Buffer-aware bounds to multi-point progressive blocking in priority-preemptive NoCs

Leandro Soares Indrusiak*, Alan Burns*, Borislav Nikolić†
*Real-Time Systems Group, Department of Computer Science, University of York, York, UK
†CISTER Research Centre, ISEP/IPP, Porto, Portugal
{leandro.indrusiak, alan.burns}@york.ac.uk, borni@isep.ipp.pt

***Abstract*—This paper aims to reduce the pessimism of the analysis of the multi-point progressive blocking (MPB) problem in real-time priority-preemptive wormhole networks-on-chip. It shows that the amount of buffering on each network node can influence the worst-case interference that packets can suffer along their routes, and it proposes a novel analytical model that can quantify such interference as a function of the buffer size. It shows that, perhaps counter-intuitively, smaller buffers can result in lower upper-bounds on interference and thus improved schedulability. Didactic examples and large-scale experiments provide evidence of the strength of the proposed approach.**

I. Introduction

Networks-on-chip (NoCs) with priority-preemptive arbitration have been widely studied for their ability to provide hard real-time guarantees [2], [11], [10] and support for mixed-criticality traffic [3]. Such guarantees are based on analytical models that are able to show that, even in the worst-case scenario, packet latencies will not exceed their deadlines.

Over the years, many analytical models of increasing complexity have attempted to calculate upper-bounds to the latency of packets injected in such a NoC [11], [7]. Those models make assumptions about the traffic generated by the real-time applications running on the NoC (e.g. bounds on packet inter-arrival interval, jitter, size) as well as the NoC itself (e.g. deterministic routing). As the state-of-the-art advances, the assumptions behind each analytical model become more realistic. The most recent development in this area was the identification of the multi-point progressive blocking (MPB) problem by Xiong *et al.* in [12]. Their observation has shown that an assumption made by all previous analyses, namely that each flit of a packet can cause interference on another packet at most once, was not valid. In Section III we look into that problem in further detail, showing that most analytical models produce optimistic latency upper-bounds in MPB scenarios, except for the analysis reported in Xiong *et al.* in [13]. We then show in Section IV that their analysis is unnecessarily pessimistic, and propose a novel approach that reduces significantly the pessimism while still producing safe upper-bounds even in the case of MPB. The paper is closed with extensive experimental work with realistic and synthetically-generated benchmarks, aiming to show the reduced pessimism of the proposed approach.

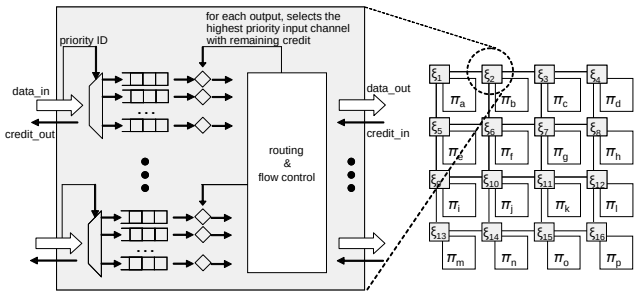

Fig. 1: Wormhole on-chip network with 2D mesh topology and detail of a router with priority-driven virtual channels

II. System Model

Figure 1 shows some details of the internal structure of a router in a priority-preemptive NoC. It follows the architectural templates first presented in [2], where each router includes a flow controller based on priority-preemptive virtual channels (VCs). By assigning priorities to packets, and by allowing high priority packets to preempt the transfer of low priority ones, network contention scenarios become more predictable and an upper bound to the packet latency can be found. In each input port, a different FIFO buffer stores flits of packets arriving through different virtual channels (one for each priority level). The router assigns an output port for each incoming packet according to their destination. A credit-based approach [1] guarantees that a router only forwards data to the next when there is enough buffer space to hold it in the downstream router. At any time, a flit of a given packet will be sent through its respective output port if it has the highest priority among the packets routed to that port, and if it has at least one credit. If the highest priority packet cannot send data because it is blocked elsewhere in the network and its buffers are full (i.e. no credit), the next highest priority packet can access the output link.

Let us model such a network as a set of nodes $\Pi = \{\pi_a, \pi_b, \ldots, \pi_z\}$, a set of routers $\Xi = \{\xi_1, \xi_2, \ldots, \xi_m\}$, and a set of unidirectional links $\Lambda = \{\lambda_{a1}, \lambda_{1a}, \lambda_{12}, \lambda_{21}, \ldots, \lambda_{zm}, \lambda_{mz}\}$. The function $vc(\xi_i)$ denotes the number of VCs supported by router ξ_i, which in this model also means the number of priority levels it is able to distinguish. The function $buf(\xi_i)$ denotes the FIFO buffer size implementing a single VC of that router. A network router is able to transmit flits over its links at a fixed rate. The amount of time taken by a router ξ_i to transmit a flit over any of its links is represented by the link latency

function $linkl(\xi_i)$. The routing of a packet header flit by a router, i.e. the routing logic to decide which of its output ports should arbitrate and transmit the flits of the input VC of that packet, also introduces a latency which is likewise represented by the routing latency function $routl(\xi_i)$. In the case of a homogeneous network, i.e. all the routers are identical, all the functions defined over a specific router (e.g. $buf(\xi_i)$, $routl(\xi_i)$) are also defined over the complete set (i.e. $buf(\Xi)$, $routl(\Xi)$) with the same meaning.

The route between any two nodes of the network is given by the function $route(\pi_a, \pi_b) = \{\lambda_a, \ldots, \lambda_b\}$, denoting the totally ordered subset of Λ used to transfer packets from node π_a to node π_b (including the links connecting a node to its respective router). The number of links of a route is given by $|route_i|$. We then define the function $order_{a,i}(\lambda_a, route_i)$ to denote the order of a link λ_a over a route $route_i$ (i.e. 1 for first, 2 for second, etc.), and the respective convenience functions $first(route_i)$ and $last(route_i)$ to single out the first and last links of $route_i$.

To model the traffic load injected to the network, we define a set Γ of n real-time traffic-flows (or just *flows* for short) $\Gamma = \{\tau_1, \tau_2, \ldots \tau_n\}$. Each flow τ_i gives rise to a potentially unbounded sequence of *packets*. A flow has a set of properties and timing requirements which are characterised by a set of attributes: τ_i = (P_i, C_i, T_i, D_i, J_i, π_i^s, π_i^d). All the flows which require timely delivery are either periodic or sporadic. The lower-bound interval on the time between releases of successive packets is called the period (T_i) for the flow. Each real-time flow also has a relative deadline (D_i) which is the upper-bound restriction on network latency, assumed to be $D_i \leq T_i$ (so that the possibility of interference between packets of the same flow can be dismissed). Any flow can suffer a release jitter J_i, which denotes the maximum deviation of successive packet releases from the flow's period. That is, a packet from τ_i will be released for transmission at most J_i time units after its periodic tick, e.g. due to the time it takes for its source node to generate it. Each flow also has a priority P_i; the value 1 denotes the highest priority and larger integers denote lower priorities. It also has source and destination nodes on the network (π_i^s and π_i^d). Considering the routes of any two packet flows τ_i and τ_j, we define a contention domain $cd_{i,j}$ as the ordered set of links shared by those flows: $cd_{i,j} = route_i \cap route_j$. We assume that a contention domain will never be a disjoint set of links, which is the case in all NoCs with dimension-order routing (e.g. XY).

The maximum zero-load network latency (C_i) is the maximum latency experienced by a packet of that flow, between the release of its first flit to the reception of its last, when no flow contention exists over the network. This value is a function of the maximum number of flits L_i of the packet, and the length of its route. For convenience, we extend the notation of the function $route$ to also represent the route of a packet from its source node to its destination: $route(\tau_i) = route_i = route(\pi_i^s, \pi_i^d)$. We can then formulate C_i as follows:

$$C_i = routl(\Xi) \cdot (|route_i| - 1) + linkl(\Xi) \cdot |route_i| + linkl(\Xi) \cdot (L_i - 1) \quad (1)$$

Equation 1 shows that C_i is equal to the zero-load latency of the header flit plus one additional link latency cycle per payload flit (since they follow the header in a pipeline fashion). The zero-load latency of the header is the time $routl(\Xi)$ it takes to be routed at each hop (in a route with $|route_i| - 1$ routers, since the hop count includes the links connecting nodes to their respective routers) plus the time $linkl(\Xi)$ it takes to cross each of the $|route_i|$ links along its way. Once the header reaches the destination, the payload of the packet takes one additional link latency time $linkl(\Xi)$ for each of its $L_i - 1$ flits. Other formulations of C_i are also possible [12], but that does not impact the approach presented here.

The goal of all the approaches reviewed in Section III, and of the one we propose, is to use (part of) the model presented above to calculate the worst-case latency R_i for each flow $\tau_i \in \Gamma$. R_i is the highest latency experienced by a packet produced by flow τ_i, and takes into account the packet's own zero-load latency plus the worst possible delays resulting from blocking and preemptions from higher priority packets. A system is then said to be schedulable if $R_i \leq D_i$ for every $\tau_i \in \Gamma$.

III. Related Work

In [11], Shi and Burns proposed an analytical model (referred to as SB) that calculates the upper-bound interference suffered by a given traffic flow τ_i considering both direct and indirect interferences from other flows. Following Kim *et al.* [9], they define a direct interference set S_i^D of τ_i as the set of flows that have higher priority than τ_i and that share with it at least one network link (i.e. a non-empty contention domain): $S_i^D = \{\tau_j \in \Gamma \mid P_i < P_j, cd_{i,j} \neq \emptyset\}$. Similarly, the indirect interference set S_i^I of τ_i is the set of flows that are not in S_i^D, but that interfere with at least one flow in that set (i.e. interfere with the flows that interfere with τ_i, but not directly with τ_i itself): $S_i^I = \{\tau_k \in \Gamma \mid \tau_k \in S_j^D, \tau_j \in S_i^D, \tau_k \notin S_i^D\}$. In the case of direct interference, they assume that a packet of τ_i may suffer interference from all packets of every flow $\tau_j \in S_i^D$. The amount of interference on each "hit" of a τ_j packet on τ_i is upper-bounded by C_j, and the number of "hits" is bounded by the number of packets of τ_j appearing during the lifetime of τ_i (which can be found by the ceiling of the ratio between R_i and T_j). Indirect interference is handled as the increased interference a packet from τ_i can suffer from two subsequent packets of a flow $\tau_j \in S_i^D$. This can happen if τ_j itself suffers interference from a flow $\tau_k \in S_i^I$, delaying the first of its packets to the point that it interferes on τ_i right before the second one causes interference (the so-called "back-to-back hit").

Kashif and Patel proposed SLA [7], [8], aiming to reduce the pessimism in SB, i.e. reduce the difference between the upper-bounds provided by the model and the actual worst-case behaviour of the NoC. They did that by calculating interference on a link-by-link basis, and claimed that their

approach will always be tighter and upper-bounded by SB. Experimental results show that their bounds are the same as SB with minimal buffer sizes, and get increasingly tighter in cases with larger buffer storage per VC.

Xiong *et al.* [12] have found a significant shortcoming in both SB and SLA. They have identified using simulations that downstream indirect interference can sometimes cause a single packet of τ_j to directly interfere on τ_i by more than its basic latency C_j, disproving one of the assumptions made by those models. Specifically, they stated that a flit of a packet of τ_j may interfere multiple times on a packet of τ_i over multiple shared links, in case τ_j (1) suffers interference from a packet τ_k that does not interfere with τ_i and (2) shares links with τ_k downstream from the links it shares with τ_i. This is referred to as multi-point progressive blocking (MPB), and both SB and SLA produce unsafe latency bounds under such scenarios.

To account for the MPB problem, Xiong *et al.* proposed a slightly different partitioning of indirect interference sets. They define the upstream indirect interference set $S_{I_i}^{up_j}$ as the set of flows $\tau_k \in S_i^I$ that interfere with the flows $\tau_j \in S_i^D$ before τ_j interferes with τ_i. Similarly, the downstream indirect interference set $S_{I_i}^{down_j}$ is the set of flows $\tau_k \in S_i^I$ that interfere with the flows $\tau_j \in S_i^D$ after τ_j interferes with τ_i. The notion of "before" and "after" used here refers to whether the contention domain between τ_k and τ_j (i.e. the links they share) appears upstream or downstream in τ_j, in comparison with the contention domain between τ_i and τ_j. For clarity, we review Xiong *et al.*'s definition of those two sets using the notation introduced in Section II:

$$S_{I_i}^{up_j} = \{\tau_k \in S_i^I \cap S_j^D \mid order(last(cd_{jk}), route_j) < order(first(cd_{ij}), route_j)\}$$

$$S_{I_i}^{down_j} = \{\tau_k \in S_i^I \cap S_j^D \mid order(first(cd_{jk}), route_j) > order(last(cd_{ij}), route_j)\}$$

Based on those two sets, Xiong *et al.* defined two worst-case interference terms I_{ji}^{up} and I_{ji}^{down} to denote the worst-case interference I_{kj} suffered by τ_j from flows τ_k that interfere with it, respectively, upstream or downstream from its contention domain with τ_i:

$$I_{ji}^{up} = \sum_{\tau_k \in S_{I_i}^{up_j}} I_{kj} \quad (2) \qquad I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} I_{kj} \quad (3)$$

Their formulation for the worst-case response time R_i bounds upstream indirect interference by I_{ji}^{up} and models downstream indirect interference suffered from every τ_j as direct interference over τ_i (i.e. by adding I_{ji}^{down} to C_j):

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + I_{ji}^{up}}{T_j} \right\rceil (C_j + I_{ji}^{down}) \quad (4)$$

Indrusiak *et al.* [6] show with a counter-example that such formulation is unsafe as the use of I_{ji}^{up} as an interference jitter term in Equation 4 is unable to properly capture all possible upstream indirect interference effects, and thus can produce optimistic results. They also propose a fix to the analysis by using $J_j^I = R_j - C_j$ instead of I_{ji}^{up}, as it was the case in the SB model. A corrected version of the analysis, using the fix proposed in [6], has appeared in [13], which we refer as XLWX and consider to be the current state-of-the-art:

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down}) \quad (5)$$

IV. Proposed Analysis

The key motivation for the approach presented in this paper is the treatment of the MPB problem in the XLWX analysis. While Xiong *et al.* have clearly identified a type of interference that has not been considered in the previous approaches, we argue that their analysis approach does not properly address the indirect interference effects that happen in wormhole networks. Their handling of downstream indirect interference as if it were direct interference is unnecessarily pessimistic, so we aim to provide a tighter analysis by considering more carefully the impact of MPB.

Let us carefully revisit that problem, caused by the downstream indirect interference identified in [12]: a single packet of τ_j can directly interfere on τ_i by more than its basic latency C_j when it suffers interference from any packet τ_k that does not interfere with τ_i, and shares links with τ_k downstream from the links it shares with τ_i. In this situation, every time τ_j is blocked by τ_k, it can allow τ_i to flow through the network and potentially overtake τ_j flits that had already blocked it earlier. XLWX analysis correctly takes into account that the amount of additional interference that τ_i can suffer from τ_j is upper-bounded by the amount of time that τ_i is allowed to overtake τ_j (and subject itself to additional interference), which is in turn upper-bounded by the downstream indirect interference that τ_j can suffer from any τ_k (which is expressed by I_{ji}^{down}, as shown in Equation 3).

Such scenario can be better understood through a simple example with only three flows τ_i, τ_j and τ_k, as shown in Figure 2, aiming to clearly depict the nature of the MPB problem. Assume that τ_i and τ_j have much larger periods and longer packets (therefore larger C) than τ_k, and that τ_k's releases are not in phase with the other two. The priority order has τ_i with the lowest and τ_k with the highest priority. In Figure 2(a), τ_i and τ_j are released at the same time from node *a*, and the higher priority τ_j gains access to the network, blocking τ_i.

In Figure 2(b), a packet of τ_k is then released and interferes with τ_j (downstream from its contention domain c_{ij} with τ_i). Since τ_k has the highest priority, it stops τ_j's flits from using the link between routers *3* and *4*, which generate backpressure on all subsequent flits of that packet of τ_j, forcing them to stay

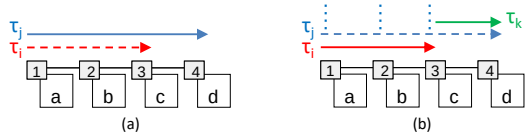

Fig. 2: Downstream indirect interference

buffered along the route (depicted as stacked dots) all the way to the source in node a. Once τ_j flits stop using the links on τ_i's route, τ_i then becomes the highest priority flow with buffer credits so the routers starts transmitting its flits.

When τ_k finishes, the scenario returns to the situation depicted in Figure 2(a), where only τ_j flows through the network. However, before new flits of τ_j can flow out of node a, its buffered flits must first make way and release the backpressure along the route. This is key to the MPB problem: it is those buffered flits of τ_j, which have already caused interference on τ_i when they were first released out of node a, that will again cause interference and as a consequence will delay τ_i by more than τ_j's zero-load latency C_j. We refer to this effect as *buffered interference*, which in turn causes MPB.

Using examples like this one, Xiong *et al.* show in [12] and [13] that SB and SLA analyses do not capture the MPB problem caused by downstream indirect interference, and thus produce optimistic results, while XLWX analysis provides an upper bound in all cases.

By understanding the notion of buffered interference, one can clearly see the intuition behind XLWX analysis, and why its upper bound does not suffer from the same issues as SB and SLA: the interference beyond C_j imposed by τ_j on τ_i will never be larger than the amount of downstream interference that τ_j suffers from τ_k, since that is the maximum amount of interference from τ_j that could be buffered along its way. Thus, by adding the maximum downstream interference I_{ji}^{down} to C_j Xiong *et al.* effectively provides a safe upper-bound to the multiple times τ_j can interfere with τ_i.

We claim, however, that such upper bound is unnecessarily pessimistic, given that the amount of buffered interference will also be upper-bounded by the maximum amount of buffer space along the route of τ_j. Furthermore, we claim that the amount of buffered interference of a single packet of τ_j that can interfere multiple times with τ_i is proportional to the length of their contention domain cd_{ij}. The intuition behind our claims is based on the following observations regarding the behaviour of a τ_j packet which is blocked due to a downstream interference "hit" by τ_k:

- Flits of τ_j stored in buffers of routers that are downstream to the contention domain cd_{ij} will not cause any further interference on τ_i, so they will not contribute to MPB.

- Flits of τ_j stored in buffers within the contention domain cd_{ij} are the only ones that will contribute to MPB.

- If τ_j does not suffer upstream interference, its flits arrive into the contention domain cd_{ij} in a perfect pipelined transmission (i.e. no gaps between flits), and if the packet is long enough it will also be buffered over routers upstream from cd_{ij}. When the downstream interference by τ_k is over, τ_j starts flowing again. If there are flits stored upstream, the amount of supplied flits into the contention domain is equal to the amount of departed flits, so no buffering occurs, or if there was buffering, the amount of buffered flits stays constant. When τ_j is preempted once more by another τ_k "hit" downstream, the build-up of its flits in the contention domain reoccurs. Each downstream hit by an indirectly interfering flow τ_k can cause at most one full contention domain worth of buffered interference.

Based on that, we can define a formulation for the maximum buffered interference over the contention domain cd_{ij}:

$$bi_{ij} = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}| \quad (6)$$

We then use that value to propose a new upper-bound for the downstream indirect interference:

$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil bi_{ij} \quad (7)$$

The ceiling function in Equation 7 determines the number of hits suffered by τ_j from every τ_k in the downstream indirect interference set of τ_i, which is multiplied by the buffered interference of each hit calculated by Equation 6, i.e. the time it takes for the flits of τ_j buffered along cd_{ij} to flow and potentially hit τ_i again. That time is given by the product of the amount of buffer space per router $buf(\Xi)$ on the virtual channel of τ_j, the time it takes for each one of the buffered flits to cross a network link, given by $linkl(\Xi)$, and the number of links in the contention domain of τ_j and τ_i given by $|cd_{ij}|$.

While the proposed upper bound in Equation 7 is often tighter than the one presented by Xiong *et al.*, that is not always the case. In the cases that the downstream interference on τ_j is not large enough to generate backpressure to fill up all the buffers along the contention domain cd_{ij}, it is likely that the maximum buffered interference bi_{ij} could be larger than the maximum downstream interference $C_k + I_{kj}^{down}$, making the XLWX analysis tighter. Therefore, we rewrite Equation 7 to use, for every downstream interference hit, the smallest value between bi_{ij} and $C_k + I_{kj}^{down}$:

$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down}) \quad (8)$$

The upper bound in Equation 8 can be optimistic in cases when τ_j suffers from both upstream and downstream indirect interference. In such cases, its packets can be "chopped-up" by interfering flows and thus arrive in waves into the contention domain cd_{ij}. If that happens, the amount of supplied flits into the contention domain will not be equal to the amount of departed flits, causing variations to the buffer interference.

Therefore, the proposed analysis (which we refer as IBN) is applied as follows:

- Equation 8 calculates I_{ji}^{down} when computing downstream indirect interference caused by flows that do not themselves suffer from both upstream and downstream interference. This can make the proposed analysis tighter, but never less tight than XLWX.
- Equation 3 calculates I_{ji}^{down} when computing downstream indirect interference caused by flows that do suffer from upstream interference. In such cases, the proposed analysis is exactly the same as XLWX.
- The appropriate values of I_{ji}^{down} are fed into Equation 5 to calculate the worst case response time of each flow.

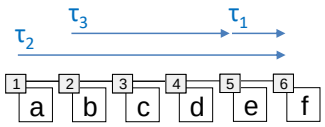

Fig. 3: Flow routes

V. Didactic Example

Let us consider a small didactic example to compare the proposed analysis against XLWX and SB analyses. We assume three flows τ_1, τ_2 and τ_3 with sources, destinations and routes shown in Figure 3, and with the flow parameters shown in Table I, chosen to highlight the effects of the downstream indirect interference of τ_1 over τ_3 through τ_2.

We applied SB, XLWX and IBN analyses to this example, which produced latency upper-bounds R for each flow. To provide evidence that the proposed analysis can capture the influence of the buffer and contention domain sizes on the downstream indirect interference, we tabulate the results of the proposed analysis considering different buffer sizes (2 and 10-flit buffers per VC), which are identified by the subscript $b = buf(\Xi)$. We also produced cycle-accurate simulation results for the same scenarios, and tabulated the worst observed latency for each flow (using the same subscripts to identify the buffer sizes used in each simulation scenario).

The results in Table II show, as expected, that both the proposed analysis and XLWX provide upper-bounds to the values found using simulation while SB provides optimistic bounds. It also shows that the proposed analysis has much tighter results than XLWX for τ_3 (348 vs 460 for 2-flit buffer networks, or 396 vs 460 for 10-flit buffer networks). This happens because in this example the amount of buffered interference limits the amount of additional interference caused by MPB, showing the real extent of the pessimism introduced by XLWX in its accounting of that problem. The results for the proposed analysis using different buffer sizes show that the common practice of using small buffers in wormhole NoCs is also advantageous in terms of time predictability, since smaller buffers allow the proposed analysis to have tighter bounds because of the limited amount of buffered interference that can build up in the network.

TABLE I: Flow parameters

flow	C (L, $\mid route \mid$)	T	D	J	P
τ_1	62 (60, 3)	200	200	0	1
τ_2	204 (198, 7)	4000	4000	0	2
τ_3	132 (128,5)	6000	6000	0	3

TABLE II: Analysis and simulation results

flow	R^{SB}	R^{XLWX}	$R^{IBN}_{b=10}$	$R^{IBN}_{b=2}$	$R^{sim}_{b=10}$	$R^{sim}_{b=2}$
τ_1	62	62	62	62	62	62
τ_2	328	328	328	328	324	324
τ_3	336	460	396	348	352	336

VI. Large-Scale Quantitative Evaluation

We now provide additional evidence on the tightness of the proposed analysis. First, we performed a large-scale comparison using synthetically-generated flow sets of increasing load. We used two configurations of a priority-preemptive wormhole network-on-chip platform: a 16-core (4x4) and a 64-core (8x8). We used flow sets of increasing workload by varying the number of flows in each set. The flows on each set are based on the following characteristics: periods uniformly distributed between 0.5 s and 0.5 ms, maximum packet lengths uniformly distributed between 128 and 4096 flits, and deadlines equal to the respective periods. Sources and destinations of packet flows are randomly selected, so the average route is longer in the larger platform. Rate-monotonic priority assignment is used despite sub-optimality, given that no optimal assignment is known for this problem.

Figure 4 shows the percentage of cases that each of the analyses is able to guarantee full schedulability: the proposed analysis considering network routers with 2-flit buffers per VC (referred to as IBN2), with 100-flit buffers per VC (referred as IBN100), the unsafe SB analysis and the safe baseline XLWX. Each point represents the percentage of schedulable flow sets using each analysis out of a set with 100 flow sets, each of them with the number of flows indicated over the X-axis.

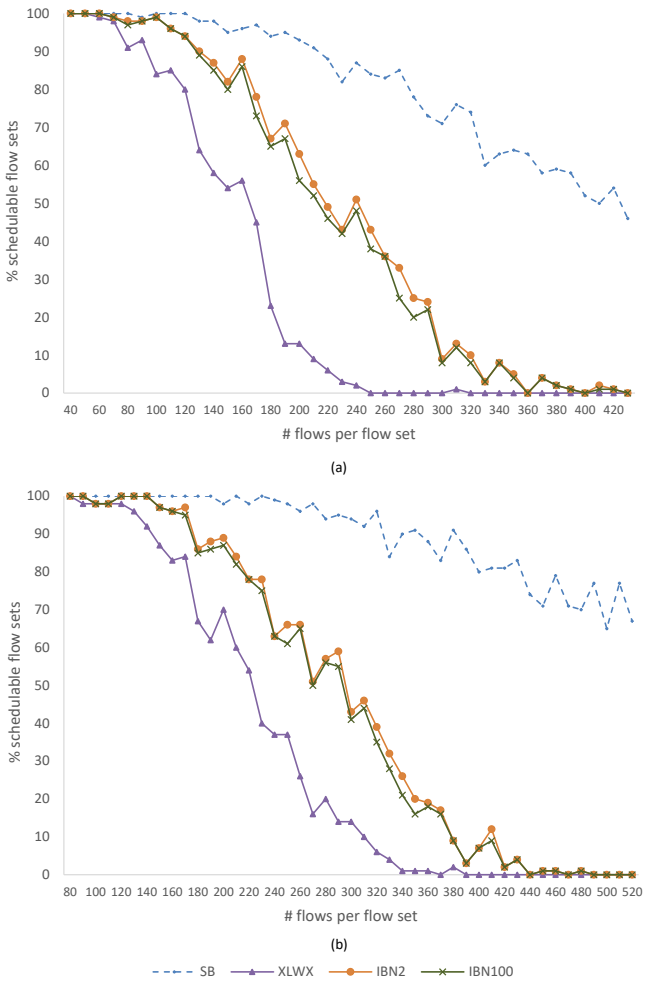

Fig. 4: Schedulability results for the proposed analysis against the SB and XLWX baselines, for (a) 4x4 and (b) 8x8 NoCs.

The lines of IBN2 and IBN100 are very close, but a careful look reveals a difference of up to 8%. This corroborates the statement made in the previous section that large buffers can decrease the predictability of the network because of the more significant buffered interference effects they can produce. We have performed the same experiments with a range of different buffer sizes between 2 and 100, but did not include them in Figure 4 to avoid cluttering the plots. We have consistently observed that, in every case, the analysis was able to guarantee schedulability of a smaller number of flow sets when considering routers with larger buffers.

More importantly, both plots show that the difference in schedulability between XLWX and IBN can be up to 58% in the 4x4 case and up to 45% in the 8x8, showing how much tighter the proposed analysis can be.

We then performed additional experiments using the autonomous vehicle (AV) benchmark from [5] and using a larger variety of NoC topologies, aiming to show the generality of the proposed approach under a realistic scenario. We randomly generated 100 mappings of the AV benchmark onto each of the 26 chosen NoC topologies (from 4 to 100 nodes), and applied the proposed analyses IBN2 and IBN100 as well as the XLWX baseline to determine how many of those mappings are deemed fully schedulable by each of them. Figure 5 shows the results, and again the proposed analysis is shown to be significantly better than XLWX for all topologies: its improved tightness allows it to provide schedulability guarantees to more mappings (up to 67% more). The comparison between both variations of the proposed analysis shows that IBN2 can provide schedulability guarantees to up to 6% more mappings than IBN100.

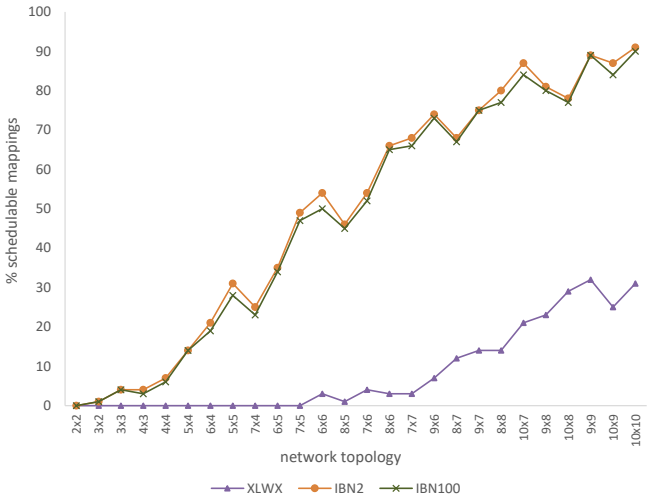

Fig. 5: Schedulability results for the proposed analysis against the XLWX baseline for the AV benchmark mapped onto the topologies indicated over the X-axis).

VII. Conclusions

In this paper, we have reviewed the latest developments in real-time analyses of priority-preemptive NoCs, focusing specifically on the newly-identified problem of multi-point blocking. We claim that XLWX, which is the only analysis that is known to be safe under MPB, is unnecessarily pessimistic as it treats indirect interference as if it were direct interference. In practice, this means that it could deem unschedulable a large number of network configurations that are in reality schedulable and viable. We then propose a novel analysis that takes into account buffering bounds, achieving tighter results than XLWX while still safe under MPB scenarios, therefore establishing the new state-of-the-art in this area. Extensive experimental evidence backs our claim, and also shows a counter-intuitive trade-off between buffer sizes and predictability, as large buffers (which are known to provide improvements on average-case performance) can result in more pessimistic worst-case latencies using the proposed analysis.

We chose to provide intuitions, insight and experimental evidence on the proposed analysis and its improvements, rather than theorems or proofs. We claim that this does not reduce the value of our contribution, since the analyses behind SB [11], SLA [8] and the original XLWX [12] were all backed by theorems and proof sketches, but that did not prevent each of them from being subsequently found to be unsafe. We therefore leave as future work the formalisation of such a proof, as well as the evaluation of proof assistance approaches (as those addressed in [4]) which could prevent such analyses from being shown unsafe. At this point, we only claim is that ours is the tightest analysis that has not been proven optimistic by a counter-example.

References

[1] T. Bjerregaard and S. Mahadevan. A survey of research and practices of Network-on-chip. *ACM Comput Surv*, 38(1):1, 2006.

[2] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. QNoC: QoS architecture and design process for network on chip. *J Syst Arch*, 50(2-3):105–128, 2004.

[3] A. Burns, J. Harbin, and L.S. Indrusiak. A Wormhole NoC Protocol for Mixed Criticality Systems. In *IEEE Real-Time Systems Symposium*, pages 184–195, 2014.

[4] F. Cerqueira, F. Stutz, and B. B. Brandenburg. PROSA: A Case for Readable Mechanized Schedulability Analysis. In *ECRTS Conf*, pages 273–284, 2016.

[5] L. S. Indrusiak. End-to-end schedulability tests for multiprocessor embedded systems based on networks-on-chip with priority-preemptive arbitration. *J Syst Arch*, 60(7):553–561, 2014.

[6] L. S. Indrusiak, A. Burns, and B. Nikolic. Analysis of buffering effects on hard real-time priority-preemptive wormhole networks. *arXiv:1606.02942 [cs]*, 2016.

[7] H. Kashif, S. Gholamian, and H. Patel. SLA: A Stage-Level Latency Analysis for Real-Time Communication in a Pipelined Resource Model. *IEEE Trans Comp*, 64(4):1177–1190, 2015.

[8] H. Kashif and H. Patel. Buffer Space Allocation for Real-Time Priority-Aware Networks. In *RTAS Symposium*, pages 1–12, 2016.

[9] B. Kim, J. Kim, S. Hong, and S. Lee. A real-time communication method for wormhole switching networks. In *Int Conf on Parallel Processing*, pages 527–534, 1998.

[10] M. Liu, M. Becker, M. Behnam, and T. Nolte. Tighter time analysis for real-time traffic in on-chip networks with shared priorities. In *IEEE/ACM NOCS Symposium*, 2016.

[11] Z. Shi and A. Burns. Real-Time Communication Analysis for On-Chip Networks with Wormhole Switching. In *IEEE/ACM NOCS Symposium*, pages 161–170, 2008.

[12] Q. Xiong, Z. Lu, F. Wu, and C. Xie. Real-Time Analysis for Wormhole NoC: Revisited and Revised. In *GLSVLSI Symposium*, pages 75–80, 2016.

[13] Q. Xiong, F. Wu, Z. Lu, and C. Xie. Extending Real-Time Analysis for Wormhole NoCs. *IEEE Trans Comput*, 66(9), 2017.