Analytical Model for Performance Evaluation of Token-Passing-Based WiNoCs

Ibrahim Krayem, Joel Ortiz Sosa, Cédric Killian, Daniel Chillet

Analytical Model for Performance Evaluation of Token-Passing Based WiNoCs

Ibrahim Krayem, Joel Ortiz Sosa, Cédric Killian, and Daniel Chillet
Inria, IRISA, Univ Rennes
Lannion, France
{first-name}.{last-name}@irisa.fr

Abstract

Recent advances in technology integration have introduced new on-chip interconnects, such as wireless Network-on-Chips (NoCs), making the design space too large to be efficiently explored with time-consuming standard simulators. In this paper, we propose an analytical model based on queuing theory to evaluate the latency of manycore architecture interconnects. We consider a hybrid interconnection that utilizes electrical and wireless NoCs for both intra- and inter-cluster communications. The results demonstrate that our proposed model significantly reduces the simulation execution time by up to $500\times$ while maintaining an error rate of less than 5% compared to the Noxim cycle-accurate simulator.

Index Terms

Wireless Network-on-Chip, Performance Analysis, Latency, Queuing theory, Token Passing.

I. Introduction

Several years ago, the rise of manycore architectures became greatly noticeable, primarily due to the implementation of massive parallelism on a single chip. With the integration of thousands of heterogeneous cores, they allow huge parallel computation capabilities suitable for High Performance Computing (HPC) [**?**]. These parallelism capabilities obviously generate an enormous amount of data exchanges making the communication medium a key element in the overall system performance. Recent advances in silicon integration offer opportunities to use emerging on-chip interconnects such as Wireless Network-on-Chips (WiNoCs) [**?**]. Utilizing the very wide frequency band offered by the Complementary metal–oxide–semiconductor (CMOS) process is the key benefit of WiNoCs. On-chip wireless signal speed propagation allows long distance communications without latency increase, whereas electrical Network-on-Chips (NoCs) suffers from more router crossings. WiNoCs can be used alone, hence all data exchanges between two nodes are done with wireless communications. However, the wireless bandwidth is generally lower than electrical NoCs, hence they can be combined as a hybrid NoC. In this case, the communications can either be handled by wireless links for long-distance communications or by electrical routers for the short ones.

TABLE I: Summary comparison of different analytical models.

	[7]	[10]	[11]	[12]	[13]	This work
Queue	M/G/1	M/D/1	G/G/1	G/G/1	General Geometric distribution	M/G/1
Arbitration	Round Robin	Round Robin	Fixed Priority	Round Robin	Weighted Round-Robin	Round Robin
Implementation	C++	C++	C++	Matlab	C++	Python
Validation tools	Wormsim	OPENET	-	Booksim	-	Noxim
Hybrid NoCs	✗	✗	✗	✗	✗	✓

In the literature, several wireless and hybrid architectures have been proposed, as illustrated by Fig. **??**, highlighting two different architectures. In Fig. **??**-a, the manycore is divided into clusters of cores, where intra-cluster communications are conducted with electrical NoCs composed of electrical routers, while inter-cluster communications are only possible through wireless routers that integrate an antenna [**?**]. In Fig. **??**-b, the manycore is fully connected with a classic 2-D mesh NoC, while the wireless routers are shared by several cores and accessible by using the electrical routers [**?**]. Several other architectures exist, such as architectures with one dedicated wireless router per core [**?**], or architecture with a heterogeneous placement of wireless routers within a classic electrical NoC [**?**]. It is worth mentioning that, for each communication infrastructure, there are several routing protocols, along with numerous technological parameters and related performances. Each interconnect architecture utilizing wireless interconnect has its own advantages and drawbacks, but they are generally not compared in the literature. Indeed, the design space for exploring such architectures is complex and time-consuming, as the performance evaluation mostly relies on simulations that are slow processes, particularly for large multi-core systems. On the other hand, mathematical modeling offers a good trade-off between computation times and accuracy level. Analytical models for on-chip interconnects are based on the queuing theory and provide performance metrics such as average buffer utilization and average packet latency [**?**]. However, current analytical models only support homogeneous interconnects and are not adapted for hybrid ones. The main features of hybrid models that are not satisfied by conventional models are: i) wireless channel access differs from classic electrical router and is based on a token passing medium access control, ii) communication bandwidths between

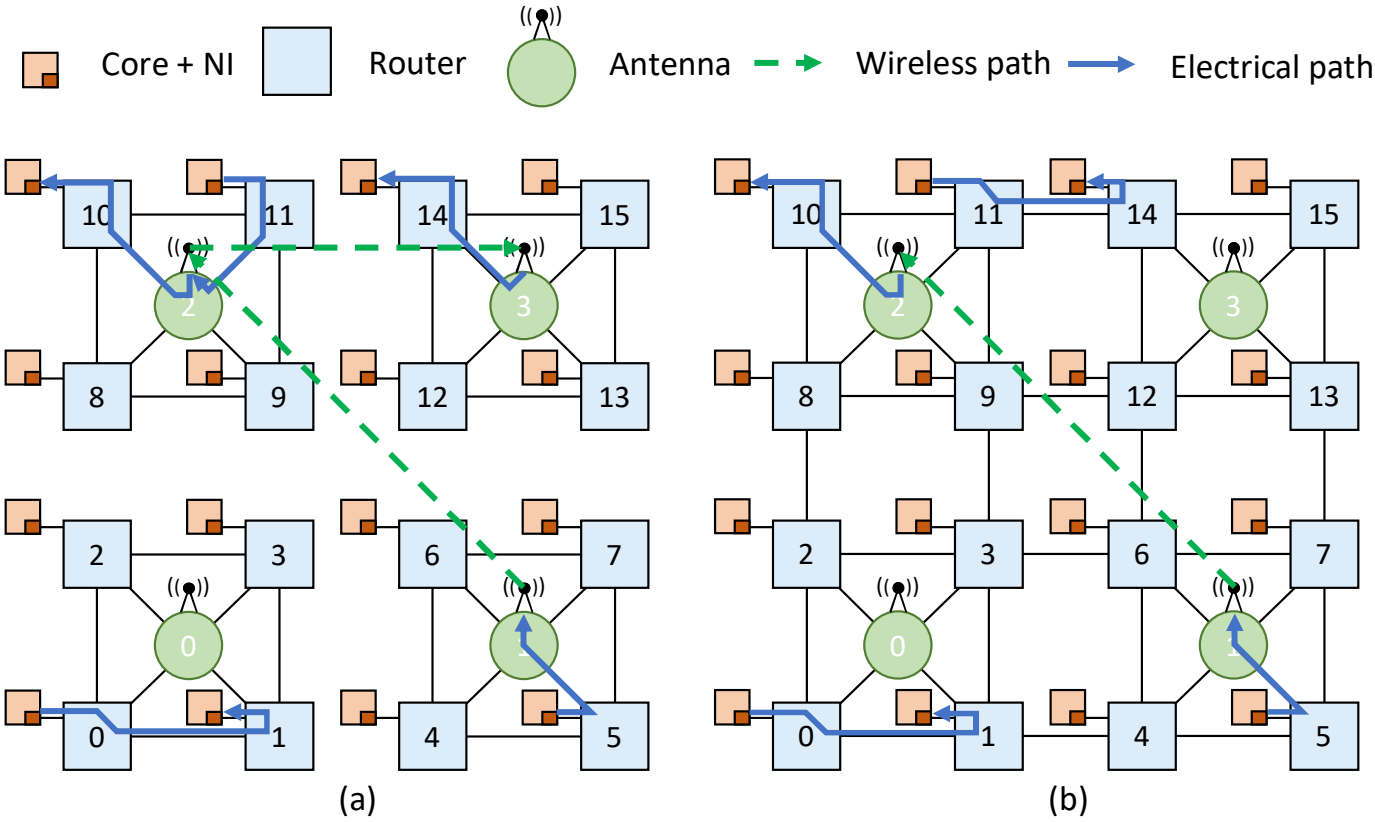


Fig. 1: Example of Hybrid NoC architectures [**?**], [**?**].

electrical and wireless mediums are heterogeneous, iii) electrical and wireless routers latencies (service time) may differ. The proposed analytical model addresses these limitations, and, from our knowledge, this is the first analytical model for a hybrid on-chip interconnect.

Herein, we propose an analytical model, based on the $M/G/1$ queuing model, to evaluate the average packet latency and the network throughput of a hybrid NoC, where packets can be routed on electrical, wireless, and mixed paths between cores. This model considers Poisson distribution for communication traffic. Routers are based on round-robin arbitration, while wireless routers are based on token passing channel access. The service time and channel capacities of electrical and wireless can be set separately to meet real hardware designs [**?**]. This model provides a degree of heterogeneity usefull to explore future design paradigms, such as chiplet or accelerators-based architectures [**?**].

The rest of this paper is organized as follows: section **??** reviews the related work on NoC performance evaluations. In section **??**, we describe our hybrid wired/wireless interconnection model, for computing packet latency and the network throughput. In order to validate our approach, the experimental evaluations and the results are shown and discussed in Section **??**. Finally, we conclude the paper in Section **??**.

II. Related Work

There are two main categories of NoC performance evaluation approaches: simulation and non simulation based methods. In simulation based methods, several simulators have been developed [**?**]. It is worth noting that amongst all proposed simulators, Catania et al. [**?**] propose an open-source cycle-accurate NoC simulator supporting the WiNoC. On the other hand, the queuing theory is an effective technique to analyze the performance of NoC. Generally speaking, it is a mathematical study of how waiting lines originate, work, and get congested. It is represented by three parameters $P_1/P_2/P_3$:

- P_1: the nature of the arrival process, e.g. M for the Poisson process, G stands for a general distribution of inter arrival times, D stands for deterministic inter-arrival times. In NoCs, it is how packets are injected.
- P_2: the nature of the probability distribution of the service times, e.g. G for a general distribution, D for a deterministic distribution, M for a exponential distributions. In NoCs, it is how packet sizes vary.
- The number of servers is indicated by the third parameter. In NoCs, it is how many outputs route a given packet.

An analytical model was proposed by Ogras et al. [**?**] for wormhole-switched NoCs. This proposal is based on a $M/G/1$ queuing model with a relative error around 9%. In [**?**], Zhang et al. proposed an analytical model for the evaluation of the performance of a NoC with constant service time system, based on a $M/D/1$ queuing model with a mean of error 7.87%.

Kiasari et al. proposed a flexible $G/G/1$ queuing model for estimating the average packet latency with less than 10% error in NoCs with arbitrary network topology and deterministic routing [**?**]. More recently, in order to represent packet flow in a NoC as an open feed-forward queuing network, Bhaskar and Venkatesh [**?**] proposed a mathematical model, with 10 and 15% of errors for NoCs with 36 routers and 64 routers respectively. Weighted Round-Robin arbitration has been proposed for priority based NoCs [**?**].

Table **??** summarizes the before-mentioned works, and also highlights the validation tool used. These models only address homogeneous Electrical Network-on-Chips (ENoCs), hence are not adapted for hybrid NoCs where WiNoC accesses are granted with a token passing channel access, and where throughput and latency of electrical and wireless interconnects may differ. It has to be noticed that some features of existing analytical models may be meaningful and complementary for future emerging hybrid interconnects, for instance using priority packet to reach a wireless router. However, it requires to also rethink the hardware architecture implementation and the simulator used to validate the analytical model, hence is out of scope of this paper.

III. Hybrid interconnection models

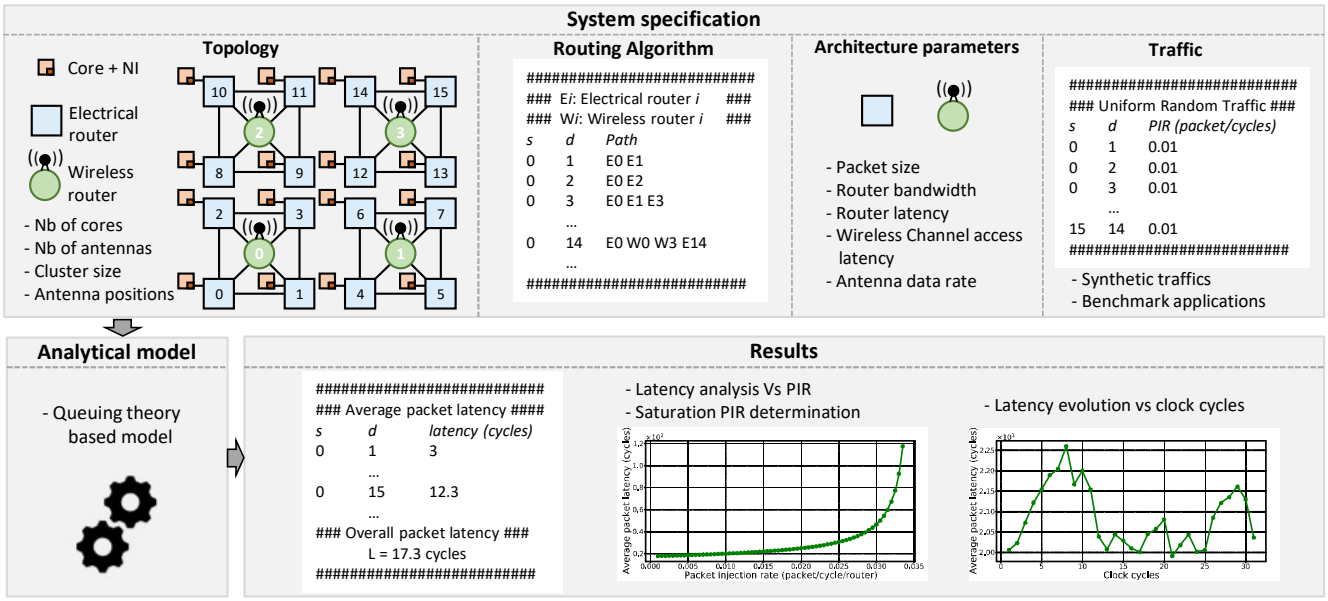


Fig. 2: Overview of the analysis flow based on our analytical model.

In this section, we present the proposed analytical model to estimate the average packet latency for a hybrid interconnection containing wired and wireless interconnections. The section **??** presents a global overview of the analysis flow, while the section **??** presents a comprehensive description of our analytical model.

A. Overview of the analysis flow

Fig. **??** illustrates the analysis flow to evaluate the communication latency of a hybrid wireless-electrical NoC according to system specifications. The flow takes as inputs the interconnect architecture, the routing paths from sources to destinations (i.e. routing algorithm), and traffic between cores. The architecture is described by the amount of cores, routers, and antennas, and how they are interconnected together. The routing algorithm specifies the path selected between any pair of source s and destination d, and with respect to the architecture. The design flow relies on architectural parameters since they impact the communications performance. For instance, packet size, router bandwidth, antenna data rate, and wireless channel access latency are considered. The latency of communications within a NoC is impacted by the injected traffic. Our design flow takes as input either synthetic traffic, such as uniform random or transpose communication patterns, or traffic from real application benchmarks such as the Parsec Benchmark suite [**?**]. The aim of the flow is to evaluate the communication latency. Our analytical model, based on queuing theory, is in charge of this computation and is detailed further in section **??**. As results, we obtain the average latency for each couple of source and destination, and also the overall latency. From this, it is easy to draw the classic latency versus Packet injection rate (PIR) plot which represents the core metric to evaluate a NoC, see section **??**. Statistics are obtained, such as the amount of packets using antennas, minimal and maximal latency with respect to PIR, which are not highlighted in this paper for reason of page limitation. It is also possible, based on traces analysis of applications, to plot the communication latency during the execution time, see section **??**.

B. Analytical Model

We propose an analytical model based on the $M/G/1$ queuing model and we consider wormhole routing. The model is generic and compatible with any kind of hybrid interconnect, such as illustrated in Fig. **??**. This model is based on a set of parameters defined in Table **??**.

TABLE II: List of the parameters used in our model

Notation	Definition
m, n	The number of input and output ports
N_c	The number of cores
N_a	The number of antennas (equal to the number of wireless routers)
N_f, ρ	The number of flits in a packet with ρ (bits) each flit
a_i, A	a_i denotes the average number of packets at input buffer i, and A the vector of average number of packets for the entire routing element ($packets$)
T, $\overline{T}$, $\overline{T^2}$	T denotes the packet service time, $\overline{T}$ and $\overline{T^2}$ denotes the first and the second order moments of service time respectively ($cycle$)
λ_i, Λ	λ_i denotes the arrival rate at input buffer i, and Λ the arrival rates to the entire routing element ($packet/cycle$)
f_{ij}	Forwarding probability: probability that a packet arrives at input i and leaves the routing element through output j
γ_{ij}	Arrival rate at input i and routed toward the output j ($packet/cycle$)
β	Time needed by the token to pass between consecutive antennas ($cycle$)
σ	The average delay required to access the wireless channel ($cycle$)
q	Waiting time in queue, q^w and q^e for wireless and electrical routers respectively ($cycle$)
r_i, R	r_i denotes the residual service time at input buffer of input i and R the residual time for the entire routing element ($cycle$)
δ_{ij}, Δ	δ_{ij} denotes the contention probability between input i and input j and Δ the contention matrix for the entire routing element
S^e, S^w	Electrical and wireless routers service time ($cycle$)
B	Electrical router bandwidth (bit/s)
D	Antenna data rate (bit/s)
θ	Frequency ($Hz => cycles/s$)
x_{sd}	Rate of the traffic transmitted from source s to destination d ($packets/cycle$)
L_{sd}	Average latency for any packet from source s to destination d ($cycle$)
L	Overall average packet latency ($cycle$)
$PIR, SPIR$	PIR denotes the packet injection rate and SPIR denotes the saturation packet injection rate ($packet/cycle/router$)
Th	Network throughput ($packet/cycle/router$)

In these hybrid NoCs, different types of communication paths exist between a source core s and a destination core d.

The path can be i) only with electrical routers, ii) only with wireless routers, and iii) a mix of both electrical and wireless routers, as illustrated with Fig.**??**.

To compute the average latency between a source s and a destination d, the proposed model first calculates the latency to cross each routing element. Then, with respect of injected traffic pattern, it calculates the probability of congestion which increases the overall latency. A routing element can be an electrical or a wireless router. An electrical router has the same amount of input and output ports, with respect to number of connected neighbors. A wireless router has one or several electrical inputs and outputs, with respect to the amount of connected neighbors, and one input and output connected to an antenna. In the following, we first introduce the model by illustrating how packets go through a simple one-input one-output routing element, hence without possible congestion, then we describe the complete model which can be applied to any routing elements with any number of inputs and outputs.

1) A simple one-input one-output routing element: The Fig. **??** presents the model used to evaluate the traffic bandwidth for a service based on a single pair of input and output of a routing element.

In this case, using the *Little's Theorem*, the average number of packets a in the system is computed from the waiting time q within a queue and the traffic arrival rate λ of packets into the system:

$$a = \lambda q \tag{1}$$

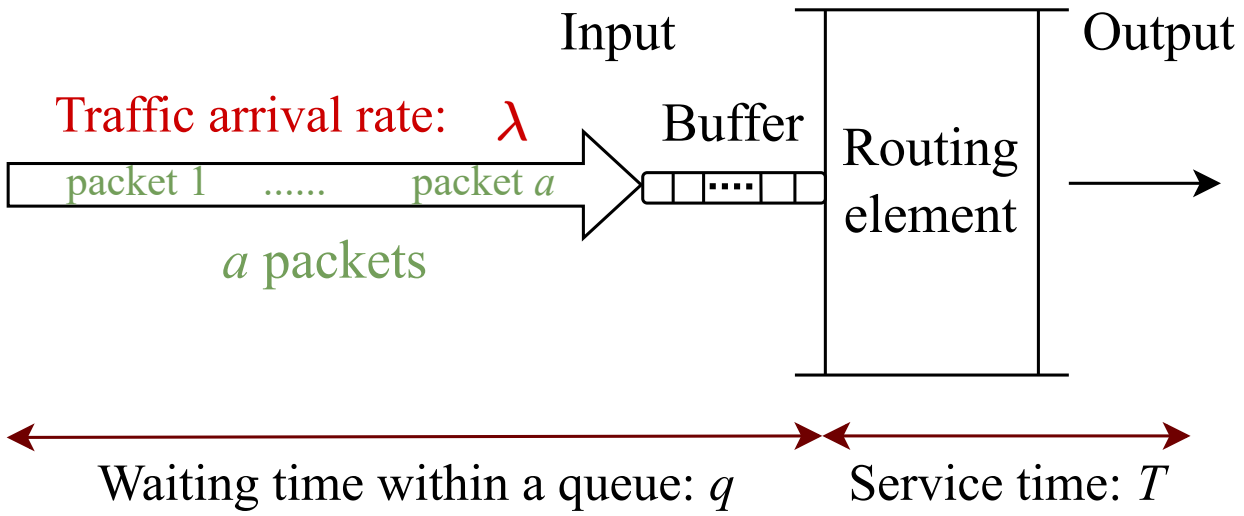


Fig. 3: Example of a single queue routing element.

On the other hand, we can calculate the waiting time q of each packet by the *Pollaczek-Khinchin* (P-K) formula depending on the queuing model. For the $M/G/1$ model, the waiting time q is given by:

$$q = \frac{r}{(1 - \lambda T)} \tag{2}$$

where r is the *residual service time*, the time a new arriving packet needs to wait until the service time of already present packet is complete:

$$r = (\lambda \overline{T^2})/2 \tag{3}$$

T, $\overline{T}$ and $\overline{T^2}$ are respectively *the packet service time*, the first and the second order moments of service time without the queuing delay respectively. T is given by:

$$T = \begin{cases} S^e + \frac{N_f \times \rho \times \theta}{B} & \textit{Electrical router} \\ S^w + \frac{N_f \times \rho \times \theta}{D} + \sigma & \textit{Wireless router} \end{cases} \tag{4}$$

where S^e, S^w are electrical and wireless router service times respectively, N_f is the number of flits in the packets, ρ is the size of flits, B and D the electrical router bandwidth and the antenna data rate, and σ is the *Average delay token*. We assume a round-robin token circulation among N_a antennas, where σ is the average delay corresponding to the number of cycles required to access the wireless channel. In the best case, an antenna requiring to send a packet already posses the token and will directly send it without any delay. On the other hand, the worse case is to wait a full round to receive the token. If the time needed by the token to pass between consecutive antennas is β *cycles*, the average delay σ can be stated as:

$$\sigma = \beta \times N_a/2 \tag{5}$$

2) General case: In a NoC, there are m inputs and n outputs for a routing element as shown in Fig. **??**. Hence, the equations **??** and **??** have to be computed as matrices. For instance, for a routing element y, the average number of packets, the arrival rates and the residual service times are respectively defined by:

$$A^y = \begin{bmatrix} a_1^y & a_2^y & \dots & a_m^y \end{bmatrix}^T \tag{6}$$

$$\Lambda^y = diag(\lambda_1^y, \lambda_2^y, ..., \lambda_m^y) \tag{7}$$

$$R^y = \begin{bmatrix} r_1^y & r_2^y \dots r_m^y \end{bmatrix}^T \tag{8}$$

where the average number of packets for a routing element y is done by the equilibrium condition [**?**]:

$$A^y = (I - \Lambda^y \Delta^y)^{-1} \Lambda R^y \tag{9}$$

Contention probability: Δ is the contention matrix, where δ_{ij} is contention probability between input i and input j:

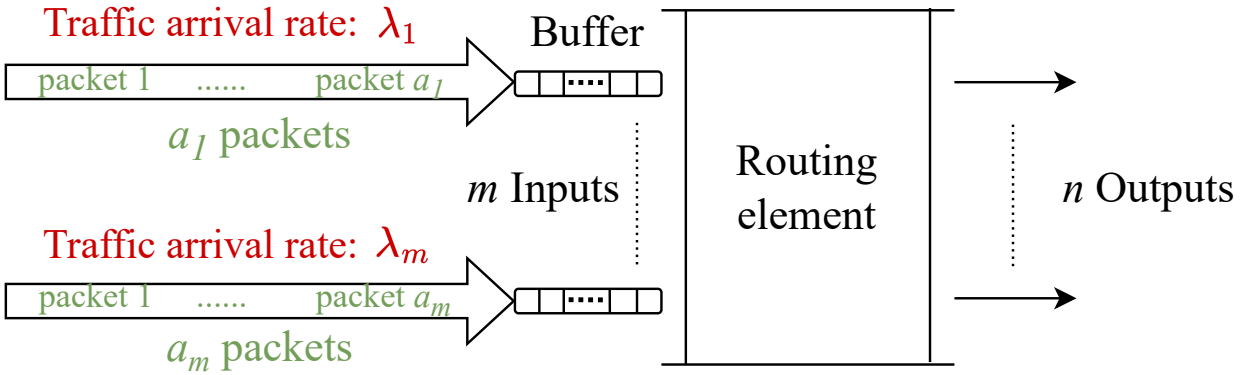


Fig. 4: General case of service managing m inputs and n outputs.

$$\Delta^y = \begin{bmatrix} 1 & \delta^y_{12} & \dots & \delta^y_{1m} \\ \delta^y_{21} & 1 & \dots & \delta^y_{2m} \\ \dots & \dots & \ddots & \dots \\ \delta^y_{m1} & \delta^y_{m2} & \dots & 1 \end{bmatrix}_{m \times m}$$

with

$$\delta^y_{ij} = \begin{cases} \sum_{k=1}^{m} f^y_{ik} f^y_{jk} & i \neq j \\ 1 & i = j \end{cases} \tag{10}$$

Where f_{ij} is the forwarding probability, the probability that a packet arrives at input i and leaves the routing element through output j:

$$F^y = \begin{bmatrix} 0 & f^y_{12} & \dots & f^y_{1m} \\ f^y_{21} & 0 & \dots & f^y_{2m} \\ \dots & \dots & \ddots & \dots \\ f^y_{m1} & f^y_{m2} & \dots & 0 \end{bmatrix}_{m \times m}$$

$$f^y_{ij} = \begin{cases} \frac{\gamma^y_{ij}}{\sum_{k=1}^{m} \gamma^y_{ik}} & i \neq j \\ 0 & i = j \end{cases} \tag{11}$$

γ^y_{ij} is the arrival rate at input i and routed toward the output j. The traffic arrival rate at input j at routing element y:

$$\lambda^y_j = \sum_{\forall s} \sum_{\forall d} x_{sd} \Re(s, d, y, j) \tag{12}$$

Where $\Re$ is the routing function:

$$\Re(s, d, y, j) = \begin{cases} 1 & \textit{if } (y, j) \textit{ in } \Pi_{sd} \\ 0 & \textit{otherwise} \end{cases} \tag{13}$$

Where Π_{sd} is the set of routers pairs traversed by the packet. Then, the average packet latency is done by:

$$L_{sd} = L_e + L_w \tag{14}$$

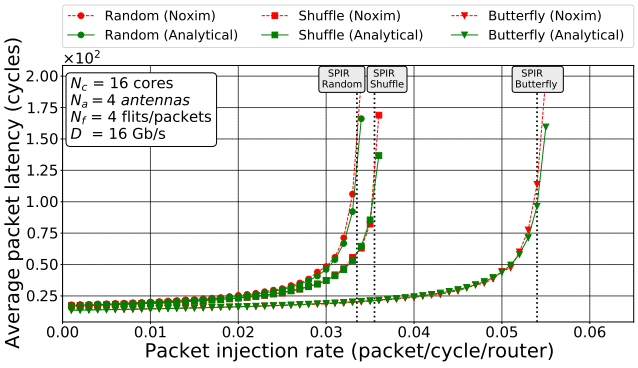
Where L_e and L_w are given by:

$$L_e = \begin{cases} 0 & \textit{if path fully wireless} \\ \sum_{(i,j) \in \pi_{sd}} (q^e_{ij} + S^e) + \frac{N_f \times \rho \times \theta}{B} & \textit{otherwise} \end{cases} \tag{15}$$

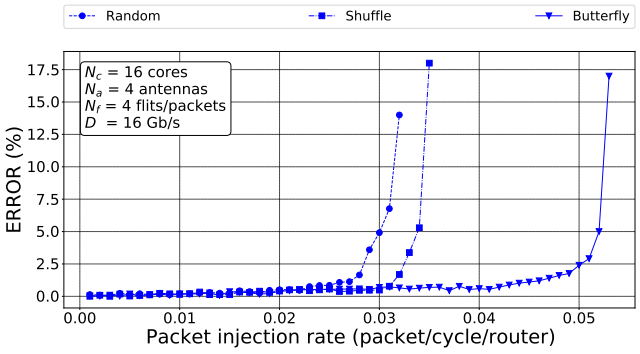
$$L_w = \begin{cases} 0 & \textit{if path fully electric} \\ q^w_{src} + q^w_{dst} + 2S^w + \frac{N_f \times \rho \times \theta}{D} + \sigma & \textit{otherwise} \end{cases} \tag{16}$$

Finally the overall average packet latency L is given by:

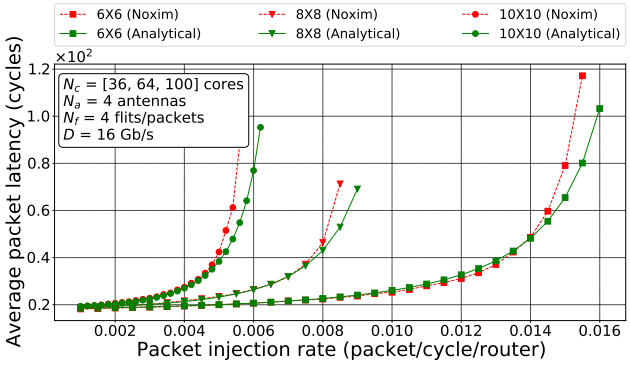
$$L = \frac{\sum_{\forall s,d} (x_{sd} \times L_{sd})}{\sum_{\forall s,d} x_{sd}} \tag{17}$$



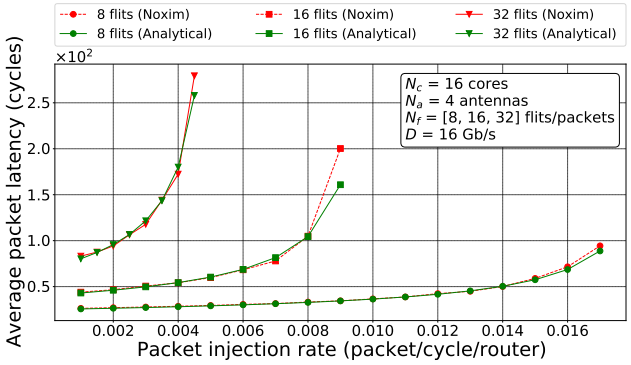
(a) Traffic patterns exploration.



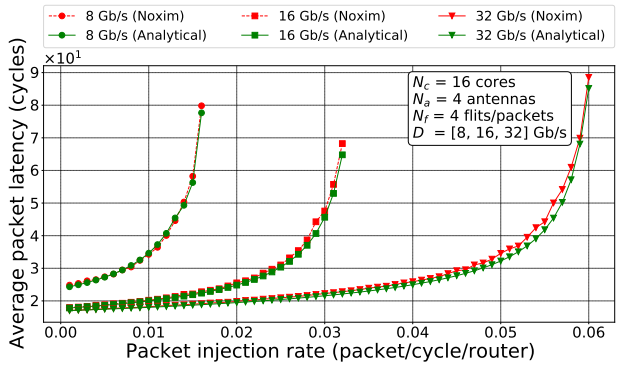
(b) Analytical model Vs Noxim error.



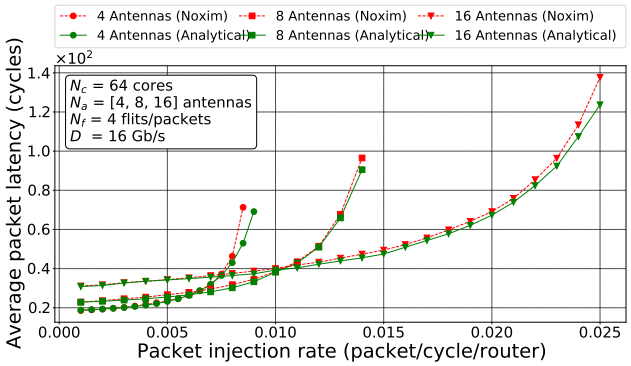
(c) Architecture size exploration.



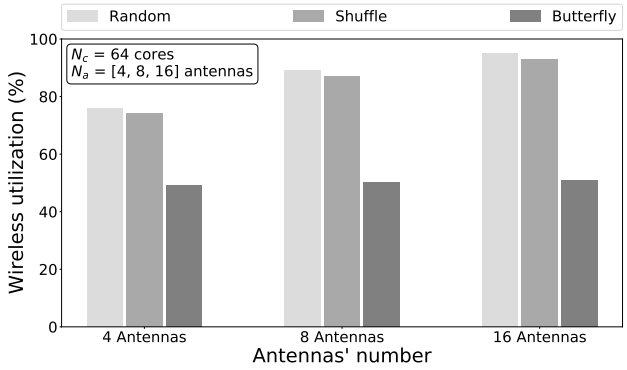
(d) Packet size exploration.



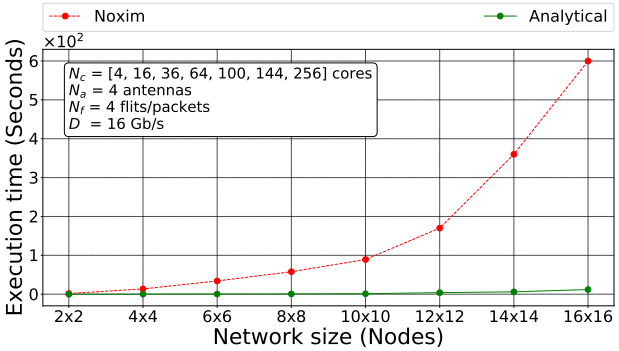
(e) Wireless datarate exploration.



(f) Exploration of number of antennas.



(g) Wireless utilization rate.



(h) Execution time evaluation.

Fig. 5: Validation of the analytical model for different parameters: (a) and (c) to (f) represent the average packet latency Vs. PIR, and (b) represents the percentage of error of (a) between our analytical model and the Noxim simulator, (g) shows the Wireless utilization rate with respect to the number of antennas, and (h) shows the execution time with respect to architecture size.

IV. Experimental Evaluations

In Section **??**, we present the experimental setup considered for our study. The model's validation is performed for synthetic traffic and application benchmarks in Sections **??** and **??**, respectively. Network throughput validation is carried out in Section **??**. Additionally, we analyze the execution time and evaluation error in comparison to the Noxim cycle-accurate simulator, and the results are presented in Sections **??** and **??**, respectively.

A. Experimental Setup

To validate our model, we compare our results with the Noxim cycle-accurate simulator [**?**]. Noxim incorporates a hybrid NoC combining both electrical and wireless routers.

In order to limit methodological bias, and ensure repeatability of results, all the results are obtained by respecting the Noxim hybrid topology, which corresponds to Fig. **??**-a. Moreover, the primary goal of this paper is not to determine the optimal topology for hybrid on-chip interconnect or to compare ENoCs and WiNoCs, but to propose a method that enables rapid performance analysis and exploration of such architectures. For comparative studies, please refer to [**?**], [**?**].

The N_c core architecture is divided in clusters of equal size, in accordance with the number of implemented antennas N_a. Each cluster is organized in a 2-D mesh ENoC with one wireless router shared among all the cores from within the same cluster. The wireless router is then connected to each electrical router from the cluster, and can be accessed by a core after crossing the associated electrical router. Intra-cluster communications are handled by the ENoC using the XY routing algorithm, while inter-cluster communications are achieved by using the WiNoC with a token-passing channel access scheme. For instance, in a configuration with $N_c = 16$ cores and $N_a = 4$ antennas, where the architecture closely resembles Fig. **??**-a. In this setup, the architecture is decomposed into 4 clusters that are not electrically interconnected.

This architecture may be likened as a chiplet-based one where inter- and intra-chiplet communications are using a different interconnect.

We considers routers without virtual channels and a router bandwidth B set to 32 Gbit/s (32-bit flits at 1GHz). The service times S^e and S^w are both set to 2 clock cycles, while the token passing latency β is set to 1 clock cycle. The simulations were conducted on an Ubuntu $18.04.5$ LTS Linux distribution, executed on a 48-cores Intel© Xeon(R) Silver 4214 CPU @ 2.20GHz. Moreover, the proposed analytical model is implemented using Python, and the results are compared with those obtained from the Noxim cycle-accurate NoC simulator [**?**].

It should be noted that any analytical method is deterministic in nature, which guarantees consistency in the obtained results without requiring multiple iterations. Simulation-based approaches, on the other hand, can exhibit a degree of variability. Nevertheless, to guarantee correctness on the analysis, we defined the simulation time that allows latency convergence toward a stable value. This value depends on the PIR value. Indeed, a low PIR requires more clock cycle to be simulated than higher PIR, as fewer packets are injected per cycle.

B. Parameters exploration with synthetic traffics

In this section, we explore the correctness of our model by exploring several parameters. We vary the traffic pattern, packet size, wireless data rate, and number of antennas. Each parameter value is summarized within the figures. The results are shown in Fig. **??** where we plot the average packet latency versus the PIR (packet/core/cycle). For each result, we vary the PIR starting from 0.001 in increments of 0.001, until crossing the Saturation Packet Injection Rate (SPIR) (defined as $10\times$ the zero-load latency). Moreover, we simulate 100,000 clock cycles into Noxim for each PIR value, while our analytical model only requires a single computation as it only depends on the PIR.

In Fig. **??**, we can see results from the analytical model and the simulator exhibit the same NoC behaviour. At low PIR the average latency is low (around 17 clock cycles), and the NoC starts reaching the saturation while the PIR increases. The NoC saturates faster for the uniform random traffic pattern, which creates more congestion. To better appreciate the correctness between the analytical model and simulator results, Fig. **??** plots the latency difference (error in %) for each PIR of Fig. **??**. We can see that the error remains below 2% until reaching the saturation, where a maximum error of 12% is found. It is important to note that the increase in error when the network gets highly congested is common to all analytical models [**?**], [**?**], [**?**], [**?**], [**?**]. This is the trade-off for extremely short computation times compared with very long simulation times (see Section **??**).

The proposed model is able to compute the routing latency for hybrid electrical-wireless paths as well as fully electrical paths, such as intra-cluster communications illustrated in Fig. **??**. For instance, on the results of Fig. **??** with the execution of traffic random, 76% of the packets used a hybrid path, while 24 % use a fully electrical path. Compared to Noxim cycle-accurate simulator, the latency evaluation accuracy is for the two types of paths are accurate.

If we focus on determining the SPIR value, these values are, respectively for the analytical model and Noxim: 0.0541 and 0.0514 for a butterfly traffic, 0.0346 and 0.0331 for a shuffle traffic, 0.0325 and 0.0317 for a uniform random traffic. Hence, on average the difference is 4.3% in determining the SPIR.

Due to space limitation, we do not present the error evaluation for the other explored parameters. however, it is evident from the figures that our model performs well in terms of accuracy compared to the cycle-accurate simulator.

The following results are performed with a uniform traffic pattern. Fig **??** highlights the impact of architecture sizes. We can see that the 100 cores architecture get congested at a lower PIR compared to 64 and 36 cores architectures. This can be attributed to the fact that more cores are sharing a wireless router, which increases the congestion in using wireless communication as it is the only way to communicate between clusters in this type of topology. The purpose of this paper is to provide a model for efficiently determining the latency of hybrid NoCs, not to identify the optimal architecture.

In Fig. **??**, it is not surprising that increasing the number of flits per packet leads to higher latency and a reduction in the saturation PIR value. Contrarily, increasing the antenna data rate allows for an increase in the saturation PIR, as shown in Fig. **??**. Fig. **??** shows the impact on the number of antennas within a 64 cores architecture. Surprisingly, increasing the amount of antenna increases the latency at low PIR. This is due to the token passing channel access. Indeed, with 16 antennas and 64 cores (i.e. cluster of 4 cores with 1 antenna), each communication needs to get the token before being able to communicate, then the latency is around 35 clock cycles. However the NoC saturates with a higher PIR as the number of hops between any source and destination is kept low thanks to a wireless communication. Fig. **??** illustrates the utilization of the wireless interconnects for different traffic patterns. For instance, for random traffic with an 8×8 architecture size and 8 antennas, it was noticed that 89% of packets communicate via wireless interconnects, and 11% with electrical ones.

C. Network throughput

It is straightforward to evaluate the throughput Th (number of packet per cycle) of an architecture with the proposed analytical model, as it may be directly derived from the latency evaluation. We proceed as follow: i) compute the latency Vs PIR, ii) determine SPIR (10× the zero-load latency for a given traffic), iii) we apply the Equation **??**:

$$Th = \begin{cases} PIR & if\ PIR < SPIR \\ SPIR & if\ PIR \geq SPIR \end{cases} \tag{18}$$

In the Noxim simulator, the throughput is directly given in the output simulation results.

The Fig. **??** shows network throughput Vs PIR of a 16 cores architecture with the same parameters as for Fig. **??**, and the SPIR values determined in section **??**. This result validates our approach to compute both SPIR and throughput.

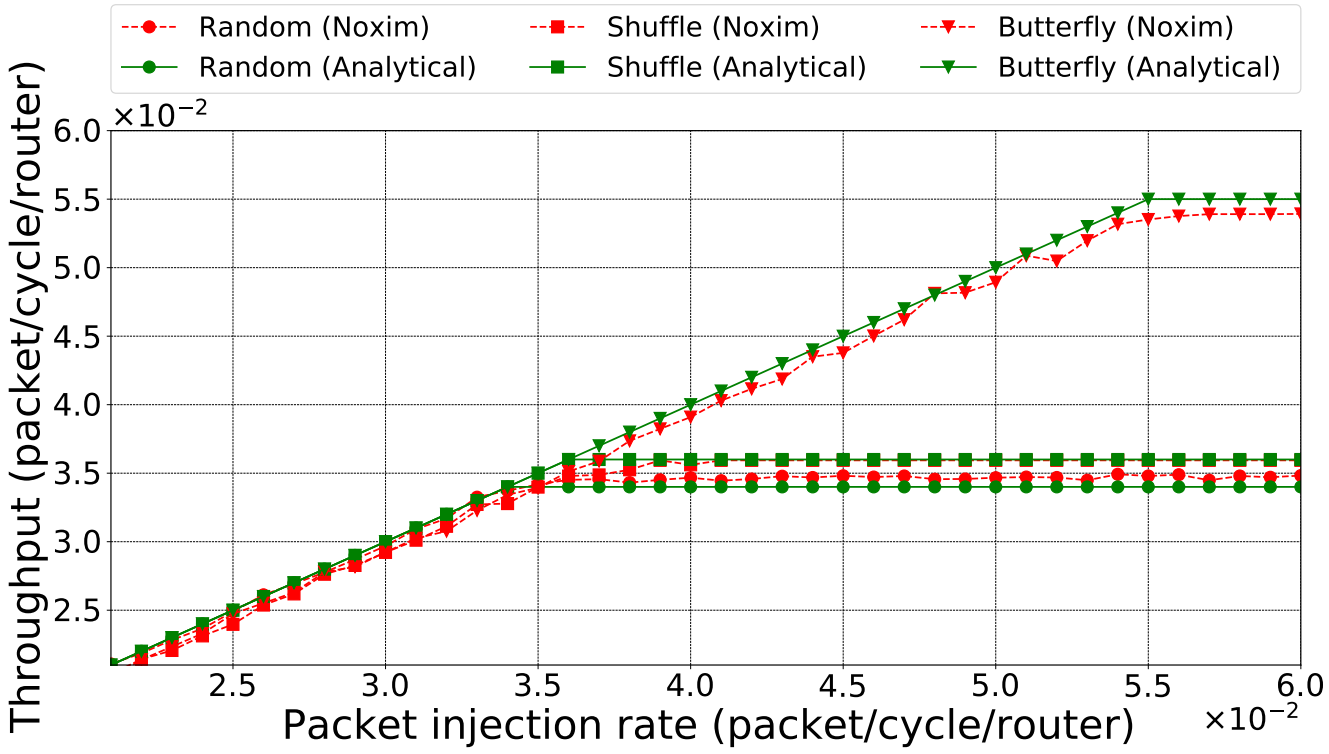


Fig. 6: Throughput Vs. PIR

D. Execution time

Fig. **??** shows the execution times of our analytical model and Noxim versus the number of cores in the architecture. It is evident that the speed up increases with respect to the architecture size. For instance, for a 16×16 architecture size, our analytical model requires only 1.22 seconds versus 603 seconds for a Noxim simulation, leading around $500\times$ speed up. This validates the benefits of using an analytical model instead of a simulator to explore a plethora of parameters and architecture topologies. Indeed, 603 seconds with Noxim is only for one simulation. For instance with a 16×16 architecture, if we want to explore the following design space:

- $N_a = [4, 8, 16]$
- $D = [8, 16, 32]$ Gb/s
- $N_f = [8, 16, 32]$ flits/packets
- Traffic = [Random, Shuffle, Butterfly]

The number of simulations required is $N_a \times D \times N_f \times$ number of traffics $= 81$ simulations. With Noxim simulations, 13 hours are required, while our analytical model only requires 1 minute and 37 seconds. Obviously, both Noxim and our analytical model can be executed with multiple instances (i.e. in parallel) for larger design spaces. For instance evaluating the impact of antenna positions in a 16×16 architecture with 4 antennas requires $C\binom{4}{256} = 174.8 \times 10^6$ simulations. With 10 instances running in parallel, Noxim would require more than 5 years of simulation time, while our analytical model only take 4 days.

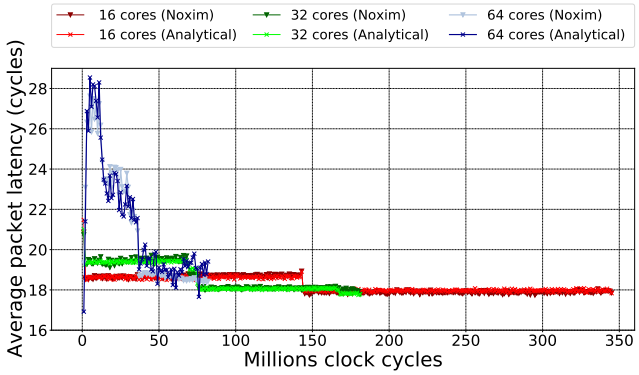


Fig. 7: Validation of the analytical model using the dedup benchmark application

E. Benchmark applications

To validate the accuracy of our model with heterogeneous traffic patterns, we adopt the PARSEC [**?**] benchmark suite, which is commonly used to study the performance of NoCs. We have chosen five representative applications, based on their parallelization level, working set, and data usage. The benchmarks are simulated using the SNIPER [**?**] simulator to extract communications traces. Communication traces list all injected packets during application execution. Each packet injected has as information the source, destination, and injection time. We simulate architectures with 16, 32 and 64 cores, along with 4 antennas (i.e cluster of 4, 8 and 16 cores respectively). The memory organization follows a distributed shared memory model, and the cache coherency protocol is MESI. Other parameters are set to default values.

From the communication traces obtained with the Sniper simulator, we performed a traffic characterization. We used a windowing approach to reproduce the individual packets injection of real traffic into the analytical model where packets are considered injected following a given distribution law. More specifically, we divided the traffic traces into windows of 1 million cycles for a matter of readability in our results (window size can be adapted). For each window, we calculated the packet injection rate (PIR) between each source and destination couple. These PIR values were then used as inputs to the analytical model, which applies queuing theory to simulate contention. Hence, packets are injected following a Poisson distribution as we use a M/G/1 queuing theory model.

Fig. **??** shows the average latency versus the execution time of the dedup application. Each plotted point represents the average latency of the hybrid NoC within the time window. It shows the evolution of latency during execution with respect to application needs. It also shows the application speed-up when additional cores are utilized, at the cost of increased communications, leading to higher latency. Indeed, the more cores used, the faster the execution becomes, thanks to increased computing parallelism. However, more communications occur in a shorter time, resulting in increased congestion. This is clearly visible with both 64 cores curves, which finishes at 80 millions clock cycles versus almost 350 millions clock cycles for a 16-core architecture. In terms of communication latency, there is a latency peak around 28 clock-cycles for 64 cores, while kept around 19 clock-cycles for 16 cores. Both results from Noxim and our model exhibit similar behaviours validating the efficiency while considering heterogeneous traffic patterns [**?**], which is particularly important for non-homogeneous system-on-chip such as accelerator-based architectures.

Finally, Table **??** summarizes the average latency percentage error of our analytical model compared to Noxim for the different applications and architecture sizes. We can notice that our error is less than 5.5%, with an average of 3.87%.

It has to be noticed that using application traces for on-chip interconnect performance analysis is a classic technique. It offers faster exploration than full system simulation. As limitation of this technique, increasing or reducing latency of communications will not modify the system level behavior, i.e. will not impact the execution time of the application as only traces are injected. On the other hand, it provides i) a quick coarse-grained analysis to explore interconnect parameters, and ii) to know whether the interconnect is congested or not regarding heterogeneous traffic, hence whether the application will be slowed down or accelerated. This analysis is useful to verify whether hybrid NoC satisfies the communication needs of real applications or suffers from congestion.

TABLE III: Average latency error between the analytical model compared to Noxim for different benchmark applications.

Applications	Average latency error (%)		
	16 cores	32 cores	64 cores
$blackscholes$	2.24	4.56	1.9
$dedup$	3.75	3.67	1.99
$raytrace$	4.02	3.17	3.93
$streamcluster$	4.56	5.5	2.4
$x264$	2.8	3.95	2.55

F. Relative Error

Analytical models are based on queuing theory which mathematically compute the packet latency. It determines the behavior of queues (here electrical and wireless routers) and predicts the time it takes for packets to be processed and transmitted while considering congestion. In analytical models, packets are considered to define the NoC traffic, and this traffic is modeled by an injection rate. On the other hand, cycle-accurate simulator will simulate the behavior of the NoC at each clock cycle while considering each packet injected into the network. Most of the error appears when the network become highly congested (see section **??**). In fact, the increasing latency due to congestion in each router will modify the arrival time in the following routers along the path, which is more complex to accurately predict in analytical models. This error is common to all analytical models, and This is the trade-off for shorter computation times compared to simulators (see Section **??**).

During the extensive experimentation conducted in Section **??**, we performed more than 1,000 simulations with the different parameters highlighted. The relative error between the analytical model and Noxim simulator was computed using the equation **??** and was found to be around 4%.

$$E = \frac{1}{N_s} \times \sum_{i=0}^{N_s} \frac{|L_A[i] - L_S[i]|}{L_S[i]} \tag{19}$$

where N_s is the number of simulations ($N_s = 1000$), $L_A[i]$ is the latency computed by our model for simulation i, and $L_S[i]$ is the latency computed by Noxim for the same simulation i.

V. Conclusion

This paper proposes an analytical model to evaluate the communication latency and the network throughput of manycore architectures using a hybrid NoC based on both ENoCs and WiNoCs. The proposed model is validated with Noxim simulation experiments. We validated the correctness with several parameter explorations on synthetic traffics. The packet latency error of our model is less than 5% compared to the cycle-accurate Noxim simulator with Parsec application traffic. Compared to Noxim, our model requires up to 500 times less time to perform the results on a 16×16 architecture, hence enabling design space exploration of such complex multi-parameter architectures. As future work, we plan to explore other emerging on-chip interconnects, such as guided-RF or photonic on silicon where the channel accesses differ. For example, for photonic on silicon, the optical link may be shared with different optical network interfaces, and the bandwidth can be adapted on demand thanks to the optical wavelength allocations [**?**].

The code is open-source and can be found at https://gitlab.inria.fr/taran/AMHNOC.git.

ACKNOWLEDGMENTS

This work benefited from the support of the projects SHNoC ANR18-CE25-0006, OpticALL2 ANR-18-CE24-0027 and RAKES ANR-18-CE25-0017 of the French National Research Agency (ANR).