A performance analytical model for Network-on-Chip with constant service time routers

Nikita Nikitin
Univ. Politècnica de Catalunya
Barcelona, Spain
nnikitin@lsi.upc.edu

Jordi Cortadella
Univ. Politècnica de Catalunya
Barcelona, Spain
jordicf@lsi.upc.edu

ABSTRACT

Performance models for Network-on-Chip (NoC) are essential for design, optimization and Quality of Service (QoS) assurance. Classical queueing theory has been often used to provide fast analytical models to estimate average performance. This paper presents a new analytical model that focuses on QoS assurance. It assumes that the NoC has an underlying synchronous behavior with constant service time routers. The comparisons with simulation results show a tangible improvement with regard to the classical M/D/1 models when estimating the worst-case latencies and queue delays. The model can be applied to any network modeled as a queueing system with constant-time routers.

1. INTRODUCTION

1.1 Network-on-Chip modeling

The rapid advance of VLSI technology demands efficient and scalable methods for the design of complex systems. In this context it is essential to have tools that can effectively estimate the cost of a system and enable a broad design space exploration. Performance is one of the crucial parameters in design. Simulators are commonly used to estimate the performance of complex systems when the analytical models are too abstract and distant from reality. However, simulators are very time consuming and not scalable.

Nowadays, complexity is handled by designing modular and scalable systems. The concepts of System-on-Chip (SoC) and Network-on-Chip (NoC) [3, 8] are often used to denote systems that are fully integrated on a chip through interconnect networks that can serve the communication demands among the different components of the system.

NoC provides a trade-off in terms of area and power when compared to the traditional bus or peer-to-peer connections. At the same time NoC requires a thorough design process including topology selection and mapping, link planning, routing, buffer allocation and various optimization tasks. The automation approaches for NoC design aim at optimizing the cost of the solution (generally power, area or traffic distribution) under a set of quality-of-service (QoS) and capacity constraints. Many of the design tasks can be efficiently solved through iterative optimization processes that enable to change the design solution in case some of the constraints are violated.

For guaranteeing the QoS constraints of an NoC, an accurate delay estimation of the transit time of the packages is required. The most common strategy is the use of simulation tools for delay estimation. While simulation provides highly accurate estimations, it is a very time consuming process that can hardly be used during the iterative exploration phase of the design. For this reason, various analytical models have been proposed as an alternative to replace simulation with efficient and reasonably accurate estimations.

The basic method of delay estimation by hop-count does not consider the contention latency at the input queues of the routers. This latency has a tangible impact in the transit time for medium and high traffic loads. The analytical models for NoC are aimed at predicting the contention delays. They can be classified into two groups: the probabilistic models and the ones based on queueing theory (QT) [6]. An example of probabilistic model can be found in [7]. In this work we focus on the QT methods for NoC delay modeling.

An important aspect of NoC design is that most of the on-chip interconnect networks are synchronous. An NoC can be represented by a system of synchronous routers with input queues (FIFOs) (see Fig. 1). When the packet length in the network is fixed, the service times of the routers are equal to a constant value. Thus a network can be modeled as a *constant service time system*. With this assumption, it is convenient that the models for NoC are suitable for discrete time analysis.

1.2 Previous work

The general approach for network modeling using QT considers that a router is a single server in which the input buffers are treated as queues. To calculate the transit time of a packet in the network with sufficient accuracy, the waiting time in the input buffers must be calculated and added to the hop-count delay.

In this work we use one of the common routing schemes: the wormhole routing [2]. Multiple analytical models have been proposed for wormhole routers, but most of them are based on the standard M/G/1 and M/M/1 models. In [9], an analytical model is presented for wormhole flow considering finite buffers. The model is not restricted to any topology, extends the M/G/1 router representation and assumes Pois-

ICCAD'09, November 2–5, 2009, San Jose, California, USA.

son processes at the inputs of every router. However, the authors estimate the average packet latency in the network that is not sufficient for the QoS design parameters that need to guarantee individual end-to-end delay constraints. Another simplified M/G/1-based model can be found in [4]. In this case, the Poisson assumption is still required for the router inputs and, thus, the constant service time cannot be considered. In [12] an M/M/1 approximation of link delay is used for the capacity and flow allocation task that can be applied to general networks. Finally, [5] extends the approximation of the M/M/1 model by an empirical estimation for capacity allocation under the assumption of finite buffers.

1.3 Contribution

All the previously mentioned approaches use an assumption that is common to all NoC analytical models: the process at every router input has a Poisson distribution. While this is an acceptable assumption for the traffic sources (by definition of the model), it is known that the service times become correlated with the packet length as the packet propagates over the network [1]. According to this fact the distribution of the flow for the intermediate routers changes. To relax this effect, the widely applied *Kleinrock independence approximation* allows one to treat the input flows at intermediate routers still having Poisson properties. This approximation is reasonable when the packet lengths have a distribution close to exponential so that the packet service times are nearly exponential as well. However, as simulations show in the common situation of fixed packet length, this assumption makes the analytical model too pessimistic predicting too high waiting times.

This paper presents a QT-based analytical model for a constant service time network, i.e. a network modeled as a system with constant-time routers. Each router is treated as a QT server that has fixed service time T for the incoming packets. This is a reasonable assumption since we want to model networks with fixed packet lengths. Unlike the other analytical models for NoC, we do not use the Kleinrock approximation. Instead, we present simple and accurate empirical equations to estimate delays at the input buffers. Our contribution is an accurate analytical model for constant service time network. It is important to note that the scope of the presented methodology goes beyond NoC design and can be applied to any queueing system with constant-time servers satisfying the same set of assumptions.

1.4 Paper organization

The papers starts in Sect. 2 by introducing the problems that the classical QT methods manifest when modeling networks with constant-time routers. We illustrate our observations with a simple example. An explanation about how to adjust the model is presented.

Section 3 presents the main contribution of the paper, starting with a simple 2-input router and generalizing for n-input routers. The extended equations for estimating the waiting times are also discussed.

Section 4 presents a network as a system of constant-time routers. It discusses how to estimate the contention delay at a particular channel as well as the full net delay assuming a wormhole routing strategy.

Finally, Sect. 5 compares the accuracy of the analytical model with simulations carried out for a wide range of input

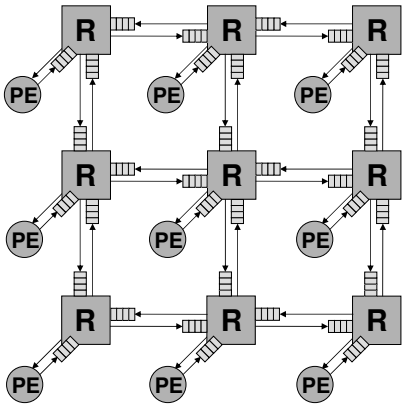

Figure 1: 3x3 mesh Network-on-Chip example.

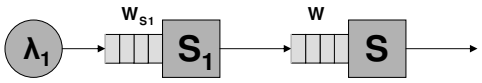

Figure 2: A simple server chain.

parameters. A comparison with the classical M/D/1 model is also performed and discussed.

2. MODEL OVERVIEW

Modeling a network as a system of routers requires an approach for calculating packet delays depending on the network topology. An example of a 3x3 mesh network is presented in Figure 1. Each router (R) is connected to a processing element (PE) and a set of neighboring routers by two unidirectional links. In our model we focus on the delay estimation (waiting time) at the input buffers. Hence, we assume each router to have an input buffer from each neighbor and one from the PE.

A net is a end-to-end route in the network, from one PE to another, and can be represented as a sequence of routers that a packet must propagate through. Each router may have an arbitrary number of input channels.

Consider the simple example in Fig. 2: a chain of two servers with constant service time T and a Poisson arrival process with rate λ_1. The classical QT model for a constant-time server with Poisson input is the M/D/1 model. The waiting time in the first input queue W_{S1} can be estimated using Pollaczek-Khinchin (P-K) formula [6]:

$$W = \frac{\lambda T^2}{2(1-\lambda T)}. \quad (1)$$

However, unlike the systems with exponentially distributed service times, the output from an M/D/1 system is no longer a Poisson process: the time between two consecutive outputs is guaranteed to be greater than or equal to the service time value T. In other words, a constant-time server produces a *de-randomization* of the Poisson process, thus reducing the degree of randomness of the inter-arrival times, which is the main cause of contention delays at the input queue.

While W_{S1} can be accurately predicted with (1), using

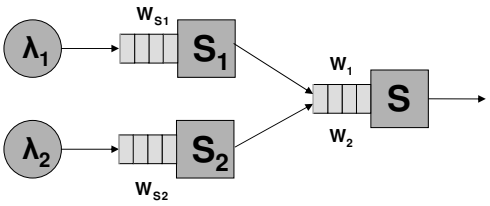

Figure 3: Merging two M/D/1 output flows.

the same equation to estimate the waiting time at the input queue of S may result in significant errors. More precisely, the waiting time W will be equal to zero as the successive arrivals of packets at S occur not earlier than T, which is exactly the time to process a packet by server S. That means that the incoming packet will never wait for his service. This fact illustrates the inaccuracy of applying (1) for the estimation of the waiting time at server S.

In the general case, a router may have an arbitrary number of inputs, including constant-time server output flows. The goal of this work is to derive the equations for an accurate queueing delay estimation, as an alternative to the P-K equation.

The qualitative importance of an accurate queueing delay estimation is shown in the following example. Consider now two inputs to server S, each one being an output from the constant service time systems S1 and S2 (Fig. 3). Table 1 reports the estimation of the waiting times W_1 and W_2 by the M/D/1 model and the constant service time model (CTM) of this paper depending on different input rates (λ_1 and λ_2). The total delay at the server for the packets of the first flow is $D_1 = T + W_1$, where T is the packet service time (assumed to be 1 cycle for the example).

The traffic rates of the flows (flits/cycle) are specified in the first two columns of the table. The following three columns show the waiting times and the delay of the first flow (in cycles) obtained by the CTM model. Next, the estimations for W_1, W_2 and D_1 obtained with the M/D/1 model are reported. The column `SIM` displays the delay of the first flow obtained by simulation. Finally, the error of D_1 with regard to the simulation is reported in the last two columns.

One can easily observe the reduction of waiting times produced by the de-randomization (W_1, W_2 of CTM vs those by M/D/1). Another important aspect is that the Poisson-input models assume equal waiting times for all input streams regardless their rate. This is not true in the case of constant service time systems. This results into larger differences in delay estimation (see the difference between W_1 and W_2 in the last row of the CTM model). We also note that the difference between our model and simulation is less than 1% in this example, while M/D/1 reveals up to 36% of overestimation. This simple example shows that proposed CTM approach can provide more accurate estimations. This is essential for QoS optimization.

Table 1: Modeling results for the example in Fig. 3.

		CTM			M/D/1		SIM	Error	
λ_1	λ_2	W_1	W_2	D_1	W_1, W_2	D_1	D_1	CTM	M/D/1
0.1	0.1	0.07	0.07	1.07	0.13	1.13	1.07	0.1%	6%
0.3	0.3	0.53	0.53	1.53	0.75	1.75	1.54	0.7%	14%
0.5	0.1	0.29	0.47	1.29	0.75	1.75	1.29	0.3%	36%

3. QT MODEL

This section presents the main contribution of this paper: the QT model for constant-time routers.

3.1 Definitions and assumptions

We use the following definitions:

- A *router* is a basic entity that routes traffic in a network. The router is also referred to as *server* in the nomenclature of queueing theory.
- A *packet* is a data transmission entity. A packet consists of one or more *flits* that are the minimum transmission units.
- An *input flow*, also referred to as *input process*, is an arrival process at one of the router input buffers.
- A *traffic source (sink)* of the net is a processing element that injects (consumes) packets to (from) the network.
- The *traffic rate* λ_k of the net k is the average rate of packet generation at the net source.
- The *waiting time* at the input buffer of a router is the average steady-state time the packets spend in the buffer before being processed by the router. In the QT nomenclature it is referred to as a *queueing delay* at the input queue (buffer).

The following assumptions are considered:

- Traffic sources generate packets according to a Poisson distribution.
- Traffic sinks consume packets immediately.
- The input buffers of the routers have infinite capacity.
- The packets have fixed size and the routers take constant time to process them.

3.2 A simple model for a 2-input router

We have already stated an important difference between a system with exponentially distributed service times and one with constant-time service, assuming a Poisson input to both. The output process from the former is also a Poisson process of the same rate and exponentially distributed interarrival times. Thus we may use (1) for M/M/1 systems to estimate the waiting time for any of the routers. The output flow from a constant service time system with Poisson input has a complex distribution discussed in [10] and [11]. The important property due to its deterministic service time is that the time between two successive outputs is not less than the service time T. Because of this fact, the waiting time for all the routers in a chain will be equal to zero except for the first one (Fig. 2). The first router delay can be successfully estimated with (1), since the packet generators are said to generate packets according to a Poisson distribution.

Consider the two-input server example presented in Fig. 3. All three servers S1, S2 and S, have a constant service time T. Both sources generate packets assuming a Poisson distribution with average traffic rates λ_1 and λ_2. Our goal is to model the waiting times W_1 and W_2 at the input queue of the server S.

For example, note that if the flow λ_2 were not sending packets, i.e. $\lambda_2 = 0$, then W_1 would be equal to zero as in the single-input server case. This fact supports the idea that the waiting time W_1 is generated by the packets of the complementary flow λ_2 and its value depends on the traffic rate of both flows.

To simplify the analysis, we use the concept of *mean residual service time* $R(\lambda)$ for an input flow [1]. If incoming packet P_i arrives at the server queue at time t_i while some other packet P_j is being processed by the server, then the residual time R_i for the packet P_i is the time left for P_j to finish its service. The mean residual service time is an average value of residual times for each packet defined by the service time and the flow traffic rate. The following equation represents its steady-state value [1]:

$$R(\lambda) = \frac{1}{2}\lambda T^2. \tag{2}$$

Using the definition of residual time, the Pollaczek-Khinchin equation can be rewritten as

$$W(\lambda) = \frac{\lambda T^2}{2(1-\lambda T)} = \frac{R(\lambda)}{1-\lambda T}. \tag{3}$$

We generalize the above expression by distinguishing traffic rates in the P-K formula. Let us consider the traffic flow of rate λ_{tr} that is merged with some complimentary flow of rate λ_{res} at the router input. As our experiments show, the waiting time for the packets of flow λ_{tr} will depend on both rates. Then we can rewrite (3) as

$$W(\lambda_{tr}, \lambda_{res}) = \frac{\lambda_{res} T^2}{2(1-\lambda_{tr} T)} = \frac{R(\lambda_{res})}{1-\lambda_{tr} T}. \tag{4}$$

This generalized waiting time can be treated as that of the traffic flow λ_{tr} experiencing a delay produced by the complementary flow with rate λ_{res}, inducing a residual time $R(\lambda_{res})$. Using (4) in combination with the standard M/D/1 equation (3) we propose the following empirical expression that was found to provide an accurate estimation for W_1 and W_2:

$$\begin{aligned} W_k &= W(\lambda_1+\lambda_2) - W(\lambda_1) - W(\lambda_2) + W(\lambda_k, \lambda_j) = \\ & W(\sum_{i=1,2}\lambda_i) - \sum_{i=1,2} W(\lambda_i) + W(\lambda_k, \lambda_j), \end{aligned} \tag{5}$$

where $k \in \{1,2\}$ and j is the complementary flow for k.

The form of (5) was suggested intuitively resulting from the numerous experimental observations and was further empirically verified for a large range of input parameters λ_i and T. The intuition behind this equation can be explained with the following considerations. Expression (5) has three terms: the first one is the M/D/1 waiting time (3) that input packets would observe if both inputs were Poisson processes. The second term is the sum of the M/D/1 waiting times for each separate flow. It can be considered as the measure of "de-randomization" introduced by the source constant-time router. In fact this is the waiting time packets of each input process spend at the source router. The last term estimates the impact of the complementary inputs on the k-th input, similarly as it was discussed in (4).

An important fact that is not evident and not observed in M/D/1 systems modeling is that the waiting time for the input flows differs when the traffic rates are not equal. This phenomenon is also proved by simulation. When two flows interact at the router input, the packets of the flow with greater traffic rate have less average input delay. Indeed, if a packet belongs to the flow with the smallest rate it has greater probability to be blocked by a packet of the complementary flow. As a result, in average its waiting time will be greater than that of the packet of complementary flow. An illustration of this fact is presented in Fig. 4. The traffic rate ratio λ_1/λ_2 ranges from 1 to 10 while their sum is kept constant ($\lambda_1 + \lambda_2 = 0.4$ flits/cycle). The waiting time W_1, W_2 estimations by both models are depicted.

One can observe the increasing difference between W_1 and W_2 estimated by the CTM model as the rate ratio increases. In contrast, the M/D/1 model provides a pessimistic constant waiting time for both flows that does not depend on the ratio of traffic rates.

3.3 Generalization for an N-input router

Equation (5) can be generalized for an arbitrary number of inputs $N > 2$. The waiting time W_k^N at input k can be calculated as

$$W_k^N = W(\sum_{i=1}^{N}\lambda_i) - \sum_{i=1}^{N} W(\lambda_i) + W(\lambda_k, \sum_{i=1, i\neq k}^{N}\lambda_i) \tag{6}$$

It is also valuable to notice that this equation holds for the case of one input flow ($N = 1$). In this case we get $W_1^1 = 0$ that stands in correspondence with zero delay at all routers in a chain, except for the first one.

3.4 Hybrid process at the router input

Another important case to consider when modeling a network is a hybrid input process consisting of Poisson flows and constant-time router outputs. It is necessary for modeling the traffic coming from two different sources: the Poisson processes from the PE's and the traffic coming from constant-time routers. We start again considering a simple two-input server example where one net is an M/D/1 output and another is a Poisson source (Fig. 5).

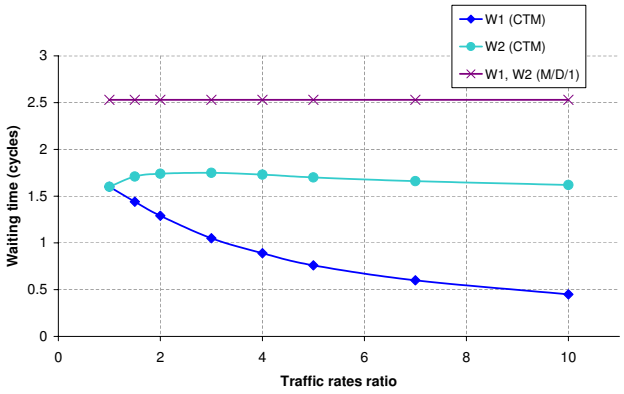

Figure 4: Estimated waiting time for different traffic rates ratio.

Combining (2), (3) and (4) we obtain the following expressions:

$$W_1 = W(\lambda_1 + \lambda_2) - W(\lambda_1) - R(\lambda_2) + W(\lambda_1, \lambda_2), \quad (7)$$

$$W_2 = W(\lambda_1 + \lambda_2) - W(\lambda_1) + R(\lambda_1). \quad (8)$$

This result can be extended to the general case presented in Fig. 6. Consider a complex arrival process at the input queue of server S: an arbitrary number d of M/D/1 output flows (λ_1, .. , λ_d) and p Poisson flows (λ_{d+1}, .. , λ_{d+p}). Let us denote the set of all M/D/1 outputs as D and set of all Poisson sources as P. Let also $N = d + p$ be the total number of inputs.

As the sum of Poisson processes with rates $\lambda_k,\ k = d + 1,\ ..\ , d + p$ is also a Poisson process of the total rate $\lambda_p = \sum_{k=d+1}^{d+p} \lambda_i$, we can treat the source flows as a single input with the total rate λ_p. This results into packets of all the flows in P experiencing equal waiting time.

Finally we present the generalized equations for the waiting time W_k^N estimation. For any input flow $\lambda_k \in D$

$$W_{k,k\in D}^N = W(\sum_{i=1}^{N} \lambda_i) - \sum_{i\in D} W(\lambda_i) - \sum_{i\in P} R(\lambda_i) + W(\lambda_k, \sum_{i=1, i\neq k}^{N} \lambda_i), \quad (9)$$

and for any Poisson input flow $\lambda_k \in P$

$$W_{k,k\in P}^N = W(\sum_{i=1}^{N} \lambda_i) - \sum_{i\in D} W(\lambda_i) + \sum_{i\in D} R(\lambda_i). \quad (10)$$

We note that in case $P = \emptyset$, equation (9) reduces to (6), thus demonstrating that (9) is the generalized case of the server not having traffic source inputs. Also note that in case of pure Poisson input flows, i.e. $D = \emptyset$, equation (10) is reduced to

$$W_{k,k\in P}^N = W(\sum_{i=1}^{N} \lambda_i), \quad (11)$$

that is exactly the waiting time expression for the M/D/1 system. Thus, *our equations are consistent with the M/D/1 model and provide a simple and accurate extension for the hybrid case of input processes.* Another interesting fact is that we can apply (10) for waiting time estimation at the sources of the nets and at any point in which the input traffic can be modeled by a pure Poisson process estimated by the Pollaczek-Khinchin formula (1).

We are using the pair (9)-(10) to predict contention delays at the input buffers of the network routers and build our delay model around them. As we show in the experimental section of the paper these equations allow accurate estimations for a wide range of input model parameters.

4. NETWORK MODEL

4.1 Router model abstraction

We represent a network as a system of connected routers. Given the single router model presented in the previous section, we now use it to model the behavior of a network. We use the router model under the following assumptions:

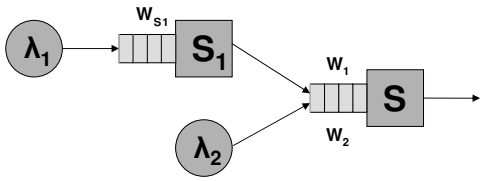

Figure 5: Merging an M/D/1 output and a Poisson flow.

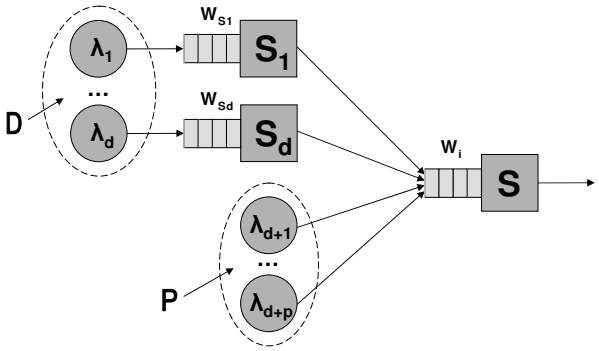

Figure 6: Hybrid input process at a constant-time router.

- The router is regarded as a black-box with constant service time for incoming packets and infinite capacity buffers.
- The router can only transmit one packet at a time. Simultaneous packet transmission is not considered.
- The strategy of processing inputs in order may vary. We discuss it in the experimental section.

4.2 Extension of the model for a network

As we have already discussed, (9) and (10) provide the waiting time estimation at the input of an arbitrary router depending on the network topology and traffic rates at the inputs. The traffic process in a network is a complex process that is the result of merging and splitting the traffic flows of individual nets. In this paper we address the waiting time estimation at the input buffers in case of merging N flows at the router input but we do not discuss the split process. However, as will be shown in the experimental section, by only considering merging processes we already obtain a significant accuracy in the estimation.

An important feature of a Poisson process served by a constant-time router is that it accumulates "de-randomization" introduced by the latter. Formally, let us consider N processes that have constant-time outputs, each one having some traffic rate λ_i. Once these processes have been merged at the constant-time router they become a single constant-time output process with the rate $\lambda = \sum_{i=1}^{N} \lambda_i$ that satisfies (9)-(10). We represent this fact in Fig. 7. Here two flows λ_1 and λ_2 that are M/D/1 outputs merge at router S to form a single flow at its output. We use (9) to estimate the waiting times at router S. Now the new flow travels to the router S',

where it merges with the flow λ_3. In order to estimate the waiting times at router S', we apply (9) again assuming two input flows with rates $\lambda = \lambda_1 + \lambda_2$ and λ_3.

This fact basically says that the waiting time of a constant service time system output process does not depend on the way it propagates in the network but only on the traffic rate at a particular router. As a result, we do not require information about the flow propagation and are able to calculate delays at each router independently. Below we show how independent router delays are joined to form the end-to-end delay of each net.

4.3 Net delay estimation for wormhole routing

The estimation of the full delay for a particular net is performed based on the wormhole routing strategy. In wormhole routing, a packet is transmitted by processing the request of its first flit, also referred to as a *header flit*. The header flit notifies the router to be served as soon as it arrives at the router input queue. Once the router has granted the connection to the header flit, the rest of the packet flits follow in a pipeline manner. To introduce the equation for the delay we use following nomenclature:

- The *routing path* P_i of a net i is the sequence of routers traversed by the packets of this net, from source to destination routers (including both).
- The *packet size* S represents the number of flits in a packet. The packet size includes the header flit and is a constant value for all packets in our model.
- The *header service time* HS denotes the time necessary for the router to grant a connection, i.e. find the appropriate output channel and establish the connection. HS does not include the waiting time at the queue.
- The *flit transmission time* FT is the time necessary to transmit one flit that is not a header in a pipeline manner. In our model we assume $FT = 1$ cycle.
- The *end-to-end average packet delay* D_i of a net i is the average time the packets of the net spend in the network, starting from the injection at source router and exiting at the output of the destination router.
- The *average waiting time* W_{ij} at router $j \in P_i$ of the packets of net i is the waiting time the packet header spends in the router queue before being processed.

The end-to-end net delay model for wormhole routing that we use incorporates two terms. The first term depends on the topology and geometrical distance between the net endpoints, thus it can be predicted statically. It is usually referred to as a hop-count delay D_i^{hc} of net i. As follows from the wormhole routing strategy, the hop-count delay consists of the time to propagate header (HS) and the time necessary for the rest of packet flits to reach the destination router:

$$D_i^{hc} = \sum_{j \in P_i} HS + FT(S-1) \tag{12}$$

The second term of the net delay is the contention delay D_i^c, that is the sum of the input buffer delays W_{ij} over all routers in the path P_i:

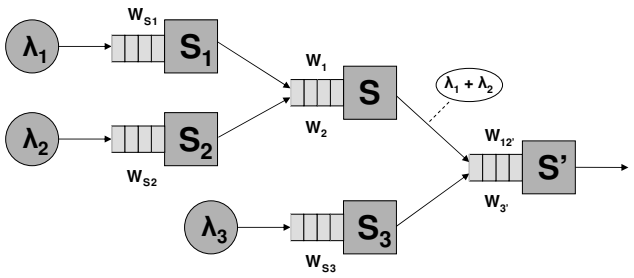

Figure 7: Successive flow merge.

$$D_i^c = \sum_{j \in P_i} W_{ij} \tag{13}$$

The contention delay occurs when several input packets compete for the same router outputs. It is estimated with the model we present in Sect. 3. Hence, the full end-to-end delay of net i is defined by the following expression:

$$D_i = D_i^{hc} + D_i^c = \sum_{j \in P_i} (HS + W_{ij}) + FT(S-1) \tag{14}$$

Finally we use (14) to calculate the end-to-end delay of every net assuming that (9)-(10) provide the waiting times W_{ij}. The constants HS, FT and S are the input parameters. According to the wormhole strategy, the packet router service time T used in the calculation of W_{ij} is the sum of the header service time and the propagation time for the remaining flits. Formally, T is defined as:

$$T = HS + FT(S-1). \tag{15}$$

5. EXPERIMENTAL RESULTS

Traffic flows in a network are complex flows that can be described as a system of merging and splitting processes at every router. The delays experienced by every input also depend on the priority scheme of the merging inputs. As it was discussed, the equations (9)-(10) provide an estimation of the waiting time in the input buffers of constant-time router assuming a first-come, first-served (FCFS) scheme of merging the arrival processes at the router inputs.

In this section we first show the accuracy of our equations by analyzing networks characterized by the merge of processes. We present an experimental proof of the fact that our equations are capable of estimating the delay within a wide range of input parameters such as network load and packet size. Then we also compare our model with M/D/1 for different priority schemes at the router inputs, namely *round-robin* (RR) and *longest queue* (LQ). LQ stands for prioritizing service of the input with the largest number of flits in the buffer. Finally we apply (9)-(10) to model arbitrary networks and show that our equations provide a significant improvement in the estimation of end-to-end delays, even without a detailed analysis of network flow splitting.

An important fact about the experiments is that we investigate *worst-case* delay errors over the network, i.e. the largest error estimating the net (buffer) delays from all nets

(buffers) in the network. Many of the models suggested so far only estimate average net delays. This is not suitable for QoS evaluation with end-to-end delay constraints.

We compare our CTM model and the classical M/D/1 model with the simulations performed by an accurate flit-level simulator written in C++. Even though we use mesh topologies in the experiments, our methodology is applicable to any type of network since the delay estimation is independent from the network topology (discussed in Sect. 4). The simulations on meshes are performed to simplify the benchmark suite and the architecture of the simulator.

5.1 Single-output router networks

This is a special type of networks we use to emphasize the accuracy of our model for constant service time networks. As the pair (9)-(10) provides delay estimation for a merging process, we first focus our analysis on this type of networks to avoid splitting at the output of the router, i.e. every router sends packets only to one direction. We start with a 3x3 mesh network with uniformly distributed traffic and FCFS priority scheme. In this experiment we measure the deviations between the net and buffer delay values obtained by simulation and estimation by CTM and M/D/1 models. We show the modeling error dependency on the maximum router utilization (flits/cycle) determined by the network load.

5.1.1 Packet size variations

Varying the packet size from 1 to 100 flits and the router utilization from 0.10 to 0.90 flits/cycle, we note that the CTM worst-case error does not exceed 0.25% for net delays and 2% for buffer delays, while the errors produced by the M/D/1 model are 9% for net delays and almost 100% for buffer delays. In the other experiments we fix packet size to 5 flits per packet.

5.1.2 Changing priority schemes

Although our experiments assume a FCFS priority scheme for the router inputs, we show that it can also be a good approximation for other schemes such as RR and LQ. Table 2 presents the relative errors of the worst-case estimation of net and buffer delays versus simulation at different traffic loads.

The first two columns of the table represent the utilization and the priority scheme. The values in the other columns report the errors between the estimated (D_{est}) and simulated (D_{sim}) delays, calculated as

$$Err = \frac{|D_{sim} - D_{est}|}{D_{sim}} \cdot 100\%. \quad (16)$$

The third and fourth columns represent the worst-case errors of the net and buffers delays estimated by the CTM model with respect to simulation. The last two columns are the errors of the M/D/1 estimation. One can see a significant improvement in the accuracy of CTM model against the classical M/D/1 model assuming low and medium traffic loads (utilizations). At high loads, the FCFS scheme still provides a high accuracy. For the other schemes, the gap between CTM and M/D/1 tends to reduce.

5.1.3 Non-uniform traffic

In this experiment (Table 3), we change the traffic distribution to be highly non-uniform. We observe that the M/D/1 model provides an overly pessimistic estimation of the delays for a medium traffic load. On the other hand, the CTM model still provides an accurate estimation for the FCFS scheme.

The difference between the CTM and M/D/1 models increases for all priority schemes. Hence, the CTM model is more accurate for the delay estimation assuming a non-uniform traffic distribution.

5.2 General networks

In this section, we present results for two general networks without the single-output router assumption. Although our model does not consider the splitting of the traffic flows, its application improves the delay estimation in comparison with the M/D/1 model, even for general networks with arbitrary configuration.

Table 4 represents the results for 3x4 mesh with highly communicating central nodes. The conclusions for this experiment are similar to those for the previous experiments. The relative error is sometimes reduced by several tens of percent.

Table 2: Relative delay error between simulation and analytical models for a 3x3 single-output network.

Utilization (flits/cycle)	Priority scheme	CTM error (%)		M/D/1 error (%)	
		net	buffer	net	buffer
0.10	FCFS	0.06	1.67	1.39	95.46
	RR	0.05	2.55	1.39	94.22
	LQ	0.05	2.58	1.41	95.29
0.42	FCFS	0.07	0.76	4.94	75.88
	RR	0.29	6.12	5.21	86.18
	LQ	0.66	5.24	5.49	84.63
0.84	FCFS	0.22	0.60	4.93	57.19
	RR	15.48	32.38	19.19	72.19
	LQ	12.85	20.80	13.30	68.18

Table 3: Relative delay error between simulation and analytical models for a 3x3 network with non-uniform traffic.

Utilization (flits/cycle)	Priority scheme	CTM error (%)		M/D/1 error (%)	
		net	buffer	net	buffer
0.42	FCFS	0.41	1.19	19.91	102.54
	RR	5.38	18.45	21.08	105.31
	LQ	13.40	37.45	23.45	148.34

Table 4: Relative delay error between simulation and modeling for a 3x4 network with highly communicating central nodes.

Utilization (flits/cycle)	Priority scheme	CTM error (%)		M/D/1 error (%)	
		net	buffer	net	buffer
0.12	FCFS	0.34	19.02	1.97	99.92
	RR	0.58	27.96	2.23	114.94
	LQ	0.68	25.86	2.25	111.41
0.50	FCFS	2.01	14.15	7.05	71.41
	RR	5.75	34.89	9.39	85.60
	LQ	3.93	31.21	8.83	84.74
0.75	FCFS	4.60	20.07	10.02	68.64
	RR	23.63	68.78	27.71	89.77
	LQ	11.71	45.58	15.22	86.50

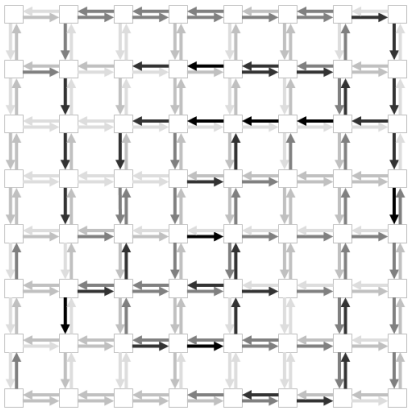

Figure 8: Traffic distribution for an 8x8 mesh NoC example.

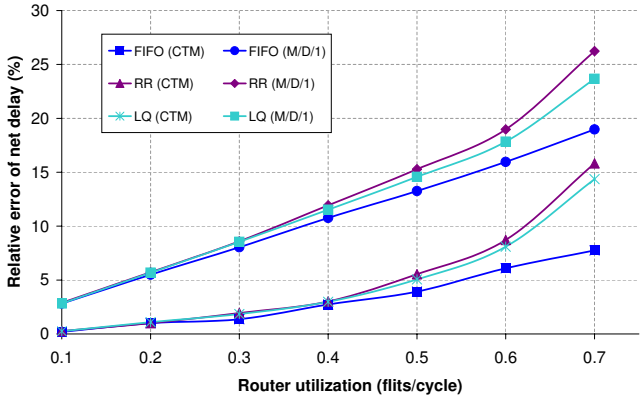

Figure 9: Relative errors between simulation and modeling for 8x8 NoC.

A large example: We also present comparative results for an 8x8 mesh NoC under variety of traffic loads. We have generated an arbitrary 8x8 network with 64 nets that are randomly distributed over the network, each net having the same traffic rate. Fig. 8 depicts the traffic distribution between the network links for this particular example, so that the darker arrows correspond to the links with higher traffic rates. The experiments were carried out for the utilizations in the range of 0.1 to 0.7 flits/cycle. The relative errors for the net delays are shown in Fig. 9. There is a tangible reduction of the error of CTM model at low and medium loads when compared to M/D/1 model. Another important fact is that the simulation of every configuration took several minutes to ensure a level of confidence smaller than 1%. The application of the CTM model took about 0.3 msecs.

To sum up, the CTM model with different priority schemes provides a notable improvement in accuracy in comparison with the M/D/1 model within wide range of traffic loads.

The buffer delay estimation improvement reaches up to several times in absolute value, while the net delay mainly determined by the hop count at low and medium loads improves up to several tens of percent. This is a significant result for designs that have QoS constraints for end-to-end latencies.

6. CONCLUSIONS

We have addressed the problem of modeling constant service time systems via queueing theory. A novel performance analytical model for Network-on-Chip has been presented. Unlike the classical approaches based on the M/D/1 model, the new method eliminates the Poisson assumption for packet distributions at the intermediate routers. It provides an accurate empirical model to estimate the input queue contention delays. The relevance of the model is emphasized by the capability to be used as a simple and accurate approximation of the queueing delay for constant-time routers, which is a realistic assumption for synchronous systems. Adjusting the NoC model with these expressions provides a tangible accuracy increase within a wide range of traffic loads. The suggested methodology can be applied to any network modeled as a system of constant-time routers.

7. REFERENCES

[1] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

[2] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Trans. Comput.*, 36(5):547–553, 1987.

[3] W. J. Dally and B. Towles. Route packets, not wires: on-chip inteconnection networks. In *DAC '01: Proceedings of the 38th conference on Design automation*, pages 684–689, New York, NY, USA, 2001. ACM.

[4] J. T. Draper and J. Ghosh. A comprehensive analytical model for wormhole routing in multicomputer systems. *J. Parallel Distrib. Comput.*, 23(2):202–214, 1994.

[5] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. Efficient link capacity and qos design for network-on-chip. In *DATE '06: Proceedings of the conference on Design, automation and test in Europe*, pages 9–14, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association.

[6] L. Kleinrock. *Theory, Volume 1, Queueing Systems*. Wiley-Interscience, 1975.

[7] O. Lysne. Towards a generic analytical model of wormhole routing networks. *Microprocessors and Microsystems*, 21(7-8):491 – 498, 1998. IEEE 1355.

[8] G. D. Micheli and L. Benini. *Networks on Chips: Technology and Tools (Systems on Silicon)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[9] U. Y. Ogras and R. Marculescu. Analytical router modeling for networks-on-chip performance analysis. In *DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 1096–1101, San Jose, CA, USA, 2007. EDA Consortium.

[10] C. D. Pack. The effects of multiplexing on a computer-communications system. *Commun. ACM*, 16(3):161–168, 1973.

[11] C. D. Pack. The Output of an M/D/1 Queue. *OPERATIONS RESEARCH*, 23(4):750–760, 1975.

[12] E. C. G. Wille, M. Mellia, E. Leonardi, and M. A. Marsan. Algorithms for ip network design with end-to-end qos constraints. *Comput. Netw.*, 50(8):1086–1103, 2006.