A Comprehensive Power-Performance Model for NoCs with Multi-Flit Channel Buffers

Mohammad Arjomand, Hamid Sarbazi-Azad
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
arjomand@ce.sharif.edu, azad@shairf.edu

ABSTRACT

Large Multi-Processor Systems-on-Chip use Networks-on-Chip with a high degree of reusability and scalability for message communication. Therefore, network infrastructure is a crucial element affecting the overall system performance. On the other hand, technology improvements may lead to much energy consumption in micro-routers of an on-chip network. This necessitates an exhaustive analysis of NoCs for future designs. This paper presents a comprehensive analytical model to predict message latency for different data flows traversing across the network. This model considers channel buffers of multiple flits which were not previously studied in NoC context. Also, architectural descriptions of the overall consumed power in the network components are extracted considering message arrival and service rates. The results obtained from simulation experiments confirm that the proposed performance and power models exhibit good accuracy for various network configurations and workloads.

Categories and Subject Descriptors
B.4.4 [Hardware Input/Output and Data Communications]: Performance Analysis and Design Aids; I.6.4 [Simulation and Modeling]: Model validation and Analysis

General Terms
Performance, Design.

Keywords
Network-on-Chip, Performance, Power, Analytical modeling, Queuing system.

1. INTRODUCTION

NoCs are used to connect a number of processing cores on a single chip multiprocessor system or IPs in SoCs [2]-[3],[17]. They have been utilized not only for high performance micro-architectures but also for cost-effective embedded devices such as mobile wireless devices or PDAs [2]. Such embedded applications often demand tight design constraints in terms of cost and performance; thus the silicon budget available for their on-chip network infrastructure should be kept low while performance and power requirements are met [3]. NoC architectures have many design parameters including network topology, routing algorithm, and router architecture; all of which severely affecting the system performance and energy. Therefore, network architecture for such embedded systems should be carefully selected to meet desired requirements [2].

Wormhole routing is the most common switching scheme for NoCs as it minimizes communication latency, requires low buffer space, and is relatively simple for implementation [2], [5]. However, most performance evaluation studies of wormhole-switched NoCs have heavily relied on simulations [1],[9],[12],[18]. The use of simulation experiments makes the task of searching for efficient designs computationally intensive and does not scale well with the size of network [19]. An alternative approach is the utilization of an analytical model for the NoCs [6], [16]. An appropriate analytical model can estimate the desired performance metrics in a fraction of time that simulation would take. Thus, it is justified to derive accurate analytical models for popular NoCs. Unfortunately, a detailed and accurate analytical model of a complex NoC is intractable, and approximations may also lead to inaccurate models.

Two contributions are reported in this paper. First, finite size queuing systems are used to develop an analytical model for message delay prediction through various flows of an application. The routers are assumed to be equipped with multiple finite-sized channel buffers which were not previously studied analytically in NoC context. The exact analysis of the time which takes a message to completely traverse a link is realized through estimation of the forwarding hops affecting the time which a message reserves a physical channel in wormhole switching. The proposed model can predict the network behavior under a wide range of traffic loads. Given any NoC configuration (network topology and routing algorithm) and its communication demands (in terms of packet length, message destination address distribution, and generation rate), the proposed analytical model can estimate the delay experienced through various flows within an acceptable accuracy level. Also, power descriptions for each router component and links are developed considering the effects of traffic load, link bandwidth, and average number of messages communicated. Traffic specifications including arbitrary message generation rate and message length are considered in this model. The proposed models are then used to efficiently explore design space of a router in a given scenario.
The rest of this paper is organized as follows. Section 2 reviews previous studies and highlights our contributions. In section 3, we provide assumptions and introduce the performance model. Section 4 is devoted to the evaluation of consumed power. Comparison of results gathered by the proposed model and those obtained through simulations is realized in section 5. We then utilize these models for an efficient router design example in section 6. Finally, concluding remarks are given in section 7.

2. RELATED WORKS

Scalable NoCs are introduced to provide high bandwidth communication infrastructure for SoCs [2],[3]. Researchers brought the ideas of using interconnection networks in multi-computers and multi-processors into SoCs in order to overcome shortcomings of previous on-chip communication architectures. Several key research issues on NoC design are discussed in [2],[17]. While it is intuitively accepted that NoCs provide scalable communication with small area overhead, most studies rely mainly on simulations to justify such findings [1],[9],[12],[17],[18]. So, network traffic used in such studies is either synthetic [18] or approximating real applications via customization of synthetic traffic generators [1],[9],[12]. For example, several studies present design methodologies for application mapping [8],[13], buffer allocation [7], virtual channel planning [10], and NoC topology synthesis [1] and demonstrate the effectiveness of these methodologies via simulation. Although exploration of the design space in such algorithms is mostly NP-Hard, they generally utilize heuristic [13] or branch-and-bound [8] techniques to prune design space within an acceptable time. However, simulation would inherently take long time to perform and provide a little insight on how the design parameters affect the actual network performance.

Another approach may use analytical modeling of NoCs. In [16], a generalized router model was proposed which allows computing the average number of packets at each buffer in the network as a function of arrival traffic process. The work in [6] presents a wormhole delay model applicable to routers with single flit buffers assuming that the packet size is the main contributor to the overall latency.

Related work about analytical modeling for wormhole switching comes mainly from parallel computing. Many studies target specific topologies or routing strategies [4]-[5],[14],[20]. The work in [9] is not restricted to any specific topology and routing algorithm. They use a sophisticated analysis to predict waiting time of a message in forwarding hops in wormhole switching for an arbitrary network topology.

Energy and power models for NoCs are discussed in several studies. Authors in [22] develop a power model for router components and introduce a performance-power simulator for interconnection networks. The work in [23] predicts the consumed energy per bit from any source to a destination using a system level approach. However, none of these works considers the effect of traffic arrival process of application tasks on various network components in a closed form solution.

In this paper, we present a complete inter-related analytical model for performance and power prediction in NoCs. The model is used to perform an exhaustive analysis of wormhole switching with arbitrary message length distribution and multi-flit channel buffers under various traffic patterns. Furthermore, the model supports arbitrary network topology and deterministic routing. To the best of our knowledge, this is the first study which presents a new power-performance model for NoCs with arbitrary channel buffer size using queuing theory.

3. THE PERFORMANCE MODEL

We consider input buffered routers with arbitrary number of ports that connect it to the adjacent nodes and local IP core through bidirectional (two unidirectional) links. Also, we target wormhole switching flow control under deterministic routing algorithm without cyclic dependency in channel dependency graph [15]. IP cores independently inject messages into the local buffers of routers following a Poisson process. The messages are generated with average rate $X_F$ for a flow $F$ from source $S$ to destination $D$. Message length is denoted by a random variable $m$, extracted from the driven application. Destination of a message is determined synthetically or by the driven application. Also, messages will be ejected from the network immediately when arrived at destination routers.

Routers have $B$ flit input buffers. Also, the width of each channel and flit size are assumed to be $W$ and $K$ bits, respectively. Figure 1 shows router configuration assumed during model development.

![Figure 1. Router configuration in the proposed analytical model.](image)

In wormhole switching, a message is divided into a sequence of flits, traversing a link in a pipeline manner. When the header flit (containing routing information) reaches the head of a buffer, it takes $T_R$ cycles to be routed. Then, it traverses the crossbar switch toward an output port in $T_S$ cycles. It then traverses the channel to the next router in $T_W$ cycles. However, the body flits, which follow the header, only experience switching and wire delays.

The message latency through flow $F$ is composed of the network latency, $\bar{S}_F$, that is the time to cross the network, and the mean waiting time seen by a message in the source node, $\bar{W}_F$:

$$T_F = \bar{S}_F + \bar{W}_F$$

Therefore, we can write the latency through flow $F$ as:

$$T_F = \sum_{i=1}^{m} (FW_i + W) + WS + (m-1)WB$$

where

- $FW_i$: Flit width
- $W$: Number of flits
- $WS$: Switching time
- $WB$: Wire delay

This formula provides an accurate prediction of the latency through a wormhole switching network.
The network latency for a message consists of two parts: one is the delay due to actual message transmission time and the other is due to blocking time in the network. The network latency of a message through flow $F$ with length $d_F$ hops can therefore be written as:

$$
\bar{S}_F = d_F \left( T_r + T_s + \left( \frac{K}{W} \right) T_w \right) + \left( m - 1 \right) \left( T_s + \left( \frac{K}{W} \right) T_w \right) + \sum_{i=1}^{d_F} b_i^F
$$

(2)

where $m$ is the mean message length and $b_i^F$ is the mean blocking time seen by header flit at hop $i$ of flow $F$ ($1 \leq i \leq d_F$).

The multi-flit channel buffers as well as variable message length limit the number of subsequent routers through a flow, where waiting time of the header in their buffers affects the holding time of the output port in the current router by a message. It should be mentioned that we assume one virtual channel for a physical channel. So, if a message reserves an output port, no other message can use that port. If consecutive buffers from the current router to the destination are not large enough to hold the entire message, all the subsequent routers to the destination affect this holding time. So, the effective number of downstream routers for an $m$-flit message at hop $i$ of flow $F$, $\Lambda_i^F(m)$, can be written as:

$$
\Lambda_i^F(m) = \begin{cases} 
\left\lfloor \frac{m}{B} \right\rfloor & \frac{m}{B} < d_F - i \\
\text{otherwise} & 
\end{cases}
$$

(3)

Figure 2 shows these two states for an 8-flit message in its third and fourth hops through a flow with 5 hops length and 3 flits channel buffers.

Owing to variable message length, the effective number of downstream routers, $\Lambda_i^F(m)$, is inherently stochastic. So, the probability that exactly $k$ routers affect the delay at the current node is given by:

$$
\Psi_i^F(k) = P\{\Lambda_i^F(m) = k\}
$$

\begin{align*}
\Psi_i^F(k) &= \begin{cases} 
M \left\{ m < (k+1)B \right\} - M \left\{ m < k B \right\}, & \frac{m}{B} < d_F - i \\
1 - M \left\{ m < k B \right\}, & \text{otherwise}
\end{cases} \\
&= \begin{cases} 
M \left\{ m < (k+1)B \right\} - M \left\{ m < k B \right\}, & \frac{m}{B} < d_F - i \\
1 - M \left\{ m < k B \right\}, & \text{otherwise}
\end{cases}
\end{align*}

(4)

where $M$ is the probability density function of the stochastic variable $m$.

Consider a message at hop $i$ of flow $F$. Blocking delay, $b_i^F$, is the time that takes a message to get header of next hop buffer when it is at the head of current buffer, excluding routing delay, $T_r$. $b_i^F$ is composed of the delay due to contention with other messages to acquire the output port at the current router, $b_i^F$, and the time to reach the head of queue at the next hop, $q_i^F$. Hence:

$$
b_i^F = h_i^F + q_i^F
$$

(5)

Using Laplace Stieltjes Transform (LST)\(^1\) for equation 5 results in:

$$
Q_{i+1}^F(s) = \frac{B_i^F(s)}{H_i^F(s)}
$$

(6)

We consider a fair arbitration. So, contention for an output channel can be modeled by a finite FIFO buffer. Hu and Kleinrock in [12] stated that more accurate results can be achieved if both the contention queue and input buffer at the next hop are modeled as a single queue. Now the question is: What is the size of the equivalent queue, if a router has $P$ ports each with a $B$-flit buffer?

Each slot of a contention queue refers to a message which tries to acquire an output port, while each slot of an input buffer can hold only a flit. So, the buffer queue should be scaled such that it refers

---

\(^1\) Since service distribution of the queues is unknown in design stage, LST is used to estimate its behavior using multiple moment approximation [21].
to the number of the messages it can hold. The variable message size makes the buffer size (in messages) to have stochastic nature. If stochastic variable $m_i$ refer to the size of message $i$ in a sequence of messages, the probability that input queue hold first $Y$ messages, $\Gamma(Y)$, can be written as:

$$\Gamma(Y) = P \left\{ m_1 + \cdots + m_Y \leq B + \frac{m}{2} \right\} = P \left\{ m_1 + \cdots + m_{Y+1} \leq B + \frac{m}{2} \right\} - P \left\{ m_1 + \cdots + m_Y \leq B - \frac{m}{2} \right\}$$

(7)

Note that the queue size is extended by $\bar{m}/2$ flits to capture the effect of space which does not hold an entire message [9]. The message length distribution function must be a descending function, so that $\Gamma(Y)$ is positive. Therefore, blocking time of a message at hop $i$ of flow $F$ is the average of waiting time in equivalent queue with weight $\Gamma(Y)$:

$$B_i^F(s) = \sum_{Y=0}^{\infty} \Gamma(Y) \times W_i^F(Y+p,s)$$

(8)

where $W_i^F(Y+P,s)$ is the LST of the mean waiting time in equivalent queue with size $Y+P$ messages. In equation 8, $P$ is the number of ports which may use the same output as current message flow at router $i$ ($1 \leq i \leq d_x$). Thus, we define contention matrix $C$ in which $c_{ij}$ denotes the number of ports in router $i$ ($1 \leq i \leq n$) which may chose output port $j$ ($1 \leq j \leq P_{MAX}$) as its next hop (see figure 1). $P_{MAX}$ refers to the maximum port size of all routers. So, we have:

$$C = \begin{bmatrix} c_{0,0} & \cdots & c_{0,P_{MAX}} \\ \vdots & \ddots & \vdots \\ c_{n,0} & \cdots & c_{n,P_{MAX}} \end{bmatrix}$$

(9)

To determine the mean waiting time, $w_i^F$, the equivalent queue is treated as an M/G/1/k queue with an LST of mean waiting time probability density function (PDF) [21]:

$$W_i^F(s) = \frac{\Pi_j \left\{ 1 - \left( \frac{\lambda_i^F \cdot S_i^F(s)}{\lambda_i^F - s} \right)^k \right\}}{s - \lambda_i^F + \lambda_i^F \cdot S_i^F(s)}$$

(10)

$$\lambda_i^F = \frac{\lambda_i^F}{1 - P_B}$$

(11)

where $\lambda_i^F$ is the normalized traffic arrival rate on the channel located at hop $i$ of flow $F$, $S_i^F$ is its service time, $\Pi_j$ is the probability that exactly $j$ messages exist in the equivalent queue, and $P_B$ is the probability that queue is full. The rate of messages arriving at a channel consists of the rate of flows in the running application, which uses this channel and is given by:

$$\lambda_i^F = \sum_{\forall F_i \in \text{Intermediate Channels}} X_i^F$$

(12)

The full buffer probability is estimated using steady state probability when analyzing M/G/1/k queuing system as [21]:

$$P_B = 1 - \frac{1}{\sum_{\forall F_i \in \text{Intermediate Channels}} \lambda_i^F \times S_i^F}$$

(13)

We start computing service time at the ejection channel, which is defined by the user or running application, and return back to the source channel hop by hop. Consider router $i$ ($1 \leq i \leq d_x$) of flow $F$. In wormhole switching, the service time consists of the contention time for the output channel at the current router, $h_i^F$, and the time during which such output channel is reserved by a message, $u_i^F$. Hence:

$$S_i^F = h_i^F + u_i^F$$

(14)

$$S_i^F(s) = H_i^F(s) \times U_i^F(s)$$

(15)

Finally, service time of the buffer connected to link $L$ is the average of service time of all flows using this channel with appropriate weight as:

$$S_L = \frac{\sum_{\forall F_i \in \text{Intermediate Channels}} X_i^F \cdot S_i^F}{\sum_{\forall F_i \in \text{Intermediate Channels}} X_i^F}$$

(16)

To calculate $u_i^F$, two random variables $x_i^F$ and $y_i^F$ are defined. $x_i^F$ is the time which take the header flit to reach the position where consecutive buffers (from current router to that position) can hold the entire message or the router at hop $\lambda_i^F(m)$. However, if the current router is destination, the header flit only meets input buffer delay with no contention for the ejection channel. $x_i^F$ can therefore be written as [9]:

$$x_i^F = \begin{cases} \lambda_i^F(m) - 1 \left( T_a + T_s + \frac{K}{W} T_w \right) + q_i^F + h_i^F + \sum_{j=i+1}^{\infty} b_j^F, & \lambda_i^F(m) \geq 1 \\ q_i^F, & \text{Otherwise} \end{cases}$$

(17)

The LST of random variable $x_i^F$ can be written as:

$$X_i^F(s) = \frac{\lambda_i^F(m) - 1 \left( T_a + T_s + \frac{K}{W} T_w \right)}{s} \times \sum_{j=1}^{\infty} \Psi_i^F(j) Q_i^F(s) H_i^F(s) \prod_{k=1}^{j-1} B_k^F(s)$$

(18)
Random variable $y_i^F$ consists of variable $x_i^F$ and the time in which body flits completely cross a link. So, $y_i^F$ for an $m$-flit message can be written as [9]:

$$y_i^F = x_i^F + (\overline{m} - 1)(T_s + \left[ \frac{K}{W} \right] T_w)$$  \hspace{1cm} (19)

$$Y_i^F(s) = X_i^F(s) \times \frac{(\overline{m} - 1)(T_s + \left[ \frac{K}{W} \right] T_w)}{s}$$  \hspace{1cm} (20)

It is obvious that $y_i^F$ is the upper bound of the time which takes an $m$-flit message to traverse the hop $i$ of flow $F$. However, its lower bound depends on the traffic of routers utilized by flow $F$. In high traffic loads, message contention in the downstream routers dominates body flits transmission delay. However, transmission time of the body flits may exceed blocking time in the subsequent channels. So, we can write:

$$\min \left\{ x_i^F, (\overline{m} - 1)(T_s + \left[ \frac{K}{W} \right] T_w) \right\} \leq u_i^F \leq y_i^F$$  \hspace{1cm} (21)

Authors in [20] propose an approximation for $u_i^F$ as:

$$U_i^F(s) = \begin{cases} \frac{(Y_i^F - X_i^F)Y_i^F(s) + 2X_i^F}{Y_i^F + X_i^F}, & r > X_i^F \\ \frac{(Y_i^F - X_i^F)Y_i^F(s) + 2X_i^F}{Y_i^F + X_i^F}, & r \leq X_i^F \end{cases}$$  \hspace{1cm} (22)

$$r = (m - 1)(T_s + \left[ \frac{K}{W} \right] T_w)$$  \hspace{1cm} (23)

where $\overline{X_i^F}$ and $\overline{Y_i^F}$ are the first moment of the associated variables.

Finally, the contention delay $h_i^F$ in equation 5 is approximated using M/G/1/k queue with size $P$ and service time $u_i^F$ [9].

Modeling the local queue in source node as an M/G/1 queue with mean arrival rate $\lambda_s$, and service time $S_0$ with approximated variance $(S_0 - T_R + \overline{m}(T_S + T_W))^2$ yields the mean waiting time as [9]:

$$\overline{W_P} = \frac{\lambda_s S_0^2 + \left( 1 - \frac{T_R + \overline{m} T_S + \left[ \frac{K}{W} \right] T_w \right)^2}{2(1 - \lambda_s S_0)}$$  \hspace{1cm} (24)

determine, different variables of the model are computed using iteration [21]. Also, substitution equations 11 and 13 lead to a non-linear equation. So, bisection, a standard iteration method, can be efficiently applied to solve $P_w$.

To reduce model complexity, we assume several approximations. The LST of different variables are simply modeled by two first moments approximations [9]. If $s$ (Laplace variable) sets to zero in the $r^{th}$ derivative of a variable, its $r^{th}$ moment is achieved. Also, message length is exponentially distributed. This yields straight calculations for equations 4 and 7.

### 4. THE POWER MODEL

In this section, a high level approach for modeling power consumption of an NoC router and its links is proposed. The router configuration is depicted in figure 1. A router has $P$-1 ports of $B$-flit buffers and a plenty of buffer slots at local queue connected to IP core. The buffers are implemented as convoluted queues, using SRAM cells. If a buffer slot holds no data, its clock is gated for power saving. A routing logic decides on the output port for a header flit when it reaches head of a buffer. An arbiter fairly (or with a priority strategy) controls the flit flow crossing the crossbar switch to an output port. A crossbar switch uses arbitration signals to provide synchronous connections between various input/output ports.

The main power consumption source in buffers is due to read and write operations. We assume that $P_{\text{bi}}^w$ and $P_{\text{bi}}^r$ denote the power consumption for one bit write and read operation, respectively. Therefore, overall write (read) operation power is modeled as write (read) rate multiplied by $P_{\text{bi}}^w$ ($P_{\text{bi}}^r$). Messages write in a buffer connected to link $L$ with rate $\lambda'_L$, while each message consists of $m$ flits, in average. Also, each flit has $K$ bits width. Similarly, the rate of read operation is calculated. The dynamic power of a buffer can be therefore approximated as:

$$P_{\text{Dynamic}} = \overline{m} \times K \times \left( \lambda'_L \times P_{\text{bi}}^W + \frac{P_{\text{bi}}^R}{S_L} + Q_L \times P_{\text{CLK}} \right)$$  \hspace{1cm} (25)

where $P_{\text{CLK}}$ refers to the mean power consumed when one bit SRAM receive a clock switch. As previously stated, clock signal of SRAM cells that have no data is gated. Therefore, $Q_L$ in equation 25 that is the mean number of messages at the current buffer and takes the effect of SRAM cells which receive clock. In the steady state, the average number of messages in input queue $L$, $Q_L$, is given by [21]:

$$Q_L = \sum_{l=0}^{\infty} \frac{\Gamma(Y)}{\Pi_{n \neq 0} \lambda'_n S_L} + Y(1 - \frac{1}{\frac{\sum_{l=0}^{\infty} \lambda'_l S_L}{\Pi_{n \neq 0} \lambda'_n S_L}})$$  \hspace{1cm} (26)

Leakage power, which is the major static power component, must be considered, especially in deep submicron technologies. As all SRAM cells of a buffer (with/without clock gate) consume static power, we can write:

$$P_{\text{Leakage}} = B \times \overline{m} \times P_{\text{bi}}^L$$  \hspace{1cm} (27)
where $P_{\text{bit}}^t$ is the mean leakage power per bit.

The power consumption in routing logic, arbiter, and crossbar switch depend on message arrival rate over all ports of a router. To shorten the calculations, we integrate the first two elements in a single equation. Also, our simulations show that the consumed power in a crossbar switch has quadratic dependency on message arrival rate of different ports. Using calculations similar to that for buffer component, we have:

$$P_{\text{bit}}^{RL-A} = P_{\text{bit}}^{XR} \sum_{L=1}^{D} \alpha_L \left( W \times m \sum_{L=1}^{D} \alpha_L \right)$$

(28)

$$P_{\text{bit}}^{XR} = P_{\text{bit}}^{XR} \left( \frac{W \times m}{L} \right)$$

(29)

where $P_{\text{bit}}^{XR}$ in equation 28 refers to the dynamic power consumed when a header is routed and arbitrated. Also, $P_{\text{bit}}^{XR}$ in equation 29 counts for the dynamic power due to crossbar traversal for a bit. Note that routing and arbitration rate follows message (or header) arrival rate, since these functions do not perform on the body flits.

Let us now describe how to calculate the quantities $P_{\text{bit}}^{W}$, $P_{\text{bit}}^{R}$, $P_{\text{bit}}^{CL}$, $P_{\text{bit}}^{L}$, $P_{\text{bit}}^{RL-A}$, and $P_{\text{bit}}^{XR}$. For a given router configuration and CMOS technology, limited simulations are performed to extract fine granularity values for power in router components, including input buffers, routing unit, arbiter, and crossbar. Each set of simulation experiments takes the effect of various parameters in equations 25 to 27. These parameters are average message size, flit size, link bandwidth, arrival traffic rate, and service rate. In order to derive the quantities $P_{\text{bit}}^{W}$, $P_{\text{bit}}^{R}$, $P_{\text{bit}}^{CL}$, $P_{\text{bit}}^{L}$, $P_{\text{bit}}^{RL-A}$, and $P_{\text{bit}}^{XR}$, simulation results for power consumption of buffers as well as equations 25 to 27 are fed to non-linear MATLAB regression functions. For other quantities in equations 28 and 29, similar processes are followed.

As technology scales, the fraction of power consumption in links becomes significant. Also, crosstalk capacitances can not be ignored in deep submicron technologies, since the gap between adjacent wires becomes smaller. So, if $f_{\text{clk}}$ and $V_{DD}$ denote router frequency and supply voltage, consumed power in a $D$ millimeter link of $W$ bits width can be written as:

$$P_{\text{Cross}} = \frac{1}{2} \times \bar{X}_C \times m \times \left( \alpha_L WC_L^0 + \alpha_C \left(W - 1\right)C_L^0 \right) \times D \times f_{\text{clk}} \times VDD^2$$

(30)

where $\alpha_L$ and $\alpha_C$ denote the probability that different bit values cross over a link and adjacent links, respectively. Also, $C_L^0$ (or $C_L^0$) in this equation is wire (crosstalk) capacitance per millimeter. Note that data flits are transmitted over link $L$ with rate $\lambda_L \times m$.

At last, the total power consumption of a router can be written as:

$$P_{\text{Router}} = P_{\text{Cross}} + P_{\text{Buffer}}^{\text{Dynamic}} + P_{\text{Buffer}}^t$$

(31)

5. VALIDATION RESULTS

This section reports on the accuracy of the proposed models. The above models have been validated through a discrete event NoC simulator that performs a time-step simulation of the network operations at the flit level. Each simulation experiment was run until the network reached its steady state, that is, until the simulated network cycles do not change the collected statistics appreciably. To this end, up to 50,000 messages are grouped into 10 batches. First batch statistics are inhibited due to their distortion effects. To analyze simulation outputs, batch-mean method [12] is used with standard deviation less than 2% of the mean value. Both simulator and analytical model are implemented in C++ and tested on a Pentium 4 system with 2G RAM running Windows OS. Also, we integrated the Orion library [22] in the simulator for power estimation in router components.

In zero load, routing takes 3 cycles, crossbar switching or transmission over a link take 1 cycle. Message length follows exponential distribution, since its probability density function is a descending function which guarantees a positive value for $\Gamma(Y)$ in equation 7.

Extensive validation experiments were performed for several combinations of network topology, routing algorithm, buffer size, mean message length, and flit size. However, for the sake of specific illustration and brevity, we provide results for the following cases only:

![Figure 3. Mean message latency and total power predicted by the models and simulation in the 4×4 Mesh; $m = 9, 16$, and 32 flits, $B = 4$ and 6 flits under uniform traffic. $T_2 = 3$ cycles, $T_3 = 1$ cycle, $T_w = 1$ cycle.](image-url)
• 4×4 Mesh network with XY routing for exponential message length with mean 9, 16, and 32 flits. The buffers have 4 and 6 flits depth. Messages are uniformly destined to other IP cores.

• 4-dimentional Hypercube with dimension order routing. Mean message length is 15 flits and buffers have 2 flits length. A portion of 0.16 or 0.24 messages are destined to the hotspot node (node <0000> here) and others use uniform traffic.

Through the provided experiments, wires have 8 bits width. In the architectural level, we use 110 nm CMOS technology, 100 MHz router frequency, and 1 millimeter inter-router links.

Figure 3 provides detailed results for the mean message latency and overall power consumption predicted by the analytical model plotted against those obtained by the simulator for the first scenario. The accuracy of the model against non-uniform traffic in the second scenario is validated in figure 4. In all figures, the horizontal axis represents the generation traffic rate (λ) while the vertical axis shows the mean message latency or total power consumption for crossing from the source to destination router.

The figures reveal that prediction of the mean message latency and total power is done with a high degree of accuracy when the network has not reached the saturation point. However, there are some discrepancies in the results provided by the model and simulation when the network is under medium or heavy traffic and approaches the saturation point. This is due to the approximations that have been made to ease the model development. For instance, two moments approximation for LST is a rough assessment. Nonetheless, given the most studies focus on network evaluation in non-saturation region, we can conclude that the proposed model can be a practical evaluation tool to help SoC designers to gain insight into the dynamic behavior of the infrastructure and to utilize it in an NoC optimization process loop.

Figure 5. Communication task graph for MMS benchmark mapping on a 100 MHz NoC with Mesh topology [7]. Single or cluster of tasks, which are assigned to an IP core connected to node (x,y), are grouped into a box. Communication volumes shown on the edges have a unit of 10 KB/sec.
As a case study, we used the communication task graph for a generic MultiMedia System (MMS) [7] and manually mapped the tasks on 16 IP cores connected via a mesh network (see figure 5). The MMS is an integration of H263 video decoder/encoder and mp3 audio decoder/encoder. It is partitioned into 40 tasks with communication volumes shown on edges (in terms of 10 KB/sec), for a video clip simulation. Each message carries an 8x8 block and each pixel value is also represented by 16 bits. Therefore, each message has 8 flits length for 128-bit flits. Also, all routers have 3-flit buffers. The delay experience at each router and power consumption for each router component are performance metrics at the finest granularity level. Indeed, it can be used for router customization for use-case NoCs. Figure 6 shows these metrics obtained from analytical models and those extracted from simulations for MMS application. We have opted to show the results for the power consumption in west channel buffers and routing, arbitration logic, and crossbar only for the sake of brevity (see figure 6). The average prediction error is 3.77% for mean latency and 6.2% for power consumption.

6. ROUTER DESIGN EXAMPLE

Buffer depth, \( B \), and links bandwidth, \( W \), are two major concerns in router design. To analyze the effect of these parameters consider MMS application mapping on a 4x4 Mesh with XY routing (see figure 5). Routing, switching, and wire delay are all set to 1 cycle. Messages have 16 flits length each with 64 bits width. Note that fixed length of messages simplifies model’s approximations. We consider overall number of flits in routers as the utilization metric (the ratio of the number of message flits in all routers to the buffer capacity). To this end, values of equation 27 and message length are multiplied for this estimation. Figure 7 shows utilization rate as a function of \( B \) and \( W \). In the presence of fixed width, increasing buffer depth more than a threshold value is not effective, since the routers are fast (2-stage pipeline, each
with one cycle). For instance, buffers with more than 6 flits length, for \( W=64 \), are not effectively utilized. Also, such a fast router is suitable for low buffer size (4 slots or less), if bandwidth increases.

7. CONCLUSIONS

In this paper, a generic power-performance analytical model for networks-on-chip is presented. The model considers the effect of different components on the message latency and power consumption. The proposed approach exhibits various statistics from message latency per a component, per a flow or in average. Also, different granularity levels exist for the power model. Generic behavior of this model was shown through different validation scenarios. Furthermore, we demonstrated the model utilization in the design context through a router customization example.

Future work in this line can consider the use of virtual channels for each physical channel. Virtual channels are useful to improve system throughput and to develop deadlock free routing protocols.

8. REFERENCES


