Using Segmentation to Improve Schedulability of RRA-based NoCs with Mixed Traffic

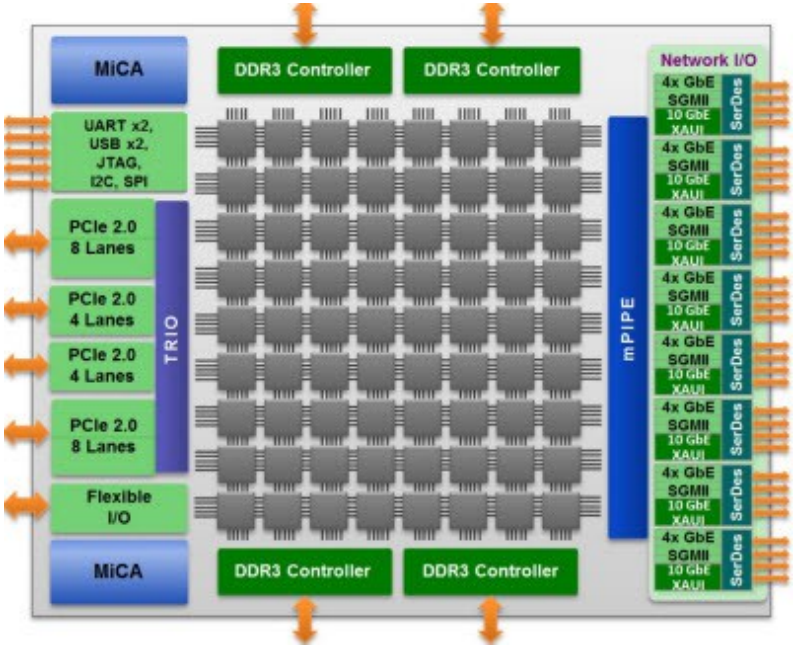
Meng Liu, Matthias Becker, Moris Behnam, Thomas Nolte

ASP-DAC, Chiba, Japan
19. January 2017

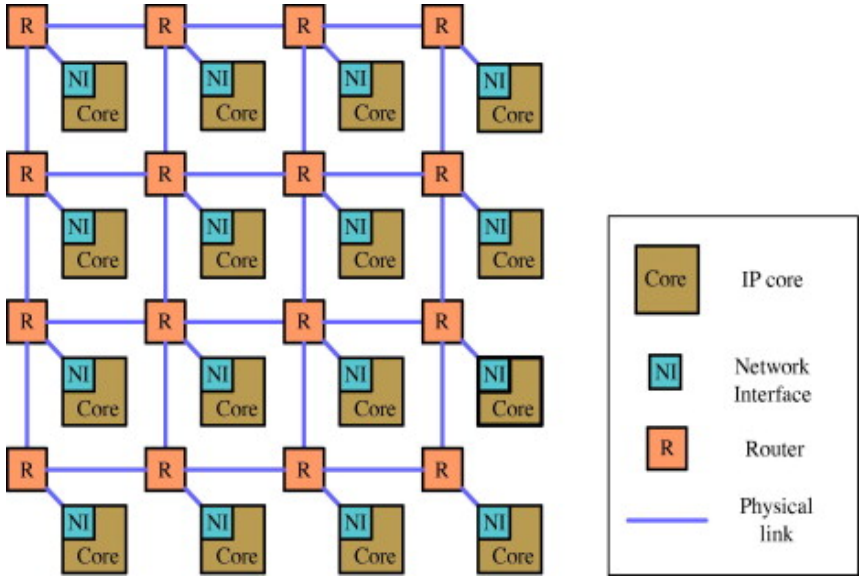
Outline

- Introduction
- Motivation
- Segmentation Algorithm
- Evaluation
- Conclusion

Many-core Platforms and NoCs



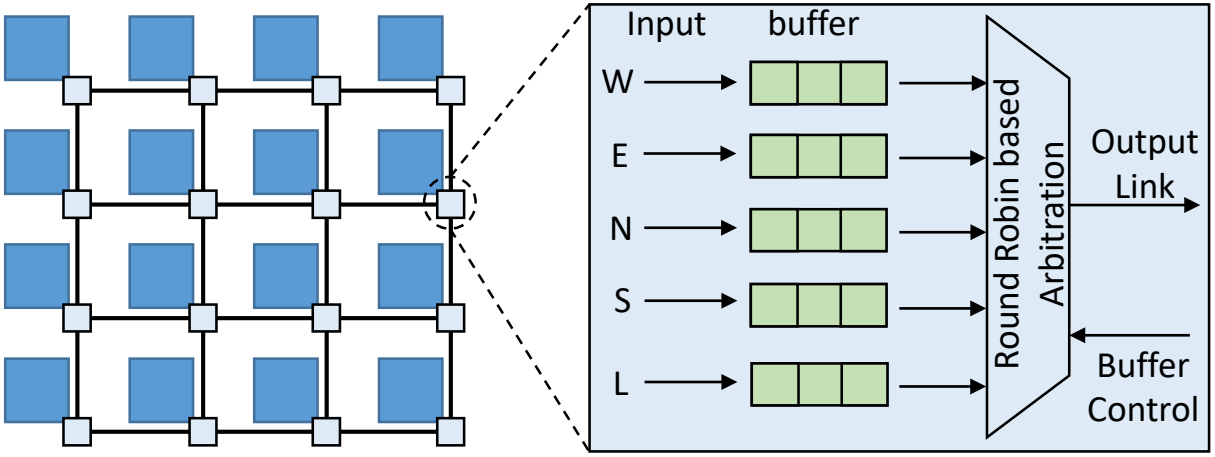
Pics from Tilera



Pic by Nuno Roma

System Model - NoC

- 2D-mesh based NoC
  - Wormhole-switching
  - Round-robin based

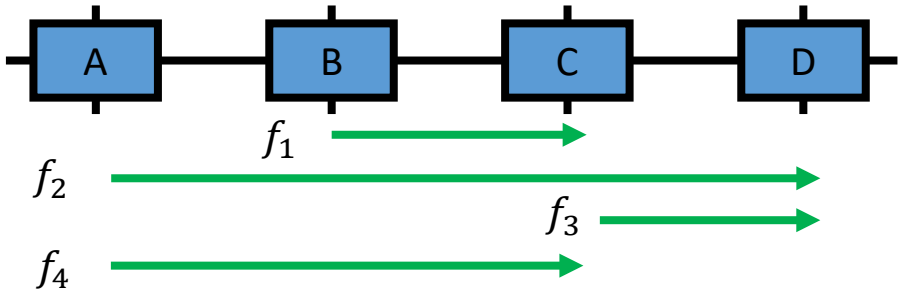


System Model - Flow

- Real-time Flows
  - Periodic or sporadic
  - L_i - Packet size
  - T_i - Minimum Inter-arrival Time (MIT)
  - D_i - relative deadline ($D_i \leq T_i$)
  - R_i - fixed route/path

- Best-effort Flows
  - R_i - fixed route/path

- W_i - Worst-Case Traversal Time of f_i



Motivation (1/2)

- Timeliness is important for real-time applications
  - Each packet should be delivered within its deadline (i.e. $W_i \leq D_i$)

Motivation (1/2)

- Timeliness is important for real-time applications
  - Each packet should be delivered within its deadline (i.e. $W_i \leq D_i$)

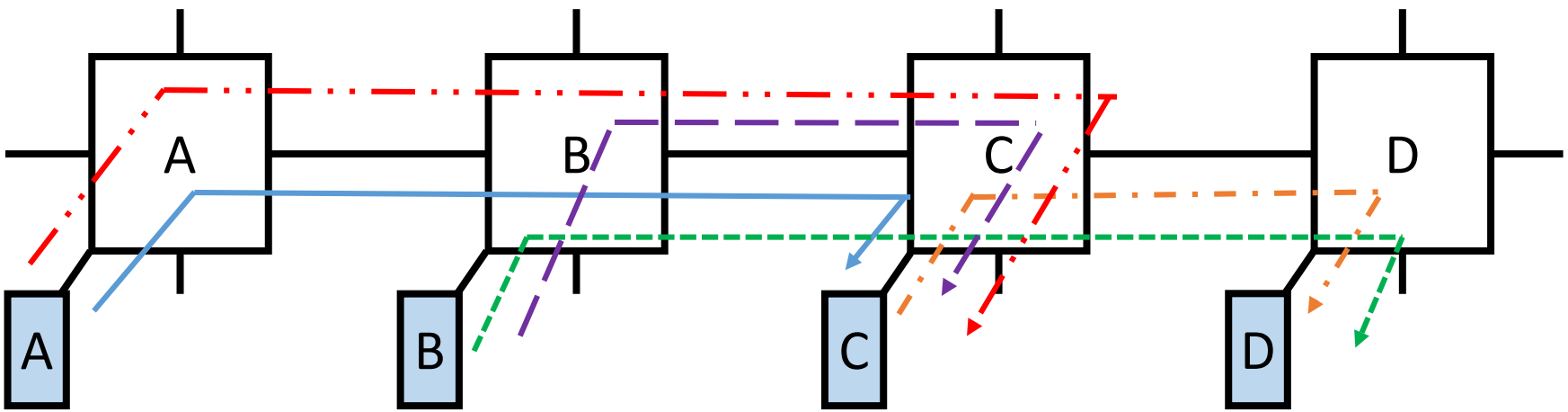
Motivation (1/2)

- Timeliness is important for real-time applications
  - Each packet should be delivered within its deadline (i.e. $W_i \leq D_i$)

If the timing analysis shows a result of *Unschedulable*, what shall we do to remove deadline misses?

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?



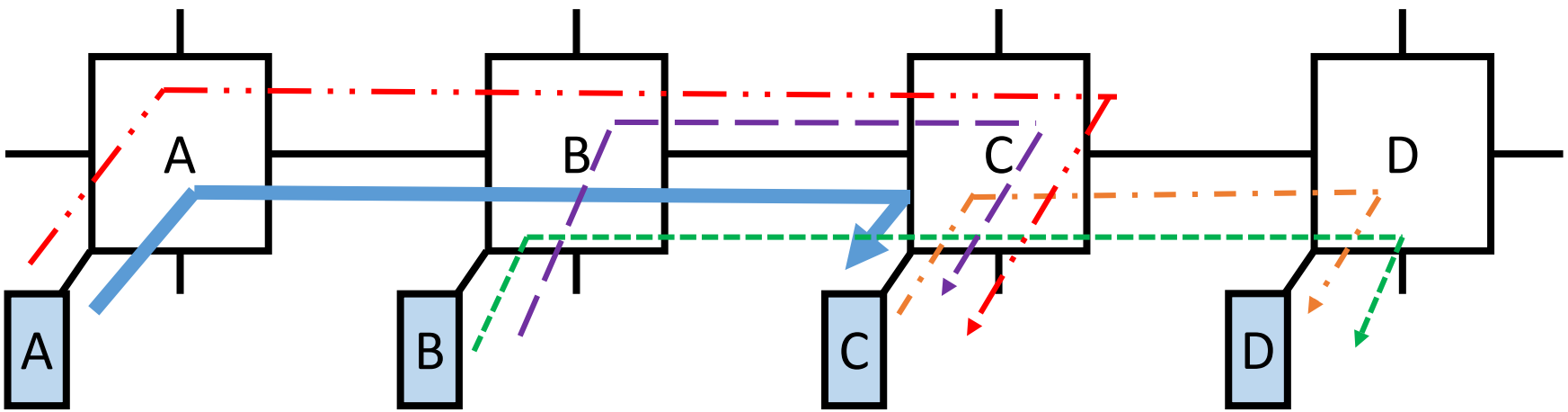
f_i ($Sr_i \rightarrow Ds_i$)

- f_1 (A → C)
- f_2 (B → D)
- f_3 (B → C)
- f_4 (C → D)
- f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?

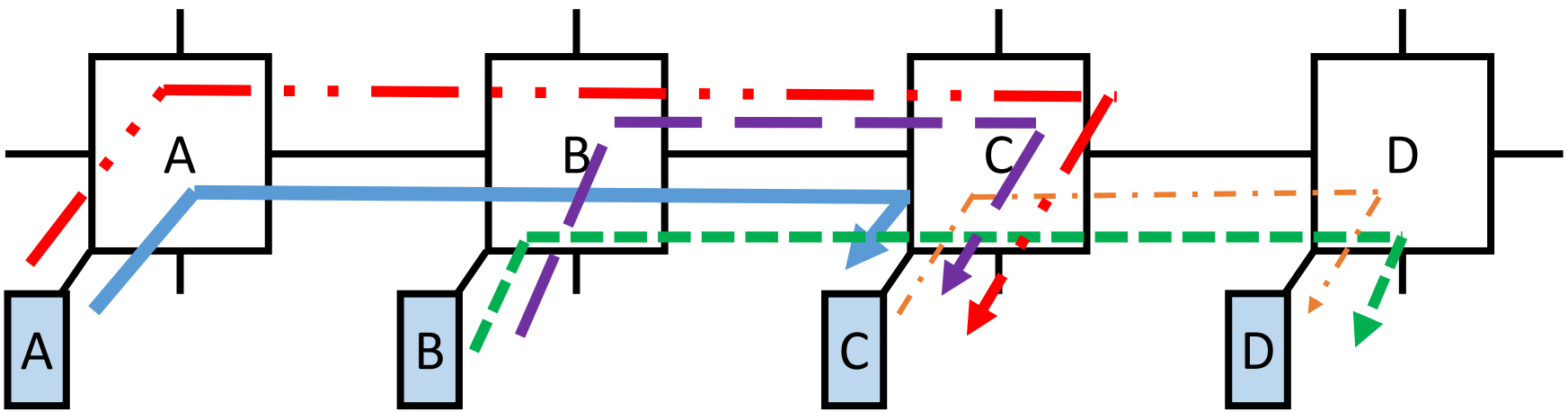


f_i ($Sr_i \rightarrow Ds_i$)
f_1 (A → C)
f_2 (B → D)
f_3 (B → C)
f_4 (C → D)
f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?

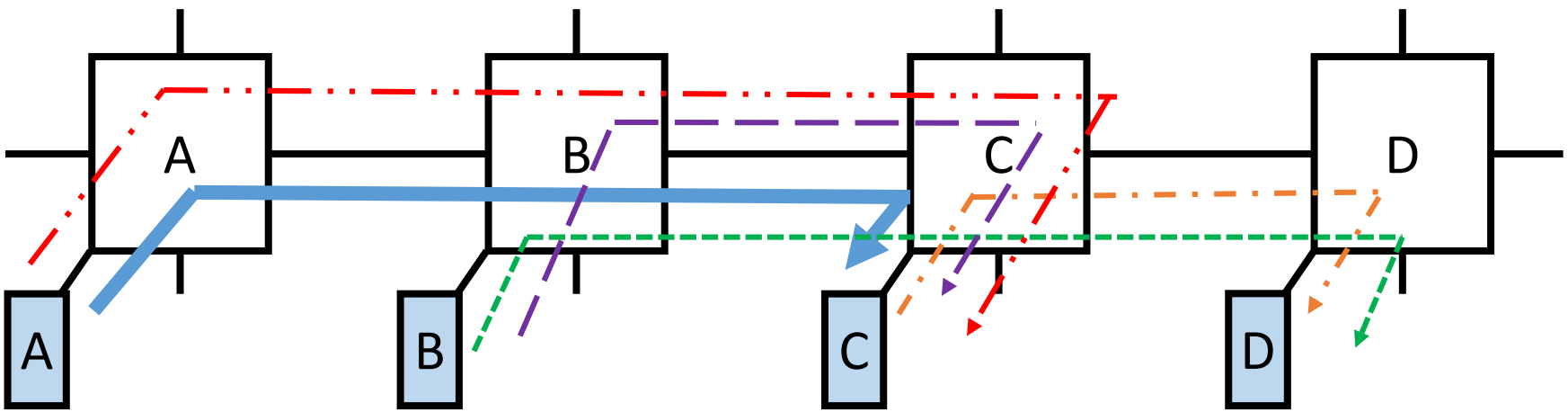


f_i ($Sr_i \rightarrow Ds_i$)
f_1 (A → C)
f_2 (B → D)
f_3 (B → C)
f_4 (C → D)
f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?

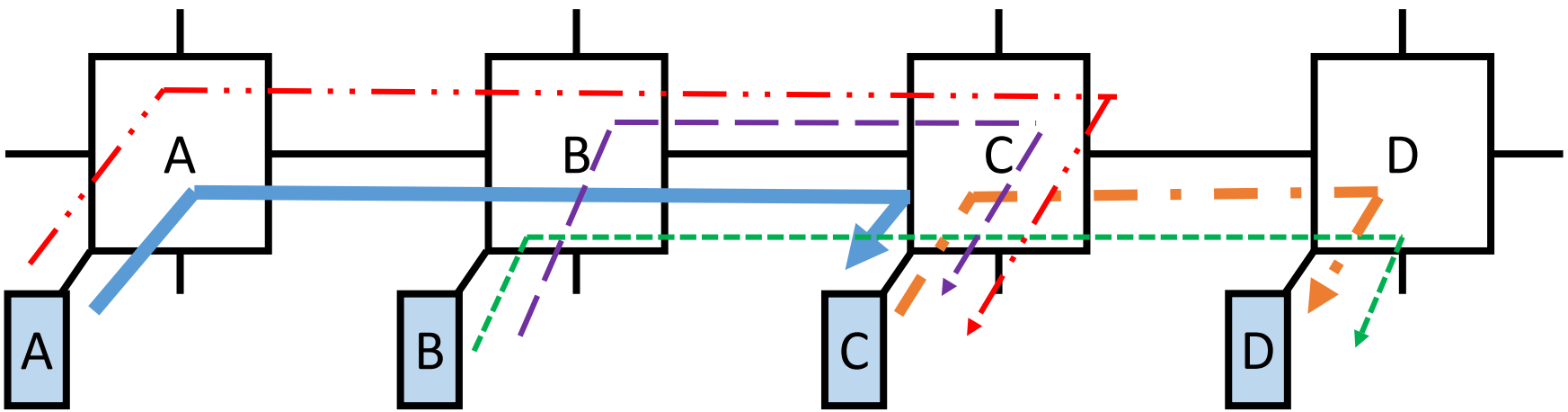


f_i ($Sr_i \rightarrow Ds_i$)
- f_1 (A → C)
- f_2 (B → D)
- f_3 (B → C)
- f_4 (C → D)
- f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?

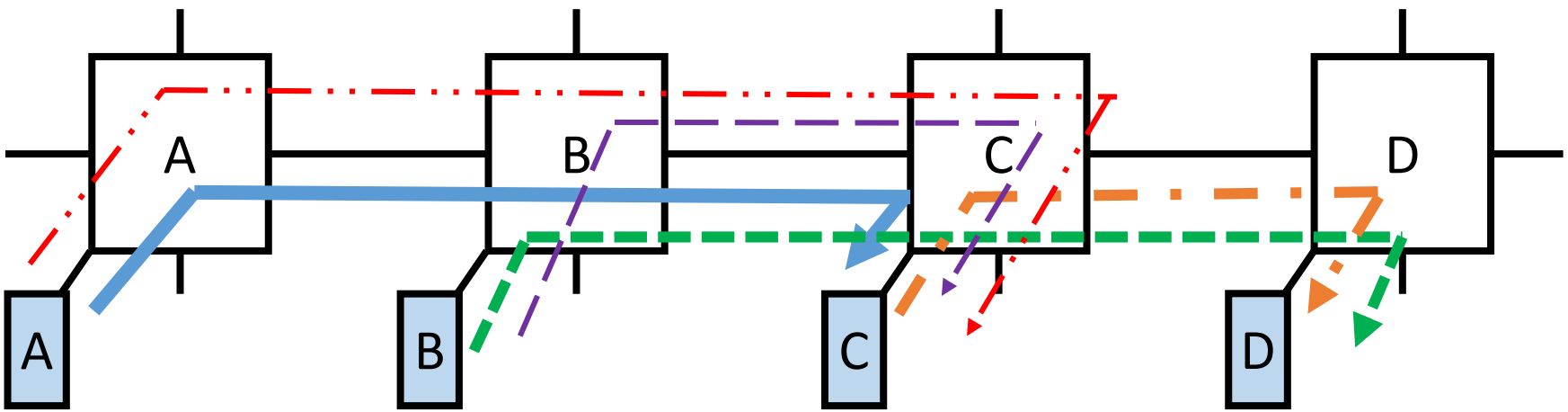


f_i ($Sr_i \rightarrow Ds_i$)
- f_1 (A → C)
- f_2 (B → D)
- f_3 (B → C)
- f_4 (C → D)
- f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?

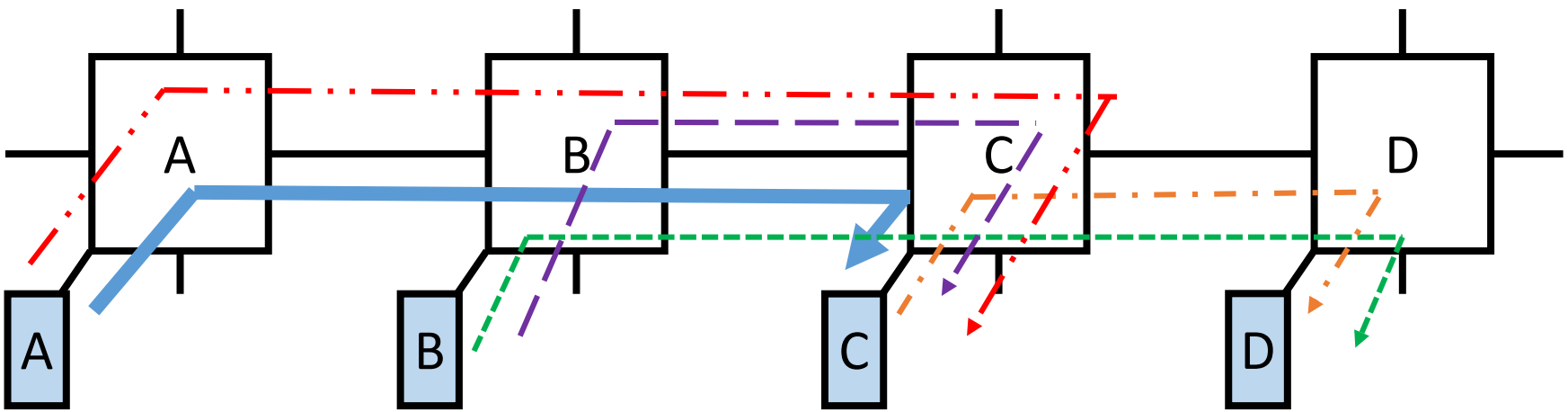


f_i ($Sr_i \rightarrow Ds_i$)
f_1 (A → C)
f_2 (B → D)
f_3 (B → C)
f_4 (C → D)
f_5 (A → C)

- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Motivation (2/2)

- How can we prevent a real-time flow from missing its deadline?



f_i ($Sr_i \rightarrow Ds_i$)
f_1 (A → C)
f_2 (B → D)
f_3 (B → C)
f_4 (C → D)
f_5 (A → C)

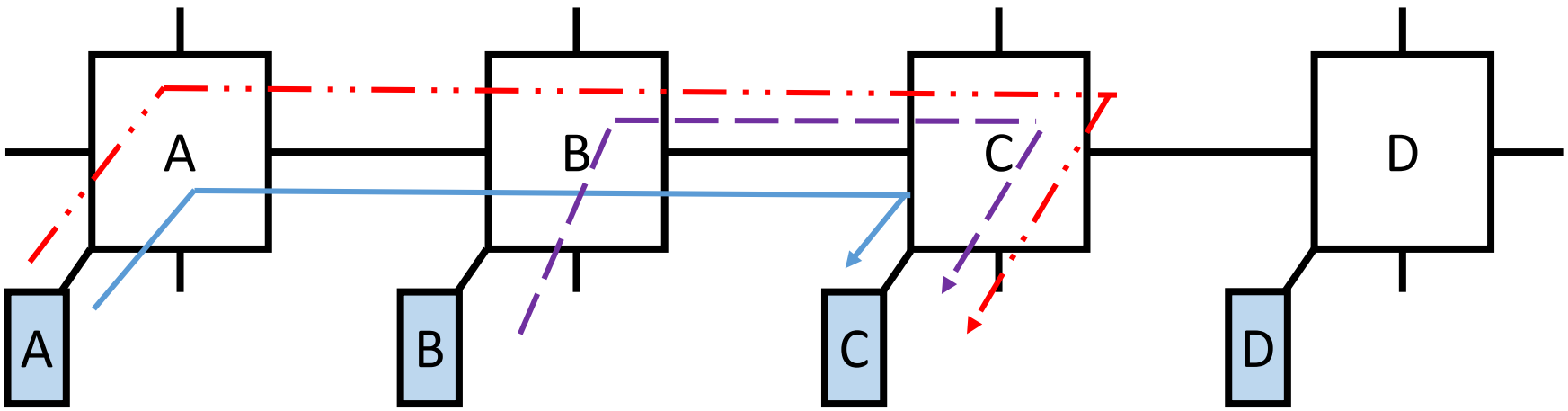
- f_1 may get direct blocking from f_2, f_3 and f_5
- f_1 may get indirect blocking from f_4 through f_2
- The WCTT of f_1 consists of its **basic transmission time** and the **blocking delay**

Contributions

- A segmentation approach to improve schedulability of real-time traffic
  - Segmentation – dividing a NoC packet into a number of smaller packets
- An algorithm for selecting proper segment sizes of best-effort traffic such that timeliness of real-time traffic is still guaranteed

Segmentation Approach (1/2)

- Why segmentation?



f_i ($Sr_i \rightarrow Ds_i$)

f_1 (A → C)

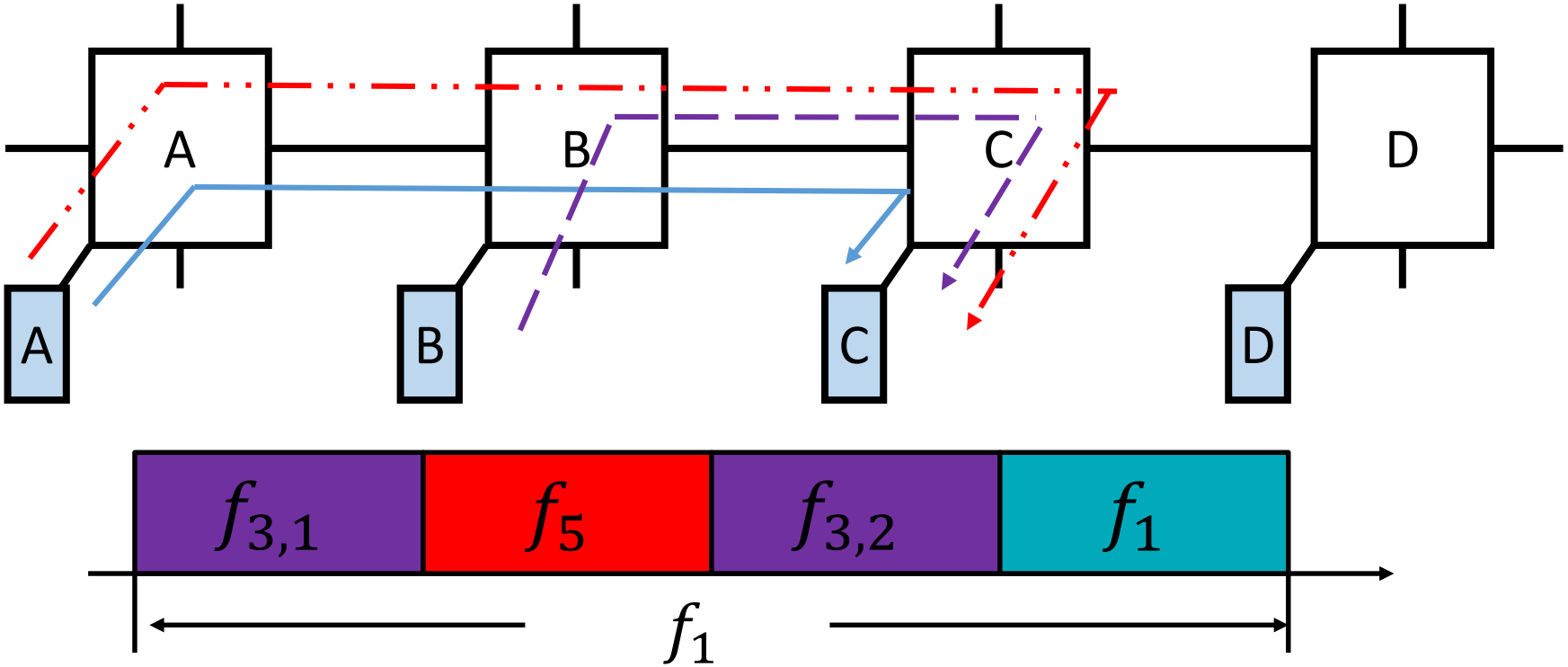
f_3 (B → C)

f_5 (A → C)

- f_1 can get blocking from f_3 for at most **TWO** time
  - f_3 blocks f_5, which further blocks f_1
  - f_3 directly blocks f_1

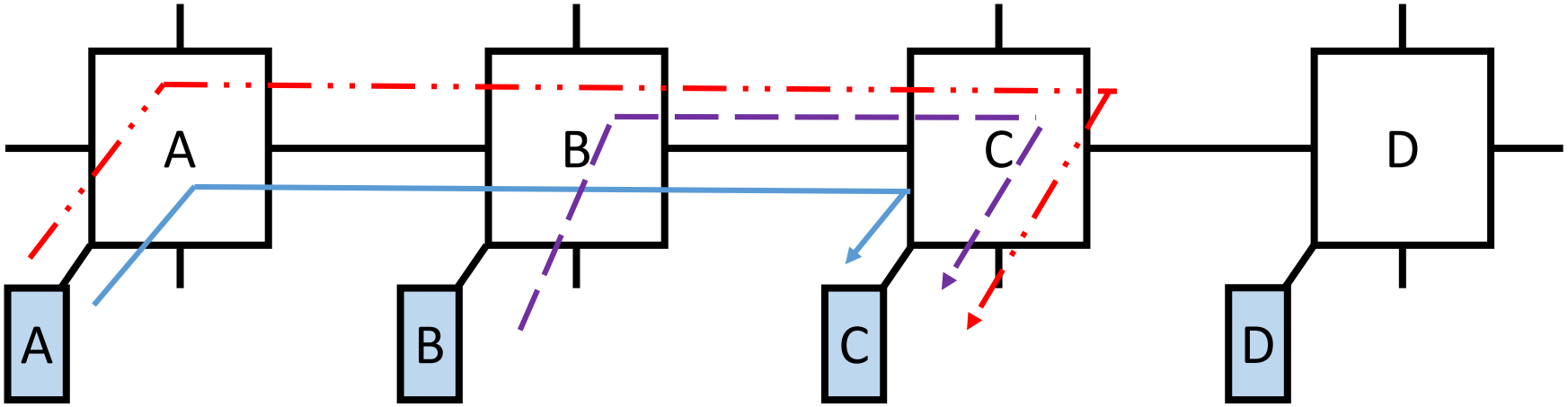
Segmentation Approach (1/2)

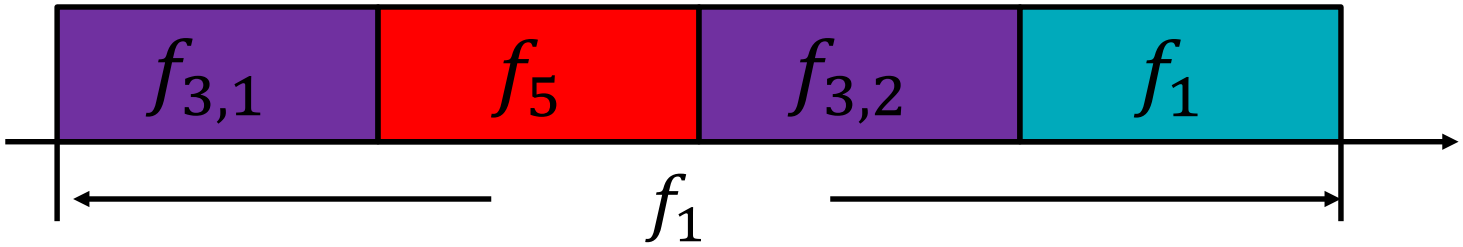
- Why segmentation?

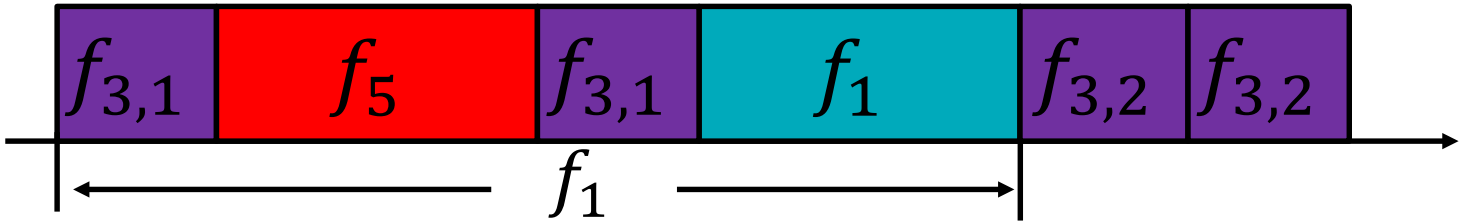


Segmentation Approach (1/2)

- Why segmentation?





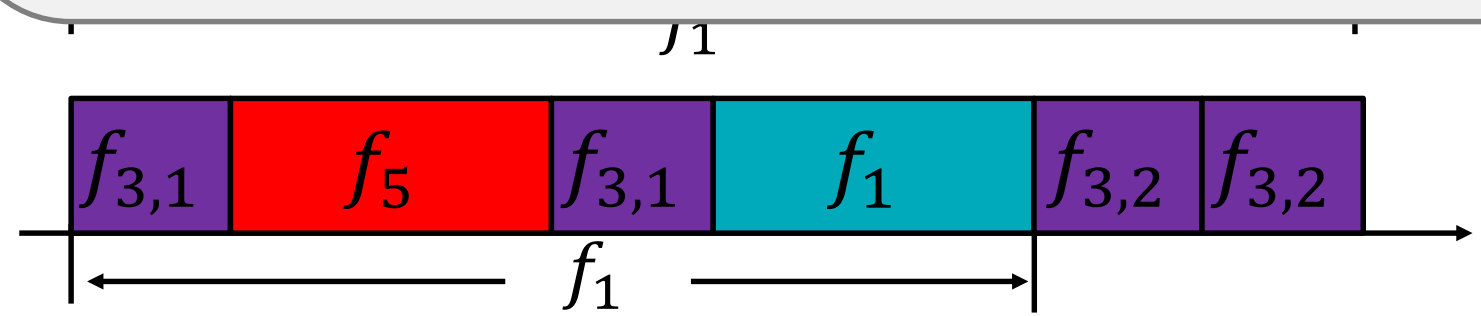


Segmentation Approach (1/2)

- Why segmentation?

Solution Principle:

- Bandwidth is a fixed shared resource
- We cannot decrease latency of all the flows at the same time



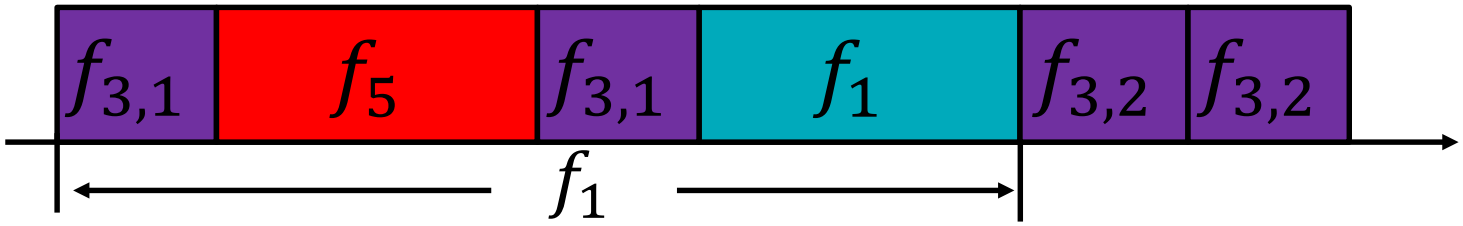
Segmentation Approach (1/2)

- Why segmentation?

Solution Principle:

- Bandwidth is a fixed shared resource
- We cannot decrease latency of all the flows at the same time

- Decrease latency of flows which may miss their deadlines, and increase latency of some flows while keeping them schedulable



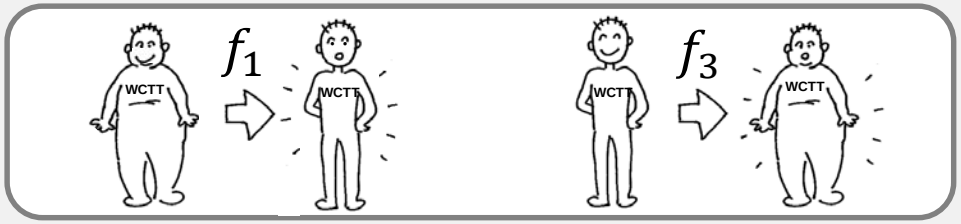
Segmentation Approach (1/2)

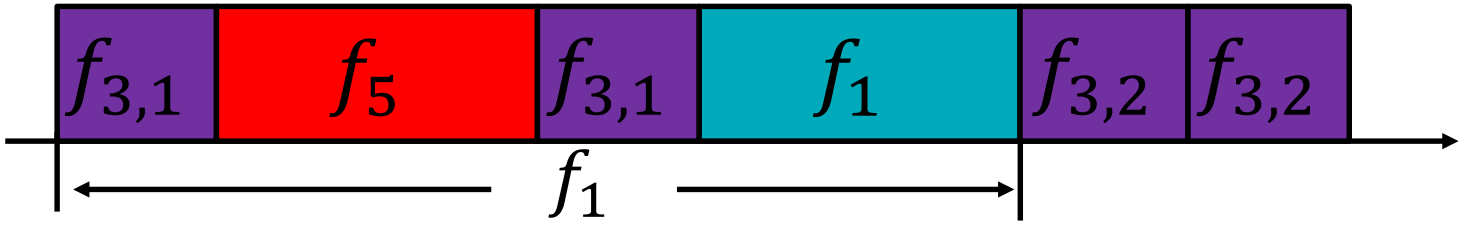
- Why segmentation?

Solution Principle:

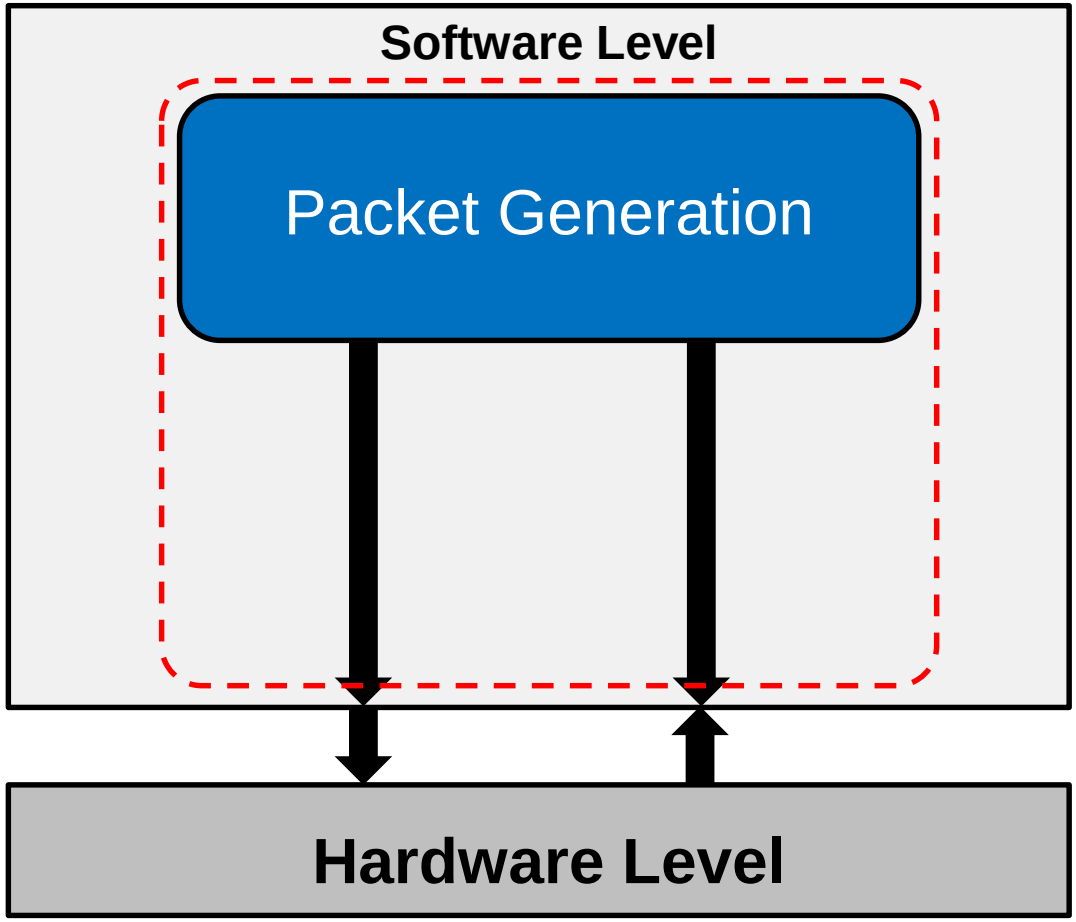
- Bandwidth is a fixed shared resource
- We cannot decrease latency of all the flows at the same time

- Decrease latency of flows which may miss their deadlines, and increase latency of some flows while keeping them schedulable

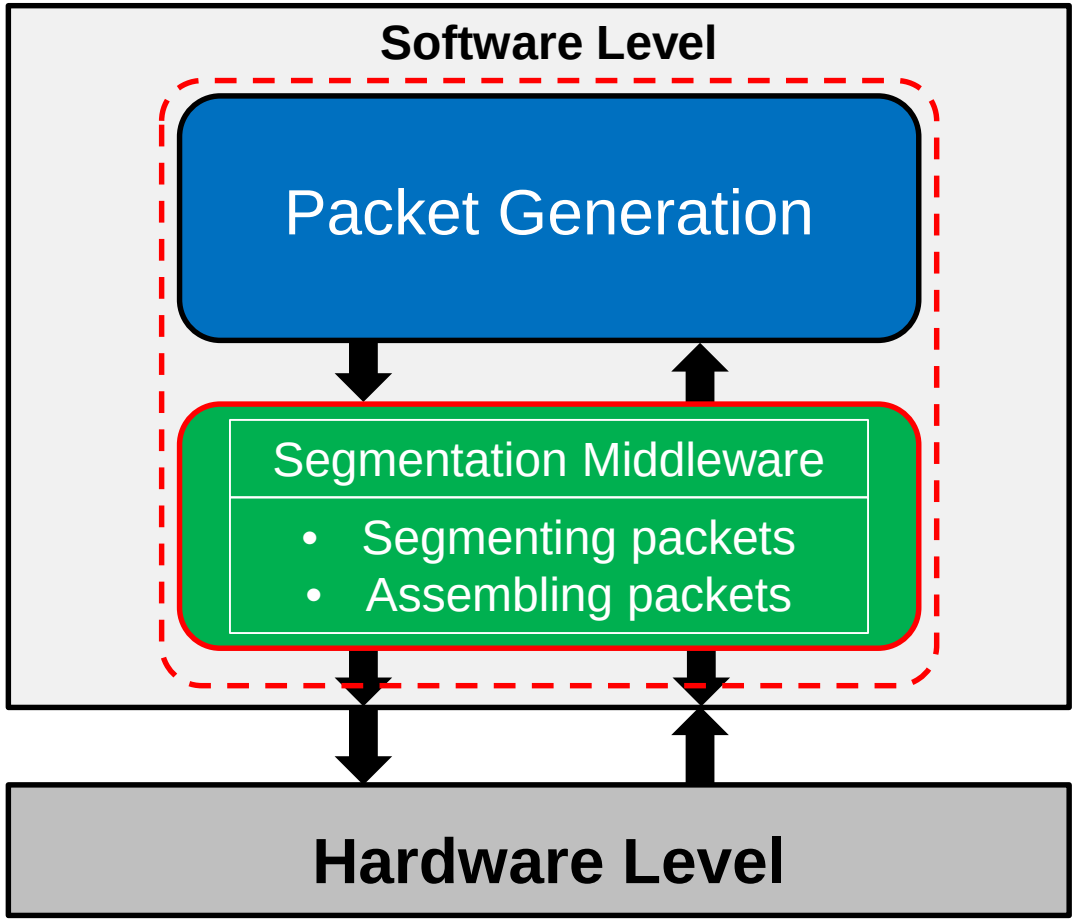




Segmentation Approach (2/2)



Segmentation Approach (2/2)



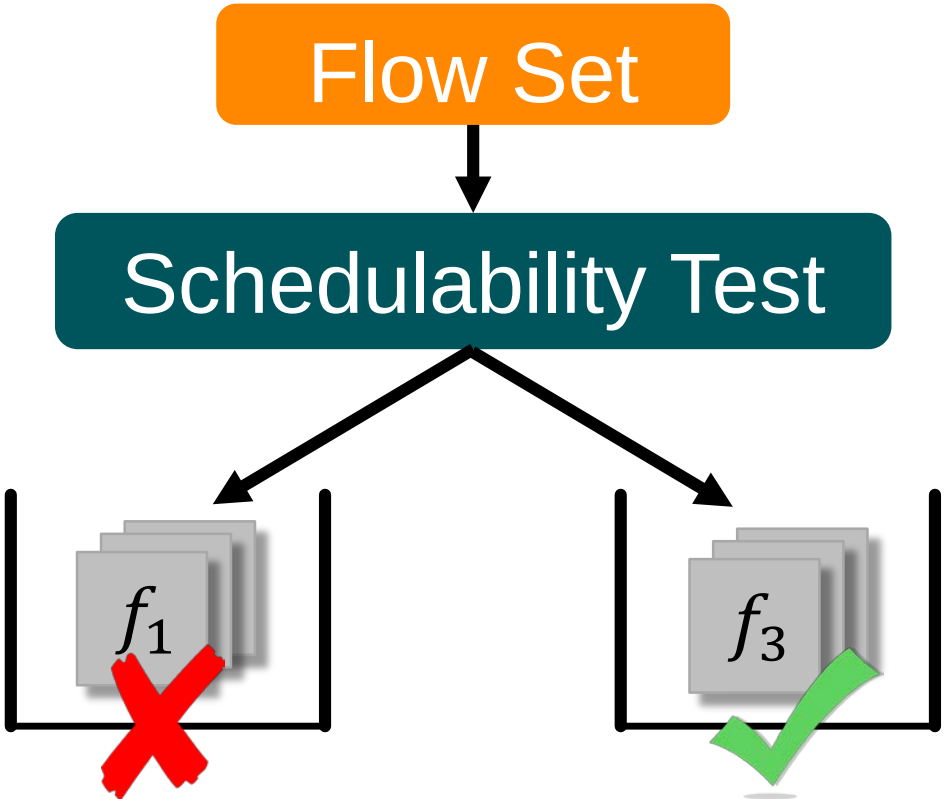
Segmentation Alg. Outline (1/2)

Flow Set

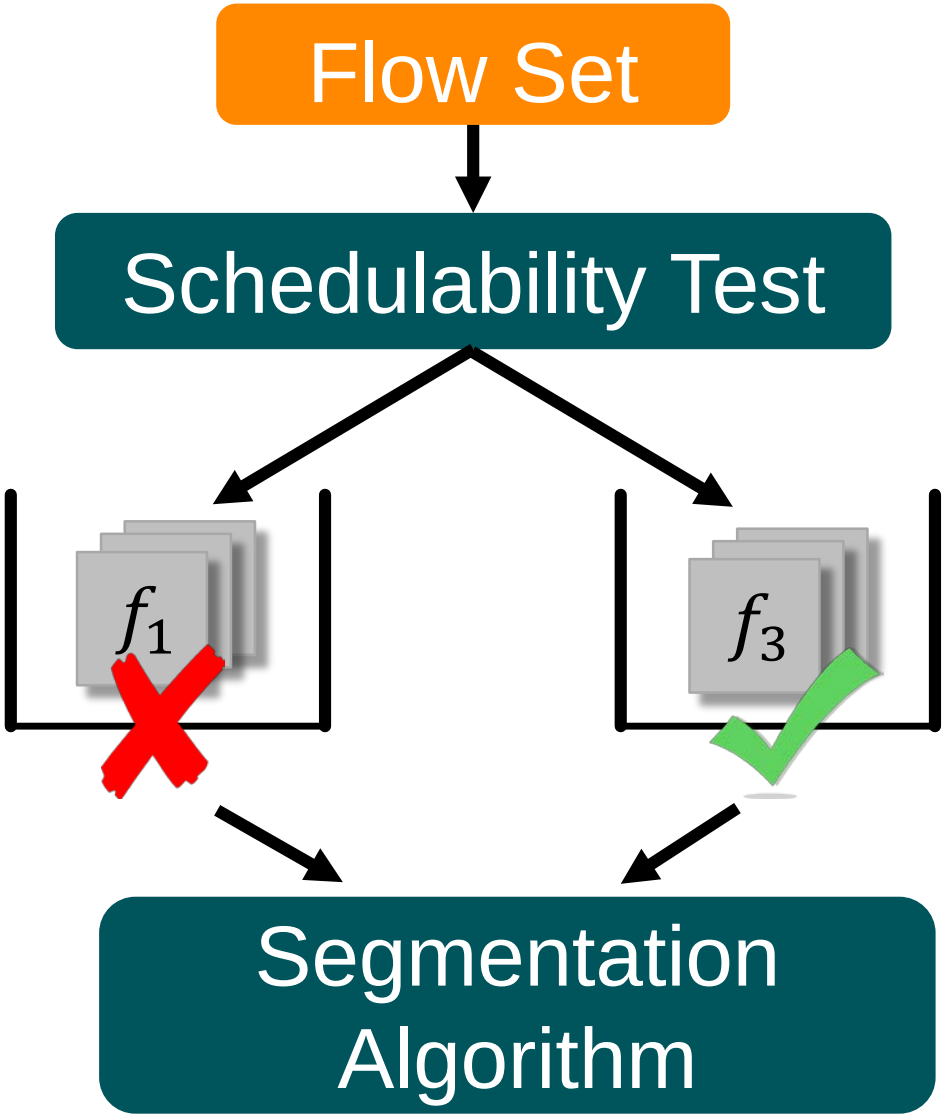
↓

Schedulability Test

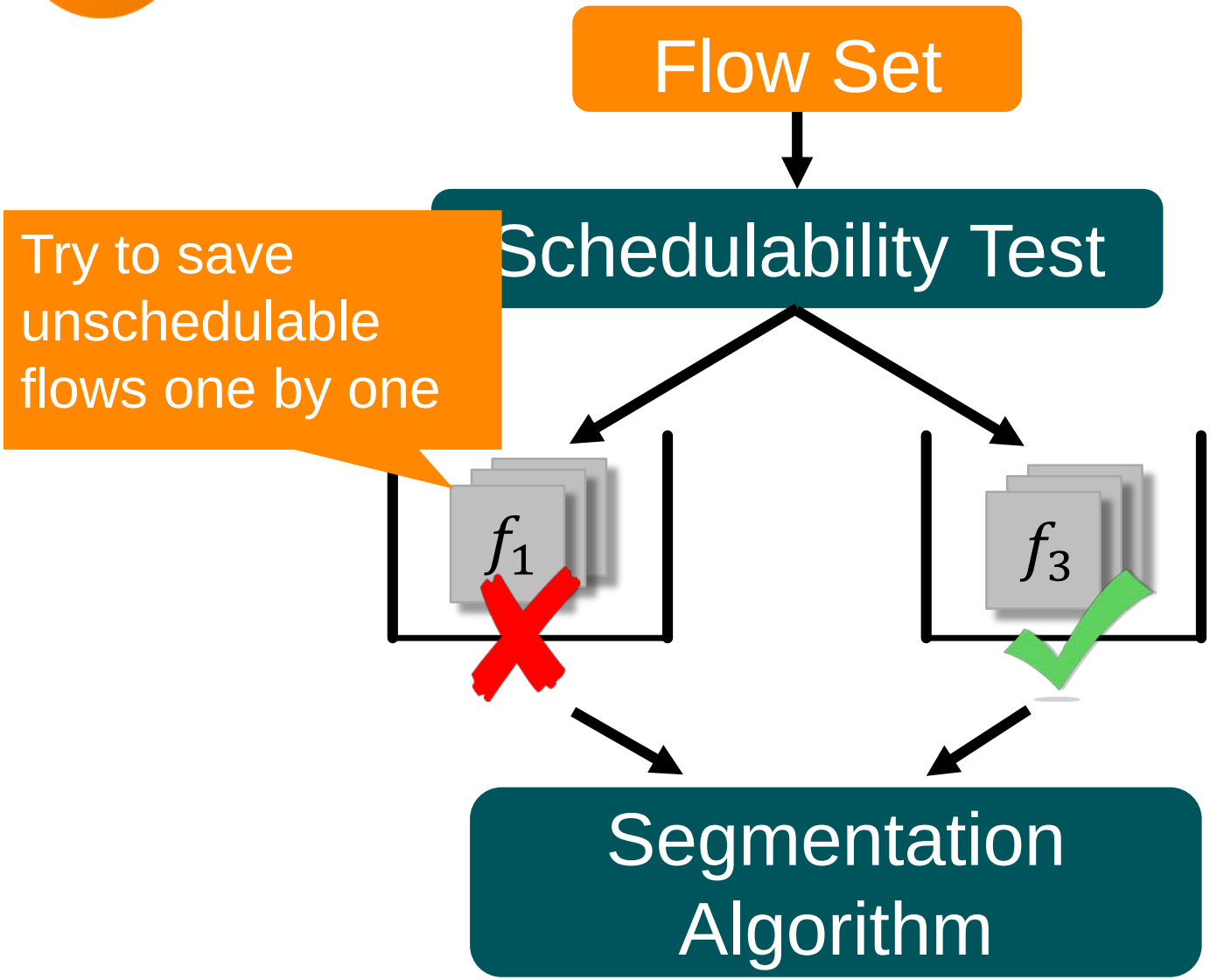
Segmentation Alg. Outline (1/2)



Segmentation Alg. Outline (1/2)



Segmentation Alg. Outline (1/2)



Segmentation Alg. Outline (2/2)

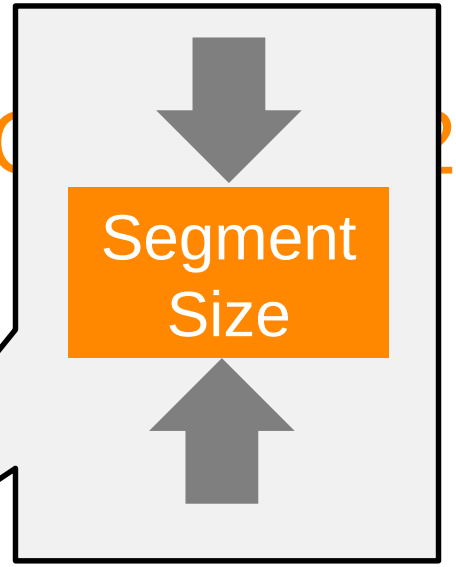
- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actually affect current W_i

Segmentation Alg. Outline (2/2)

- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actually affect current W_i

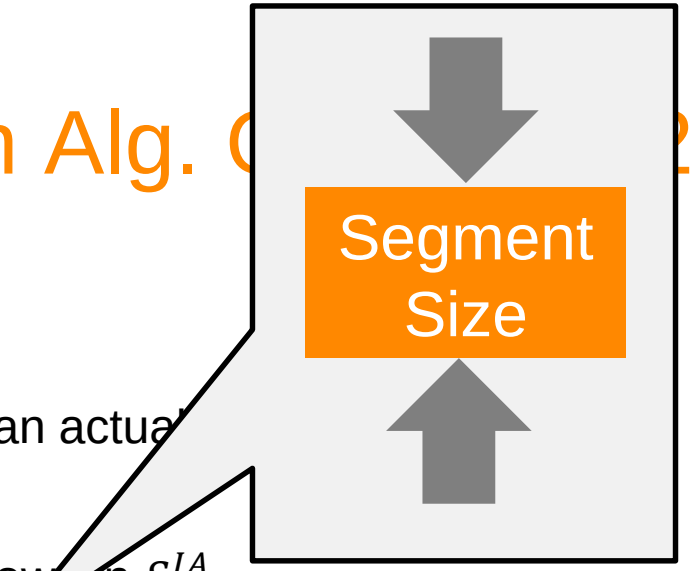
- Apply segmentation on the flows in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced

Segmentation Alg. (



- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actua
- Apply segmentation on the flow in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced

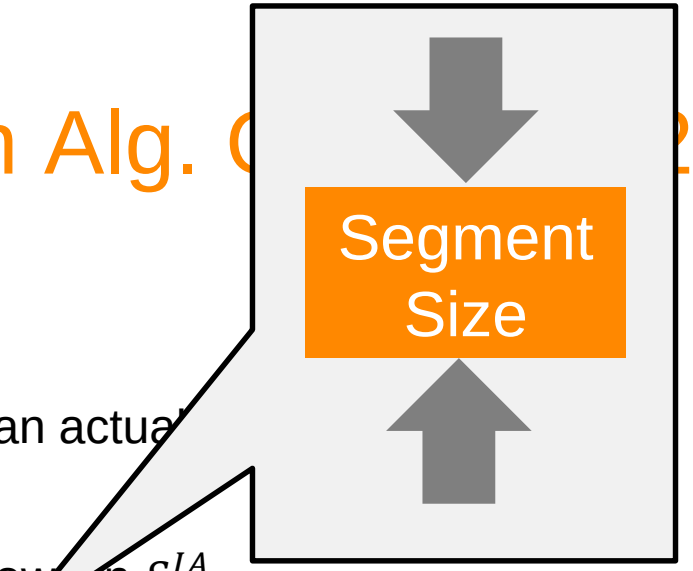
Segmentation Alg. (2)



- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actual

- Apply segmentation on the flows in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced
  - On the other hand, ***f_j may suffer from more blocking***
  - Thus, the schedulability of f_j has to be rechecked after each segmentation

Segmentation Alg. (

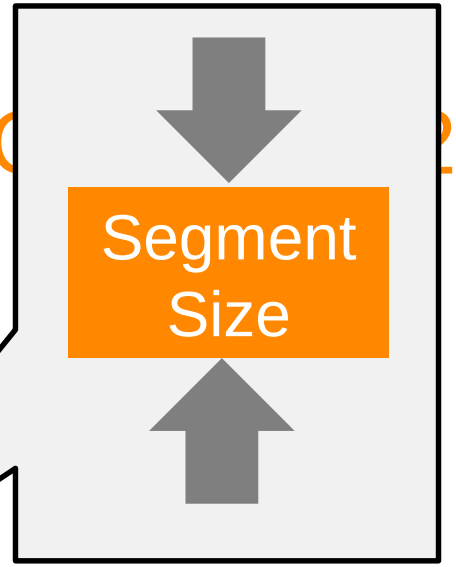


- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actua

- Apply segmentation on the flow in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced
  - On the other hand, ***f_j may suffer from more blocking***
  - Thus, the schedulability of f_j has to be rechecked after each segmentation

- The segmentation process is repeated until
  - f_i becomes schedulable → Success
  - No flows in S_i^{IA} can be segmented any more → Failure

Segmentation Alg. (2)

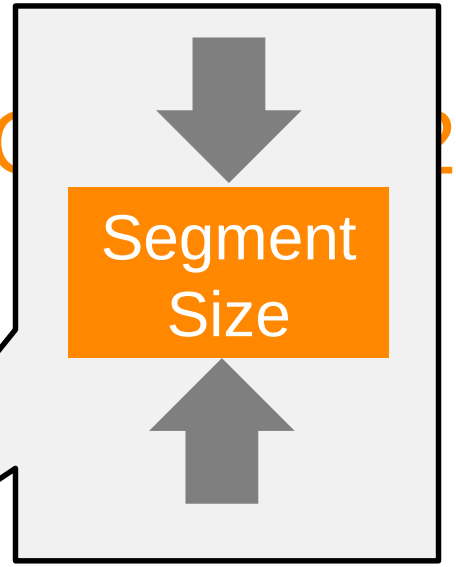


- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actual

- Apply segmentation on the flows in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced
  - On the other hand, ***f_j may suffer from more blocking***
  - Thus, the schedulability of f_j has to be rechecked after each segmentation

- The segmentation process is repeated until
  - f_i becomes schedulable → Success
  - No flows in S_i^{IA} can be segmented any more → Failure

Segmentation Alg. (



- f_i is the flow to be saved
- S_i^{IA} consists of flows which can actua

- Apply segmentation on the flow in S_i^{IA}
  - Gradually decrease segment size of f_j ($f_j \in S_i^{IA}$), so that blocking on f_i can be reduced
  - On the other hand, ***f_j may suffer from more blocking***
  - Thus, the schedulability of f_j has to be rechecked after each segmentation

- The segmentation process is repeated until
  - f_i becomes schedulable → Success
  - No flows in S_i^{IA} can be segmented any more → Failure

Adding best-effort traffic

- A packet with larger segment size (i.e. fewer number of segments) can have shorter WCTT
  - ***Increase segment size of each BE packet as much as possible while still guaranteeing the timeliness of real-time traffic***

Adding best-effort traffic

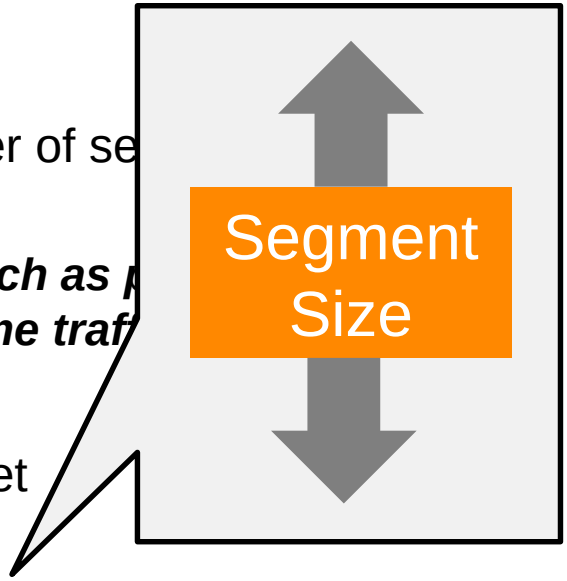
- A packet with larger segment size (i.e. fewer number of segments) can have shorter WCTT
  - ***Increase segment size of each BE packet as much as possible while still guaranteeing the timeliness of real-time traffic***

- Start with minimum segment size for each BE packet

- Increase segment size of all the BE flows at the same time
  - If the increased segment size of a BE flow can make any real-time flow unschedulable, stop increasing this BE flow any more

Adding best-effort traffic

- A packet with larger segment size (i.e. fewer number of se
  have shorter WCTT
  - ***Increase segment size of each BE packet as much as p***
    while still guaranteeing the timeliness of real-time traff

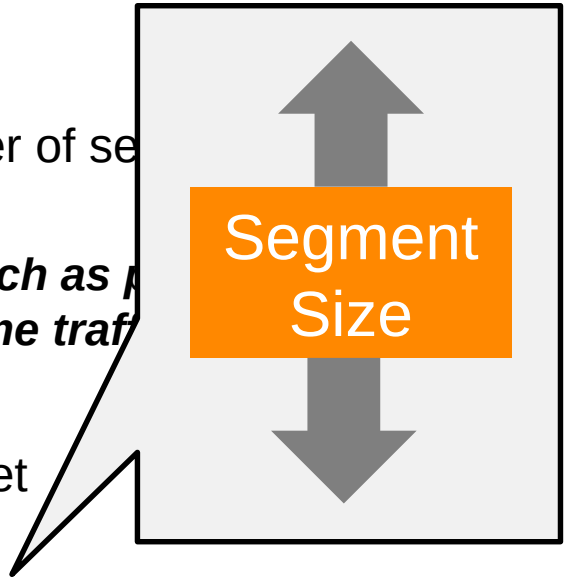


- Start with minimum segment size for each BE packet

- Increase segment size of all the BE flows at the same time
  - If the increased segment size of a BE flow can make any real-time flow unschedulable, stop increasing this BE flow any more

Adding best-effort traffic

- A packet with larger segment size (i.e. fewer number of se
  have shorter WCTT
  - ***Increase segment size of each BE packet as much as p***
    while still guaranteeing the timeliness of real-time traf

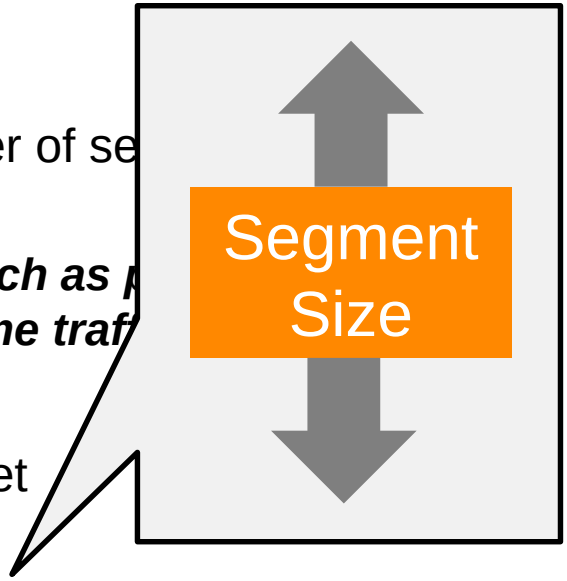


- Start with minimum segment size for each BE packet
- Increase segment size of all the BE flows at the same time
  - If the increased segment size of a BE flow can make any real-time flow unschedulable, stop increasing this BE flow any more

- The algorithm terminates when no BE flows can have further increased segment size

Adding best-effort traffic

- A packet with larger segment size (i.e. fewer number of se
  have shorter WCTT
  - ***Increase segment size of each BE packet as much as p***
    while still guaranteeing the timeliness of real-time traff



- Start with minimum segment size for each BE packet
- Increase segment size of all the BE flows at the same time
  - If the increased segment size of a BE

As long as the actual segment size of a BE flow does not exceed the selected bound, the timeliness of real-time flows is always guaranteed

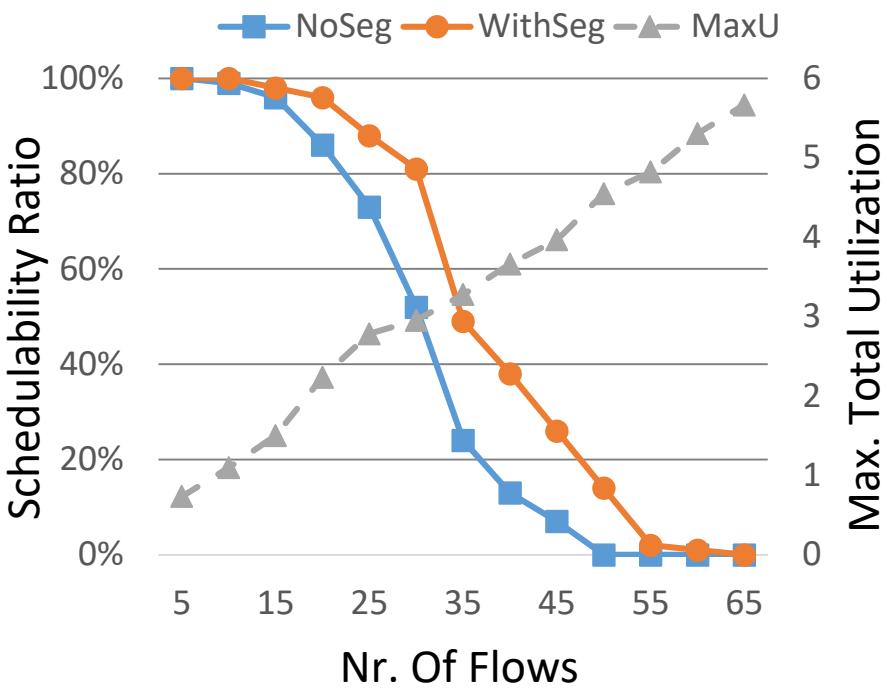
  - ...algorithm terminates when no BE flows can have further increased segment size

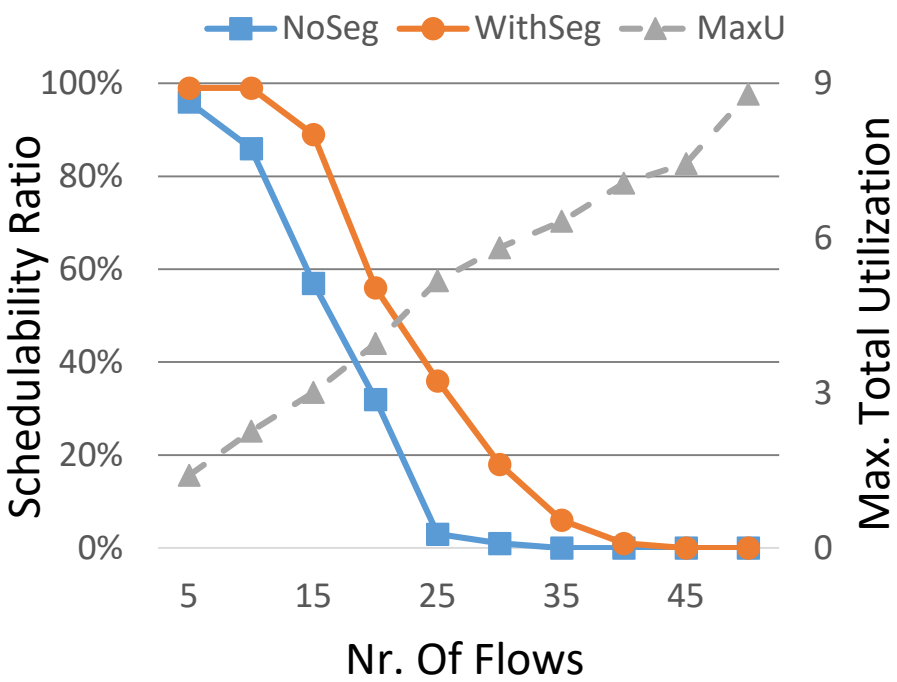
Evaluation (1/4)

- 8 × 8 2D-meshed NoC
- Randomly generated flows
  - Following uniform distribution
  - Packet size – [5, 25], [5, 50], [5, 100] flits
  - Utilization – [0.003, 0.1], [0.01, 0.2]
- Results are presented by schedulability ratio
  - Percentage of schedulable flow sets among all the generated flow sets (100 samples for each data point)

Evaluation (2/4)

Packet size – [5, 25]



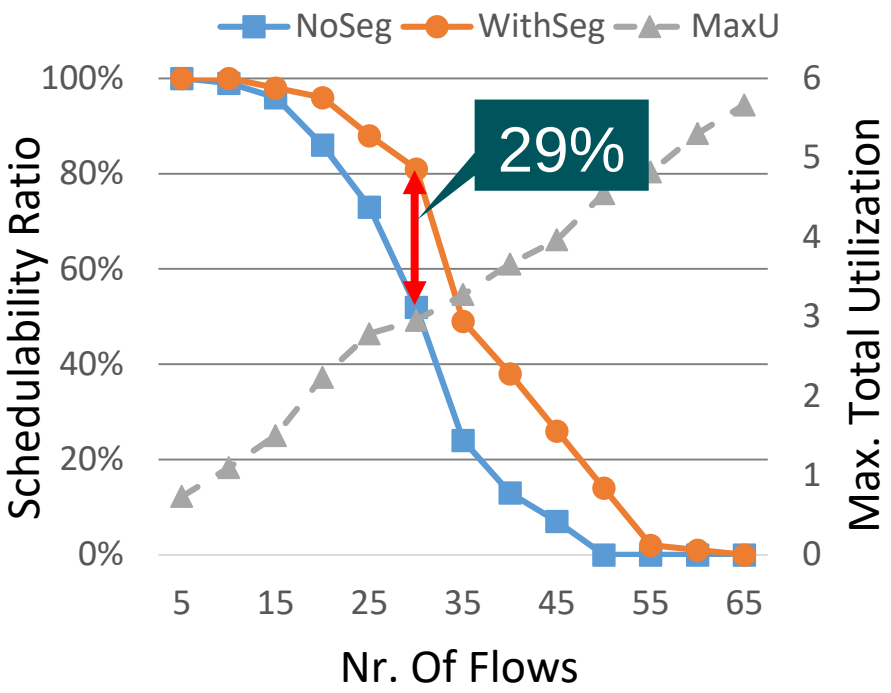


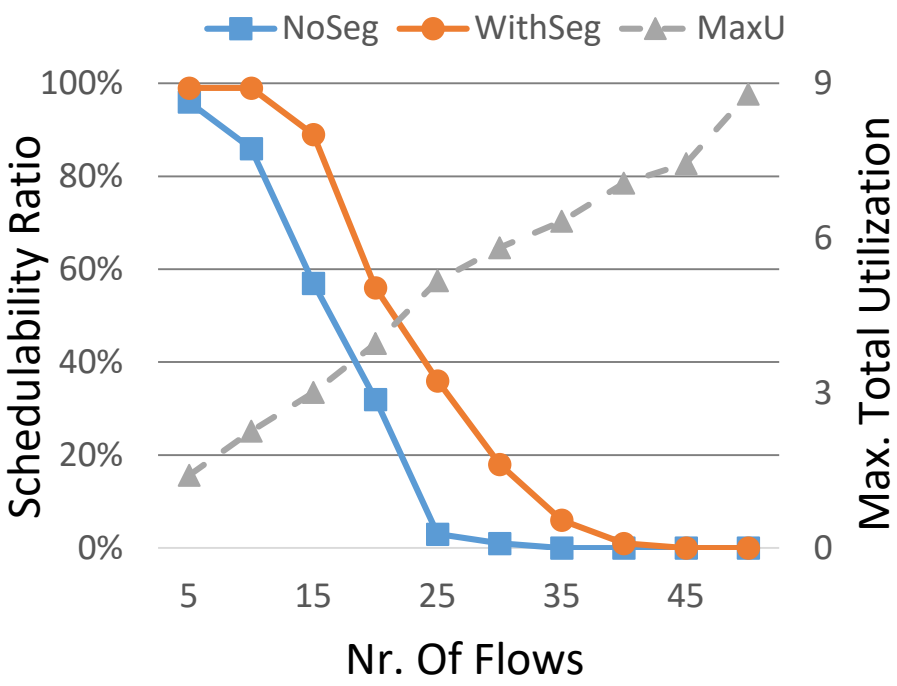
Utilization – [0.003, 0.1]

Utilization – [0.01, 0.2]

Evaluation (2/4)

Packet size – [5, 25]



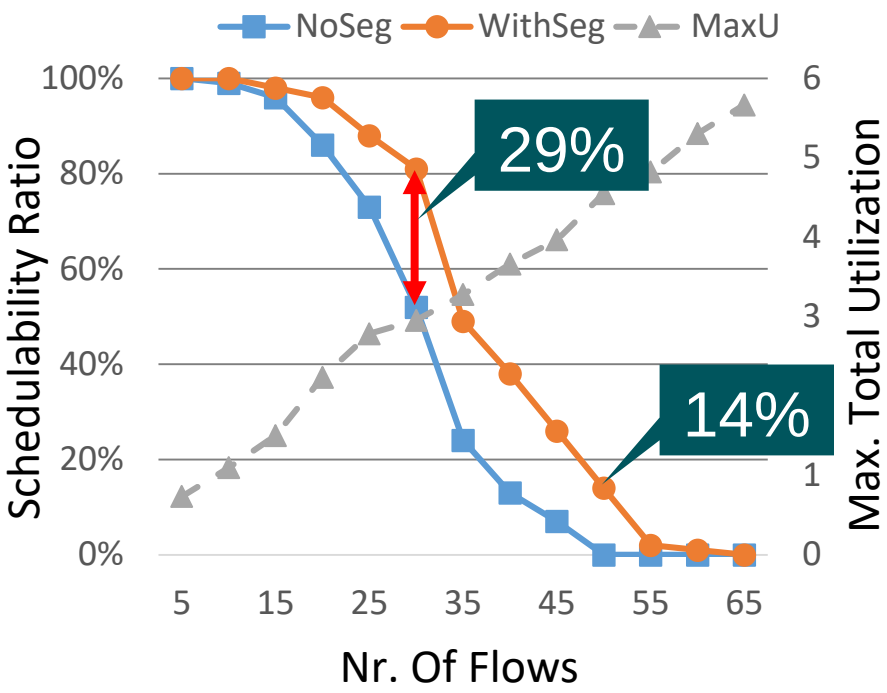


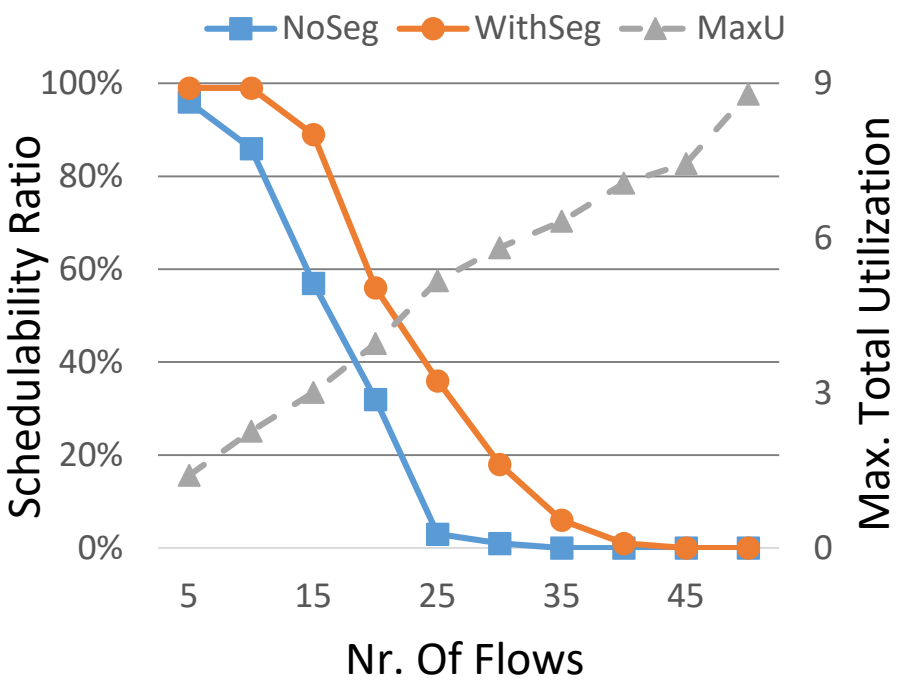
Utilization – [0.003, 0.1]

Utilization – [0.01, 0.2]

Evaluation (2/4)

- Packet size – [5, 25]



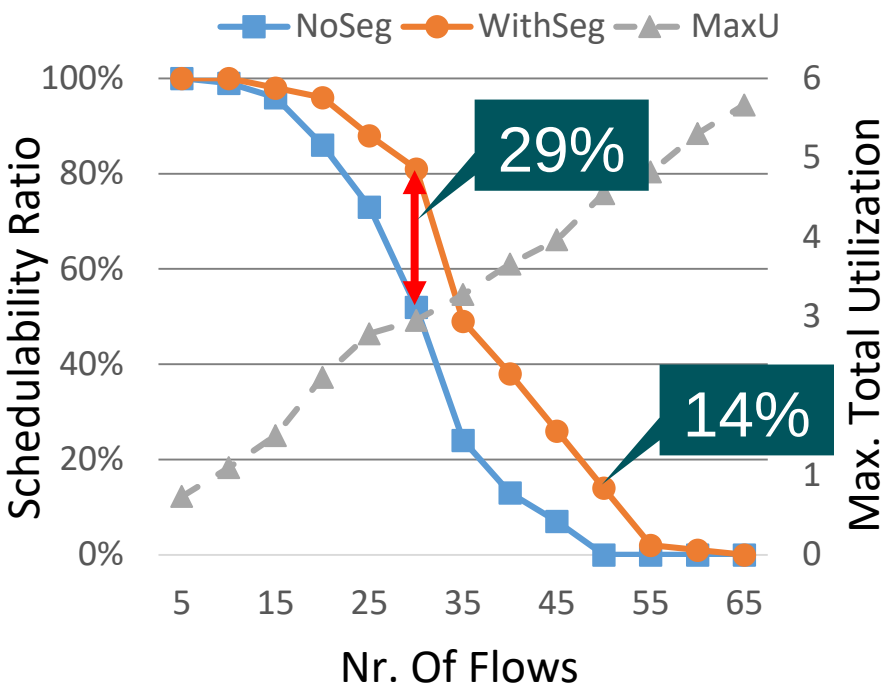


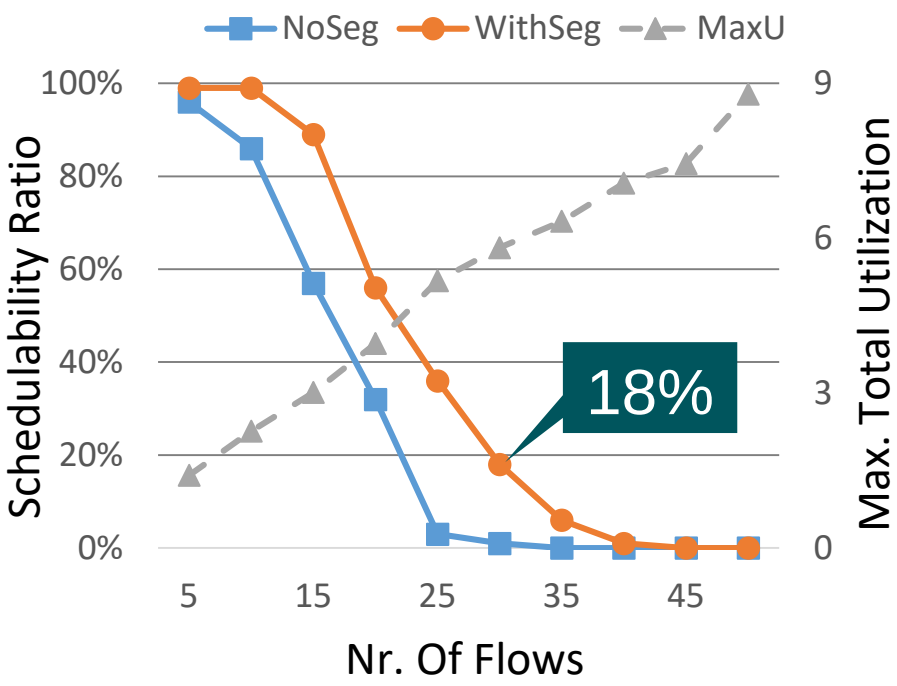
Utilization – [0.003, 0.1]

Utilization – [0.01, 0.2]

Evaluation (2/4)

- Packet size – [5, 25]



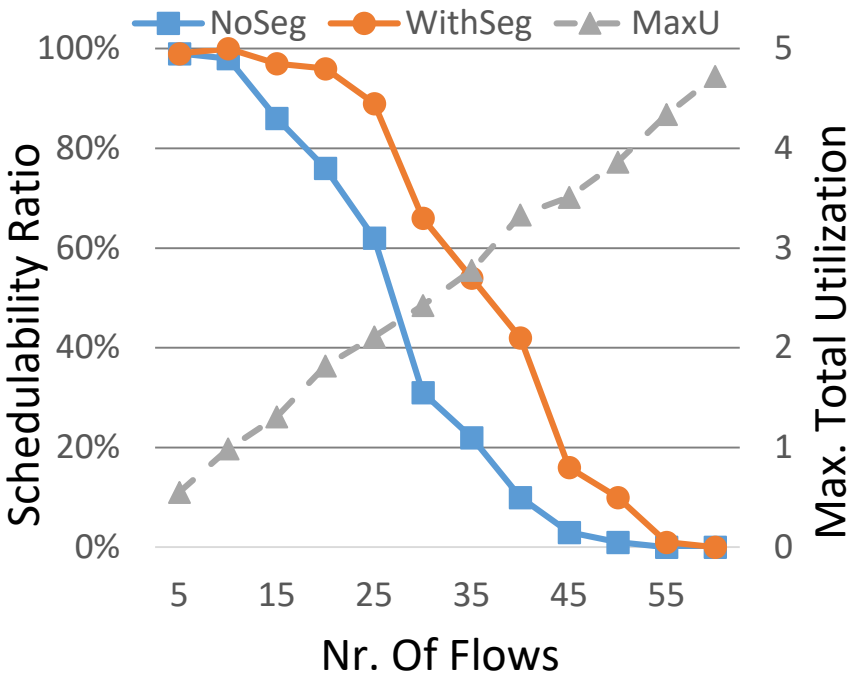


Utilization – [0.003, 0.1]

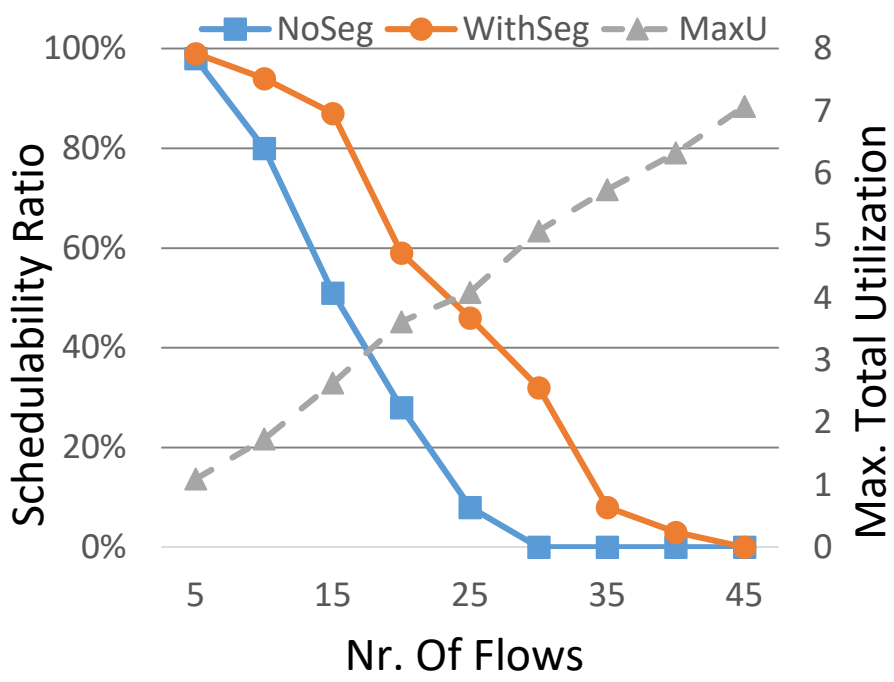
Utilization – [0.01, 0.2]

Evaluation (3/4)

- Packet size – [5, 50]



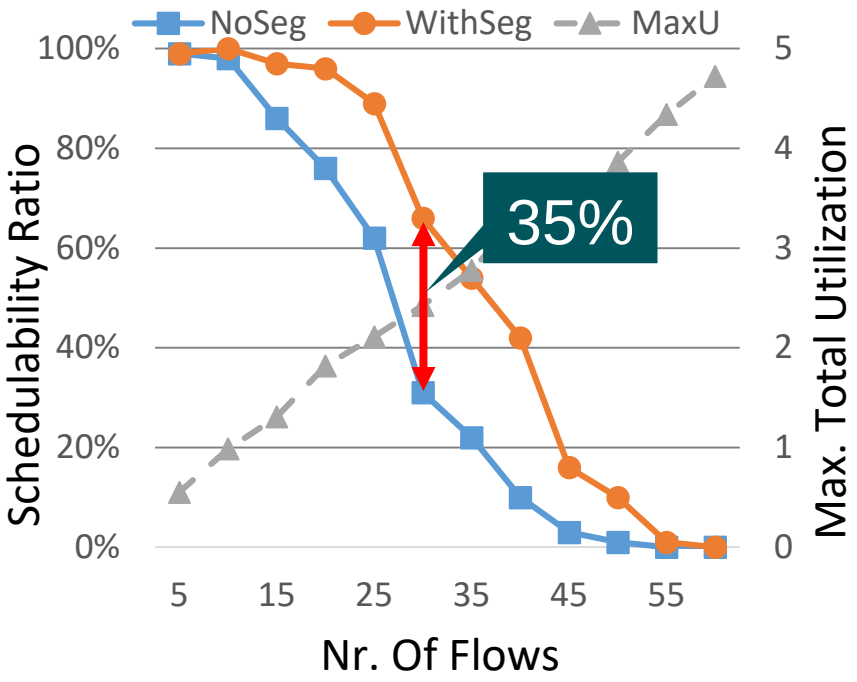
Utilization – [0.003, 0.1]



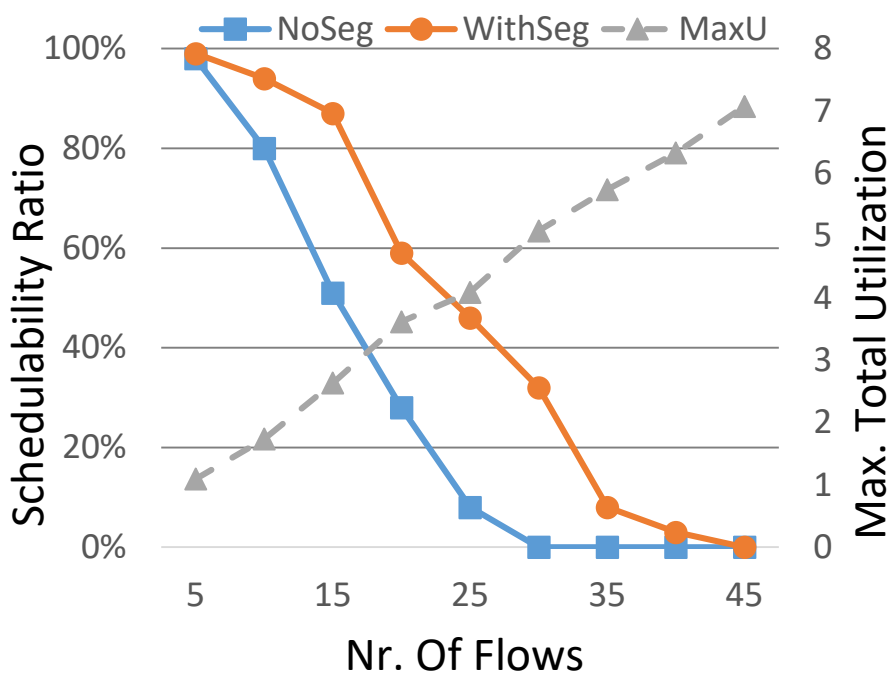
Utilization – [0.01, 0.2]

Evaluation (3/4)

- Packet size – [5, 50]



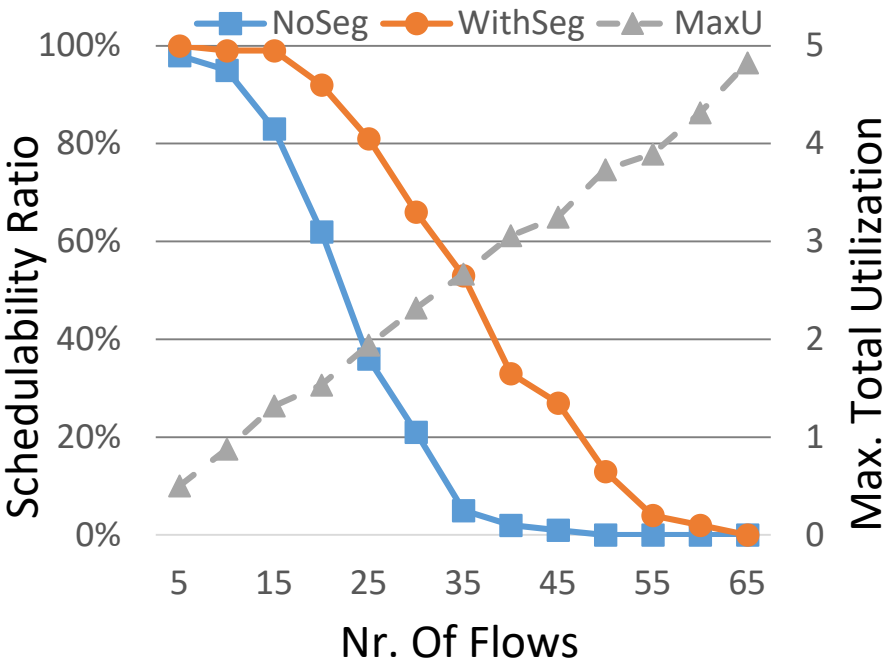
Utilization – [0.003, 0.1]



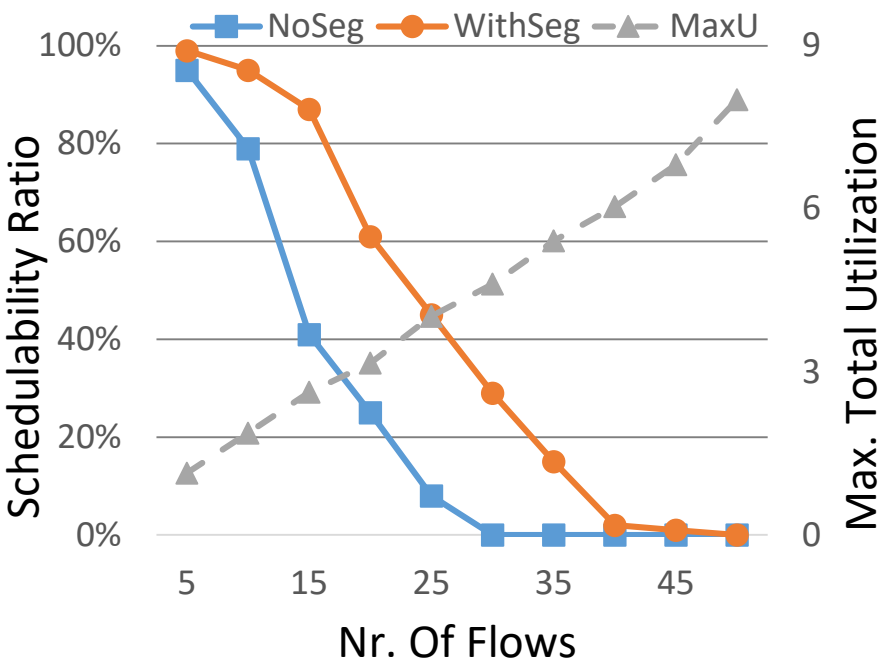
Utilization – [0.01, 0.2]

Evaluation (4/4)

- Packet size – [5, 100]



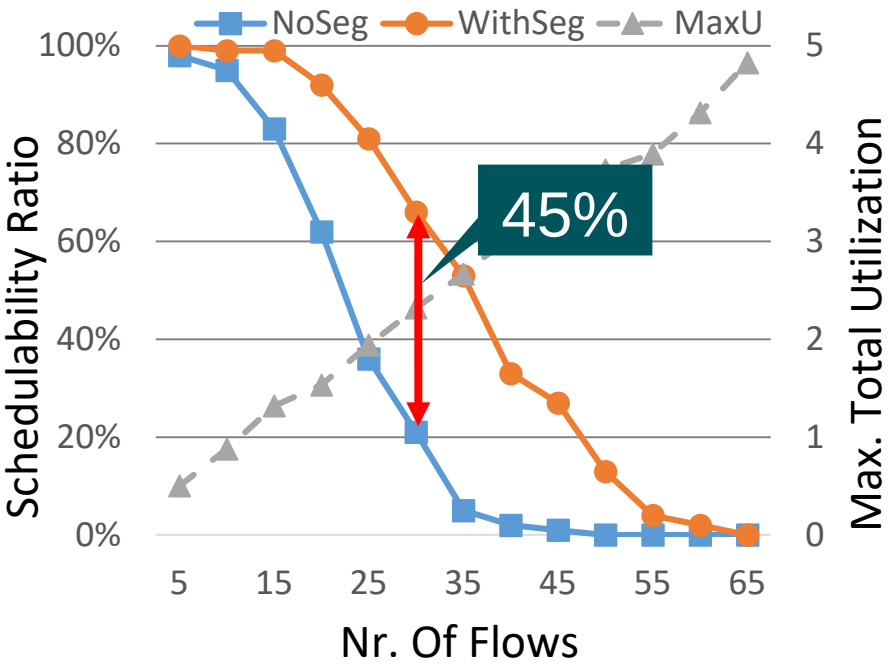
Utilization – [0.003, 0.1]



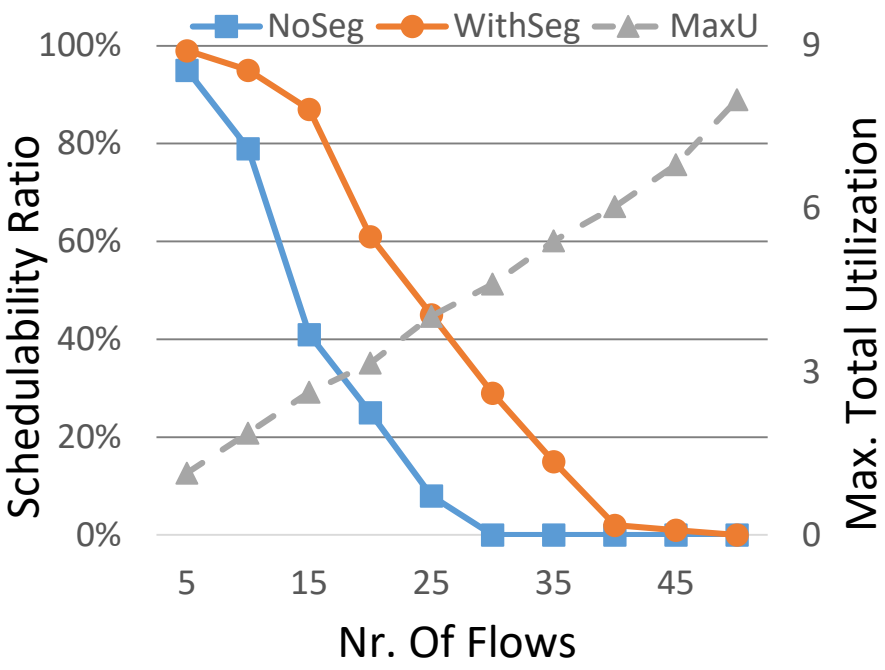
Utilization – [0.01, 0.2]

Evaluation (4/4)

- Packet size – [5, 100]



Utilization – [0.003, 0.1]



Utilization – [0.01, 0.2]

Conclusion

- ✓ We present a segmentation-based approach to improve schedulability of real-time traffic in RRA-based NoCs.

- ✓ The solution also considers how to add best-effort traffic in the same NoC while guaranteeing the timeliness of real-time traffic

  - Combining with the task mapping
    - Further improve schedulability of real-time traffic
    - Further reduce latency of best-effort traffic

Thank you for the attention!

Questions?

MÄLARDALEN UNIVERSITY
SWEDEN