Improving arrival curves for flow splitting

Authors: Edoardo Loni, Raffaele Zippo, Giovanni Stea, Matteo Andreozzi (Arm)

Presenter: Edoardo Loni

University of Pisa

WoNeCa 2025 – NTNU, Trondheim

Motivation – timing predictability in RT systems

Ensuring predictable time behaviour is crucial to meet safety and performance requirements

Challenge

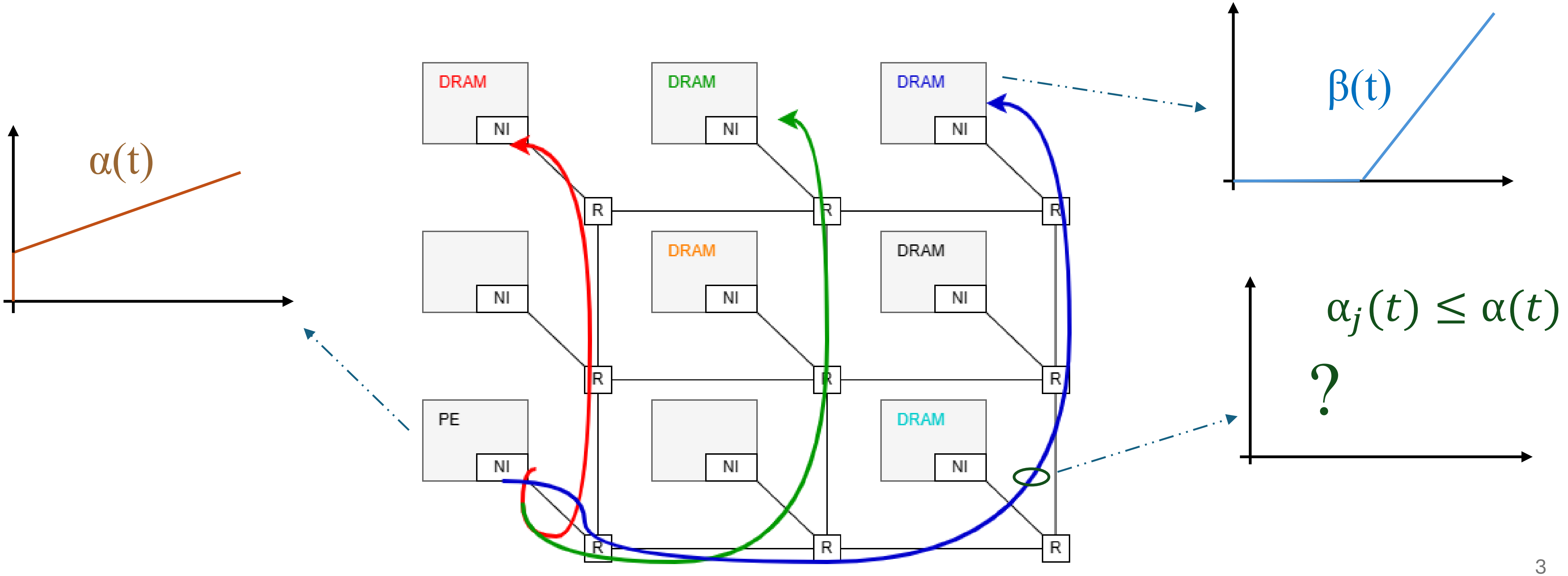
- Data or service requests are heterogeneous
    - Multiple targets (e.g. memory banks, cores)
    - Different paths

Objective

- Finer-grained traffic analysis

Case study

Address-aware splitting towards DRAM banks connected to different NoC nodes



Related work – 1/3

Mao et al. – Multipath delay minimization

Approach

- Use DNC to minimize end-to-end delay in multipath transport
- Traffic split via **leaky buckets:**

$$\alpha = (\sigma, \rho) \rightarrow \alpha_1 = (\sigma_1, \rho_1), \alpha_2 = (\sigma_2, \rho_2)$$

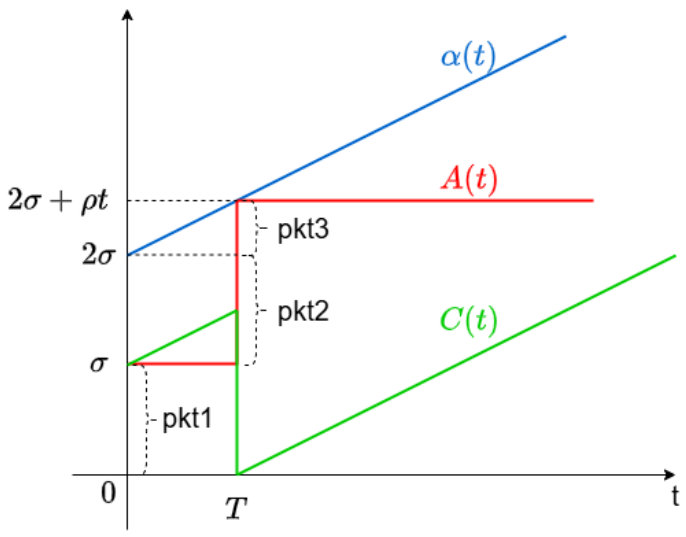
such that

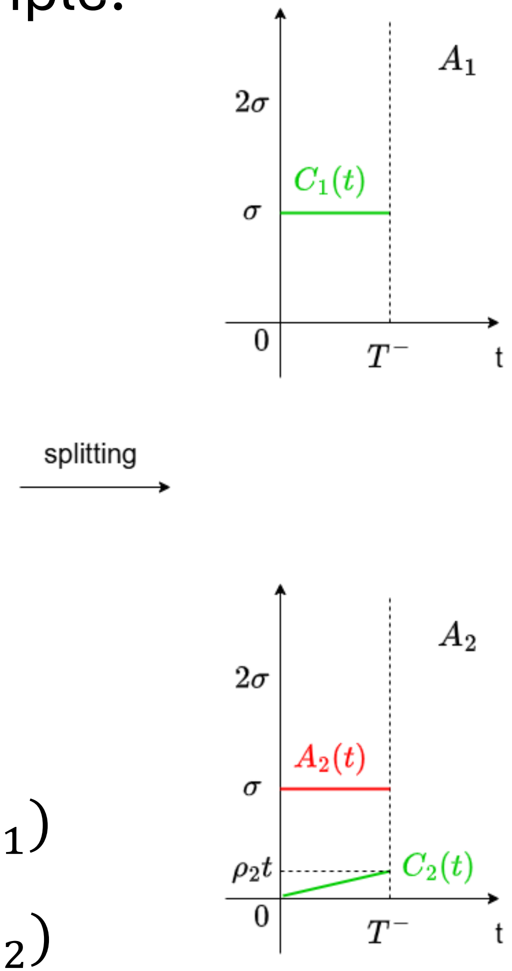
$$\sigma_1 + \sigma_2 = \sigma, \rho_1 + \rho_2 = \rho$$

This assumption is not always valid

Related work – 2/3

Disproving counterexample:





$$\alpha = (2\sigma, \rho) \begin{cases} \alpha_1 = (\sigma, \rho_1) \\ \alpha_2 = (\sigma, \rho_2) \end{cases}$$

At time $t = 0$:

- *Pkt1* goes to subflow A_2

At time $t = T^-$:

- Total credits:

$$C = \sigma + \rho T$$

- Subflow credits:

$$\left.\begin{aligned} C_1 &= \sigma \\ C_2 &= \rho_2 T < \rho T \end{aligned}\right\} C_1 + C_2 < C$$

Related work – 3/3

Key observation

- Sum of subflow credits smaller than total credit

Not all α-shaped flows can be split into valid α_i-shaped subflows

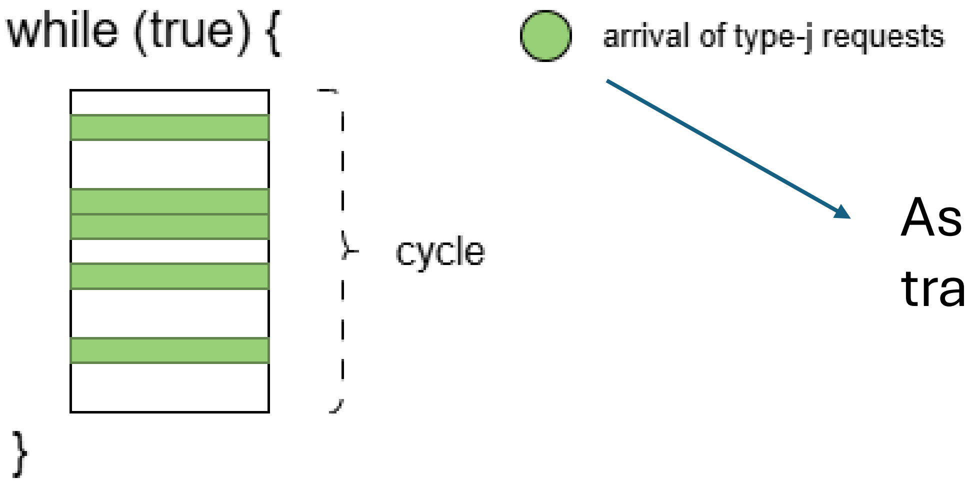
Conclusion

- Mao et al. rely on an incorrect partitioning assumption
- Our method does not require traffic to conform to decomposable leaky buckets
- Instead, we derive tighter bounds by analyzing spatial structure

System model – key idea

Underlying intuition

➢ Consider a processing element which repeatedly executes a task, generating requests



As an example, type-j requests can be memory transaction requests towards a specific DRAM bank

We can define a **cycle** as the sequence of requests issued during an iteration of the task execution

System model – formal definitions (1/2)

T: computational entity that executes instances of a task, generating a sequence of **requests**, or packets

Cycle: ordered sequence of requests issued in a task execution

$$C_k = (r_1, r_2, \dots, r_{n_k})$$

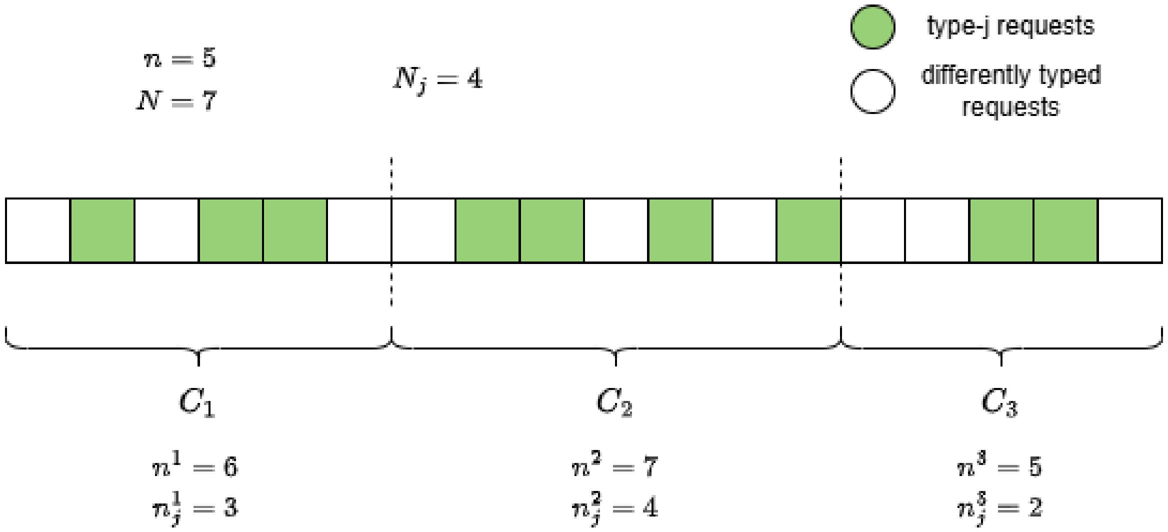
Request type: $\tau(r_i)$

Type-j subsequence: $C_j^k = \left(r_{j_1}, r_{j_2}, \dots, r_{j_{n_j^k}}\right)$

System model – formal definitions (2/2)

Traffic flow: ideally infinite sequence of cycles

$$\Gamma = (C_1, C_2, C_3, \dots)$$



Known parametres:

- n, N: bounds on the number of per-cycle requests
- N_j: max number of type-j requests

- α(t): sub-additive arrival curve for the cumulative traffic flow

Method – key insights (1/2)

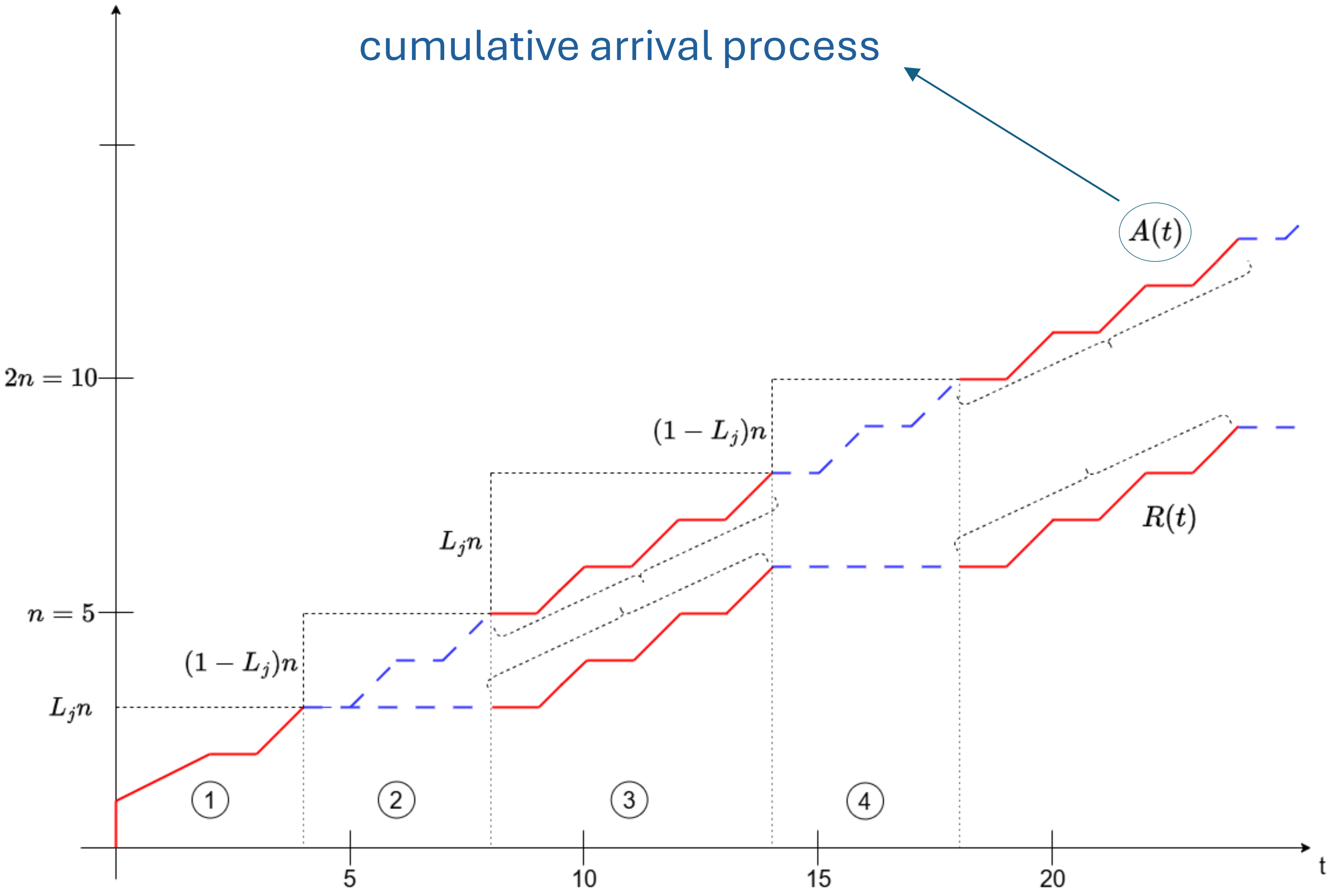
$\boldsymbol{\alpha_j(t)}$: arrival curve for the subflow of interest. It can be computed as:

$$\alpha_j(t) = R(t) \oslash R(t)$$

upper bound to the max arrival process for the subflow

- When $L_j = \frac{N_j}{n} < 1$, each cycle **must** contain non type-j packets
- Use this to bound spacing between type-j packets

Method – key insights (2/2)



Infer refined bounds by analyzing:

- temporal behaviour, via $A(t)$

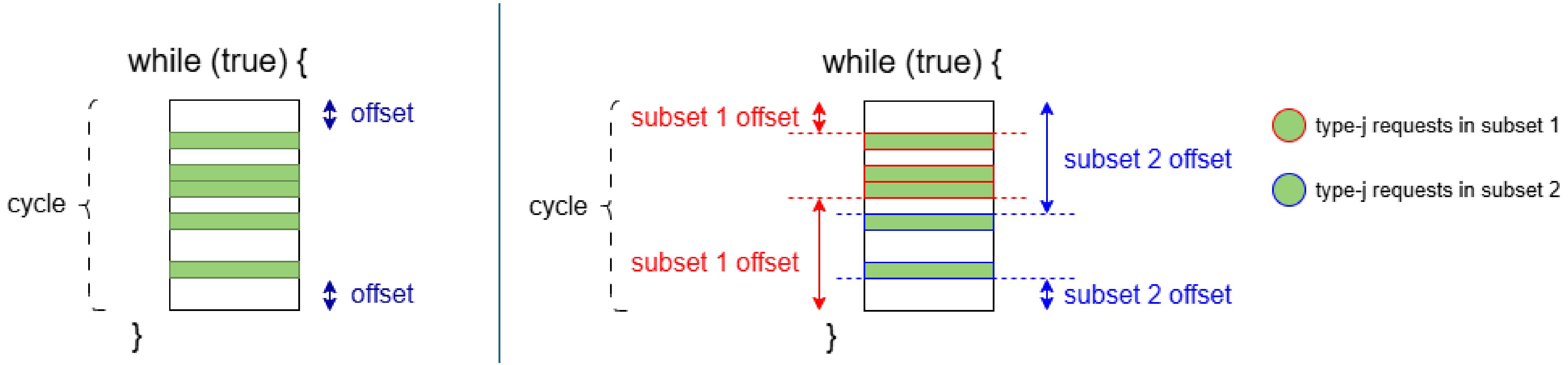
  must be the maximum one

- spatial arrangement of packets

  must be the worst-case one

Method – adding further system knowledge

Introduction of **offset** and **subsets** to model non-periodic packet behaviour within a cycle



- ➢ Knowing lower bounds on their size can help us to model the spatial arrangement of packets in a more accurate way

Method – defined versions

Version	Required Knowledge	Main Features
Basic	n, N, N_j	Uses type-j packets count per cycle
Offset-aware	d_j, a_j: min gaps before/after type-j, packets arrival in a cycle	Refines inter-cycle spacing via spatial layout
Subset-aware	Partition of type-j packets into subsets, each with L_j^k, d_j^k, a_j^k	Tracks subset layout for precise type-j packets positioning

- The three versions differ in how the worst-case spatial arrangement of type-j packets is computed

$\mathbf{J_M(b)}$: max number of type-j packets within the first b ones in the cumulative flow

Results – 1/5

- WCD analysis performed on a numerical example using the proposed splitting method

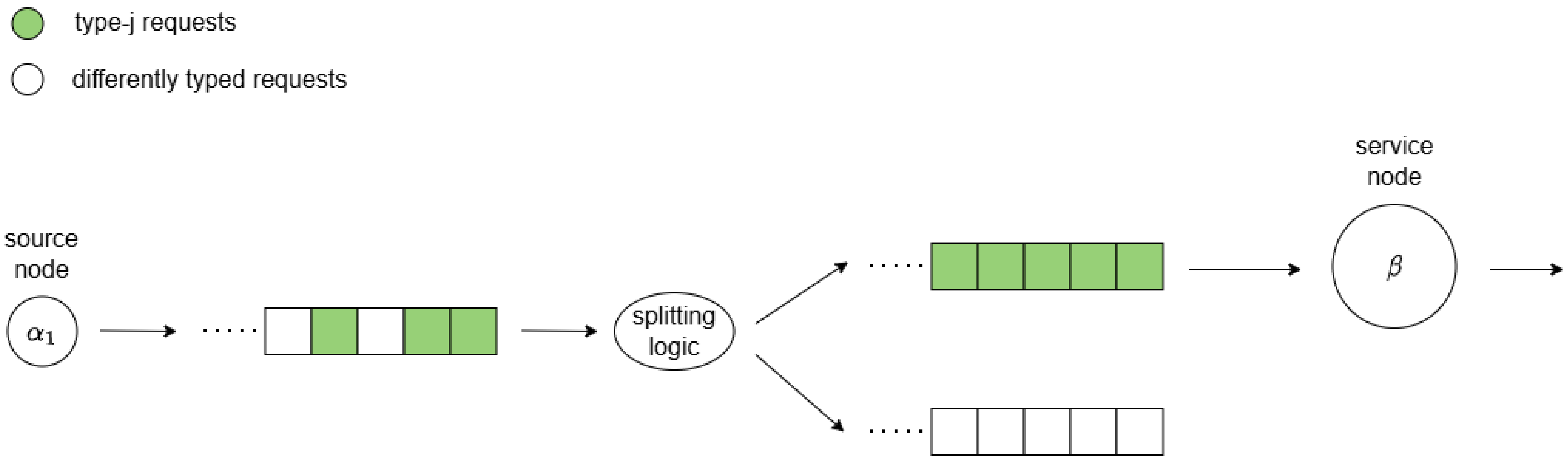
Parametre setup

Subset	L_{i_j}	d_{i_j}	a_{i_j}
Subset 1	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{5}{6}$
Subset 2	$\frac{1}{6}$	$\frac{9}{10}$	$\frac{1}{4}$

$n = 15, N = 25$

Results – 2/5

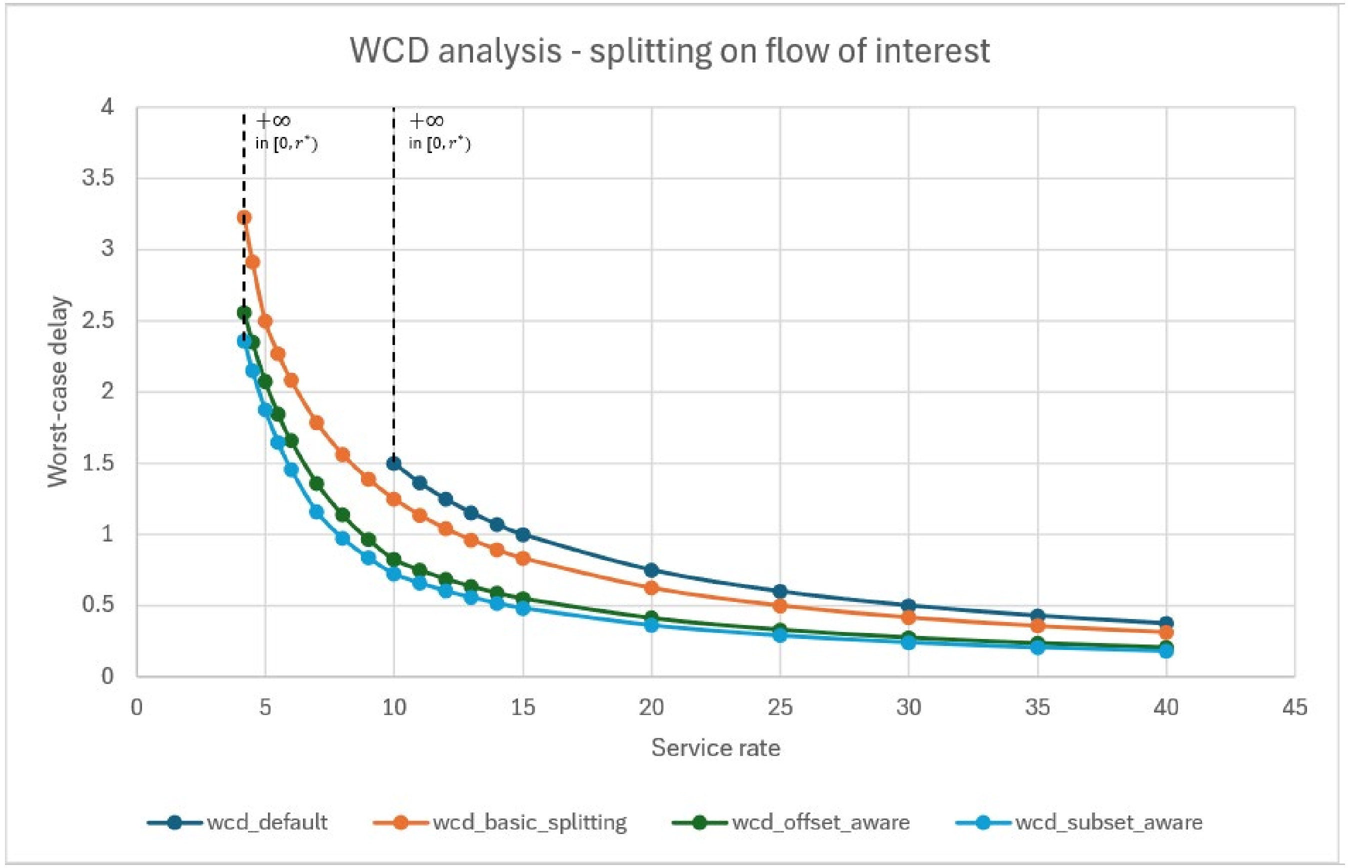
Scenario 1 – splitting on subflow of interest



➢ We are interested in computing WCD for type-j packets being served by node β

Results – 3/5

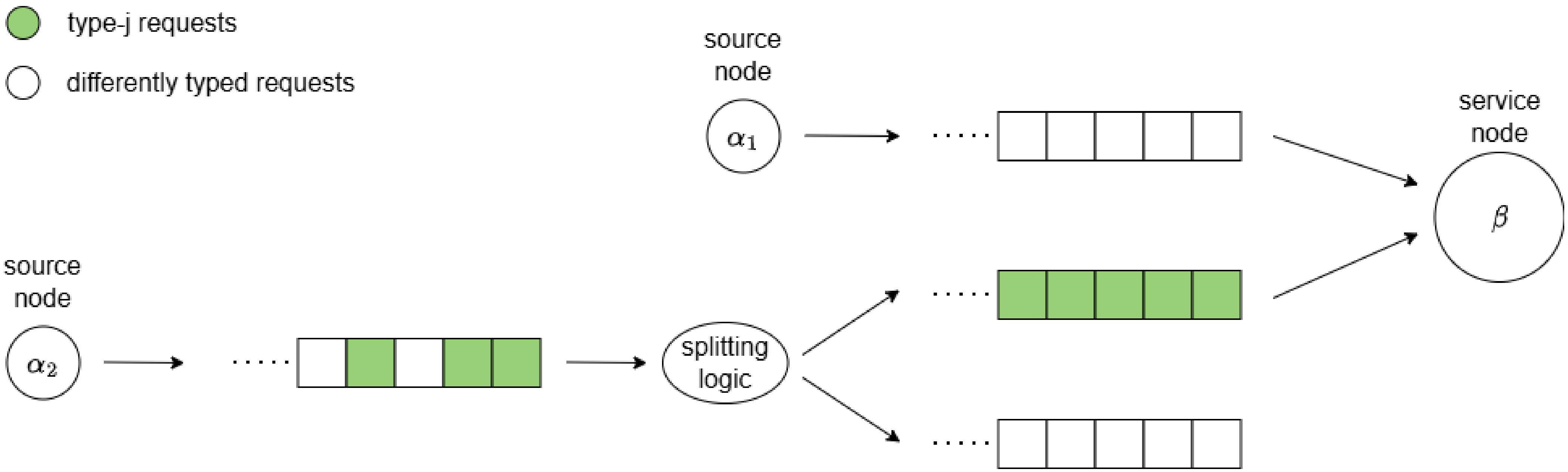
Scenario 1 – splitting on subflow of interest – WCD analysis



- WCD becomes finite also with lower service rates
- Lower WCD bounds using refined arrival curves for subflow j
- Offsets and subsets are helpful in reducing the initial burst

Results – 4/5

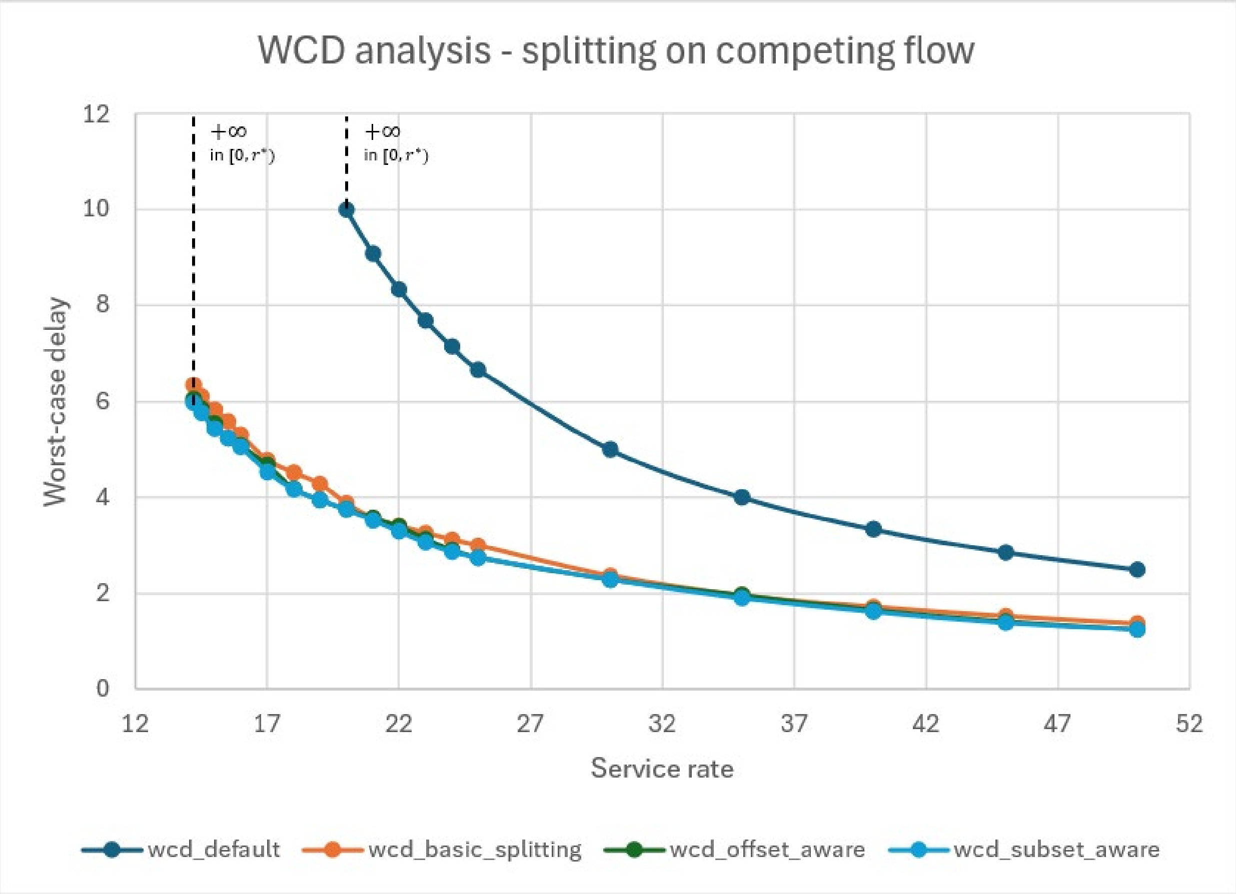
Scenario 2 – splitting on a competing subflow



- We are interested in computing WCD for α_1 packets given that subflow j of α_2 is competing for the same service

Results – 5/5

Scenario 2 – splitting on a competing subflow – WCD analysis



- Noticeable WCD reduction
- Obtaining lower arrival curves has a positive impact on the residual service curve for α_1
- Slight improvements exploiting offsets and subsets

Final remarks

Key contribution

- Formal method to derive subflow-specific arrival curves
- Enables tighter worst-case bounds by exploiting spatial behaviour of packets
- Generalizable to complex request patterns

Future work

- Wide range of applications to be investigated

i.e. cache-aware splitting, compiler/MMU support for optimized splitting

Thank you

Edoardo Loni

PhD student - Department of Information Engineering

University of Pisa

edoardo.loni@phd.unipi.it

References

- Mao, Shiwen and Panwar, Shivendra S and Hou, Y Thomas, “*On minimizing end-to-end delay with optimal traffic partitioning*”- IEEE Transactions on Vehicular Technology - 2006