Analytical Router Modeling for Networks-on-Chip Performance Analysis

Umit Y. Ogras and Radu Marculescu
Department of Electrical and Computer Engineering
Carnegie Mellon University, USA
{uogras,radum}@ece.cmu.edu

Abstract

Networks-on-Chip (NoCs) have recently emerged as a scalable alternative to classical bus and point-to-point architectures. To date, performance evaluation of NoC designs is largely based on simulation which, besides being extremely slow, provides little insight on how different design parameters affect the actual network performance. Therefore, it is practically impossible to use simulation for optimization purposes. In this paper, we first present a generalized router model and then utilize this novel model for doing NoC performance analysis. The proposed model can be used not only to obtain fast and accurate performance estimates, but also to guide the NoC design process within an optimization loop. The accuracy of our approach and its practical use is illustrated through extensive simulation results.

1. Introduction

Networks-on-Chip (NoC) communication architectures target single chip multiprocessor systems that implement multiple applications [1,5,9]. The complexity of such systems, as well as the tight requirements in terms of power, performance, cost and time-to-market, place a tremendous pressure on the design team. To cope with this situation, application and platform models are usually developed separately [12]. After that, the application is mapped to the target platform and the resulting system is evaluated to ensure its compliance with the design specifications.

The success of this methodology depends critically on the availability of adequate performance analysis tools that can guide the overall design process. In order to be used in an optimization loop (Figure 1), the analysis needs to be tractable, while providing meaningful feedback to the designer. Time consuming simulations can only come into the picture at later stages, typically *after* the design space is already reduced to only a few practical choices (the outer loop in Figure 1).

For traffic with guaranteed service [5,13], accurate performance figures can be easily derived. However, the performance analysis of best effort traffic relies largely on simulation or simple performance metrics derived under idealized conditions. For example, the average hop count is commonly used to approximate the average packet latency [14]. While this metric ignores the queuing delays and network contention, approaches that do consider queuing delays often make other idealistic assumptions such as exponential service times, infinite buffers, *etc.* [6,7,10].

In this paper, we present an analytical performance analysis methodology for NoCs based on a novel router model. The router model allows us to compute the average number of packets at each buffer in the network as a function of the traffic arrival process. This model is then used to analyze each router in the network, given the topology, routing algorithm, driver application and its mapping to the network. The proposed approach, which is developed for wormhole flow control, provides three performance metrics, namely average buffer utilization, average packet latency per flow, and network throughput. These metrics can be conveniently used for design and optimization purposes, as well as obtaining quick performance estimates.

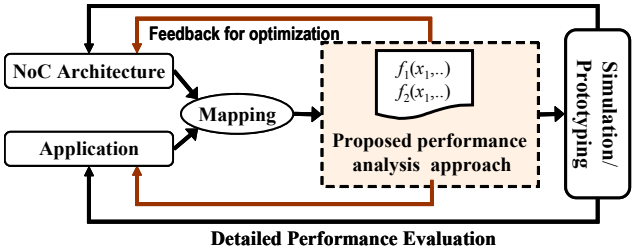


Figure 1. The use of the proposed performance analysis approach is illustrated with the Y-chart scheme [12].

The remaining of this paper is organized as follows. Section 2 reviews related work and highlights our contributions. Section 3 presents the router model, while Section 4 discusses its use in network performance analysis. Experimental results appear in Section 5. Finally, Section 6 concludes the paper by summarizing our main contribution.

2. Related Work and Novel Contributions

The design of application-specific NoCs is commonly formulated as a constrained optimization problem [9,14,15]. Therefore, performance analysis techniques that can be used in optimization loops are extremely important. The authors in [10] consider the buffer sizing problem and present a performance model based on queuing theory. However, the approach is applicable to only packet switched networks. The work in [8] presents a wormhole delay model applicable to routers with single flit buffers and assume that packet size dominates the overall latency.

Related work about analysis techniques for wormhole routing comes mainly from parallel computing and macro-network research communities. Many studies target specific network topologies such as *k*-ary *n*-cubes [3,6] and hypercubes [16]. The study presented in [7] is not restricted to a particular topology, but it assumes an exponential message length distribution and it has a very high complexity for high dimensional networks. A more general analytical model for wormhole routing is presented in [11]. The model provides average packet latency estimates using a sophisticated analysis.

978-3-9810801-2-4/DATE07 © 2007 EDAA

The main *contribution* of the work herein is a novel performance model for on-chip routers which generalizes the traditional delay models for single queues and captures the classical results as a special case. This model is used to develop a thorough performance analysis for wormhole routing with *arbitrary size messages and finite buffers* under application-specific traffic patterns. Furthermore, the model supports arbitrary network topologies and deterministic routing. Finally, the proposed model provides not only aggregate performance metrics, such as average latency and throughput, but also useful feedback about the network behavior at a fine-level granularity (*e.g.,* utilization of all buffers, average latency per router and per flow). Hence, it can be invoked in any optimization loop for NoCs (*e.g.,* application mapping [9,14], network architecture synthesis [15], buffer space allocation [10]) for fast and accurate performance estimations.

3. Router Modeling for Performance Analysis

3.1. Basic assumptions and notations

We consider input buffered routers with P channels and target wormhole flow control under deterministic routing algorithms. The size of the packets (*bits*) is denoted by the random variable $\boldsymbol{S}$, as listed in Table 1 along with other parameters. The probability distribution of $\boldsymbol{S}$ is determined by the driver application.

The network channel bandwidth is denoted by W (in *bits*/*sec*). The router service time for the header flit is given by H_s. We note that H_s is a function of the router design and includes the time to traverse the router (t_R) and the link (t_L). Since the remaining flits follow the header flit in a pipelined fashion, the service time of a packet, excluding the queuing delay (this will be accounted for in Section 4), is given by:

$$\boldsymbol{T} = H_s + \left\lceil \frac{\boldsymbol{S}}{W} \right\rceil \tag{1}$$

We denote by x_{sd} (*packets/sec*) the rate of the traffic transmitted from the source node at router s to the destination node at router d. Likewise, the traffic arrival rate of the header flits to input channel j of router i is given by λ_{ij} (*packets/sec*). We assume that the arrival process of the *header flits* to the router inputs (λ_{ij}) follows a Poisson process. Note that under this model, the arrival process for the body flits is *not* assumed to be Poisson; the Poisson assumption refers only to the header flit distribution. This matches the reality since the arrival of the header flit implies that all body flits will follow in sequence.

This assumption, which is quite common [6,8,11], enables us to derive closed loop solutions and show that our model generalizes the classical results for single queue systems. However, in general, the real arrival process may exhibit a more deterministic or long-range dependent behavior [18] depending on the type of traffic. Nevertheless, our model provides insight into router design and reasonably accurate results for pruning the design space at early design stages, as shown in Section 5. Removing the Poisson assumption completely is left for future work.

3.2. Analytical model of the router

This section focuses on modeling a single router as a set of first-come first-serve buffers connected by a crossbar switch. The parameter of interest is the *average number of packets at the input buffers*, at *each input channel* $1,...,P$, *i.e.,* $N = [N_1, N_2, ... N_P]^T$.

Table 1: List of input parameters used in the paper. Bold symbols (e.g., *S* and *T*) denote random variables.

Input	*Explanation*	*Depends on*
$\boldsymbol{S}$	Random variable (rv) denoting the packet size.	Application
x_{sd}	Packet transmission rate from node s to node d.	
R	Residual packet waiting time.	
H_s	Service time for the header flit.	Router
W	Network channel bandwidth.	
B_{ij}	Size of the input buffer at router i, channel j.	
$\boldsymbol{T}$, T, $\overline{T^2}$	rv $\boldsymbol{T}$ denotes the packet service time. T and $\overline{T^2}$ are its 1st and 2nd order moments.	Application & Router
c_{ij} , C	Contention probability between channel i and j.	
λ_{ij}	Traffic arrival rate at router i, channel j.	Topology, routing, application

Since Poisson arrivals see time averages, the following equilibrium equation is valid for the input buffer at channel j:

$$\lambda_j = \frac{N_j}{\tau_j} \tag{2}$$

where τ_j denotes the *average time* an incoming packet spends in queue j. τ_j is composed of the following three components:

I. Service time of the packets already waiting in the same buffer;

II. The packets waiting in the other buffers of the same router and served before the incoming packet;

III. The residual service time seen by an incoming packet (R).

Therefore, τ_j can be written as:

$$\tau_j = \underbrace{TN_j}_{I} + \underbrace{T \sum_{k=1,\,k\neq j}^{P} c_{jk}N_k}_{II} + \underbrace{R}_{III} \tag{3}$$

where the coefficients c_{jk} denote the contention probabilities, *i.e.,* the probability that channels j and k compete for the same output. The second component of the average waiting time (*i.e. II* in Equation 3) applies only to those packets that will be served before the incoming packet. Depending on the output channel requested by the incoming packet and the router scheduling policy (*e.g.* priority, round robin, *etc.*), an incoming packet can be served earlier than a packet that is already waiting in one of the other buffers. In the following, we assume the round robin policy, but the results can be easily extended for other scheduling disciplines.

Let C_j be the row vector $C_j = [c_{j1}, c_{j2},...,c_{jP}]$ of the contention probabilities, where $c_{jj} = 1$. Then, Equation 2 can be written using τ_j from Equation 3 as:

$$\lambda_j = \frac{N_j}{TC_jN + R}, \text{ since } C_jN = N_j + \sum_{k=1,\,k\neq j}^{P} c_{jk}N_k$$

so rearranging the terms yields:

$$\lambda_j TC_jN + \lambda_j R = N_j \tag{4}$$

Equation 4 describes the *equilibrium condition* of the buffer at the input channel j only. For the entire router, we denote the arrival rates (Λ), the contention matrix (C) and the residual time ($\overline{R}$) by:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_P \end{bmatrix}_{P\times P}, \quad C = \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_P \end{bmatrix}_{P\times P}, \quad \overline{R} = R \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix}_{P\times 1}$$

Then, the equilibrium condition for the router can be written as:

$$T\Lambda CN + \Lambda R = N$$
$$(I - T\Lambda C)N = \Lambda\overline{R}$$

Finally,

$$N = (I - T\Lambda C)^{-1}\Lambda\overline{R} \quad (5)$$

The router model described by Equation 5 provides a closed form expression for the average number of packets at *each* buffer of the router, given the traffic arrival rates (Λ), the packet contention probabilities (C), router design specifications (H_s, W) and target application ($\boldsymbol{S}$). Equation 5 generalizes the single queue model; this is one of the major contributions of this paper.

We note that when $det(I - T\Lambda C) = 0$, the packet population in the router grows to infinity. This corresponds to the case when the utilization is 1 for a system with a single queue. The following example gives more intuition for Equation 5.

Example: Consider the case $P = 1$ (*i.e.,* single queue system) and infinite buffers. In this case, Equation 5 simply becomes:

$$N = \frac{\lambda R}{1 - T\lambda}$$

Furthermore, the residual waiting time $R = 1/2\lambda\overline{T^2}$ where $\overline{T^2}$ is the second moment of the service time [2]. As a result:

$$N = \frac{\lambda^2\overline{T^2}}{2(1 - T\lambda)}, \quad (6)$$

which is precisely the average number of packets in an *M/G/*1 system. Hence, the commonly studied distributions *M/G/*1, *M/M/*1 and *M/D/*1 become *special cases* of our newly proposed model.

3.3. Computation of the contention matrix

Let f_{ij} be the probability that a packet arrives at channel i and leaves the router through channel j. The *forwarding probability matrix* is:

$$F = \begin{bmatrix} 0 & f_{12} & f_{13} & \dots & f_{1P} \\ f_{21} & 0 & f_{23} & \dots & f_{2P} \\ \dots & \dots & \dots & \dots & \dots \\ f_{P1} & f_{P2} & f_{P3} & \dots & 0 \end{bmatrix}, \text{ where } f_{ij} = \frac{\lambda_{ij}}{\sum_{k=1}^{P}\lambda_{ik}}, 0 \le i, j \le P \quad (7)$$

Assuming that the forwarding probabilities are independent for a deterministic routing algorithm, the contention probabilities can be written in terms of the forwarding probabilities as:

$$0 \le i, j \le P \ , i \ne j, \quad c_{ij} = \sum_{k=1}^{P} f_{ik} f_{jk} \quad (8)$$

3.4. Using Equation 5 for router design

The router model described by Equation 5 provides an analytical approach to analyze the effect of various router parameters on network performance. Consider a multimedia system design [9] where the packets in the network carry data of 8×8 blocks. Each pixel value is represented by 16 bits, so $\boldsymbol{S} = 1024$ *bits*. We assume that the channel bandwidth is given by $W = 256\times f_{ch}$, where f_{ch} is the clock frequency of the router.

Two major concerns in router design are the number of pipeline stages, *i.e.,* the number of cycles it takes to route the header flit (H_s), and the size of the input buffers (B). To analyze the impact of these parameters on router utilization, we first map the system to a 4×4 mesh network running under XY routing, and determine arrival rates (Λ) and the contention matrix C for the bottleneck router. Then, we use Equation 5 to analyze the impact of H_s and B on buffer utilization.

Figure 2 shows the average number of flits in the router (at all buffers) as a function of H_s and B. For a given buffer size, the average number of flits in the router increases with increasing service time, as expected. This increase is more severe for larger buffers, since more flits are stored in the buffer before being blocked. Likewise, for a given service time, the router utilization saturates, as the buffer size increases. The saturation occurs earlier for lower service times, as depicted in Figure 2. For example, when H_s =2, increasing the buffer size beyond B_j =2 for $1 \le j \le 5$ does not increase the buffer utilization (see point **"A"** in Figure 2), since the router is very fast. On the other hand, for larger service times (*e.g.,* H_s =8, **"B"** in Figure 2), the saturation point moves further away, *i.e.,* more flits wait in the buffer before being served.

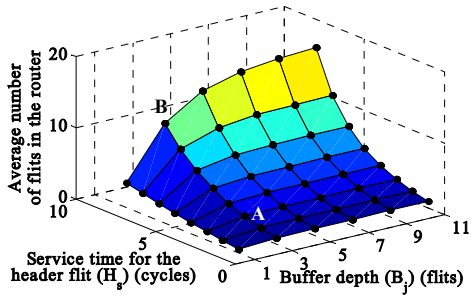


Figure 2. The average number of flits in the router (*i.e.,* the sum of the flits in all five buffers) is shown as a function of the buffer size and service time of the router.

This case study illustrates the use of the proposed router model as a powerful tool for routers design. Indeed, this model can be used by designers to evaluate possible trade-offs offered by different design choices (*e.g.,* buffer size, channel width) that are nowadays pre-determined mostly in an ad-hoc manner.

4. Network Performance Analysis

The router model presented in Section 3 enables calculation of the average utilization of the input buffers given the traffic input to the router. In this section, we discuss how this model can be actually utilized to analyze the performance of the entire network. More specifically, we compute the average buffer utilization in the router, average packet latency and the maximum network throughput using the proposed model.

We start by computing the traffic arrival rate (λ_{ij}) at channel j of any router i. λ_{ij} is given by:

$$\lambda_{ij} = \sum_{\forall s}\sum_{\forall d} x_{sd}\Re(s, d, i, j) \quad (9)$$

where $\Re$ is the *indicator function* such that $\Re(s, d, i, j) = 1$ if the packet sent from the source PE s to the destination PE d is routed through the input channel j of router i, and $\Re(s, d, i, j) = 0$ otherwise.

We note that the routers are typically interconnected in a network. Hence, the service time for the header flits may increase due to chained blocking at the downstream routers. In general, the blocking probabilities, hence the expected waiting time of the header flit due to blocking, can be computed using an iterative process similar to [6] or through computation ordering [11]. In our experimental work, the blocking probability for the buffer at input channel j, $p_b(\lambda_j, \boldsymbol{T}, B_j)$, is found using an $M/G/1/m$ queuing model [17] for a single iteration. After that, the forwarding probabilities for each router (F in Equation 7) and the contention matrix (C in Equation 8) are computed. Since our goal is to demonstrate the router model developed in Section 3, we omit here the details of these calculations.

4.1. Average buffer utilization and packet latency

Given the arrival rates, λ_{ij}, at each input channel in the network, the contention matrix for each router, and $\boldsymbol{T}$, we use Equation 5 to find the average number of packets in the input buffer at each router. This information can be used for optimization purposes (*e.g.* to determine the buffer sizing), since buffer utilization provides information about the distribution of the traffic load across the network. The average buffer utilization can also be used to compute the average waiting time in buffers. By Little's theorem:

$$W_{ij} = N_{ij}/\lambda_{ij} \tag{10}$$

where W_{ij} is the average waiting time in the channel j buffer at router i. Since we already know the packet service time**,** W_{ij} enables us to compute the average packet latency at each router.

The delay experienced at each router is a performance metric with very fine granularity. Indeed, it can be used to compute the average latency for each traffic source/destination pair separately, as well as the average packet latency in the network. When a packet is sent from the source node s to the destination node d, it traverses a set of routers and the corresponding input buffers denoted by Π_{sd}. The average latency for any packet from node s to node d (denoted by L_{sd}) is given by:

$$L_{sd} = W_s + \sum_{(i,j) \in \Pi_{sd}} (W_{ij} + T)$$

where W_s is the queueing delay at the source, W_{ij} is the queuing delay at channel j of router i, and T is the average service time. W_s is computed using the $M/G/1/m$ model, since the buffers in the PEs are also finite. Then, the *overall average packet latency* in the network is found as:

$$L = \frac{1}{\sum_{\forall s,d} x_{sd}} \sum_{\forall s,d} x_{sd} \times L_{sd} \tag{11}$$

This relation provides fast and accurate estimates of L for a variety of traffic patterns and application mappings, as in Section 5. It can be applied to a wide range of optimization problems, since average packet latency is a common performance metric.

4.2. Network throughput

The network throughput is defined as the rate at which the packets are delivered to the destination. At low traffic loads, the packet delivery rate is equal to the packet injection rate. However, as the traffic load increases, the throughput starts saturating. To find the saturation value of the throughput, we start with source traffic generation rates αx_{sd}, where α is a positive scaling factor which ensures that the network is not saturated. Then, the arrival rates to the router inputs, $\alpha\lambda_{ij}$, are computed, and the equilibrium equation for *each* router is written as:

$$N(\alpha) = (I - \alpha T \Lambda C)^{-1} \alpha \Lambda R \tag{12}$$

where $N(\alpha)$ is the average number of packets in the router as a function of α. When the utilization of the input buffers approaches unity, the router will be always busy so its throughput will saturate. The approximate value of α that will saturate a given router can be found by solving the following equation:

$$\sum_{j=1}^{P} N_j(\alpha) = 1 \tag{13}$$

We solve equations 12 and 13 to find the minimum value of α over all the routers, *i.e.* α_{min}. Then, the traffic generation rates at which the application throughput saturates is found as $\alpha_{min} x_{sd}$. Finally, the *saturation throughput* of the network is found as:

$$\Gamma = \alpha_{min} \sum_{\forall s,d} x_{sd} \tag{14}$$

In summary, the basic idea of our approach is to identify the bottleneck router, which happens to be the router with the highest amount of traffic through it. The critical load of this router defines the critical load of the overall network, since the congestion propagates quickly across the entire network. While techniques to compute average packet latency have been proposed before, to the best of our knowledge, the presented model provide the *first analytical approach* for finding the maximum network throughput for arbitrary traffic patterns.

4.3. Overview of the analysis methodology

The proposed analysis technique is summarized for convenience in Figure 3. First, the traffic input rates to the routers and packet service time (including the waiting time due to blocking) are computed using equations 1 and 9. In order to find the average packet latency, we follow the path on the left in Figure 3. The average utilization of the input buffers in the routers are found using Equation 5 (Step 2a). Next, the average packet latency in the network is found using Equation 11 (Step 3a). Finally, to find the saturation throughput, we identify the bottleneck router and use Equation 14 (Steps 2b and 3b in Figure 3).

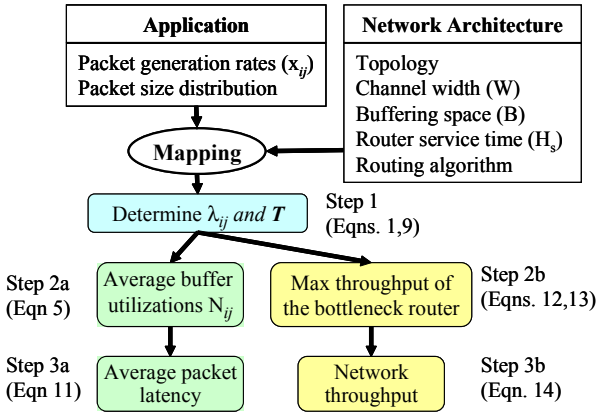


Figure 3. Overview of the proposed approach.

5. Experimental Results

This section provides a detailed study on the accuracy and run-time of the proposed approach. The analytical results obtained using the proposed method are compared against those obtained with a cycle-accurate (flit-level) NoC simulator. Both the simulator and the analytical model are implemented in C++ and tested on a Pentium 4 computer with 512M memory running Linux OS.

Throughout the experiments, the size of the input buffers in the routers is 5×64 bits, *i.e.,* each buffer can hold five 64-bit flits. In the absence of contention, the router service time is 4 cycles. Simulations run for 50000 cycles with an initial warm-up period of 20000 cycles. Also, multiple simulations are performed with different seeds in order to collect relevant averages.

5.1. Average packet latency

We first consider a multimedia application [9] which is manually mapped to a 4×4 2D mesh network. We compare the average packet latency obtained using the proposed approach against the values obtained by simulation. The average packet latency as a function of the packet injection rate is shown in Figure 4.

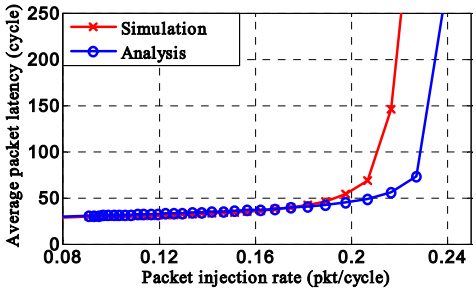


Figure 4. Average packet latency values found with the proposed approach and by simulation are shown.

We observe that the latency values estimated by the proposed approach follow the simulation results closely. More precisely, for packet injection rates below 0.2 *pkt/cycle*, the relative error between the analytical and simulation results is within 5%. After that, the latency values start increasing abruptly, since at this critical traffic load the network enters the congestion region. Our approach is also capable of estimating this critical value, as we demonstrate in Section 5.3.

Next, we assess the accuracy of our approach for different application mappings. We performed experiments for 1000 random mappings. For each mapping, the average packet latency is computed using the proposed approach and by simulation, at 0.16 *pkt/cycle* injection rate, which is a possible operating point (see Figure 4). We repeat each simulation 50 times with different seeds; the results are averaged such that the measured latency is within one standard deviation of the actual value with 95% confidence. More formally, let $L_S(i)$ be the average packet latency for mapping i obtained by simulation and $L_A(i)$ be the corresponding latency obtained using Equation 11. The relative error between the analytical and simulation results, for 1000 different mappings, is:

$$Err = \frac{1}{1000}\sum_{i=1}^{1000}\frac{|L_S(i)-L_A(i)|}{L_S(i)}$$

Using this definition, the relative error between the analytical and simulation results is about 9%. This is actually a very good accuracy level, given that the relative error is very sensitive even to small differences in data values.

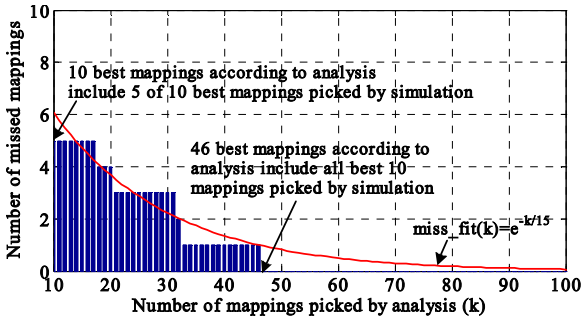


Figure 5. The top mappings selected by simulation, but missed by analysis are shown.

5.2. Case study 2: Application mapping

In general, the NoC design space is too big to explore by simulation. For instance, there are *n!* different ways to map a given application to a network with *n* nodes. Since the proposed performance model targets NoC design and optimization, we illustrate the effectiveness of our approach using application mapping, which is a common optimization problem for NoCs [9,14]. More precisely, based on the average packet latency, we first rank order 1000 different mappings obtained in Section 5.1. It takes about *22 hours* to find the best mapping through simulation, whereas our approach completes the analysis of all possible solutions in about *7 seconds*, which is about *4 orders of magnitude faster!*

According to the simulation results, the best among all 1000 mappings is the mapping with ID 268 with an average latency of 35.5 cycles. According to the analysis we propose, the best mapping is the one with ID 732 which has average latency of 35.3 *cycles*. The latency for mapping ID 732 found by simulation is 36.2 *cycles*. As such, the analysis approach selects a mapping whose latency is within 2% of the best one found by simulation. Additionally, the analysis discovers the best mapping about 10000 times faster and so much more mappings can be explored, within the same time budget, using the proposed analytical technique.

To evaluate the analysis approach from a different angle, assume now that the objective is to select the 10 best mappings for more detailed evaluations. Therefore, we denote the top 10 mappings obtained via simulation as being the golden set. Then, we find the top k mappings based on the analysis results, where $1 \le k \le 100$. When we pick strictly the top 10 mappings based on analysis, only 5 mappings selected by simulation are missed. However, the number of misses drops exponentially to zero as k increases. For instance, the top 20 mappings picked by our approach include 7 best mappings found by simulation, while top 46 mappings contain all 10 best mappings, as shown in Figure 5.

To sum up, the proposed method can be used to prune the large design space accurately in a very short time compared to simulation. Experiments performed on larger networks show *several orders of magnitude achievable* speed-up compared to a single simulation run. Considering that many simulations are needed to obtain high confidence intervals, the overall speed-up due to the analytical approach is impressive. Moreover, the simulation run-time grows faster for heavier traffic, while the run-time of the analytical approach remains pretty much the same.

5.3. Network throughput

Next, we compare the maximum network throughput obtained via simulation against the analysis results found using Equation 14. In order to test the robustness of our approach to *non-uniform* traffic conditions, each node communicates only with the nodes that are located within a forwarding radius. Furthermore, if the distance between the source and destination nodes is given by $dist(s,d)$, then the forwarding probability $p_f(s,d)$ is:

$$p_f(s,d) = \begin{cases} \sim 1/dist(s,d) & dist(s,d) \le F_R \\ 0 & dist(s,d) > F_R \end{cases} \quad (15)$$

where F_R (number of hops) is the radius of the forwarding region. The maximum network throughput of a 8×8 mesh network, as a function of the traffic locality is given in Figure 6. As expected, the network throughput increases with the level of the locality. Furthermore, our technique provides a close approximation to the simulation results over a wide range of characteristics in the traffic locality.

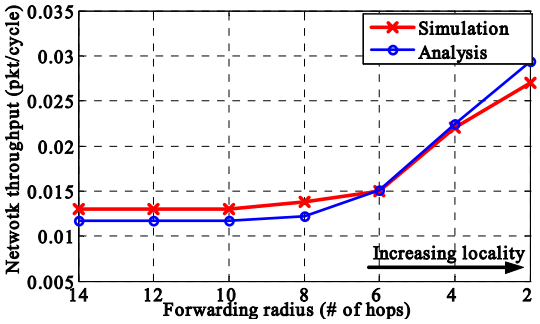
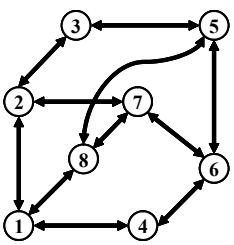


Figure 6. Sustainable network throughput for 8×8 2D mesh network with local traffic described by Equation 15.

5.4. Application to arbitrary topologies

Since the proposed performance analysis is general, we apply it now on arbitrary topologies [15]. To this end, we analyze and simulate the simple network in Figure 7(a). Figure 7(b) describes the traffic pattern and the deadlock-free routing algorithm used in the network. The entries of routing matrix, RM(i,j) $1 \le i,j \le 8$, show whether there is communication between nodes i and j, and the routing choice in case they communicate. For instance, in Figure 7(b), RM(1,5) = – implies that node 1 does not send packets to node 5. On the other hand, RM(1,6) = 4 means that node 1 forwards the packets to node 4, when it needs to communicate with node 6. Finally, the traffic load between all pairs of communicating nodes is uniform and the packets consist of 15 flits.



(a)

RM =

Id	1	2	3	4	5	6	7	8
1	–	2	2	4	–	4	8	8
2	1	–	3	–	–	–	7	7
3	2	2	–	–	5	–	–	–
4	1	1	1	–	–	6	–	–
5	3	3	3	–	–	6	–	–
6	4	4	5	4	5	–	7	–
7	2	2	2	–	–	–	–	8
8	1	5	–	–	–	–	7	–

(b)

Figure 7. (a) Arbitrary network topology used to test the proposed technique and (b) the Routing Matrix (RM).

The maximum throughput of this network is found as 0.2 *packets/cycle* using our technique. To evaluate the accuracy of this value, we also run 50 simulations with different random seeds and identify the maximum throughput as 0.18 *packets/cycle*. As such, the difference between simulation and analysis is about 11%. More detailed evaluations (such as the one shown in Figure 4) are omitted here due to lack of space.

6. Conclusions

In this paper, we presented a novel router model for NoC performance analysis. Our approach provides not only aggregate performance metrics such as average latency and throughput, but also feedback about the network characteristics (*e.g.,* buffer utilization, average latency per router and per flow) at a fine-level of granularity. Furthermore, our approach makes the impact of different design parameters on the performance explicit so it provides invaluable insight into NoC design. As a result, the proposed approach can be used as a powerful design and optimization tool. Experimental results demonstrate the accuracy and efficiency of the analysis on real and synthetic benchmarks.

Acknowledgements: This work is supported by GSRC Marco.

7. References

[1] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, 35(1), Jan. 2002.

[2] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1992.

[3] W. J. Dally. "Performance analysis of *k*-ary *n*-cube interconnection networks," *IEEE Trans. on Computers*, 39(6), June, 1990.

[4] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.

[5] J. Dielissen, *et. al.*, "Concepts and implementation of the Philips network-on-chip," *in Proc.IP-based SoC Design,* Nov. 2003.

[6] J. Draper and J. Ghosh, "A comprehensive analytical model for wormhole routing in multicomputer systems," *Journal of Parallel and Distributed Computing*, 23(2), Nov. 1994.

[7] W. Guan, W. Tsai, and D. Blough, "An analytical model for wormhole routing in multicomputer interconnection networks," *in Proc. Intl. Parallel Processing Symposium*, Apr., 1993.

[8] Z. Guz, *et. al.*, "Efficient link capacity and QoS design for wormhole network-on-chip," *in Proc. DATE*, March 2006.

[9] J. Hu and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 24(4), Apr. 2005.

[10] J. Hu, *et. al.*, "System-level buffer allocation for application-specific networks-on-chip router design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 25(12), Dec. 2006.

[11] P. Hu and L. Kleinrock, "An analytical model for wormhole routing with finite size input buffers," *15th Intl. Teletraffic Congress*, June 1997.

[12] P. Lieverse, *et. al.*, "A methodology for architecture exploration of heterogeneous signal processing systems," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 29(3), Nov. 2001.

[13] M. Millberg, E. Nilsson, R. Thid and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," *in Proc. DATE*, Feb. 2004.

[14] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," *in Proc. DATE,* March 2004.

[15] U. Y. Ogras and R. Marculescu, "'It's a small world after all': NoC performance optimization via long-range link insertion," *IEEE Trans. on VLSI*, 14(7), 2006.

[16] M. Ould-Khaoua and H. Sarbazi-Azad, "An analytical model of adaptive wormhole routing in hypercubes in the presence of hot spot traffic," *IEEE Trans on Parallel and Distributed Systems,* 12(3), March, 2001.

[17] H. Takagi, *Queueing Analysis, Vol. 2: Finite Systems*. Elsevier, 1993.

[18] G. Varatkar and R. Marculescu, "On-Chip traffic modeling and synthesis for MPEG-2 video applications," *IEEE Trans. on VLSI,* 12(1), 2004.