A Flexible Analytic Model for the Design Space Exploration of Many-Core Network-on-Chips Based on Queueing Theory

Erik Fischer, Albrecht Fehske, and Gerhard P. Fettweis
Vodafone Chair Mobile Communications Systems
Technische Universität Dresden
Dresden, Germany
Email: {erik.fischer, albrecht.fehske, fettweis}@ifn.et.tu-dresden.de

***Abstract*—A continuing technology scaling and the increasing requirements of modern embedded applications are most likely forcing a current multi-processor system-on-chip design to scale to a many-core system-on-chip with thousands of cores on a single chip. Network-on-chip emerged as flexible and high-performance solution for the interconnection problem. There will be an urgent need for fast, flexible and accurate simulation models to guide the design process of many-core system-on-chip. In this paper, we introduce a novel analytic approach for modeling on-chip networks to fulfill these requirements. The model is based on queueing theory and very flexible in terms of supported topology, routing scheme and traffic pattern. The approach overcomes the limitations of the mean value analysis introduced in the existing work. Instead, it provides information about a steady-state distribution of the network routers. This allows to dimension network resources, such as buffers, links, etc. We show the high accuracy of the model by comparison with a cycle-accurate simulation. The model is able to estimate the mean network latency with an accuracy of about 3%.**

Keywords-network-on-chip; noc; queueing theory; analytic model.

I. Introduction

In embedded computing, today's applications show a common trend towards a continuously increasing computational effort and reliability. This is especially true in the area of multi-media and mobile communication. These requirements can only be fulfilled by massively exploiting parallelism. Taking also emerging technologies like 3D chip stacking [1] into account, today's multi-processor system-on-chips (MPSoCs) soon scale to many-core SoCs with thousands of processors on a single chip [2]. Already in 2015, we may have 1000 or more cores on a chip [3].

If we assume such a large number of cores, the interconnection problem becomes a serious challenge. Classical interconnection architectures, such as busses or crossbar switches, cannot offer the necessary flexibility and scaling with respect to throughput or area overhead. Network-on-chip (NoC) evolved as a flexible and high-performance solution for the interconnection problem during the last decade [4]. NoC is a packet-switched on-chip network where packets are forwarded from a source to a destination via several intermediate router nodes. We call the processing nodes that are connected to the NoC cores, modules or processing elements (PEs). Their functionality is thereby transparent to the NoC, i.e., this could be a processors, memory or an external interfaces. The smallest unit, to be transmitted over a NoC, is called the *flit* (flow control digit).

Finding an optimal NoC interconnect for many-core SoCs is a very challenging task, since many different design objectives and constraints have to be considered, like choosing routing and switching methods, selecting topology, application mapping, etc. [5]. This leads to a huge design space. Therefore, fast and accurate NoC models will be required that give an insight into the system and enable us to reduce the design space already in early design stages. Cycle-accurate simulation based approaches are too slow for this purpose. Simple high-level system models (e.g. only considering the propagation latency and ignoring queueing delays), on the other hand, are able to provide results in very short time. Due to the high abstraction, however, these models loose quite some accuracy. Analytic models provide a good trade-off between both approaches and are thus well suited for the NoC exploration of a many-core SoC.

In this paper, we propose an analytic NoC model based on queueing theory [6] that provides a high degree of flexibility regarding topology, routing and traffic scheme. In contrast to existing models, it is not restricted to mean value analysis but provides information about the state distribution functions of the routers. It enables us to easily derive arbitrary performance metrics, such as mean latency, buffer usage or blocking probability, and makes the model a very flexible tool for NoC performance analysis.

The remainder of this paper is structured as follows. In Section II, we discuss related work. Section III shows the system model and its assumptions. Then, Section IV introduces the analytic NoC model on network level (IV-A) and router level (IV-B). We evaluate the accuracy of the proposed approach against cycle-accurate simulation in Section V. Finally, Section VI concludes the work.

II. Related Work

Much effort has been spent for more than two decades for finding adequate traffic models for the analysis of off-chip and (later) on-chip networks. In 1990, Dally [7] developed analytic tools for investigating latency and throughput in networks, but restricting to k-ary n-cube topologies. Recent approaches focus on the mean value analysis of latency, throughput and energy consumption. Kiasari et al. presented an M/G/1 queueing model for wormhole switched two-dimensional (2D) torus NoC topologies, assuming deterministic routing [8]. A different approach has been published in 2009 in [9], which introduces an empirical model to estimate contention delays for constant service time routers. Thereby, the hybrid router model takes into account Poisson input flows as well as output flows from preceding constant service time routers. Ogras et al. presented a fast and flexible analytic approach in 2010 [10] for the mean value performance analysis of virtual channel first-come first-serve (FCFS) input buffered routers for arbitrary topology and service time

 ISBN: 978-1-61208-234-9

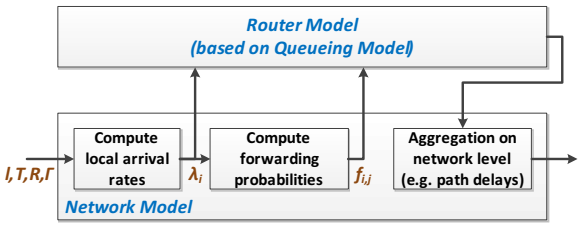

Figure 1. The Hierarchical structure of the proposed analytic model.

distribution. Other recent approaches for modeling on-chip networks [11] focus on the theory of the Network Calculus [12]. This theory provides a powerful tool for an estimation of performance bounds in NoCs, which is essential for giving statements about the realtime capabilities of a network in early design stages. However, for the exploration of the average network behavior, other methods, like stochastic models, are more expedient.

III. System Model

We assume the routers to be arranged in an arbitrary topology. An arbitrary number of cores is allowed tobe connected to a single router. Due to the low buffer requirements, wormhole switching is the most favored switching technique for realizing best-effort services in on-chip networks today [5]. Therefore, we restrict our model to this technique. The routing protocol, on the other hand, shall not be restricted. Concerning the arbitration scheme, we restrict to the first-come first-serve method. Extensions to other arbitration schemes, like the popular round-robin, are left for future work. Routers consist of an arbitrary number of buffered input ports and an arbitrary number of (unbuffered) output ports. We assume infinite buffer size.

Furthermore, we assume external packet arrivals from PEs to possess Poisson characteristic [6], i.e., they have exponentially distributed inter-arrival times with known mean values. This assumption is often made to approximate real network traffic while reducing the model complexity at the same time. The router service times include processing delay for arbitration as well as forwarding delay for the packet and are assumed to be exponentially distributed. Furthermore, knowledge of the mean router service rate and router service latency is required. We assume it w.l.o.g. to be equal for all routers in the network. Finally, we imply a common clock for all routers.

IV. An analytic Model for Networks-on-Chip

To provide a flexible as well as a fast analytic model we propose to follow a hierarchical approach as depicted in Figure 1. We split the NoC model into an analysis on network level and on router level. By performing the analysis on router level and combining the results on network level, we thus reduce the complexity.

The network model receives multiple inputs that have to be specified by the user. The traffic scenario is described by the *traffic characterization matrix* $\boldsymbol{T}$ and the *external arrival rate vector* $\boldsymbol{l}$. The topology and interconnection is specified via the *connectivity matrix* $\boldsymbol{\Gamma}$. Finally, information about the applied routing scheme is provided via the *routing matrix* $\boldsymbol{R}$. An overview of the notation and a more detailed explanation is given in Table I.

Based on this information, the network model is able to compute local parameters for each router node individually, i.e., the inputs for the router model. The local parameters comprise the local arrival rates λ_i that is the accumulated arrival rate over all traffic flows that cross router input i. Furthermore, the forwarding probabilities $f_{i,j}$ are computed. $f_{i,j}$ defines the probability that a packet arriving at a router input i is forwarded to a router output j (please note that the indices i and j correspond to the unique identifier of the link that is connected to router input or output). The computation of the local arrival rates and forwarding probabilities is discussed in more detail in Section IV-A.

The local parameters can now be applied to a queueing model on router level. It is responsible for deriving the compound distribution for the number of packets in the input queues, which represent the router state. Consequently, the knowledge of the compound distribution enables a computation of key performance indicators, such as average buffer usage, blocking probabilities or mean queueing delays. The queueing model on router level is introduced in Section IV-B.

Finally, the performance metrics, computed on router level, have to be combined on network level, e.g., to derive path delays by summing up the queueing delays and the fix router propagation latencies.

A. Analysis on Network Level

We can derive the vector of local arrival rates $\boldsymbol{\lambda}$, with elements λ_i ($1 \leq i \leq N_E$), by summing up all traffic flows that cross a specific link (and router input queue, respectively). Therein, N_E is the number of links in the network. The traffic characterization matrix $\boldsymbol{T}$ provides information about a pairwise traffic flow probability between each module s and d. By weighting $\boldsymbol{T}$ with the external arrival rates $\boldsymbol{l}$, we get the traffic intensities (in packets/cycle) for each pair of modules. Finally, we multiply the traffic intensities with the probability that the flow will pass link i (given by routing matrix $\boldsymbol{R}$) and sum up the fractions of the contributing traffic flows:

$$\lambda_i = \sum_{s=1}^{N_M} \sum_{d=1}^{N_M} l_s \cdot t_{s,d} \cdot r_{s,d,i}, 1 \leq i \leq N_E. \quad (1)$$

The notation is given in Table I. By applying the definition of the Frobenius inner product [13], we can rewrite (1) as matrix equation as follows:

Table I: Model parameters and notation

N_M	Number of modules
N_R	Number of router nodes
N_E	Number of edges
$\boldsymbol{T} = [t_{s,d}]$	Traffic characterization matrix (of size $N_M \times N_M$) with elements $t_{s,d}$ that specify the send probability from module s to module d
$\boldsymbol{l} = [l_s]$	External arrival rate vector (of size $N_M \times 1$) with elements l_s representing the arrival rate (packets/cycle) from source module s
$\boldsymbol{\Gamma} = [\gamma_{s,d}]$	Connectivity matrix (of size $(N_M+N_R)\times(N_M+N_R)$) with elements $\gamma_{s,d}$; $\gamma_{s,d} > 0$, if there is a directed connection from s to d; the value $\gamma_{s,d}$ represents the link ID for this connection (sgn($\boldsymbol{\Gamma}$) ≡topology matrix)
$\boldsymbol{R} = [r_{s,d,i}]$	Routing matrix (of size $N_M \times N_M \times N_E$) with elements $r_{s,d,i}$ defines the probability that link i is occupied for routing a packet from source module s to target module t ($\sum_{\forall i} r_{s,d,i} = 1$)
$\overline{x}$	Average router service time

 ISBN: 978-1-61208-234-9

$$\lambda_i = \mathrm{tr}\left(\left(\boldsymbol{T}\cdot\boldsymbol{L}^{\boldsymbol{D}}\right)^T \boldsymbol{R_i}\right). \quad (2)$$

In (2), tr represents the trace of the matrix, $\boldsymbol{L}^{\boldsymbol{D}}$ is the $N_M \times N_M$ diagonal matrix representation of vector $\boldsymbol{l}$:

$$\boldsymbol{L}^{\boldsymbol{D}} := \mathrm{diag}(\boldsymbol{l}),$$

and $\boldsymbol{R_i}$ the corresponding submatrix of $\boldsymbol{R}$ that consists of all elements $r_{s,d,i}$ with $1 \le s,d \le N_M$. We can select the set of local arrival rates Λ^r for a single router node r by exploiting the knowledge of the topology that is contained in the connectivity matrix $\boldsymbol{\Gamma}$. I.e. we collect all λ_i where i is the ID of an input edge of router r:

$$\Lambda^r := \{\lambda_i \mid \exists s; 1 \le s \le N_M + N_R; \gamma_{s,r} = i\}. \quad (3)$$

We continue to compute the forwarding probability matrix $\boldsymbol{F}$. The matrix element $f_{i,j}$ $(1 \le i,j \le N_E)$ can be defined as traffic intensity between router input i and router output j normalized to the total arrival rate at input i, i.e., λ_i:

$$f_{i,j} := \frac{\sum_{s=1}^{N_M}\sum_{d=1}^{N_M} l_s \cdot t_{s,d} \cdot r_{s,d,i} \cdot r_{s,d,j} \cdot \delta_{i,j}}{\lambda_i}, 1 \le i,j \le N_E. \quad (4)$$

We call the term $\delta_{i,j}$ the *link selector matrix*. It ensures that there is only a forwarding probability $f_{i,j} > 0$, if (i,j) represents an input/output link pair of the same router:

$$\delta_{i,j} := \begin{cases} 1, & if\ \exists s,r,d\ with\ \gamma_{s,r} = i\ \wedge \gamma_{r,d} = j \\ 0, & otherwise \end{cases}.$$

Therein, $\gamma_{s,r}$ and $\gamma_{r,d}$ are corresponding elements of the connectivity matrix $\boldsymbol{\Gamma}$. Equation (4) can be rewritten in matrix form:

$$f_{i,j} := \frac{\delta_{i,j}}{\lambda_i} \cdot \mathrm{tr}\left(\left(\boldsymbol{T}\cdot\boldsymbol{L}^{\boldsymbol{D}}\right)^T \left(\boldsymbol{R_i} \circ \boldsymbol{R_j}\right)\right), \quad (5)$$

where $\circ$ represents the entry-wise multiplication (i.e., the Hadamard product) of two matrices. Finally, we also restrict the set of forwarding probabilities F^r to a single router node r, similar to the approach in (3), and come to (6):

$$F^r := \{f_{i,j} \mid \exists s,d; 1 \le s,d \le N_M + N_R; \gamma_{s,r} = i \wedge \gamma_{r,d} = j\}. \quad (6)$$

B. An Analytic Router Model based on Queueing Theory

Based on the assumptions that we made in Section III, an M/M/1 queueing system [6] with exponential interarrival and service times will be appropriate to model the router behavior. However, in reality, the traffic situation within a router looks more complicated, as the example in Figure 2 (left) shows.

Therein, we find splitting and merging of traffic flows that contend with other input queues for multiple output ports. Furthermore, each input has different probabilities of being forwarded to a specific output. To be able to represent the router system by a queueing model, we propose using a simplified equivalent system, as depicted in Figure 2 (right). The idea is to include the contention delays into the service times and thereby receiving port specific service times. In fact, if a packet in front of a (FIFO) queue is blocked due to a contending queue, this is nothing else than a delayed service. Therefore, it is reasonable to consider the contention delay as a service time increase. Consequently, we come to a reduced

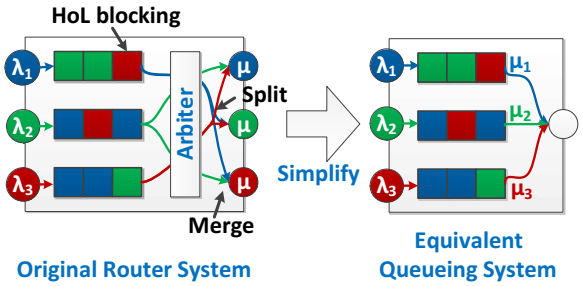

Figure 2. The equivalent queueing system simplifies a view onto the traffic situation in a router and can easily be expressed as a Markov model.

equivalent system that now only consists of a single output server and multiple input queues, each having an individual service time (and service rate μ_i respectively).

Due to the memoryless property of the exponentially distributed arrival and service processes, the state of the equivalent router system can now solely be defined by the number of flits contained in the input queues. If we represent the state by a vector where each element represents the fill level of a single input queue, we can model the system by means of a multidimensional Markov chain. This is illustrated in Figure 3 for the case of a router with two inputs (please ignore the depicted macro states for now). Therein, the transition rates are defined by the arrival rate λ_i and service rate μ_i for each input independently. Let $\boldsymbol{x}$ be the current state vector of the router. Then, a transition from state $\boldsymbol{x} \to \boldsymbol{x} + e_i$ (where e_i is the unit vector for dimension i) has an intensity of λ_i. On the other hand, a transition $\boldsymbol{x} \to \boldsymbol{x} - e_i$ has an intensity of μ_i. The boundaries of the Markov chain are an exception to that rule (first column/row in Fig. 3). There, we find a different contention situation. In the case of two inputs, we have no contention caused by the second input anymore. Therefore, the transition rates for $\boldsymbol{x} \to \boldsymbol{x} - e_i$ change to μ, i.e., the basic router service rate without contention delay.

For solving the Markov chain, we still need to know the

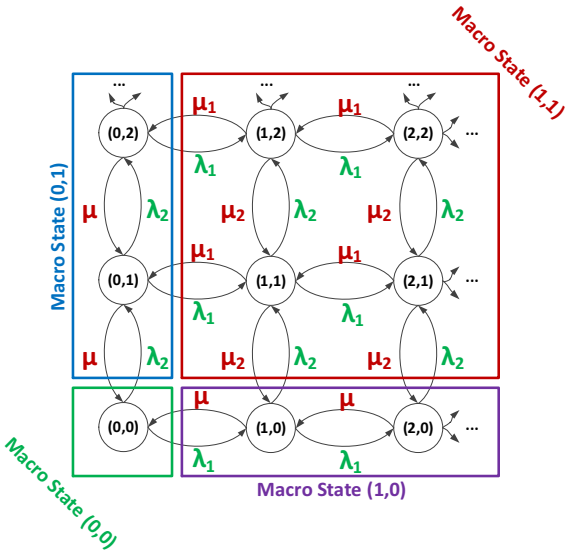

Figure 3. Example of a two-dimensional Markov model for a router with two inputs and the decomposition into reversible sub-chains.

 ISBN: 978-1-61208-234-9

service rates μ_i that include the contention delay to be able to define the transition rates. For this purpose, we apply an idea that was proposed in [10] to determine the mean waiting time for a similar input buffered router model assuming an FCFS arbiter. We modify this approach to find an estimation for the mean service time, i.e., the waiting time of the flit in front of the queue. Similar to [10], we first compute the pairwise contention probability $c_{i,j}$ for all inputs pairs (i, j) of a router with P inputs based on the forwarding probabilities $\boldsymbol{F}$ that can be derived according to (5):

$$c_{i,j} = \sum_{k=1}^{P} f_{i,k} f_{j,k}, i \neq j, 1 \leq i, j \leq P. \quad (7)$$

From (7), an equivalent matrix equation can be derived

$$\boldsymbol{C} = \boldsymbol{F} \cdot \boldsymbol{F}^T. \quad (8)$$

Note that the main diagonal of the contention probability matrix $\boldsymbol{C}$ in (8) is set to "1" which makes the following computation more convenient. Based on the contention probabilities, we can derive an expression to estimate the mean service times $\overline{x}_i(\boldsymbol{y})$ under contention:

$$\overline{x}_i(\boldsymbol{y}) := \overline{x} + \overline{x} \sum_{j=1, j\neq i}^{P} c_{i,j} y_j, 1 \leq i \leq P. \quad (9)$$

The first summand $\overline{x}$ of (9) represents the mean router service time for the packet in front of queue i. The second summand considers the contention delay. Therein, the vector $\boldsymbol{y}$ represents the instantaneous fill state of each input queue, i.e., $y_i = 0$, if input queue i is empty and does not contribute to the contention delay and $y_i = 1$ otherwise. We will call $\boldsymbol{y}$ the *router macro state* in the following and can directly derive it from the router state $\boldsymbol{x}$:

$$y_i = \begin{cases} 0, & if\ x_i = 0 \\ 1, & if\ x_i > 0 \end{cases},$$

or rather informally: $\boldsymbol{y} = \mathrm{sgn}(\boldsymbol{x})$.

We can still condense (9) somewhat by exploiting the convenient definition of contention probability matrix $\boldsymbol{C}$ and provide a short form matrix equation for the mean service rates $\mu_i(\boldsymbol{y})$ (i.e. the inverse of the mean service times):

$$\mu_i(\boldsymbol{y}) := \left[\frac{1}{\mu} \boldsymbol{C_i}^T \boldsymbol{y} \right]^{-1}, 1 \leq i \leq P. \quad (10)$$

With the definition for the mean service rates $\mu_i(\boldsymbol{y})$ in (10) we have now all necessary inputs to solve the Markov chain in order to obtain the steady-state probability distribution. However, in trying to do so, we are confronted with another challenge. If we apply the Kolmogorov criterion for reversibility of Markov chains, we soon realize that it does not hold for some cases in the peripheral region of our Markov chain. Accordingly, the chain is not time reversible; see Fig. 3 and examine the following state transitions: $(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (0,0)$, and the corresponding return path. We notice that the product of the transition rates is not equal for both directions, and thus, it does not fulfill the Kolmogorov criterion [14]:

$$\lambda_1 \cdot \lambda_2 \cdot \mu_1 \cdot \mu \neq \lambda_2 \cdot \lambda_1 \cdot \mu_2 \cdot \mu.$$

Consequently, we are not allowed to apply local balance equations to solve the chain. Unfortunately, we are not able to find a closed-form solution for the infinite Markov chain solely based on the global balance equations. Fehske and Fettweis [15] recently encountered exactly the same problem when trying to solve an equivalent Markov chain. They proposed an approximation to find a solution for the stationary distribution. The approach is based on the concept of *aggregation of variables* that is well known by economics for quite some years [16]. The proposed algorithm consists of the following steps.

We start decomposing our Markov chain into reversible sub-chains. This is done by collecting all states $\boldsymbol{x}$ that belong to the same macro state (or aggregate state) $\boldsymbol{y} = \mathrm{sgn}(\boldsymbol{x})$ in a common set $\mathcal{S}(\boldsymbol{y})$:

$$\mathcal{S}(\boldsymbol{y}) := \left\{ \boldsymbol{x} \in \mathbb{N}_0^P \mid \mathrm{sgn}(\boldsymbol{x}) = \boldsymbol{y} \right\}.$$

The idea behind the definition is that all states are collected in the a macro state where we find a similar contention situation. If we consider a contending queue, it doesn't matter how many packets it contains, only if it contains at least one packet or not. Consequently, the mean service rates are homogeneous within each macro state. An example for the Markov chain decomposition for the case of two input ports is provided in Figure 3. Therein, we decompose the two-dimensional Markov chain into four macro states. Macro state $(0,0)$ contains all states where both input queues are empty (which is only a single router state $(0,0)$). Macro states $(1,0)$ and $(0,1)$ collecting the states where only one of the two queues is empty. Hence, we have no contention within these two macro states. Macro state $(1,1)$ represents all router states where both queues are not empty. In this example, this is the only macro state where contention occurs.

Since the transition rates are homogeneous within each macro state, the sub-chains are reversible and can be solved. This leads to a product form solution for the stationary probability distribution of the number of customers (i.e. packets) $\widetilde{\pi}$ in an M/M/1 queueing system that is well known from classical queueing theory [6][15]:

$$\widetilde{\pi}(\boldsymbol{x}) = \begin{cases} \prod_{i \in N_1(\boldsymbol{y})} (1 - \rho_i(\boldsymbol{y})) \rho_i^{x_i - 1}(\boldsymbol{y}) \sigma(\boldsymbol{y}), & \text{for } \boldsymbol{y} \neq \boldsymbol{0} \\ \sigma(0), & \text{for } \boldsymbol{y} = \boldsymbol{0} \end{cases} \quad (11)$$

with utilization $\rho_i(\boldsymbol{y})$ of input queue i defined as

$$\rho_i(\boldsymbol{y}) := \frac{\lambda_i}{\mu_i(\boldsymbol{y})}.$$

Note that (11) only yields an estimate for the solution of the stationary probability distribution. This is because we omit the transitions between the macro states at this consideration. Also, note that (11) is conditioned on the probabilities of the corresponding macro state $\sigma(\boldsymbol{y})$ to ensure that $\sum_{\boldsymbol{x}} \widetilde{\pi}(\boldsymbol{x}) = 1$.

So far, we have no knowledge about the macro state probabilities $\sigma(\boldsymbol{y})$. We can compute $\sigma(\boldsymbol{y})$ by solving the (now finite) Markov chain on macro state level. Figure 4 shows a solution for the transition rate $p(\boldsymbol{y}, \boldsymbol{y}')$ from macro state $\boldsymbol{y}$ to macro state $\boldsymbol{y}'$, as provided by [15]:

$$p(\boldsymbol{y}, \boldsymbol{y}') = \begin{cases} \lambda_i, & \text{for } \boldsymbol{y}' = \boldsymbol{y} + e_i \\ \mu_i(\boldsymbol{y}) - \lambda_i, & \text{for } \boldsymbol{y}' = \boldsymbol{y} - e_i \\ 0, & \text{else} \end{cases}, \quad (12)$$

 ISBN: 978-1-61208-234-9

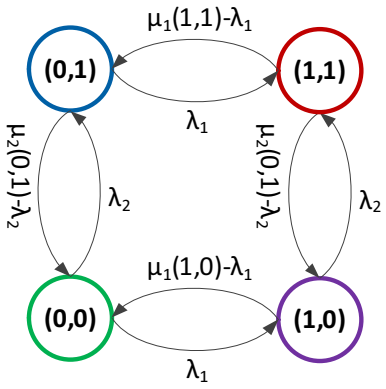

Figure 4. Example: Markov chain on macro state level assuming a router with two inputs.

where e_i again represents the unit vector for dimension i. Based on (12), we can now define the transition probability matrix $P = \left[p_{ij}\right]$ with $p_{ij} := p\left(y_i, y_j\right)$. With the definition of $p_{ii} := -\sum_{j=1}^{2^P} p_{ij}$ we normalize the row sum to 0.

Finally, we can follow the usual approach and solve the equation system for the vector of macro state probabilities σ based on the transition probability matrix P:

$$\sigma P = 0,$$

under the side condition $\sum_{\boldsymbol{y}} \sigma(\boldsymbol{y}) = 1$.

Based on (11), we can now compute the approximates for the state probabilities $\widetilde{\pi}(\boldsymbol{x})$. We can derive several key performance indicators, such as the mean number of packets in the queue $\mathbb{E}[x_i]$:

$$\mathbb{E}[x_i] \approx \sum_{\boldsymbol{x}} \widetilde{\pi}(\boldsymbol{x}) x_i = \sum_{\boldsymbol{y}} \frac{\rho_i(\boldsymbol{y})}{1-\rho_i(\boldsymbol{y})} \sigma(\boldsymbol{y}),$$

or the mean queueing delay W_i for input queue i by applying Little's law [6]:

$$W_i = \frac{\mathbb{E}[x_i]}{\lambda_i}.$$

V. Performance Evaluation

We show the accuracy of the proposed NoC model by comparing it against cycle-accurate NoC simulation. Due to the similar system model assumptions we decided to compare our approach against the model proposed in [10] as well as the NoC simulation tool that has been used therein [17].

We assumed following common simulation parameters:

- deterministic, dimension-ordered XY-routing,
- flit traffic, i.e., packet size = 1,
- input buffered routers with FCFS arbiter and service rates of $\mu = 0.5$,
- large buffer size (256 flits) to approximate the infinite buffer model and
- simulation run time of 10^5 cycles with a warm-up period of 10^4 cycles.

We investigate the following two topology/traffic scenarios under different load conditions (defined by number of injected packets/cycle) and compare the average packet transmission latency in the network.

First, we choose a very simple scenario to investigate the model behavior under a clear contention situation. Therefore, we consider a simple chain of four routers where a single PE is connected to each router. The PEs at routers 1 and 4 are sending their packets to PE 2 and 3 with a uniform distribution. PEs 2 and 3 do not send any packets. Hence, we find at router 2 and 3 a contention situation with the following forwarding probability matrix $\boldsymbol{F}$:

$$\boldsymbol{F} = \begin{pmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 1 & 0 & 0 \end{pmatrix}.$$

The result under different load conditions is shown in Figure 5. We find that the latency estimation for our proposed approach (red curve with + marker) follows very well the cycle-accurate simulation results (black curve with point marker) under a low and medium load condition. However, it significantly underestimates the network saturation limit where latency tends to infinity (0.66 packets/cycle in our model compared to 0.8 packets/cycle in the cycle-accurate simulation). The reference mean value model from [10] (blue curve with circle marker) shows a slight overestimation of the latencies under mid load conditions but estimates the network saturation point quite well.

The reason for the poor estimation of the network saturation point of our model is the applied aggregation approach for approximating the solution of a Markov chain. Therein, the stability of the overall solution is determined by the stability of the "worst-case" aggregate, i.e., the aggregate with the highest contention. If the solution for the "worst-case" aggregate tends to infinity the overall solution tends to infinity as well. To avoid this behavior, we propose to determine an average service time $\overline{\overline{x}}_i$ over all macro states for every router input. This is done by computing the expectation of the mean service times $\overline{x}_i(\boldsymbol{y})$ over all macro states based on the known macro state probabilities $\sigma(\boldsymbol{y})$:

$$\overline{\overline{x}}_i = \sum_{\boldsymbol{y} \in \{0,1\}^P} \overline{x}_i(\boldsymbol{y}) \sigma(\boldsymbol{y}) y_i. \tag{13}$$

Therein, y_i constrains the expectation to those macro states where queue i is not empty. We compute the average waiting time W_i for input queue i based on (13):

$$W_i = \frac{\overline{\overline{x}}_i}{1 - \lambda_i \overline{x}_i}.$$

The result of the refined approach is also depicted in Figure 5 (green curve with square marker). It shows a very good match compared to the cycle-accurate simulation. The latencies under low/mid load conditions, as well as the network saturation point, are estimated very accurately by this approach. The average estimation error is less than 3%.

Finally, we choose a 4x4 2D-mesh topology using a more diverse traffic pattern of the generic multimedia application from [10]. We target to compare the estimation quality of the average latencies under more complex contention situations. The results are plotted in Figure 6 and confirm the accurate results of the first scenario. Again, the average estimation error is around 3% (9% for the reference model). However, we still notice a slight underestimation of the network saturation limit of about 2.5% for that case. The reference mean value model shows a better accuracy in this region.

 ISBN: 978-1-61208-234-9

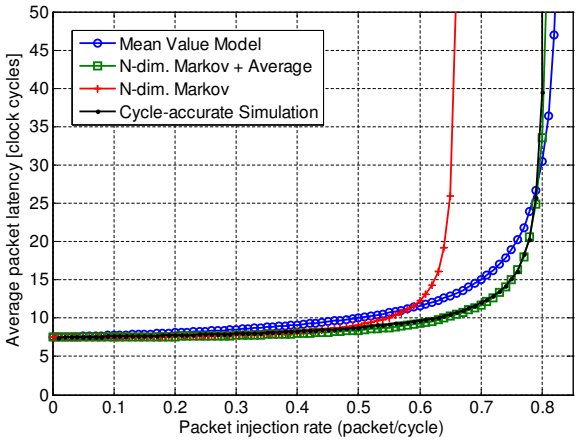

Figure 5. Performance results for 4x1 chain analyzing the average packet latency in comparison to cycle-accurate simulation.

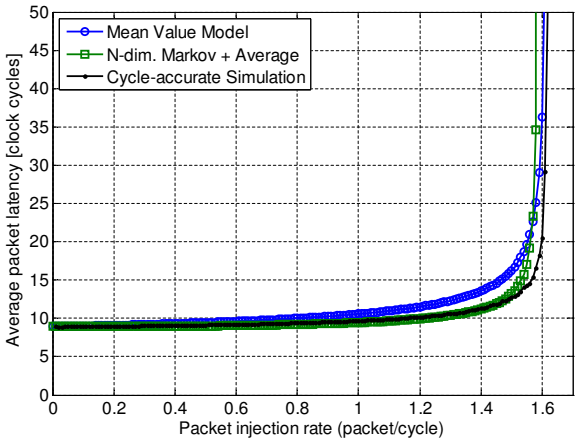

Figure 6. Performance results for 4x4 2D-mesh with generic multimedia application traffic analyzing the average packet latency in comparison to cycle-accurate simulation.

Note that the presented results only serve as proof of concept and easily scale to larger networks. The relative accuracy of the latency estimation is expected to stay in the same range under similar contention situations, independent of the NoC size. This is because the analysis of the queueing delay is done on router level and only accumulated on network level.

VI. Conclusion and Future Work

In this paper, we presented a novel analytic approach for modeling on-chip networks for many-core SoC based on queueing theory. In contrast to many existing models, the approach is very flexible in terms of supported topology, routing scheme and traffic pattern. The approach overcomes the limitations of the mean value analysis introduced in the existing work. Instead, it provides information about a steady-state distribution of the network routers. This allows to derive arbitrary key performance indicators, such as blocking probabilities or average queueing delays, which is very important information for dimensioning network resources, such as buffers, links, etc. We demonstrated the very high accuracy of the approach by comparison to a cycle-accurate simulation. The average estimation error for the mean latencies in a 4x4 2D-mesh is only 3%.

Many extensions of the NoC model are planned. We target to consider different arbitration schemes, such as the popular round-robin method. A finite buffer model extension would be interesting in order to model network acceptance behavior and back pressure effects. A generalization towards an arbitrary service time distribution is also desirable. Finally, supporting multiple clock domains (i.e., globally asynchronous locally synchronous systems) and frequency scaling is another open topic in order to explore a many-core NoC more accurately.

References

[1] T. Dresden, "Esf young investigators group; 3d chip stack intraconnects - 3dcsi," last visited on 15/10/2012. [Online]. Available: http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_elektrotechnik_und_informationstechnik/3dcsi

[2] J. Manferdelli, N. Govindaraju, and C. Crall, "Challenges and opportunities in many-core computing," *Proc. of IEEE*, vol. 96, no. 5, pp. 808 –815, May 2008.

[3] S. Borkar, "Thousand core chipsa technology perspective," in *Proc. of DAC*, 2007.

[4] L. Benini and G. De Micheli, "Networks on chips: a new soc paradigm," *Computer*, vol. 35, no. 1, pp. 70 –78, Jan 2002.

[5] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 3 –21, Jan. 2009.

[6] L. Kleinrock, *Queueing systems - 1 : Theory*. New York: Wiley, 1975.

[7] W. Dally, "Performance analysis of k-ary n-cube interconnection networks," *Computers, IEEE Transactions on*, vol. 39, no. 6, pp. 775 –785, Jun 1990.

[8] A. Kiasari, D. Rahmati, H. Sarbazi-Azad, and S. Hessabi, "A markovian performance model for networks-on-chip," in *Proc. of Euromicro PDP*, 2008.

[9] N. Nikitin and J. Cortadella, "A performance analytical model for network-on-chip with constant service time routers," in *Proc. of ICCAD*, 2009.

[10] U. Ogras, P. Bogdan, and R. Marculescu, "An analytical approach for network-on-chip performance analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 12, pp. 2001 –2013, Dec. 2010.

[11] M. Bakhouya, S. Suboh, J. Gaber, and T. El-Ghazawi, "Analytical modeling and evaluation of on-chip interconnects using network calculus," in *Proc. of NoCS*, 2009.

[12] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queueing systems for the internet*. Berlin, Heidelberg: Springer-Verlag, 2001.

[13] Seber and A. F. George, *A Matrix Handbook for Statisticians*. John Wiley & Sons, Inc., 2008.

[14] R. Nelson, *Probability, stochastic processes, and queueing theory / the mathematics of computer performance modeling*. New York ; Heidelberg [u.a.]: Springer, 1995.

[15] A. J. Fehske and G. P. Fettweis, "Aggregation of variables in load models for interference-coupled cellular data networks," in *Proc. of ICC*, 2012.

[16] H. A. Simon and A. Ando, "Aggregation of variables in dynamic systems," *Econometrica*, vol. 29, no. 2, pp. 111–138, Apr 1961.

[17] worm_sim, "Cycle-accurate noc simulator," last visited on 15/10/2012. [Online]. Available: http://www.ece.cmu.edu/~sld/software/index.php

 ISBN: 978-1-61208-234-9