Configurations and Optimizations of TDMA Schedules for Periodic Packet Communication on Networks on Chip

Tim Harde[1], Matthias Freier[2],
Georg von der Brüggen[1], and Jian-Jia Chen[1]

[1]**TU Dortmund University, Germany**
[2]**Robert Bosch GmbH, Germany**

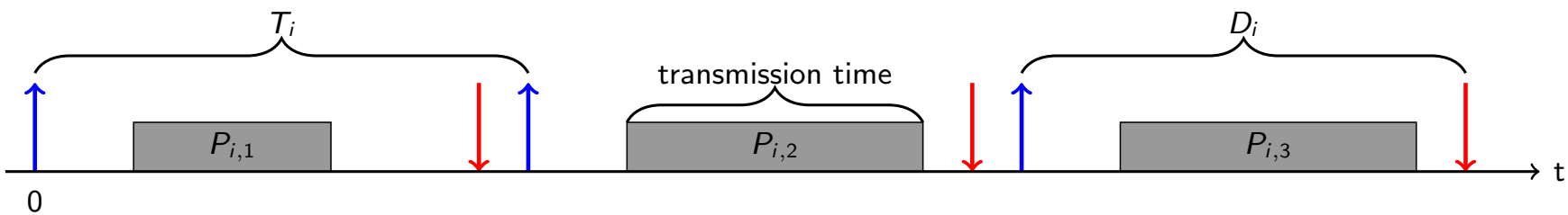
11.10.2018

Table of Content

Communication Tasks

Periodic communication task $\tau_i = (T_i, D_i, P_i, R_i, r_i, d_i)$

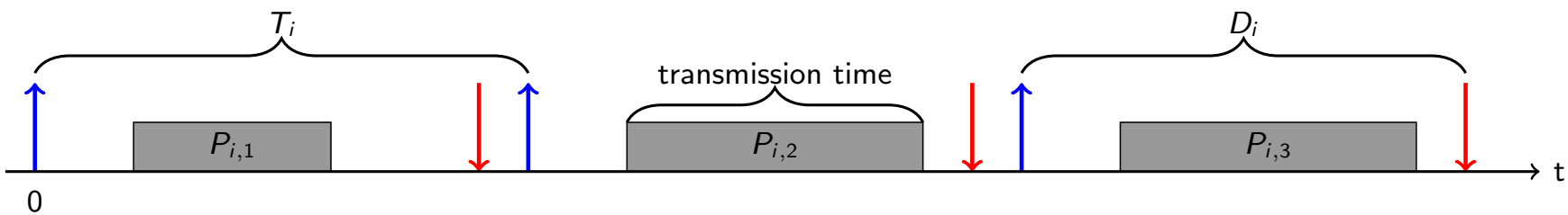
- period T_i
- relative deadline D_i
- packet size P_i
- route R_i through the network
- source node r_i
- destination node d_i
- infinite number of **packets** $P_{i,j}$



Communication Tasks

Periodic communication task $\tau_i = (T_i, D_i, P_i, R_i, r_i, d_i)$

- period T_i
- relative deadline D_i, implicit deadline: $D_i = T_i$
- packet size P_i
- route R_i through the network
- source node r_i
- destination node d_i
- infinite number of **packets** $P_{i,j}$



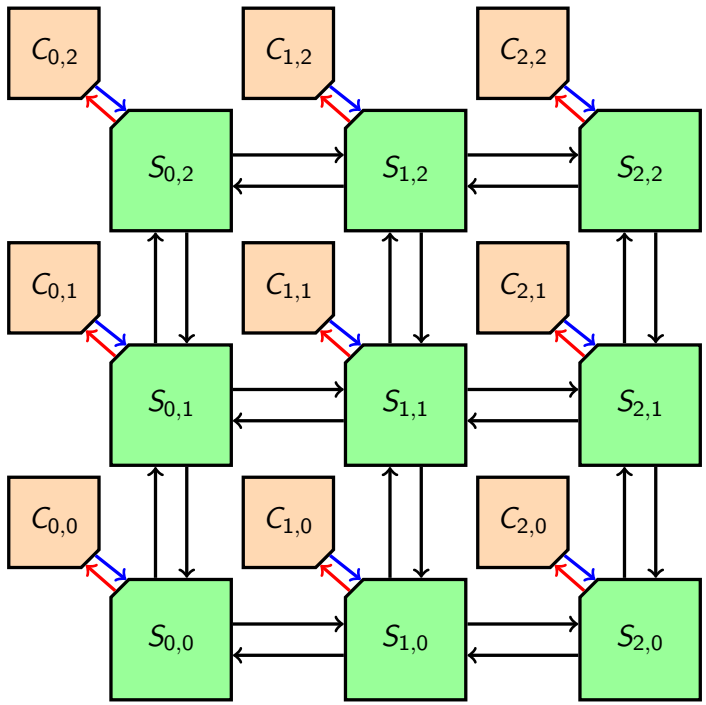
communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$

- harmonic: $\forall \tau_i, \tau_j \in \tau$: if $T_i < T_j$ then $T_j = a \cdot T_i$, $a \in \mathbb{N}$

Network on Chip

Architecture

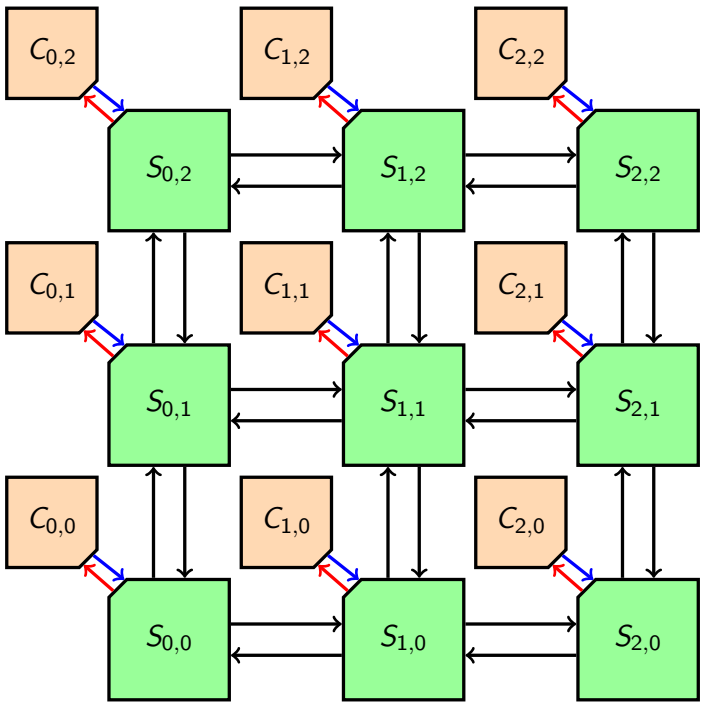
- $r \times s$ 2D-mesh
- directed graph:
  - nodes:
    - cores $C_{x,y}$
    - switches $S_{x,y}$
  - edges:
    - links $L^{node_1}_{node_2}$



Network on Chip

Architecture

- $r \times s$ 2D-mesh
- directed graph:
  - nodes:
    - cores $C_{x,y}$
    - switches $S_{x,y}$
  - edges:
    - links $L_{node_2}^{node_1}$



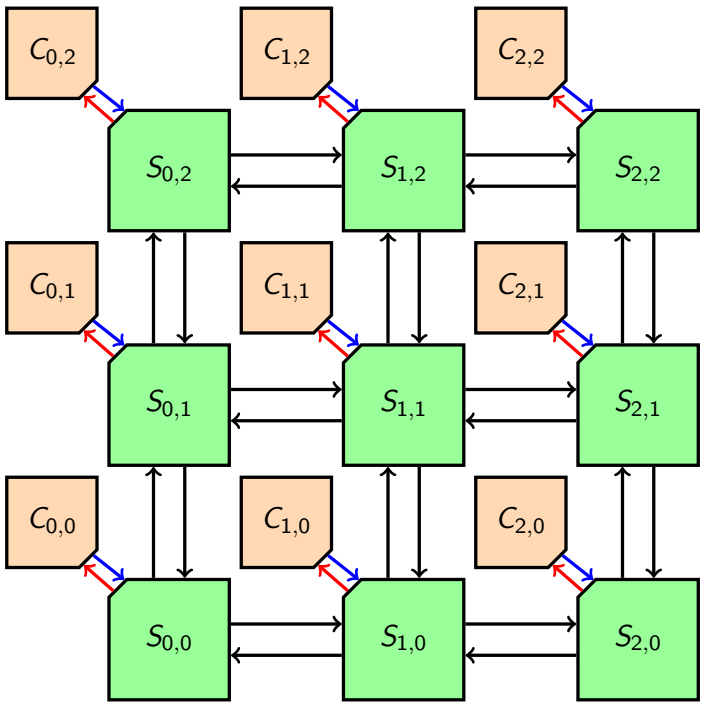
Packets and Flits

- flow control units (flits):
  - atomic units of communication
  - maximum payload p_{max}: 32 bits
- packet $P_{i,j}$:
  - segmented into flits if $P_i > p_{max}$

Network-on-Chip

Cores $C_{x,y}$

- execute computational tasks
- exchange messages (packets)
- components:
  - processing unit
  - memory
  - network interface (up-/downlink)



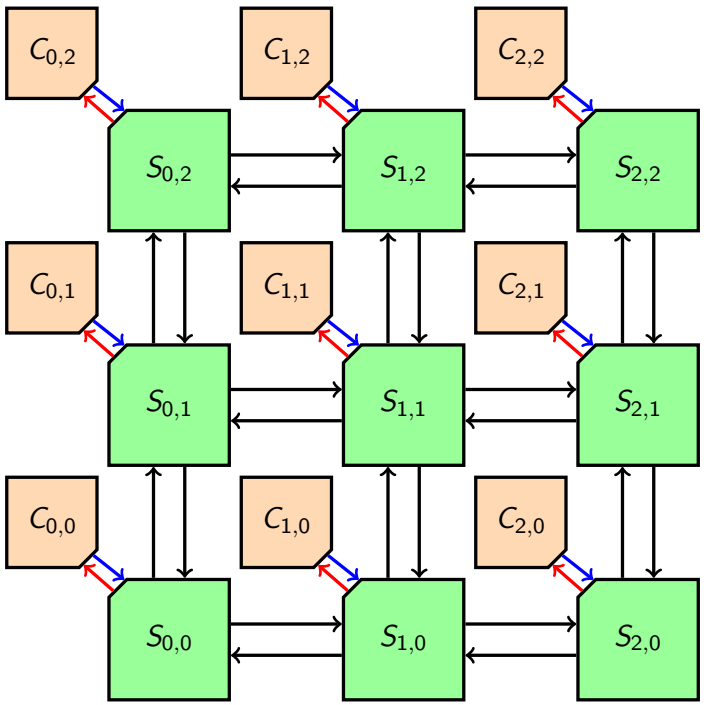
Network-on-Chip

Cores $C_{x,y}$

- execute computational tasks
- exchange messages (packets)
- components:
  - processing unit
  - memory
  - network interface (up-/downlink)

Switches $S_{x,y}$

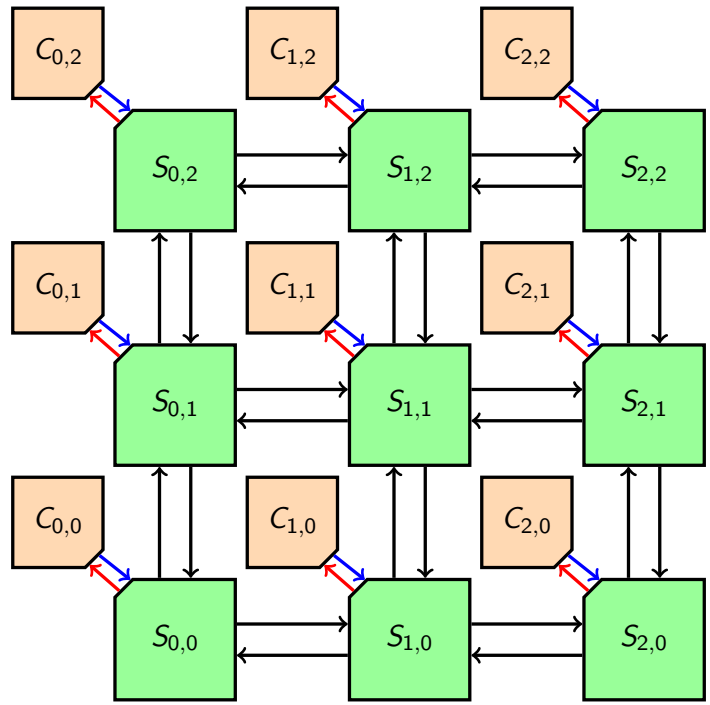
- *forward* flits through the NoC
- components:
  - routing logic
  - arbiter
  - switching fabric



Network-on-Chip

Links $L_{node_2}^{node_1}$

- unidirectional connection
- flit forwarding
  - constant link latency l



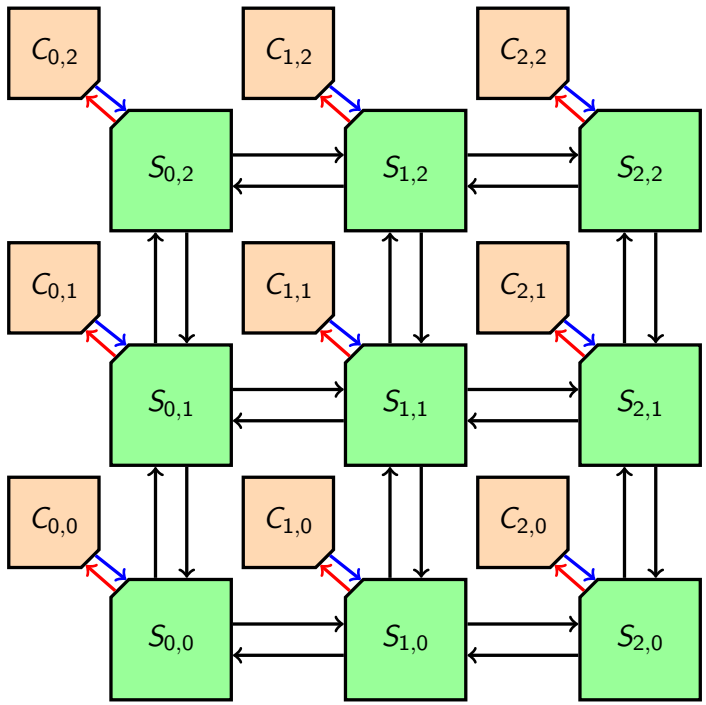
Network-on-Chip

Links $L^{node_1}_{node_2}$

- unidirectional connection
- flit forwarding
  - constant link latency l

Synchronicity

- fully synchronous system
  - global clock for NIs and switches



Network-on-Chip

Links $L^{node_1}_{node_2}$

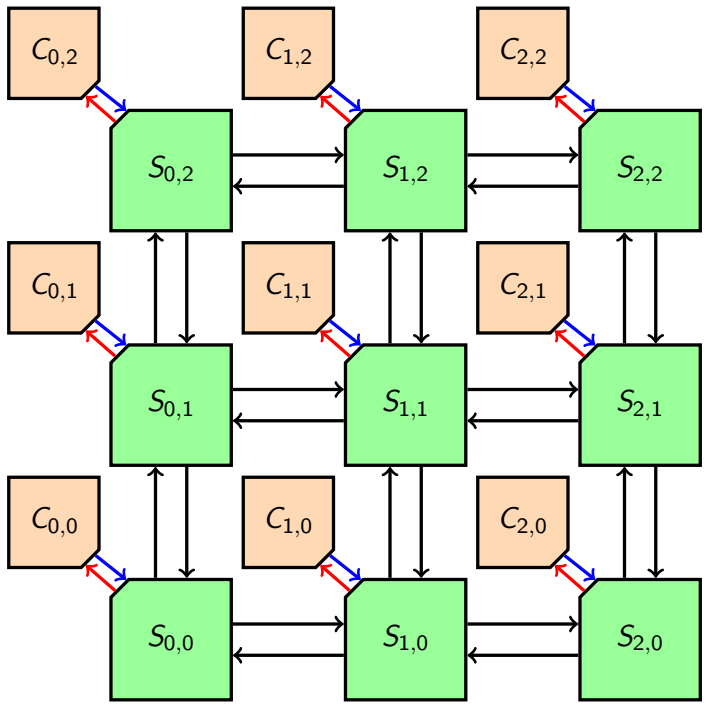
- unidirectional connection
- flit forwarding
  - constant link latency l

Synchronicity

- fully synchronous system
  - global clock for NIs and switches

Routing

- XY-Routing
  - deterministic (in-order delivery)
  - deadlock-free



Network-on-Chip

Links $L_{node_2}^{node_1}$

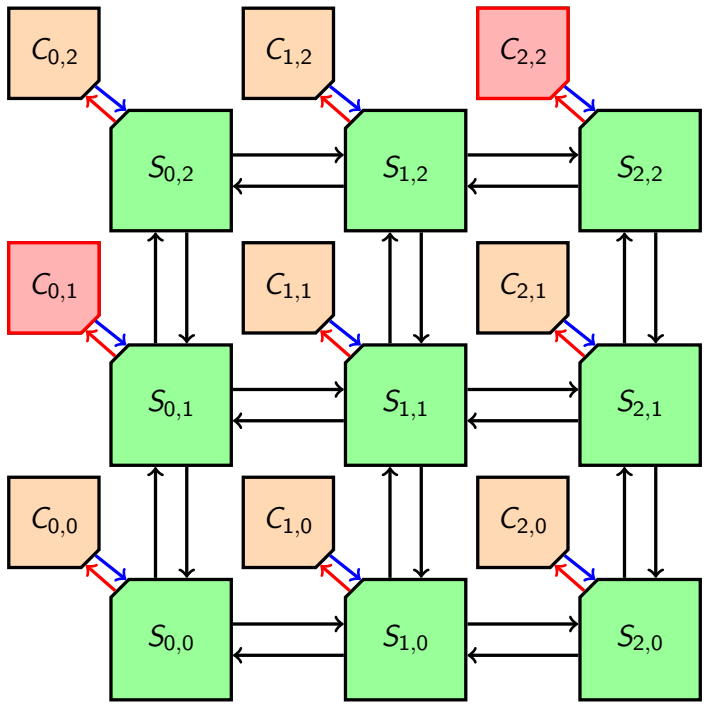
- unidirectional connection
- flit forwarding
  - constant link latency l

Synchronicity

- fully synchronous system
  - global clock for NIs and switches

Routing

- XY-Routing
  - deterministic (in-order delivery)
  - deadlock-free



Example: communication task τ_i with $r_i = C_{0,1}$ and $d_i = C_{2,2}$

Network-on-Chip

Links $L^{node_1}_{node_2}$

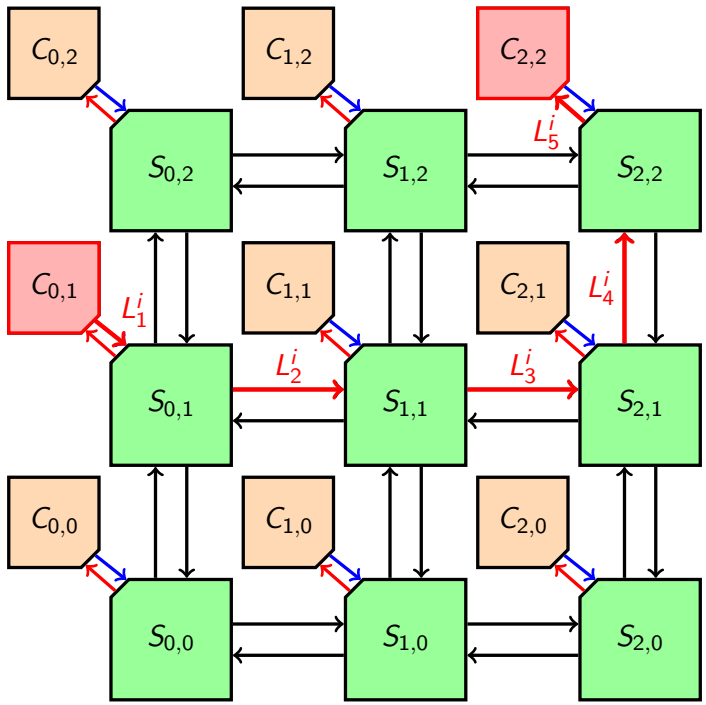
- unidirectional connection
- flit forwarding
  - constant link latency l

Synchronicity

- fully synchronous system
  - global clock for NIs and switches

Routing

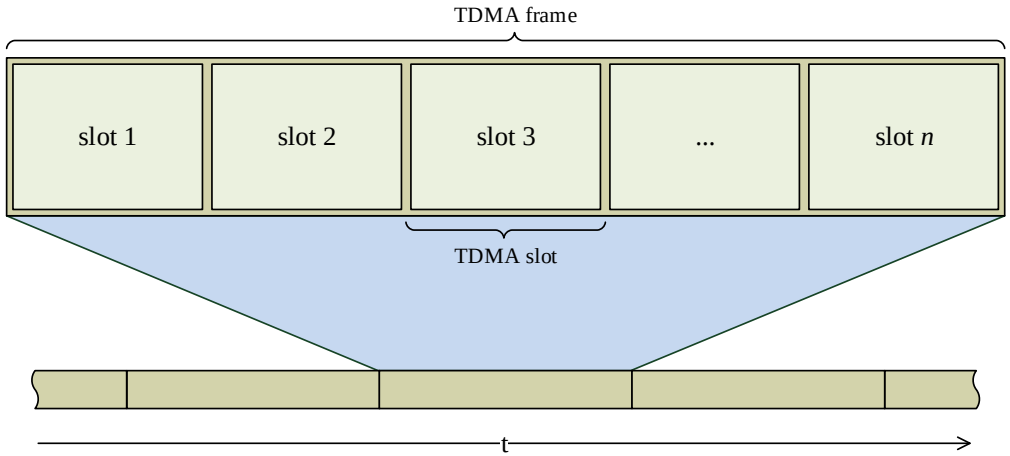
- XY-Routing
  - deterministic (in-order delivery)
  - deadlock-free



Example: communication task τ_i with $r_i = C_{0,1}$ and $d_i = C_{2,2}$

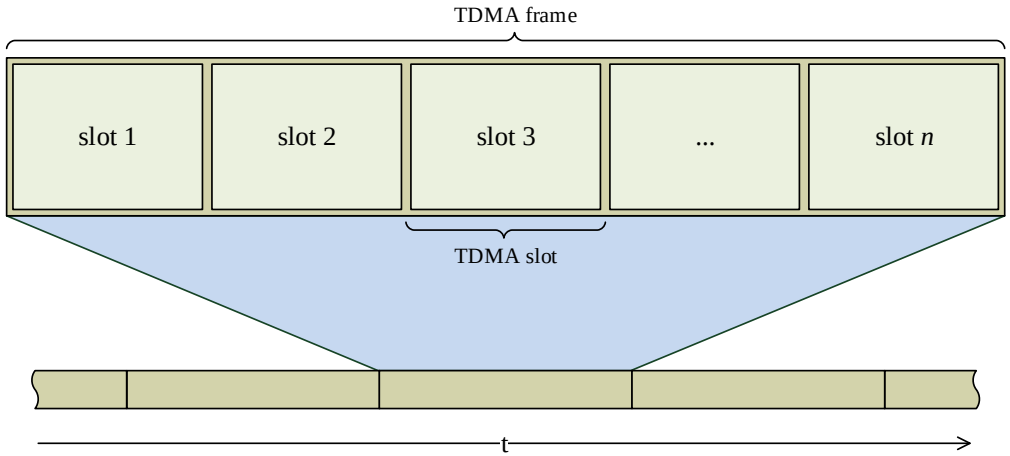
$\Rightarrow$ route R_i: $L^{C_{0,1}}_{S_{0,1}}$ $L^{S_{0,1}}_{S_{1,1}}$ $L^{S_{1,1}}_{S_{2,1}}$ $L^{S_{2,1}}_{S_{2,2}}$ $L^{S_{2,2}}_{C_{2,2}}$

Time Division Multiple Access (TDMA)



Idea: partitioning the access time to a specific resource

Time Division Multiple Access (TDMA)



Idea: partitioning the access time to a specific resource

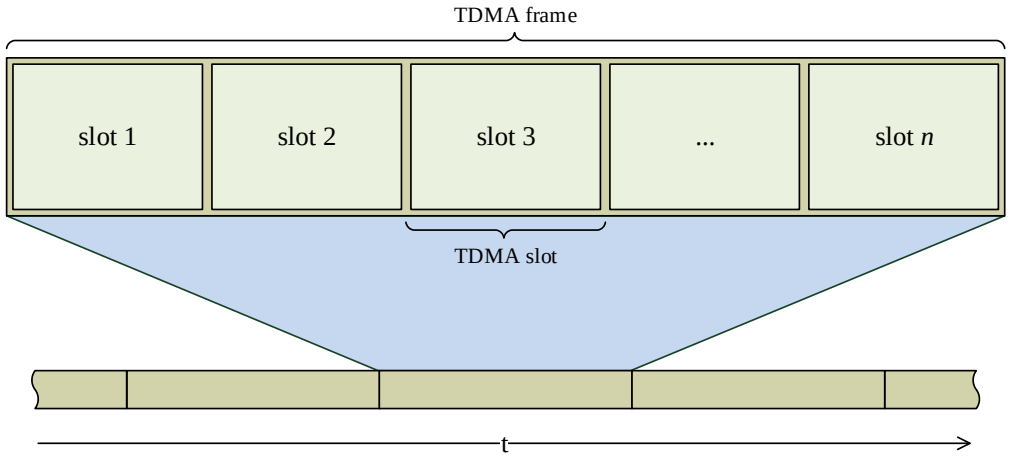
Advantages

- bandwidth guarantees
- isolation
- predictable timing behavior
- good analyzability

Disadvantages

- global synchronization
- global overhead
- tough design problem

Time Division Multiple Access (TDMA)



Idea: partitioning the access time to a specific resource

Advantages

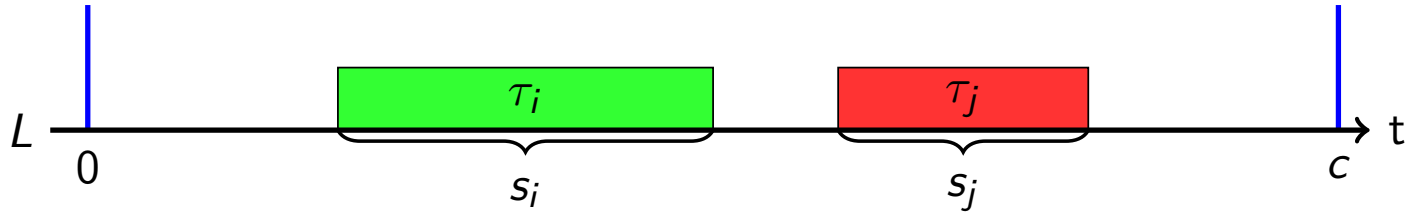
- bandwidth guarantees
- isolation
- predictable timing behavior
- good analyzability

Disadvantages

- global synchronization
- global overhead
- tough design problem

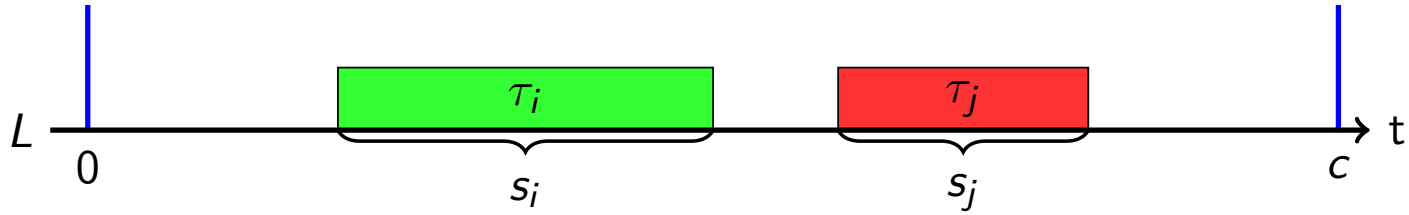
Problem: parameter selection: frame size, slot size and alignment

Global TDMA Approach



- synchronous execution of the TDMA schedule on all nodes
- *start* and *end* of a TDMA slot can be defined arbitrarily
  $\Rightarrow$ unused bandwidth possible

Global TDMA Approach

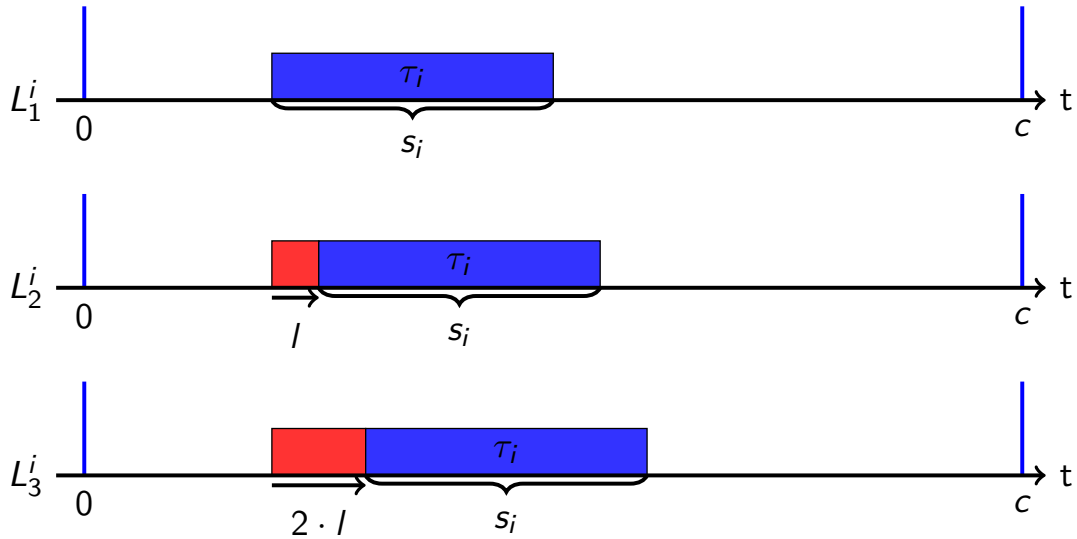


- synchronous execution of the TDMA schedule on all nodes
- *start* and *end* of a TDMA slot can be defined arbitrarily
  $\Rightarrow$ unused bandwidth possible

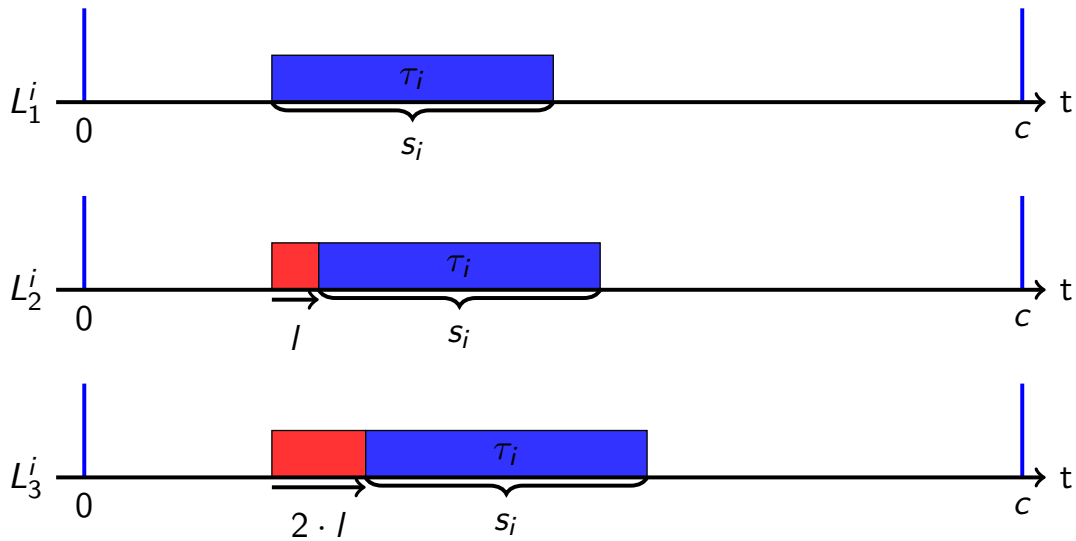
TDMA Parameters

- TDMA cycle length c
  $\Rightarrow$ identical for all TDMA arbiters
- TDMA slot length s_i
  $\Rightarrow$ sufficient bandwidth to forward the flits of τ_i

TDMA Slot Aligning

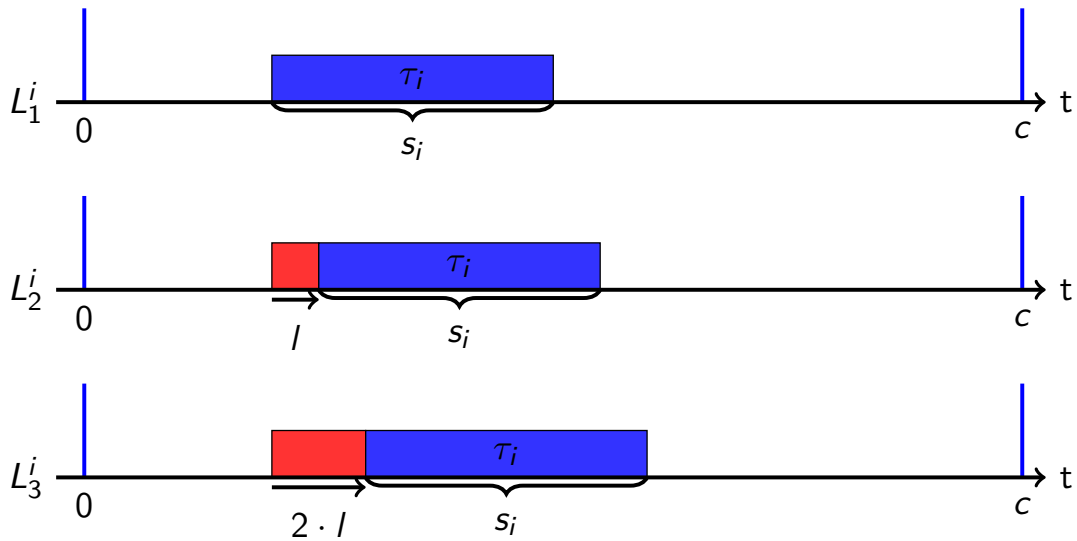


TDMA Slot Aligning



- align TDMA slots on consecutively traversed links
  - $\Rightarrow$ immediate forwarding of flits
  - $\Rightarrow$ no buffering required
  - $\Rightarrow$ no flow control mechanism required (bandwidth guarantee)
  - $\Rightarrow$ no contention from other packets (isolation property)

TDMA Slot Aligning



- align TDMA slots on consecutively traversed links
  - $\Rightarrow$ immediate forwarding of flits
  - $\Rightarrow$ no buffering required
  - $\Rightarrow$ no flow control mechanism required (bandwidth guarantee)
  - $\Rightarrow$ no contention from other packets (isolation property)

TDMA slot assignment respecting the alignment property

TDMA Parameter Selection - Cycle Length

TDMA Cycle Length

- identical for all TDMA arbiters in the NoC

TDMA Parameter Selection - Cycle Length

TDMA Cycle Length

- identical for all TDMA arbiters in the NoC

TDMA cycle length $c \in \{T_1, T_2, ..., T_n\}$

TDMA Parameter Selection - Cycle Length

TDMA Cycle Length

- identical for all TDMA arbiters in the NoC

TDMA cycle length $c \in \{T_1, T_2, ..., T_n\}$

Harmonic Property of τ:

$\Rightarrow$ packets are released at the same offset of the TDMA cycle

$\Rightarrow$ identical response times for consecutive packet releases

TDMA Parameter Selection - Cycle Length

TDMA Cycle Length

- identical for all TDMA arbiters in the NoC

TDMA cycle length $c \in \{T_1, T_2, ..., T_n\}$

Harmonic Property of τ:

$\Rightarrow$ packets are released at the same offset of the TDMA cycle

$\Rightarrow$ identical response times for consecutive packet releases

Fragmentation:

- $T_i < c$: integer number of packets releases per TDMA cycle
- $T_i = c$: one packet release per TDMA cycle
- $T_i > c$: one packet release every $\frac{T_i}{c}$ TDMA cycles

TDMA Parameter Selection - Cycle Length

TDMA Cycle Length

- identical for all TDMA arbiters in the NoC

TDMA cycle length $c \in \{T_1, T_2, ..., T_n\}$

Harmonic Property of τ:

$\Rightarrow$ packets are released at the same offset of the TDMA cycle

$\Rightarrow$ identical response times for consecutive packet releases

Fragmentation:

- $T_i < c$: integer number of packets releases per TDMA cycle
- $T_i = c$: one packet release per TDMA cycle
- $T_i > c$: one packet release every $\frac{T_i}{c}$ TDMA cycles

Directly provides the slot size s_i for each task

TDMA Schedule Design - Overview

Given:

- communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$
  - $\Rightarrow$ harmonic periods
  - $\Rightarrow$ implicit deadlines
- $r \times s$ NoC platform

TDMA Schedule Design - Overview

Given:

- communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$
  - $\Rightarrow$ harmonic periods
  - $\Rightarrow$ implicit deadlines
- $r \times s$ NoC platform

The problem is to determine

1. the TDMA cycle length
2. the TDMA slot size s_i
3. the offset of each TDMA slot in the TDMA cycles

all tasks can be feasibly scheduled and the TDMA slots are aligned

TDMA Schedule Design - Overview

Given:

- communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$
  - $\Rightarrow$ harmonic periods
  - $\Rightarrow$ implicit deadlines
- $r \times s$ NoC platform

The problem is to determine

1. the TDMA cycle length: $c \in \{T_1, T_2, ..., T_n\}$
2. the TDMA slot size s_i
3. the offset of each TDMA slot in the TDMA cycles

all tasks can be feasibly scheduled and the TDMA slots are aligned

TDMA Schedule Design - Overview

Given:

- communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$
  - $\Rightarrow$ harmonic periods
  - $\Rightarrow$ implicit deadlines
- $r \times s$ NoC platform

The problem is to determine

1. the TDMA cycle length: $c \in \{T_1, T_2, ..., T_n\}$
2. the TDMA slot size s_i: fragmentation, # releases
3. the offset of each TDMA slot in the TDMA cycles

all tasks can be feasibly scheduled and the TDMA slots are aligned

TDMA Schedule Design - Overview

Given:

- communication task set $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$
  - $\Rightarrow$ harmonic periods
  - $\Rightarrow$ implicit deadlines
- $r \times s$ NoC platform

The problem is to determine

1. the TDMA cycle length: $c \in \{T_1, T_2, ..., T_n\}$
2. the TDMA slot size s_i: fragmentation, # releases
3. the offset of each TDMA slot in the TDMA cycles

all tasks can be feasibly scheduled and the TDMA slots are aligned

Approaches:

- First-Fit Greedy Heuristic
- Rectangular Scheduling (solver-based approach)

TDMA Design - Greedy Heuristics

- Communication Task Scheduling
  - tasks are scheduled successively
  - tasks are scheduled in a specific *order*

TDMA Design - Greedy Heuristics

- Communication Task Scheduling
  - tasks are scheduled successively
  - tasks are scheduled in a specific *order*

- TDMA Slot Assignment:
  - start as early as possible
  - alignment property is maintained
  - isolation property is maintained (non-overlapping slots)

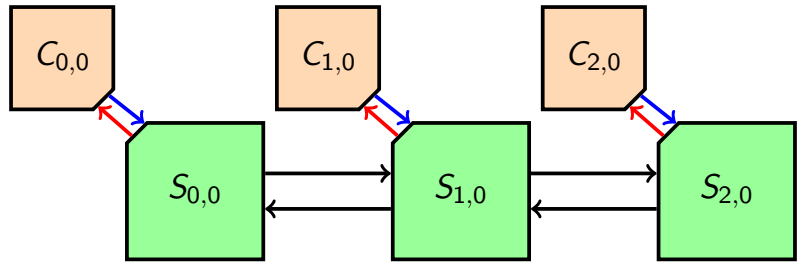
TDMA Design - Greedy Heuristics

- Communication Task Scheduling
  - tasks are scheduled successively
  - tasks are scheduled in a specific *order*
- TDMA Slot Assignment:
  - start as early as possible
  - alignment property is maintained
  - isolation property is maintained (non-overlapping slots)
- Ordering Strategies:
  - Largest Utilization First (LUF)
  - Smallest Utilization First (SUF)
  - Largest Period First (LPF)
  - Smallest Period First (SPF)
  - Random
  - Largest Hop Count First + {LUF, SUF,LPF,SPF}
  - Hop-Count Weighted LUF (HC-W-LUF)

TDMA Design - Greedy Heuristics (Example)

Communication Tasks

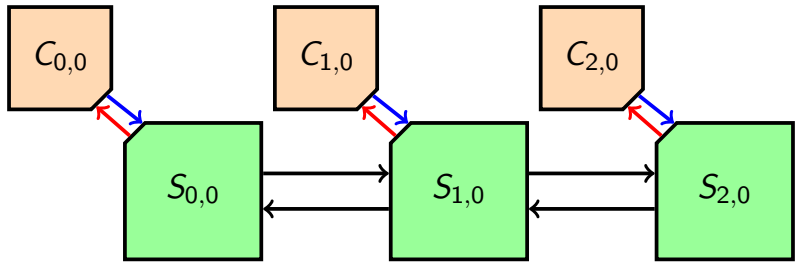
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$
  - $r_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$
  - $r_2 = C_{0,0}$, $d_1 = C_{2,0}$



TDMA Design - Greedy Heuristics (Example)

Communication Tasks

- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$
  - $r_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$
  - $r_2 = C_{0,0}$, $d_1 = C_{2,0}$



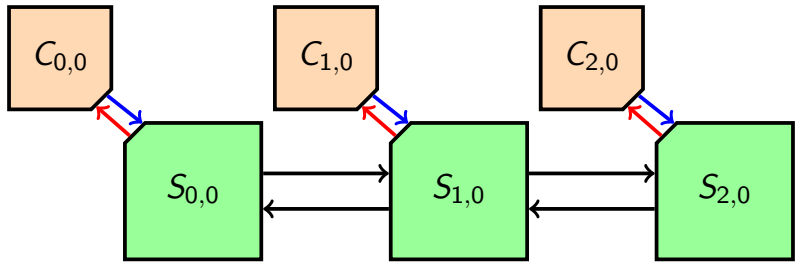
NoC Parameters

- link latency: $l = 1$

TDMA Design - Greedy Heuristics (Example)

Communication Tasks

- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$
  - $r_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$
  - $r_2 = C_{0,0}$, $d_1 = C_{2,0}$



NoC Parameters

- link latency: $l = 1$

TDMA Parameters

- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$

TDMA Design - Greedy Heuristics (Example)

Communication Tasks

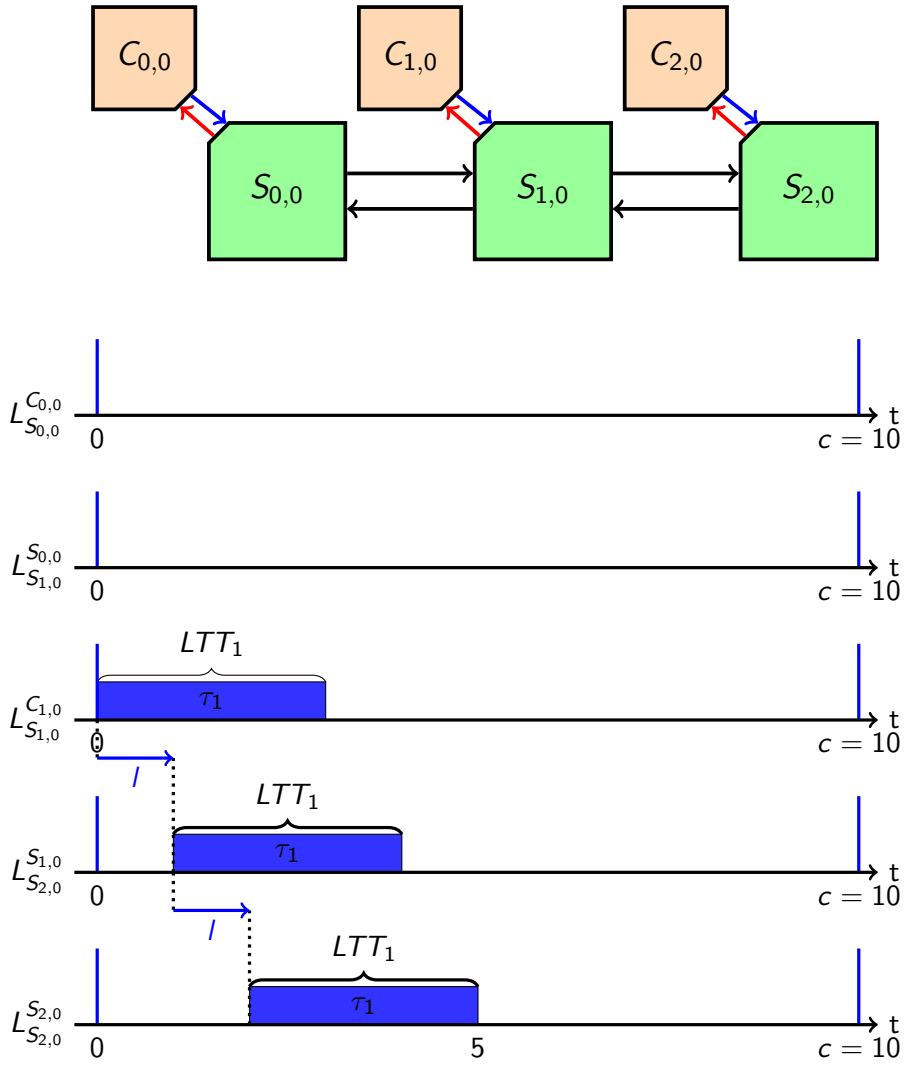
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$
  - $r_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$
  - $r_2 = C_{0,0}$, $d_1 = C_{2,0}$

NoC Parameters

- link latency: $l = 1$

TDMA Parameters

- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$



TDMA Design - Greedy Heuristics (Example)

Communication Tasks

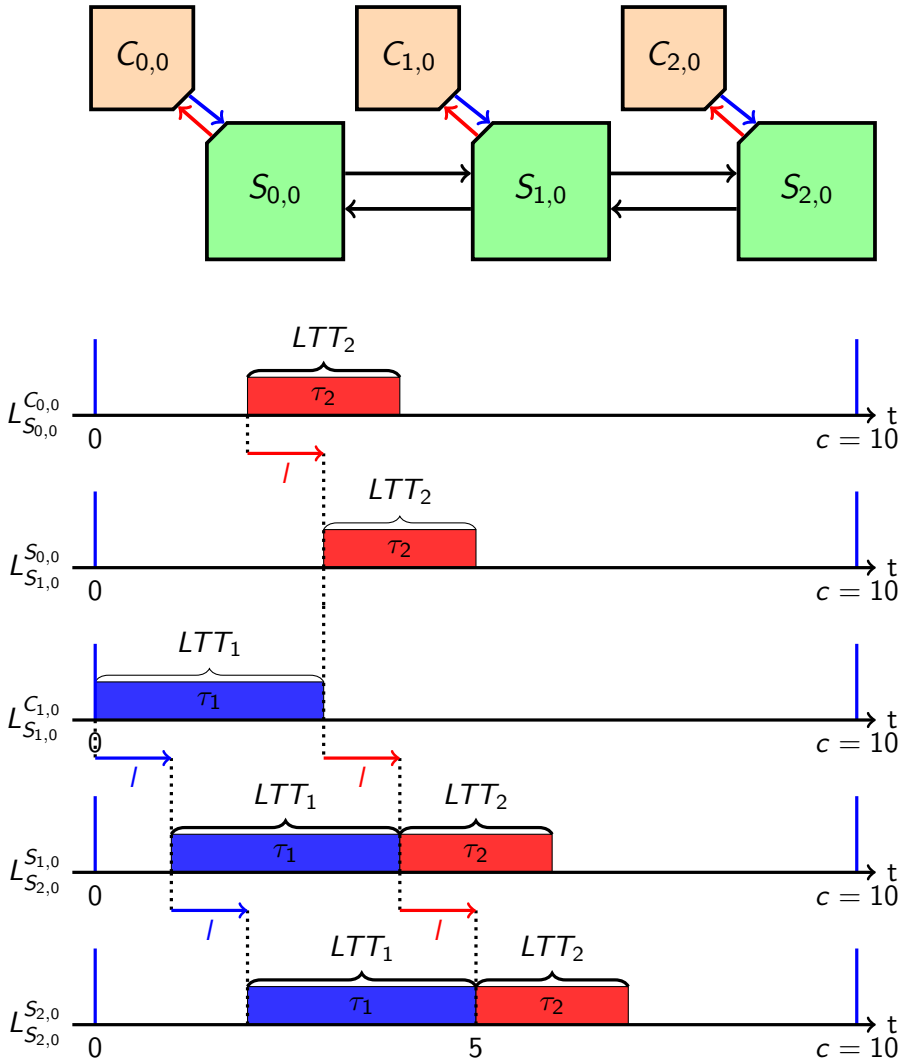
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$
  - $r_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$
  - $r_2 = C_{0,0}$, $d_1 = C_{2,0}$

NoC Parameters

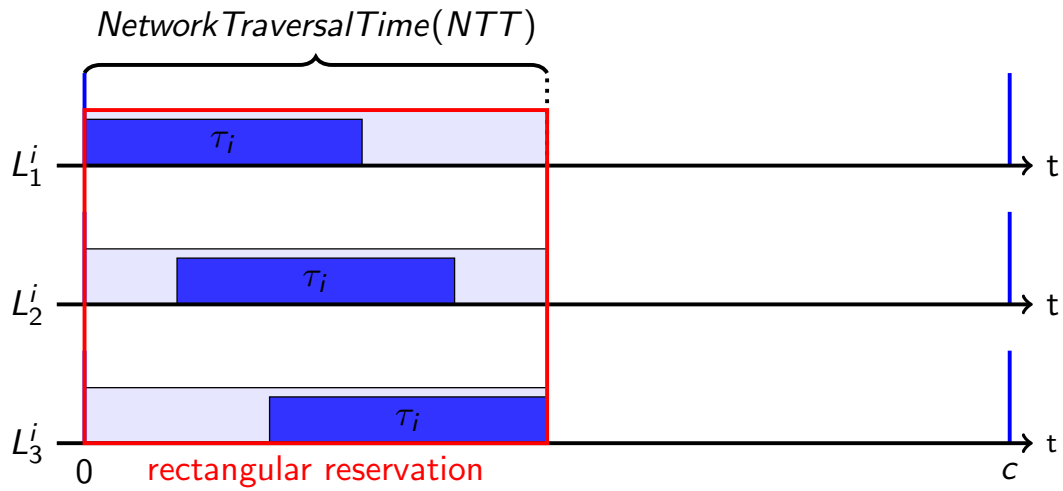
- link latency: $l = 1$

TDMA Parameters

- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$



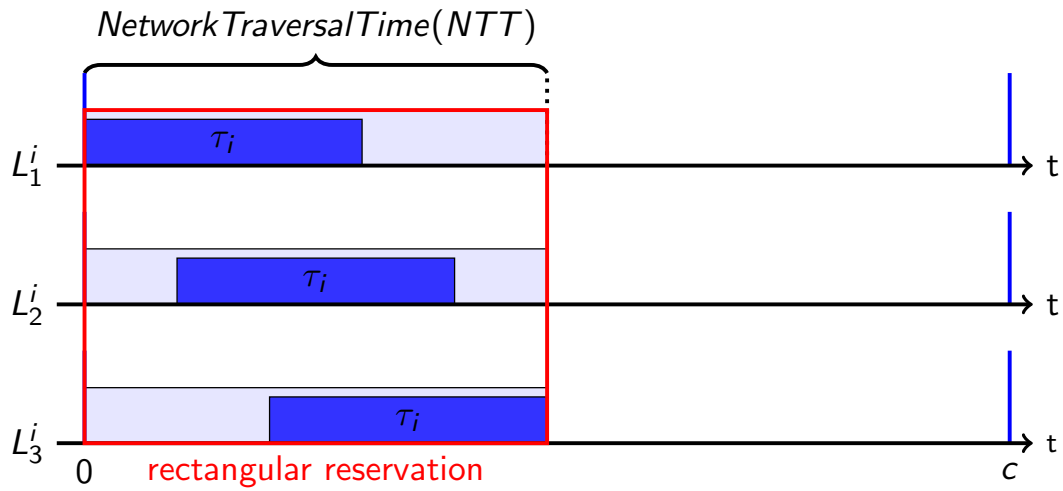
TDMA Design - Rectangular Scheduling



Scheduling Strategy

- Resource Reservations
  - reservations of width Network Traversal Time (NTT)
  - isolation property: non-overlapping *rectangles*
  - arrangement of the *rectangular reservations*:
    - $\Rightarrow$ Integer Linear Programming (ILP) formulation
    - $\Rightarrow$ Constraint Programming (CP) formulation

TDMA Design - Rectangular Scheduling



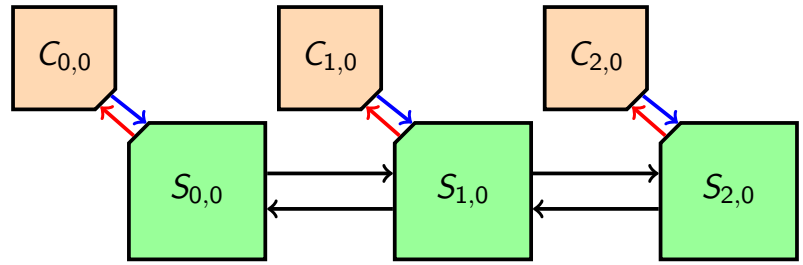
Scheduling Strategy

- Resource Reservations
  - reservations of width Network Traversal Time (NTT)
  - isolation property: non-overlapping *rectangles*
  - arrangement of the *rectangular reservations*:
    - $\Rightarrow$ Integer Linear Programming (ILP) formulation
    - $\Rightarrow$ Constraint Programming (CP) formulation
- TDMA Slot Assignment:
  - slots are allocated within rectangular reservations
  - maintain alignment property

TDMA Design - Rectangular Scheduling (Example)

Communication Tasks

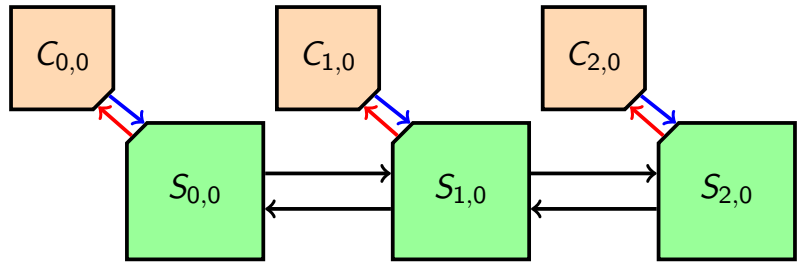
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$, $NTT_1 = 5$
  - $s_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$, $NTT_2 = 5$
  - $s_2 = C_{0,0}$, $d_1 = C_{2,0}$



TDMA Design - Rectangular Scheduling (Example)

Communication Tasks

- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$, $NTT_1 = 5$
  - $s_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$, $NTT_2 = 5$
  - $s_2 = C_{0,0}$, $d_1 = C_{2,0}$



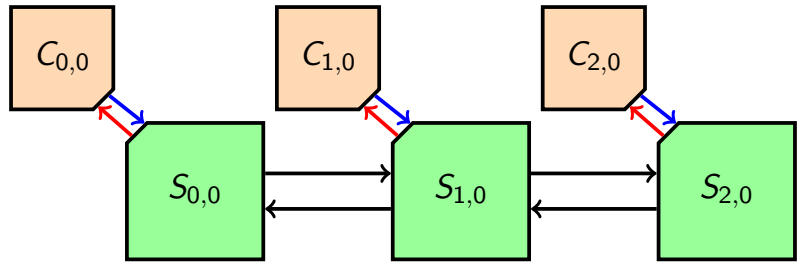
NoC Parameters

- link latency: $l = 1$

TDMA Design - Rectangular Scheduling (Example)

Communication Tasks

- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$, $NTT_1 = 5$
  - $s_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$, $NTT_2 = 5$
  - $s_2 = C_{0,0}$, $d_1 = C_{2,0}$



NoC Parameters

- link latency: $l = 1$

TDMA Parameters

- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$

TDMA Design - Rectangular Scheduling (Example)

Communication Tasks

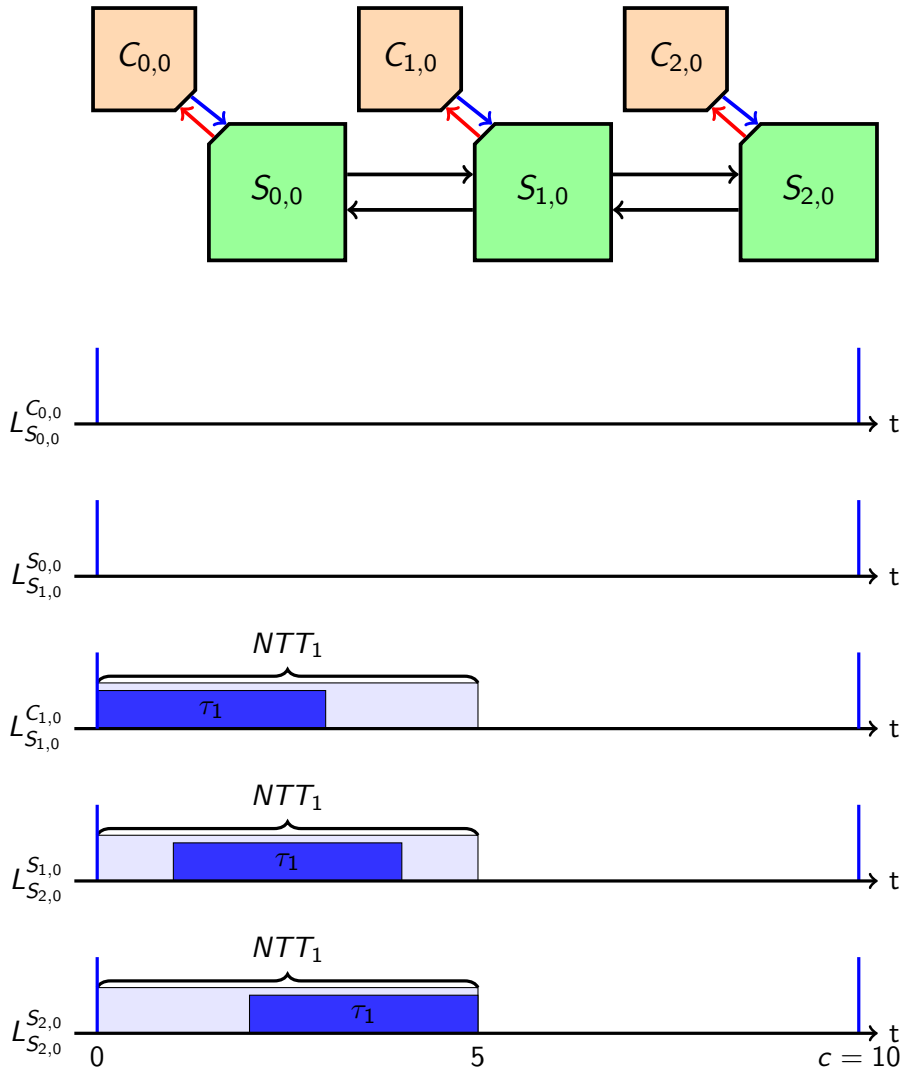
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$, $NTT_1 = 5$
  - $s_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$, $NTT_2 = 5$
  - $s_2 = C_{0,0}$, $d_1 = C_{2,0}$

NoC Parameters

- link latency: $l = 1$

TDMA Parameters

- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$



TDMA Design - Rectangular Scheduling (Example)

Communication Tasks

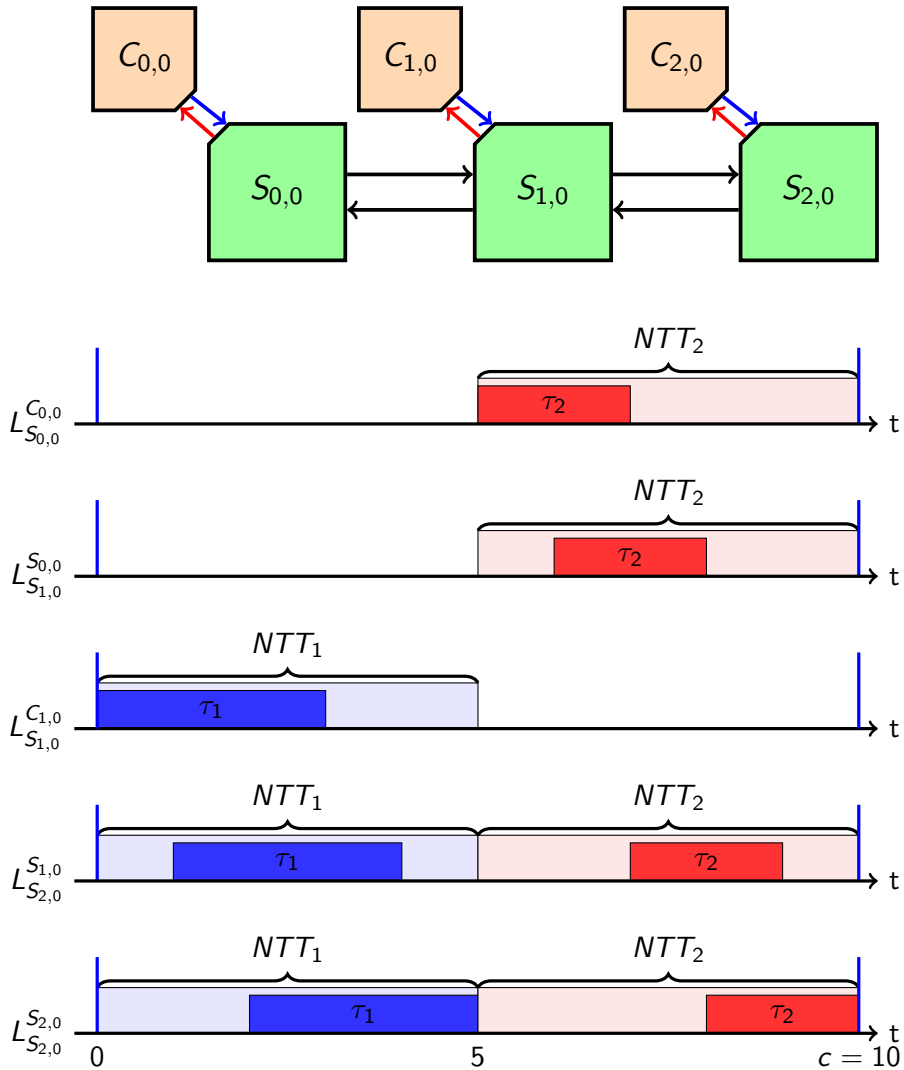
- task τ_1:
  - $T_1 = 10$, $LTT_1 = 3$, $NTT_1 = 5$
  - $s_1 = C_{1,0}$, $d_1 = C_{2,0}$
- task τ_2:
  - $T_2 = 10$, $LTT_2 = 2$, $NTT_2 = 5$
  - $s_2 = C_{0,0}$, $d_1 = C_{2,0}$

NoC Parameters

- link latency: $l = 1$

TDMA Parameters

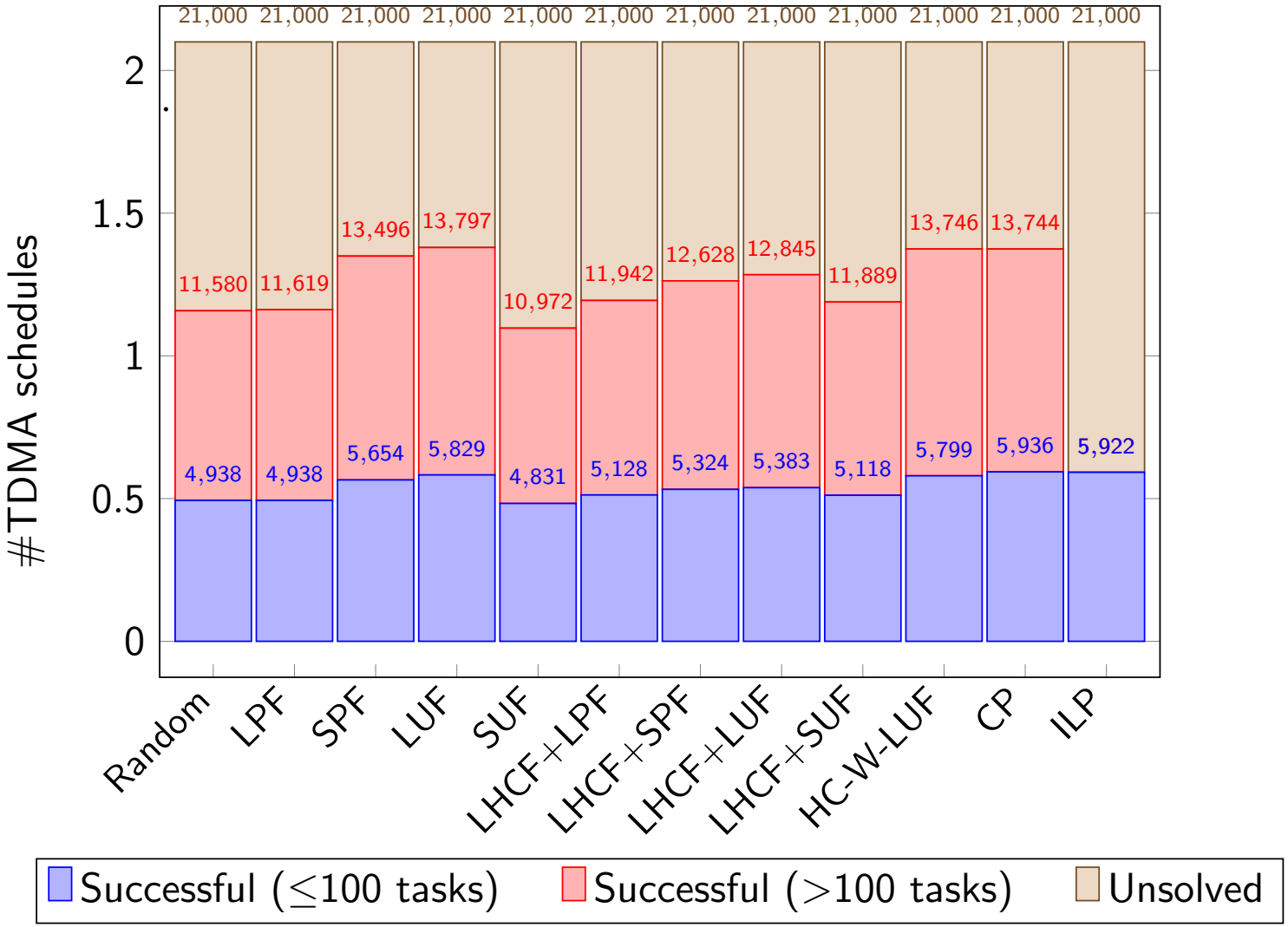
- TDMA cycle length: $c = 10$
- $s_1 = LTT_1 = 3$
- $s_2 = LTT_2 = 2$



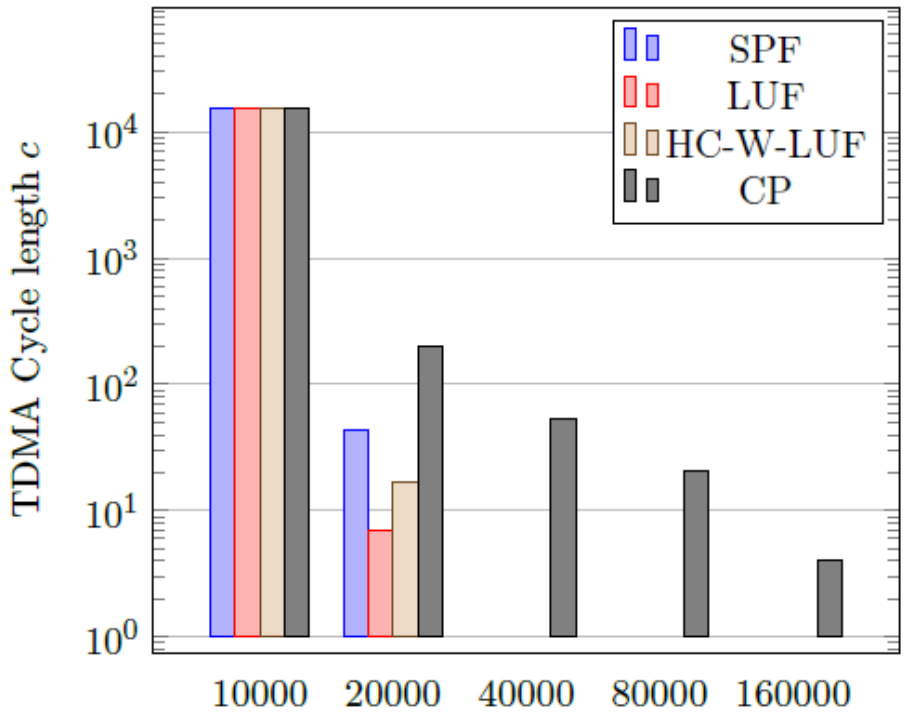
Evaluation - 3×3 parameters

flitPayload (bit)	32
linkLatency (μs)	5
numberOfTasks	20, 25, 30, 35, 40, 45, 50, 100, 200, 300, 400, 500, 750, 1000
utilization (%)	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75
simple strategies **combined strategies** **solver based**	Random, LPF, SPF, LUF, SUF LHCF+{LPF,SPF,LUF,SUF} HC-W-LUF CP, ILP
periods (μs)	10000, 20000, 40000, 80000, 160000
# experiments	100
ILPTimeLimit (s)	60.0
CPTimeLimit (s)	60.0

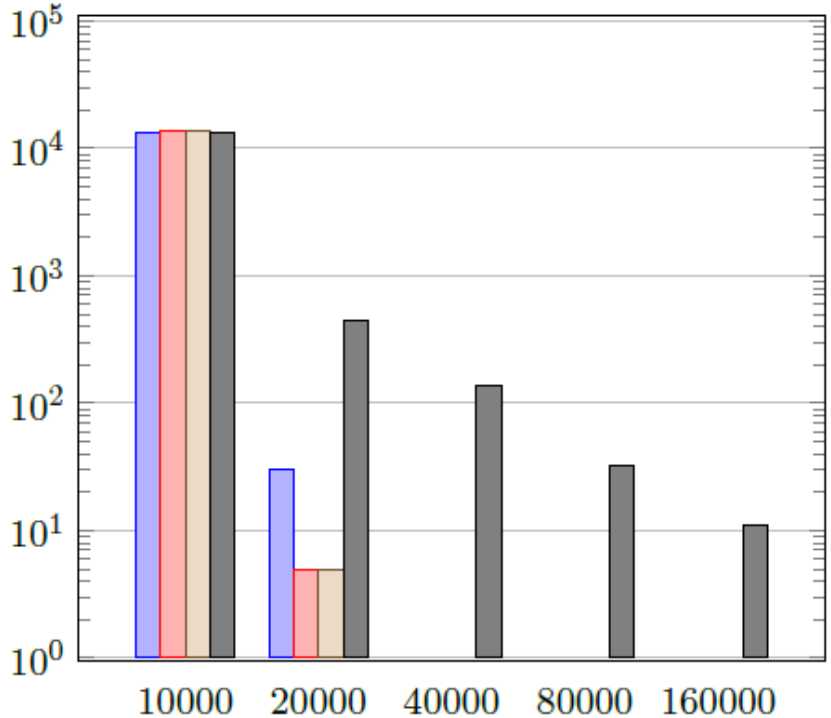
Evaluation - Success Rate Analysis (3×3 topology)



Evaluation - TDMA Cycle Length Analysis



(a) Cycle lengths for the 3×1 NoC topology.



(b) Cycle lengths for the 3×3 NoC topology.

- minimize TDMA slot length
  - $\Rightarrow$ minimize memory to store TDMA schedule
  - $\Rightarrow$ more accurate synchronization (TDMA cycle end)

Conclusion

Results:

- a framework to derive / verify a TDMA schedule for a NoC
- greedy heuristic which provides good results
- a solver-based problem formulation (rectangular scheduling)
- works with clock drift

Conclusion

Results:

- a framework to derive / verify a TDMA schedule for a NoC
- greedy heuristic which provides good results
- a solver-based problem formulation (rectangular scheduling)
- works with clock drift

Evaluation:

- TDMA efficiently applied to NoC with reasonable utilization
  - harmonic periods
  - implicit deadlines
  - synchronous systems
- short TDMA cycle length (frame size)

Conclusion

Results:

- a framework to derive / verify a TDMA schedule for a NoC
- greedy heuristic which provides good results
- a solver-based problem formulation (rectangular scheduling)
- works with clock drift

Evaluation:

- TDMA efficiently applied to NoC with reasonable utilization
  - harmonic periods
  - implicit deadlines
  - synchronous systems
- short TDMA cycle length (frame size)

Thank You!