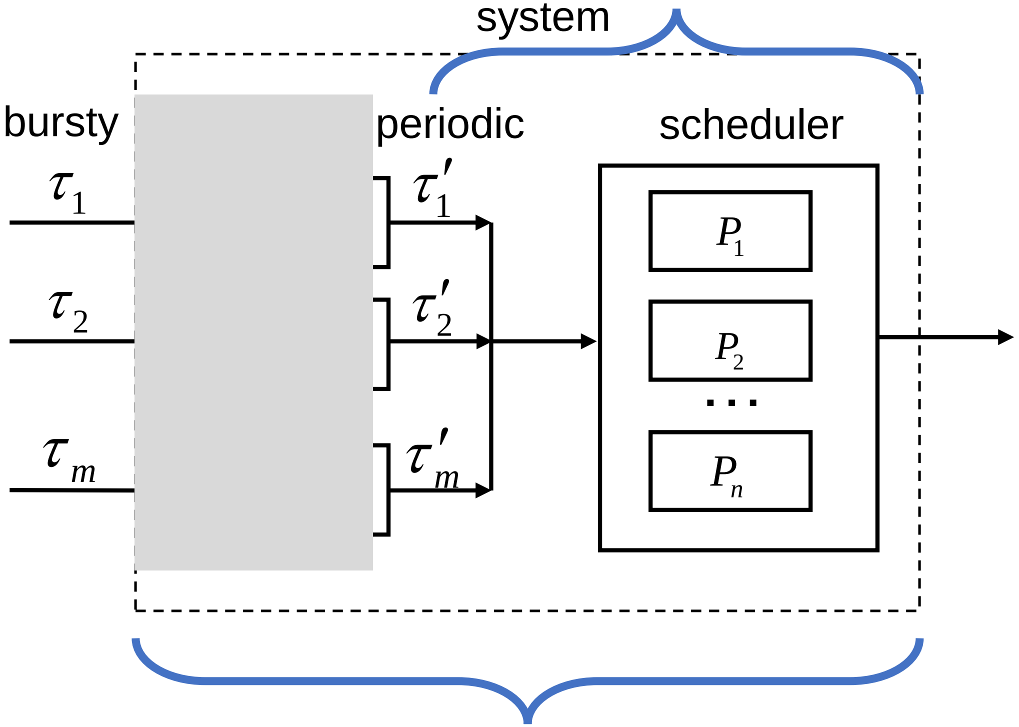
Correctness



our calculated delay bound
(changing bursty tasks as periodic)

Correctness

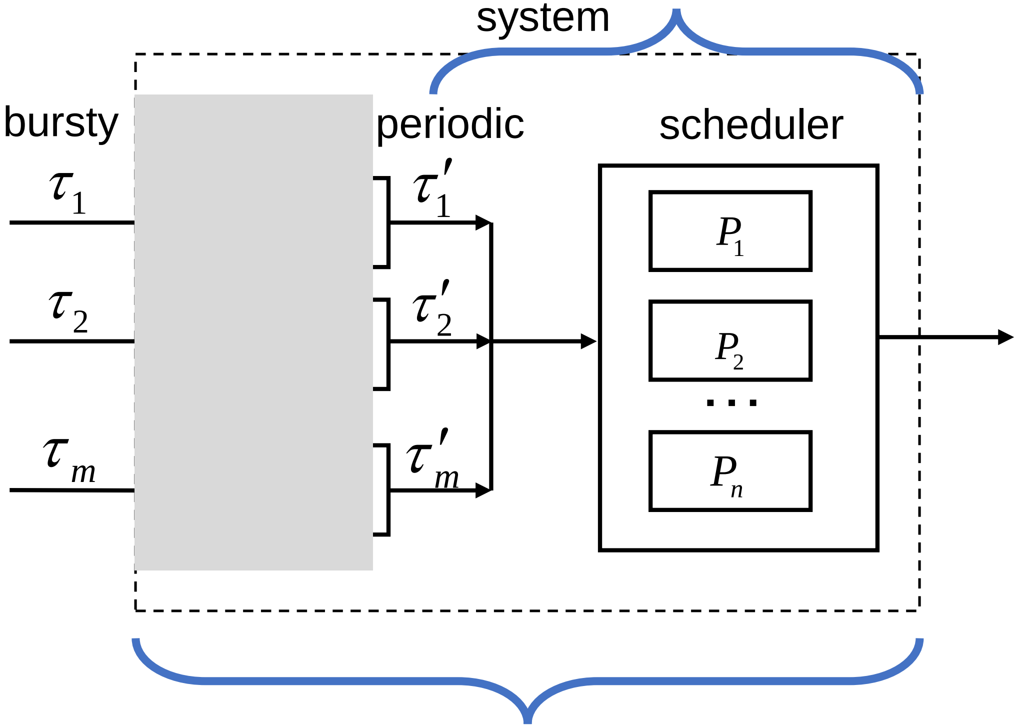
calculated delay bound
(modeling bursty tasks as periodic)



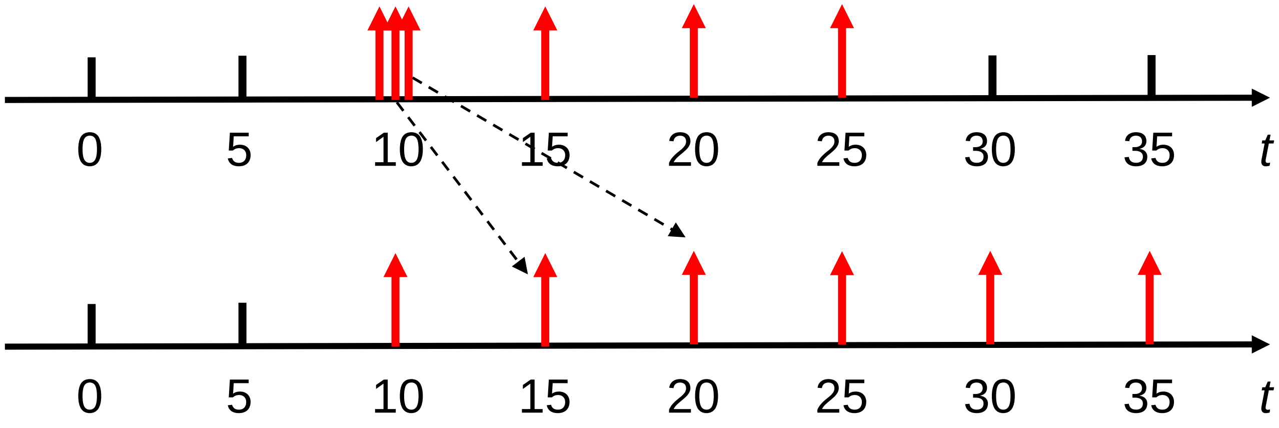
our calculated delay bound
(changing bursty tasks as periodic)

Correctness

calculated delay bound
(modeling bursty tasks as periodic)



our calculated delay bound
(changing bursty tasks as periodic)



Assume D = 5, one task on uniprocessor

If consider the delay bound at the scheduler → schedulable ✗

If consider the overall delay → unschedulable ✓

Outline

- Motivation
- Framework
- **Analysis Details**
- Evaluation

System Model

- Task Model

For each task, it has

Worst case execution time *C*

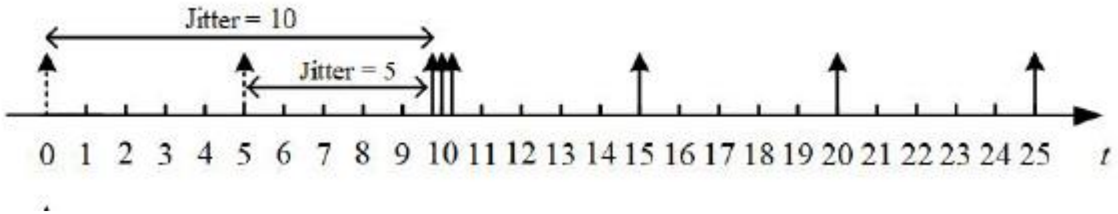
Relative deadline *D* → Relative deadline only decides a task's finish time

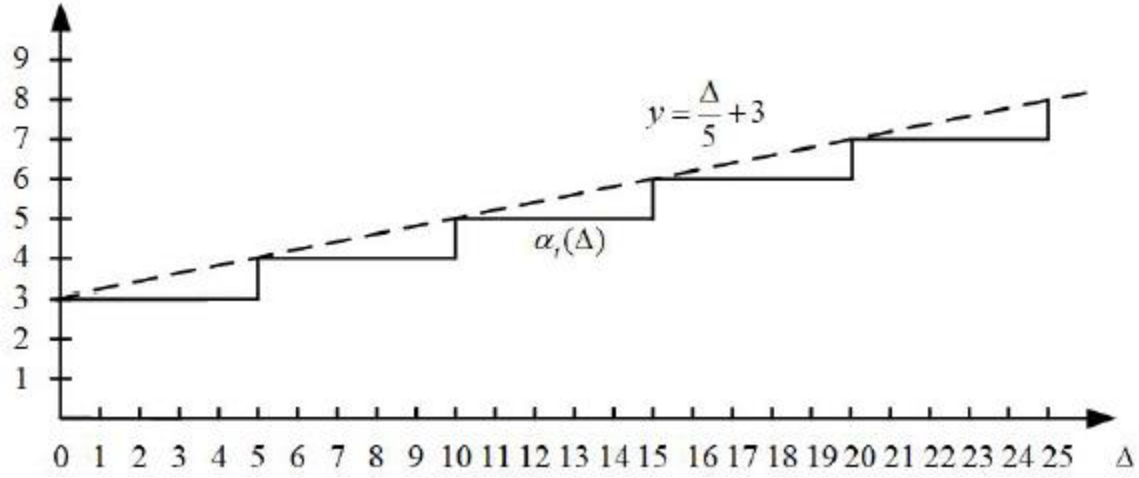
Priority indicator *λ* → The priority of a task is decided by priority indicator

- Scheduling strategy

A more general global EDF scheduling

Arrival curve *α: maximum number of arrived events in any length of time interval*

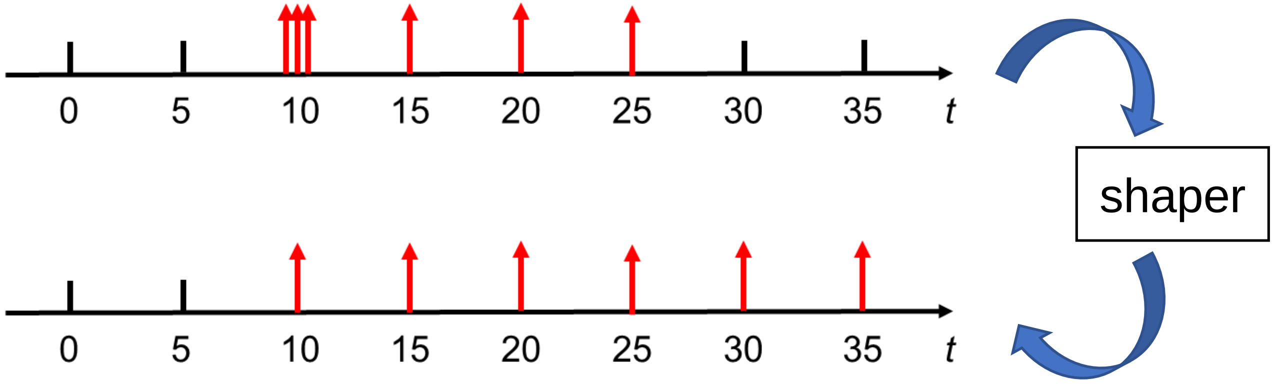




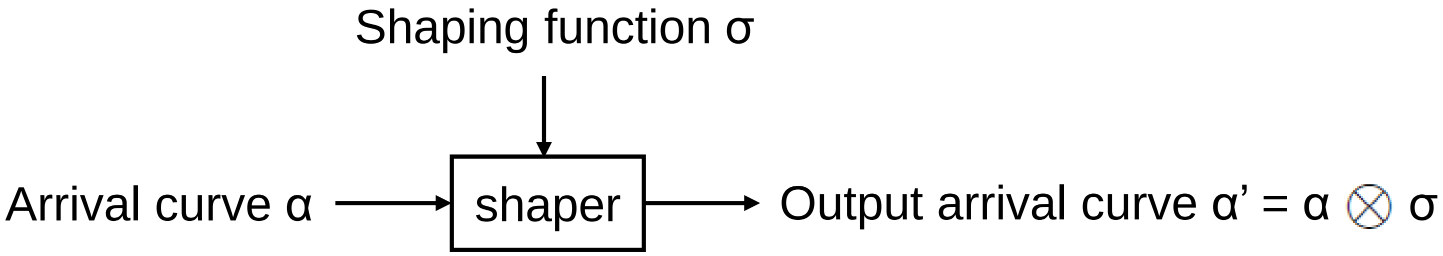
Shaper

- Behavior

  Process job sequences and force its output job sequences to confirm to some time constraints



- Shaper Component


Shaping function σ

Arrival curve α

shaper

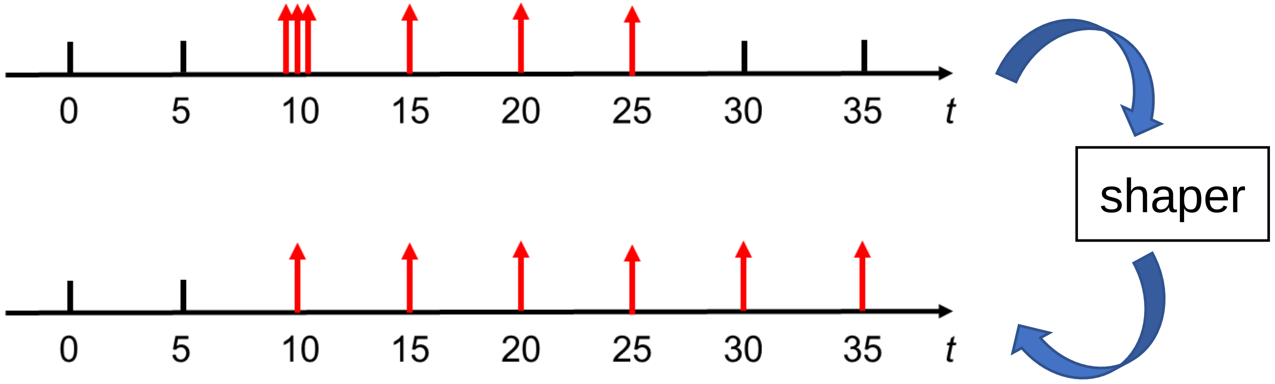
Output arrival curve α’ = α ⊗ σ

- When generating periodic outputs, the shaper/shaping function is uniquely decided by period T.

Shaper

- Behavior

  Process job sequences and force its output job sequences to confirm to some time constraints

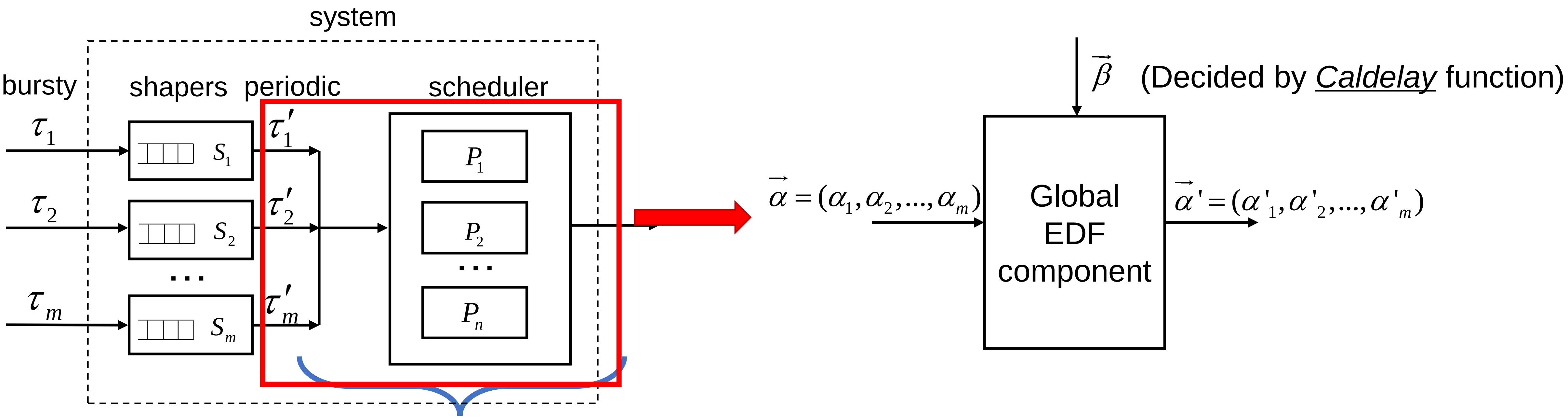


- Larger number of simultaneously arrived jobs → larger delay bound at the shaper

- Smaller period of shaper → smaller delay bound at the shaper

Global EDF Component

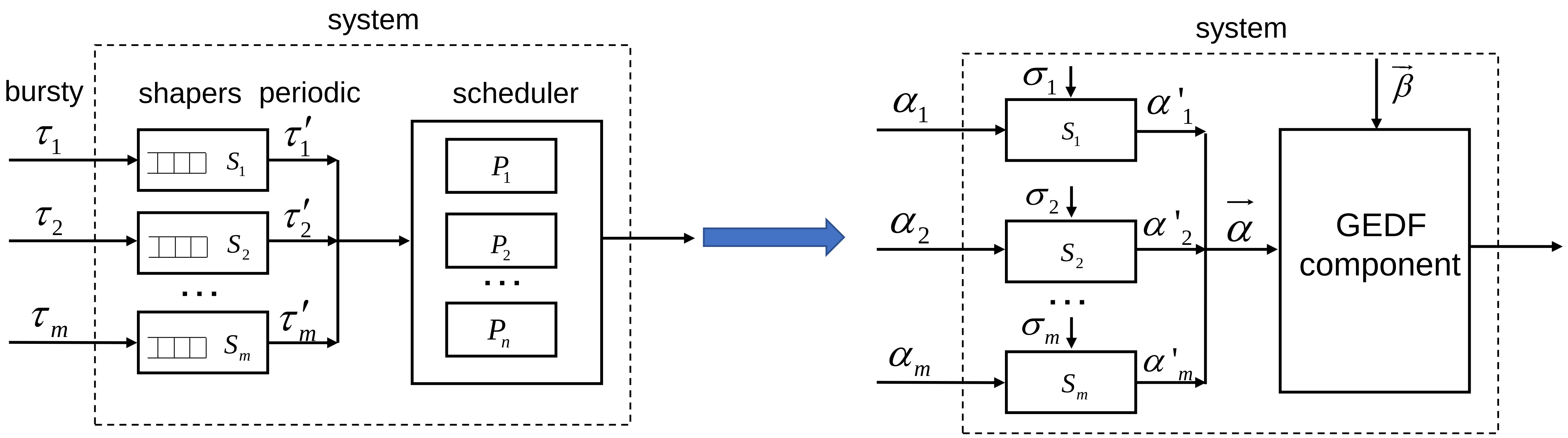
- Model the global EDF (GEDF) scheduler as a GEDF component



Delay = *Caldelay* (input tasks, analysis techniques)

Model the System with Shapers

- A system with shapers is modeled as a sequence of abstract components

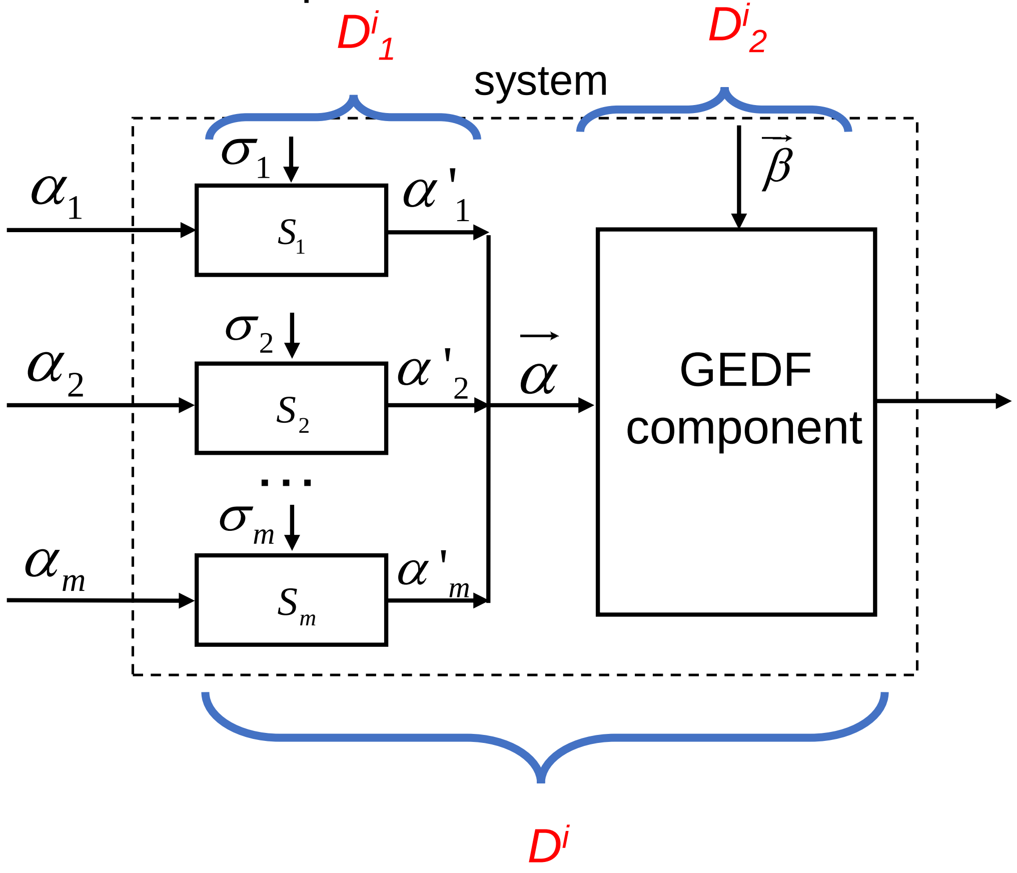


How to Analyze (for given shapers)

- Theorem

For each task, the overall delay bound is equal to the sum of that at the shaper and that of the scheduler

$D^i = D^i_1 + D^i_2$



How to Choose Shapers

- For periodic outputs, the behavior is decided by the period of shapers
- Why difficult to decide the period T

  $D^i = D^i_1 + D^i_2$

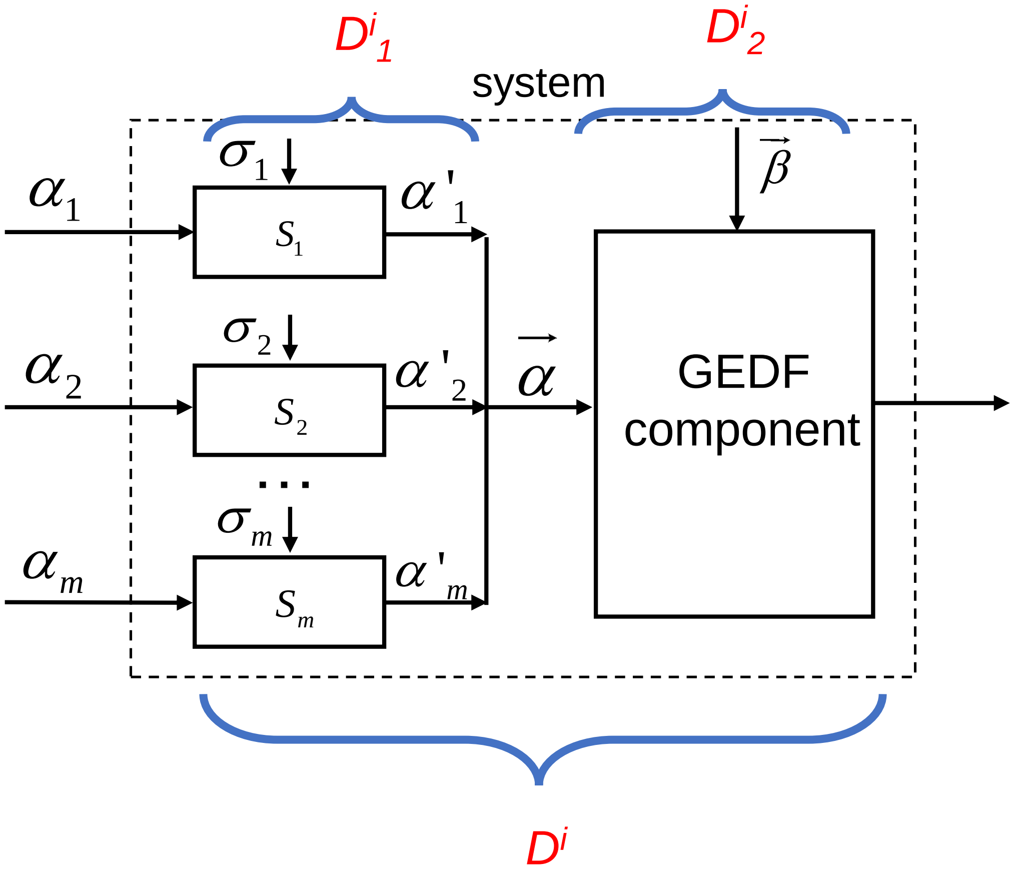
  ① Increase T → larger D^i_1 , smaller D^i_2

  Decrease T → smaller D^i_1 , larger D^i_2

  ② The calculation D^i_2 *depends on other tasks*

  ③ Enumerating all possible values is low-efficient

- We propose a heuristic algorithm for higher acceptance ratio



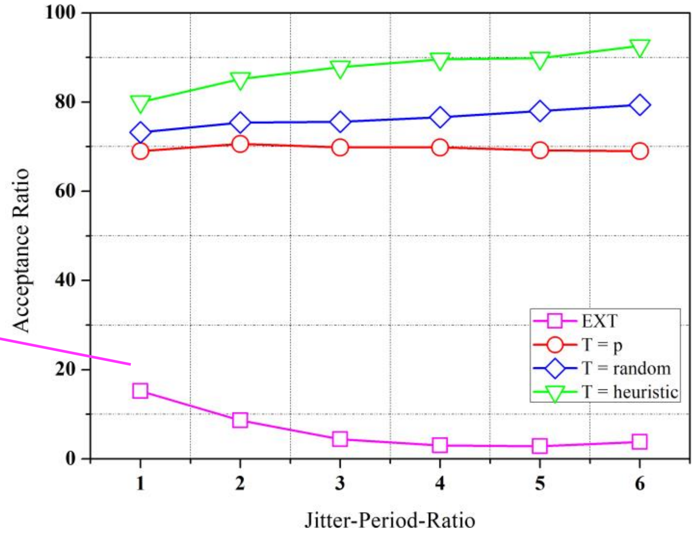
Outline

- **Motivation**
- **Framework**
- **Analysis Details**
- **Evaluation**

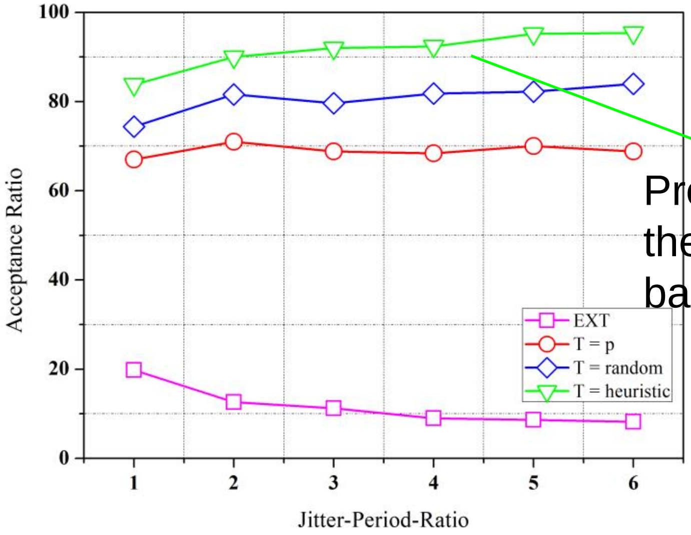
Evaluation (acceptance ratio)

With simultaneously arrived jobs

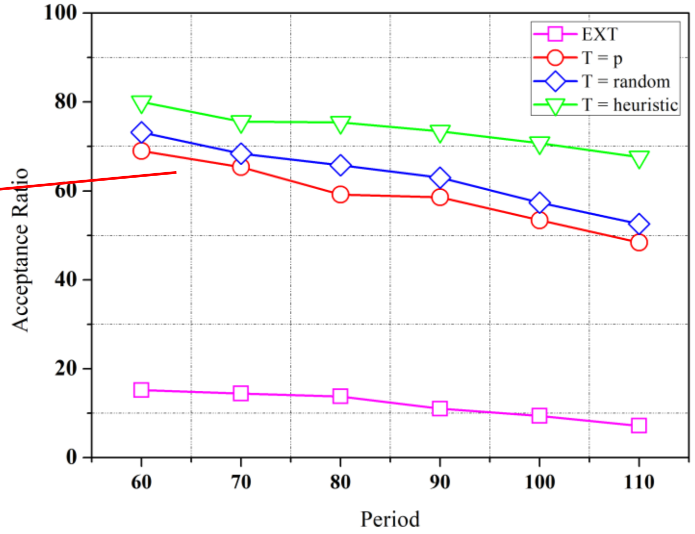
Without simultaneously arrived jobs



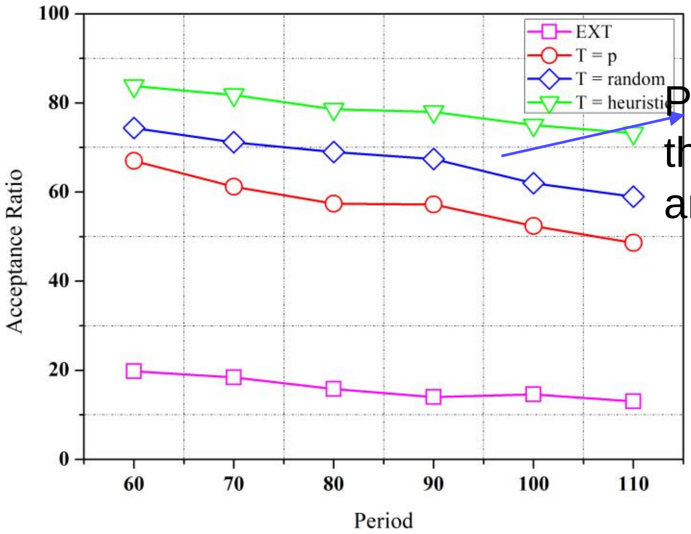
The only one existing work for bursty inputs under global EDF



Proposed approach, setting the period of the shaper based on the heuristic



Proposed approach, setting the period of the shaper based on the input task



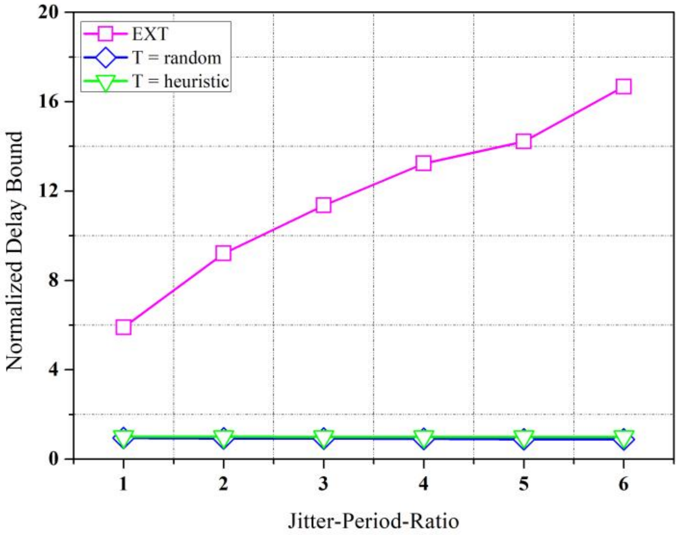
Proposed approach, setting the period of the shaper arbitrarily

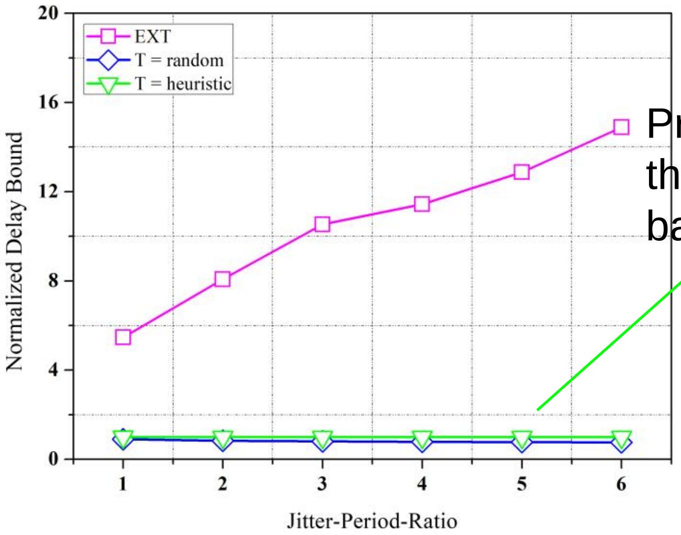
Evaluation (normalized delay bound)

With simultaneously arrived jobs

Without simultaneously arrived jobs

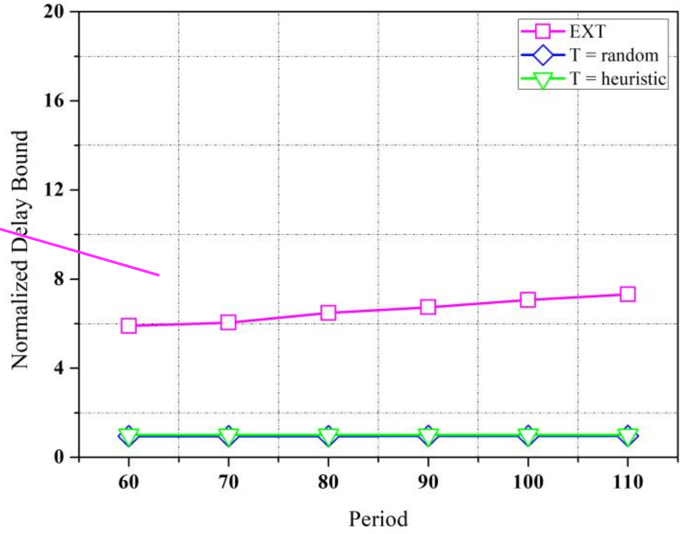
The normalized delay bound is calculated based on that when setting the period based on the input

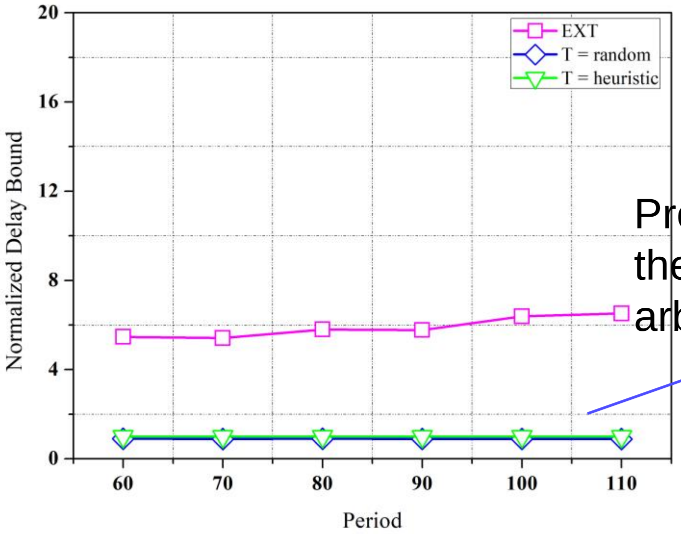




Proposed approach, setting the period of the shaper based on the heuristic

The only one existing work for bursty inputs under global EDF





Proposed approach, setting the period of the shaper arbitrarily

Thanks for attention !