Technical Report

A Tighter Real-Time Communication Analysis for Wormhole-Switched Priority-Preemptive NoCs

Borislav Nikolic

Leandro Soares Indrusiak

Stefan M. Petters

CISTER-TR-170111

A Tighter Real-Time Communication Analysis for Wormhole-Switched Priority-Preemptive NoCs

Borislav Nikolic, Leandro Soares Indrusiak, Stefan M. Petters

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

http://www.cister.isep.ipp.pt

Abstract

Simulations and runtime measurements are some of the methods which can be used to evaluate whether a given NoC-based platform can accommodate application workload and fulfil its timing requirements. Yet, these techniques are often time-consuming, and hence can evaluate only a limited set of scenarios. Therefore, these approaches are not suitable for safety-critical and hard real-time systems, where one of the fundamental requirements is to provide strong guarantees that all timing requirements will always be met, even in the worst-case conditions. For such systems the analytic-based real-time analysis is the only viable approach. In this paper the focus is on the real-time communication analysis for wormhole-switched priority-preemptive NoCs. First, we elaborate on the existing analysis and identify one source of pessimism. Then, we propose an extension to the analysis, which efficiently overcomes this limitation, and allows for a less pessimistic analysis. Finally, through a comprehensive experimental evaluation, we compare the newly proposed approach against the existing one, and also observe how the trends change with different traffic parameters.

A Tighter Real-Time Communication Analysis for Wormhole-Switched Priority-Preemptive NoCs

Borislav Nikolić
CISTER/INESC TEC, ISEP
Polytechnic Institute of Porto
Portugal
borni@isep.ipp.pt

Leandro Soares Indrusiak
Real-Time Systems Group
Dept. of Computer Science
University of York, UK
lsi@cs.york.ac.uk

Stefan M. Petters
CISTER/INESC TEC, ISEP
Polytechnic Institute of Porto
Portugal
smp@isep.ipp.pt

ABSTRACT

Simulations and runtime measurements are some of the methods which can be used to evaluate whether a given NoC-based platform can accommodate application workload and fulfil its timing requirements. Yet, these techniques are often time-consuming, and hence can evaluate only a limited set of scenarios. Therefore, these approaches are not suitable for safety-critical and hard real-time systems, where one of the fundamental requirements is to provide strong guarantees that all timing requirements will always be met, even in the worst-case conditions. For such systems the analytic-based real-time analysis is the only viable approach.

In this paper the focus is on the real-time communication analysis for wormhole-switched priority-preemptive NoCs. First, we elaborate on the existing analysis and identify one source of pessimism. Then, we propose an extension to the analysis, which efficiently overcomes this limitation, and allows for a less pessimistic analysis. Finally, through a comprehensive experimental evaluation, we compare the newly proposed approach against the existing one, and also observe how the trends change with different traffic parameters.

1. INTRODUCTION

The technological advancements in the semiconductor technology have led to a stage where further processing power enhancements related to single-core platforms are no longer affordable. Consequently, chip manufacturers took a design paradigm shift and started integrating multiple cores within a single chip [10, 17]. Nowadays, platforms consisting of several cores (*multi-cores*) and more than a dozen of cores (*many-cores*) are commonplace in many scientific areas, e.g. high-performance computing, while they are still an emerging technology in others, like safety critical and real-time systems.

The Network-on-Chip (NoC) [2] architecture became the prevailing interconnect medium in many-cores, due to its scalability potential [11]. For NoCs, one of the most popular data transfer techniques is the *wormhole switching* [12], because of its good throughput and small buffering requirements. Currently available wormhole-switched NoCs employ a wide range of diverse strategies when transferring the data, e.g. different sizes of basic transferable units – *flits*, different router operating frequencies, the (in)existence of virtual channels, different arbitration policies. These design choices have a significant impact on both the performance and the analysis.

The behaviour and performance of NoC-based platforms can be evaluated via simulations and/or runtime measurements. These evaluation techniques are suitable options for some areas, e.g. general-purpose and high-performance computing. However, in some other areas, like the safety-critical and real-time computing, the most important aspect is *not* the overall system performance, but instead, whether a platform can fulfil all timing requirements of a given application workload, even in the worst-case conditions. Simulations and runtime measurements are not suitable approaches for these domains, because they are usually time consuming, and hence can evaluate only a small fraction of all possible scenarios, without any information whether the worst-case scenario was indeed captured, or not. In such cases, the analytic-based real-time analysis is the only viable approach.

Yet, analysis-based approaches are mostly performed at design-time. Thus, their efficiency highly depends on the amount of predictability of the entire system, whereas any non-deterministic system behaviour has to be accounted for in the analysis with a certain degree of pessimism. A more pessimistic analysis may lead to a significant resource over-provisioning and/or underutilisation of platform resources. Conversely, a less pessimistic approach allows to save on design costs, e.g. by choosing a cheaper platform with fewer resources, which still guarantees the fulfilment of all timing constraints. Moreover, with the less pessimistic analysis, the resources of the platform can be exploited more efficiently, for example, by accommodating the additional workload, or by decreasing the power consumption via core shutdowns and smaller router frequencies. Thus, deriving the analysis with as less pessimism as possible is of paramount importance in the real-time domain.

Contribution: In this paper the focus is on the real-time communication analysis for wormhole-switched priority-preemptive NoCs. Specifically, the main objective is to derive guarantees that all traffic flows complete their traversal over the NoC without violating posed timing constraints, even in the worst-case conditions. First, we elaborate on the ex-

This work was completed in June 2014, and was submitted, but not selected to appear at the following two conferences: 8th IEEE/ACM International Symposium on Networks-on-Chip (NOCS 2014), and 22nd International Conference on Real-Time Networks and Systems (RTNS 2014).

isting analysis [15] (**Section 4**), and identify one source of pessimism (**Section 5.2**). Then, we propose the extension, which overcomes the aforementioned limitation, and allows for a less pessimistic analysis (**Section 5.3**). Finally, we compare the new approach with the existing one, and observe how the trends change with different traffic parameters (**Section 5.4, Section 5.5 and Section 6**).

2. RELATED WORK

Contrary to the popular belief, the wormhole switching technique [12] is not a novelty. In fact, it has been introduced more than 20 years ago. However, it has been largely neglected, since the alternative *store-and-forward switching* technique was providing satisfactory results [11]. Yet, as the data that has to be transferred kept increasing, buffering within routers became a challenge, which lately brought the wormhole switching back into focus. Nowadays, some NoC-based many-cores employ this technique, e.g. [10, 17].

When organising the access to the interconnect medium, NoC-based platforms employ a diverse range strategies. All approaches can be broadly classified into two categories: *contention-free* techniques and *contention-aware* techniques. An example of the former category is the AEthereal platform [8], where a time-division-multiple-access approach is used to organise the access to interconnect resources. Conversely, if contentions among traffic flows are allowed, the performance of the system may improve, but the analysis of the system behaviour becomes more complex as well. This is especially emphasized in cases where a platform provides only a single virtual channel. For this model, several analyses have been proposed to compute the worst-case traversal times of traffic flows [5, 6, 7, 14], however, due to complex interference patterns, the obtained values may be overly pessimistic [5].

If a platform provides multiple virtual channels [3, 4], the benefits are twofold: (i) the throughput significantly increases [3, 4], and (ii) traffic preemptions can be implemented [16]. Shi and Burns [15] proposed the real-time analysis assuming: per-flow distinctive priorities, per-priority virtual channels and flit-level preemptions. Subsequently, Nikolić et al. [13] relaxed the requirements for virtual channels, proving that the analysis can remain unaffected as long as the number of virtual channels is at least equal to the maximum number of contentions for any port of any router, which is a realistic assumption. Indeed, assuming a 2D mesh NoC interconnect with a 10 × 10 grid and 400 traffic flows, on average, only 8 virtual channels are needed [13], while platforms with 8 virtual channels (e.g. [10]) were already available more than 5 years ago.

So far, the aforementioned model (a wormhole-switched priority-aware NoC with flit-level preemptions via virtual channels) appears to be a promising steps towards real-time NoCs. Therefore, providing any improvement over the state-of-the-art analysis [15] would be a valuable contribution, primarily to the safety-critical and the real-time domain. Our work is motivated with this reasoning.

3. SYSTEM MODEL

3.1 Platform

We consider a many-core platform Π, comprised of $m \times n$ tiles, interconnected with a 2D mesh NoC, like the system depicted in Figure 1. Each tile contains a single core and a single router. Each router has a set of ports, which are used to exchange the data with the local core, as well as with the neighbouring routers. Each pair of communicating ports is connected with two unidirectional links. Additionally, the platform employs a wormhole switching technique with the credit-based flow-control mechanism, where acknowledgements use separate physical links (see the upper part of Figure 2). This means that, prior to sending, each data packet is divided into small fixed-size elements called *flits*. The header flit establishes the path, and the rest of the flits follow in a pipeline manner. Moreover, we assume a deterministic, dimension-ordered XY routing mechanism, which is deadlock and livelock free [9].

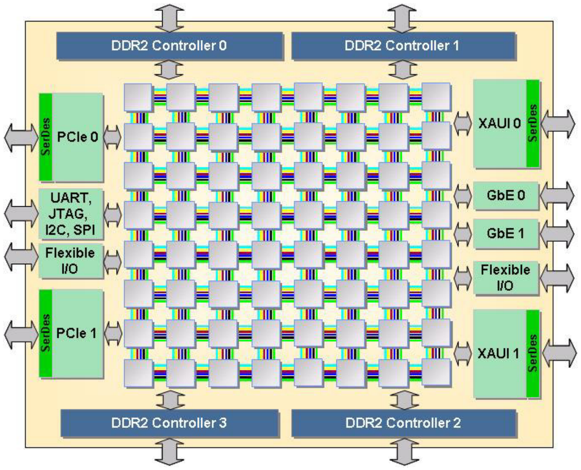


Figure 1: TILEPro64 Platform

The platform provides virtual channels. A virtual channel is implemented as an additional buffer within every port of every router. Virtual channels are used as an infrastructure to implement the priority-preemptive router-arbitration policy, i.e. to store flits of blocked/preempted packets and offer the network resources to other packets which can freely progress. This significantly improves the throughput (performance) [3, 4], and also makes the system behaviour more predictable and deterministic, which is of paramount importance for the real-time analysis. The requirement is that the number of virtual channels is at least equal to the maximum number of contentions for any port of any router. As already described (Section 2), this guarantees that flits of each packet will have an available virtual channel in every port along the path [13]. The router architecture is depicted in the upper part of Figure 2. Notice, that our target platform is very similar to existing many-core systems (e.g. [10, 17]) and that the required hardware features already exist within these platforms. Note, our method also applies to any NoC topology and any deterministic routing technique, which guarantee a single continuous contention domain between any pair of contending traffic flows. For clarity purposes, the focus of this paper is on 2D mesh NoCs with the XY routing mechanism.

3.2 Workload

The workload consists of a traffic flow-set $\mathcal{F}$, which is a collection of flows $\{f_1, f_2, ...f_{z-1}, f_z\}$. Each flow $f_i \in \mathcal{F}$ is characterised by a deterministic XY-routed path from its

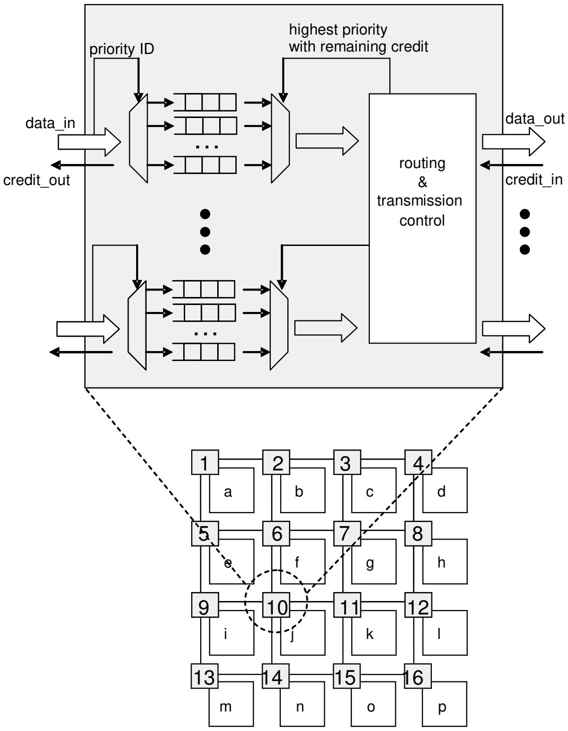


Figure 2: Platform and router architecture

source core to its destination core, expressed as a set of traversed links $\mathcal{L}_i$, where $|\mathcal{L}_i|$ denotes the cardinality of the path, which is, in the literature, often called *the number of hops*. Additional flow parameters are: its size $size(f_i)$, its unique priority P_i, its minimum inter-release period T_i, and its deadline D_i. In this paper we assume that all flows have implicit deadlines, i.e. $\forall f_i \in \mathcal{F} : D_i = T_i$. Each flow f_i generates a potentially infinite sequence of packets. A packet released at the time instant t should be received by the destination no later than $t+D_i$. If it fails to do so, it has *missed a deadline*. If guarantees can be provided that a flow will not miss its deadline, even in the worst-case conditions, it is considered *schedulable*. If all flows of the flow-set are schedulable, the flow-set is considered *schedulable*.

4. BACKGROUND AND PRELIMINARIES

In this section we will describe the fundamental concepts and properties of the real-time communication analysis for NoCs, which are essential for understanding our approach.

Prop. 1: A traversal time of a packet, belonging to a flow f_i, termed C_i (Equation 1), is equal to the time it takes for its header flit to establish the path and reach the destination, augmented by the transfer time of the rest of the flits, once the header reaches the destination. d_r is the delay to route a packet header, d_l is the latency to transfer one flit across one link, while $size(flit)$ represents the size of one flit. C_i is in the literature also known as the *basic network latency*, and it covers the case when the packet does not suffer any interference from the packets of other higher-priority flows.

$$C_i = \overbrace{|\mathcal{L}_i| \times d_l + (|\mathcal{L}_i| - 1) \times d_r}^{\text{header to reach the destination}} + \overbrace{\left\lceil \frac{size(f_i)}{size(flit)} \right\rceil \times d_l}^{\text{the rest of the flits}} \quad (1)$$

Prop. 2: A flow f_i can be preempted only by higher-priority flows which share a part of the path with it, called *directly interfering flows*. Let $\mathcal{F}_D(f_i)$ be a set of directly interfering flows of f_i. Formally:

$$\forall f_j \in \mathcal{F} : (P_j > P_i \land \mathcal{L}_j \cap \mathcal{L}_i \neq \emptyset) \Rightarrow f_j \in \mathcal{F}_D(f_i) \quad (2)$$

Prop. 3: Interference caused by a single preemption of any $f_j \in \mathcal{F}_D(f_i)$ to f_i is equal to its traversal time C_j.

Prop. 4: The worst-case scenario for the flow f_i occurs when, during its traversal, all its directly interfering flows release their packets at their respective maximum possible rates. The traversal time of f_i in this scenario is called *the worst-case traversal time*, R_i in Equation 3. J_j^R is the release jitter [1], and it denotes the maximum deviation of two successive packets released from their period T_j. J_j^I is the interference jitter, and it symbolises the interference caused to f_i by other flows which are not directly interfering with it, but are directly interfering with f_j and hence can indirectly cause additional interference to f_i. An interested reader can find more details about Equation 3 in the previous work of Shi and Burns [15]. Notice that Equation 3 has a recursive notion, i.e. the term R_i exists on both sides of the equation. Therefore, it is solved iteratively [1], until reaching a fixed converging point. If it cannot be solved (e.g. in the case of starvation), the flow is deemed unschedulable.

$$R_i = C_i + \sum_{\forall f_j \in \mathcal{F}_D(f_i)} \left\lceil \frac{R_i + J_j^R + J_j^I}{T_j} \right\rceil \times C_j \quad (3)$$

Prop. 5: If the computed worst-case traversal time of the flow f_i is less than or equal to its deadline, i.e. $R_i \leq D_i$, this means that the packets of f_i will never miss the deadline due to the other traffic existing within the same platform, and thus f_i is schedulable. By computing the worst-case traversal times of all flows of the flow-set, it can be determined whether the entire flow-set is schedulable or not.

Note, the value of R_i, obtained by solving Equation 3, does not present the exact worst-case traversal time that f_i might experience while traversing, but rather its analytically computed safe upper-bound estimate (see Figure 3). In some cases, the upper-bounds obtained by solving Equation 3 may be pessimistic, which is thoroughly discussed in the next section.

5. PROPOSED APPROACH

5.1 Work Objectives

The work objectives can be summarised as follows. Given the platform Π and the flow-set $\mathcal{F}$, provide the analysis to obtain the upper-bound estimates on the worst-case traversal times of individual flows, such that the obtained bounds are as tight as possible (with as less pessimism as possible).

We illustrate this with the example given in Figure 3. Each flow has its traversal times obtained via (i) the analysis, and (ii) measurements. The goal is to reduce the gap

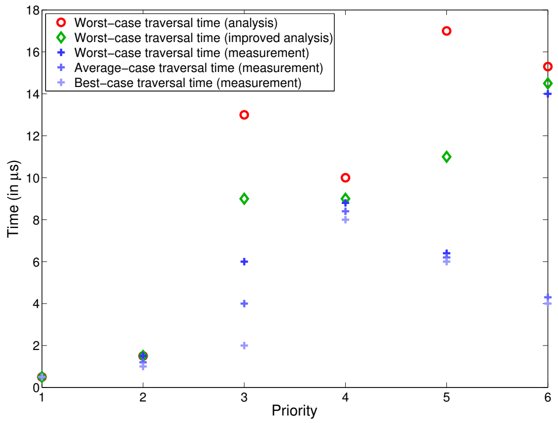


Figure 3: Example of flow traversal times

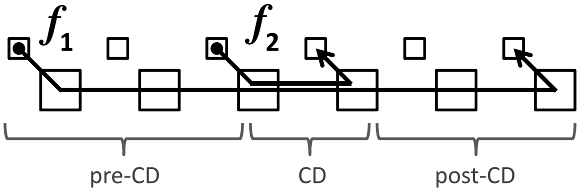
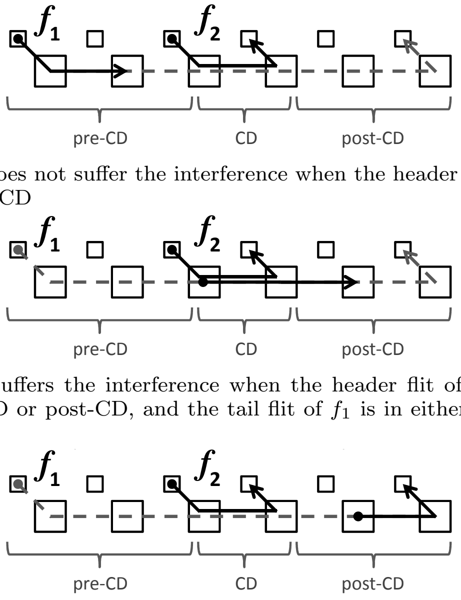


Figure 4: Example of traffic flows



(a) f_2 does not suffer the interference when the header flit of f_1 is in pre-CD

(b) f_2 suffers the interference when the header flit of f_1 is in either CD or post-CD, and the tail flit of f_1 is in either pre-CD or CD

(c) f_2 does not suffer the interference when the tail flit of f_1 is in post-CD

Figure 5: Detailed analysis of the interference that f_1 causes to f_2

between the analytic and the measured worst-case values, i.e. to provide the analysis which will produce significantly tighter (less pessimistic) estimates, e.g. green diamonds in Figure 3.

5.2 Source of Pessimism in Existing Analysis

In the existing analysis [15] (Section 4), it is assumed that the flow under analysis suffers the interference from the entire traversal of a higher-priority flow (see Prop. 3). However, that assumption may be overly pessimistic, as described below.

Consider the example of two flows, f_1 and f_2, illustrated in Figure 4, where $P_1 > P_2$. Small and big rectangles depict cores and routers, respectively. Recall, that in the existing analysis, the entire traversal of f_1 would be considered as the interference to f_2 (Prop. 3). Let us closely investigate the paths of these flows. f_1 and f_2 share the common part of the path, which consists of one link, and which we hereafter refer to as *the contention domain* – CD[1]. Let us divide the path of f_1 into 3 parts: (i) before f_1 and f_2 start sharing a common part of the path – **pre-CD**, (ii) while f_1 and f_2 share the common part of the path – **CD**, and (iii) after f_1 and f_2 stop sharing the common part of the path – **post-CD**.

Notice that while the header flit of f_1 traverses pre-CD, f_1 does not cause any interference to f_2 (see Figure 5a). Thus, in the existing analysis, the traversal of the header flit of f_1 through pre-CD is unnecessarily considered as the interference that f_1 causes to f_2. If we exclude that delay from the interference that f_1 causes to f_2, we can decrease the pessimism of the analysis and obtain a tighter upper-bound estimate on the worst case traversal time of f_2.

Once the header flit of f_1 reaches CD, f_2 starts suffering the interference (see Figure 5b). When the tail flit of f_1 leaves CD, f_2 stops suffering the interference from f_1, and may continue its progress (see Figure 5c). Thus, the traversal of the tail flit of f_1 through post-CD is also unnecessarily considered as the interference in the existing analysis. By excluding this delay from the interference that f_1 causes to f_2, we can further decrease the pessimism of the analysis and obtain even tighter upper-bound estimate on the worst-case traversal time of f_2.

5.3 Less Pessimistic Worst-Case Analysis

In this section we propose an extension to the existing analysis. Specifically, based on the observations from the previous section, we develop a new, less pessimistic method to compute the interference that directly interfering flows cause to the flow under analysis. Subsequently, we can obtain tighter upper-bound estimates on the worst-case traversal times.

Consider again the example illustrated in Figure 4. Let the parts of $\mathcal{L}_1$ constituting sections pre-CD, CD and post-CD (with respect to flow f_2) be $\mathcal{L}_{1,2}^{pre-CD}$, $\mathcal{L}_{1,2}^{CD}$ and $\mathcal{L}_{1,2}^{post-CD}$, respectively. It is obvious that the union of these parts forms the entire path of f_1, i.e. $\mathcal{L}_1 = \mathcal{L}_{1,2}^{pre-CD} \cup \mathcal{L}_{1,2}^{CD} \cup \mathcal{L}_{1,2}^{post-CD}$. Now, as described in the previous section, the less pessimistic estimate of the interference that a single traversal of f_1 causes to f_2, denoted as $I_{1,2}$ (Equation 4), can be computed by subtracting (i) the time it takes a header flit of f_1 to traverse pre-CD $\sigma_{1,2}^{pre-CD}$, and (ii) the time it takes a tail flit of f_1 to traverse post-CD $\sigma_{1,2}^{post-CD}$, from the entire traversal time of f_1.

$$I_{1,2} = C_1 - \sigma_{1,2}^{pre-CD} - \sigma_{1,2}^{post-CD} \tag{4}$$

where $\sigma_{1,2}^{pre-CD}$ and $\sigma_{1,2}^{post-CD}$ are computed as follows:

[1] The XY-routing mechanism assures that any two flows with a direct interference relationship have exactly one CD

$$\sigma_{1,2}^{pre-CD} = \overbrace{|\mathcal{L}_{1,2}^{pre-CD}| \times d_l + max\left\{0, \left(|\mathcal{L}_{1,2}^{pre-CD}| - 1\right)\right\} \times d_r}^{\text{header flit traversal through pre-CD}} \tag{5}$$

$$\sigma_{1,2}^{post-CD} = \overbrace{|\mathcal{L}_{1,2}^{post-CD}| \times d_l}^{\text{tail flit traversal through post-CD}} \tag{6}$$

If we apply this reasoning when computing the interference that a flow under analysis f_i suffers from all directly interfering flows, we can obtain a tighter worst-case traversal time R_i^*, expressed with Equation 7. Note, that similar to Equation 3, Equation 7 also has a recursive notion, thus is solved iteratively.

$$R_i^* = C_i + \sum_{\forall f_j \in \mathcal{F}_D(f_i)} \left\lceil \frac{R_i^* + J_j^R + J_j^I}{T_j} \right\rceil \times I_{j,i} \tag{7}$$

5.4 Observations

When we compare the existing and the new method to compute the worst-case traversal times of individual flows, we can notice several interesting facts:

Observation 1: When compared with the existing one, the proposed method never underperforms. Indeed, the new approach always derives upper-bounds which are either the same, or less pessimistic. In other words, it always holds that $\forall f_i \in \mathcal{F} : R_i^* \leq R_i$. We prove that with Theorem 1.

Theorem 1. *The worst-case traversal time, of any flow of the flow-set, computed with the new method, is either equal to or less pessimistic than the one obtained with the existing method.*

Proof. Proven directly. Consider two flows f_i and f_j, where $P_j > P_i$ and $\mathcal{F}_D(f_i) = \{f_j\}$. Let $\sigma_{j,i}$ be the difference in the interference caused by a single preemption of f_j to f_i, computed with the existing and the proposed method (Equation 8).

$$\sigma_{j,i} = C_j - I_{j,i} = \sigma_{j,i}^{pre-CD} + \sigma_{j,i}^{post-CD} = |\mathcal{L}_{j,i}^{pre-CD}| \times d_l + max\left\{0, \left(|\mathcal{L}_{j,i}^{pre-CD}| - 1\right)\right\} \times d_r + |\mathcal{L}_{j,i}^{post-CD}| \times d_l \tag{8}$$

Since all the terms of Equation 8 are non-negative values, it follows that $\sigma_{j,i} \geq 0$. Let $K_{j,i}$ be the number of preemptions that f_j can cause to f_i during the worst-case traversal time of f_i. Also, let σ_i be the difference in the total interference caused to f_i by all its directly interfering flows, computed with the existing and the proposed method (Equation 9).

$$\sigma_i = \sum_{\forall f_j \in \mathcal{F}_D(f_i)} \sigma_{j,i} \times K_{j,i} \tag{9}$$

Since all the terms of Equation 9 have non-negative values, it follows that $\sigma_i \geq 0$. Moreover, as the existing and the new method differ only in the way how the interference is computed, it follows that $R_i - R_i^* = \sigma_i \geq 0$. □

Theorem 2 provides a proof that the obtained upper-bounds are safe.

Theorem 2. *The traversal time of any packet belonging to the flow f_i can not be greater than R_i^* (Equation 7), even in the worst-case conditions.*

Proof. Proven directly. Shi and Burns have proven that R_i (Equation 3) presents the upper-bound on the worst-case traversal time of the flow f_i (see Theorem 1 in [15]). Thus, if we prove that there exists some discontinuous time interval σ_i which is a part of R_i, and during which f_i does not progress, nor any of its interfering flows causes the interference to it, we will prove that $R_i - \sigma_i$ is also a safe upper-bound on the worst-case traversal time of f_i.

Consider two traffic flows f_i and f_j, where $P_j > P_i$ and $\mathcal{F}_D(f_i) = \{f_j\}$. According to Prop. 3, the entire traversal of f_j is considered as the interference that f_j causes to f_i, and hence entirely contributes to R_i. Let $\sigma_{j,i}$ (Equation 8) be the sum of: (i) $\sigma_{j,i}^{pre-CD}$, which is the interval when the header flit of f_j traverses pre-CD and (ii) $\sigma_{j,i}^{post-CD}$, which is the interval when the tail flit of f_j traverses post-CD. By definition, both a necessary and sufficient condition for the contention between f_i and f_j is that both of them attempt to traverse the CD section at the same time. However, during $\sigma_{j,i}^{pre-CD}$ and $\sigma_{j,i}^{post-CD}$, the flow f_j does not traverse CD, hence f_i can safely progress. Thus, the maximum interference that one packet of f_j can cause to f_i has a safe upper-bound, which is $C_j - \sigma_{j,i}$. Subsequently, if during the traversal of f_i, packets of f_j can appear at most $K_{j,i}$ times, then the safe upper-bound on the interference that f_j can cause to f_i is $K_{j,i} \times (C_j - \sigma_{j,i})$. By elevating this reasoning, we conclude that the worst-case traversal time of f_i has a safe upper-bound R_i^* (Equation 10).

$$R_i^* = R_i - \sum_{\forall f_j \in \mathcal{F}_D(f_i)} K_{j,i} \times \sigma_{j,i} = R_i - \sigma_i \tag{10}$$

□

Let us observe the improvements of the proposed method over the existing one on a small-scale example given in Figure 4, where the flow characteristics are as follows: $size(f_1) = size(f_2) = 48B$, $P_1 > P_2$, $T_1 = D_1 = T_2 = D_2 = 1000ns$. The other analysis parameters are given in Table 1 (Section 6). For clarity purposes assume that the release jitters are equal to zero, i.e. $J_1^R = J_2^R = 0$. Moreover, since only two flows exist, there are no indirect interferences, thus interference jitters are equal to zero [15], i.e. $J_1^I = J_2^I = 0$.

f_1 is the higher-priority flow, so its worst-case traversal time will be the same with both methods:

$$R_1 = R_1^* = C_1 = |\mathcal{L}_1| \times d_l + (|\mathcal{L}_1| - 1) \times d_r + \left\lceil \frac{size(f_1)}{size(flit)} \right\rceil \times d_l = 7 \times 0.5ns + 6 \times 1.5ns + \left\lceil \frac{48B}{16B} \right\rceil \times 0.5ns = 14ns$$

However, the worst-case traversal time of f_2 is different. Let us first obtain the value with the existing analysis. For that, we need to compute C_2.

$$C_2 = |\mathcal{L}_2| \times d_l + (|\mathcal{L}_2| - 1) \times d_r + \left\lceil \frac{size(f_2)}{size(flit)} \right\rceil \times d_l = 3 \times 0.5ns + 2 \times 1.5ns + \left\lceil \frac{48B}{16B} \right\rceil \times 0.5ns = 6ns$$

We compute R_2 as follows:

$$R_2 = C_2 + \left\lceil \frac{R_2 + J_1^R + J_1^I}{T_1} \right\rceil \times C_1$$

$$R_2^0 = 6 + \left\lceil \frac{0+0+0}{1000ns} \right\rceil \times 14ns = 6ns$$

$$R_2^1 = 6 + \left\lceil \frac{6ns+0+0}{1000ns} \right\rceil \times 14ns = 20ns$$

$$R_2^2 = 6 + \left\lceil \frac{20ns+0+0}{1000ns} \right\rceil \times 14ns = 20ns$$

Now, let us compute the worst-case traversal time of f_2 with the new approach. To do so, we first have to compute $I_{1,2}$, which is the interference that f_1 causes to f_2 with a single preemption. The lengths of the relevant sections are: $|\mathcal{L}_{1,2}^{pre-CD}| = 3$, $|\mathcal{L}_{1,2}^{CD}| = 1$ and $|\mathcal{L}_{1,2}^{post-CD}| = 3$ (see Figure 4).

$$I_{1,2} = C_1 - |\mathcal{L}_{1,2}^{pre-CD}| \times d_l - max\left\{0, \left(|\mathcal{L}_{1,2}^{pre-CD}| - 1\right)\right\} \times d_r$$
$$-|\mathcal{L}_{1,2}^{post-CD}| \times d_l$$

$$I_{1,2} = 14ns - 3 \times 0.5ns - 2 \times 1.5ns - 3 \times 0.5ns = 8ns$$

We compute R_2^* as follows:

$$R_2^* = C_2 + \left\lceil \frac{R_2^* + J_1^R + J_1^I}{T_1} \right\rceil \times I_{1,2}$$

$$R_2^0 = 6 + \left\lceil \frac{0+0+0}{1000ns} \right\rceil \times 8ns = 6ns$$

$$R_2^1 = 6 + \left\lceil \frac{6ns+0+0}{1000ns} \right\rceil \times 8ns = 14ns$$

$$R_2^2 = 6 + \left\lceil \frac{14ns+0+0}{1000ns} \right\rceil \times 8ns = 14ns$$

We can see that the worst-case traversal time of f_2 obtained with the new approach is $14ns$, while it was $20ns$ with the existing analysis, which is an improvement of $6ns$, or in relative terms, an improvement of 30%.

Observation 2: The improvements of the proposed approach over the existing one depend on the lengths of the pre-CD, CD and post-CD sections of interfering flows. Consider again the example from Figure 4, where $P_1 > P_2$. Assuming that the path of f_1 is constant, i.e. $|\mathcal{L}_1| = const$, from Equations 8-9 it straightforwardly follows that the improvements in the worst-case traversal time of f_2 are greater when both $|\mathcal{L}_{1,2}^{pre-CD}|$ and $|\mathcal{L}_{1,2}^{post-CD}|$ are bigger. Since, the interference relationship between f_1 and f_2 is possible if and only if a contention between these two flows exist, i.e. $|\mathcal{L}_{1,2}^{CD}| \geq 1$, it follows that the necessary condition for the maximum improvements is: $|\mathcal{L}_{1,2}^{pre-CD}| + |\mathcal{L}_{1,2}^{post-CD}| = |\mathcal{L}_1| - 1$.

We demonstrate that with an illustrative example given in Figure 7. Consider the same flow characteristics as in the previous example, i.e. $size(f_1) = size(f_2) = 48B$, $P_1 > P_2$, $T_1 = D_1 = T_2 = D_2 = 1000ns, J_1^R = J_2^R = 0$. If we compute the worst-case traversal times with both methods we get the following results: $R_1 = R_1^* = 14ns$, $R_2 = 24ns$ and $R_2^* = 20.5ns$. When compared with the previous example, it is visible that the improvements in the worst-case traversal time of f_2 dropped from $6ns$ to $3.5ns$, or expressed relatively, from 30% to less than 15%.

Note, a special case occurs when paths of interfering flows entirely overlap, i.e. $|\mathcal{L}_{1,2}^{pre-CD}| = \mathcal{L}_{1,2}^{post-CD}| = 0$, $|\mathcal{L}_{1,2}^{CD}| = |\mathcal{L}_1|$. In such scenarios there are no improvements because both approaches return the same values.

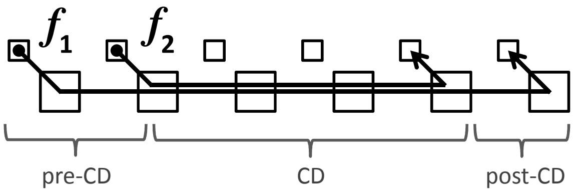


Figure 7: Example of traffic flows

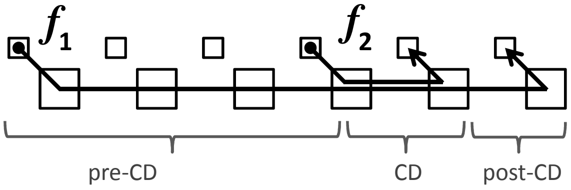
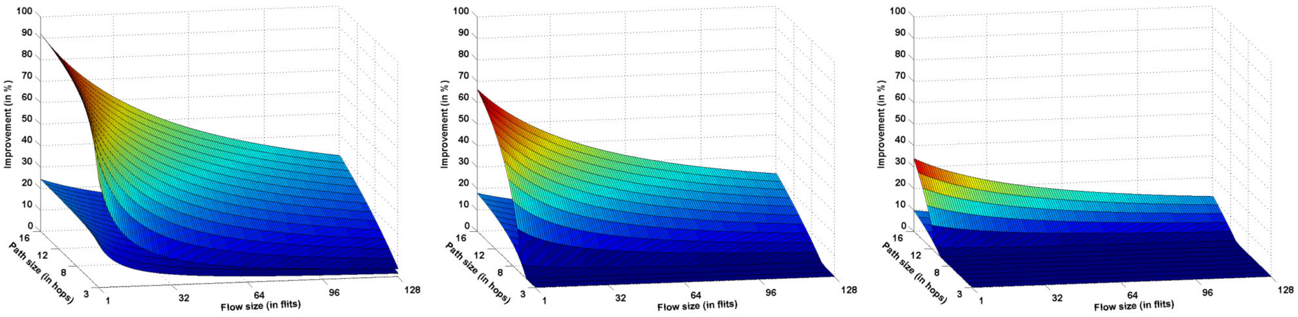


Figure 8: Example of traffic flows

Observation 3: The improvements of the proposed approach over the existing one depend on the position of the CD section of interfering flows. Consider again the example given in Figure 4, where $P_1 > P_2$. Assuming that the path of f_1 and the length of the CD section are constant, i.e. $|\mathcal{L}_1| = const, |\mathcal{L}_{1,2}^{CD}| = const$, from Equations 8-9 it straightforwardly follows that the improvements in the worst-case traversal time of f_2 are more influenced by the length of the pre-CD than the post-CD section. Thus, a necessary and sufficient condition for maximum improvements is: $|\mathcal{L}_{1,2}^{pre-CD}| = |\mathcal{L}_1| - 1, |\mathcal{L}_{1,2}^{CD}| = 1$ and $|\mathcal{L}_{1,2}^{post-CD}| = 0$.

To demonstrate that, let us analyse the example from Figure 8, which differs from the example from Observation 1 (Figure 4) only in the position of the CD section. If we compute the worst-case traversal times for both flows, assuming the same flow characteristics ($size(f_1) = size(f_2) = 48B$, $P_1 > P_2$, $T_1 = D_1 = T_2 = D_2 = 1000ns, J_1^R = J_2^R = 0$), we reach the following values: $R_1 = R_1^* = 14ns$, $R_2 = 20ns$ and $R_2^* = 12.5ns$. When compared with the example from Observation 1, it is visible that the improvements in the worst-case traversal time of f_2 increased from $6ns$ to $7.5ns$, or expressed relatively, from 30% to 37.5%.

Observation 4: The improvements of the proposed approach over the existing one do not depend on flow-sizes. Indeed, from Equations 8-9 it is visible that only the paths, but not the sizes of the flows influence the improvements. Therefore, with the increase in flow sizes, the worst-case traversal times also grow, but the improvements remain the same in absolute values, and hence report decrease in relative values. If we compute the worst-case traversal times for the example given in Figure 4, but this time with bigger flow sizes ($size(f_1) = size(f_2) = 160B$, $P_1 > P_2$, $T_1 = D_1 = T_2 = D_2 = 1000ns, J_1^R = J_2^R = 0$), we get the following results: $R_1 = R_1^* = 17.5ns$, $R_2 = 27ns$ and $R_2^* = 21ns$. Like in the equivalent case for $size(f_1) = size(f_2) = 48B$, we again have $R_2 - R_2^* = 6ns$, however, the relative improvements drop from 30% to 22.2%. This implies that, as flow sizes increase, the absolute improvements remain unaffected, however, the relative improvement decrease. Indeed, in a hypothetical case with flows of infinitely large sizes, the relative improvements asymptotically converge towards 0%.



(a) $|\mathcal{L}_{1,2}^{CD}| = 1$ (b) $|\mathcal{L}_{1,2}^{CD}| = min\{5, |\mathcal{L}_1|\}$ (c) $|\mathcal{L}_{1,2}^{CD}| = min\{10, |\mathcal{L}_1|\}$

Figure 6: Numerical example with two contending flows f_1 and f_2, where $P_1 > P_2$ and $|\mathcal{L}_1| = |\mathcal{L}_2|$

5.5 Numerical Example

In order to get a better insight into how different parameters influence analysis improvements, we use a small-scale numerical example. We consider two contending flows f_1 and f_2 where $P_1 > P_2$ and $\mathcal{F}_D(f_2) = \{f_1\}$. We vary the length of the CD section, the sizes, and the paths of the flows. For each particular scenario we compute the worst-case traversal times of f_2 with both approaches, and observe the relative improvements of the proposed approach. Figures 6a-6c demonstrate the results. In each figure a lower surface covers a corner case where the length of the pre-CD section is equal to zero, i.e. $|\mathcal{L}_{1,2}^{pre-CD}| = 0$ and $|\mathcal{L}_{1,2}^{post-CD}| = |\mathcal{L}_1| - |\mathcal{L}_{1,2}^{CD}|$, while the upper surface represents the opposite corner case, i.e. the length of the post-CD section is equal to zero, i.e. $|\mathcal{L}_{1,2}^{pre-CD}| = |\mathcal{L}_1| - |\mathcal{L}_{1,2}^{CD}|$ and $|\mathcal{L}_{1,2}^{post-CD}| = 0$. The trends in Figures 6a-6c coincide with all the conclusions from Observations 1-4. Moreover, it is visible that the improvements are equal to zero in cases where the path of f_1 entirely belongs to the CD section, i.e. $|\mathcal{L}_{1,2}^{CD}| = |\mathcal{L}_1|$, which has already been inferred in Observation 2.

6. EVALUATION

Table 1: Analysis parameters

Platform size	**8 × 8**
Link width = flit size	**16B**
NoC frequency	**2GHz**
Router delay - d_r	**3 cycles (1.5ns)**
Link delay - d_l	**1 cycle (0.5ns)**
Flow periods	**[1 - 10]ms**

In the previous section we have studied the improvements of the proposed analysis over the existing one on small-scale illustrative examples consisting of only two flows. In this section we perform a large-scale comprehensive experimental evaluation with the main objective to compare the two approaches, but this time assuming flow-sets with hundreds of flows. This will help us to investigate whether the improvement trends from a small-scale example also hold for large flow-sets, and to what extent. Subsequently, we can identify scenarios (flow characteristics) for which the proposed approach reports the biggest improvements. Analysis parameters are given in Table 1[2].

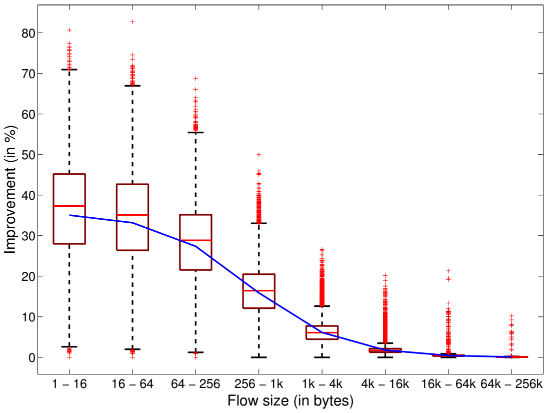


Figure 9: Improvements wrt flow sizes

Experiment 1. *Improvements wrt flow sizes*

In this experiment we investigate how different flow sizes influence the improvements of the proposed analysis over the existing one. We generate several categories of flow-sets, with different flow size ranges: $1B-16B, 16B-64B, ..., 64kB-256kB$. For each category we generate 100 flow-sets, where a flow-set consists of 200 flows. The size of each flow is randomly generated, within the limits of the respective flow-set category. Also, for each flow, the priority, the source core and the destination core are randomly generated, where flow paths comply with the XY routing policy. Subsequently, for each flow of the flow-set we compute the worst-case traversal time with both methods, and measure the improvements in relative terms.

Figure 9 shows the results. Since the improvements do not depend on flow sizes (see Observation 4), the absolute improvements of the new approach over the existing one are constant, irrespective of the flow sizes. However, as the increase in the flow size causes a uniform increase in the worst-case traversal times obtained with both methods, the relative improvements of the new approach (y-axis) decrease as

[2] A period of each flow is randomly generated, within the given limits. If a generated flow-set is not schedulable, then periods of all flows are uniformly increased (even beyond the limits) until the flow-set becomes schedulable.

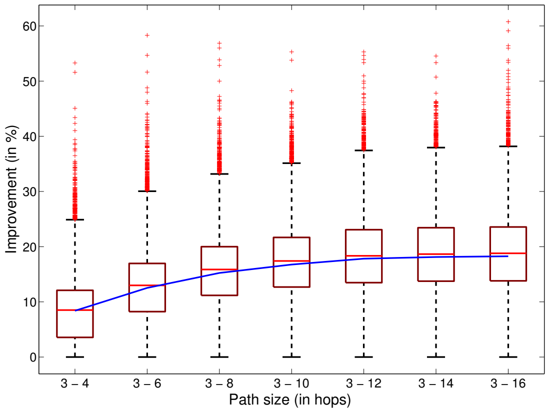


Figure 10: Improvements wrt path sizes

the flow sizes increase (x-axis). Another interesting finding is that, irrespective of the flow sizes, there are always flows for which the new approach does not yield better results. These are the highest-priority flows which do not suffer any interference, hence for them both methods return the same results.

Experiment 2. *Improvements wrt paths sizes*

In this experiment we vary the lengths of flow paths and observe their influence on the improvements. We again generate several categories of flow-sets, but this time with different lengths of flow paths: $3-4, 3-6, ..., 3-16$. For each category 100 flow-sets are generated, where a flow-set consists of 200 flows. The priority, the source and destination cores are generated randomly for each flow, but in accordance with the constraint on the maximum path size, posed by the respective category to which the flow-set belongs. Each flow has a size which is randomly generated in the range $[1B - 1kB]$. We compute the worst-case traversal times of all flows with both methods, and observe the improvements in relative terms.

Figure 10 demonstrates that as the paths of the flows increase (x-axis), the relative improvements also increase (y-axis). The explanation is that short paths substantially decrease the number of contentions and interferences, hence decreasing the scenarios in which the new approach can cause improvements. In fact, even in scenarios where contentions do occur, due to short flow paths, CD sections cover large fractions of them, which has a significant impact on the improvements (Observation 2). Conversely, longer paths cause more interferences, but also longer pre-CD and post-CD sections. All these facts have a positive effect on the improvements (see Observation 2).

Experiment 3. *Improvements wrt flow and path sizes*

The objective of this experiment is to get a better insight into how both the aforementioned flow characteristics (the flow size and the path size) together influence the improvements of the new method over the existing one. We generate different flow-set categoties where we vary both parameters. Again, each category consists of 100 flow-sets, each with 200 flows with randomly generated priorities, source and destination cores. For each flow we compute the worst-case traversal times with both methods. Subsequently, for each category we observe the average improvements achieved with the new method, expressed in relative terms.

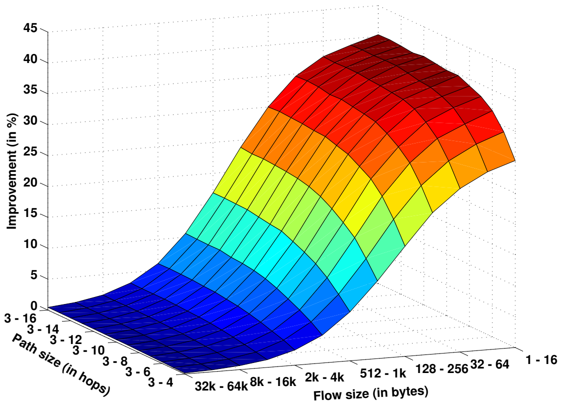


Figure 11: Improvements wrt flow and path sizes

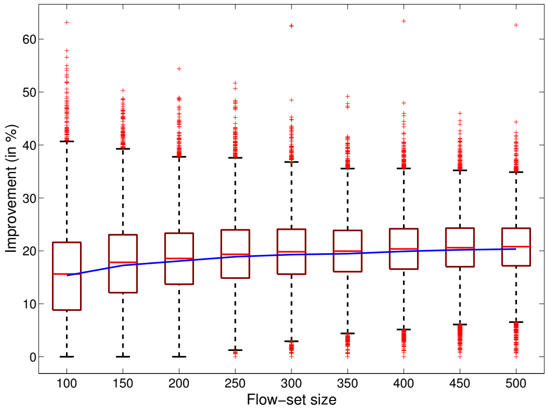


Figure 12: Improvements wrt flow-set sizes

Figure 11 shows the improvement trends (z-axis) associated with flow sizes (x-axis) and path sizes (y-axis). The improvement trends are identical to those from Experiments 1-2, inferring that the increase in the flow sizes and the decrease in the path sizes both have a negative effect on the relative improvements. This infers that small flows with long paths benefit the most from the proposed approach. Note, that a similar conclusion was reached for a small-scale numerical example with two flows (see Section 5.5 and Figure 6).

Experiment 4. *Improvements wrt flow-set sizes*

The emphasis of this experiment is on the flow-set size. In other words, we investigate how the improvement trends change with the number of flows constituting a flow-set. We generate several flow-set categories, where each category has the number of flows equal to one of these values $100, 150, ..., 500$. For each category 100 flow-sets are generated, with random priorities, source and destination cores. Moreover, each flow has a size which is randomly generated in the range $[1B - 1kB]$. Subsequently, for each flow we obtain the worst-case traversal times with both methods and compare the results in relative terms.

Figure 12 demonstrates that as the flow-set size increases, so do the improvements. The explanation is that larger flow-

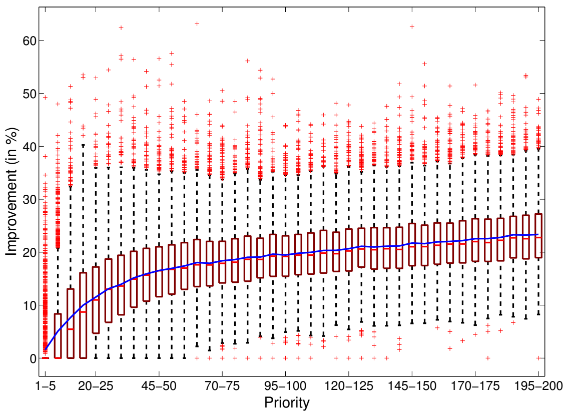


Figure 13: Improvements wrt priorities

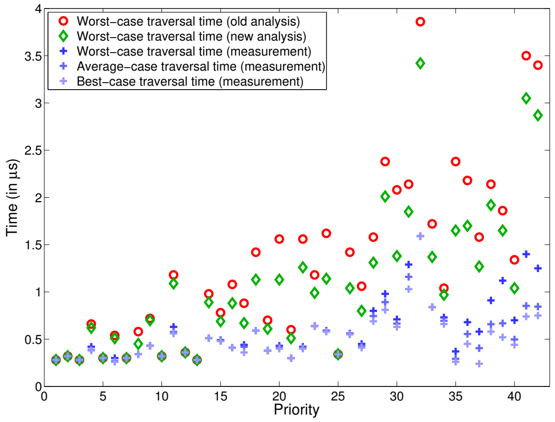


Figure 14: Analysis tightness

sets have more substantial contentions, which favours the new approach. Yet, irrespective of the flow-set size, there always exist the highest-priority flows which do not suffer interference and hence for them no improvements can be achieved. However, as the flow-set size increases, the highest-priority flows constitute smaller and smaller fraction of the entire flow-set, hence for sets with more than 200 flows these cases are below the 25^{th} percentile, and are thus classified as outliers (depicted with red crosses in Figure 12).

Experiment 5. *Improvements wrt priorities*

In this experiment we investigate how the improvement trends change with different flow priorities. We generated 100 flow-sets, each with 200 flows, where priorities, flow sizes, source and destination cores have been randomly generated. A size of each flow is a randomly generated value from the range $[1B - 1kB]$. For each flow we compute the worst-case traversal times with both methods, and express the improvements achieved by the new approach, in relative terms.

Figure 13 confirms that, as flow priorities decrease (bigger numbers on the x-axis), the relative improvements increase (y-axis). This confirms our initial assumption that the new approach does not produce significant improvements for the highest-priority flows, because these flows suffer very little interference (if at all). As flow priorities decrease, the interference that flows suffer becomes more substantial, which favours the new approach.

Experiment 6. *Analysis tightness*

The objective of this experiment is to investigate the tightness of the obtained upper-bounds on the worst-case traversal times. To achieve this, we generate a single flow-set consisting of 42 flows, and map it on a 6×6 platform with randomly generated source and destination cores. Each flow has a payload in the range $[2 - 48]$ flits, and one additional flit for a header. Moreover, each flow has a period in the range $[0.5 - 9]ms$, and a unique priority. The NoC frequency is 100MHz. We compute the worst-case traversal times of all flows with both analyses, and also simulate the execution on a cycle-accurate simulator. The simulated time is 2 hyper-periods[3]. Subsequently, we compare the analysis estimates with the values obtained via simulations, namely (i) the worst-case, (ii) the average-case and (iii) the best-case.

Results are depicted in Figure 14. It is visible that for high-priority flows both the analyses provide tight bounds. This is expected, given that these flows suffer very little interference, if at all. With the decrease in the priority (bigger numbers on the x-axis), the differences between the observed values and the analysis results become more noticeable (y-axis), because the analysis pessimism accumulates. Also, notice that the difference between the analysis estimates increases with the decrease in priorities. In fact, in some cases the proposed approach provides significantly tighter estimates, which entirely coincides with the conclusions from the previous experiments, and further motivates this work. However, the results also suggest that there is still room for improvement in the analysis, and we see this as a potential topic for future work.

7. CONCLUSIONS AND FUTURE WORK

The real-time communication analysis is the most suitable approach to determine whether a NoC-based many-core platform can accommodate the workload and always fulfil its timing requirements. In safety-critical and real-time computing areas, providing strong guarantees that all posed constraints will always be met, even in the worst-case conditions, is of paramount importance. Yet, one of the greatest challenges of analysis-based approaches is the pessimism, which can lead to a significant underutilisation of the platform and inefficient use of available resources.

In this paper we have extended the existing real-time communication analysis [15] for wormhole-switched priority-preemptive NoCs, and proposed a new, less pessimistic technique to compute tighter upper-bound estimates on the worst-case traversal times of traffic flows. Specifically, our approach reduces the pessimism of the direct interference that contending traffic flows cause to each other when competing for NoC resources. Subsequently, we have quantified the improvements (i.e. pessimism reduction, tightness) achieved with the new analysis, and also observed how these trends change with different flow parameters, namely, flow sizes, path sizes, flow-set sizes, priorities. The experiments demonstrate that the proposed approach yields significant improvements in almost all cases, while the greatest pessimism reductions are achieved in scenarios with large flow-sets, where

[3] A hyper-period is the least common multiplier of all flow-periods.

flows have small sizes and traverse long paths. These traffic characteristics correspond to control core-to-core traffic as well as to read requests and write responses in core-to-memory traffic. Given that these traffic types constitute a significant fraction of the entire NoC traffic, the proposed analysis not only can help to exploit the platform more efficiently and decrease the resource overprovisioning, but also can render many flow-sets schedulable, even though the existing analysis classified them as unschedulable. Motivated with this fact, we plan to further explore the possibilities to improve the analysis, especially in the domain of indirect interferences, which have been, for clarity purposes, kept out of scope of this paper.

8. REFERENCES

[1] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Softw. Engin. J.*, 1993.

[2] L. Benini and G. De Micheli. Networks on chips: a new soc paradigm. *The Comp. J.*, 35(1):70 –78, jan 2002.

[3] W. Dally. Virtual-channel flow control. *Trans. Parall. & Distr. Syst.*, 3(2):194 –205, Mar 1992.

[4] W. Dally and C. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Trans. Computers*, 1987.

[5] D. Dasari, B. Nikolić, V. Nelis, and S. M. Petters. Noc contention analysis using a branch and prune algorithm. *Trans. Emb. Comput. Syst.*, 2013.

[6] T. Ferrandiz, F. Frances, and C. Fraboul. A method of computation for worst-case delay analysis on spacewire networks. In *SIES*, 2009.

[7] T. Ferrandiz, F. Frances, and C. Fraboul. Using network calculus to compute end-to-end delays in spacewire networks. *SIGBED Rev.*, 2011.

[8] K. Goossens, J. Dielissen, and A. Radulescu. Aethereal network on chip: concepts, architectures, and implementations. *IEEE Design & Test Computers*, 2005.

[9] J. Hu and R. Marculescu. Energy-aware mapping for tile-based noc architectures under performance constraints. In *8th ASPDAC*, 2003.

[10] Intel. *Single-Chip-Cloud Computer.* `www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/intel-labs-single-chip-cloud-article.pdf`.

[11] N. K. Kavaldjiev and G. J. M. Smit. A survey of efficient on-chip communications for soc. In *4th Symp. Emb. Syst.*, 2003.

[12] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *The Comp. J.*, 26, 1993.

[13] B. Nikolić, H. I. Ali, S. M. Petters, and L. M. Pinho. Are virtual channels the bottleneck of priority-aware wormhole-switched noc-based many-cores? In *21st RTNS*, 2013.

[14] Y. Qian, Z. Lu, and W. Dou. Analysis of worst-case delay bounds for best-effort communication in wormhole networks on chip. In *NOCS*, 2009.

[15] Z. Shi and A. Burns. Real-time communication analysis for on-chip networks with wormhole switching. In *NOCS*, 2008.

[16] H. Song, B. Kwon, and H. Yoon. Throttle and preempt: a new flow control for real-time communications in wormhole networks. In *1997 Int. Conf. Parall. Processing*, Aug 1997.

[17] Tilera. *TILE64^{TM} Processor.* `www.tilera.com/products/processors/TILEPro_Family`.