

# Extended Recursive Analysis for Tileria Tile64 NoC Architectures: Towards Inter-NoC Delay Analysis

Hamdi Ayed

Toulouse University - IRIT - ENSEEIHT  
2 rue Charles Camichel  
Toulouse 31000, France  
hamdi.ayed@enseeiht.fr

Jean-luc Scharbarg

Toulouse University - IRIT - ENSEEIHT  
2 rue Charles Camichel  
Toulouse 31000, France  
jean-luc.scharbarg@enseeiht.fr

Jérôme Ermont

Toulouse University - IRIT - ENSEEIHT  
2 rue Charles Camichel  
Toulouse 31000, France  
jerome.ermont@enseeiht.fr

Christian Fraboul

Toulouse University - IRIT - ENSEEIHT  
2 rue Charles Camichel  
Toulouse 31000, France  
christian.fraboul@enseeiht.fr

## ABSTRACT

A heterogeneous network, where a switched-Ethernet backbone, e.g. AFDX, interconnects several end systems based on Network-on-Chip (NoC), is a promising candidate to build new avionics architectures. When using such a heterogeneous network for real-time applications, a global worst-case traversal time (WCTT) analysis is needed. In this short paper we focus on the intra-NoC communication on a Tileria TILE64-like NoC. First, we extend the Recursive Calculus (RC) to achieve tighter intra-NoC WCTT. Then, we explain how this intra-NoC WCTT analysis could be used in a compositional manner for the end-to-end inter-NoC delay analysis.

## 1 INTRODUCTION

The many-core architectures are promising candidates to support the design of hard real-time systems. They are based on simple cores interconnected using a Network-on-Chip (NoC). In Figure 1, an avionics architecture is composed of two many-core end systems. The data exchange between the cores within the same NoC is called intra-NoC communication (e.g.  $f_1$  in Figure 1), and between the cores in different NoCs, inter-NoC communication (e.g.  $f_5$  in Figure 1). The timing constraints, such as bounded delays have to be guaranteed for hard real-time avionics systems: a worst-case end-to-end delay analysis is needed. The intra-NoC communication has to take into account the wormhole switching mechanism and the possible direct and indirect blocking between communications flows. Inter-NoC communications needs a two level compositional framework: first intra-NoC communication delays due to the conflicts to reach and share the I/O (Ethernet) ports, second the inter-NoC communication delays due to the sharing of the switched-Ethernet network. Many works have been devoted to the worst-case delay analysis of a switched-Ethernet network, e.g. AFDX, using techniques such as the Network Calculus or the trajectory approach [8] [9] [5]. Moreover, an extension of the trajectory approach has been proposed for worst-case delay analysis of several CAN networks interconnected through a Switched Ethernet Backbone [10].

However, the intra-NoC worst-case traversal time (WCTT) computation strongly depends on the implemented wormhole switching mechanism. The context of this paper is a commercially existing NoC platform: Tileria TILE64. It implements the wormhole routing and a credit-based flow control in routers. A packet is divided in flow control digits (flits) of fixed size. The first flit contains routing information that define the path for all the flits of the packet. In each cycle one flit is forwarded from each router, provided that there is a free space in the input buffer of the next router. A three flits buffer is associated to each input port. The input ports are polled, based on a Round-Robin Arbitration (RRA).

Several techniques have been proposed for the WCTT analysis of a Tileria TILE64-like NoC. Among them, the Recursive Calculus (RC) [7] offers a simple way to capture the wormhole switching. The RC approach has been studied in [6], [1] and [3] to integrate the inter-release constraints and the available buffer, respectively. In this work, we describe an extended RC method dealing with both the buffer effect and the flow inter-release constraints. Then, we explain how this intra-NoC worst-case delay analysis could be used in a compositional manner for the end-to-end inter-NoC delay analysis.

## 2 INTRA-NOC TIMING ANALYSIS

The principle of the RC method [7] consists in building the set of packets that can delay (directly or indirectly) the flow under study, in the worst-case and derive a bound on its WCTT. Let's denote by  $\mathcal{S}_i = \{ \langle f_j, nb_j^i \rangle \} \cup \{ \langle f_i, 1 \rangle \}$  this set of packets. For each flow  $f_j$  impacting  $f_i$ , it gives the maximum number  $nb_j^i$  of packets may delay the flow under study  $f_i$ . Set  $\mathcal{S}_i$  is initialized with one packet from flow  $f_i$  under study, i.e.  $\mathcal{S}_i = \{ \langle f_i, 1 \rangle \}$ . The current location of this packet is its source node. This packet is forwarded till it is blocked by another flow  $f_{j_1}$  or it reaches its destination. In the later case, building of set  $\mathcal{S}_i$  is over. In the former case, one packet from  $f_{j_1}$  is added in  $\mathcal{S}_i$ , i.e.  $\mathcal{S}_i = \{ \langle f_{j_1}, 1 \rangle, \langle f_i, 1 \rangle \}$ . Its current location is the place in the network where  $f_{j_1}$  blocks  $f_i$ . For  $f_1$  in NoC 1 of the avionics architecture of Figure 1,  $\mathcal{S}_1 = \{ \langle f_4, 2 \rangle, \langle f_3, 2 \rangle, \langle f_2, 1 \rangle, \langle f_1, 1 \rangle \}$ . The scenario leading to  $\mathcal{S}_1$  is illustrated in Figure 2.

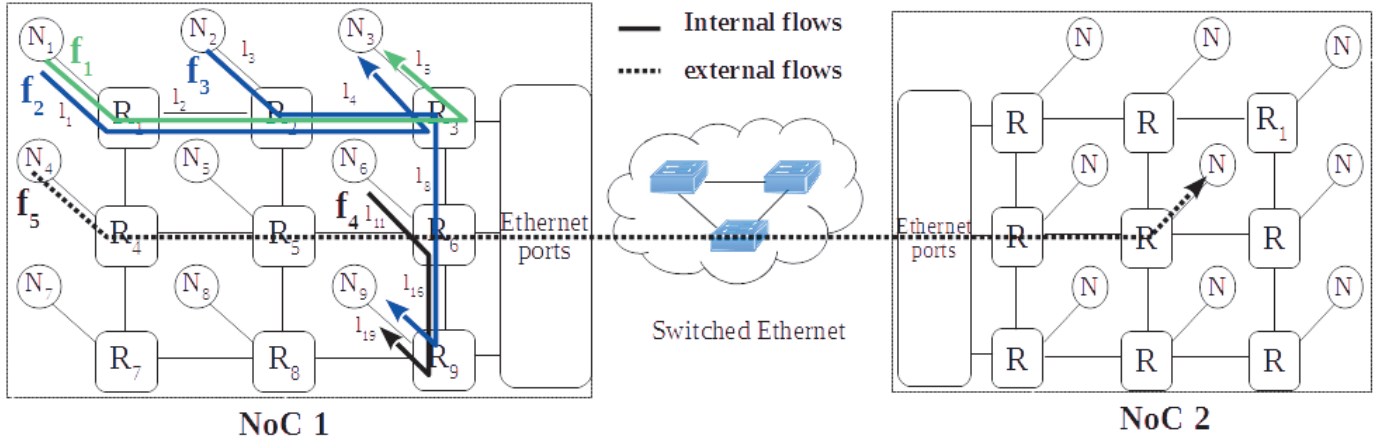


Figure 1: Illustrative example

The initial RC approach ignores the available buffer capacity in routers. It assumes that  $f_2$ ,  $f_3$  and  $f_4$  packets block  $f_1$  till they reach their destinations. This assumption simplifies the computation. However it might introduce some pessimism. let's assume, for example, a three-flit packet and a three-flit buffer (typical for Tileria Tile64). Then, a packet from  $f_3$  can be fully stored in  $R_6$  input buffer. Thus, the impact of an  $f_3$  packet on both  $f_1$  and  $f_2$  ends as soon as it leaves  $R_3$ . Since it can leave  $R_3$  even if there is a pending packet from  $f_4$ ,  $f_4$  doesn't add any extra delay for  $f_1$  and  $f_2$ . Thus the worst-case list of packets blocking  $f_1$  becomes  $\mathcal{S}_1 = \{ \langle f_3, 2 \rangle, \langle f_2, 1 \rangle, \langle f_1, 1 \rangle \}$ . The integration of available buffer space in WCTT computation has been studied in [1] (the pipeline effect). The authors establish properties to better capture the effect of buffers under wormhole routing. Based on these properties, we integrate the buffer effect in the initial RC approach. The second source of pessimism in the RC computation is due to the fact that flows are sporadic. It means that there is a minimum duration  $T_j$  between the generation times of two consecutive packets from a flow  $f_j$ . The scenario considered by the RC computation does not take into account these constraints. As illustrated in Figure 2, two packets of flow  $f_3$  are counted in the sequence of blocking packets for  $f_1$ . They are generated at time  $t'_1$  and  $t'_6$ . Assuming three flit packets for all the flows, we have  $t'_6 - t'_1 = 12$  cycles. As soon as the minimum duration  $T_3$  between two consecutive  $f_3$  packets is more than 12 cycles, this scenario cannot occur. In such a situation one single packet from  $f_3$  can delay  $f_1$ . Thus, the resulting worst-case list of packets blocking  $f_1$  becomes:  $\mathcal{S}_1 = \{ \langle f_3, 1 \rangle, \langle f_2, 1 \rangle, \langle f_1, 1 \rangle \}$ . This second source of pessimism has been addressed in [6].

The basic idea consists in enumerating all the possible sequences of blocking for a given flow, respecting the minimum inter-release constraints, and selecting the sequence leading to the WCTT. Unfortunately, the number of sequences that need to be explored grows exponentially. In order to tackle this problem, we propose an over-estimation of all the enumerated sequences. Then by integrating the minimum gap between successive packets of flows in routers, we bound the number of packet instances in each packets set. The sets are then refined in iterative manner, and the WCTT for each flow is derived. Thus, we implemented an extended RC algorithm

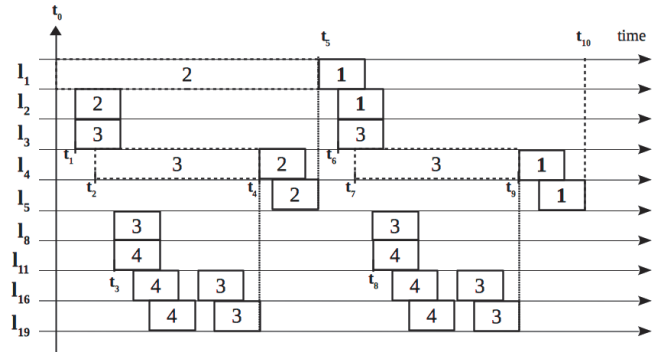


Figure 2: Basic RC sequence for  $f_1$

combining the benefits of the buffer effect and the minimum inter-release constraints of flows. We have done some experiments on  $n \times n$  2D-mesh NoC, with  $n=4$  or  $n=6$  or  $n=8$ . For these experiments, first, we obtained significant reduction on the WCTT (up to 64% compared to the initial RC) of the flows that experiment heavy indirect blocking or those who contend with flows with large inter-release periods. This can lead to guarantee the applications constraints when the classical RC method cannot.

### 3 INTER-NOG TIMING ANALYSIS: PERSPECTIVES

A Tileria-like NoC, used as a processing element within a backbone network supports two types of communication: (i) the communication between cores; and (ii) the communication between cores and the I/O interfaces to reach the backbone network. The existing works only focus on the inter-core communication and do not consider the I/O interfaces. The Tileria NoC interconnects cores but also Ethernet and DDR-SDRAM memory interfaces that are located on its edges. As each I/O interface can be accessed from the core adjacent to this interface through specific ports, each Ethernet controller of the Tile64 is connected to respectively 2 ports.

Moreover, a core can receive data directly from the Ethernet interface or through an intermediate memory controller. A similar process is used for the egress data flows where a DMA command is sent by the tile wanting to send data to the Ethernet. Efficient mapping of application on a many-core is a key issue to reduce the contention experienced by core to I/O flows [2]. Moreover the size of an Ethernet frame is several factors higher than the size of a NoC packet. Thus, several NoC packets are therefore needed to transmit to a tile an Ethernet payload. Consequently, the bridging strategies have to be optimized and accounted for when evaluating the worst-case core-to-I/O delays. Final objective will be to compute the end-to-end delay including:

- the time needed to go from a source core to the Ethernet port on the emitting NoC;
- the time needed to cross the Ethernet (AFDX) backbone;
- the time needed to go from the Ethernet port to the destination core on the receiving NoC.

One key issue will be to assess the global pessimism introduced at each level on such a heterogeneous network.

The approach proposed in this work, for Tiler TILE64-like NoC architectures, is based on the initial RC [7]. It combines the properties introduced in [1] and [6] to achieve tighter WCTT bounds for intra-NoC communication [4]. This approach, introduced for intra-chip communication (i.e. communication between cores on the same NoC), minimizes the pessimism and seems a good basis for inter-NoC end-to-end delay analysis.

## ACKNOWLEDGMENTS

This work is partially supported under CORAIL project of CORAC (Aéronautique Environnement Recherche).

## REFERENCES

- [1] L. Abdallah, M. Jan, J. Ermont, and C. Fraboul. 2015. Wormhole networks properties and their use for optimizing worst case delay analysis of many-cores. In *Industrial Embedded Systems (SIES), 2015 10th IEEE International Symposium on*. 1–10. DOI : <http://dx.doi.org/10.1109/SIES.2015.7185041>
- [2] L. Abdallah, M. Jan, J. Ermont, and C. Fraboul. 2016. Reducing the Contention Experienced by Real-Time Core-to-I/O Flows over a Tiler-Like Network on Chip. *ECRTS (2016)*, 86–96.
- [3] H. Ayed, J. Ermont, J. I. Scharbarg, and C. Fraboul. 2016. Towards a unified approach for worst-case analysis of Tiler-like and KalRay-like NoC architectures. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)*. 1–4.
- [4] Hamdi Ayed, Jérôme Ermont, Jean-Luc Scharbarg, and Christian Fraboul. 2016. Tightening worst-case timing analysis of Tiler-like NoC architectures. In *Work in Progress Session of the 28th Euromicro Conference on Real-Time Systems (ECRTS 2016)*.
- [5] H. Bauer, J. Scharbarg, and C. Fraboul. 2010. Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach. *IEEE Transactions on Industrial Informatics* 6, 4 (Nov 2010), 521–533. DOI: <http://dx.doi.org/10.1109/TII.2010.2055877>
- [6] Dakshina Dasari, Borislav Nikolić, Vincent Nélis, and Stefan M. Petters. 2014. NoC Contention Analysis Using a Branch-and-prune Algorithm. *ACM Trans. Embed. Comput. Syst.* 13, 3s, Article 113 (March 2014), 113:1–113:26 pages.
- [7] Thomas Ferrandiz, Fabrice Frances, and Christian Fraboul. 2009. A method of computation for worst-case delay analysis on SpaceWire networks. In *Proc. of the 4th Intl. Symp. on Industrial Embedded Systems (SIES)*. Lausanne, Switzerland, 19–27.
- [8] F. Frances, C. Fraboul, and J. Grieu. 2006. Using Network Calculus to optimize the AFDX Network. Proceedings of the 3rd European Congress Embedded Real Time Software, Toulouse.
- [9] H. Bauer, J. L. Scharbarg, and C. Fraboul. 2009. Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network. *IEEE Conference on Emerging Technologies and Factory Automation (2009)*, 1–8.
- [10] X. Li, J. L. Scharbarg, and C. Fraboul. 2012. Worst case delay analysis on a real-time heterogeneous network. *7th IEEE International Symposium on Industrial Embedded Systems (2012)*.