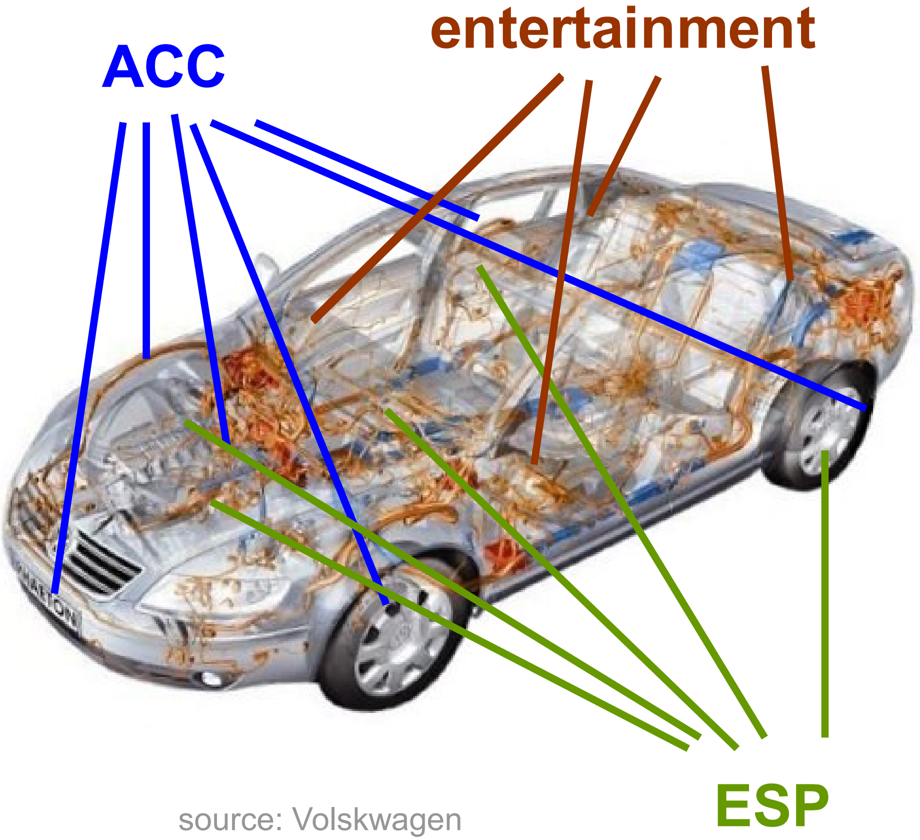
Dynamic Admission Control for Real-Time Networks-On-Chips

Adam Kostrzewa, Selma Saidi, Leonardo Ecco, Rolf Ernst
TU Braunschweig

Motivation

[redacted]

- today’s many-core real-time systems
  - many integrated functions i.e. tasks and ECUs
  - networked control
  - many suppliers → heterogeneous

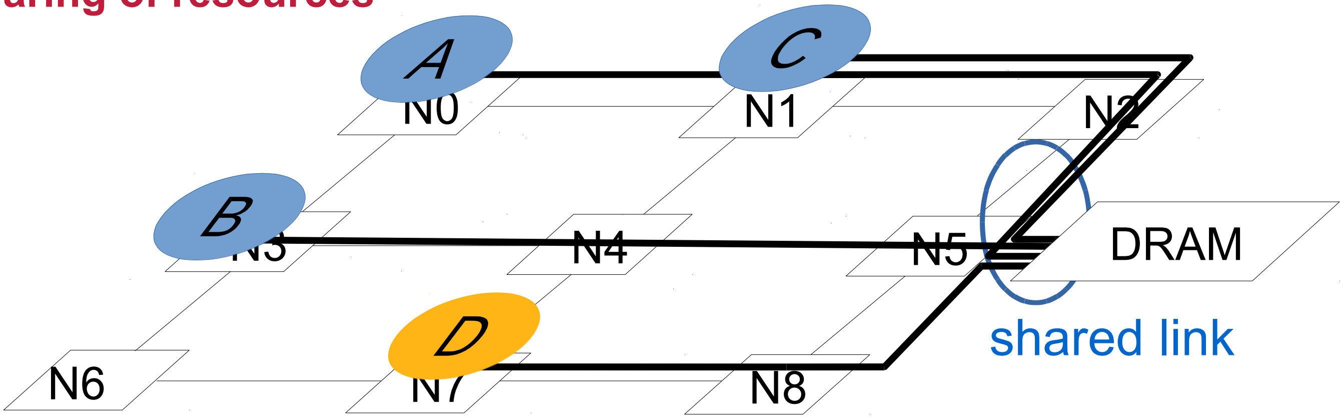


source: Volskwagen

Motivation

[redacted]

- Networks-on-Chip are an efficient platform for **systems integration**
- Running transmissions compete for shared resources
  - link bandwidth or buffer space
- **NoC must assure:**
  - **spatial and temporal independence isolation between interfering transmissions**
  - **efficiency of sharing of resources**



Challenge → Assuring predictable and efficient execution!

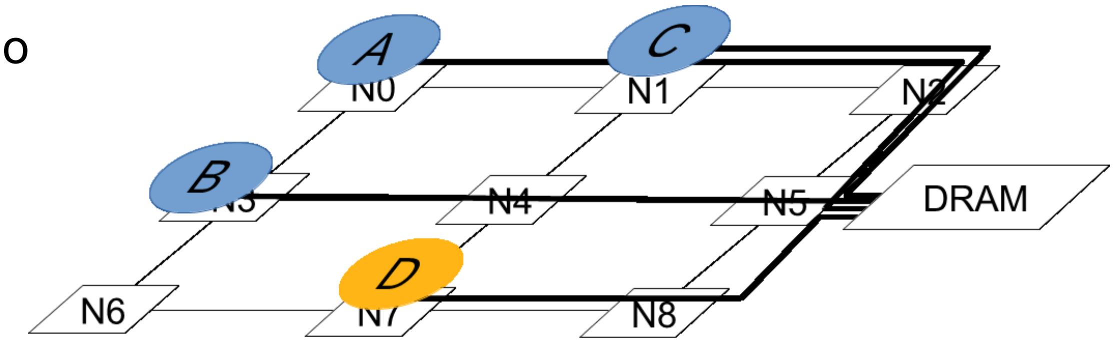
Outline

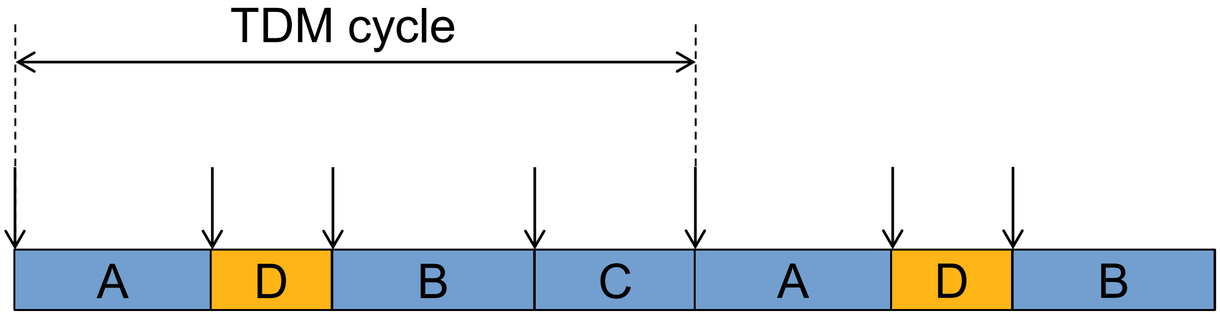
- Motivation
- TDM-based arbitration for NoCs
- Our Solution – Resource Manager
- RM’s Predictability
- Experimental Evaluation
- Conclusions

Time-Division Multiplexing

[redacted]

- TDM the most frequently deployed solution for enforcing isolation
- Resources are shared in time – **cyclic order**
- Entire **NoC is a globally shared** resource
  - each application/transmission has a time slot
  - accesses granted in a cyclic o
  - exclusive access to NoC





TDM Advantages

[redacted]

- isolation - temporal & spatial
- predictability and formal guarantees
  - compute the worst-case latency of a transmission → deadlines
- relatively simple implementation
- transmissions acquire exclusive access to the NoC
  - designer may guarantee absence of contention
  - therefore, reduce hardware overhead
    - buffers, logic in routers
- relatively simple analysis

$$\boldsymbol{R_i} = \underbrace{\boldsymbol{C_i}}_{\text{own transmission time}} + \underbrace{(\mathbf{t} - \boldsymbol{s_i})}_{\text{other slots}} * \underbrace{\left\lceil \frac{\boldsymbol{C_i}}{\boldsymbol{s_i}} \right\rceil}_{\text{max. num. of slots required}}$$

$\mathbf{t}$ – duration of whole TDM-cycle

$\boldsymbol{s_i}$ – duration of time slot for $\boldsymbol{i}$

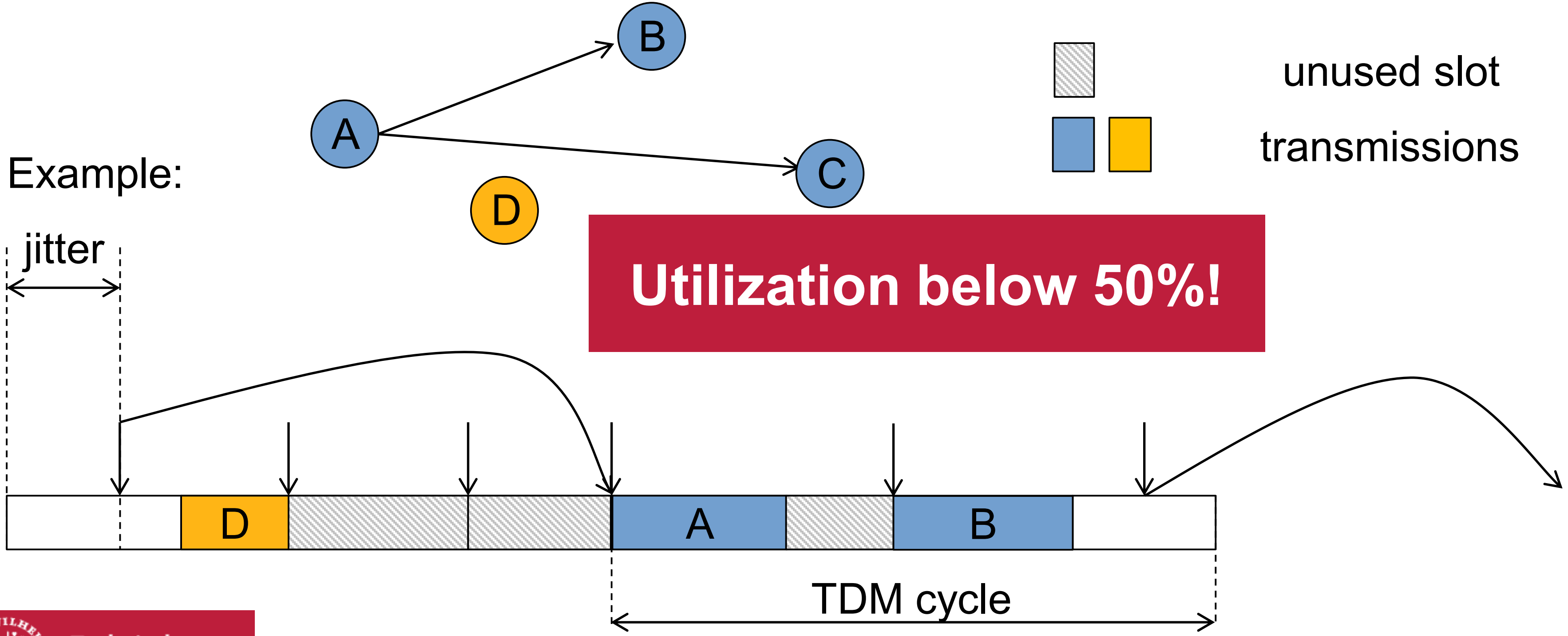
TDM with dynamics

[redacted]

Problems: **TDM is static and non work-conserving**

- unused slots are wasted
- cannot cope with dynamics (e.g. data dependent execution)
  - release jitter, execution time, communication volume
- negative effects are amplified in case of task-chains

Example:



Utilization below 50%!

TDM Utilization

[redacted]

- efficient utilization is possible only if :
  - NoC is continuously requested (highly loaded)
  - under absence of dynamics
- this implies dedicated optimized/solutions
  - is it possible to get on core scheduling resulting in continuous accesses?
  - is it possible to fully exclude dynamics?
    - changes in toolchain, integrated components, porting to different platforms
- otherwise, low utilization and average latencies close to the worst-case
  - even in lightly loaded system

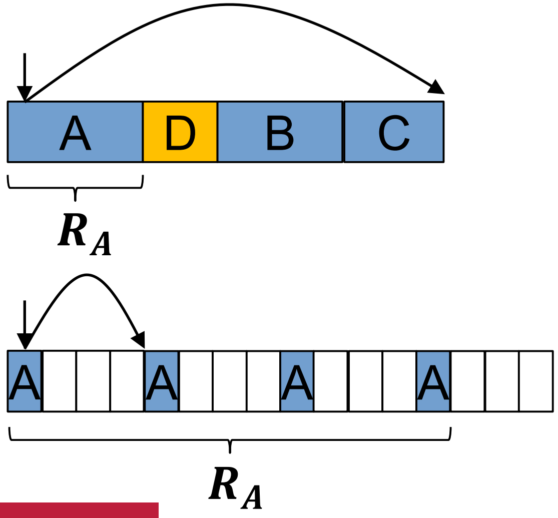
TDM is predictable but usually not efficient!

TDM Countermeasures

[redacted]

- overhead is proportional to the length of the cycle
  - number of integrated applications
  - length of the slots
  - NOT frequency of accesses

- decreasing the length of the TDM cycle by short slot-length
  - e.g. Æthereal



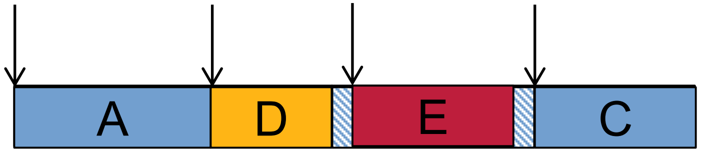
- distribution of longer transmissions over several cycles (even if NoC is free)
- underutilization of peripherals and modules
- e.g. too short transmissions towards SDRAM result in drastic increase of command overhead

TDM Countermeasures

[redacted]

- **optimized TDM scheduling** e.g. PhaseNoC or SurfNoC
  - replacing the cycle by cycle schedule with more flexible solutions e.g. domain oriented waves
  - decrease the gap between real access pattern and cyclic transmissions
  - increase complexity of the design
  - therefore hardware overhead and power consumption

- **multiplexing of time-slots between channels e.g. Channel Trees**



  - no guarantees
  - effectiveness depends on the number of VCs
  - static budgets -> same problems as in case of TDM

Technische Universität Braunschweig

Outline

- Motivation
- TDM-based arbitration for NoCs
- Our Solution – Resource Manager
- RM's Predictability
- Experimental Evaluation
- Conclusions

Our solution

[redacted]

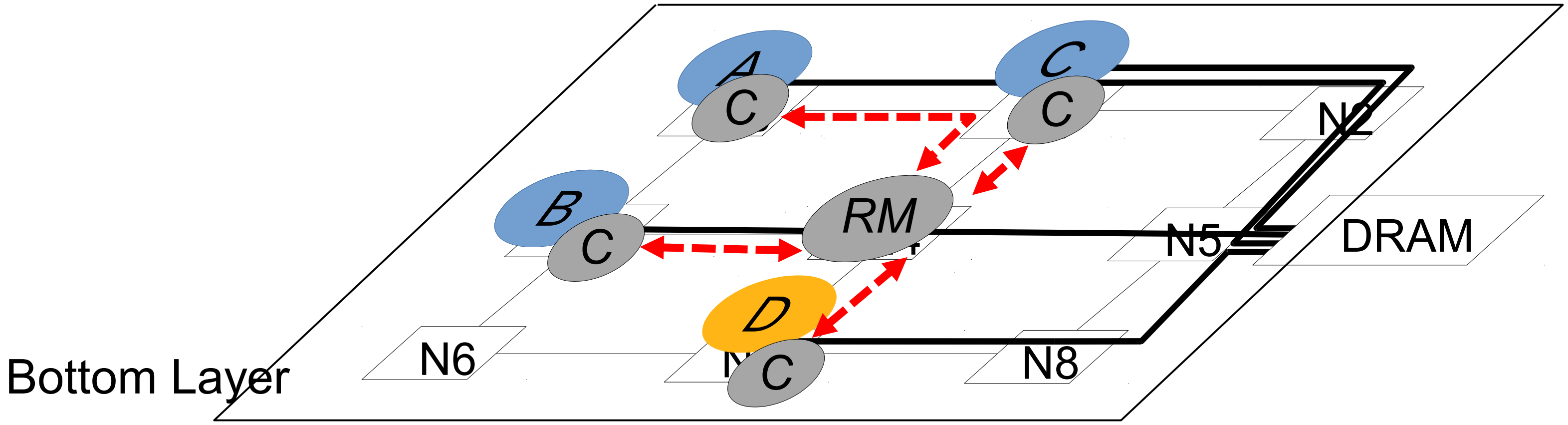
- **Predictability**
  - isolation of whole transmission composed of multiple packets
  - guarantees for the whole transmission instead of a single packet
- **Efficiency**
  - dynamically adapt arbitration to the current load
    - work conserving arbitration (round-robin based)
      e.g. skipping the slots of non-active senders
    - preserve locality of network transfers
      - DMA transfers towards DRAM
- **Low implementation overhead**
  - mechanism build on top of existing performance optimized networks
  - NoC as globally shared resource → small buffers
  - very little modifications of running components

Mechanism Description

[redacted]

- overlay network to decouple flow and admission control
- **bottom layer** – low-level flow-control method in NoCs responsible for switching packets/flits
- **virtual top layer** – global and dynamic arbitration
  - Clients - admission control locally in nodes
  - RM – central scheduling unit
  - protocol based synchronization

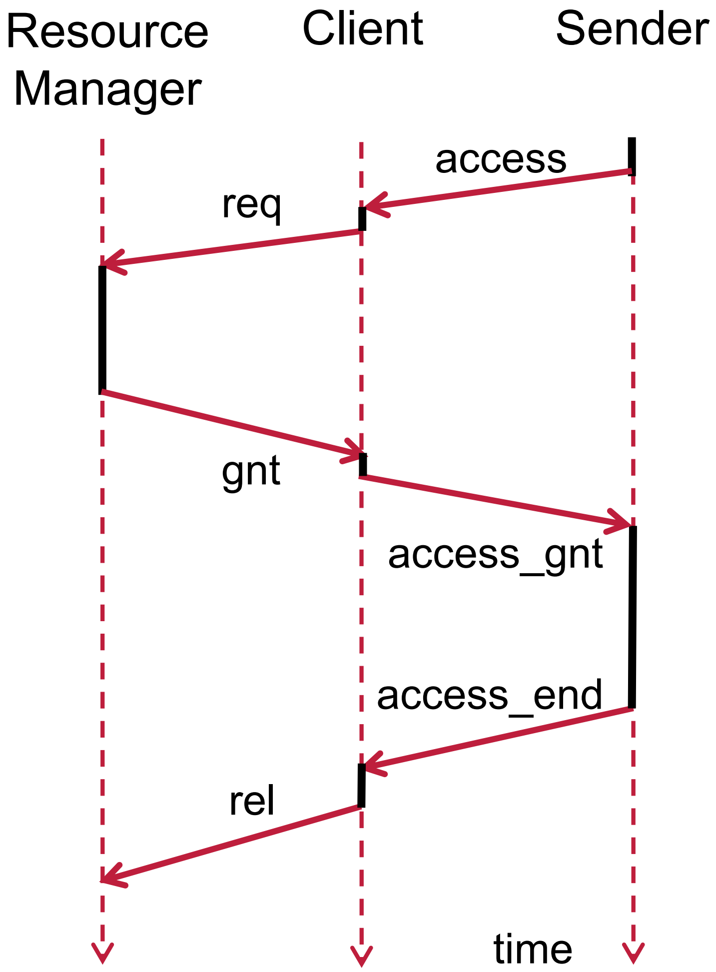
} distributed network hypervisor



Workflow

[redacted]

- Sender starts trans. → access to the NoC
- Client traps this access
- Client sends a request to RM
- RM performs scheduling
- RM sends a grant to Client
- Client permits Sender to use the NoC (whole trans. == multiple packets)
- Sender conducts transmission
- Client detects end of the transmission (timeout monitor, last flit)
- Client sends a release to RM



Requirements

[redacted]

Bottom Layer

- work-conserving scheduling done locally in routers e.g. round-robin, iSLIP
- predictable behavior of routers
  - arbiters in routers must be analyzable with one of the existing analysis methods

- protocol-based synchronization requires safe communication channel
  - dedicated VC capable of giving latency guarantees
  - control NoC for maximum efficiency

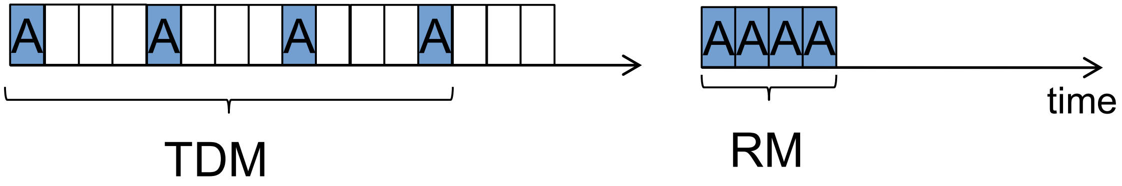
Synchronized transmissions

- overlapping streams share resources – buffers and links
  - share at least one link – path-based arbitration
  - on the same Virtual Channel

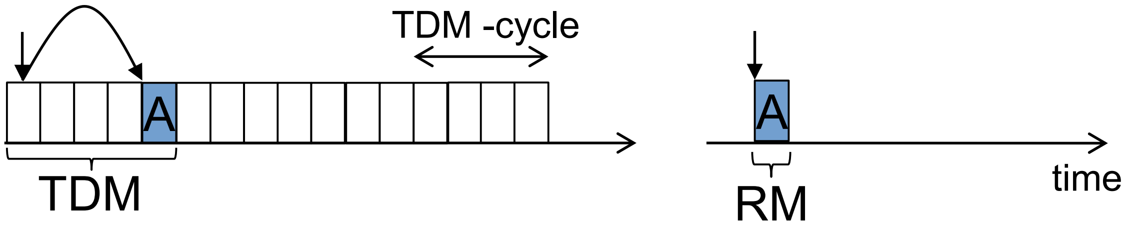
Advantages

[redacted]

- work-conserving scheduling done
  - blocking proportional frequency of transmissions
    e.g. multiple transmissions from the same sender if the network is free



  - automatically incorporates dynamics



- no need to modify routers
- useful for synchronization of longer transmissions
  - DMA-based memory transfers

Outline

- Motivation
- TDM-based arbitration for NoCs
- Our Solution – Resource Manager
- RM’s Predictability
- Experimental Evaluation
- Conclusions

Predictability

[redacted]

- mechanism description → mathematical model
- calculate the worst-case latency of a transmission
- validate against the deadlines

- ***busy window*** approach
  - assuming maximal activation rate of synchronized senders
  - and arbitrary activation patterns of transmission
- transmissions abstracted with ***event models***
  - $\boldsymbol{\eta}^{+}(\Delta \boldsymbol{t}),\ \boldsymbol{\eta}^{-}(\Delta \boldsymbol{t})$ maximum and minimum number of initiated transmissions during time period $\boldsymbol{\Delta t}$
  - framework: **Compositional Performance Analysis (CPA)**
- we focus on the top-layer applying the **round-robin scheduling** overview – details in the paper

Predictability

[redacted]

- **the worst-case time necessary to conduct q transmissions ($\mathbf{w}^{+}{}_{i}(\mathbf{q})$)**

$$\mathbf{w}^{+}{}_{i}(\mathbf{q}) = \underbrace{\mathbf{q} * \mathbf{C}^{+}{}_{i}} + \underbrace{\mathbf{3q} * \mathbf{C}^{+}{}_{i,ctrl}} + \underbrace{\mathbf{B}_{i}\left(\mathbf{w}^{+}{}_{i}(\mathbf{q})\right)}$$

duration of q trans.

protocol overhead (three ctrl.
msgs. per transmission (req, ack, rel))

the maximum blocking resulting from scheduling
of other synchronized transmissions

Predictability

[redacted]

- blocking time which q requests experience in a time window Δt can be bounded by:

$$\mathbf{B}_i(\Delta \boldsymbol{t}) = \sum\nolimits_{j\epsilon S} \underbrace{(\mathbf{C}_j + \mathbf{2} * \mathbf{C}^{+}{}_{j,ctrl})}_{\text{maximal blocking caused by a single trans from } Tj} * \boldsymbol{min}\,\{\underbrace{q}_{\text{q activations of } Ti},\ \underbrace{\eta^{+}{}_{j}(\Delta t)}_{\text{max num. of activations of } Tj}\}$$

- $\boldsymbol{min}$ function is to denote that transmissions from ***Tj*** cannot block ***Ti*** more than:
  - q times that ***Ti*** is activated
  - $\boldsymbol{\eta}^{+}{}_{\mathbf{j}}$ number of its own (***Tj***) activations

Outline

- Motivation
- TDM-based arbitration for NoCs
- Our Solution – Resource Manager
- Predictability
- Experimental Evaluation
- Conclusions

Experimental Evaluation

[redacted]

- **simulations**
  - **OMNeT++** event-based simulation framework
  - **HNOCS** library
- **CHSTONE** benchmarks

- comparison with
  - **TDM with long slots** – slot size adjusted to the duration of entire transmission

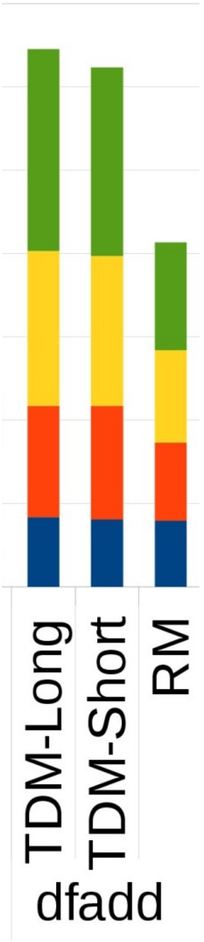
$$s_i = C^{+}{}_{i}$$

  - **TDM with short slots** – slot size adjusted to the network latency of a single packet

$$s_i = C^{+}{}_{i,pkt}$$

CHSTONE benchmark

[redacted]



Experimental Evaluation

[redacted]

- **analytical experiments**
  - **pyCPA** analysis framework
    - pragmatic Python implementation of Compositional Performance Analysis
    - worst-case timing analysis
    - using **event models**

- **synthetic** and **MPEG-4** as benchmarks

- comparison with
  - **TDM with long slots** – slot size adjusted to the duration of entire transmission

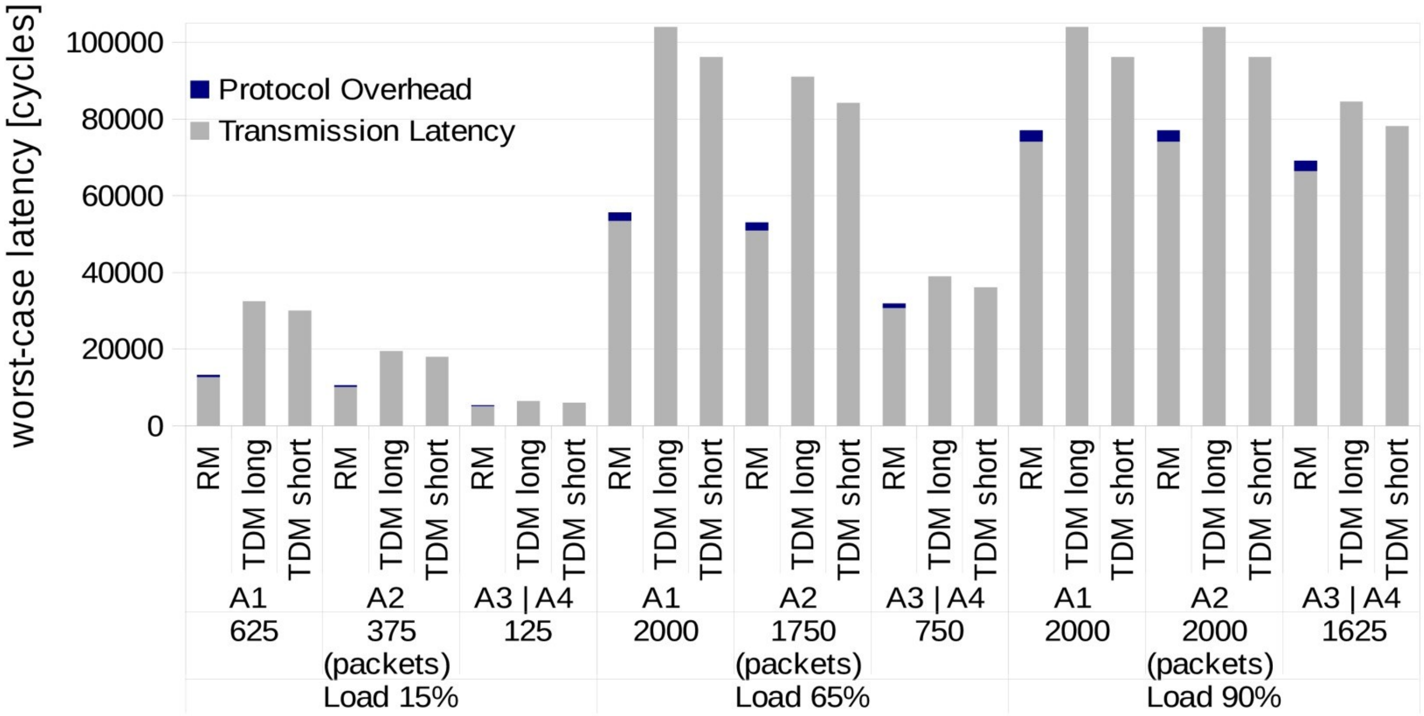
$$s_i = C^{+}{}_{i}$$

  - **TDM with short slots** – slot size adjusted to the network latency of a single packet

$$s_i = C^{+}{}_{i,pkt}$$

Worst-Case Guarantees (1)

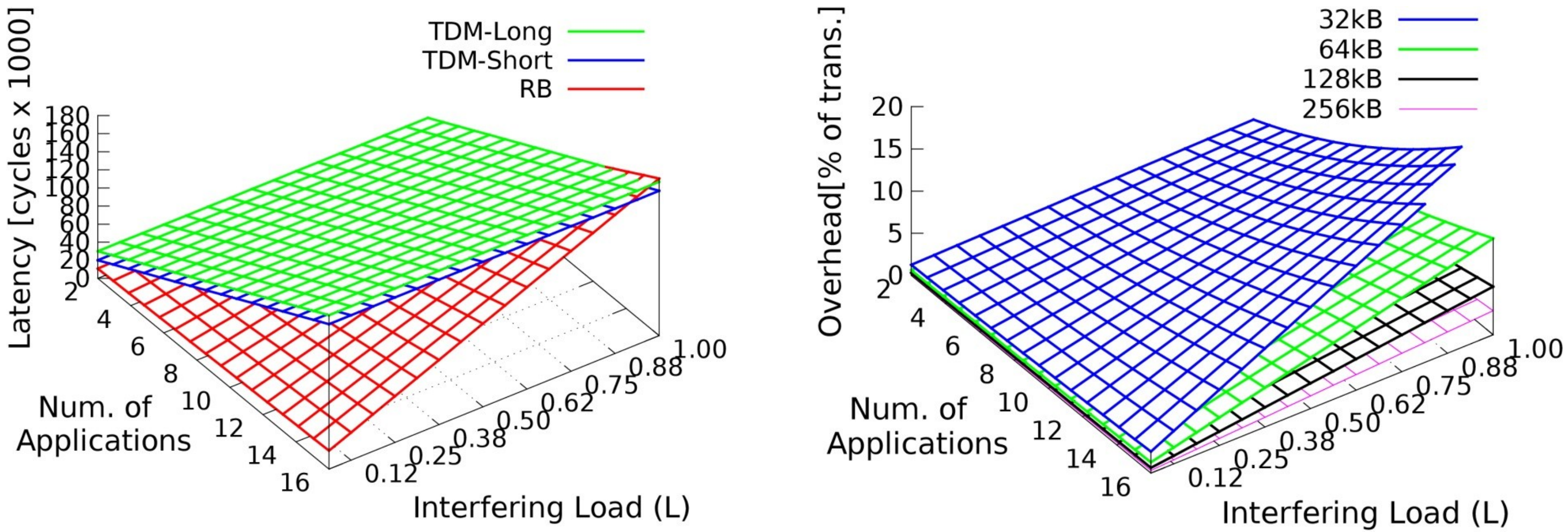
[redacted]



Analytical comparison of worst-case latency guarantees for applications (A1-A4) generating different NoC load.

Worst-Case Guarantees (2)

[redacted]



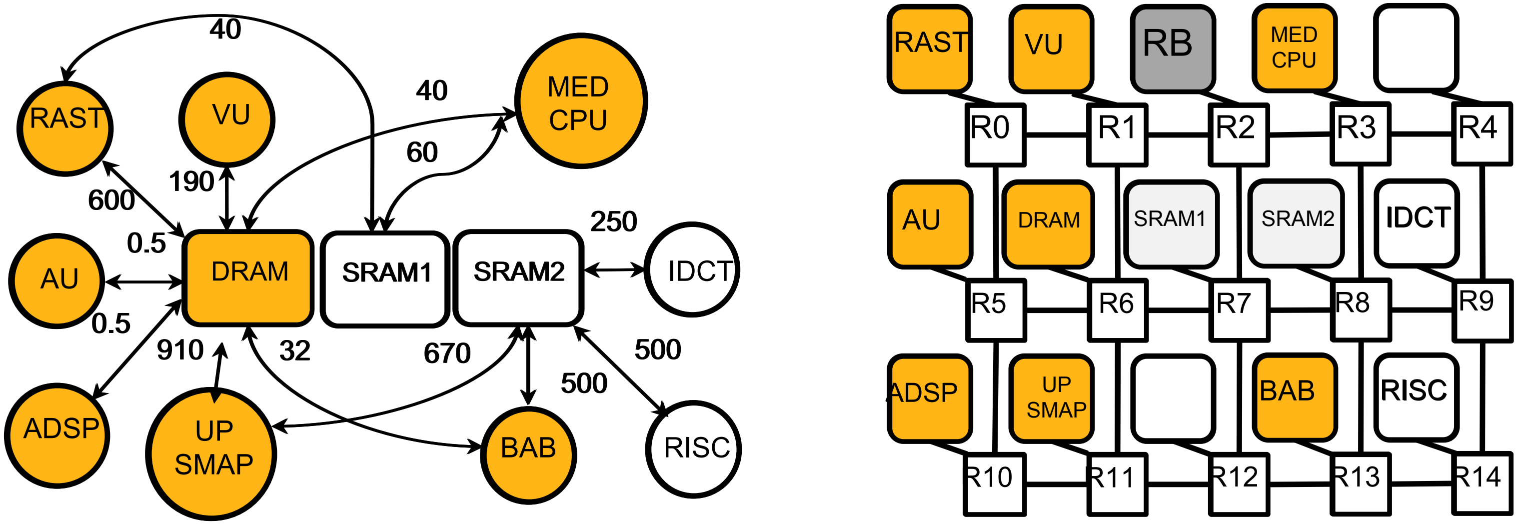
Improvement – up to 80%!

b)

Worst-case guarantees for a burst of 16 transmission with jitter = 10%P
(a) Transmission latency and (b) Protocol overhead resulting from RM.

MPEG-4 Use-case

[redacted]

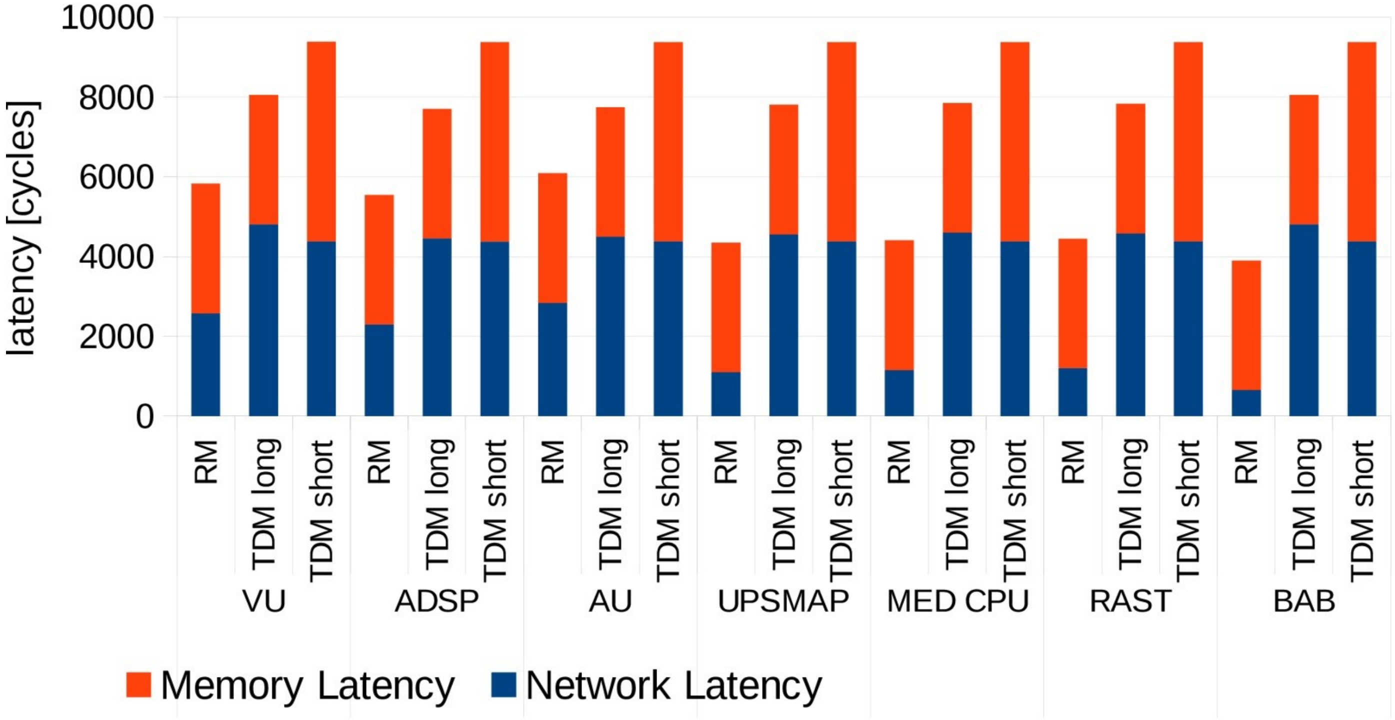


a) b)

- MPEG-4 average communication demands specified in MB/s (a) and mapping (b).
- Locality of memory transfers
    - reduction of DRAM command overhead
    - arrival order of packets in DMA transfer must be assured e.g. 8kB uninterrupted transfers for DDR3-1600 DRAM

MPEG-4 Memory Locality

[redacted]



Effect of memory locality on the total transmission latencies for MPEG-4 module using TDM and RMs.

Technische Universität Braunschweig

Outline

- Motivation
- TDM-based arbitration for NoCs
- Our Solution – Resource Manager
- RM’s Predictability
- Experimental Evaluation
- **Conclusions**

Conclusions

[redacted]

- new method for safe sharing of resources in NoCs
- global and dynamic arbitration
  - work-conserving scheduling
- high predictability
  - proved through the formal worst-case analysis
- low-hardware overhead
  - no modifications of routers
  - possibility of software implementation
- significant improvement over TDM-based solutions

Thank you for your attention!
Questions?