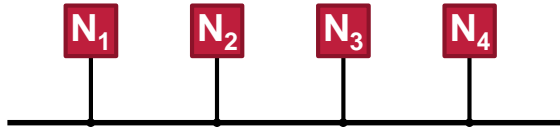
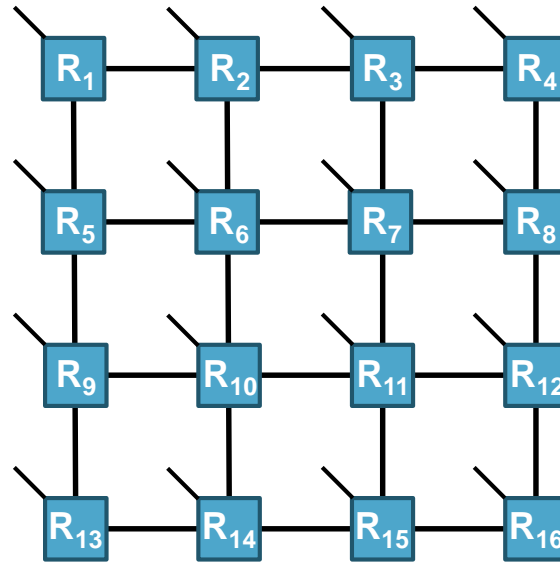


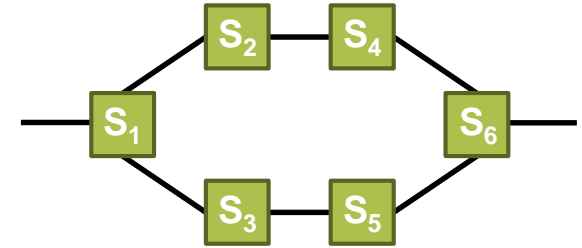
Prelude



CAN



NoC



Ethernet

What do they have in common?

1. Letter “N”

2. Arbitration

3. Transfer

4. 2 & 3 via *different* ~~same~~ medium *s*



Technische
Universität
Braunschweig



INSTITUT FÜR
DATENTECHNIK UND
KOMMUNIKATIONS-
NETZE



Slot-Based Transmission Protocol for Real-Time NoCs

SBT – NoC

Borislav Nikolić, Robin Hofmann, Rolf Ernst

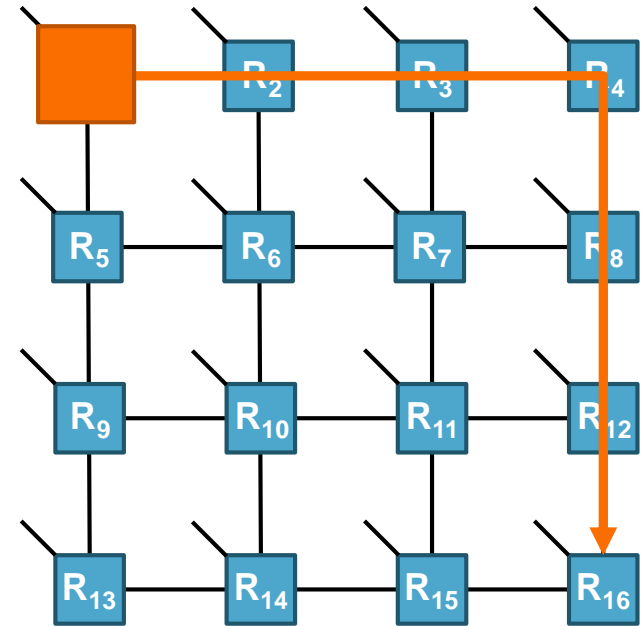
Institut für Datentechnik und Kommunikationsnetze, TU Braunschweig, Germany

ECRTS 2019, Stuttgart, 10-12 July 2019

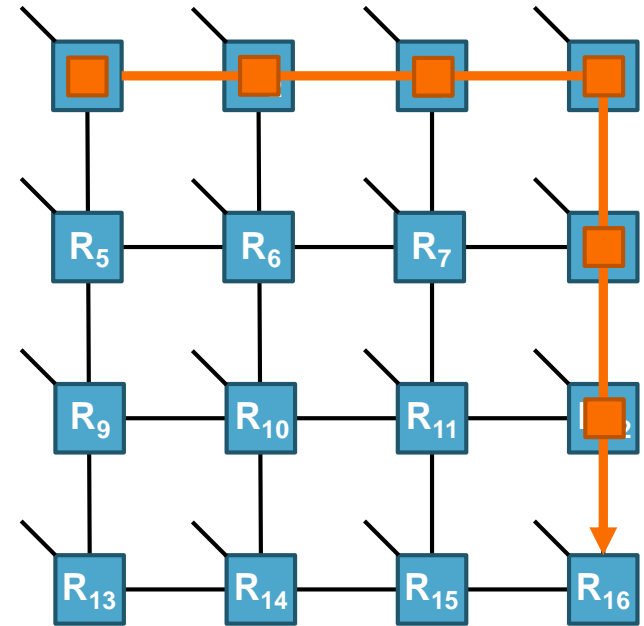
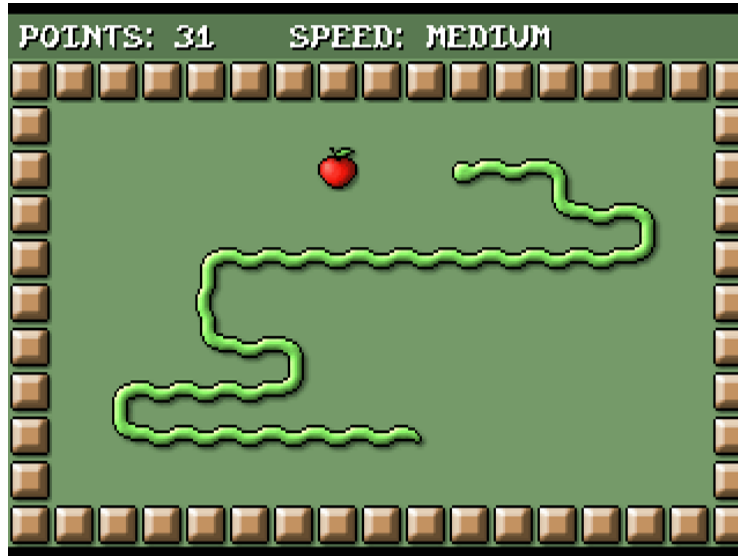
Outline

- **NoC Principles**
- **NoCs in Real-Time Domain**
- **SBT-NoC**
- **Experimental evaluation**
- **Application and Future work**

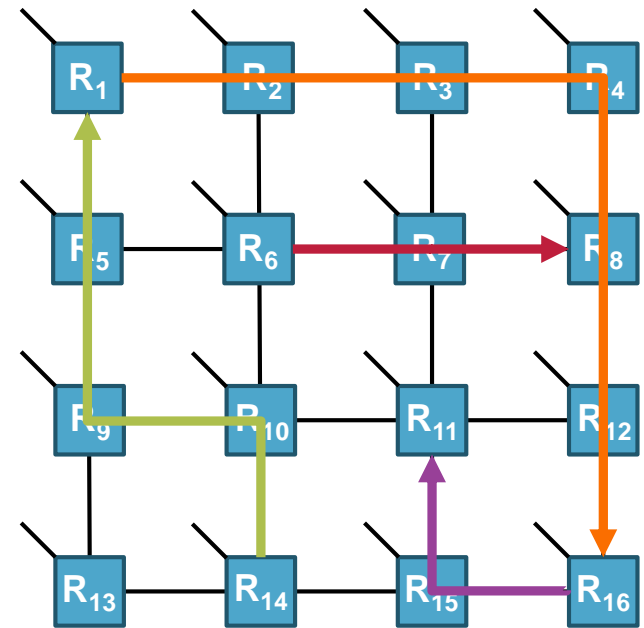
NoC Principles



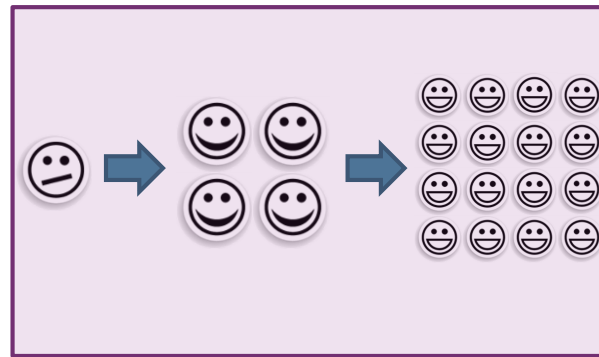
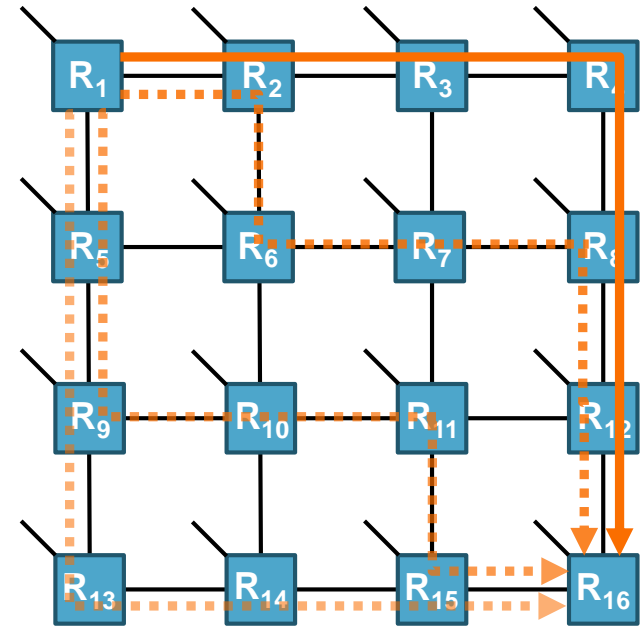
NoC Principles



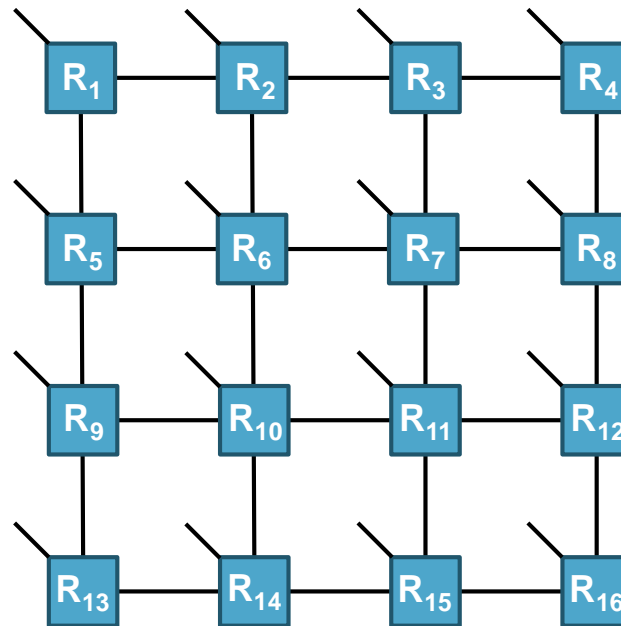
NoC Principles



NoC Principles



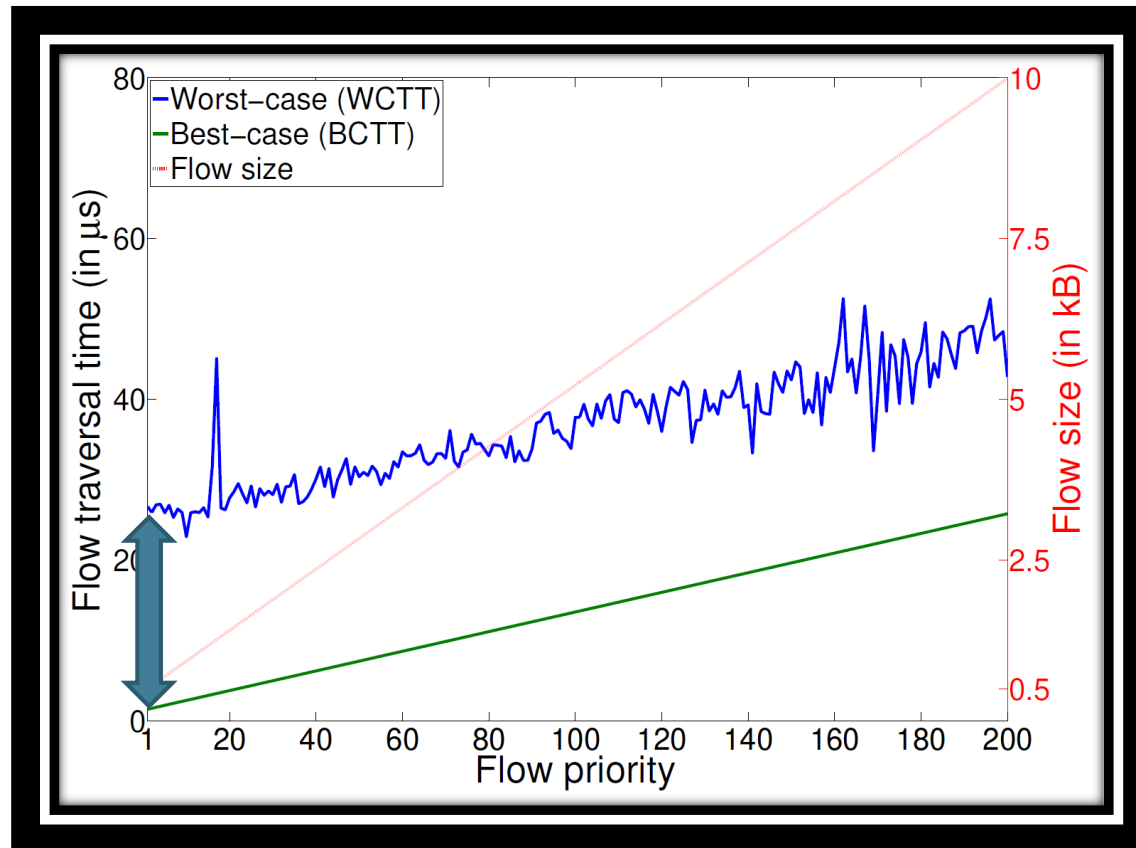
NoC Principles - Summary



NoCs are efficient transmission mediums

Transmission latency calculation valid for ~20 years

Real-Time Analysis of NoCs (Attempt 1)



NoCs without HW support not suitable for RT

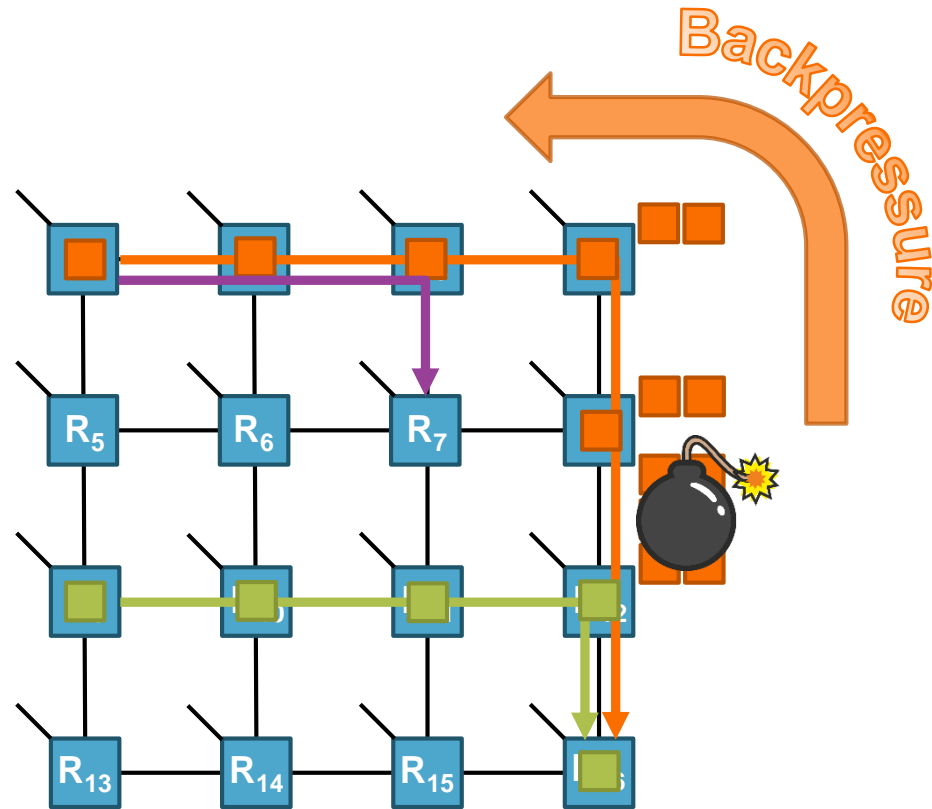
Real-Time Analysis of NoCs (Attempt 2)

Year	Authors, Title
1994	M. W. Mutka. "Using rate monotonic scheduling technology for real-time communications in a wormhole network"
1997	S. L. Hary, F. Ozguner. "Feasibility test for real-time communication using wormhole routing"
1998	B. Kim et al. "A real-time communication method for wormhole switching networks"
2005	Z. Lu et al. "Feasibility analysis of message scheduling in wormhole networks using wormhole routing"

But how come?!

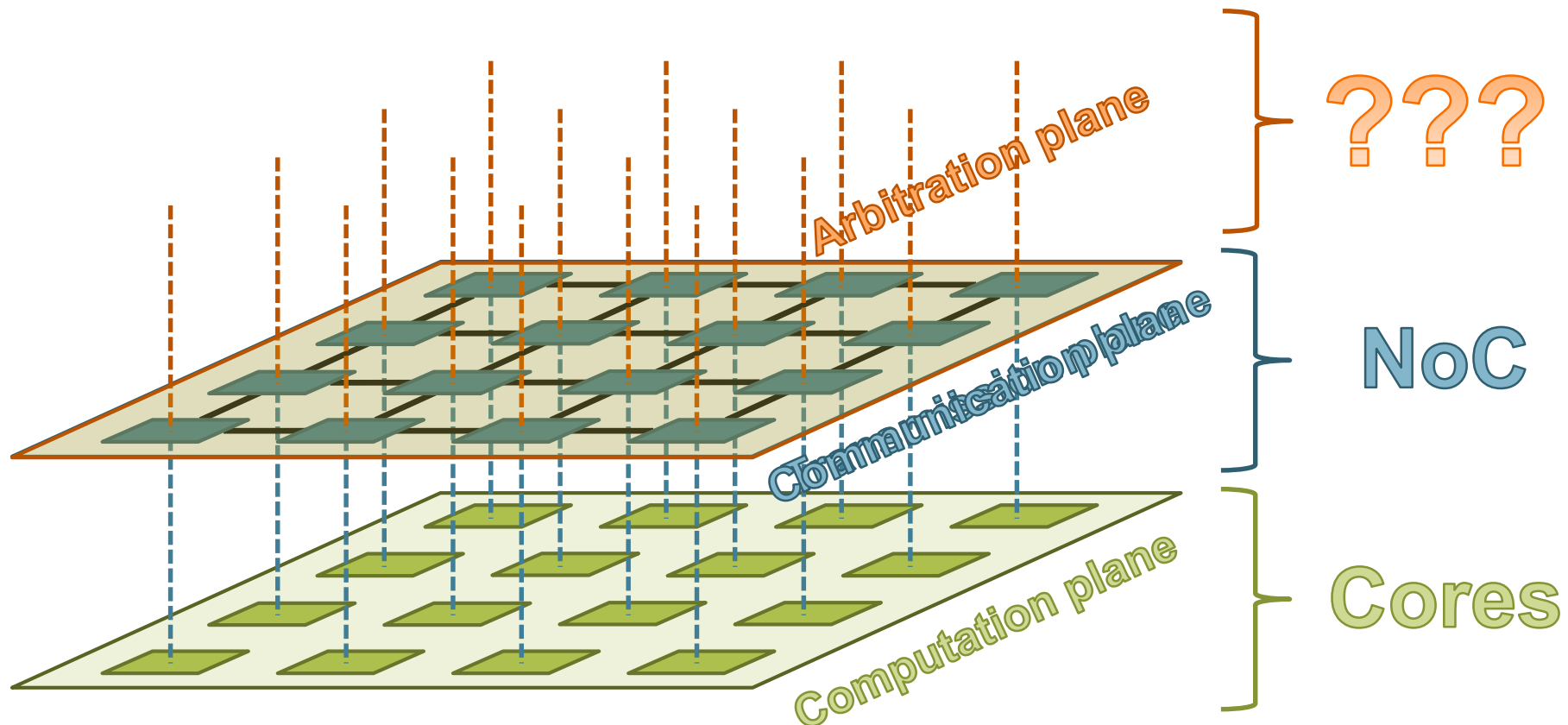
2016	M. Liu et al. "Tighter time analysis for real-time traffic in on-chip networks with shared priorities"
2016	H. Kashif, H. Patel. "Buffer Space Allocation for Real-Time Priority-Aware Networks"
2016	Q. Xiong et al. "Real-Time Analysis for Wormhole NoC: Revisited and Revised"
2018	Q. Xiong et al. "Extending real-time analysis for wormhole NoCs"
2018	L.S. Indrusiak, A. Burns, B. Nikolić. "Buffer-aware bounds to multi-point progressive blocking in priority-preemptive NoCs"
2019	B. Nikolić et al. "Real-time analysis of priority-preemptive NoCs with arbitrary buffer sizes and router delays"

NoC Challenges – Backpressure Effect

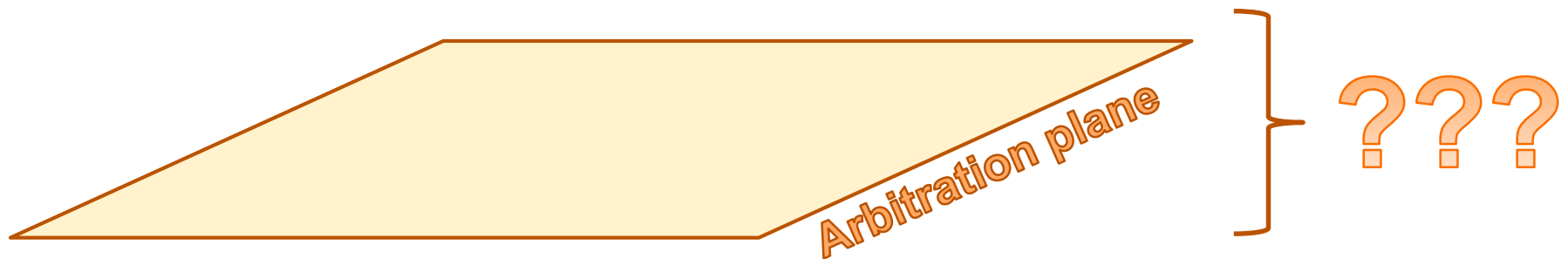


Packets often “heavy”, rethink packet injection

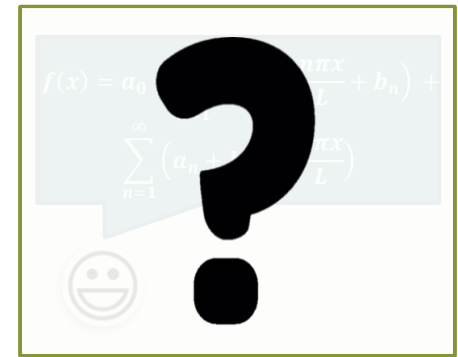
Separation of Concerns



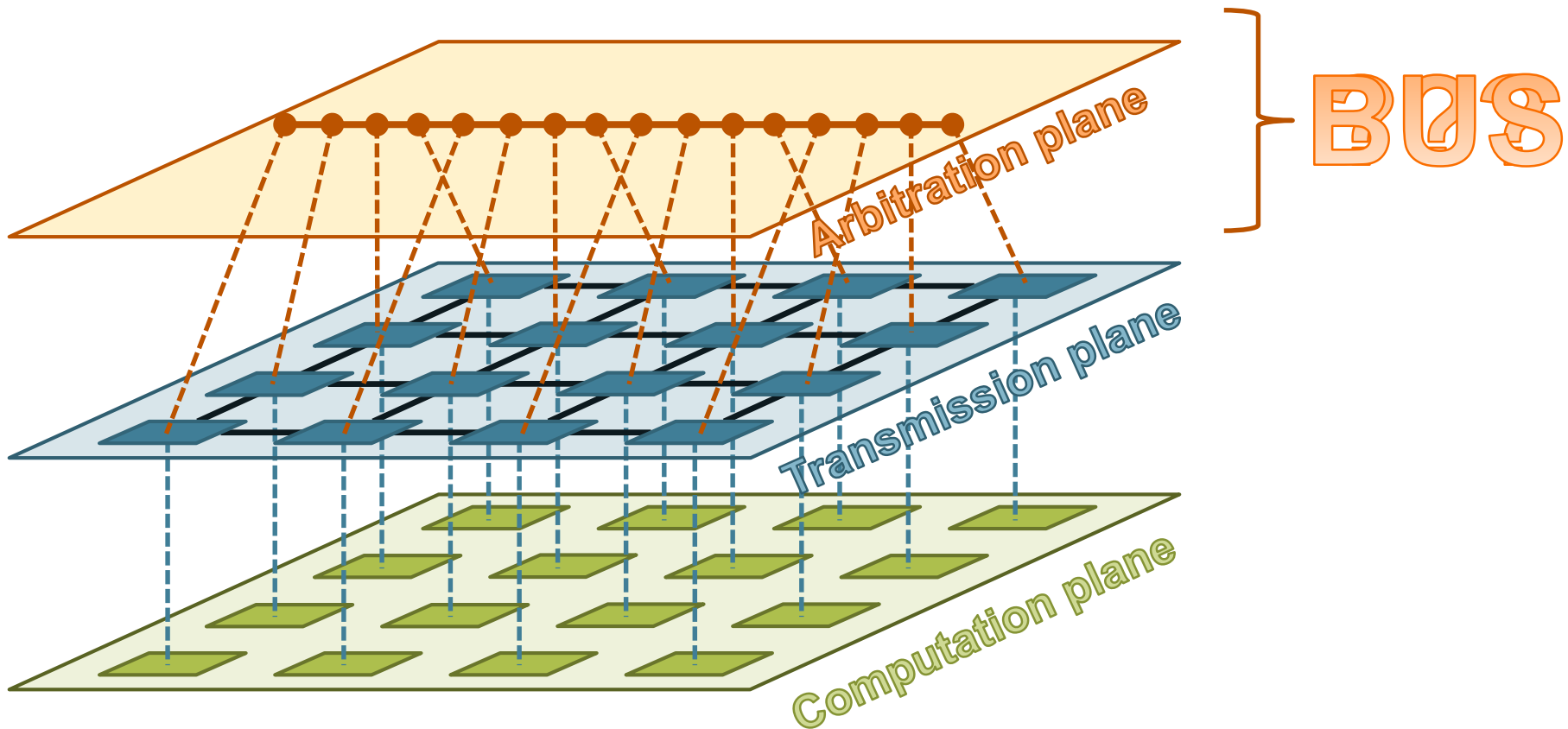
Dedicated Arbitration Medium



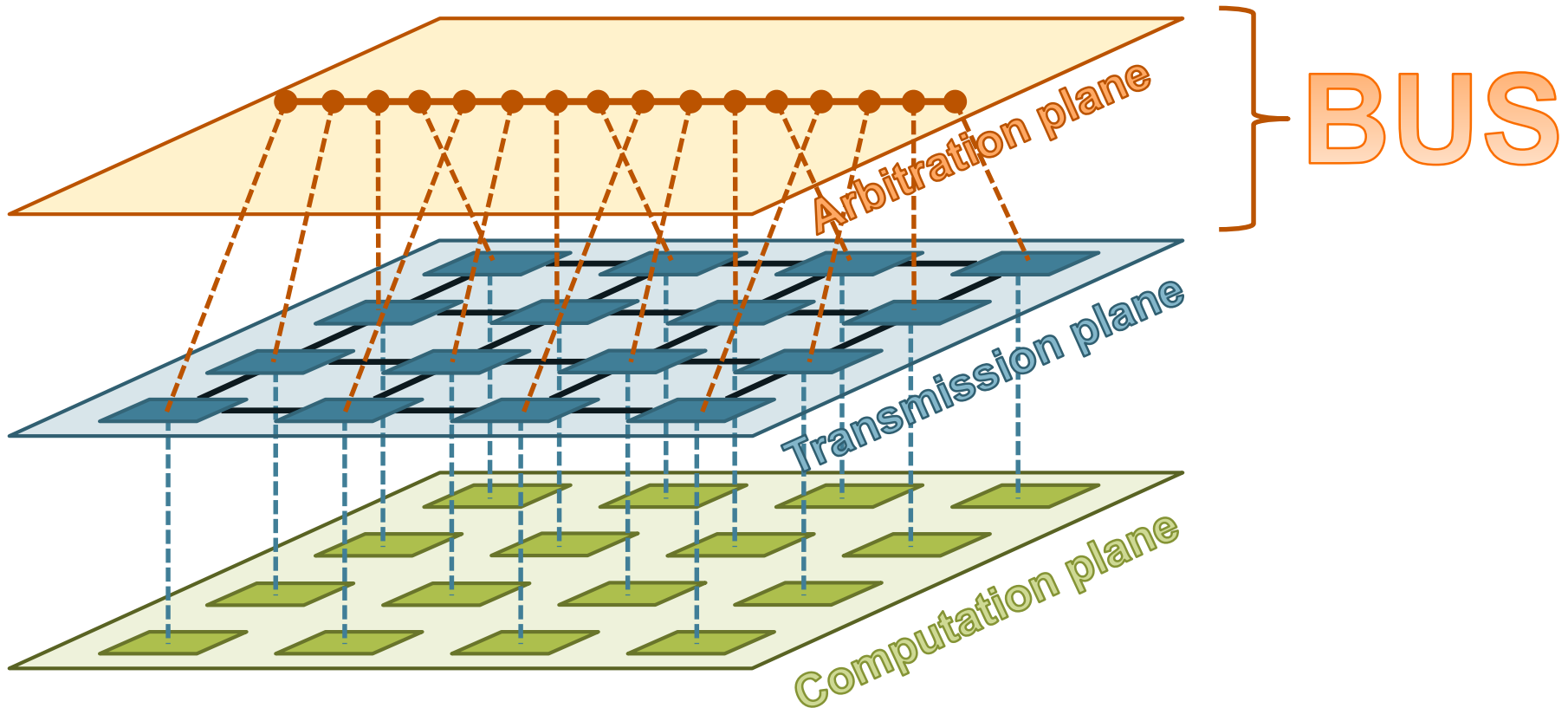
What medium & protocol CAN be good starting point?



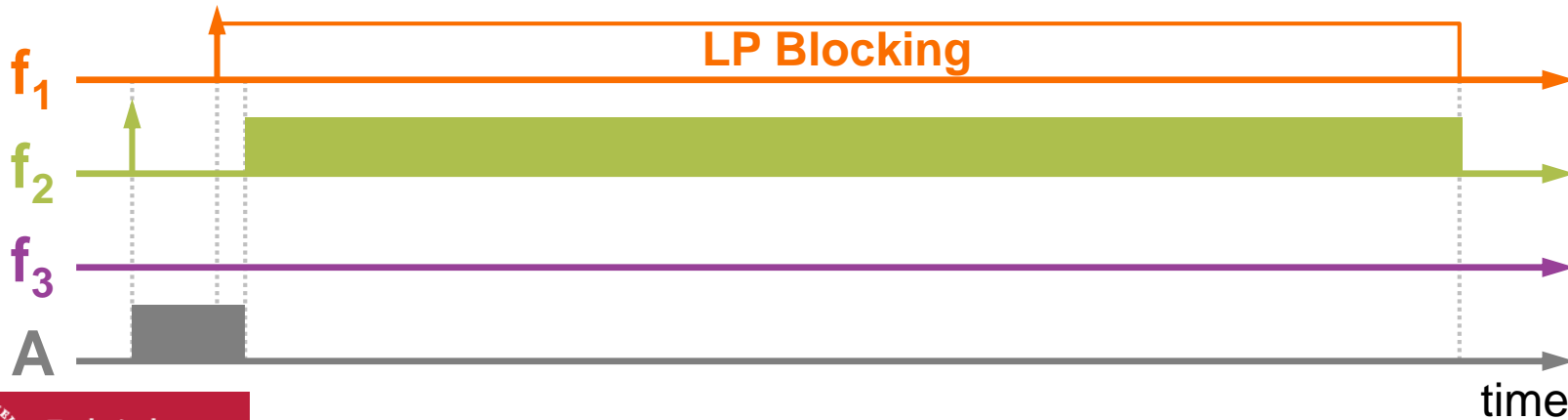
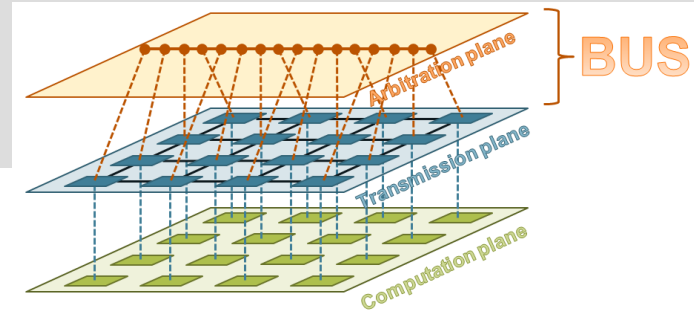
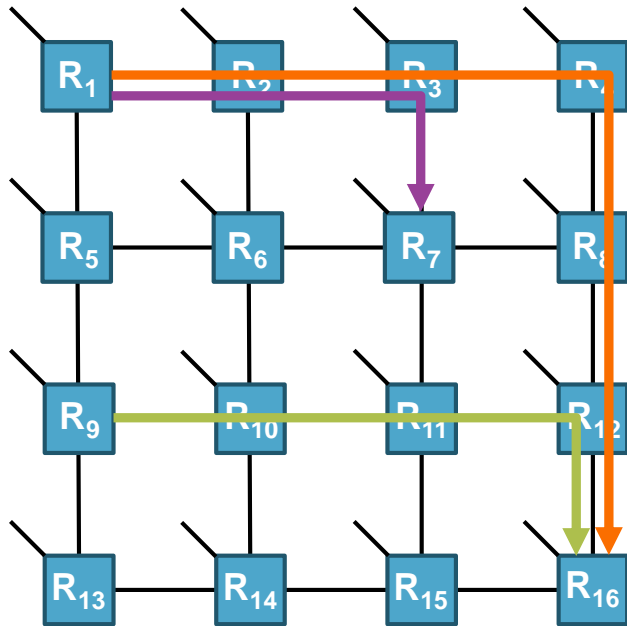
Separation of Concerns



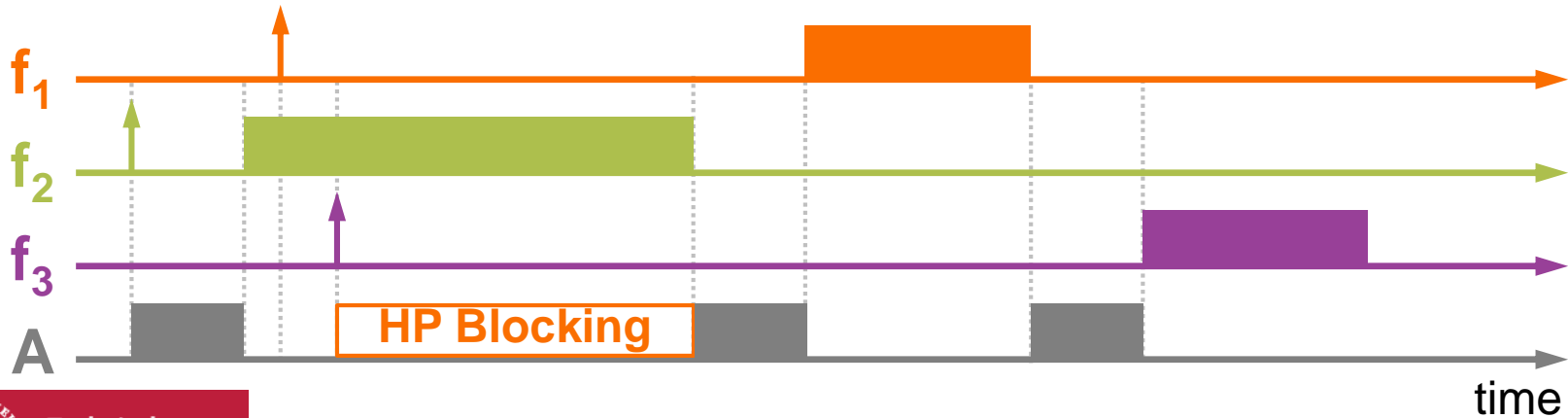
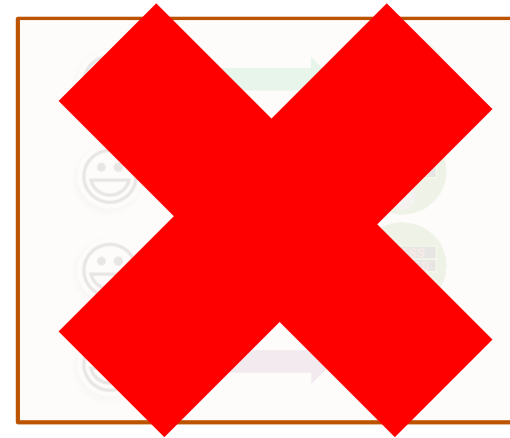
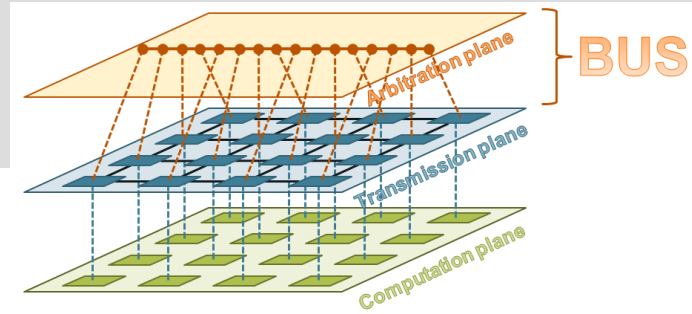
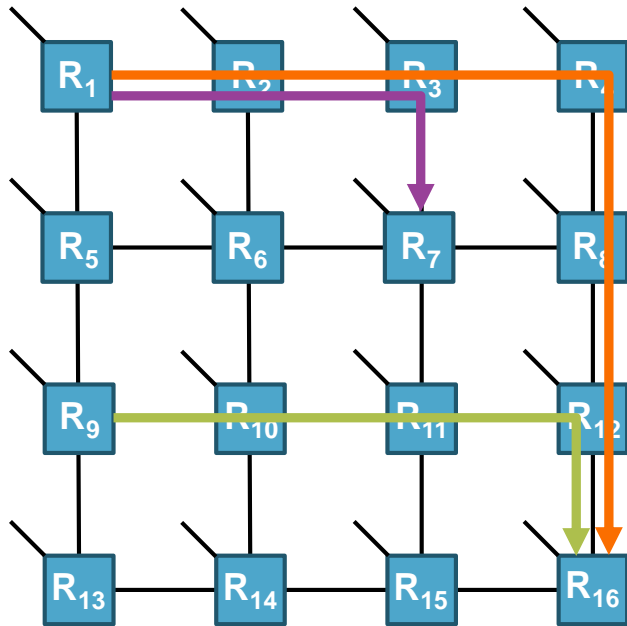
Separation of Concerns



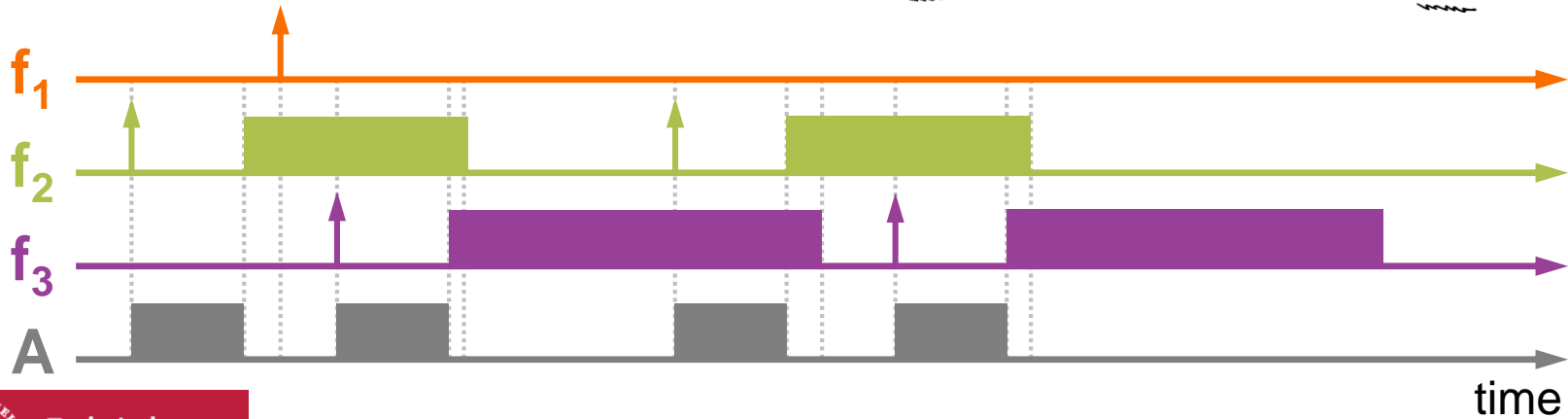
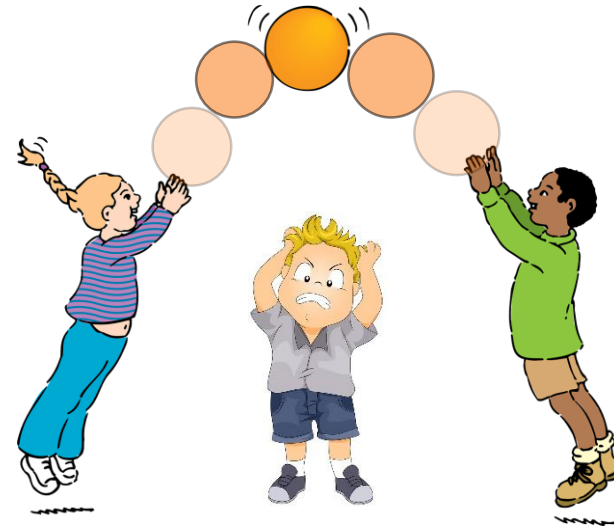
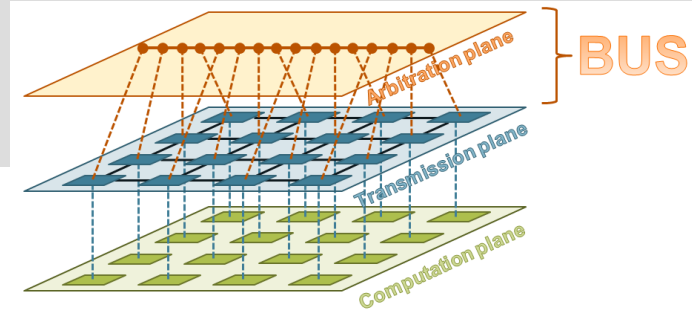
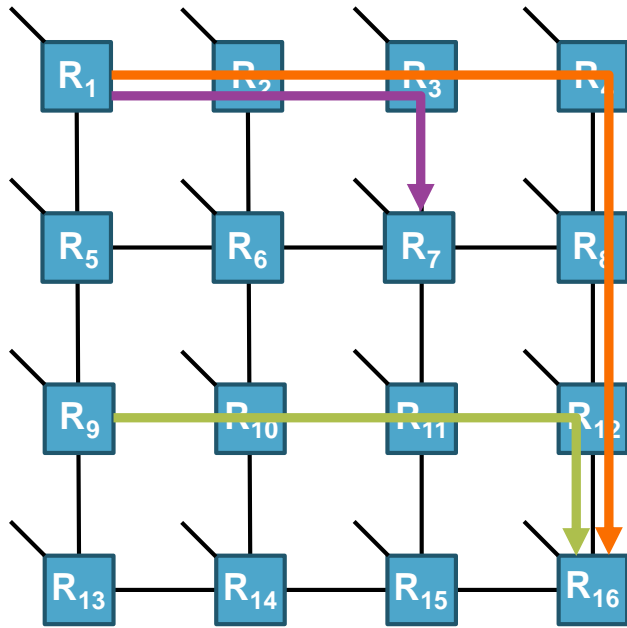
SBT-NoC (Development)



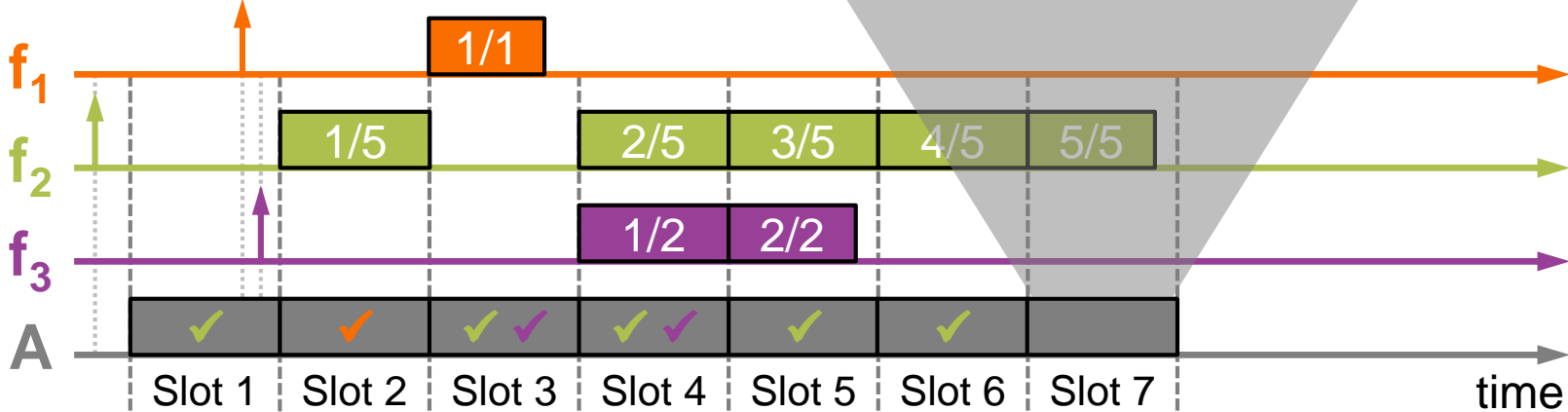
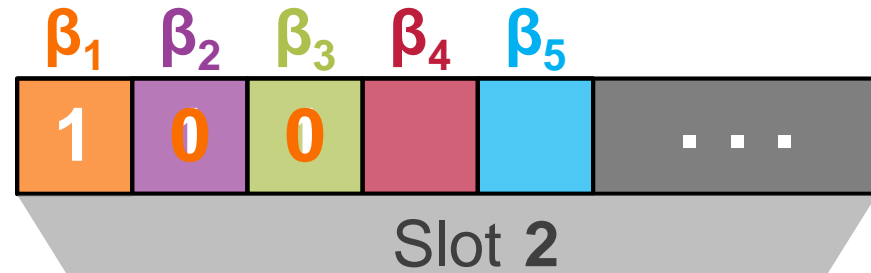
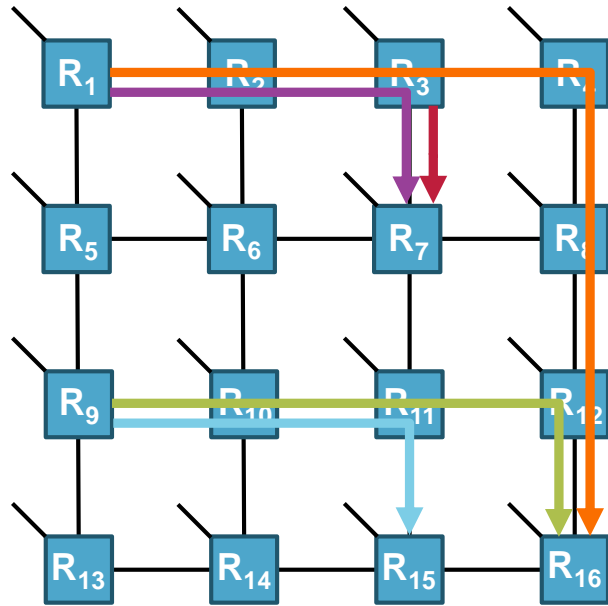
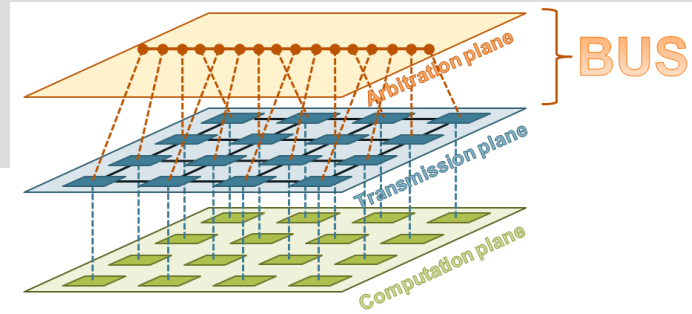
SBT-NoC (Development)



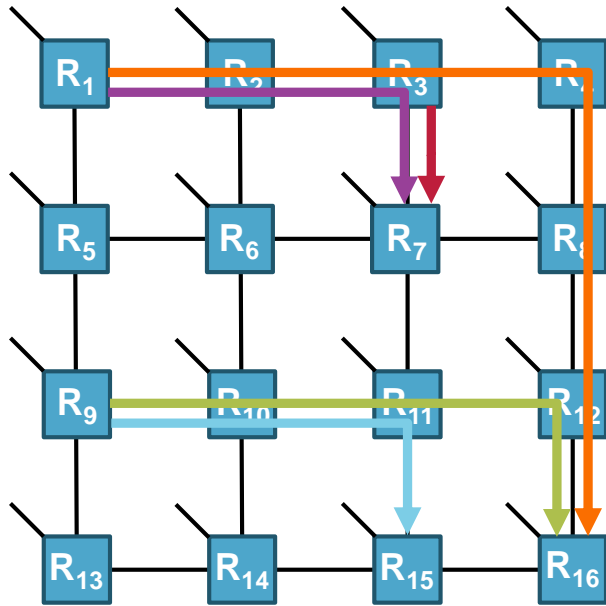
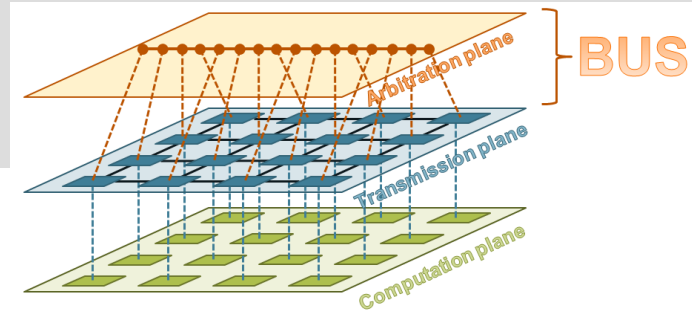
SBT-NoC (Development)



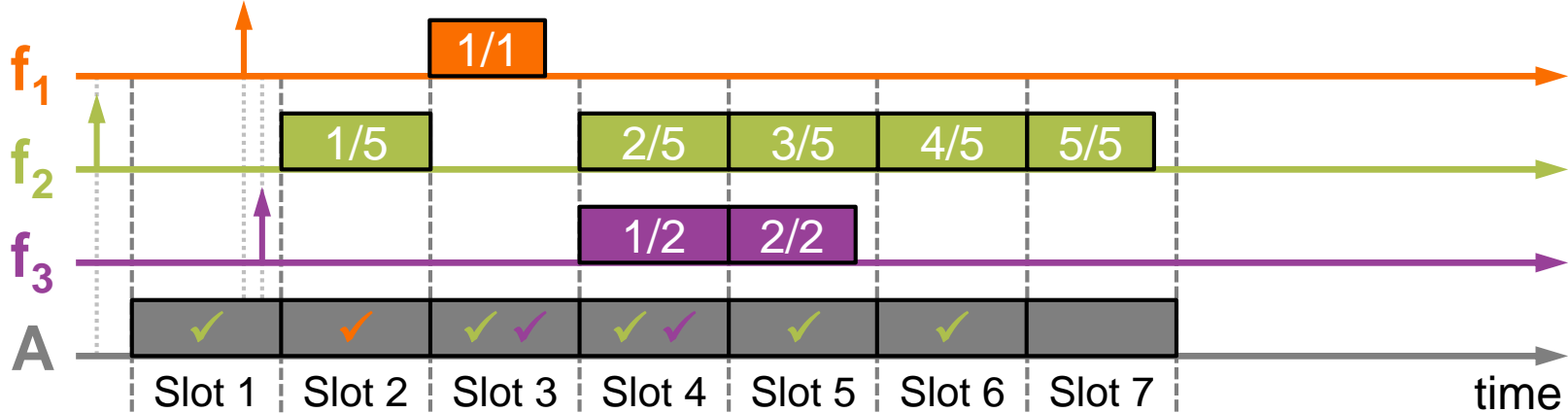
SBT-NoC (Getting there)



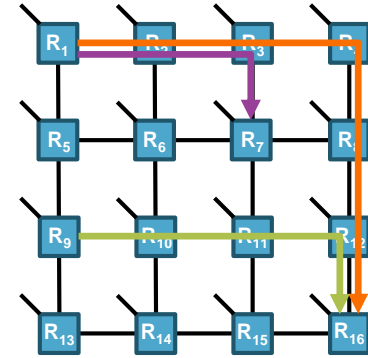
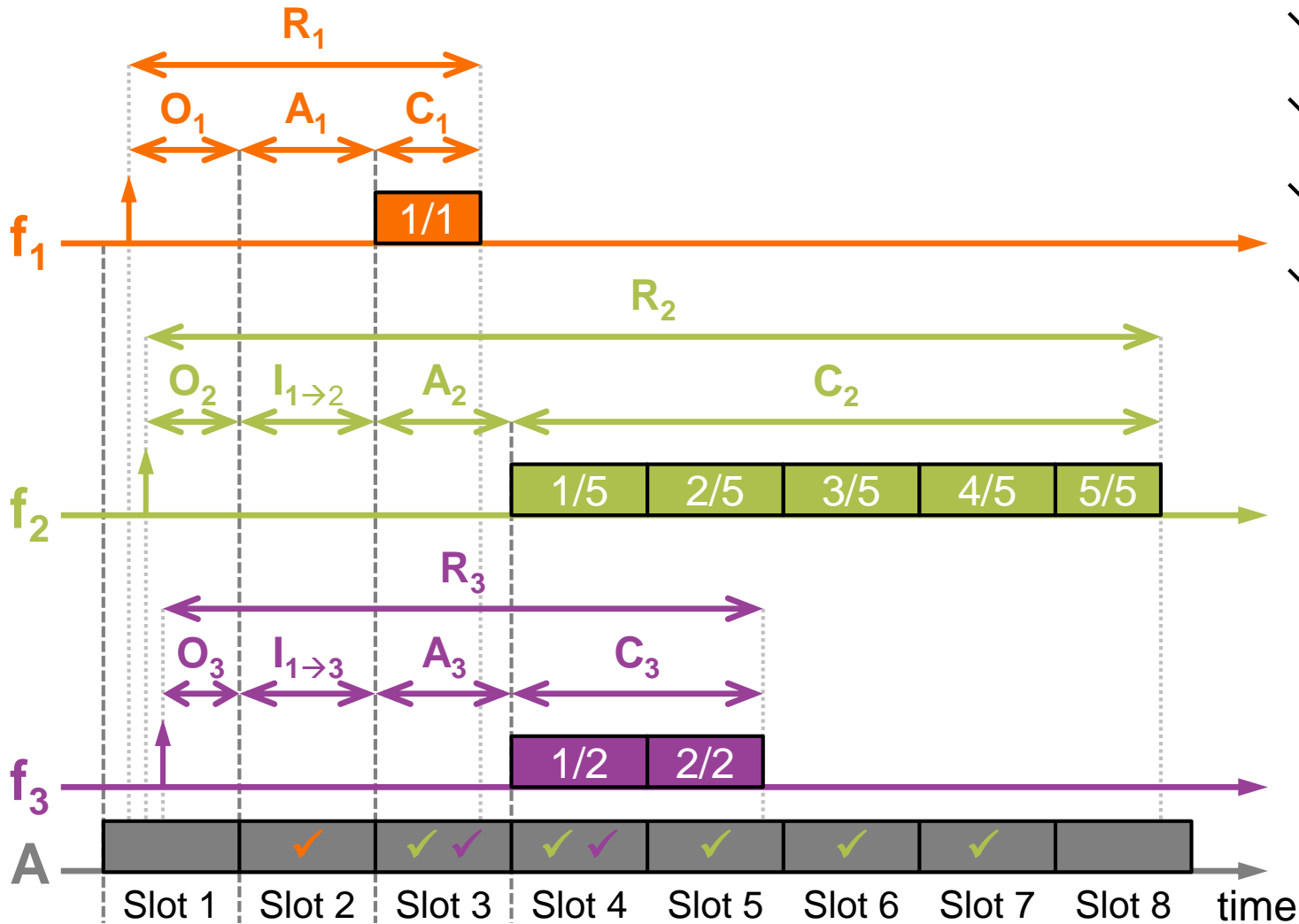
SBT-NoC (Getting there)



Slot 3



SBT-NoC (Basic)



Slot-Based Transmission Protocol for Real-Time NoCs - SBT-NoC
 Borislav Nikolić
 Institute of Computer and Network Engineering, TU Braunschweig, Germany
 borislav.nikolic@tu-bs.de

Robin Hofmann
 Institute of Computer and Network Engineering, TU Braunschweig, Germany
 robin.hofmann@tu-bs.de

Rolf Ernst
 Institute of Computer and Network Engineering, TU Braunschweig, Germany
 rolf.ernst@tu-bs.de

Abstract
 Network-on-Chip (NoC) architectures are one of the most challenging-to-realize components of multiprocessor platforms. This is primarily due to the following two reasons: (i) NoCs require extensive shared resources (i.e., routers, links, and I/O) for network traffic, often consuming a large fraction of the device resources. Consequently, complex reservation schemes using static and dynamic time slots are required to guarantee performance guarantees.

In this work, we propose a slot-based reservation protocol for NoCs (called SBT-NoC), and an accompanying router module for deriving reservation slots. The reservation of SBT-NoC in a reservation-based network, without a pre-set reservation in a dedicated network module. The main advantage of SBT-NoC is that, while not requiring an explicit hardware support (i.e., virtual channels, or flow-based admission), it enables NoCs to handle a real-time and potentially bursty traffic of high priority, non-critical flows, which is also not needed for the utilization of NoCs, as demonstrated by results in the real-time domain. The experimental evaluation, including both synthetic workloads and a set of an automotive driving-related applications, reveals that SBT-NoC offers a plethora of reservation opportunities, which makes it applicable to a wide range of diverse traffic workloads.

2017 ACM Subject Classification: Computer systems organization → Real-time systems; Computer systems organization → Embedded systems

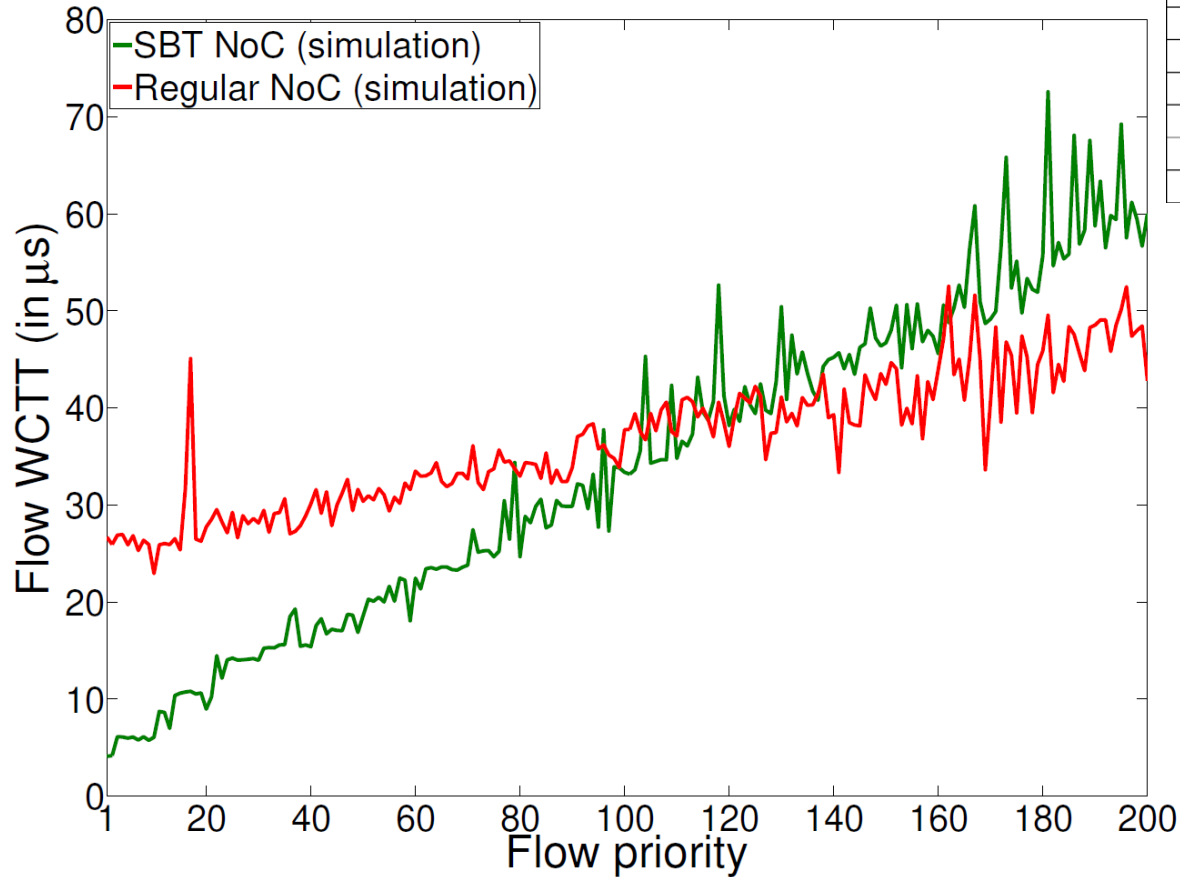
Keywords and phrases: Real-Time Systems, Embedded Systems, Network-on-Chip, Protocols

Digital Object Identifier: 10.1007/978-3-319-59292-9_16

1 Introduction
 Even though multiprocessors are now ubiquitous in almost all computing areas, they are still often considered as a new frontier technology in the real-time domain. Traditionally, the real-time analysis of multiprocessors, the requirement to use a single type of shared resources – processing elements (cores). However, due to the ever-increasing demand for the multi-processor area, multiprocessors for other shared resources, such as an interconnect network, become more important. This implies that, in order to perform a real-time analysis of real-time applications deployed on multiprocessors, it is no longer sufficient to only take into account the requirements for shared resources, but also interconnect and memory traffic need to be considered as well. Therefore, the real-time analysis of network interconnects becomes a crucial prerequisite for the development of multiprocessors in the real-time domain.

© Springer Nature, Berlin, Heidelberg, and Tokyo, 2017.
 This is a preview of the article published online in Springer Nature. The final version of the article will be published in the journal Embedded Systems, Springer Publishing Company.

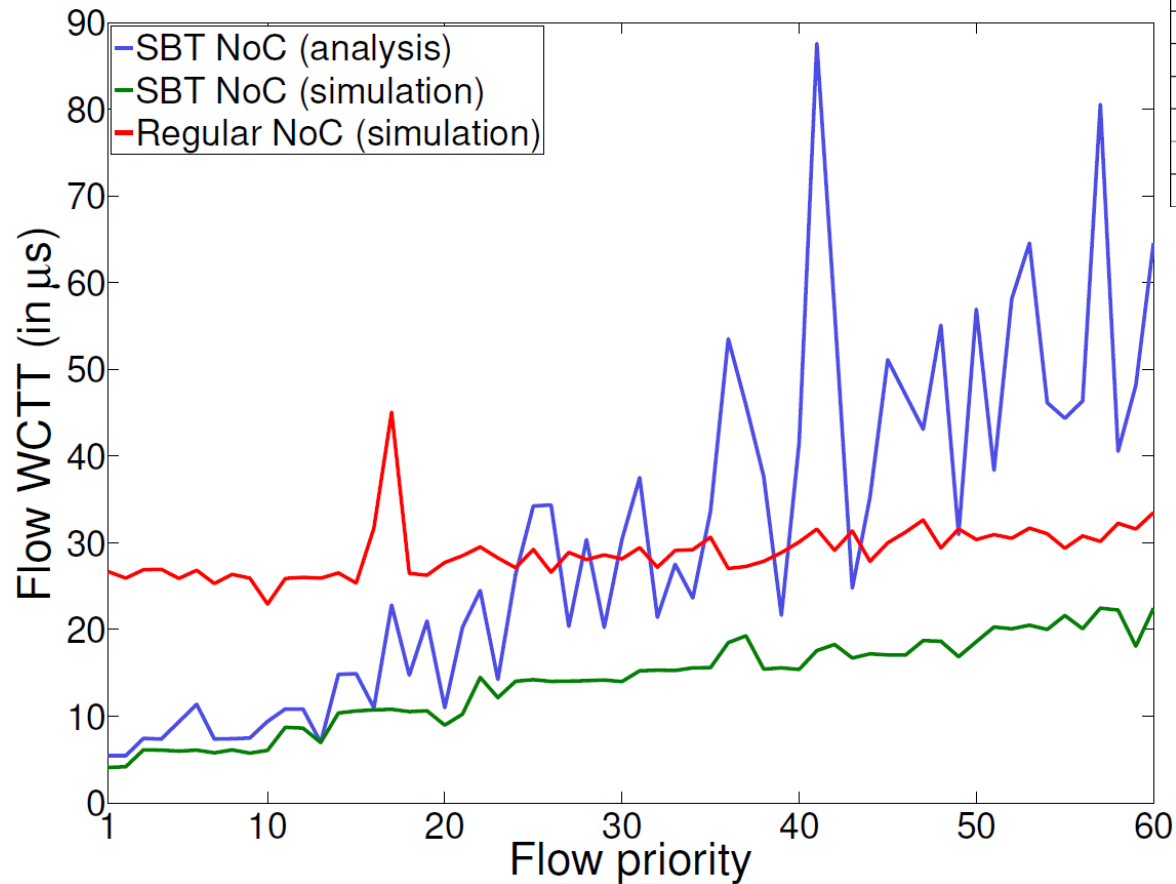
SBT-NoC (Basic)



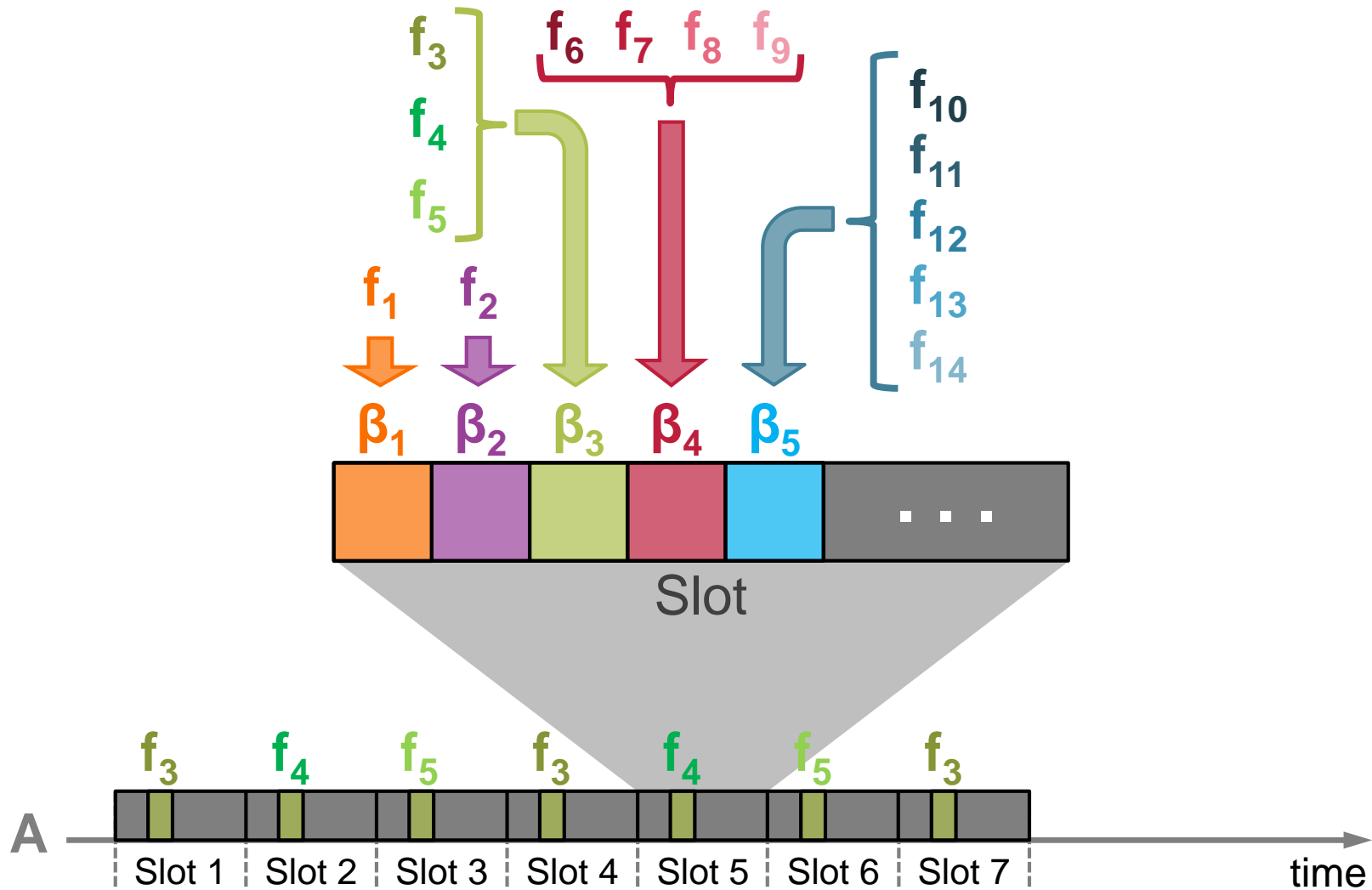
NoC topology	4×4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow source core/router ($\mu_i^{src} / \rho_i^{src}$)	Random
Flow destination core/router ($\mu_i^{dst} / \rho_i^{dst}$)	Random
Flow size (σ_i)	[500B - 10kB]
Flow deadline (D_i) = flow period (T_i)	[0.5ms - 5ms]*
Flow priority assignment policy	Rate monotonic

SBT-NoC (Basic)

NoC topology	4 × 4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow source core/router ($\mu_i^{src} / \rho_i^{src}$)	Random
Flow destination core/router ($\mu_i^{dst} / \rho_i^{dst}$)	Random
Flow size (σ_i)	[500B - 10kB]
Flow deadline (D_i) = flow period (T_i)	[0.5ms - 5ms]*
Flow priority assignment policy	Rate monotonic

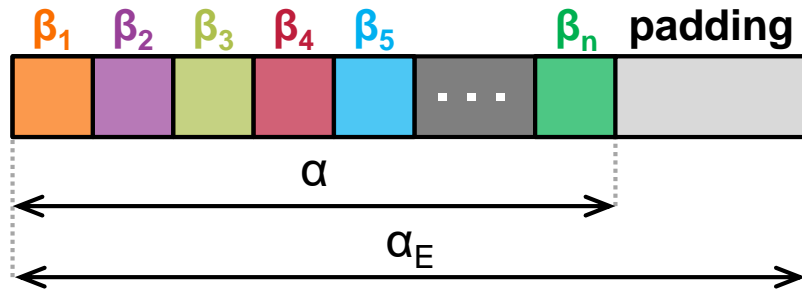


SBT-NoC (Advanced w/ Slot Reduction)

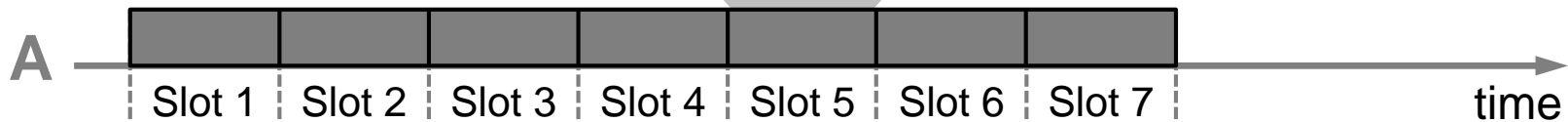
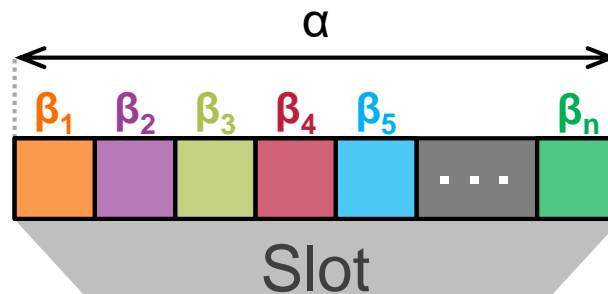
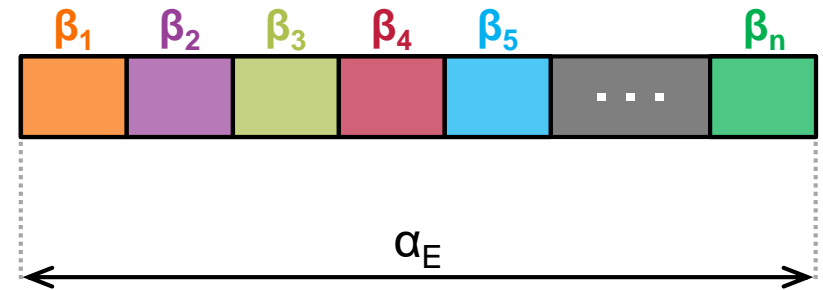


SBT-NoC (Advanced w/ Slot Extension)

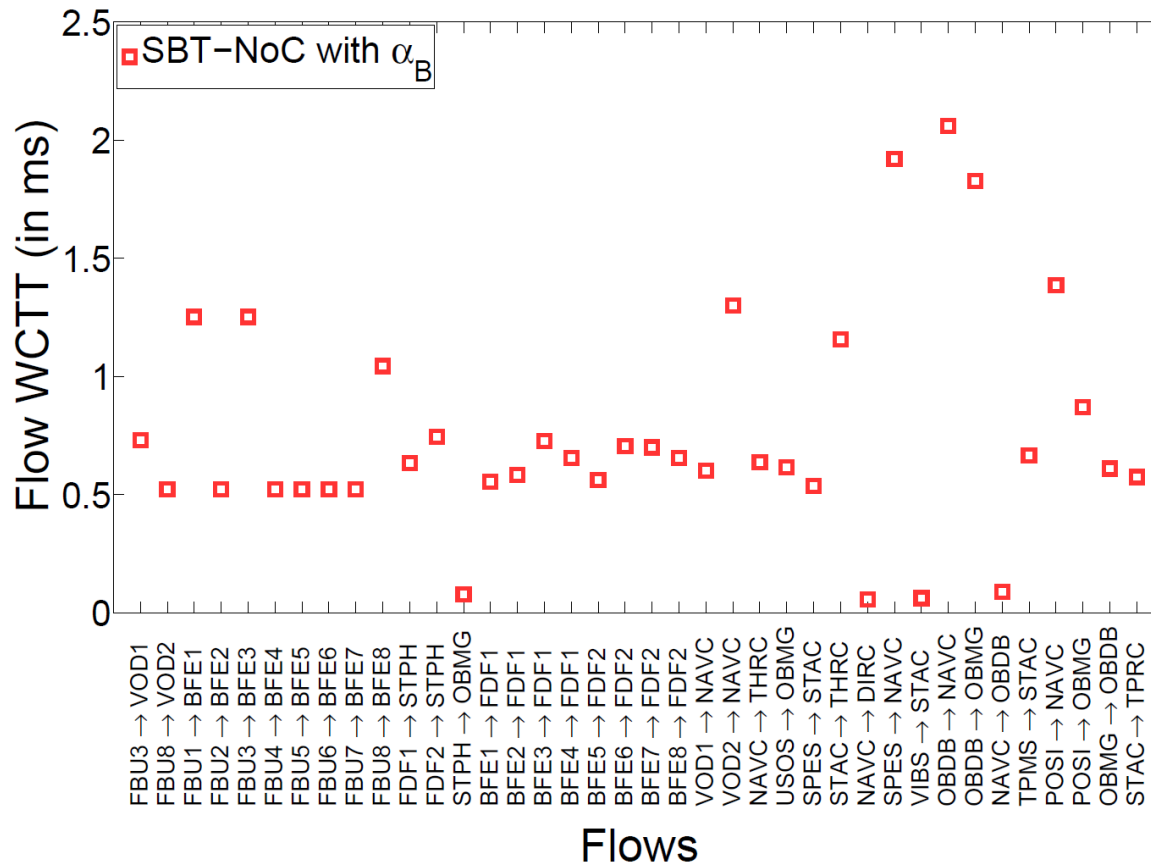
Option 1



Option 2



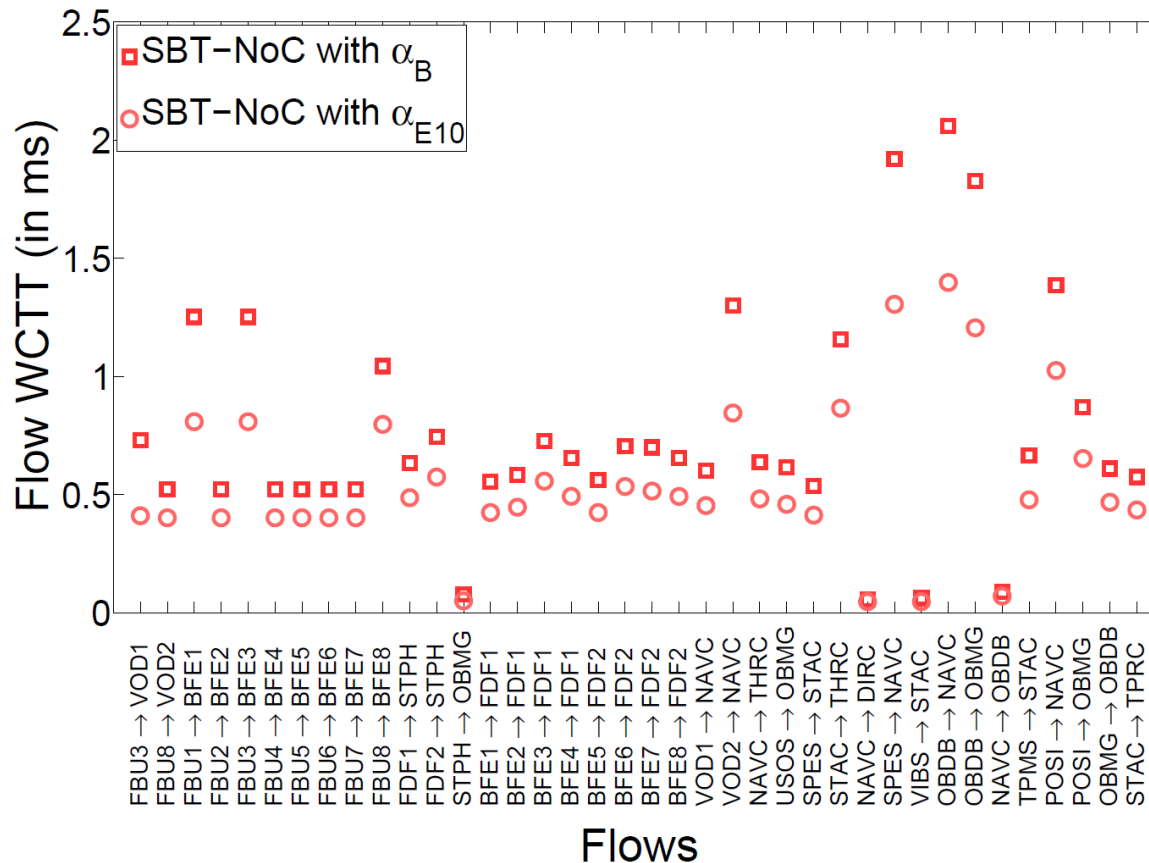
SBT-NoC (Advanced w/ Slot Extension)



NoC topology	4 × 4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow size (σ_i)	[2kB - 150kB]
Flow deadline (D_i) = flow period (T_i)	[0.04s - 1s]
Flow priority assignment policy	Rate monotonic

Use-case of autonomous driving vehicle application (from Shi et al. 2010).

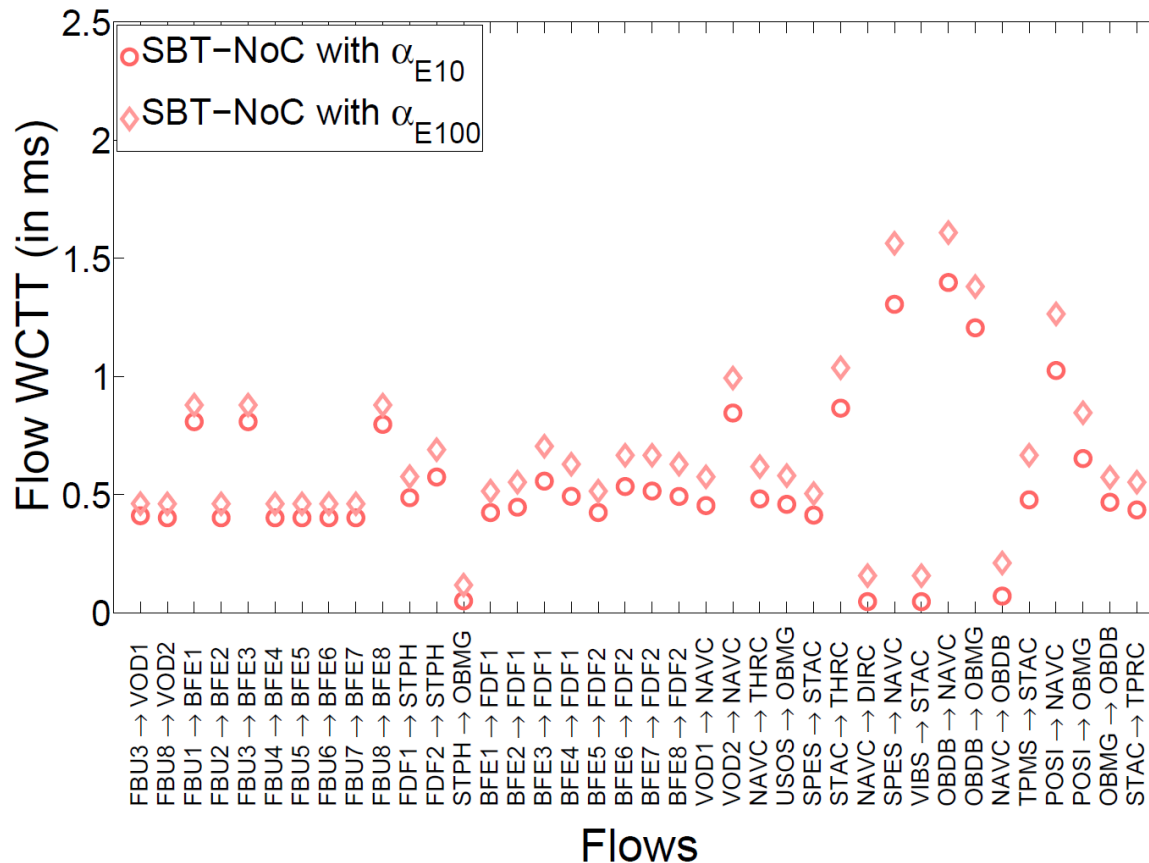
SBT-NoC (Advanced w/ Slot Extension)



NoC topology	4 × 4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow size (σ_i)	[2kB - 150kB]
Flow deadline (D_i) = flow period (T_i)	[0.04s - 1s]
Flow priority assignment policy	Rate monotonic

Use-case of autonomous driving vehicle application (from Shi et al. 2010).

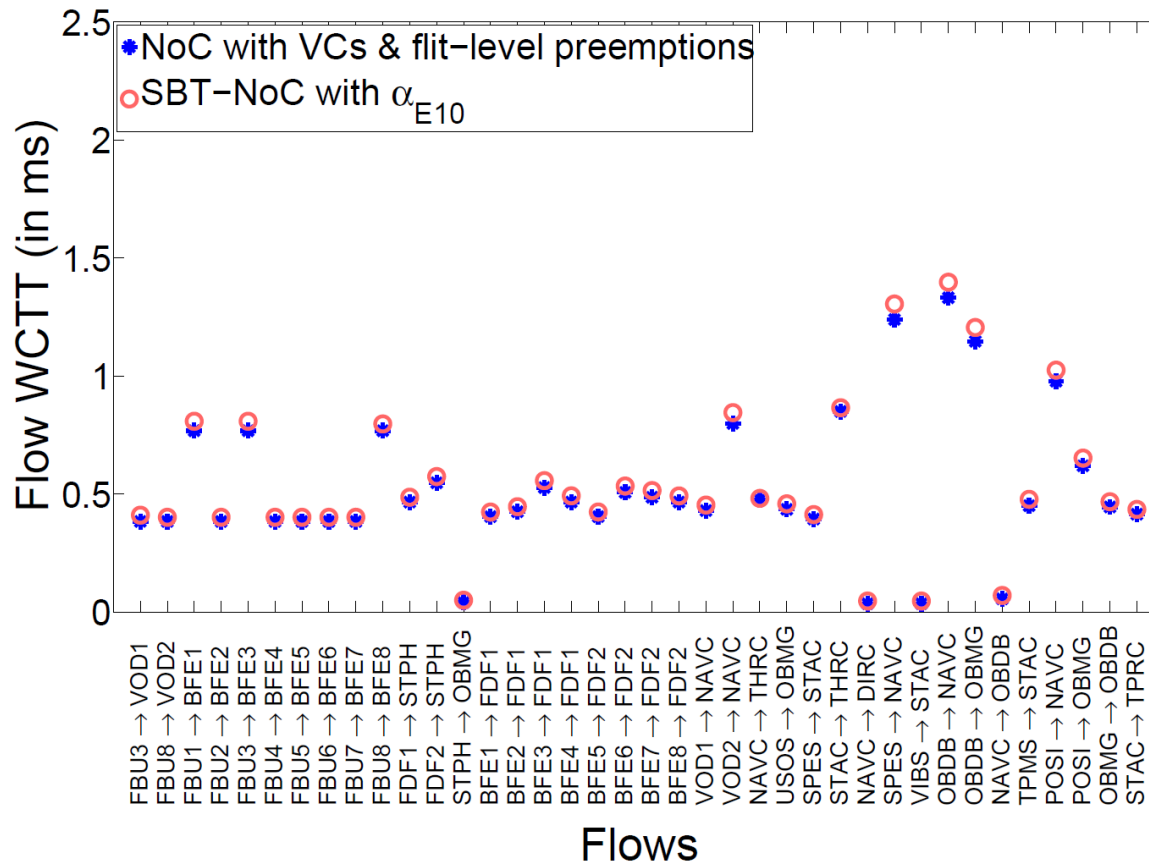
SBT-NoC (Advanced w/ Slot Extension)



NoC topology	4 × 4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow size (σ_i)	[2kB - 150kB]
Flow deadline (D_i) = flow period (T_i)	[0.04s - 1s]
Flow priority assignment policy	Rate monotonic

Use-case of autonomous driving vehicle application (from Shi et al. 2010).

SBT-NoC (Advanced w/ Slot Extension)

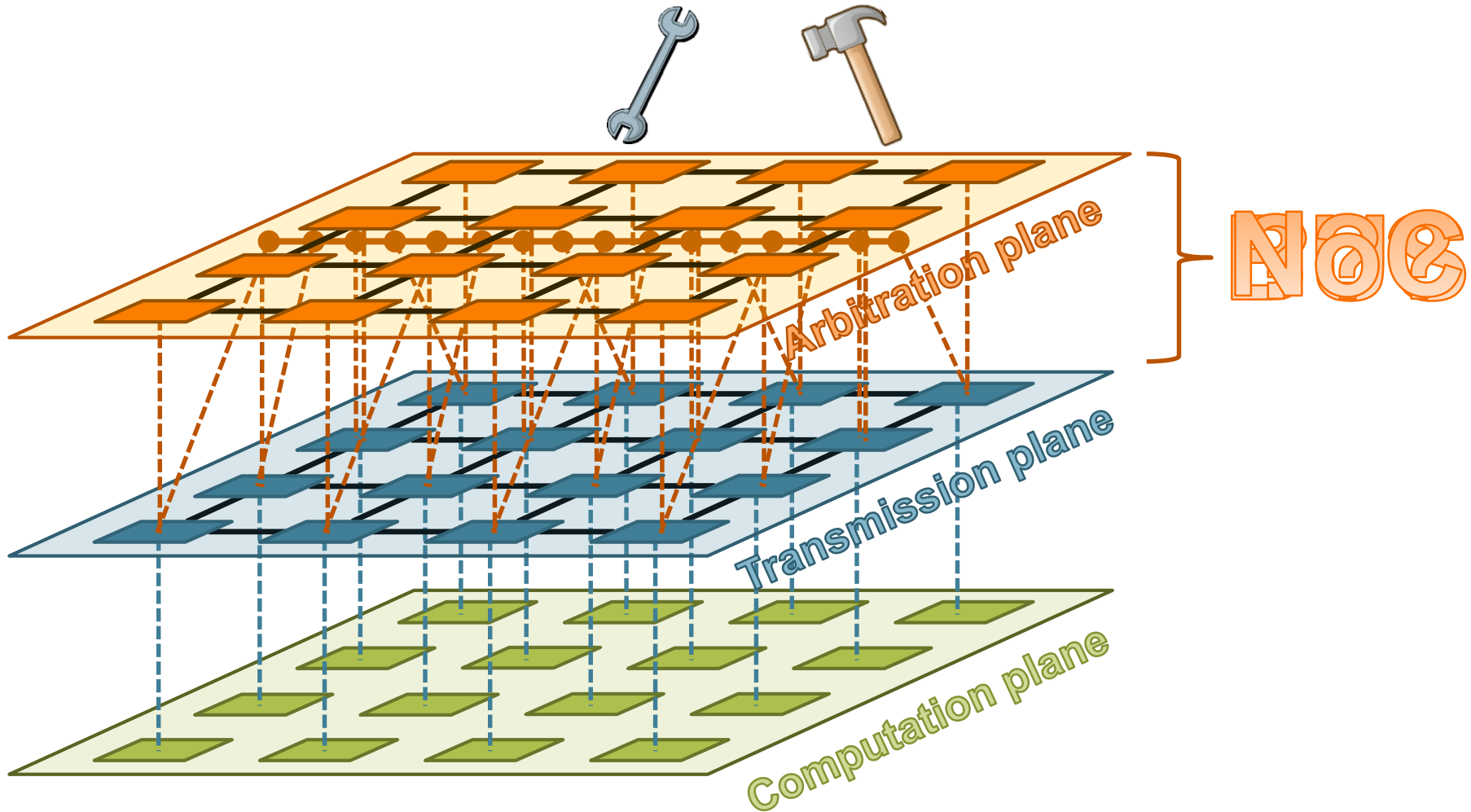


NoC topology	4 × 4 (2-D mesh)
Routing mechanism	X-Y
Router frequency (ψ)	100MHz
Router latency (d_R) + link latency (d_L)	3 + 1 cycles
Bus writing/reading latency (d_B)	1 cycle
Link width = flit size (σ_F)	4B
Flow size (σ_i)	[2kB - 150kB]
Flow deadline (D_i) = flow period (T_i)	[0.04s - 1s]
Flow priority assignment policy	Rate monotonic

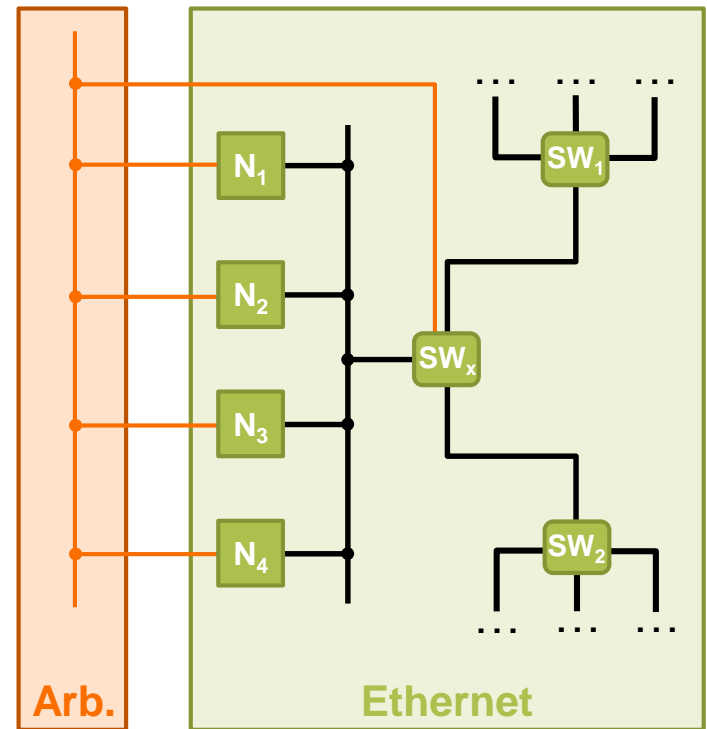
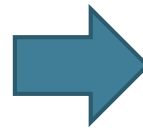
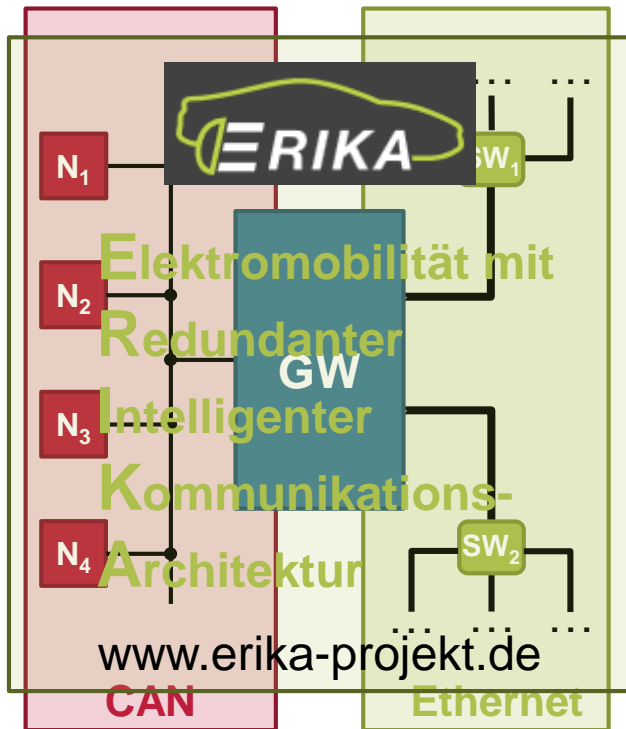
	SBT-NoC	FLP NoC
VCs	None	Per-stream
Arb	None	Flit-level
Buff	None	Credit-based

Use-case of autonomous driving vehicle application (from Shi et al. 2010).

SBT-NoC Future Work & Application



SBT-NoC Application & Future work



Thank you for your attention!

