DESIGN, AUTOMATION & TEST IN EUROPE

27 – 31 March, 2017 · STCC · Lausanne · Switzerland

The European Event for Electronic System Design & Test

Real-Time Communication Analysis for Networks-on-Chip with Backpressure

Sebastian Tobuschat and Rolf Ernst

Institute of Computer and Network Engineering (IDA)

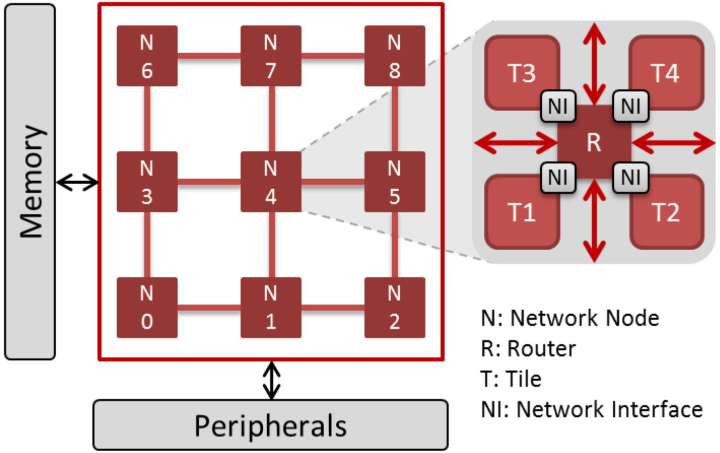
Technische Universität Braunschweig, Germany

Motivation

- many-core systems are reaching safety-critical real-time systems
  - sensor fusion and recognition in highly automated driving
  - avionics, space
- complexity increases
  - integrate previously distributed functions
  - implement new functionality
- safety standards require predictable timing

Motivation

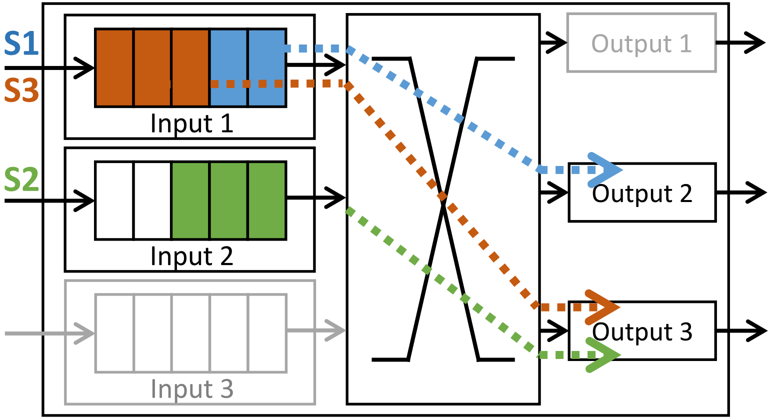
- networks-on-chip (NoCs) offer high-performance, scalability and flexibility
- allow integration of many components
- different transmissions share NoC resources
- limited resources
  - e.g. buffers are shared by different streams
  - backpressure can occur



→timing analysis for NoCs taking backpressure and shared buffers into account

Router Architecture

- input-buffered routers
  - FIFO scheduling inside
  - if buffer full, signal backpressure upstream
- wormhole switching
  - packets are composed of flits
- stream is a sequence of packets of same size (number of flits)
- round-robin arbitration at each output port (packet based)



State of the Art – NoC Timing Analysis

- **unique priorities and virtual channels:**
  - **Lu2005, Shi2008, Kashif2015**
- **shared channels:**
  - **Shi200** … **5**

→ only for

This work extends Rambo2015 to account for finite sizes buffers and backpressure

- **with backpressure, unique priorities and virtual channels**
  - **Qian2009, Kashif2016, Indrusiak2016**

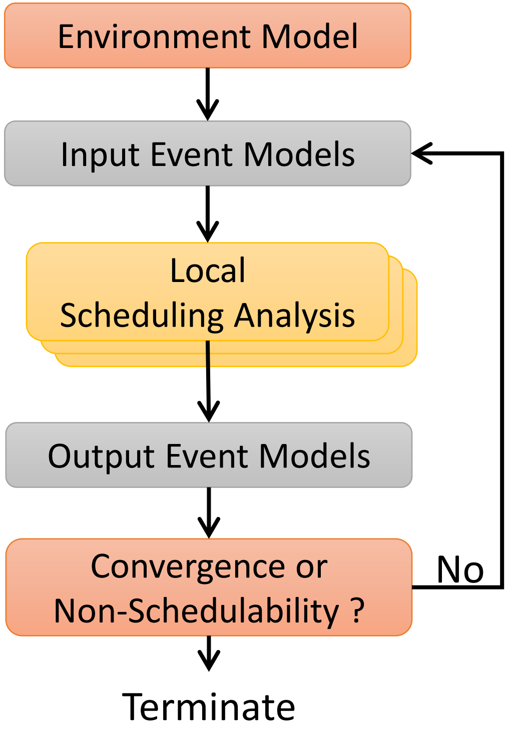
→ no channel sharing, optimistic in some cases

Outline

- Motivation
- **Compositional Performance Analysis (CPA)**
- **Backpressure in CPA**
- **Experiments**
- **Conclusion**

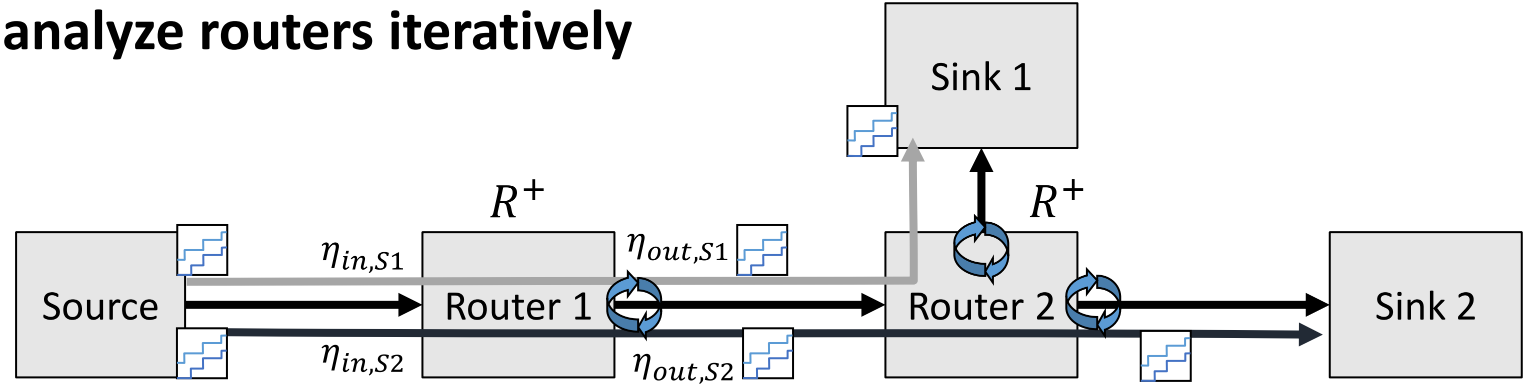
CPA Approach

- analysis performed iteratively
  - step 1: local analysis (at each router)
    - compute worst-case response time of flits based on critical instant (busy window)
    - derive output event models
  - step 2: global analysis
    - propagate event models downstream
    - go to step 1 if any event model has changed
    - otherwise, terminate
- fixed point problem



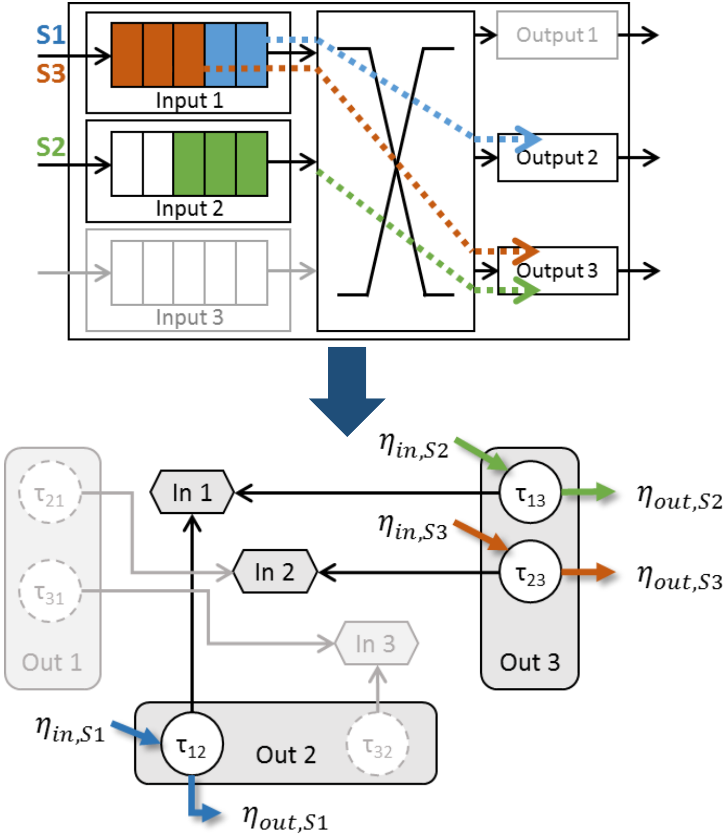
CPA Approach

- worst-case end-to-end latency relies on response times R^+ from local analyses
- for each stream
  - analyze routers along its path and propagate event models downstream
- formally analyze routers iteratively



Mapping NoC Domain to Processor Resource Model

- output ports → processing resources
- input ports → shared resources with mutually exclusive access
- traffic stream → chain of tasks mapped to resources
- flit transmission → task execution
- flit arrival → task activation
  - input and output event models



Outline

- Motivation
- Compositional Performance Analysis (CPA)
- **Backpressure in CPA**
- **Experiments**
- **Conclusion**

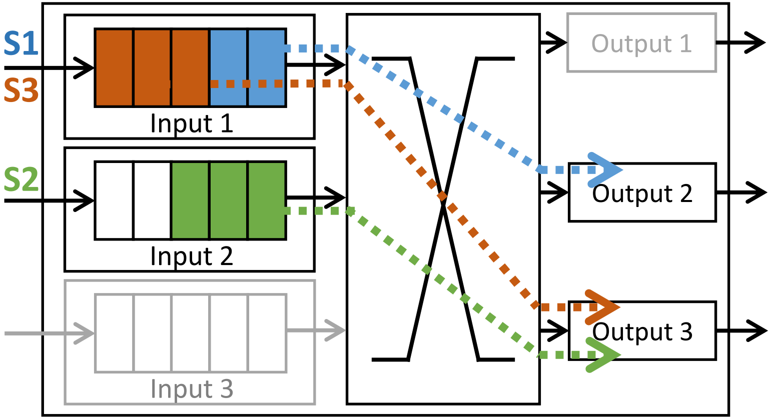
Local Router Analysis

- **worst-case multiple activation processing time for a stream B_i^+**
  - **maximum time resource (router) is busy processing q flits of a stream**
  - **used to derive worst-case latency R_i^+ of a single hop**
- **break down into sum of different terms addressing different blocking factors**
- **doing this for routers without virtual channels, round-robin arbitration and arbitrary but finite input buffer size**

Blocking factors

- **flit transfer**
- **output blocking**
- **FIFO blocking**
- **backpressure blocking**

$$B_i^+(q, a_i^q) \leq \mathbf{q} * \mathbf{C}$$
$$+B_i^{out}(B_i^+(q, a_i^q) - C, q)$$
$$+B_i^{fifo}(B_i^+(q, a_i^q), q, a_i^q)$$
$$+B_{P(i)}^{bp}(q)$$

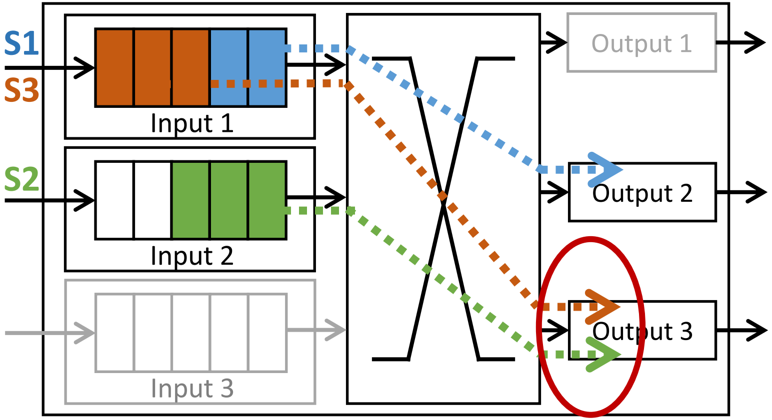


q : number of flits
a_i^q : arrival time of event q
C : single flit transmission time

For details and equations look into the paper

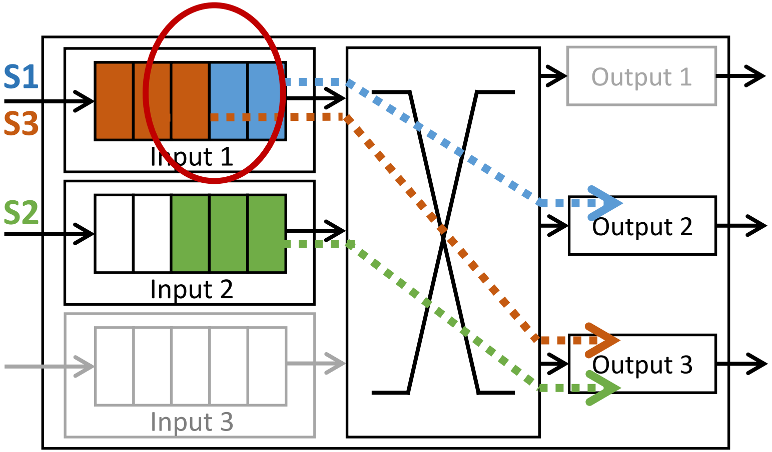
Output blocking

- **interference from other input ports which use the same output port**
- **due to round-robin:**
  - **each other input might send a full packet before each packet of the stream under analysis**
  - **account for their transmission**
  - **and backpressure they might experience**



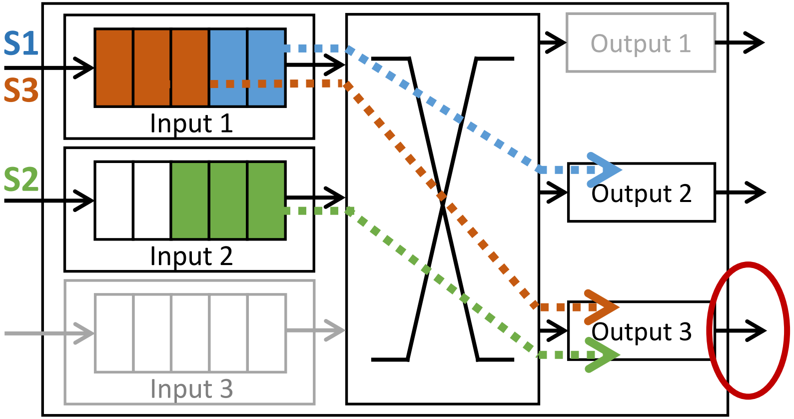
FIFO blocking

- time required to transmit other flits in the FIFO queue preceding the stream under analysis
  - account for their transmission
  - their output blocking
  - backpressure they might experience
- due to limited buffer space:
  - assume the worst candidate to be in the FIFO



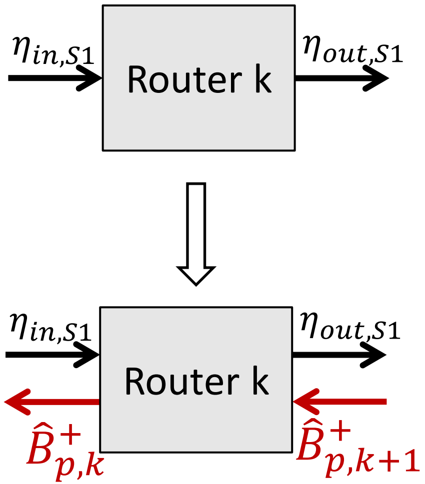
Backpressure blocking

- resulting from lack of free buffer space at the downstream router
  - worst-case waiting time at downstream router until the flits can be received
- if downstream buffer can overflow:
  - wait until enough flits are transmitted (at downstream router)
  - these experience:
    - transmission time, output blocking, backpressure (downstream)



Backpressure blocking

- backpressure depends on downstream router
  - propagate blocking time of a buffer upstream, if backlog can exceed buffer size
    → additional output event model
- influences event model propagation of interfering streams and blocking of task under analysis
- takes part in all other blocking factors
- CPA can already handle upstream propagation
  - but: need to avoid cyclic dependencies
    → be conservative



Network Latency

- derive single hop latency R^+ based on
  - multiple activation busy time
  - router's overhead (e.g. time to determine and acquire output port)
- network latency l^+:
  - sum of single hop latencies on path

    \+ injection time (including backpressure at source)

    \+ de-/packetization overhead

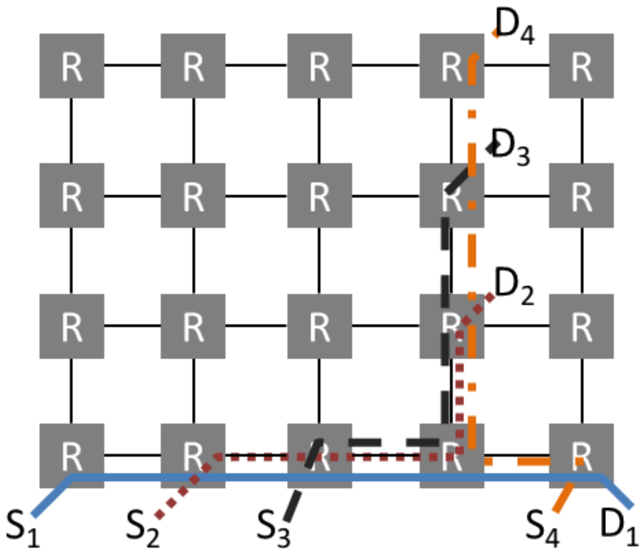
$$l_i^+(q) = InjectionTime(q) + PacketizationOverhead + \sum_{j \in Tasks(i)} R_j^+$$

Outline

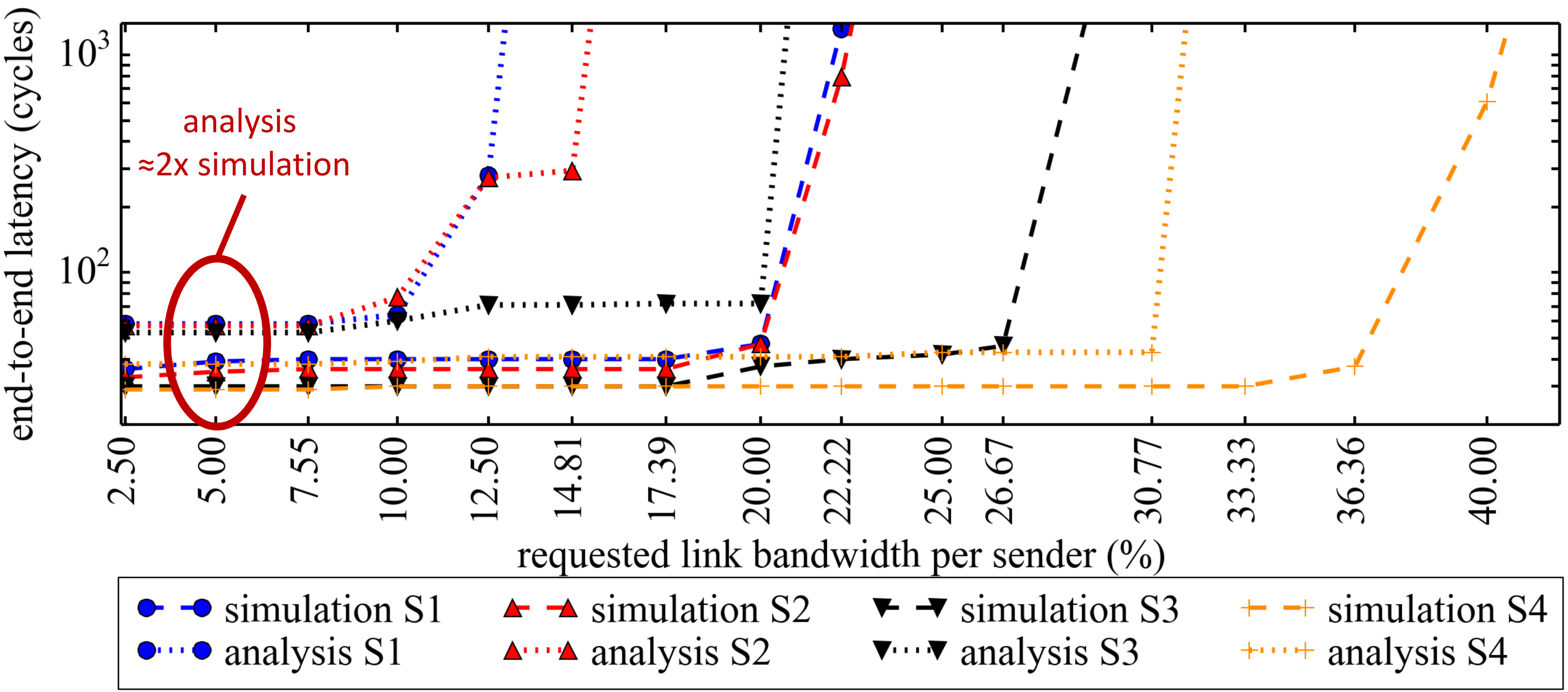
- Motivation
- Compositional Performance Analysis (CPA)
- Backpressure in CPA
- **Experiments**
- **Conclusion**

Experiments

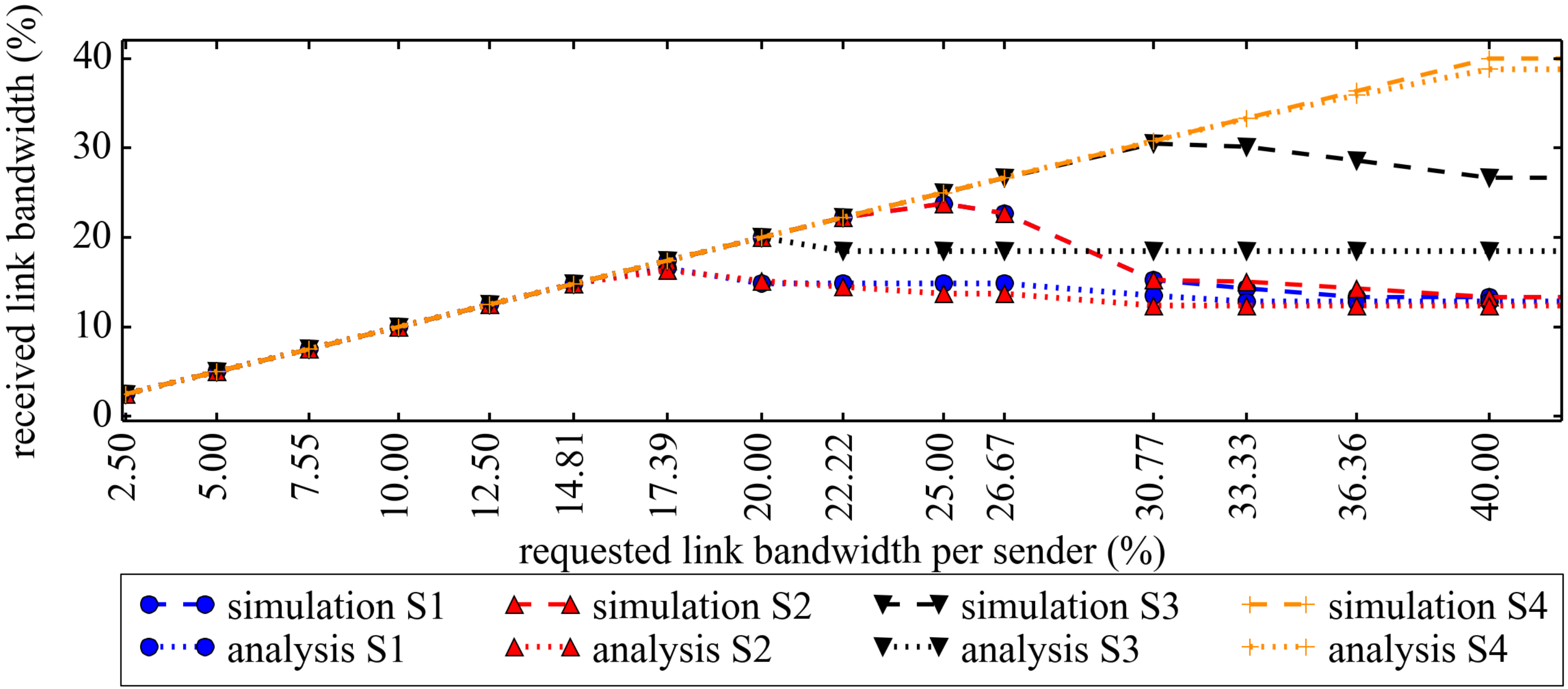
- simulation in OMNeT++
- four streams periodically injecting traffic
  - from S_x to D_x
  - packet size: 4 flits
  - 4 cycle routing overhead
  - buffer size 2 packets / 8 flits
  - injection jitter: 25% of period
- varied requested throughput (decrease of period)



Flit Worst-case Latency

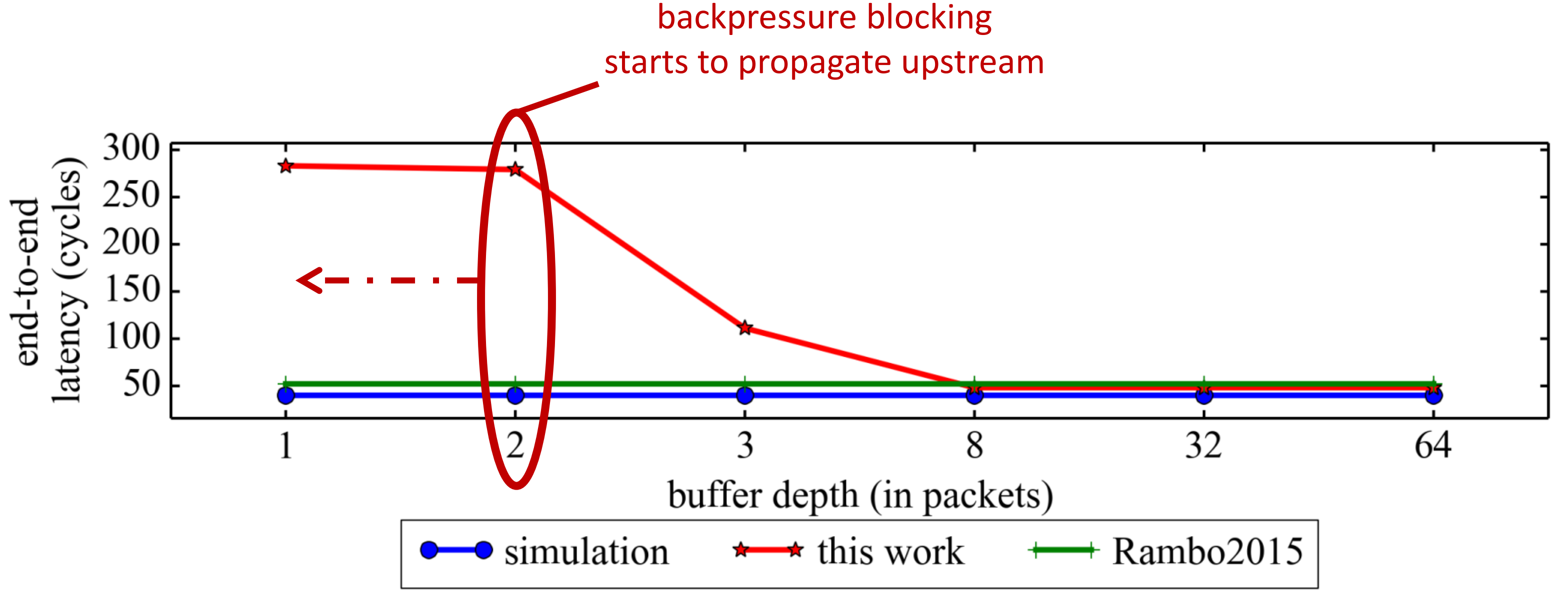


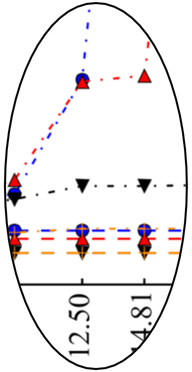
Requested and Received Bandwidth



Flit Latency for Stream S1

12.5% requested bandwidth per sender





Outline

- Motivation
- Compositional Performance Analysis (CPA)
- Backpressure in CPA
- Experiments
- **Conclusion**

Conclusion

- extended CPA to handle backpressure in NoCs
- analysis provides upper bounds on flit level
  - e.g. end-to-end latency, max backlog at each router, received throughput
- with backpressure, analysis results gets more pessimistic
  - load/saturation point can be estimated more accurately than latency
- further improvements needed:
  - correlations between different blocking terms (and routers)
  - correlations between streams

Thank you for your attention.
Questions?

References

- [**Rambo2015**] E. A. Rambo and R. Ernst, “*Worst-case communication time analysis of networks-on-chip with shared virtual channels*,” in Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE ’15, (San Jose, CA, USA), pp. 537–542, EDA Consortium, 2015.
- [**Lu2005**] Z. Lu, A. Jantsch, and I. Sander, “*Feasibility analysis of messages for on-chip networks using wormhole routing*,” in Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, 2005., vol. 2, pp. 960–964 Vol. 2, Jan 2005.
- [**Shi2008**] Z. Shi and A. Burns, “*Real-time communication analysis for on-chip networks with wormhole switching*,” in Networks-on-Chip, 2008. NoCS 2008. Second ACM/IEEE International Symposium on, pp. 161–170, Apr. 2008
- [**Shi2009**] Z. Shi and A. Burns, “*Improvement of schedulability analysis with a priority share policy in on-chip networks*,” in 17th International Conference on Real-Time and Network Systems, pp. 75–84, 2009.
- [**Kashif2015**] H. Kashif, S. Gholamian, and H. Patel, “*SLA: A stage-level latency analysis for real-time communication in a pipelined resource model*,” IEEE Transactions on Computers, vol. 64, pp. 1177–1190, April 2015.
- [**Ferrandiz2012**] T. Ferrandiz, F. Frances, and C. Fraboul, “*A sensitivity analysis of two worst-case delay computation methods for spacewire networks*,” in 2012 24th Euromicro Conference on Real-Time Systems, pp. 47–56, July 2012.
- [**Dasari2014**] D. Dasari, B. Nikoli’c, V. N’elis, and S. M. Petters, “*NoC contention analysis using a branch-and-prune algorithm*,” ACM Trans. Embed. Comput. Syst., vol. 13, pp. 113:1–113:26, Mar. 2014.
- [**Rahmati2009**] D. Rahmati, S. Murali, L. Benini, F. Angiolini, G. D. Micheli, and H. Sarbazi-Azad, “*A method for calculating hard QoS guarantees for networks-on-chip*,” in 2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, pp. 579–586, Nov 2009
- [**Qian2009**] Y. Qian, Z. Lu, and W. Dou, “*Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip*,” in Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on, pp. 44–53, May 2009.
- [**Kashif2016**] H. Kashif and H. Patel, “*Buffer space allocation for real-time priority-aware networks*,” in 2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 1–12, April 2016.
- [**Indrusiak2016**] L. S. Indrusiak, A. Burns, and B. Nikolic, “*Analysis of buffering effects on hard real-time priority-preemptive wormhole networks*,” CoRR, vol. abs/1606.02942, 2016.

Backup

Output Blocking - Comparison

- **our:** $B_i^{out}(\Delta t, q) = \sum_{j \in Out_i} C * \chi + B_{P(i)}^{bp}(\chi)$

- **Rambo2015:** $B_i^{out}(\Delta t, q) = \sum_{j \in Out_i} C * \rho_j^{+}(\Delta t)$

transmission of packets from other inputs

backpressure blocking for packets from other inputs

- **with**
- $\chi = min\left\{\left\lceil \frac{q}{n} \right\rceil * n, \rho_j^{+}(\Delta t)\right\}$
- $\rho_j^{+}(\Delta t) = \left\lceil \frac{\eta_j^{+}(\Delta t)}{n} \right\rceil * n$ **(max number of flits that arrive in Δt assuming whole packets)**
- $\eta_j^{+}(\Delta t)$ **maximum number of flits that arrive in Δt**

FIFO Blocking - Comparison

- **our:**

$$B_i^{fifo}(\Delta t, q, a_i^q) = m * C + \max_{\theta \in \Theta^k}\{A_\theta\} + \max_{\theta \in \Theta^l}\left\{B_{P(\theta)}^{bp}(m - k * n)\right\}$$

transmission, output blocking and backpressure of preceding packets

backpressure blocking for preceding packets

- **Rambo2015:**

$$B_i^{fifo}(a_i^q) = \sum_{j \in Ov_i}\{C * \rho_j^+(a_i^q)\} + \sum_{j \in In_i}\left\{C * \rho_j^+(\Delta t) + B_i^{out}\left(\delta_j^-\left(\rho_j^+(a_i^q)\right) - C\right)\right\}$$

transmission of preceding overlapping streams

transmission and output blocking of preceding packets

- **with:**
- $m = min\{Q_b - 1, \sum_{j \in Buf_i}\{\rho_j^+(a_i^q)\}\}$
- $A_\theta = \sum_{j \in \theta}\left\{B_j^{out}(\Delta t - C, n) + B_{P(j)}^{bp}(n)\right\}$

Backpressure Blocking - Comparison

- $B_p^{bp}(q) = \hat{B}_{p,k+1}^{+}(q)$, max waiting time at downstream router (k+1) for q flits

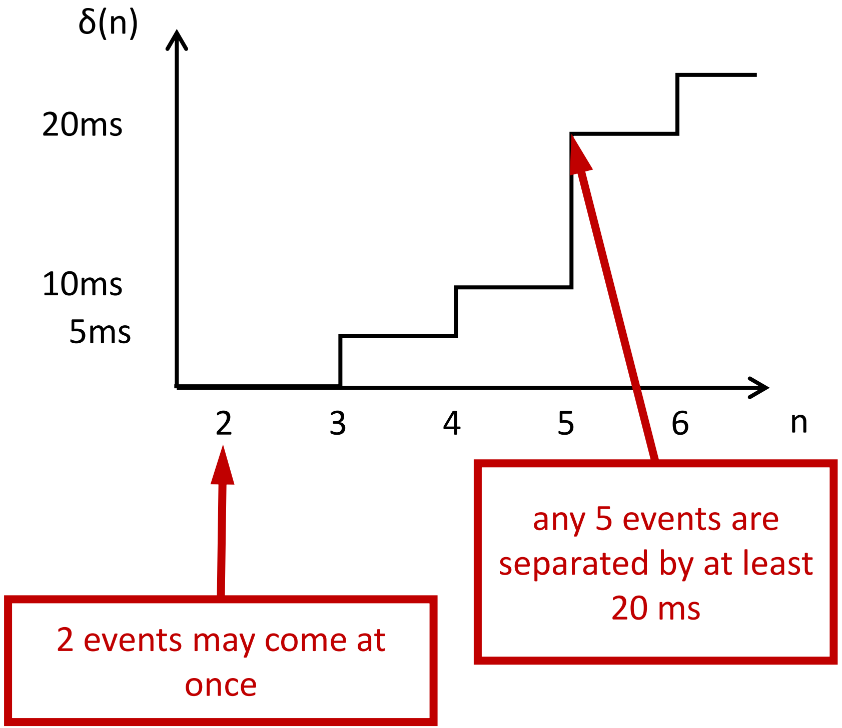
$$\hat{B}_p^{+}(q) = \begin{cases} q * C + \max_{\theta \in \Theta^k}\{A_\theta\}, & if\ b_p > Q_b \\ 0, & otherwise \end{cases}$$

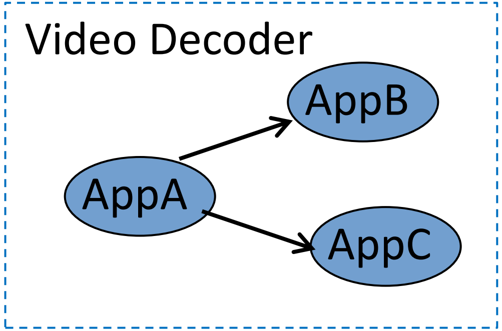
transmission, output blocking and backpressure of flits at port p

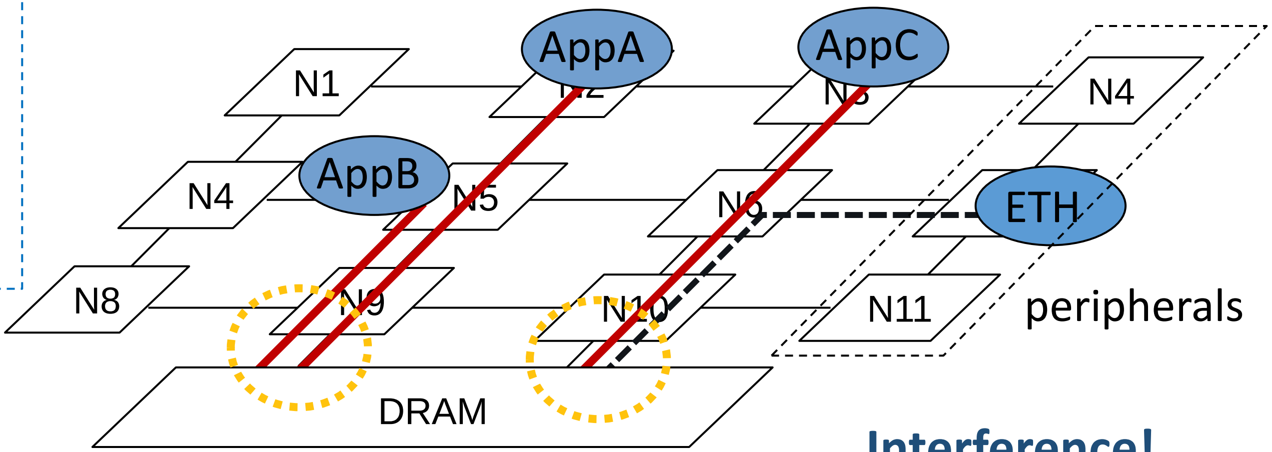
- with
- $A_\theta = \sum_{j \in \theta}\left\{B_j^{out}\left(\hat{B}_p^{+}(q) - C, n\right) + B_{P(j)}^{bp}(n)\right\}$
- b_p as worst-case backlog of port p, Q_b as the buffer size, n packet size in flits
- $k = \left\lceil \frac{q}{n} \right\rceil$, max number of packets q flits form

Complex Activation Patterns

- **variety of activation patterns used in practice**
  e.g. periodic + spontaneous, dual cyclic, on change
- **timing verification can consider them through use of minimum distance functions**
  - **i.e. specification of the minimum distance between any n consecutive events**
  - **derived from specification or rate-limiter**

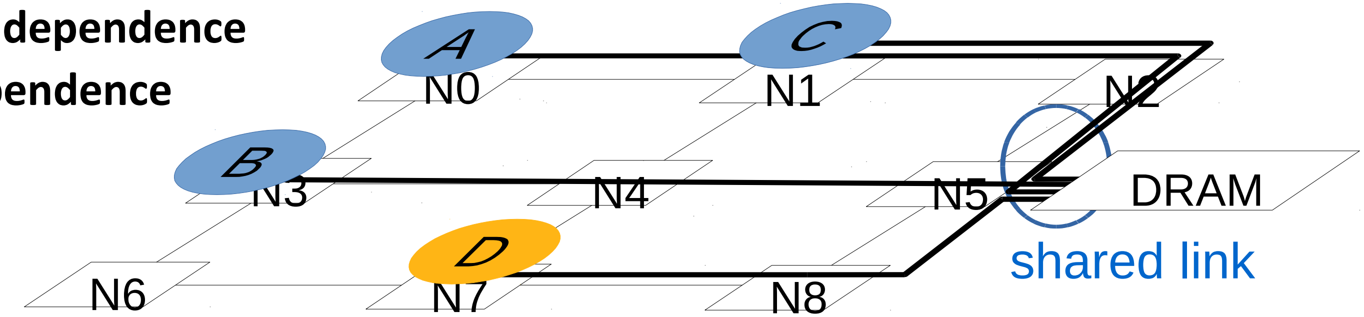






Interference!

- **Networks-on-Chip (NoCs) offer a high-performance, scalability and flexibility**
- **allow integration of many components**
- **result → different transmissions share NoC resources**
  - **links and buffers**
- **standards require separation in case of shared resources**
  - **functional independence**
  - **timing independence**

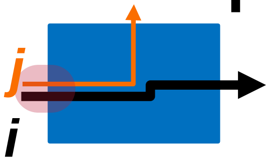


Main Challenge → QoS guarantees + high performance

Rambo2015 - Calculating the Interference

j is in a **different VC** | *j* is in the **same VC**

- **Direct Input**



j is in a different VC:

$$B_i^{DI}(\Delta t, q) = \sum_{c\in V(i)} C \cdot \min\left\{q, \sum_{j\in In_c^i} \tilde{\eta}_j^+(\Delta t)\right\}$$

Round-Robin

j is in the same VC:

$$B_i^{DIvc}(a_i^q) = \sum_{j\in In^i_{v(i)}} \left\{\rho_j^+(a_i^q)\cdot C + B_j^{DO}(\delta_j^-(\rho_j^+(a_i^q)) - C) + B_j^{IO}(\delta_j^-(\rho_j^+(a_i^q)) - C)\right\}$$

FIFO

Head of line blocking

- **Overlapping**

[Diemer11]

$$w_i^{overlap}(q) \le w_i(q) + \sum_{j\in Overlap(i)} w_j(\eta_j^+(w_i^{overlap}(q) - C))$$

Interference from *j*
+ interference suffered by *j*

Improvement

[Rambo2015]

$$B_i^{OV}(\Delta t) = \sum_{c\in V(i)} \sum_{j\in Overlap_c^i} \eta_j^+(\Delta t)\cdot C$$

Interference from *j*

i: stream under analysis; ***j***, ***k***: interfering stream