Technische Universität Braunschweig

Bringing Dynamic Control to Real-time NoCs

Rolf Ernst
TU Braunschweig
Nara, July 12, 2016

Outline

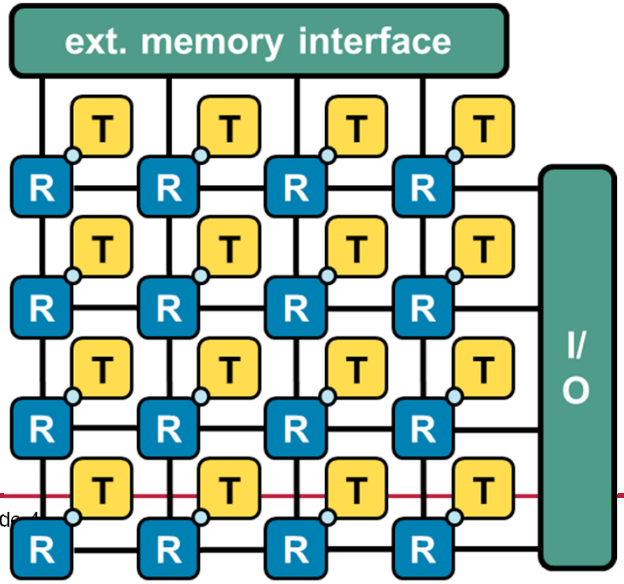
- **Motivation**
- NoCs for hard real-time and robustness
- Dynamic resource management
- Improving memory control and error handling
- Conclusions

Motivation

- **many-core systems are reaching critical embedded systems**
  - sensor fusion and recognition in highly automated driving
  - avionics, space
- **limited power and cost budget**
  - compact solutions
  - higher systems integration
- **mixed criticality**

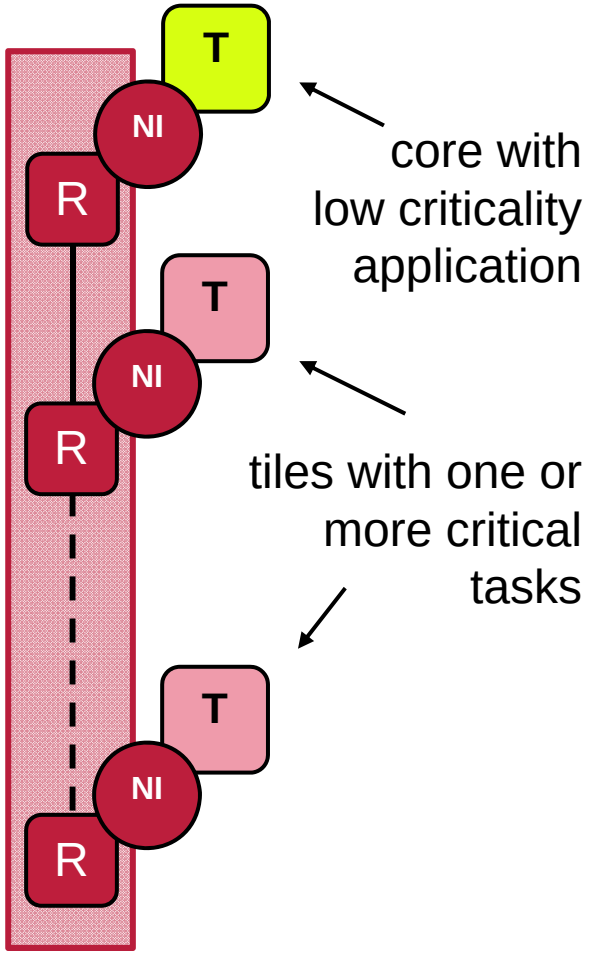
Mixed criticality challenge - Independence

- safety standards require
  - isolation of subsystems with different criticality levels (IEC 61508: sufficient independence)
  - predictable timing where timing is relevant (almost every system)
  - robustness against errors
- already challenging in current multicore implementations
  - how to meet these challenges in many-cores?
  - main difference: Communication via Network-on-Chip (NoC)



Robustness and isolation in NoCs

- the NoC must be designed according to the network traffic with highest criticality
- network interface must separate critical network from non critical tiles
- non-critical tasks cannot be trusted
  - WCET
  - activation frequency
  - communication volume
  - addressing
- approach
  - NoC with QoS guarantees for critical traffic
  - Network Interface (NI) with access control
- ***last talk at MpSoC (2011)***
  - ***used in several projects (ARAMIS, RECOMP, ...)***



Outline

- Motivation
- **NoCs for hard real-time and robustness**
- Dynamic resource management
- Improving memory control and error handling
- Conclusions

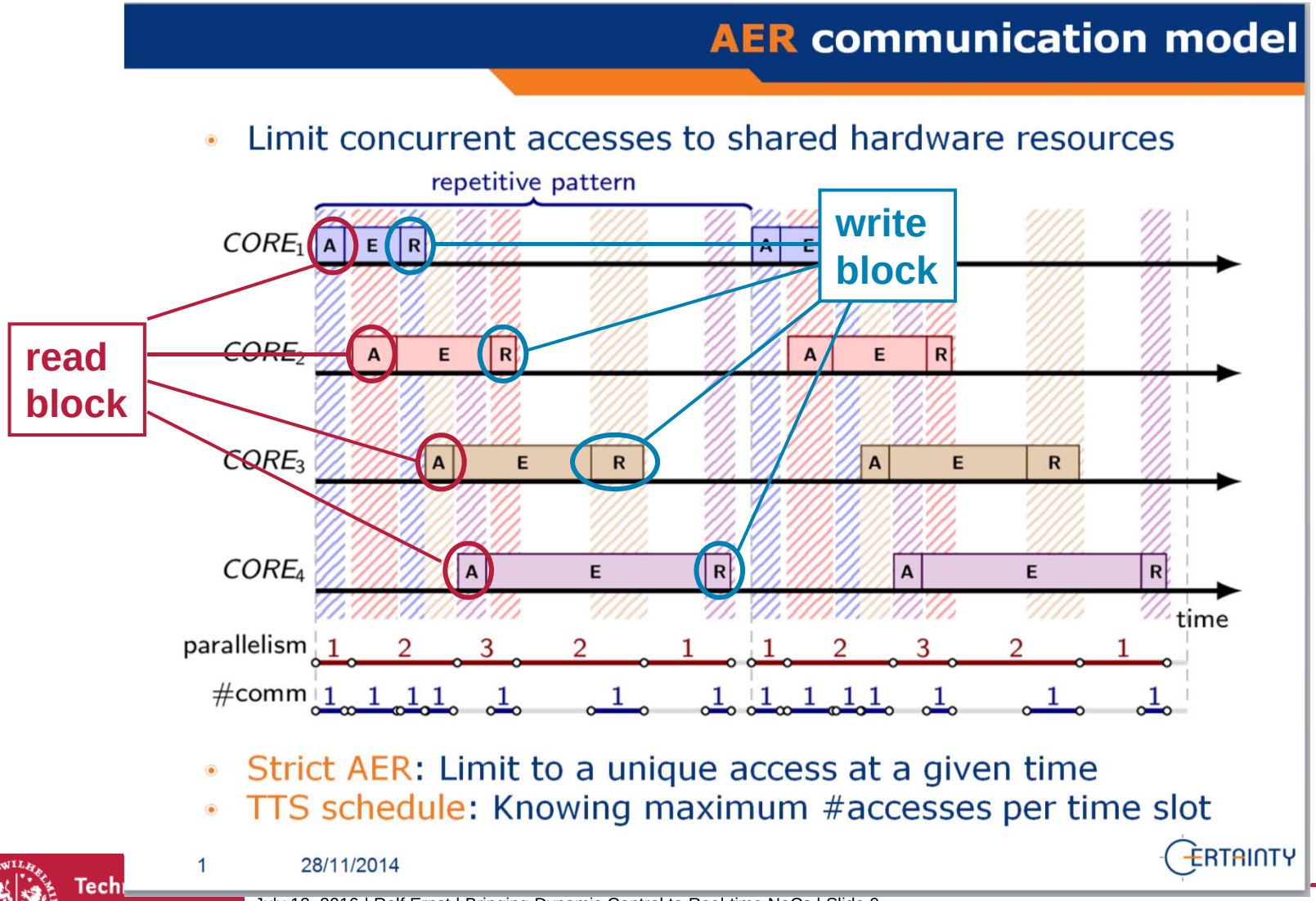
This talk: NoCs for hard real-time and robustness

- **main factors**
  - router and network control
  - robustness against errors
- **main challenge**
  - high performance requirements + resilience + safety (+ security)
    – cp. autonomous driving!
- **related work**
  - limited interference using distributed resource assignment (PhaseNOC) or distributed flow control (back suction)
  - predictability using time triggered communication (e.g. Aetheral, CompSOC)
  - predictability using static block transfers – separating communication from computation
- ***selected approach: block transfer***

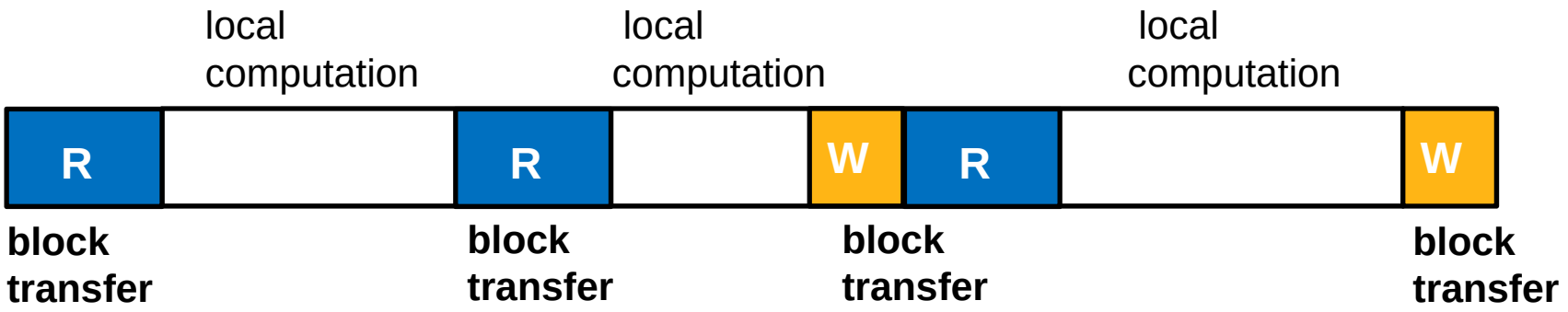
„Classical“ model in safety critical design

- read in the beginning - write in the end
  - examples: task model in automotive software, superblock model (containers)
- in multi-cores
  - clustering of memory accesses
  - deterministic access sequence (DMA)

Static repetition in multicore – CERTAINTY project

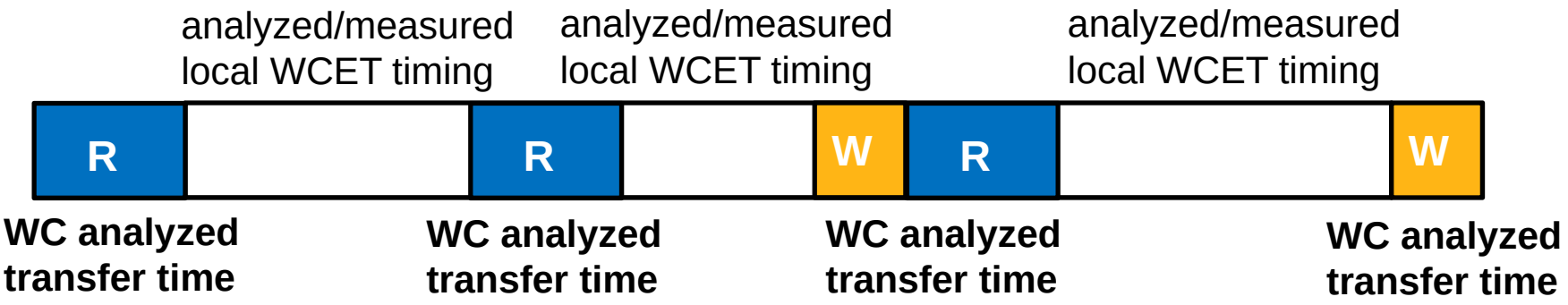


Flexible NoC block transfers - Principle



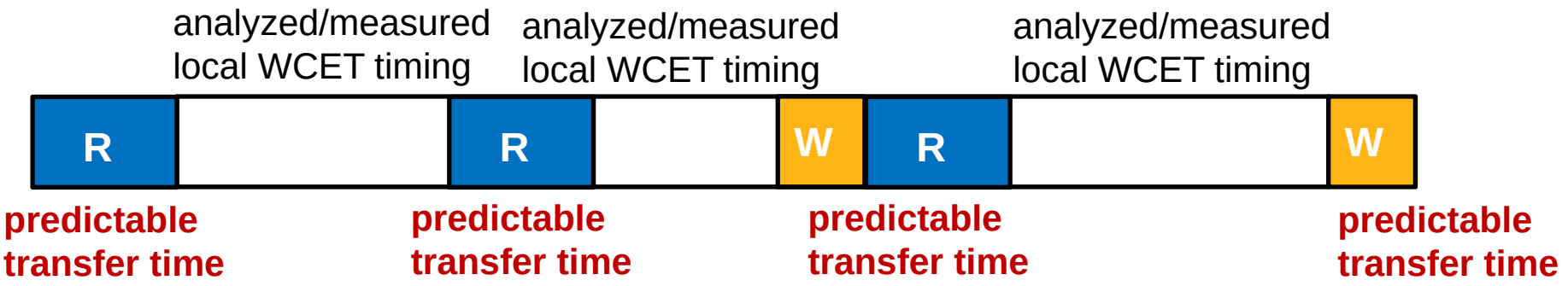
- classical model limited by local memory size and communication cost
  - flexible block transfers as simple extension
- dynamic communication resource reservation
  - Improved utilization of resources - in particular memories
  - combine with memory controller for efficient block transfer
- **challenge: optimized worst-case (WC) timing for safety critical systems**
  - allow other patterns for best effort traffic – mixed critical systems

Block transfer – Worst case (WC) timing



- previous NoC solution
  - distributed scheduling and arbitration in routers
  - result: **wide timing bounds for block (and other) transfers**

VC reservation – Predictable timing



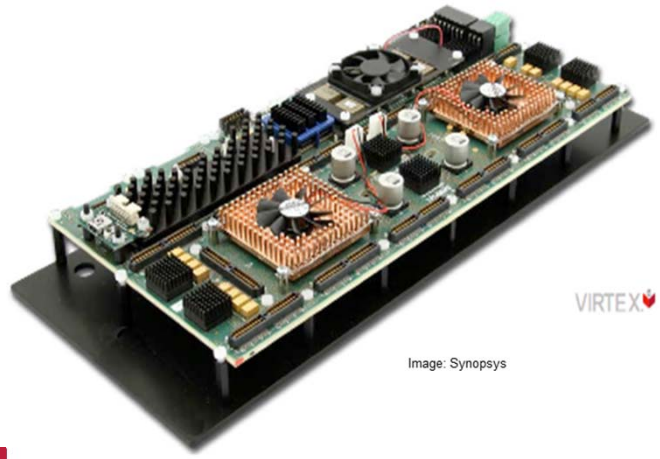
- resource reservation timing
  - **tightly bounded WC memory access and transfer time**
  - result of deterministic sequence + resource reservation
  - computing time measured or analyzed (e.g. WCET analysis)

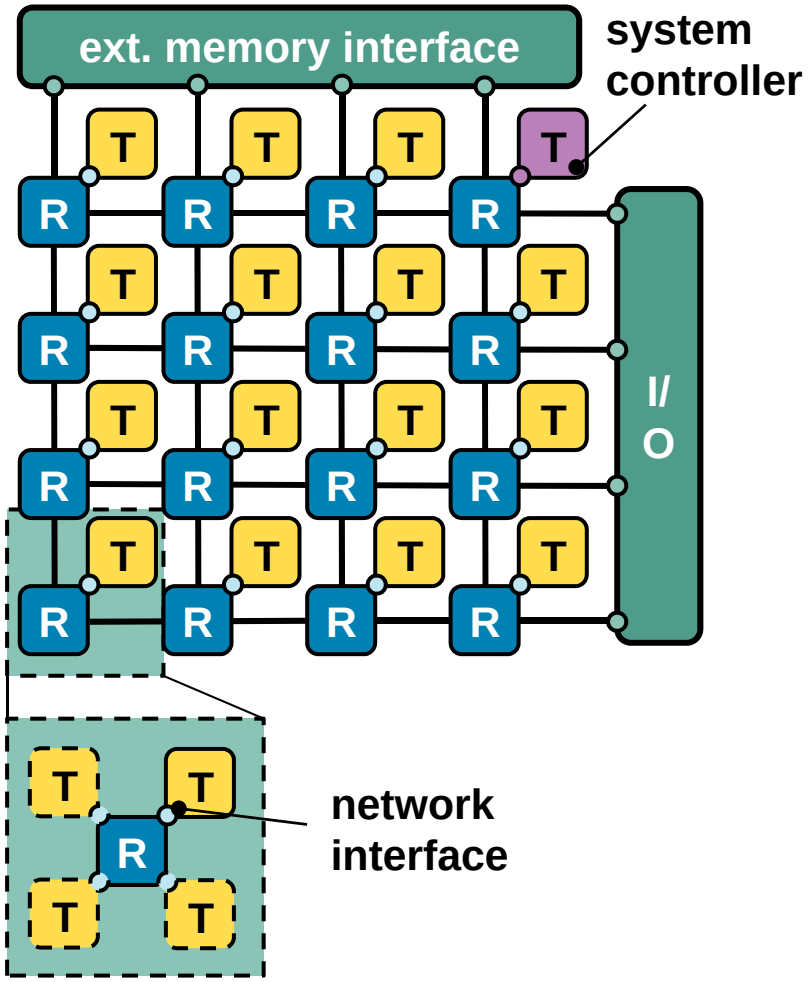
Outline

- Motivation
- NoCs for hard real-time and robustness
- **Dynamic resource management**
- Improving memory control and error handling
- Conclusions

Integrated Dependable Architecture for Many Cores (IDAMC) - Research vehicle

- **configurable NoC**
  - up to 4 links per router
  - 4-64 nodes with up to 4 AMBA based tiles
    - currently LEON3
  - NoC level virtualization
  - includes DMA controlled data transport
- implemented on FPGAs
  - Synopsys HAPS-62



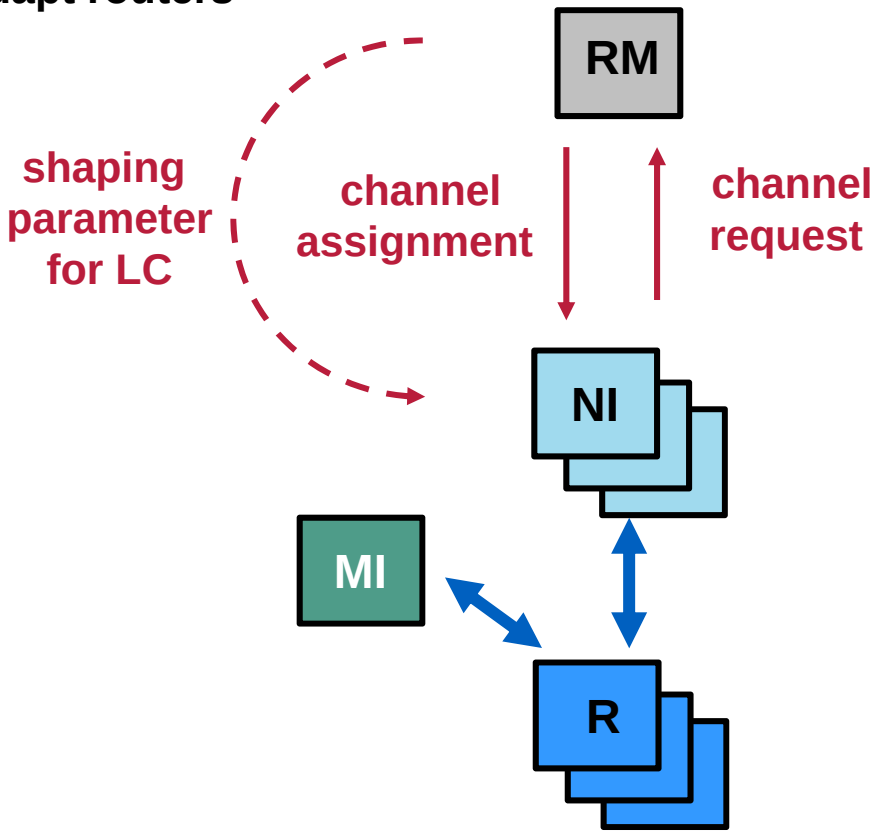


IDAMC NoC overview

- **wormhole routing**
  - flit size 128b
- **virtual channels (VC)**
  - 2-8 channels
  - classes: bandwidth critical (BC), best effort (BE)
- **router (R)**
  - 2 stage arbitration (input buffer + output iSLIP)
  - distributed control (back suction streaming option; not this talk)
- **network interface (NI)**
  - address translation w. protection
  - signaling for different DMA communication paradigms
  - tile level traffic shaping

Extension for dynamic channel reservation

- include Resource Manager (RM)
  - controls VC assignment and tile level traffic shaping
  - adapt routers



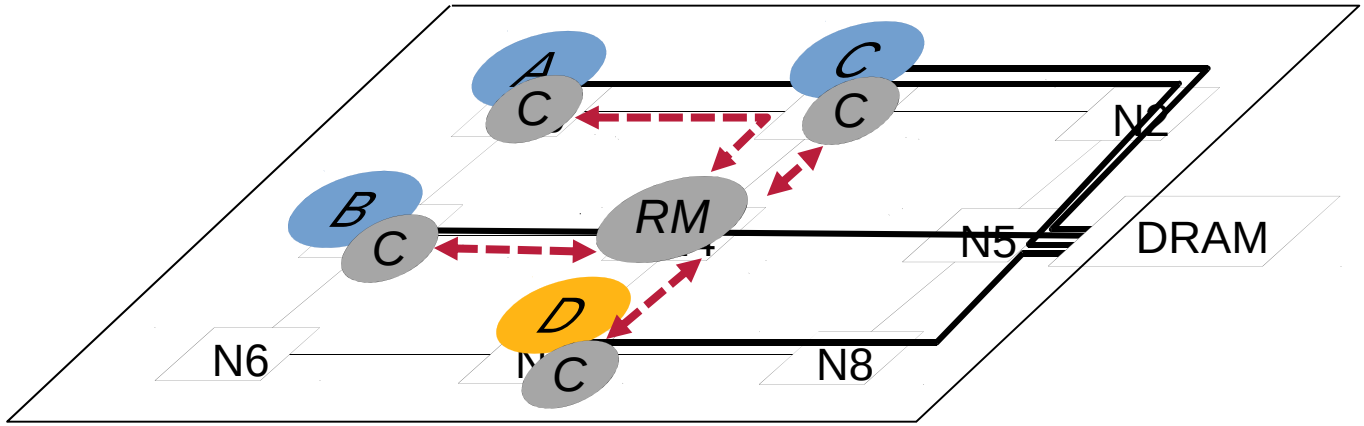
(virtual) control layer

data layer

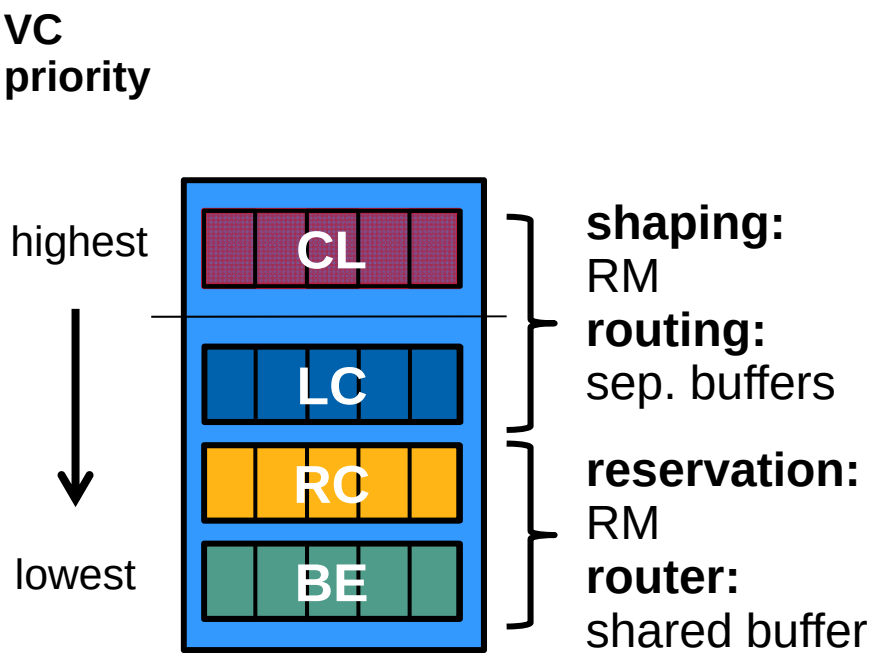
RM: resource manager
NI: network interface
R: router
MI: memory interface

Overlay network

- **overlay network to decouple data flow and control protocol**
- **data layer – data transport and data routing and arbitration**
- **control layer – global and dynamic arbitration**
  - **clients** - admission control locally in nodes
  - **RM** – central scheduling unit
  - protocol based **synchronization**



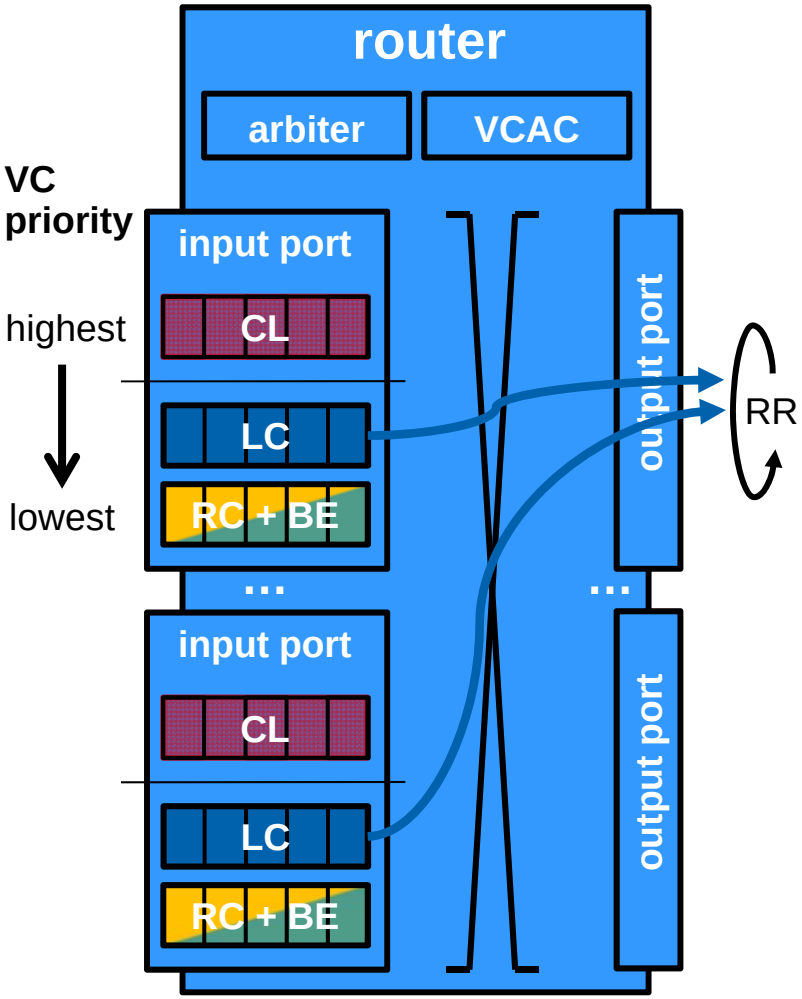
Network virtual channels



WC timing analysis for all levels and mechanisms developed

- **control layer VC: CL**
  - single flit message size
  - min.distance shaping – small buffer
- **latency critical VC: LC**
  - *optional* VC for cache traffic
  - max 32/64 bytes payload (256 bit)
  - programmable min. distance shaping
- **reserved critical VC: RC**
  - VC reserved for single traffic/link
  - reservation controlled by RM (no local arb.)
- **best effort VC: BE**
  - VC reserved for single traffic/link
  - reservation controlled by RM (no local arb.)
  - RM controlled preemption for RC traffic

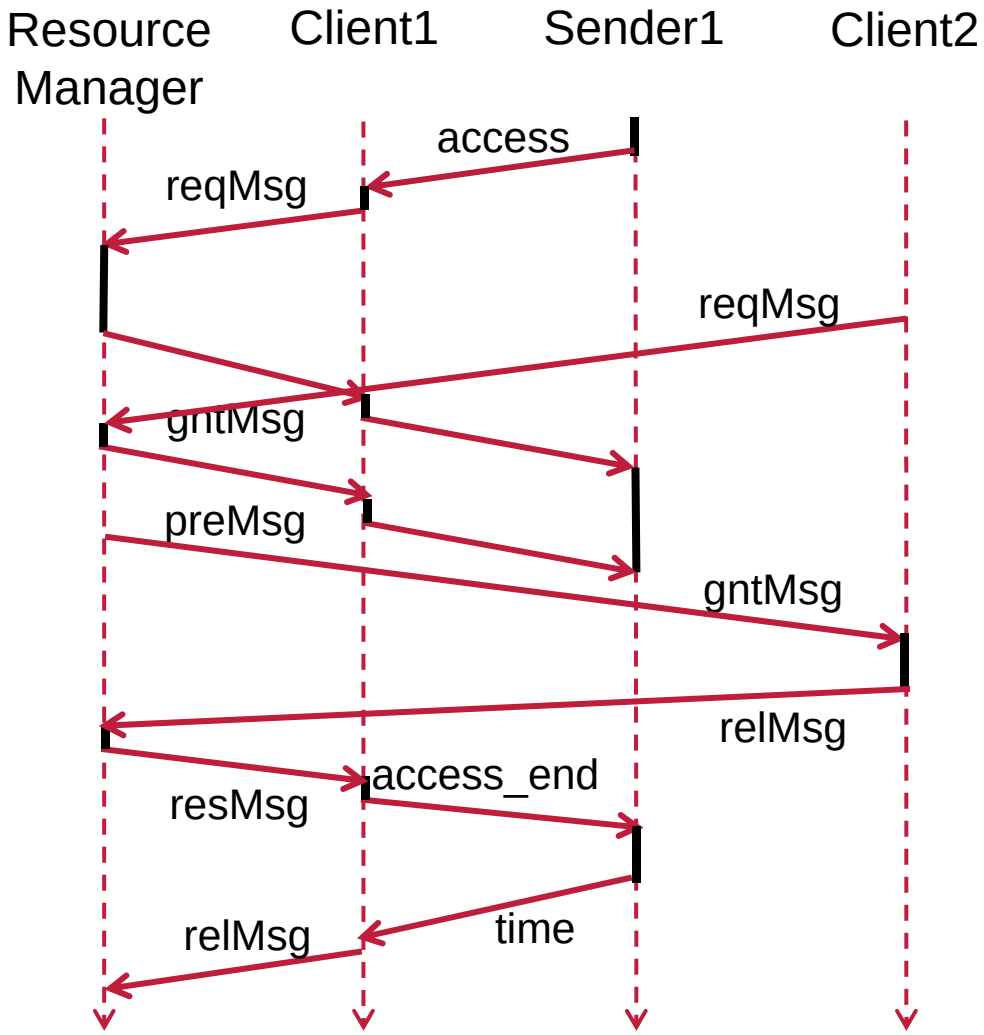
Router architecture



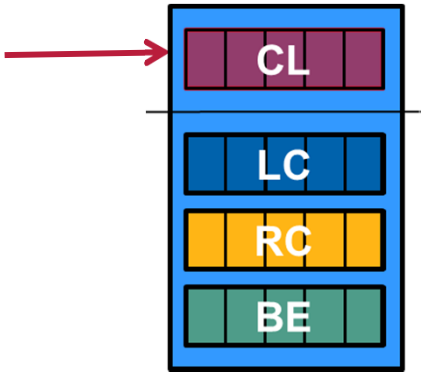
- **input buffering + wormhole switching**
  - small buffers
- **crossbar switch**
- **priority-based arbitration**
  - highest priority packet is transmitted first
  - round-robin between packets in the same priority
  - FIFO inside VCs

WC timing analysis for all levels and mechanisms developed

Resource reservation protocol

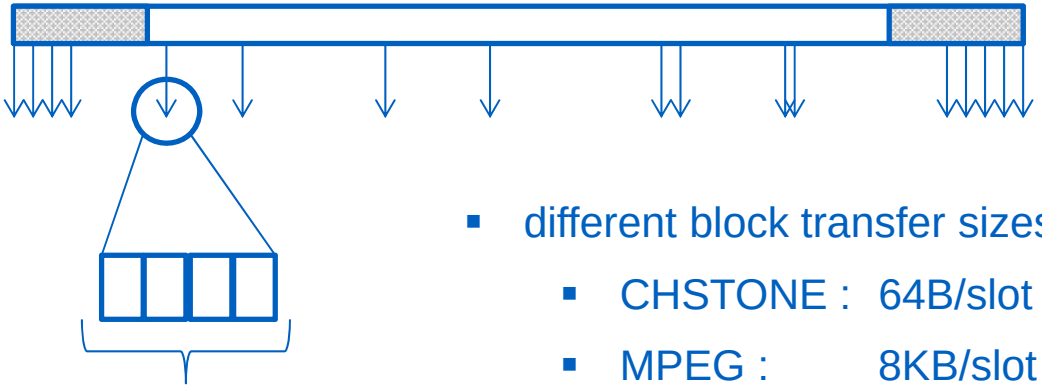


single flit control message size (128 bit)



Experiments

- **analytical WC experiments**
  - pyCPA analysis framework
- **simulations**
  - OMNeT++ event-based simulation framework
  - HNOCS library
- **input data**
  - memory access traces

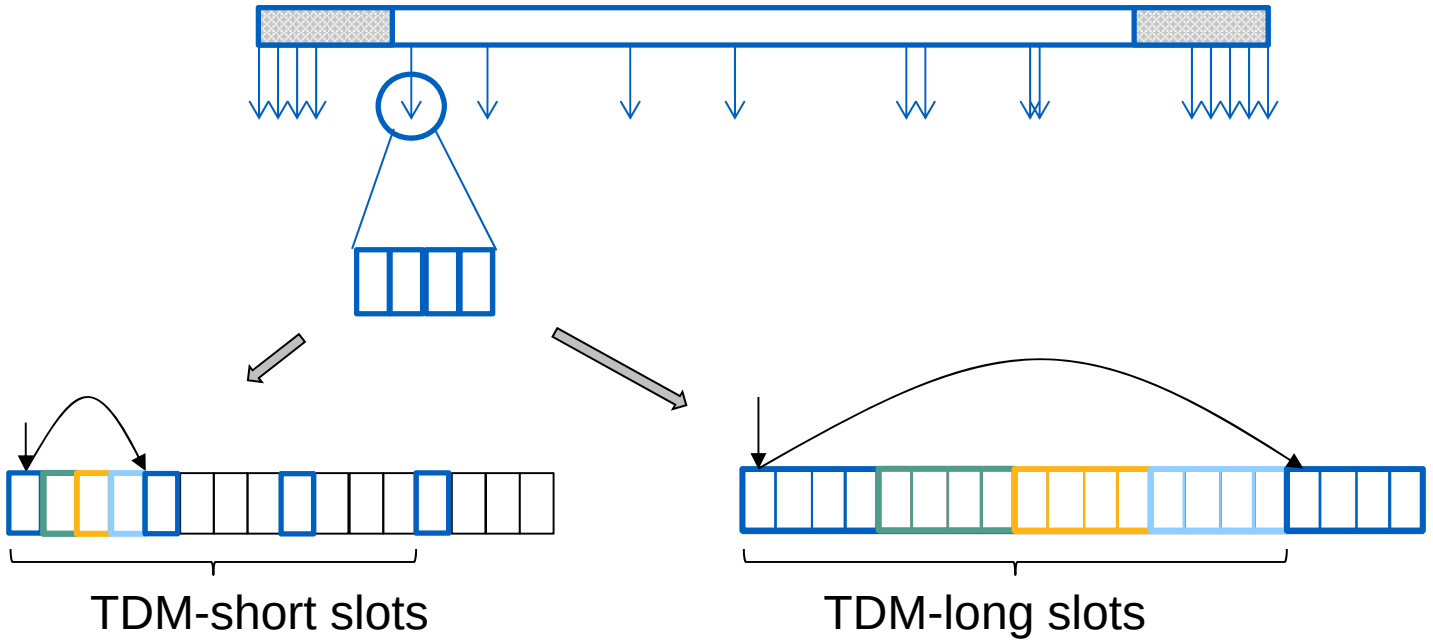


block transfer

- different block transfer sizes
  - CHSTONE : 64B/slot (4 packets)
  - MPEG : 8KB/slot (125 packets)

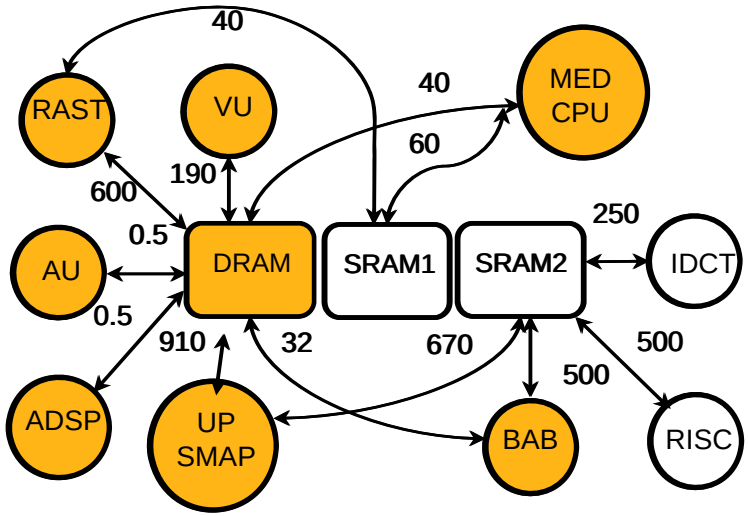
Experiments

- **comparison with**
  - **TDM with short slots** (cp. Aetheral) – slot size adjusted to the network latency of a single packet
  - **TDM with long slots** – slot size sufficient for complete block transmission (64B/slot – 4 packets; 8KB/slot – 125 packets)

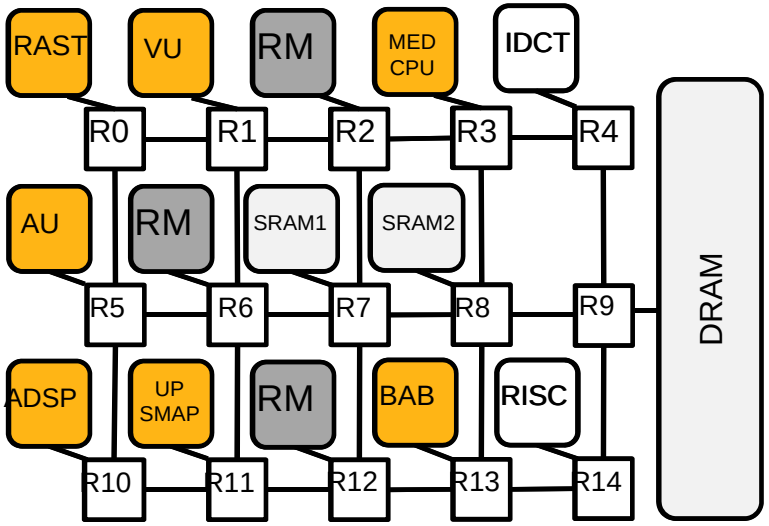


MPEG-4 Use-case

MPEG function

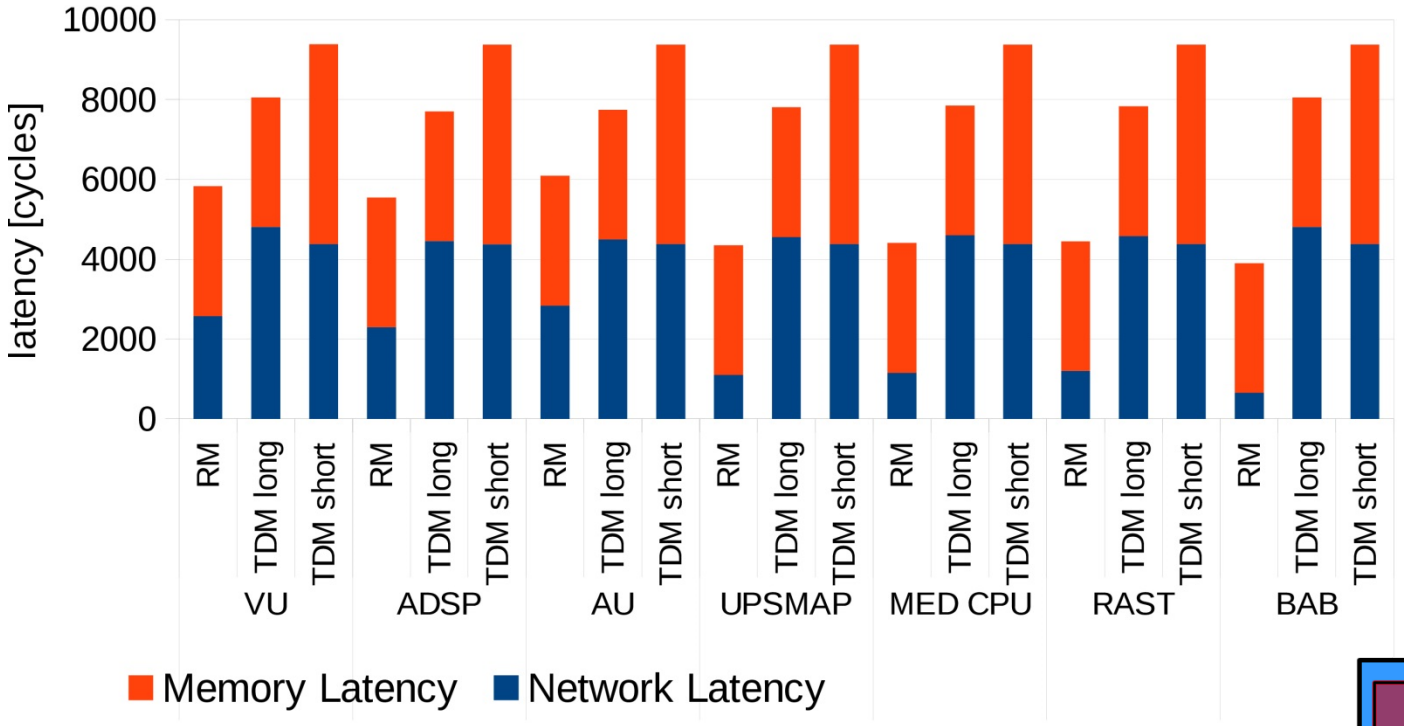


function mapping

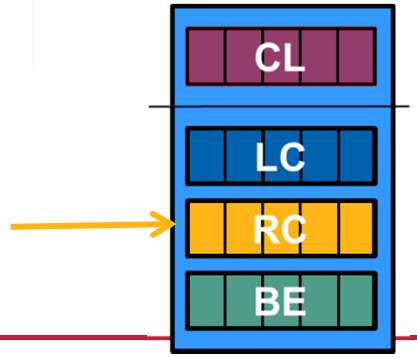


- average bandwidth demands in MB/s

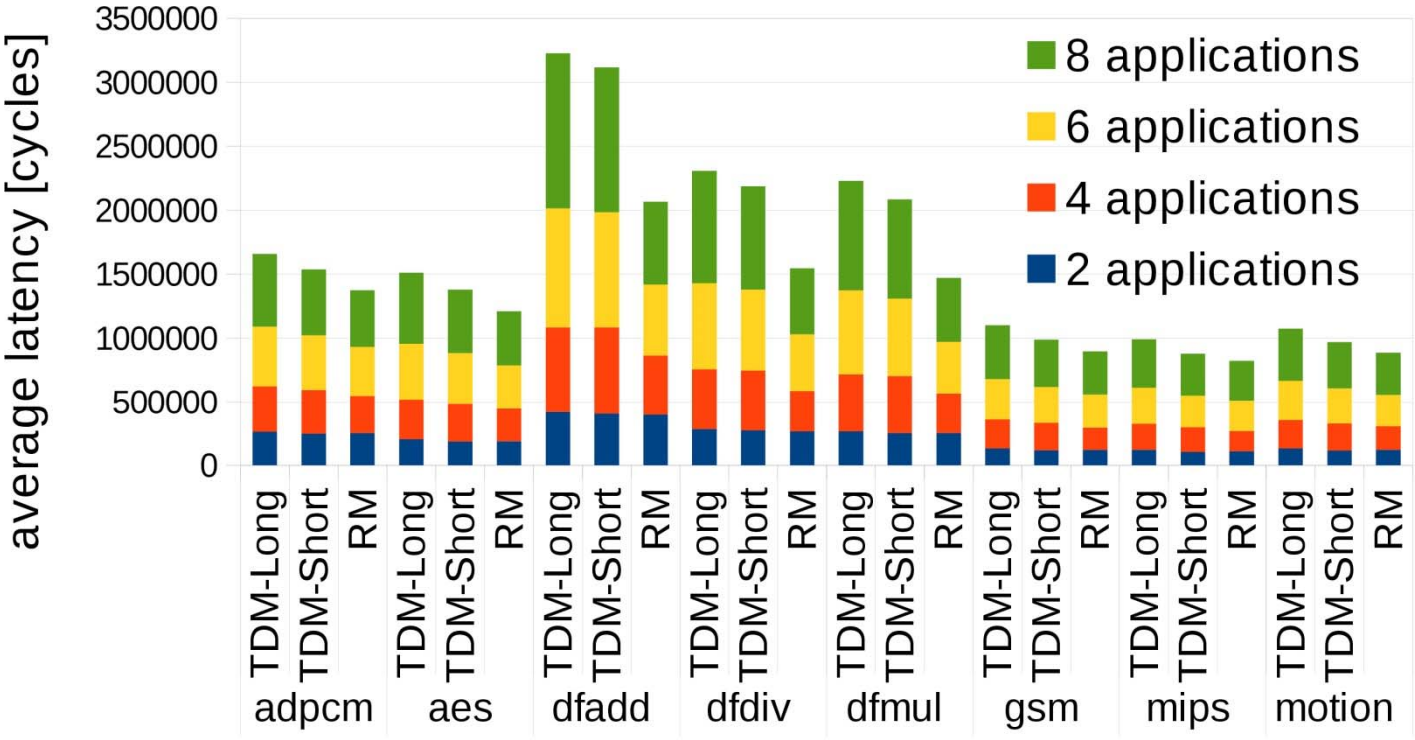
MPEG-4 worst case block transfer timing



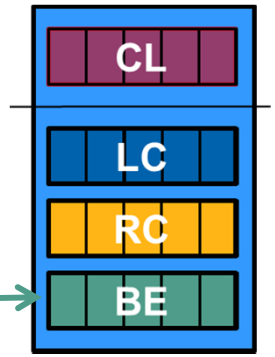
- **worst case latency per block transfer**
- **8KB block transfer**
- **uses RC channel**



RM for BE traffic



- **average total latency for CHSTONE benchmark**
- **64B block size**
- **uses BE channel**

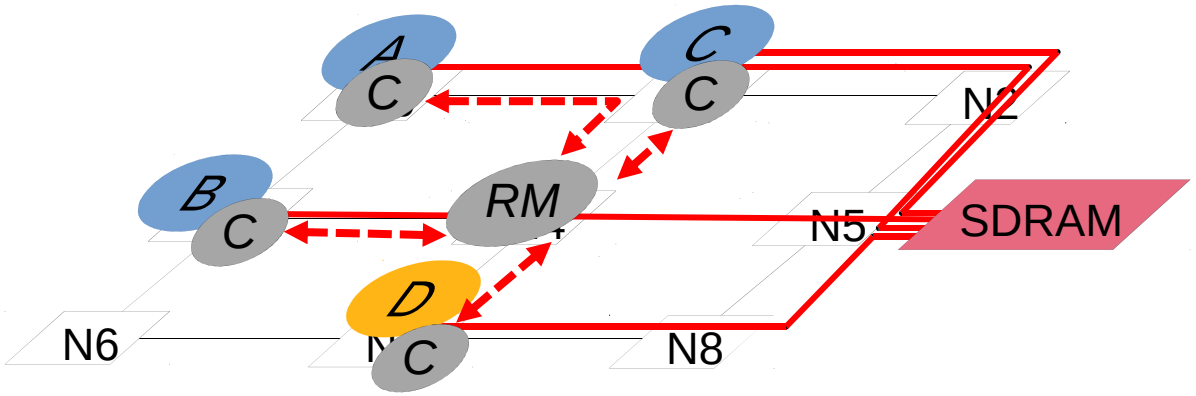


Outline

- Motivation
- NoCs for hard real-time and robustness
- Dynamic resource management
- **Improving memory control and error handling**
- Conclusions

Resource Manager and SDRAM scheduling

- due to the preservation of spatial locality, a simple SDRAM controller that serves requests in FCFS order suffices
  - SDRAM scheduling is implicitly delegated to RM
  - hence, *mixed criticality* can be addressed at the RM level

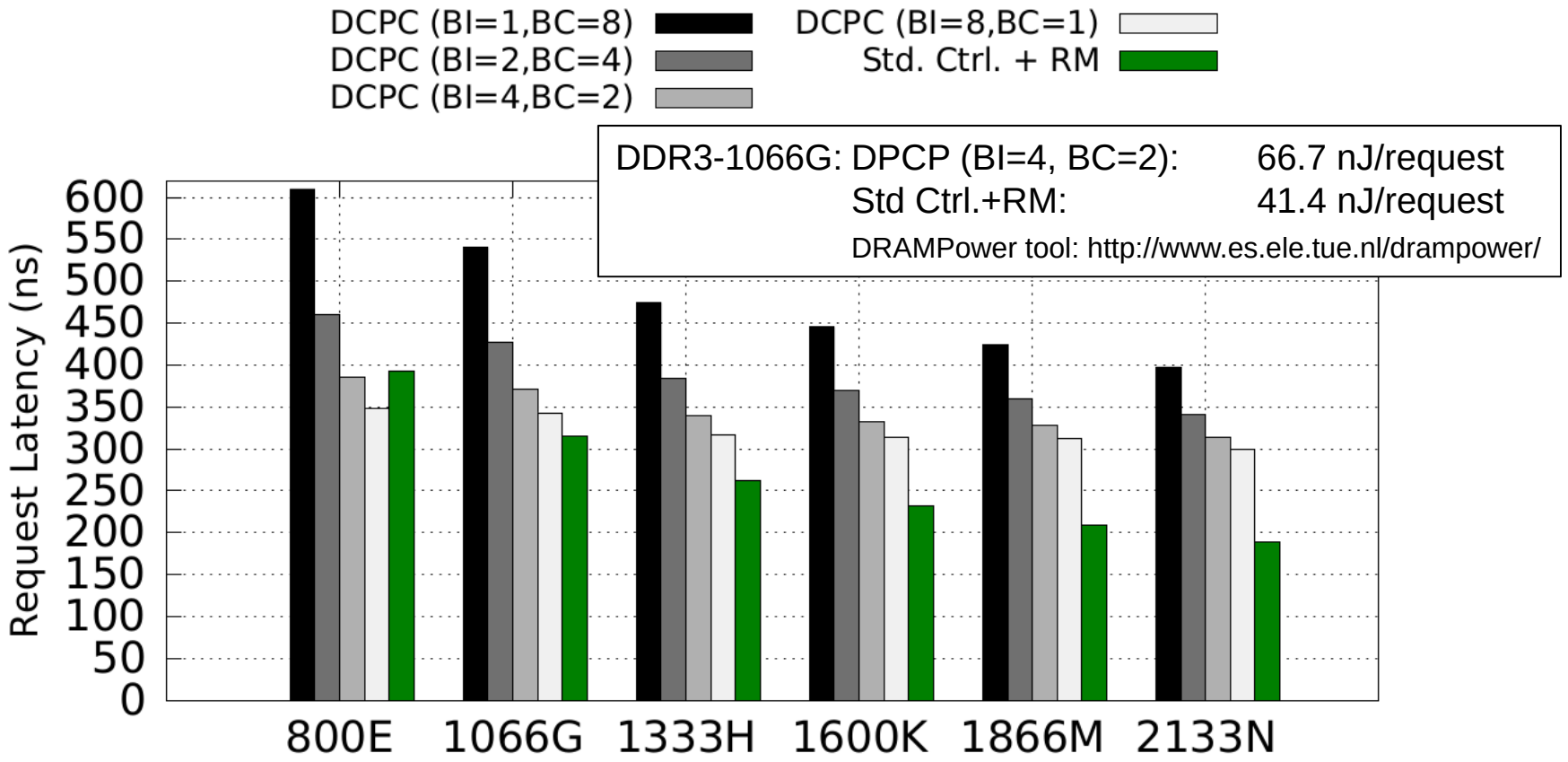


RM vs. *predictable* SDRAM scheduling

- **predictable SDRAM schedulers**
  - where spatial locality of transfers is not enforced (e.g. TDM with small slots), *predictable* SDRAM controllers are required
  - to deal with lack of locality, such controllers employ *close-page* policy and/or bank interleaving - **D**edicated **C**lose **P**age-**C**ontrollers **(DCPC)**.
  - the operation DCPCs is controlled by two parameters: BI and BC
    - BI (Bank Interleaving) no of banks per access
    - BC (Burst Count) no of *read* or *write commands executed per bank*

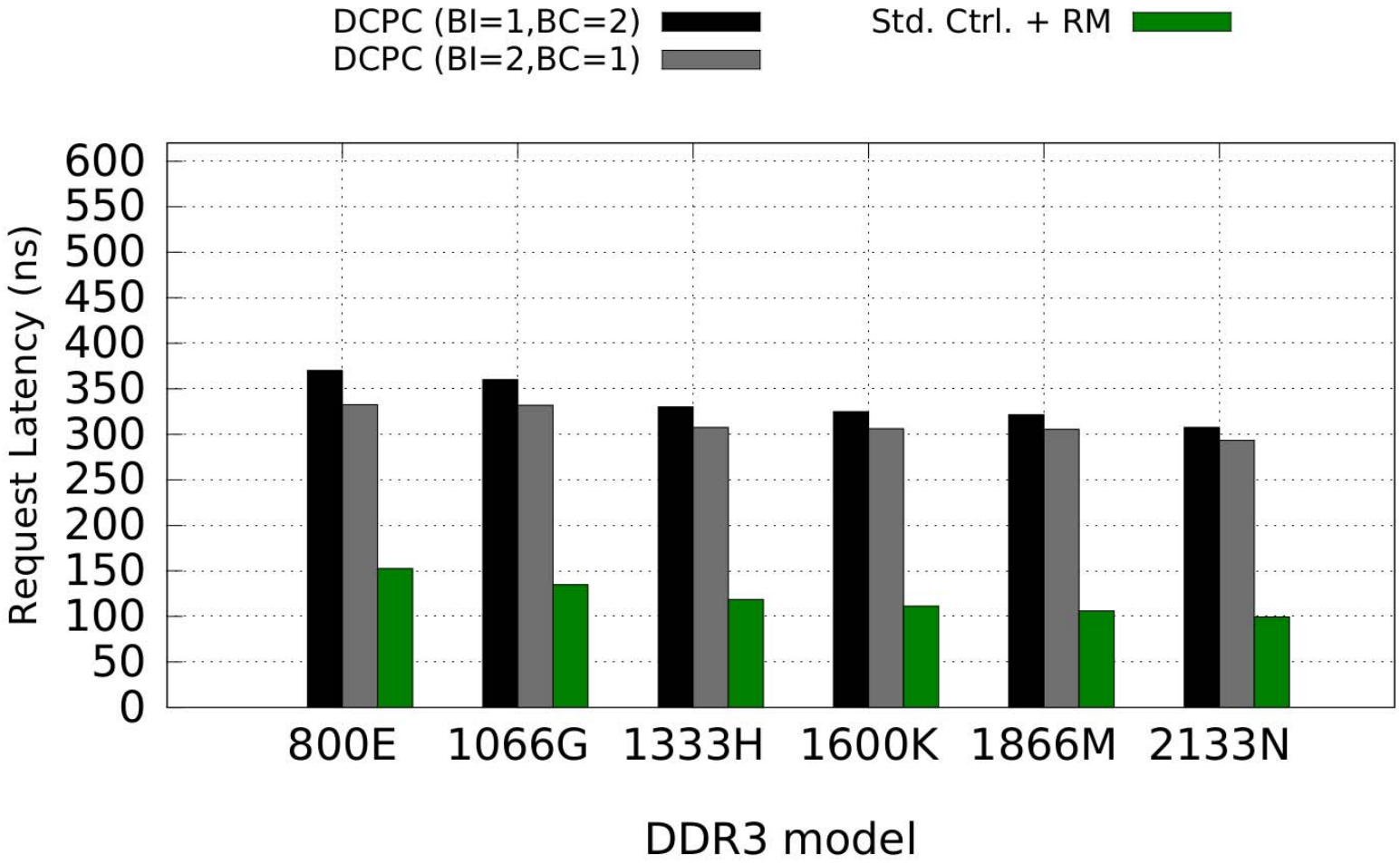
- **RM + standard SDRAM controller**
  - keeps spatial locality
  - allows reduction of row buffer open and close operations
  - out-of-order optimization must be turned off (FCFS memory scheduling)
  - choose BI = 1 → leads to predictable access timing

Resource Managers and SDRAM scheduling



Worst-case latency for a 4x64B request on DDR3 devices (with 8-bit wide interfaces)

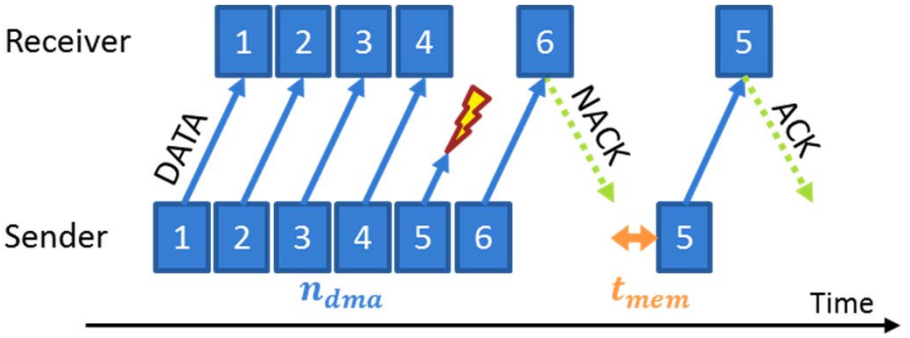
Resource Managers and SDRAM scheduling



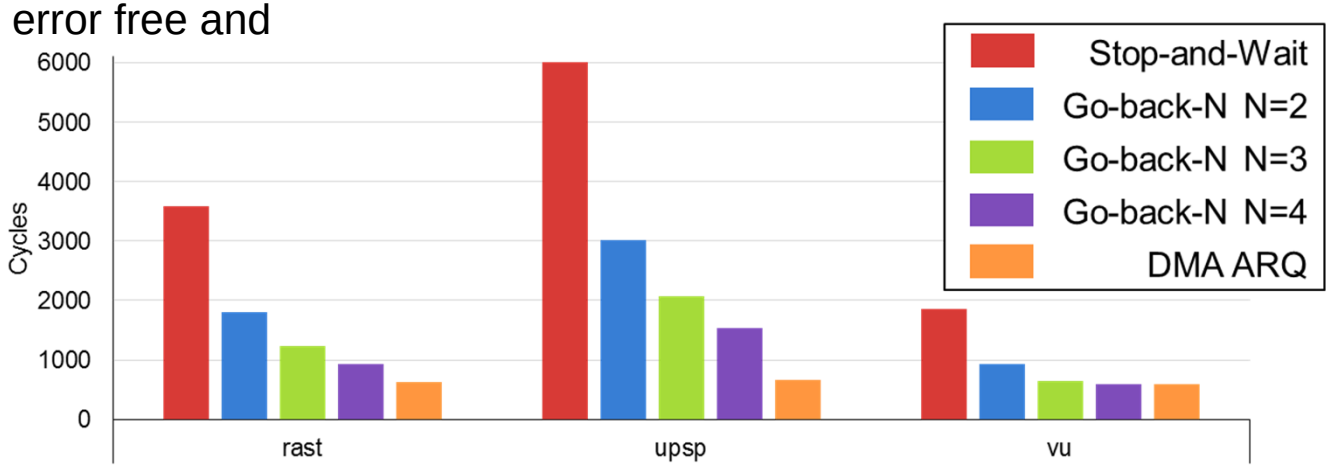
Worst-case latency for a 4x64B request on DDR3 devices (with 32-bit wide interfaces)

Handling packet losses in block transfers

- **transient error** or **packet drop (BE)**
- **DMA ARQ - variant of Go-back-N and Selective Repeat**
  - data retransmitted from memory
  - acknowledge entire DMA transfers (send window $\boldsymbol{n} = \boldsymbol{n_{dma}}$)
  - selective retransmission
- low overhead end-to-end protection
  - WC timing analysis for error free and error cases available
- ***very efficient error protection***



DMA transfer time (8KByte, error free)



processes from MPEG4 example

Outline

- Motivation
- NoCs for hard real-time and robustness
- Dynamic resource management
- Improving memory control and error handling
- **Conclusions**

Conclusion

- **NoC based many-cores are entering safety critical system design**
- **mixed criticality is result of function integration**
- **dynamic resource management using a research manager is a highly efficient NoC control mechanism for such NoCs providing worst case guarantees**
- **mechanism supports simpler memory control and transient error handling**

Thank you!

Acknowledgement: Some of the slide contents have been provided by Leonardo Ecco, Adam Kostrzewa, and Eberle Rambo

Bibliography

- A. Kostrzewa, S. Saidi, and R. Ernst, “Dynamic control for mixed-critical networks-on-chip,” in RTSS, 2015.
- A. Kostrzewa, S. Saidi, L. Ecco, and R. Ernst, “Dynamic admission control for real-time networks-on-chips,” ASP-DAC 2016.
- Rambo, Eberle A., and Rolf Ernst. "Worst-case communication time analysis of networks-on-chip with shared virtual channels." 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2015.
- Rambo, Eberle A., Selma Saidi, and Rolf Ernst. "Providing formal latency guarantees for ARQ-based protocols in Networks-on-Chip." 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2016.