Boosting Guaranteed Performance in Wormhole NoCs with Probabilistic Timing Analysis

Mladen Slijepcevic[‡,†], Carles Hernandez[†], Jaume Abella[†], Francisco J. Cazorla[†,*]
[‡]Universitat Politecnica de Catalunya (UPC), Spain
[†]Barcelona Supercomputing Center (BSC), Spain
[*]Spanish National Research Council (IIIA-CSIC), Spain

***Abstract*—Wormhole-based NoCs (wNoCs) are widely accepted in high-performance domains as the most appropriate solution to interconnect an increasing number of cores in the chip. However, wNoCs suitability in the context of critical real-time applications has not been demonstrated yet. In this paper, in the context of probabilistic timing analysis (PTA), we propose a PTA-compatible wNoC design that provides tight time-composable contention bounds. The proposed wNoC design builds on PTA ability to reason in probabilistic terms about hardware events impacting execution time (e.g. wNoC contention), discarding those sequences of events occurring with a negligible low probability. This allows our wNoC design to deliver improved guaranteed performance. Our results show that WCET estimates of applications running on top of probabilistic wNoCs are reduced by 40% and 75% on average for 4x4 and 6x6 wNoC setups respectively when compared against deterministic wNoCs.**

I. Introduction

Wormhole-based NoCs (wNoCs) are deployed in high-performance domains to connect a high number of cores on-chip. However, wNoCs efficient use in the context of critical-related real-time applications – such as those that can be found in aircraft, cars or trains – has not been shown yet. Unlike buses or other existing centralized network architectures, wNoCs perform the arbitration of communication flows in a distributed manner, which severely complicates the derivation of request contention bounds as required in real-time domains. In this line, some works show that, while reliable contention upper bounds can be provided for Commercial off-the-shelf (COTS) wNoCs [21] [19] [18], those bounds are pessimistic, preventing an efficient use of high-performance wNoCs for mixed-criticality real-time embedded systems (RTES).

wNoC bounds are pessimistic because, whenever timing events can lead to the stall of a request, they are assumed to occur systematically, and hence factored in the derived contention bounds. At the NoC level, since many different flows with different criticality levels might potentially contend for different resources, e.g. router ports, timing analysis techniques are forced to make the pessimistic assumption that all contenders will simultaneously request the same resources. A simple and intuitive way to reduce such pessimism consists in getting information about when and where communication flows in the wNoC will occur such that the exact interference that requests experience can be reliably and tightly factored in. Unfortunately, obtaining this low-level information is not only out of the ability (and will) of end users, but it also breaks time composability. The lack of time composability occurs because one task's load on the wNoC affects the worst-case execution time (WCET) estimates of its corunners, with devastating consequences in (incremental) system integration: any change in a task requires reanalyzing all other tasks (i.e. performing regression tests), which ultimately results in prohibitively high integration costs. Even worse, the WCET of a critical task could depend on the accuracy of the information obtained for a lower criticality task.

Measurement-based probabilistic timing analysis (MBPTA) has been proposed recently as an industrially-friendly timing analysis method to derive WCET estimates and proven in bus-based multicore industrial case studies [25]. MBPTA relies on hardware designs which break systematic pathological behavior so that increasingly high contention scenarios occur with decreasing probabilities, thus leading to low probabilistic WCET (pWCET) estimates by discarding execution times with negligible accumulated probability.

In this paper we propose a wNoC design based on a new randomized wormhole router design, which makes the contention in the network have a probabilistic behavior compatible with MBPTA requirements, thus leading to reduced contention bounds. The contributions of this paper can be summarized as follows:

(1) We integrate efficiently random permutation arbitration [6] in wNoCs routers to avoid systematic bad behavior and make them amenable for MBPTA.

(2) We propose a mechanism based on limiting the number of in flight requests in the wNoC. While limiting the number of in flight requests in deterministic wNoCs (i.e. with round-robin arbitration) does not help reducing contention bounds, it helps reducing significantly those bounds in a probabilistic wNoC and thus, improves WCET estimates.

Results obtained with a cycle-accurate simulator confirm that the proposed wNoC achieves tighter bounds than existing wNoCs and thus, enables the derivation of much tighter WCET estimates. We have observed that contention in the wNoC can be reduced significantly and such reduction becomes more significant as the size of the network increases. In particular, we obtain an average reduction of the WCET estimates for EEMBC [20] workloads of 22%, 40% and 75% for networks of 3x3, 4x4 and 6x6 cores, respectively.

II. MBPTA for wNoCs

Probabilistic timing analysis (PTA) resorts on having platforms on which the execution time of applications can be modelled with true probabilities. Note that probabilities differ from frequencies. While frequencies provide information about past

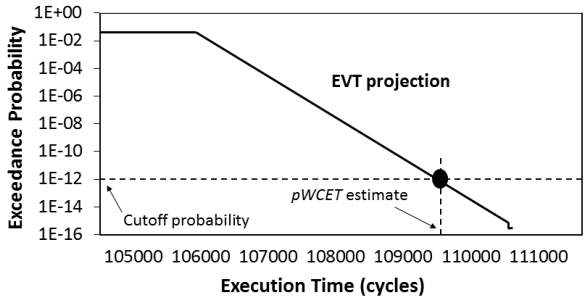


Fig. 1: EVT projection (i.e. probabilistic WCET)

events, probabilities allow reasoning about the future and thus, make predictions. In particular, we focus on the measurement-based variant of PTA (MBPTA), since the measurement-based timing analysis has been shown to be closer to industrial practice in many systems [14], [10], [16]. In this section we review some of the key elements of MBPTA for its reliable application for WCET estimation.

A. MBPTA Application Process

MBPTA relies on collecting a number of execution time measurements – typically in the order of few hundreds – of the *program under analysis* on top of a MBPTA-compliant hardware/software platform [8]. The fact that some variability in execution times occurs does not bring the probabilistic behavior needed by PTA unless such variability is strictly caused by random events.

Execution time measurements need to fulfill requirements such as the following: (1) the upper tail of execution time distributions can be modelled with an exponential distribution and (2) the collected execution time sample needs to attain statistical independence and identical distribution. Both requirements are assessed empirically for the sample of execution times used for prediction with appropriate statistical tests [3]. Some authors point out that exponential tails may be optimistic in some scenarios where either measurements from multiple paths are placed in a single sample or programs with unbounded execution time are analyzed [13]. In our case, as shown in [15], programs have finite execution time and paths are analyzed separately since otherwise the application of MBPTA could be unreliable.

Once those tests are passed, execution time measurements are used as input for Extreme Value Theory (EVT) [9], which is a powerful statistical method to approximate the tail of a distribution. In the case of MBPTA, the tail of the distribution corresponds to high execution times. This results in a probabilistic WCET (pWCET) associated with the probability that one run of the program exceeds a particular time value (see the example in Figure 1). The particular cutoff probability is chosen to be low enough so that it can be regarded as residual risk, in line with safety standards requirements [5].

B. Requirements on the wNoC

MBPTA application requires the sources of jitter (execution time variation) to be properly controlled so that they match or upper-bound operation time conditions in either a deterministic or a probabilistic way. As discussed before, probabilistic modelling allows discarding contention scenarios that occur with negligible probability. This is, for instance, the case of contention in the wNoC. If arbitration decisions are deterministic, the worst-case contention scenario could occur systematically. Instead, if those decisions are randomized, worst contention scenarios occur with (provable) low probability even if time composability is enforced by assuming that all contenders send requests at the maximum possible rate to the worst possible target node. Thus, our approach differs from Network Calculus, since the latter builds on knowledge about contender traffic. Instead, our approach assumes worst-case traffic and enables fully time-composable WCET bounds. Next, we describe the conditions under which timing measurements have to be collected. In Section III we describe how to randomize wNoC timing behavior.

C. Upper-bounding Contention in Probabilistic wNoCs

To be able to reliably apply MBPTA, we have to ensure that measurements for the task under *analysis* are collected under contention conditions that upper-bound those that can occur during *operation* [8]. Failing to do so prevents EVT from actually capturing unobserved contention effects into the pWCET, which could therefore be optimistic. For instance, measurements collected under contention-free conditions lead to unreliable pWCET estimates since EVT cannot reason about the events (contention in the wNoC) not captured in those contention-free measurements.

Such upper-bounding can be performed deterministically or probabilistically. Upper-bounding latency deterministically only requires forcing all wNoC requests to experience the worst-possible delay [19]. To illustrate probabilistic upper-bounding, let us assume a hardware resource whose analysis-time latency can be 1 or 2 cycles with the same probability: $etd_a =< (1,2),(0.5,0.5) >$ where the first vector corresponds to the different latencies and the second to their associated probabilities. If during *operation* its execution time distribution is $etd_o =< (1,2),(0.6,0.4) >$, then etd_a probabilistically upper-bounds etd_o since the exceedance probability for any value is higher at *analysis* than during *operation* (e.g., latency of 2 cycles is exceeded with probability 0.4 during *operation* and 0.5 at *analysis*).

In a wNoC setup with all-to-all traffic the worst contention situation can be reproduced by considering that the flows contending for the resources with the flow of the core under analysis (F_i) are all *worst-possible destination flows* [19]. The destination of contenders' flows is chosen such that it causes the worst contention to F_i packets (i.e. it prevents packets of F_i from crossing each hop for the longest possible time). In the wNoC setup considered in this paper, with XY routing, the worst destination of contenders flow corresponds to the farthest node that can be reached from the next F_i hop's input port[1], depending on the traversing direction.

[1]This assumption is not always valid and depends on the number of ports potentially contending with F_i at the different routers along the path. In general the worst possible destination is computed iterating contending flows to the possible destination and selecting the one causing the highest contention.

Probabilistic upper-bounding allows WCET to have a time-composable behavior, i.e. independent of the actual traffic generated by contending applications.

III. Probabilistic wNoC Designs

Unlike deterministic wNoCs, probabilistic network designs do not require the timing analysis to consider that all accesses systematically experience their worst possible contention. Therefore, the probabilistic analysis made by MBPTA arises as a suitable approach to reduce the pessimism factored in the contention in wNoCs. To enable the derivation of pWCET estimates with MBPTA, two conditions must hold in the wNoC design: (i) conflicts in the wNoC must have a probabilistic nature (i.e. should occur with a given probability); and (ii) the execution conditions (contention) under which the timing measurements of the application are collected at *analysis* are actually an upper bound of those that will occur during *operation*. Condition (i) requires modifications in the arbitration unit of the router (Section III-A) and condition (ii) requires defining a contention scenario which safely upper-bounds the worst possible one (Section II-C). In this section we present how a COTS wNoC must be adapted to enable the derivation of tight pWCET estimates with MBPTA.

A. MBPTA-compliant wNoC Router Design

To make a wNoC design MBPTA-compliant, we have to make packet jitter follow a probabilistic behavior (under maximum contention). To do so, hardware changes are required in the arbitration unit of the NoC router. From the different MBPTA-friendly arbitration policies, we choose random permutations as it delivers superior performance and bounded contention [6]. Random permutations grant access to N contenders in a round-robin fashion, but in a random order. Such order changes every N arbitrations, so that each contender is granted access once every N slots, but in a random order.

To implement random permutations in the wNoC router, we modify the arbiter to be able to generate a random permutation P_i of all four inputs for every output port, where the four inputs and the output port belong to the group $(X+, X-, Y+, Y-, In)$. Whenever one or more packets request access to a given output port, the arbiter grants access according to P_i and an arbitration pointer. When a permutation is generated the arbitration pointer points to the first input port in the permutation. If the first input in the permutation is not requesting the output port then the next input port in the permutation is selected. This process is repeated until an input port with a pending request is selected. Then the arbitration pointer is moved to the input port in the permutation after the one granted access. When the pointer reaches the end of the permutation, a new random permutation window is generated.

This router is the basic component on top of which the later proposed probabilistic wNoC design in this paper relies on. The random arbitration performed in this router allows introducing delays in the measurements to probabilistically represent the contention in the wNoC. While arbitration decisions are random, they carry out dependences across arbiters since the actual requests contending for a given output port in a router often depend on the (random) decisions taken in other routers. Any state of the wNoC in terms of contention moves to any other state with a given probability due to the purely random nature of all arbitration choices. Therefore, each sequence of states occurring during the execution of the task under analysis occurs with a given probability and hence, each potential execution time has a true probability to occur, as needed to apply MBPTA.

B. Reducing Contention in Probabilistic wNoCs

By combining worst-contention scenarios with the probabilistic router architecture proposed in section III-A we can produce execution conditions during *analysis* that upper-bound those during *operation*. Execution time measurements collected during the *analysis* phase can be used reliably to apply MBPTA in order to derive WCET estimates. However, if we do not anyhow limit the contention in the network, the stalls experienced by the requests of the task under analysis can be very high and thus, WCET estimates will account for high contention for all requests, similarly to the case of time-deterministic wNoCs. In time-deterministic wNoCs the worst-case contention with, for instance, round-robin arbitration, is accounted for all requests regardless of the degree of contention in the network. In the case of probabilistic wNoCs, requests experience the actual contention of the worst-case scenario modeled at *analysis*, which is enforced not to be exceeded during *operation*. Therefore, decreasing maximum contention by design opens the door to obtaining lower WCET estimates with probabilistic wNoCs, as already shown for tree wNoCs [23].

In order to decrease contention and derive tighter WCET estimates in the wNoC, we propose a mechanism consisting of limiting the number of in-flight requests (**LNR**). It is important to mention that reducing contention by reducing the number of requests in the network is suitable for probabilistic wNoCs because, in such designs, requests interleave probabilistically and therefore, worst-case alignment of flows and arbitration decisions do not need to be accounted for systematically (as opposed to the case of time-deterministic wNoCs). In other words, already proposed techniques for reducing contention in time-deterministic wNoCs, such as injection throttling [24], cannot obtain tighter WCET estimates [19].

Contention in the network can be reduced by limiting the number of requests in-flight for all the nodes in the network. With our proposed MBPTA-compliant router design, we remove the need to know the exact alignment and we only need to ensure that, during the *analysis* phase, the task under analysis can have *up to* n requests in-flight and all the other cores have *always* exactly n requests in flight. In this case, execution times obtained for the task under analysis are obtained under worst-case contention conditions.

Since we do not assume any specific pattern and maximum contention is enforced during *analysis*, time-composability is preserved. During *operation* all cores can inject requests with the same restrictions imposed during analysis: at most n requests in flight per core.

IV. Evaluation

A. Methodology

Target Processor Architecture. We model a wNoC-based manycore processor with pipelined in-order cores[2] with a simulator based on the SoCLib simulation framework [2]. Each core has separated first level instruction (I1) and write-through data (D1) caches, a partitioned-across-cores write-back L2 cache and main memory. I1 and D1 are 16KB, 4-way and 16B/line and the L2 has 128KB 4-way per core to discount L2 cache effects from the analysis. All caches implement random placement and replacement policies [7]. Hit/miss latencies are 1 and 3 cycles for I1/D1 and 2 and 7 cycles for L2.

wNoC. We model the wNoC with an enhanced version of the gNoCsim [1] simulator, that has been integrated with the SoCLib framework. Cores and memories are connected using a mesh network topology with XY routing. For a NxN mesh we index routers from R(0,0) to R(N-1,N-1). The shared L2 cache memory and a shared memory controller are connected at router R(N-1,N-1). The memory controller implements random permutation policy. Two virtual networks are used to split requests and responses. Routers are pipelined and consist of 4 stages: input buffer, routing, switch allocation, and crossbar traversal. In line with other works [19], [18] in all wNoC setups we use single-flit packets only to improve performance guarantees. The number of VCs is 1. Additional virtual channels would not provide higher guaranteed performance in our setup, as discussed in [19].

Authors in [19] showed that measuring inter-task interferences in a wNoC using the Worst Contention Delay ($\overline{WCD}$) metric results in much tighter WCET estimates than using Worst-Case Traversal Time and it allows obtaining time-composable WCET estimates. Therefore, we compare our probabilistic wNoC design with that proposal. The rest of the platform features are kept identical in both setups, deterministic and probabilistic ones, to understand the differences in the guaranteed performance provided by the wNoC since both setups are compatible with MBPTA.

Workload. As representative real-time workloads we use the EEMBC Autobench suite [20], which reflects current real-world demand of some automotive critical real-time embedded systems.

WCET estimation. We follow MBPTA process to obtain pWCET estimates [3]. For each task we collect 1000 runs and present pWCET estimates for a cutoff probability of 10^{-13} per run, although the same trends are obtained for other cutoff probabilities.

B. Performance Evaluation

For evaluating the performance guarantees of our probabilistic wNoC setup, we use the EEMBC single-threaded workloads as the task under analysis. In these experiments, the task under analysis is placed at the node attached to $R(0,0)$, which is the one experiencing highest contention, and the rest of the cores are forced to cause worst-possible contention as mentioned in Section III. Figure 2 shows the pWCET estimates achieved by limiting the number of in-flight requests (**LNR**) for the different benchmarks. Results are normalized w.r.t. the case of the deterministic wNoC. All the existing task communications target the shared memory controller that is attached to $R(2,2)$ in the 3x3 case and to $R(3,3)$ and $R(5,5)$ in the 4x4 and 6x6 case[3]. Other possible task placements provide similar comparative results. As shown in Figure 2, **LNR** wNoC setup outperforms the performance guarantees achieved by the deterministic wNoC design. In particular, average WCET reductions are 22%, 40% and 75% for 3x3, 4x4 and 6x6 setups respectively (6x6 results are not depicted due to space constraints).

It was already shown in [19] that limiting the injection of the flow under analysis, has no impact on $\overline{WCD}$ since this only affects intra-task contention and not the contention due to inter-task interferences. The reason is that time-composable WCET estimates require considering the worst possible interleaving of requests in the wNoC and this causes, in general, worst-case scenarios to be also possible when allowing only one in-flight request per flow.

We have discovered that contention in a probabilistic wNoC setup follows a probabilistic distribution which is centered around $\overline{WCD}$ (the worst contention in a deterministic wNoC setup). The shape and location of the distribution is almost identical regardless of the number of in-flight requests. Then it might not seem obvious the reason why **LNR** behaves better than a deterministic setup. For a deterministic approach, where a particular alignment of requests cannot be assumed, it needs to be considered systematically that requests will be absorbed at a constant rate that is equal to $\overline{WCD}$. On the contrary, in a probabilistic approach roughly 50% of the requests will get absorbed faster than $\overline{WCD}$. These fast requests make possible to take advantage of the store buffer of the pipeline more frequently than for a deterministic approach and allow a higher overlapping between computation and communication, thus leading to smaller execution time[4]. In particular, the behavior of the deterministic wNoC is a *fill-and-stall* behavior of the store buffer in front of store bursts, thus stopping pipeline progress always. Conversely, the probabilistic wNoC allows releasing store buffer entries often earlier due to lower contention delay, and for the time a new store arrives at the store buffer, there is space available so that the pipeline keeps progressing in parallel with the processing of stores in the wNoC.

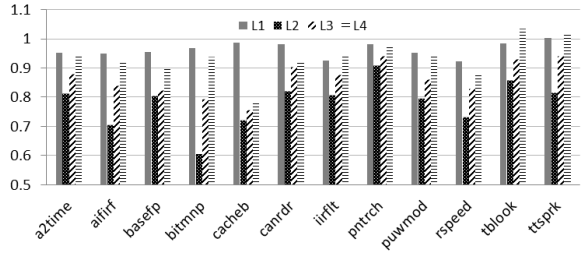
V. Related work

Some NoCs, whose complexity may vary, have been designed specifically to meet real-time constraints [4], [22], but they differ significantly from COTS wNoCs, thus challenging reconciling their different objectives: average performance vs time-composable WCET estimates.

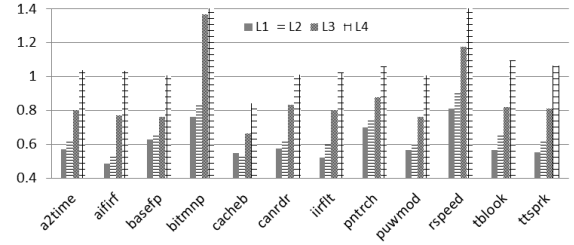
[2]Note that more complex processor cores, although compatible with our proposals, are very hard to time-analyze and thus, not suitable for hard-real time applications.

[3]Although our approaches scale smoothly regardless the core count, we do not consider larger manycores due to the increasingly poor scaling of deterministic wNoCs for larger core counts.

[4]This holds for timing-anomaly-free processors like the one used in our experiments.



(a) 3x3



(b) 4x4

Fig. 2: LNR pWCET estimates normalised w.r.t. a deterministic wNoC (Ln means up to n requests in-flight allowed).

Works based on Network Calculus [11] abstract communication flows using arrival curves that upper-bound the amount of traffic within any time interval, thus sacrificing time-composability to obtain tighter WCET bounds. Network Calculus is appropriate when traffic information available during analysis is accurate, which may be for off-chip traffic, but is generally unaffordable for on-chip traffic.

Another set of works focuses on determining wNoC packets worst-case traversal time (WCTT) by considering worst-case conditions [12], [21]. Authors in [19] show that measuring inter-task interferences in a wNoC using the Worst Contention Delay ($\overline{WCD}$) metric results in much tighter WCET estimates than using WCTT.

MBPTA compliance has been achieved for bus designs with appropriate randomized arbitration policies [6]. TDMA-based buses have also been proven amenable for MBPTA by padding execution time measurements conveniently [17]. A tree NoC implementing wormhole routers with random arbitration and intended for all-to-one communication has also been shown to be amenable for MBPTA [23]. However, trees do not fit well all-to-all communication.

Differently to those works, we propose minor changes to COTS wNoC mesh designs to enable reliable, tight and time-composable WCET estimates. In particular, we build upon MBPTA and the use of random arbitration to discard pathological cases with negligible probability, thus outperforming deterministic wNoCs similar to COTS ones.

VI. Conclusions

In this paper we show that a probabilistic approach is highly efficient dealing with contention in wNoCs. Pathological worst-contention scenarios occur with (provable) negligible probability and hence, there is no need to account for them. We propose a wNoC setup, **LNR**, able to provide better performance guarantees than deterministic approaches by making use of a wormhole router with randomized arbitration. **LNR** is particularly suitable for applications that are very sensitive to latency and its gains w.r.t. deterministic setups increase in pace with the wNoC size.

Acknowledgments

This work has also been partially supported by the Spanish Ministry of Science and Innovation under grant TIN2015-65316-P and the HiPEAC Network of Excellence. Mladen Slijepcevic is funded by the *Obra Social Fundación la Caixa* under grant Doctorado "la Caixa" - Severo Ochoa. Carles Hernández is jointly funded by the Spanish Ministry of Economy and Competitiveness (MINECO) and FEDER funds through grant TIN2014-60404-JIN. Jaume Abella has been partially supported by the MINECO under Ramon y Cajal postdoctoral fellowship number RYC-2013-14717.

References

[1] *NanoC: NaNoC design platform. http://www.nanoc-project.eu*.
[2] Soclib, http://www.soclib.fr/trac/dev, 2012.
[3] J. Abella et al. Measurement-based worst-case execution time estimation using the coefficient of variation. *ACM Trans. Des. Autom. Electron. Syst.*, 22(4), June 2017.
[4] K. Goossens, et. al. Aethereal network on chip: concepts, architectures, and implementations. *Design Test of Computers, IEEE*, 2005.
[5] International Organization for Standardization. *ISO/DIS 26262. Road Vehicles – Functional Safety*, 2009.
[6] J. Jalle et al. Bus designs for time-probabilistic multicore processors. In *DATE*, 2014.
[7] L. Kosmidis et al. A cache design for probabilistically analysable real-time systems. In *DATE*, 2013.
[8] L. Kosmidis et. al. Fitting processor architectures for measurement-based probabilistic timing analysis. *Microprocess. Microsyst.*, 47(PB):287–302, November 2016.
[9] S. Kotz and S. Nadarajah. *Extreme value distributions: theory and applications*. World Scientific, 2000.
[10] S. Law and I. Bate. Achieving appropriate test coverage for reliable measurement-based timing analysis. In *ECRTS*, 2016.
[11] J.-Y. Le Boudec and P. Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag, 2001.
[12] Sunggu Lee. Real-time wormhole channels. *Journal Of Parallel And Distributed Computing*, 63:299–311, 2003.
[13] G. Lima et al. Extreme value theory for estimating task execution time bounds: A careful look. In *ECRTS*, 2016.
[14] E. Mezzetti and T. Vardanega. On the industrial fitness of wcet analysis. In *WCET Workshop*, 2011.
[15] S. Milutinovic et al. On uses of extreme value theory fit for industrial-quality WCET analysis. In *SIES*, 2017.
[16] M. Di Natale, J. Abella, J. Reineke, A. Hamann, and G. Farrall. Predictable system timing – probab(ilistical)ly? In *DAC (panel in automotive track)*, 2016.
[17] M. Panic, et. al. Enabling TDMA arbitration in the context of MBPTA. *DSD*, 2015.
[18] M. Panic, et. al. Improving performance guarantees in wormhole mesh noc designs. In *DATE*, 2016.
[19] M. Panic, et. al. Modeling high-performance wormhole nocs for critical real-time embedded systems. *RTAS*, 2016.
[20] J.A. Poovey et al. A benchmark characterization of the EEMBC benchmark suite. *IEEE Micro*, 29, 2009.
[21] D. Rahmati, et. al. Computing accurate performance bounds for best effort networks-on-chip. *IEEE Transactions on Computers*, 2013.
[22] Z. Shi and A. Burns. Real-time communication analysis for on-chip networks with wormhole switching. In *NoCS*, 2008.
[23] M. Slijepcevic, et. al. pTNoC: Probabilistically time-analyzable tree-based noc for mixed-criticality systems. In *DSD*, 2016.
[24] M. Thottethodi et al. Self-Tuned Congestion Control for Multiprocessor Networks. In *HPCA*, 2001.
[25] F. Wartel et al. Timing analysis of an avionics case study on complex hardware/software platforms. In *DATE*, 2015.