Schedulability analysis analysis of real-time distributed systems

UNIVERSIDAD DE CANTABRIA

1. Introduction
2. Single processor systems
3. ***Distributed systems***
4. **Schedulability analysis: holistic approach**
5. **Schedulability analysis: offset-based approaches**
6. Schedulability analysis: EDF
7. Modelling techniques: MAST
8. Conclusion and future work

Real-time networks

UNIVERSIDAD DE CANTABRIA

Very few networks guarantee real-time response

- many protocols are based on collisions
- no priorities or deadlines in most protocols
  - Wireless: IEEE 802.11e, Sensor networks,...
  - IEEE 802.1p, with 3-bit priority field (implementation-specific)

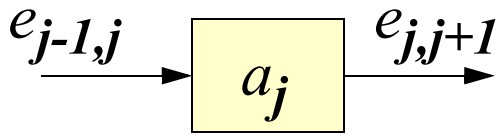
Some solutions

- CAN (Controller Area Network) bus, now an ISO standard
- Priority-based token passing (e.g., RT-EP)
- Time-division multiplexed access
- Point to point networks

No commercial or standard EDF networks yet

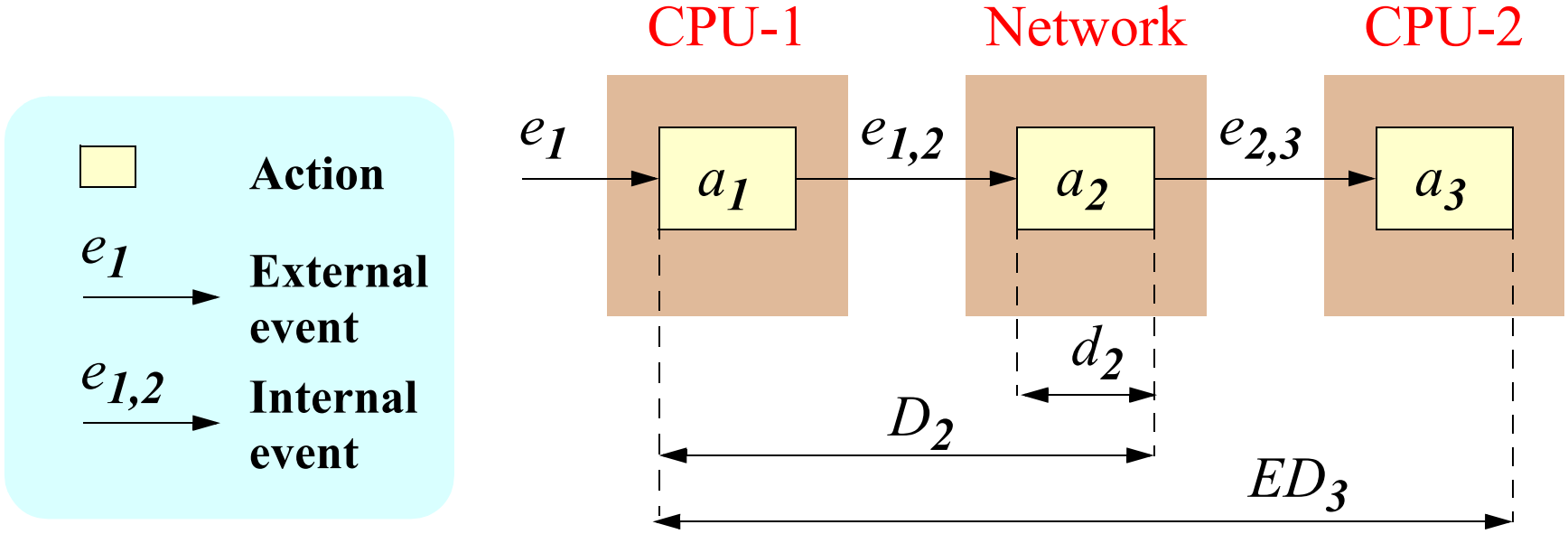
Distributed system model

Linear Action



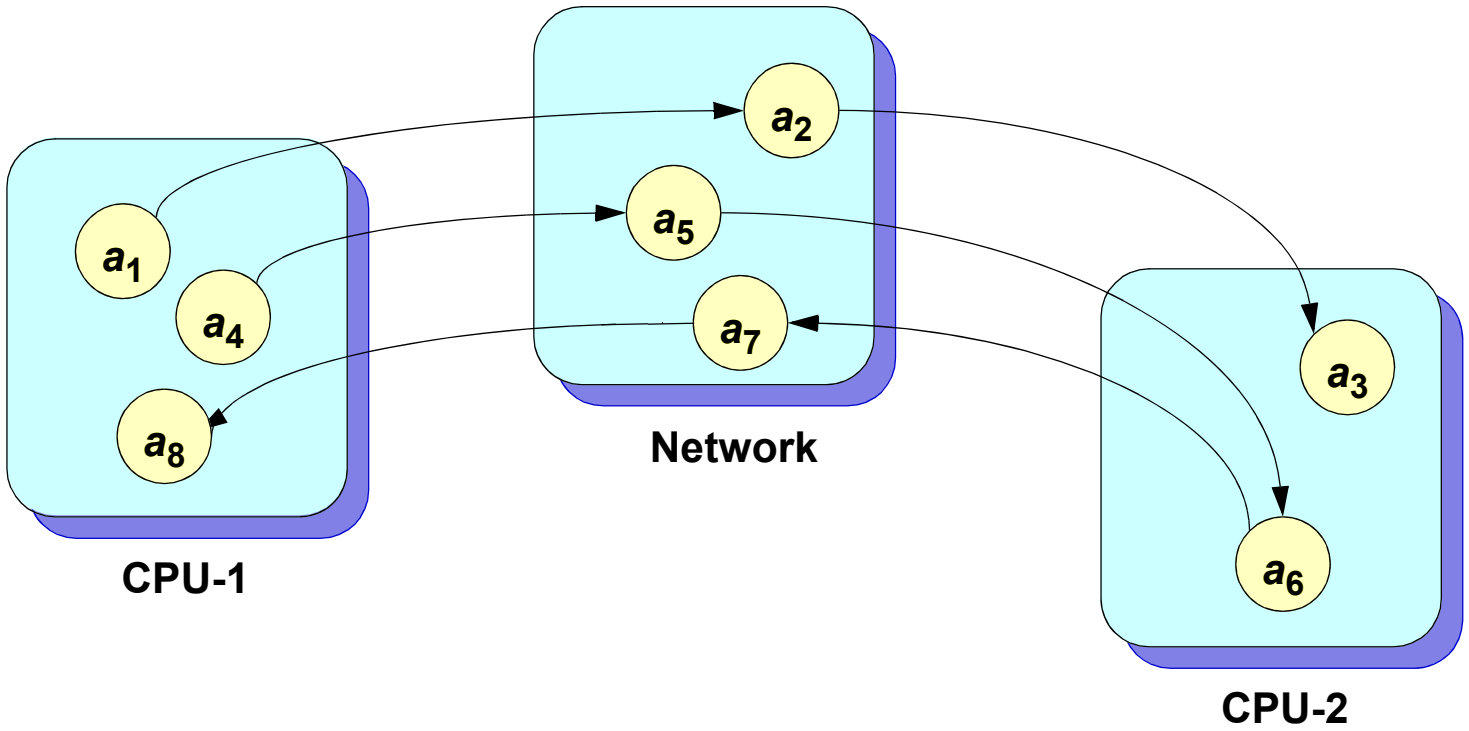
$$T_{j-1,j} = T_j$$

Linear Response to an Event



Jitter in distributed systems

In this example, actions a_2,a_3 and a_5,a_6,a_7,a_8 do have jitter, even though a_1 and a_4 are purely periodic and therefore jitterless



The problem

UNIVERSIDAD DE CANTABRIA

Mutual dependencies in the flowchain

- Jitter in one resource depends on the response times in other resources
- Response times depend on jitters

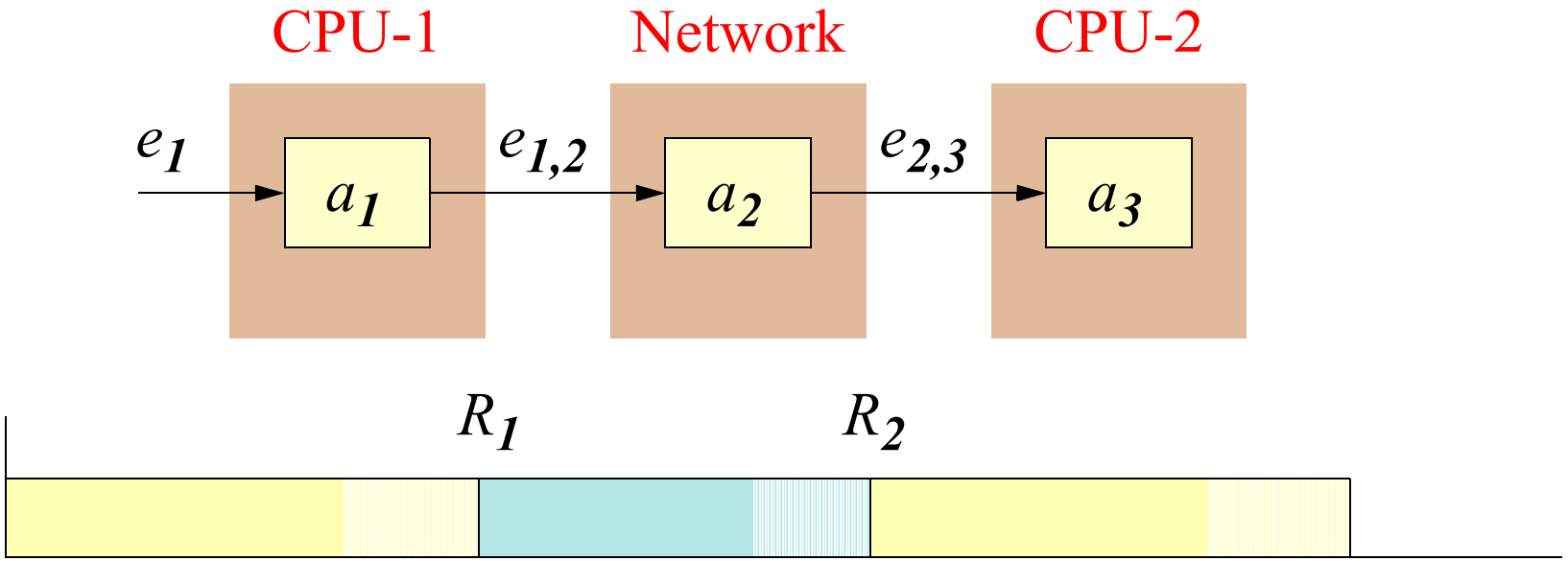
Solutions

- Algorithms that can calculate jitter and response times
- Change the scheduler
  - to avoid jitter: phase control (requires global clocks)
  - to eliminate the effects of jitter, with sporadic server scheduling (remember: release jitter worsens interference)

Avoiding jitter

Release each task (or message) at specific times in the schedule

- **wait until the worst-case response time before releasing the next task or message**
- **requires global clocks, and OS and network driver cooperation**



Eliminating the effects of jitter

UNIVERSIDAD DE CANTABRIA

Employ Sporadic Server scheduling

- guarantees an execution capacity (C_R)
- at every replenishment period (T_R)

Features

- minimum guaranteed bandwidth for aperiodic events
- bounded interference on lower-priority tasks
  - effects like those of a periodic task
  - reduces the effects of jitter

Not usually available in network schedulers

POSIX specifies sporadic server scheduling, but it has a flaw!

Schedulability analysis of real-time distributed systems

UNIVERSIDAD DE CANTABRIA

1. Introduction
2. Single processor systems
3. Distributed systems
4. ***Schedulability analysis: holistic approach***
5. Schedulability analysis: offset-based approaches
6. Schedulability analysis: EDF
7. Advanced modelling techniques: MAST
8. Conclusion and future work

Holistic analysis technique

UNIVERSIDAD DE CANTABRIA

Mainly developed at the University of York

Each resource (CPU, interconnect) is analyzed separately

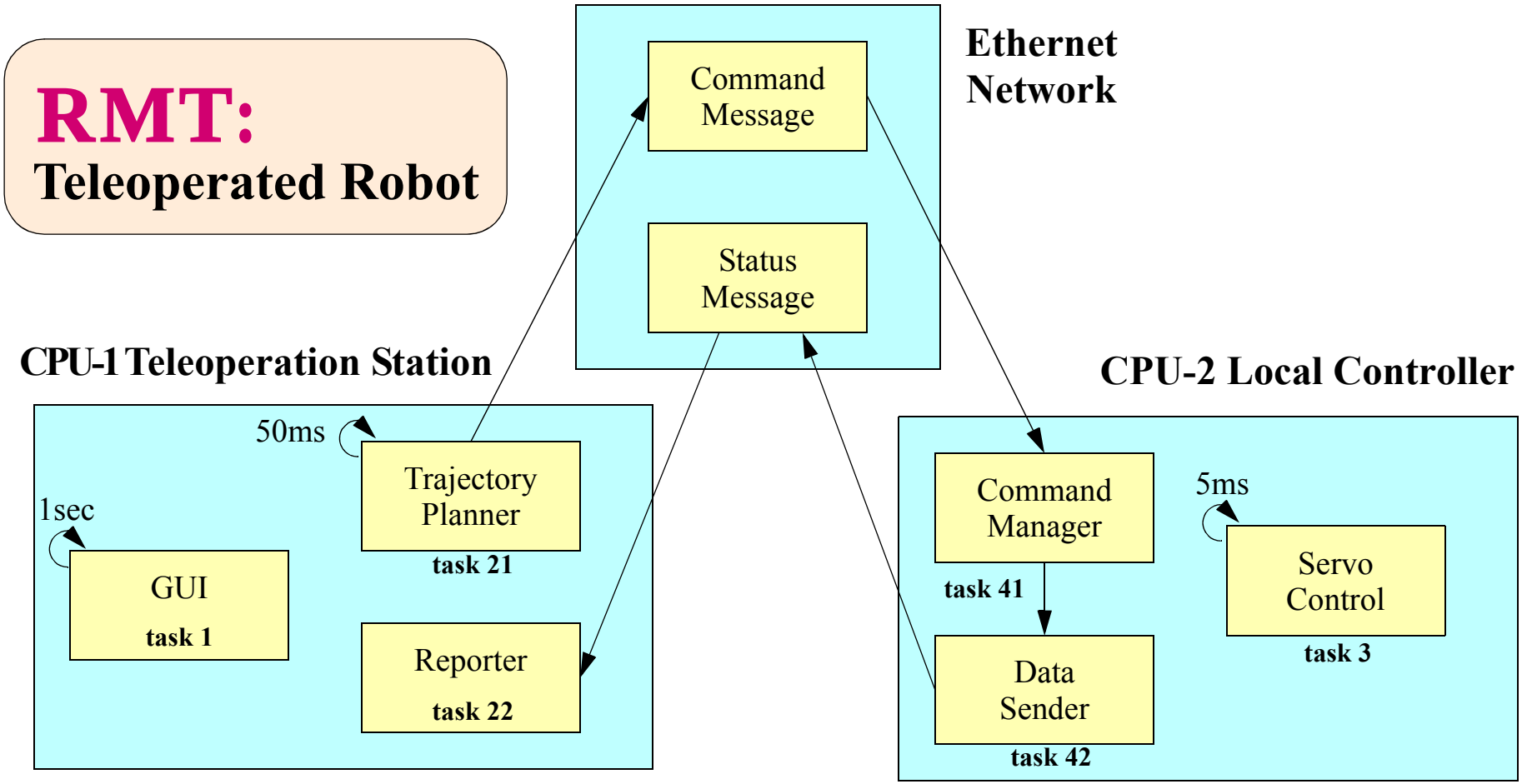
- all activations after the first have *jitter*
- *jitter* in one action is considered equal to the worst-case response time of the preceeding actions
- analysis in one resource affects *response times* in other resources
- the analysis is repeated *iteratively* until a stable solution is achieved
- the method converges because of the *monotonic* relation between jitters and response times

Analysis in the distributed system

```
algorithm WCRT is
begin
    initialize jitter terms to zero
    loop
        calculate worst-case response times;
        calculate new jitters, equal to response
            times of preceding actions
        exit when not schedulable or
            no variation since last iteration;
    end loop;
end WCRT
```

Assumption: $J_i = R_i - R^b_i$, R^b_i = best-case response time=0

Running example



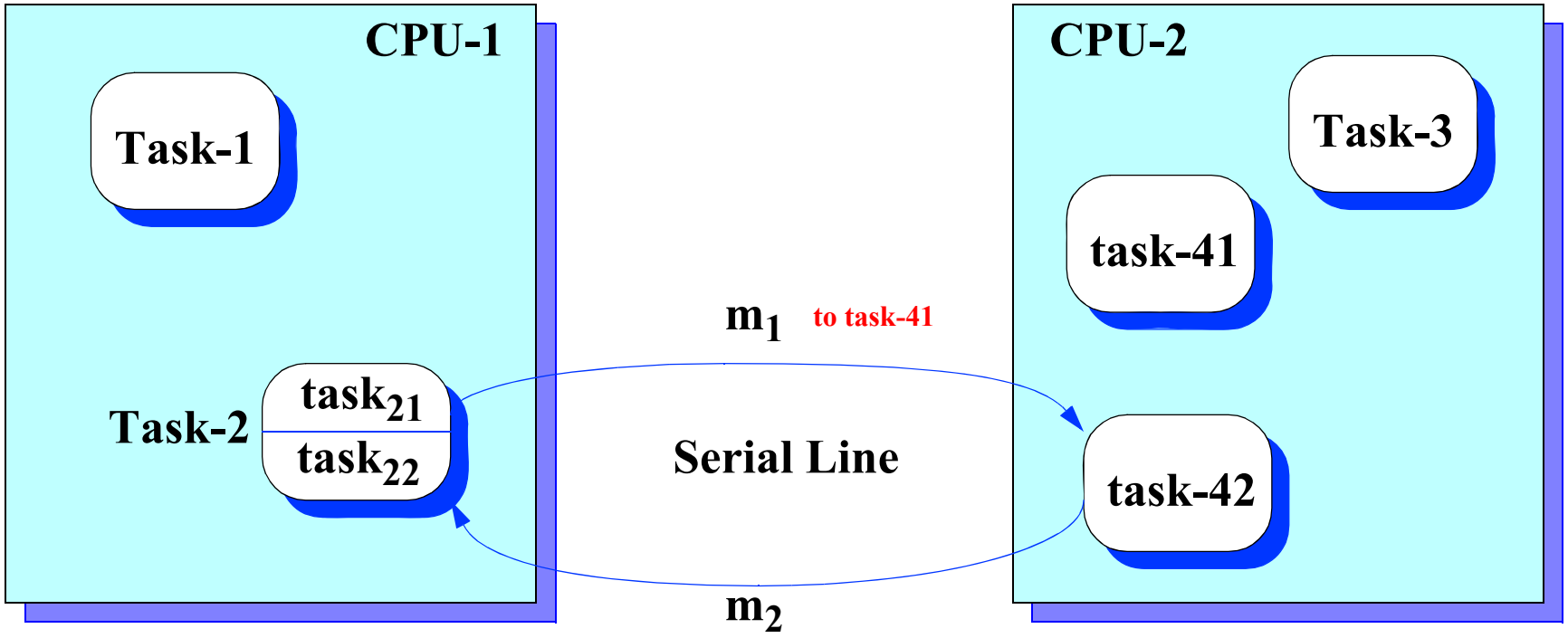
Schedulability analysis of real-time distributed systems

UNIVERSIDAD DE CANTABRIA

1. Introduction
2. Single processor systems
3. Distributed systems
4. Schedulability analysis: holistic approach
5. ***Schedulability analysis: offset-based approaches***
6. Schedulability analysis: EDF
7. Advanced modelling techniques: MAST
8. Conclusion and future work

Pessimism in holistic analysis

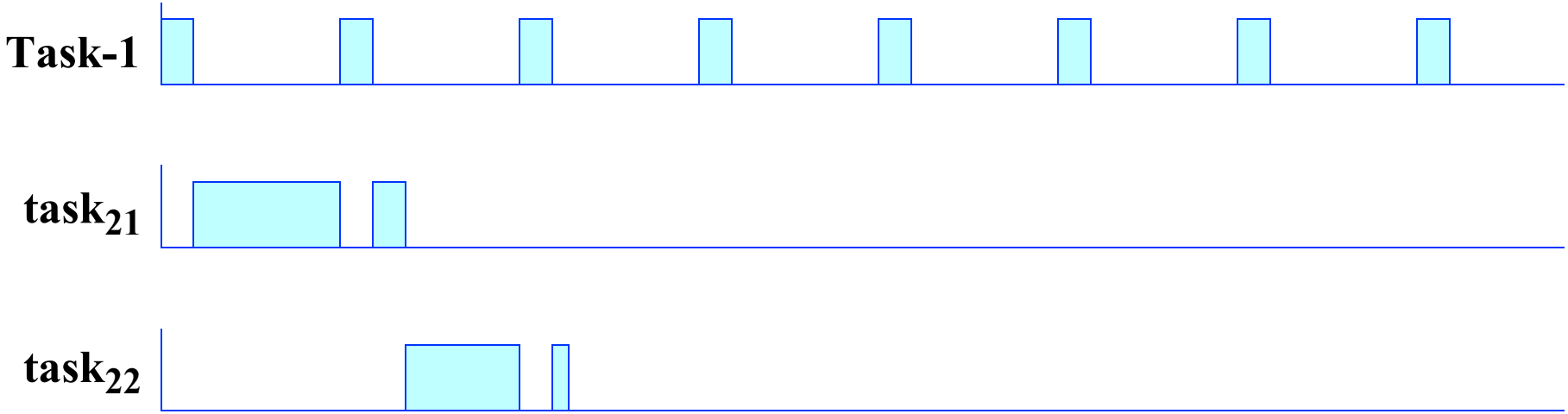
Holistic analysis assumes *independent* task activations in each resource



Independent task analysis

UNIVERSIDAD DE CANTABRIA

Execution timeline for $task_{22}$ on CPU-1 in previous example:

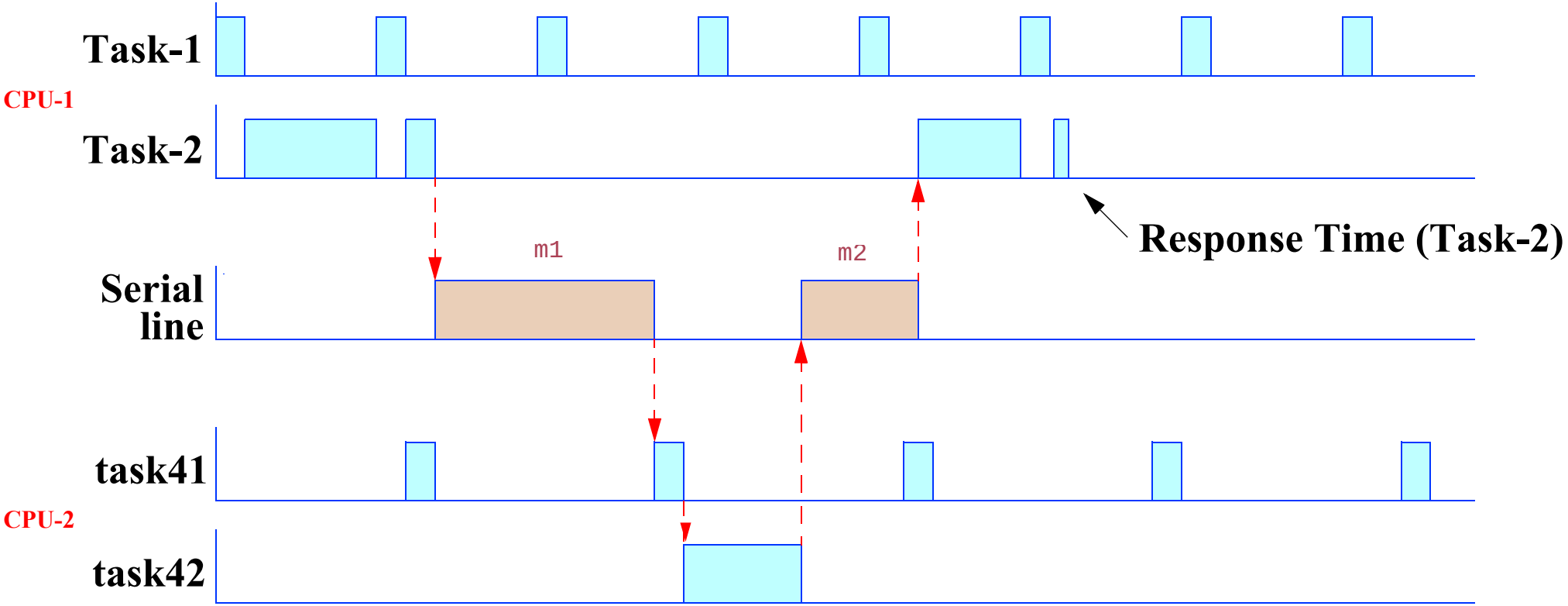


The response time for Task-2:

- includes durations for $task_{21}$, m_1, Task-4, m_2, $task_{22}$
- pessimistic!

Using offsets to reduce pessimism

Execution timeline for analysis of Task-2:



Objectives of the offset-based techniques

UNIVERSIDAD DE CANTABRIA

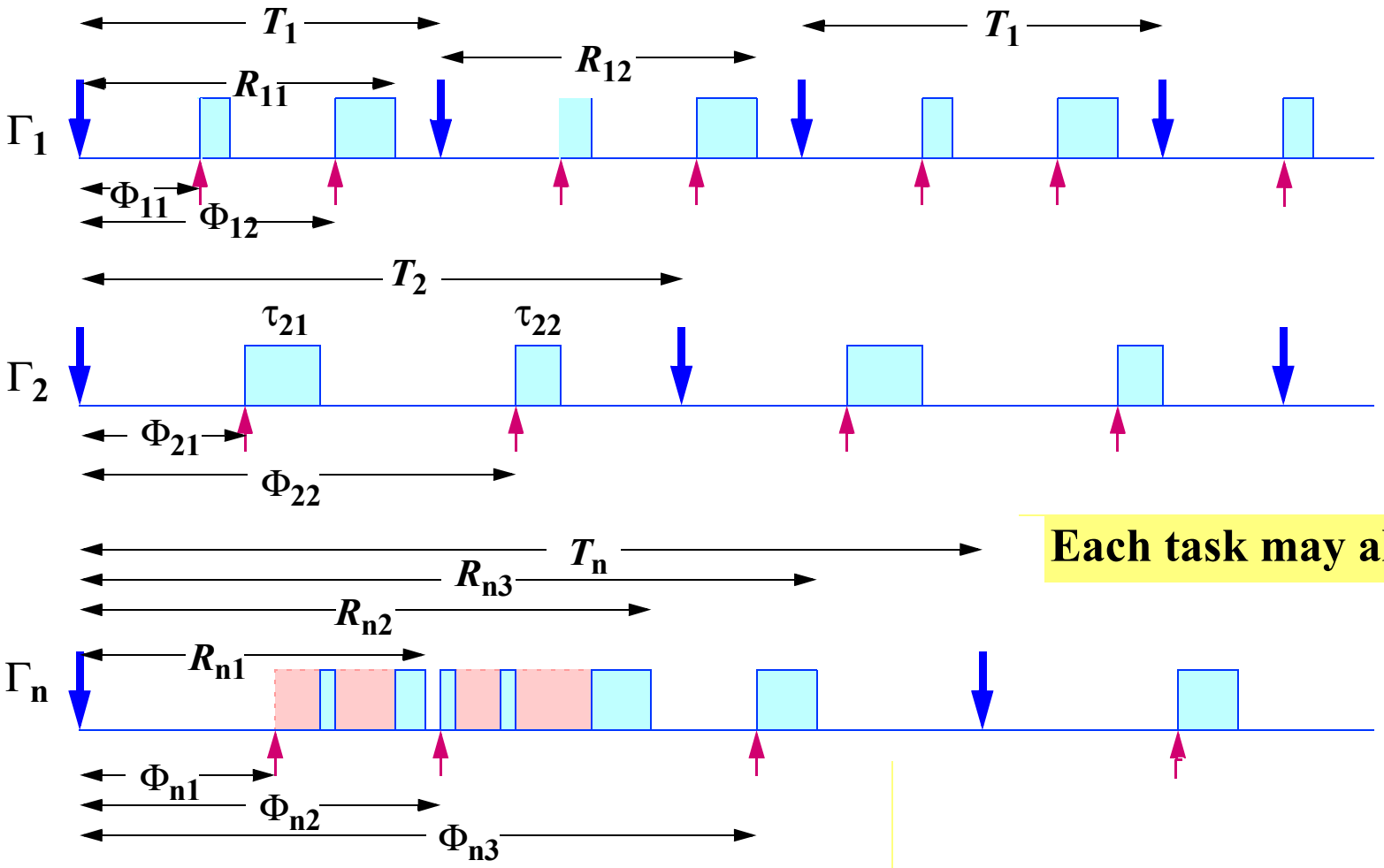
To reduce the pessimism of the worst-case analysis in multiprocessor and distributed systems:

- by considering offsets in the analysis
- offsets can be larger than task periods
  - this is important if deadlines > task periods
- offsets can be static or dynamic
  - offsets are dynamic in distributed systems
  - also in tasks that suspend themselves

This enhancement comes “for free” as there is no change to the application

- although better results can be obtained if best-case execution times are measured

System model with offsets



Each task may also have jitter

Analysis with offsets

UNIVERSIDAD DE CANTABRIA

Developed by Ken Tindell at the University of York

- The exact analysis is intractable: the worst case is not known
- An upper-bound approximation provides good results (equal to exact analysis in 93% of tested cases)

Main limitations

- Offsets are static
  - not applicable to general distributed transactions (precedence-constrained set of actions)
- Offsets are less than the task periods
  - for distributed systems, deadlines would have to be strictly smaller than or equal to the task periods

Extended in Cantabria to dynamic offsets and arbitrary deadlines

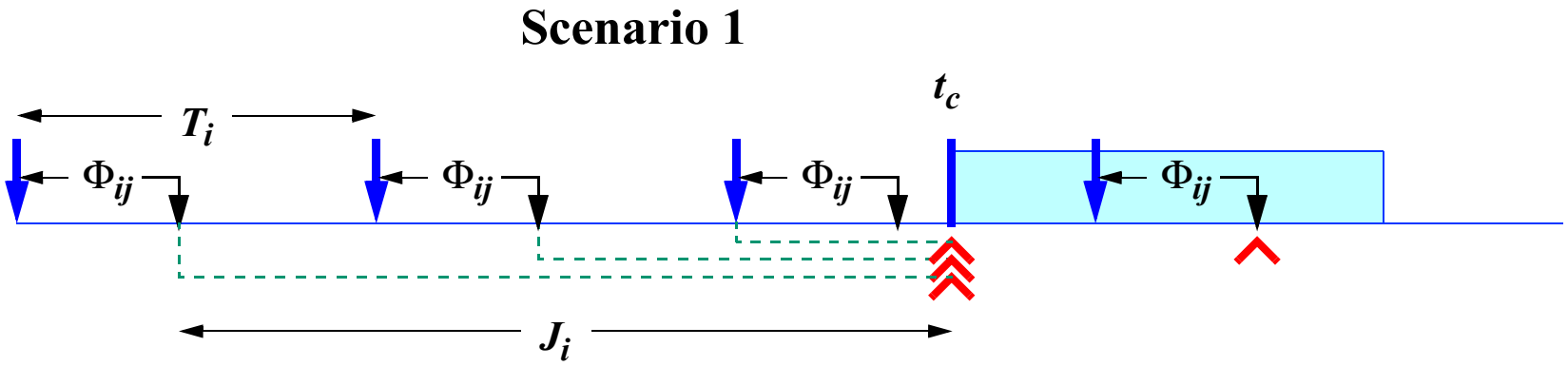
Exact analysis with static offsets

UNIVERSIDAD DE CANTABRIA

Contribution of task τ_{ij} to the response time of lower-priority tasks:

- ***Set 0***: activations that occur before the critical instant and that cannot execute inside the busy period
- ***Set 1***: activations that occur before or at the critical instant, and that may execute inside the busy period
  - **Theorem 1**: worst-case when they all occur at the critical instant
  - **Theorem 2**: worst-case when the first one experienced its maximum jitter
- ***Set 2***: activations that occur after the critical instant
  - **Theorem 1**: worst-case when they have zero jitter

Scenario for calculating the worst-case contribution of τ_{ij}



Theorem 2: job j of task i experiences maximum jitter so that it can cause interference

Upper-bound approximation for worst-case analysis

UNIVERSIDAD DE CANTABRIA

Exact analysis is intractable

- The task that generates the worst-case contribution of a given transaction is unknown
- The analysis has to check all possible combinations of tasks

Tindell developed an upper-bound approximation

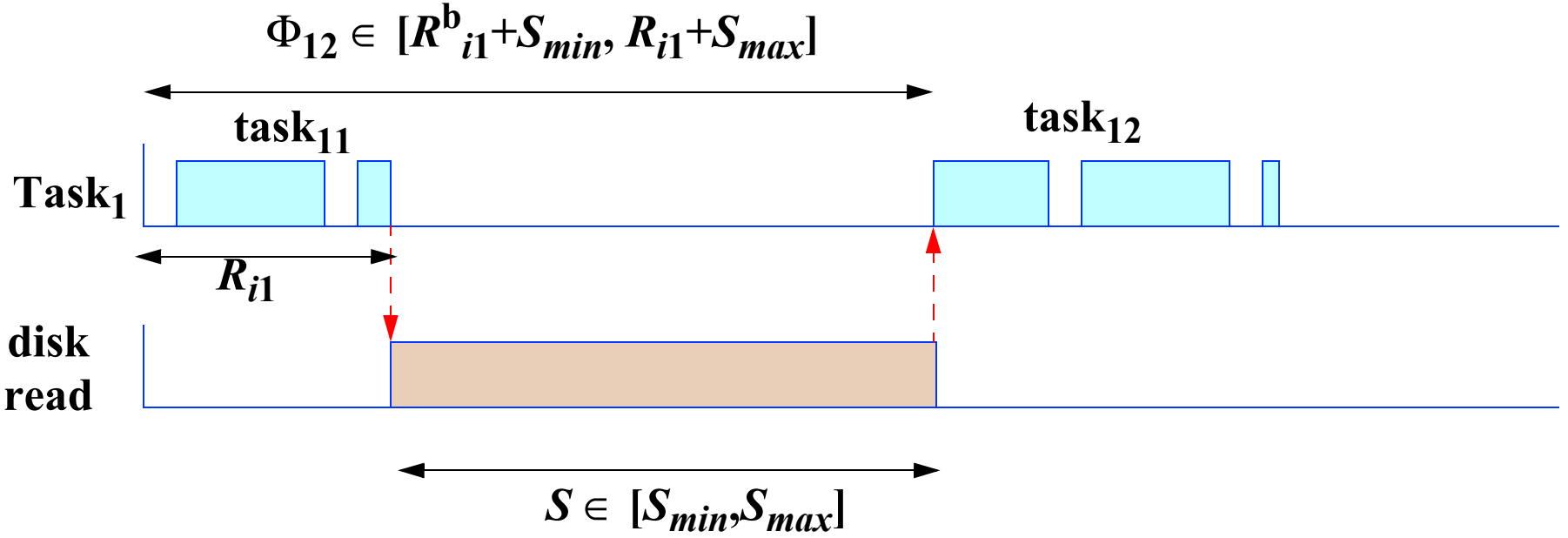
- For each transaction we consider a function that is the maximum of all the worst-case contributions, assuming each task of the transaction to initiate at the critical instant
- This technique is pessimistic, but pseudo-polynomial
- In 93% of the tested cases, the response times were exact

Analysis with dynamic offsets

In many systems the offset may be dynamic:

$$\Phi_{ij} \in [\Phi_{ij,min}, \Phi_{ij,max}]$$

Example: task with a suspending operation



Analysis with dynamic offsets (cont'd)

UNIVERSIDAD DE CANTABRIA

Dynamic offsets can be modeled as static offset + variable jitter

- **Equivalent static offset:**

$$\Phi'_{ij} = \Phi_{ij,min}$$

- **Equivalent variable jitter:**

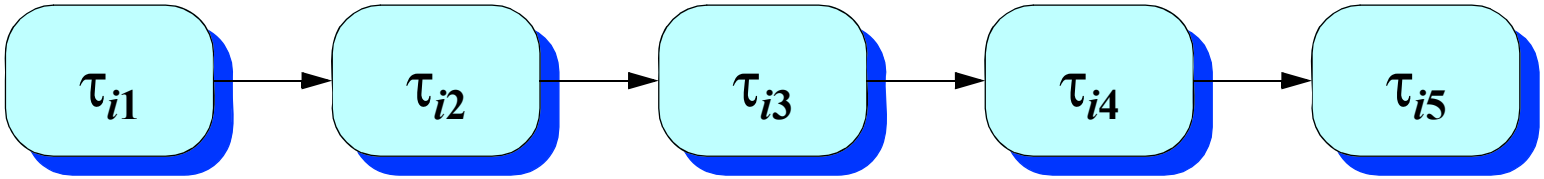
$$J'_{ij} = J_{ij} + (\Phi_{ij,max} - \Phi_{ij,min})$$

The problem is that now offsets depend on response times and response times depend on offsets

- **The solution is to apply the analysis iteratively, starting with response times = zero, until a stable solution is achieved**
- **We call this algorithm WCDO (Worst-Case Dynamic Offsets)**

UNIVERSIDAD DE CANTABRIA

Distributed transaction Γ_i



Dynamic offsets in distributed transactions can be modeled with static offset and variable jitter:

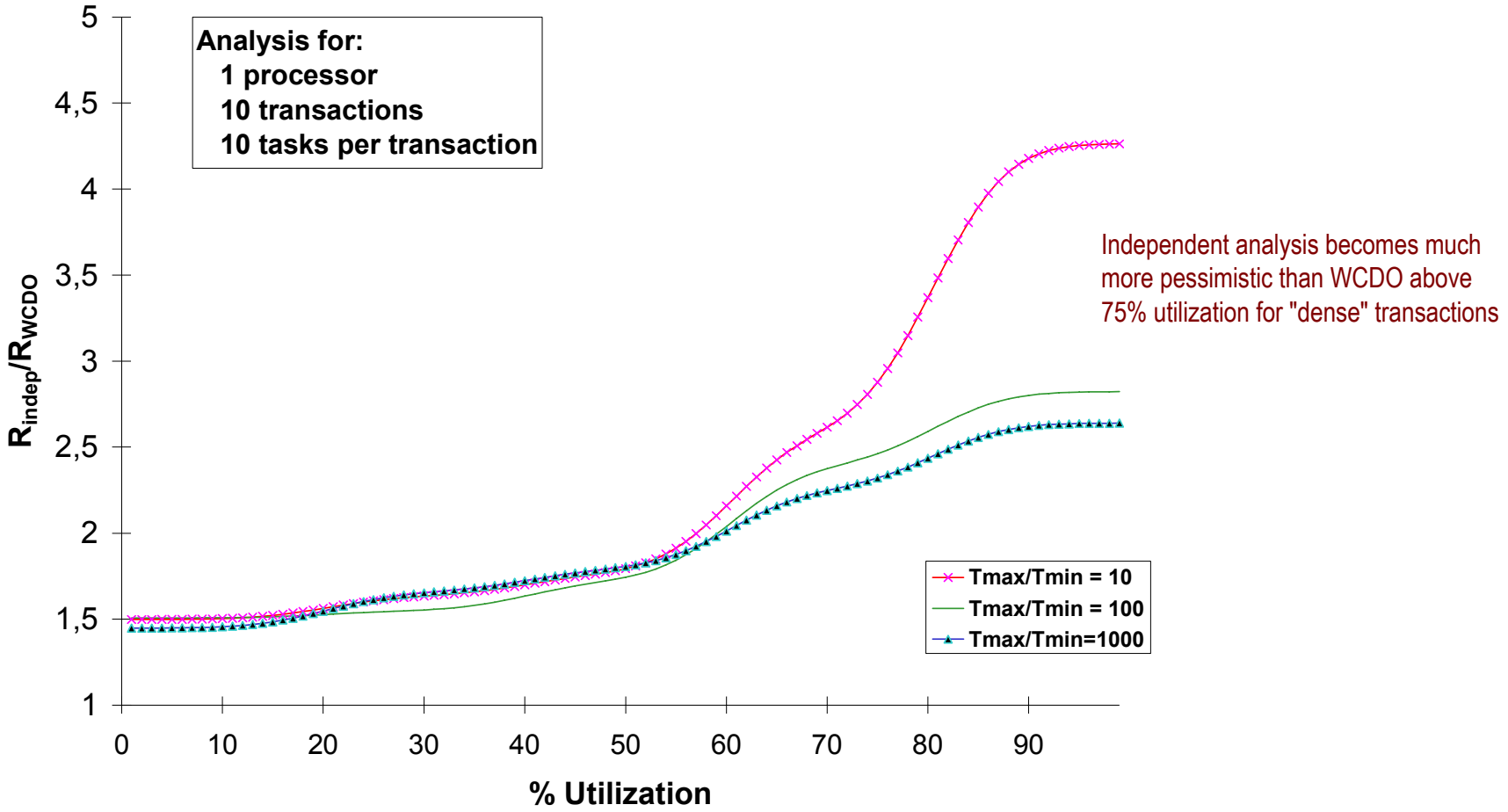
- **Equivalent static offset:**

$$\Phi'_{ij} = \Phi_{ij,min} = R^b_{ij-1}$$

- **Equivalent variable jitter ("own" jitter = 0):**

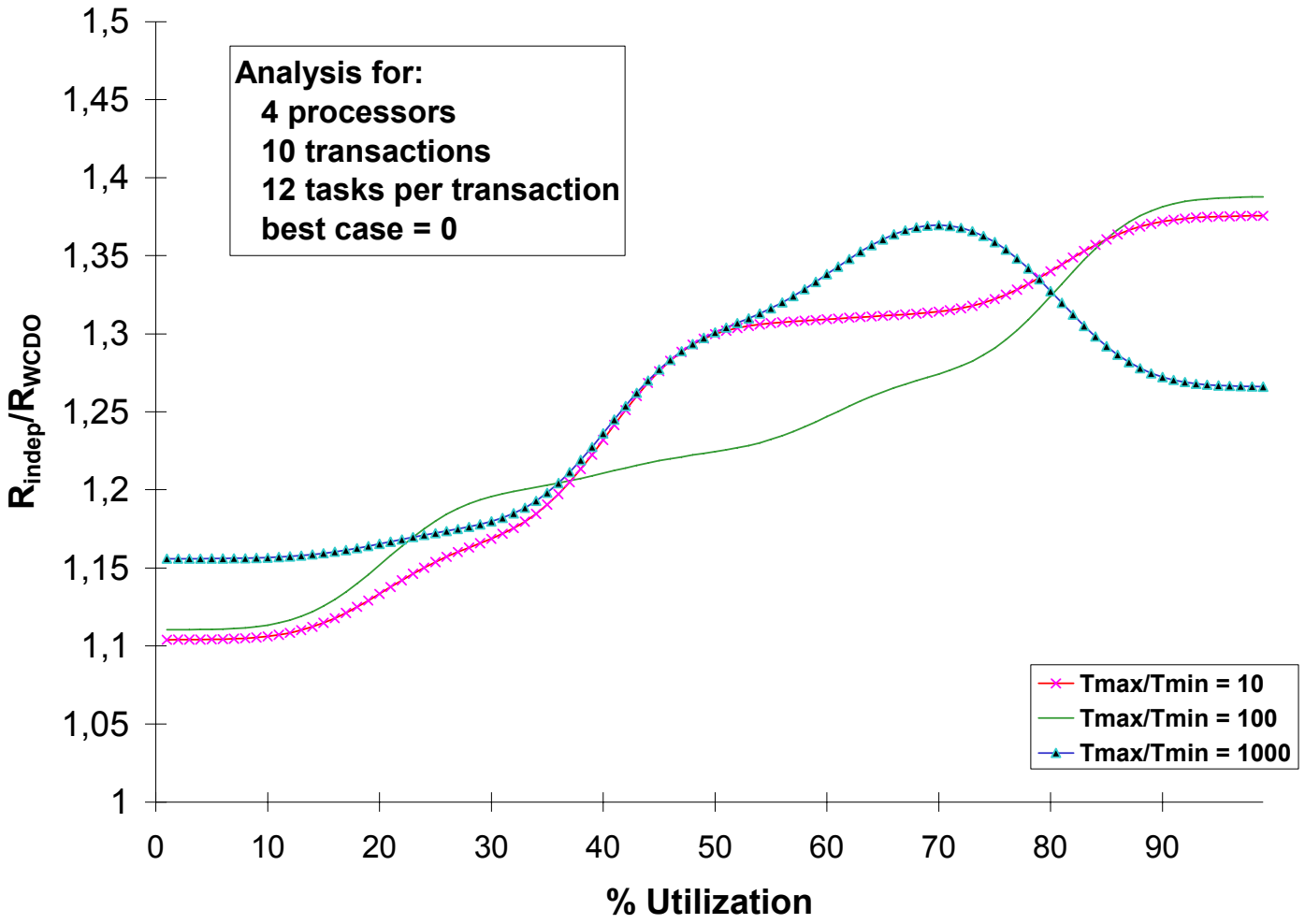
$$J'_{ij} = J_{ij} + (\Phi_{ij,max} - \Phi_{ij,min}) = R_{ij-1} - R^b_{ij-1}$$

Comparison with existing technique: 1 processor

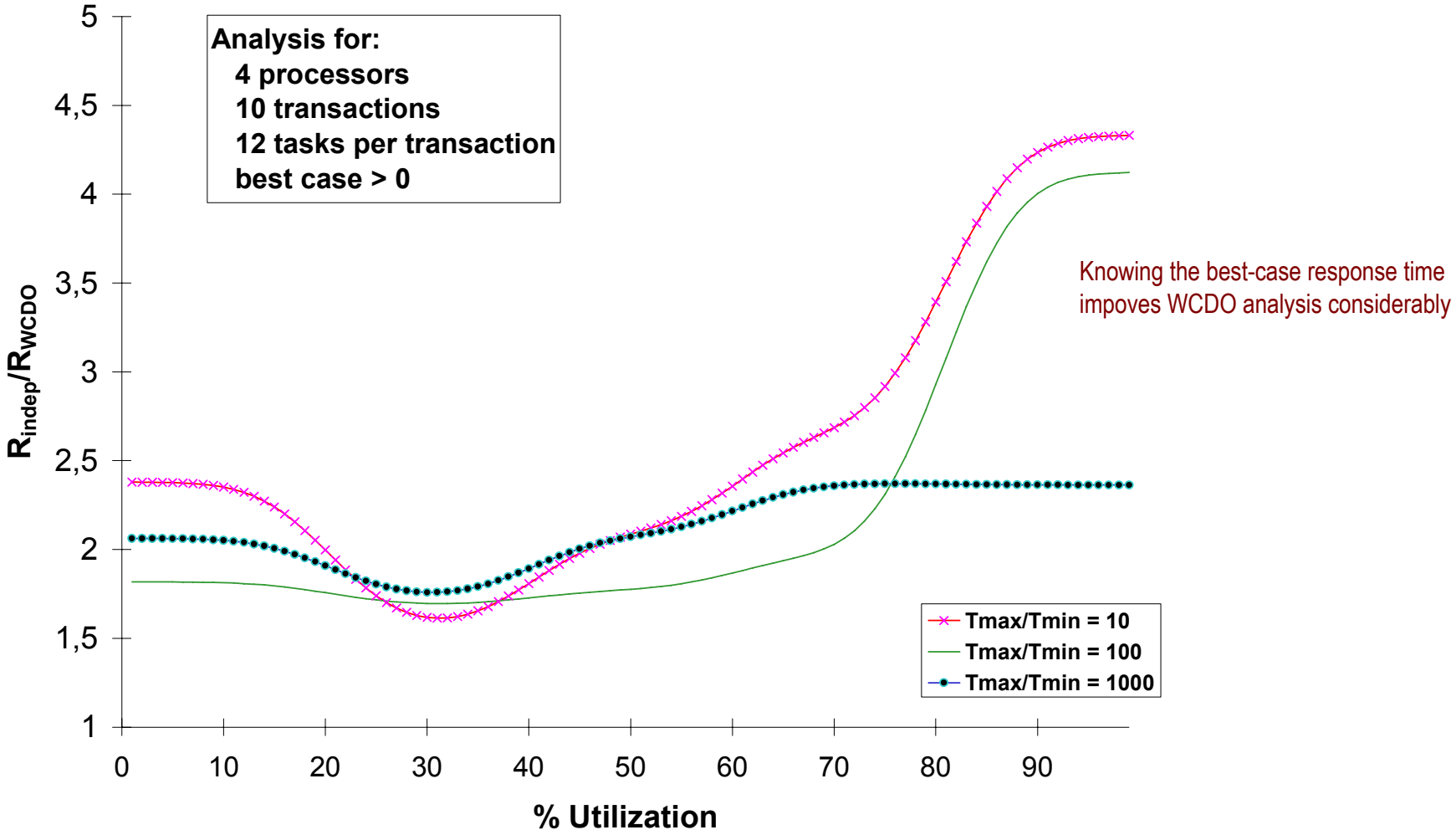


Independent analysis becomes much more pessimistic than WCDO above 75% utilization for "dense" transactions

Comparison with four processors, and best-case=0

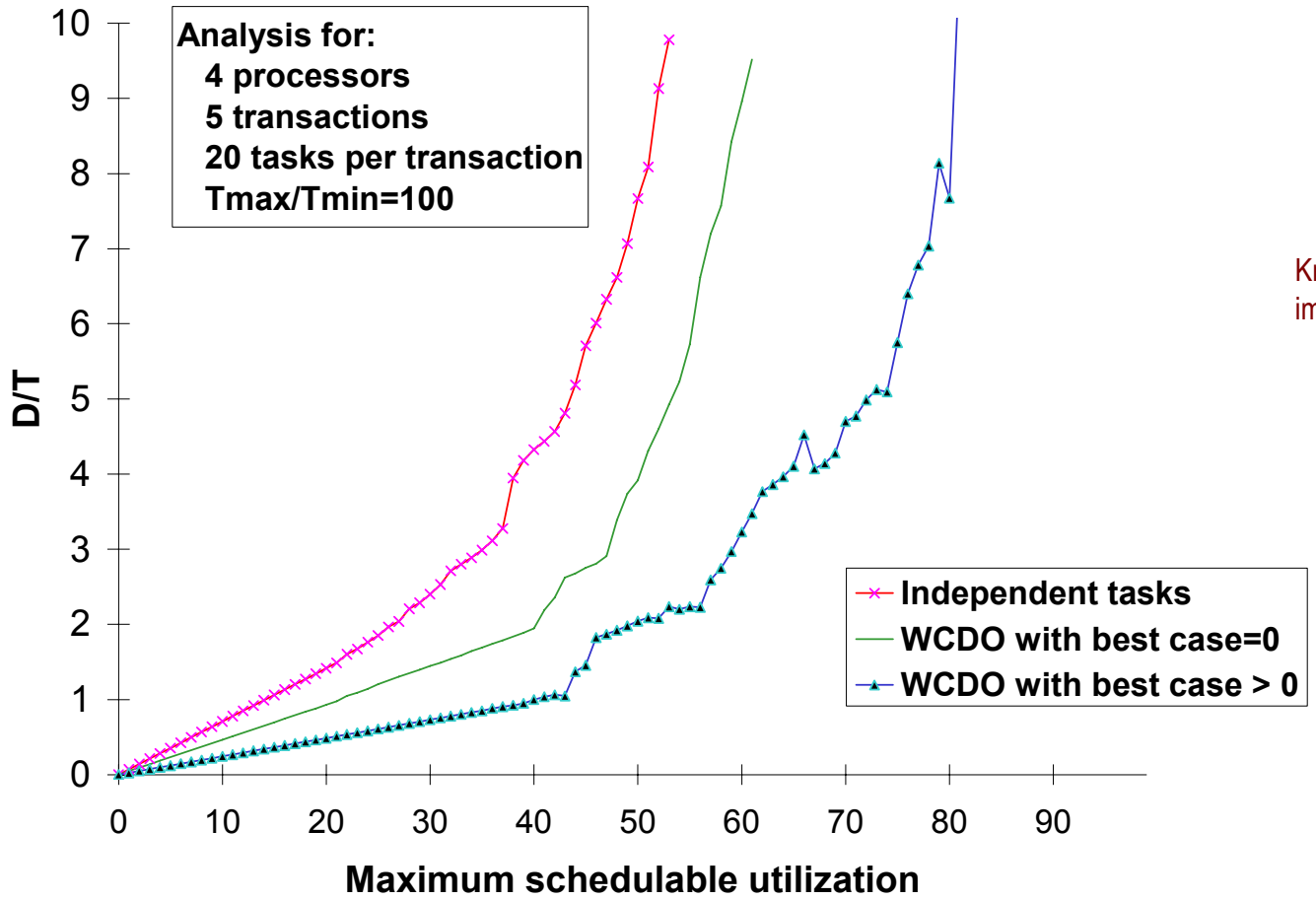


Comparison with four processors and best-case>0



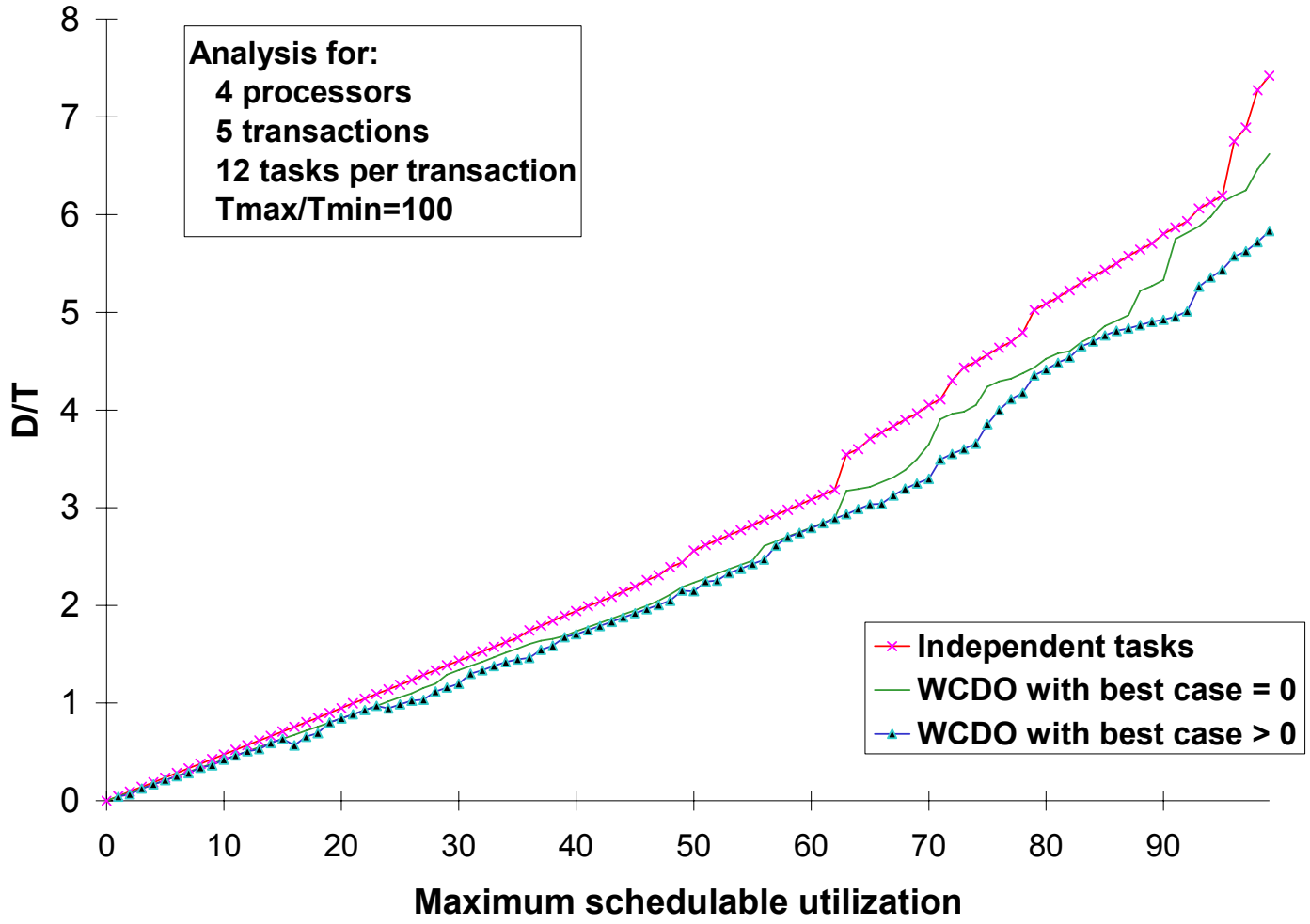
Knowing the best-case response time impoves WCDO analysis considerably

Maximum schedulable utilization with 20 tasks per transaction



Knowing the best-case response time impoves WCDO analysis considerably

Maximum schedulable utilization with 12 tasks per transaction



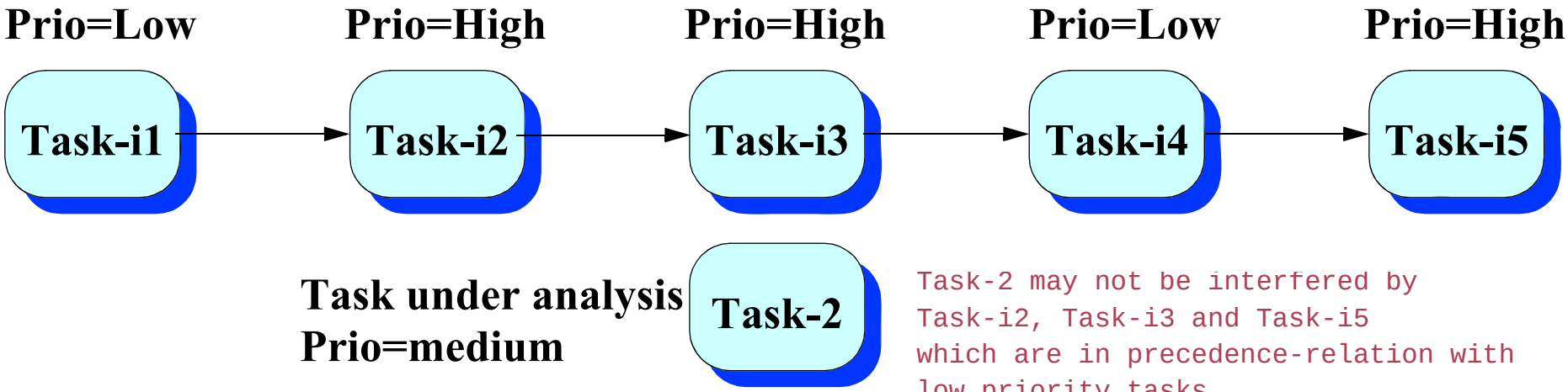
Room for improvement in offset-based analysis

UNIVERSIDAD DE CANTABRIA

Offset-based analysis produces results that are much less pessimistic than holistic analysis

But offset-based analysis still has room for improvement

- **a high-priority task that is preceded by a low-priority task may not be able to execute before a medium-priority task under this analysis**



Objectives of optimized offset-based analysis

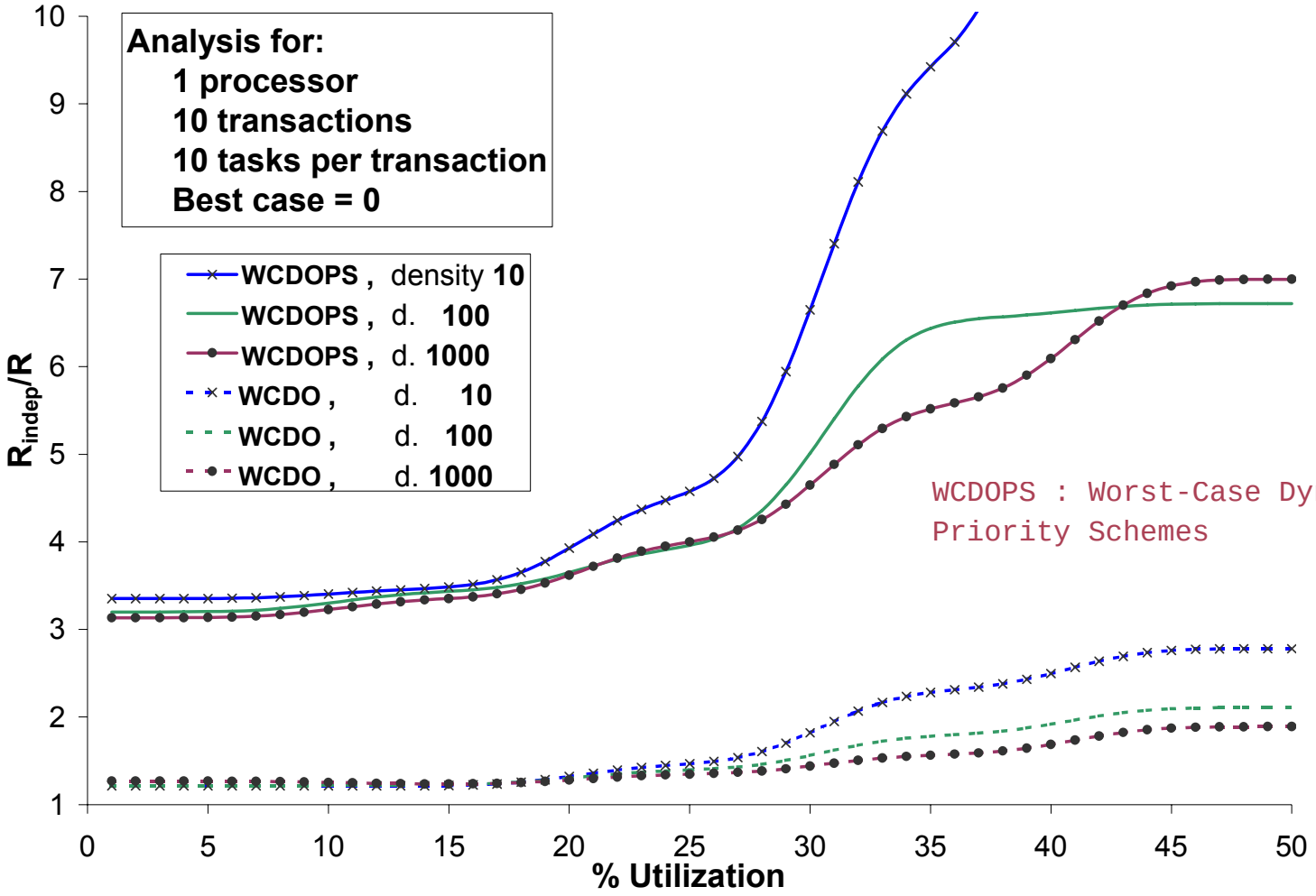
UNIVERSIDAD DE CANTABRIA

To enhance the offset-based schedulability analysis by:

- Eliminating from the analysis the effects of higher priority tasks that cannot execute due to precedence constraints
- Eliminating the effects of the tasks that are preceded by the task under analysis

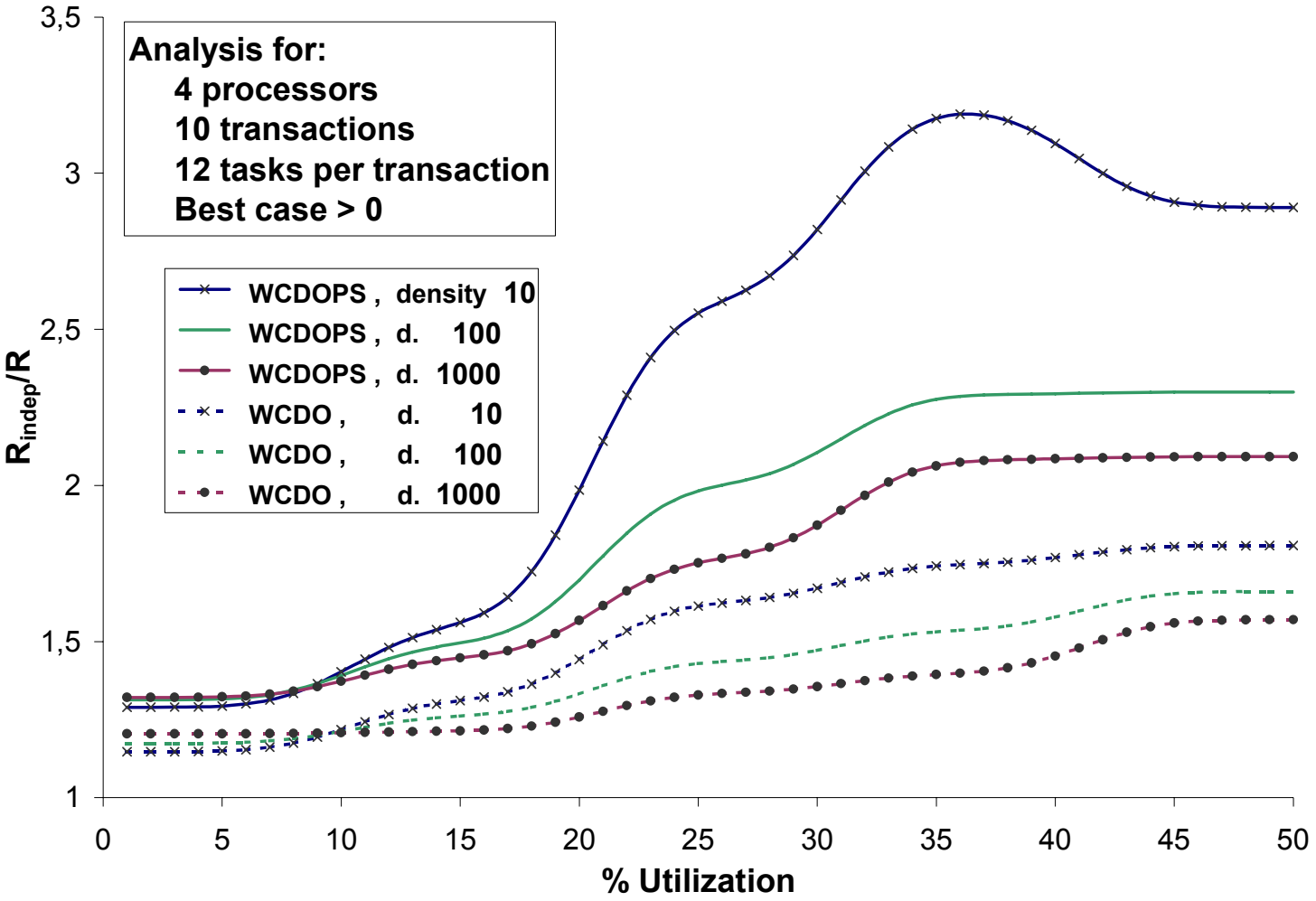
These enhancements reduce much of the pessimism in the analysis of distributed systems

Simulation results: response times

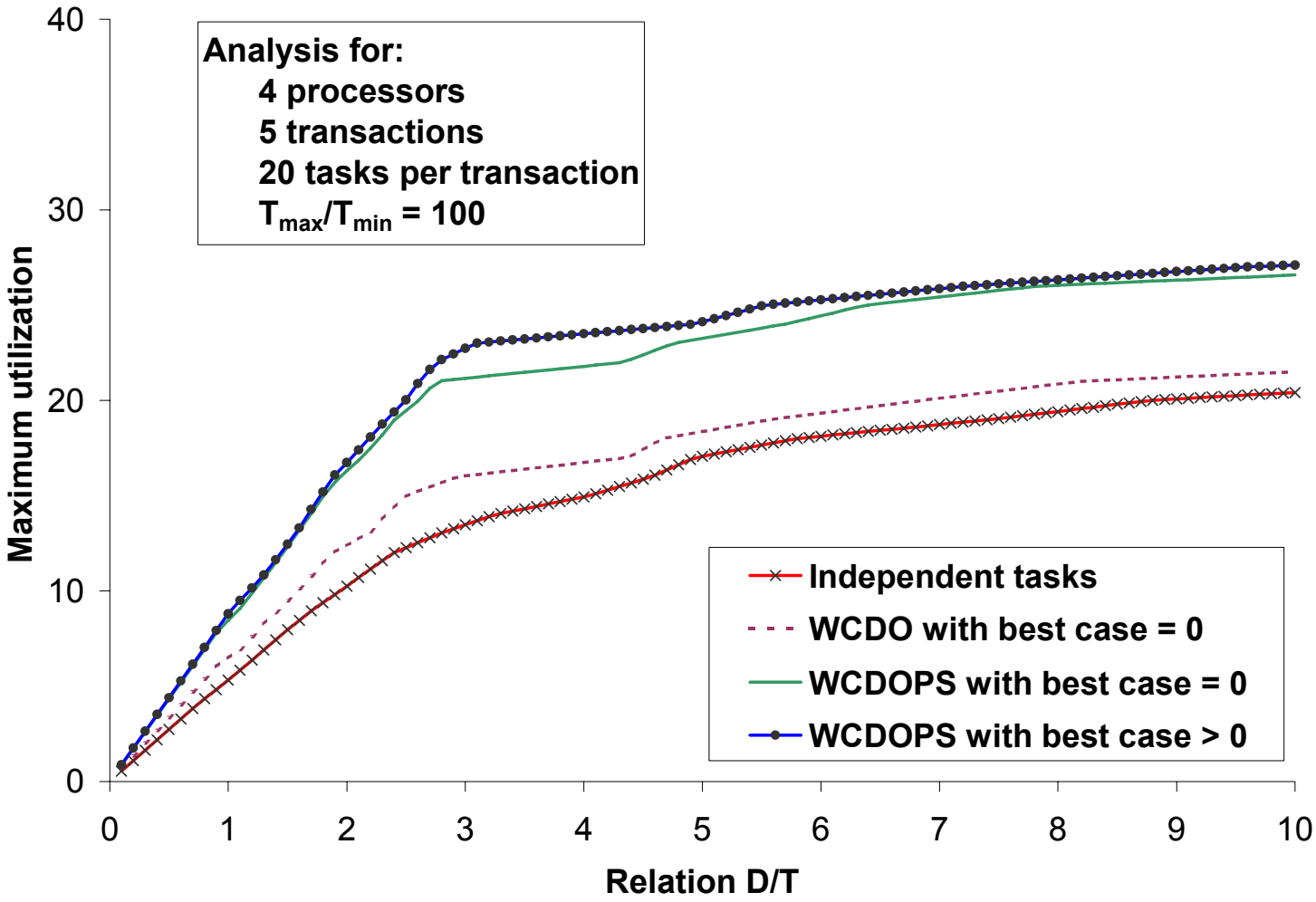


WCDOPS : Worst-Case Dynamic Offsets with Priority Schemes

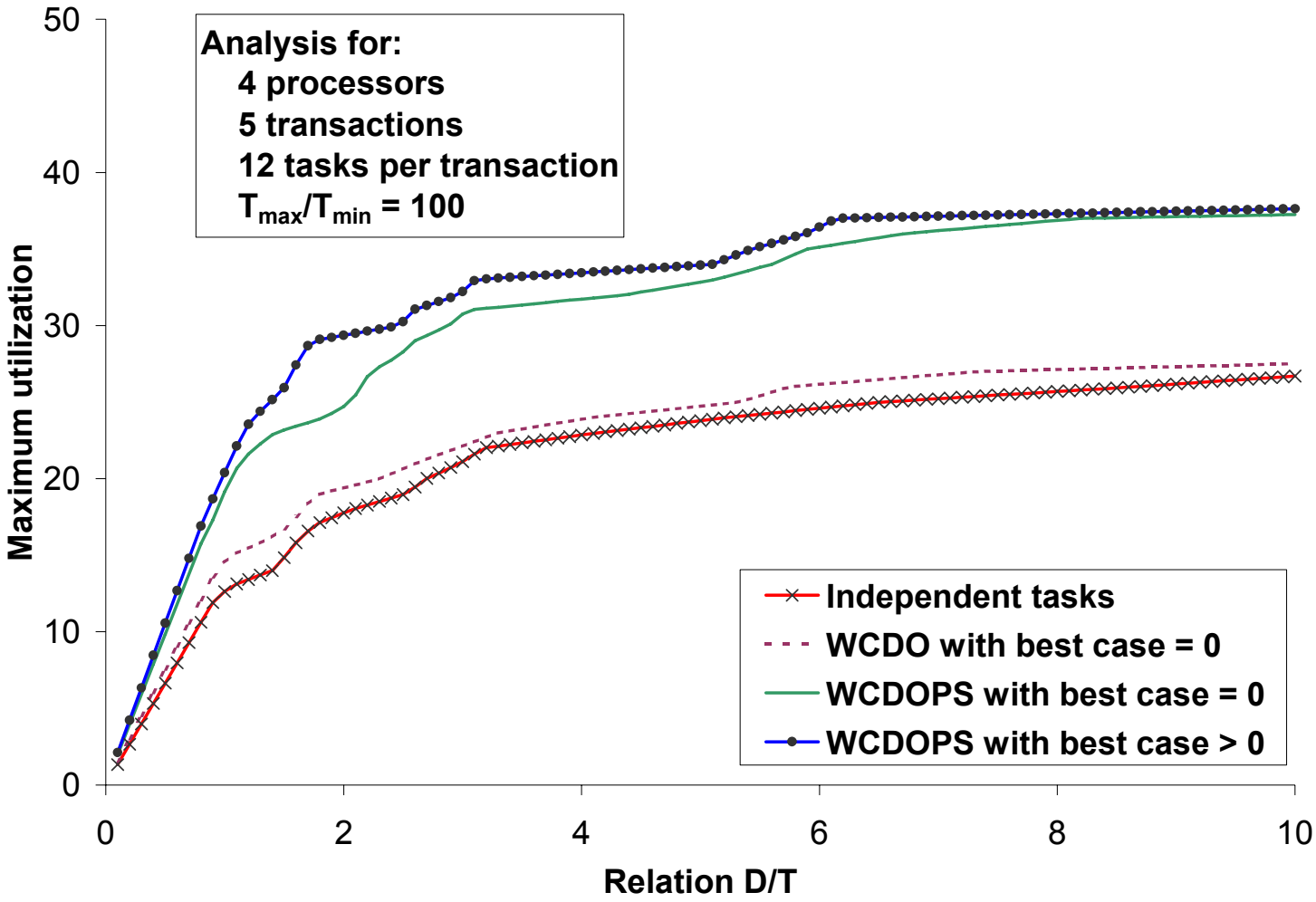
Simulation results: response times



Simulation results: utilization



Simulation results: utilization



Priority assignment techniques

UNIVERSIDAD DE CANTABRIA

No known optimum priority assignment

Simulated Annealing:

- Standard optimization technique
- Finds solutions by successively exchanging the priorities of a pair of tasks, reapplying the analysis to determine if results are worse or better
- The probability of the change surviving is a function of the results
- Does not guarantee finding the solution

Priority assignment techniques (cont'd)

UNIVERSIDAD DE CANTABRIA

HOPA (Heuristic Optimized Priority Assignment)

- Heuristic algorithm based on successively applying the analysis
- Much faster than simulated annealing
- Usually finds better solutions
- Does not guarantee finding the solution