New Latency Model for Dynamic Frequency Scaling on Network-on-Chip
Sheng-Nan Li and Wen-Ming Pan

Abstract—Modulating both the clock frequency and supply voltage of the network-on-chip (NoC) during runtime can reduce the power consumption and heat flux, but will lead to the increase of the latency of NoC. It is necessary to find a tradeoff between power consumption and communication latency. So we propose an analytical latency model which can show us the relationship of them. The proposed model to analyze latency is based on the M/G/1 queuing model, which is suitable for dynamic frequency scaling. The experiment results show that the accuracy of this model is more than 90%.

Index Terms—Dynamic programming network, latency model, network-on-chip, power budgeting, regression.

1. Introduction

As the structure size down to 22 nm and smaller[1], there are two big challenges that many network-on-chip (NoC) designers have to face: First, the proportion of time which is spent on the communications of different intellectual property (IP) cores is increasing compared with the time that is spent on calculating in the IPs, so the delay of NoC acts as the performance bottleneck in many NoC designs[2]. Second, as the integration density of a chip increases in an amazing speed, the power consumption and the power density of NoC are also increase rapidly. Eventually, the point will be reached where heat constraints forbid adding more parallelism structures, which becomes a new limiting factor to system performance[3]. The many-core architecture also faces steep design challenges with respect to power consumption[4]. Now more and more people want to control the power budget during the runtime dynamically to make the NoC more power saving, and most designs allow the modulation of both clock frequency and supply voltage during runtime by a feature called dynamic voltage and frequency scaling (DVFS)[5]. But two problems became the limitation which seriously affects the development of performance of NoC. In general, if latency is reduced, the power consumption will increase; if power is reduced, the latency of NoC will increase. So it is very important to build a latency model which can show us the relation between the latency and the frequency of NoC and the average latency of NoC in DVFS procedure. But the present analytical model only can be used when the frequency of routers is fixed.

To solve this problem, in this work, we present a latency model which can not only calculate the latency in a normal way, but also analyze the latency when the frequency of routers is changed dynamically. So it will suit for the works which care about both the latency and power and want to make a tradeoff between them or to find the best budget when there is a constraint for power.

The rest of the paper is arranged as follows: In Section 2, we will discuss the related work about latency models and power management by means of DVFS. The new latency model is proposed in Section 3. In Section 4, evaluations on different router frequency and different workloads are given. And the conclusions are drawn in Section 5.

2. Related Work

Two latency models using the queuing theorem were presented in [5] and [6]. The model presented in [5], which is based on an M/G/1 queuing model, is used to develop a thorough performance analysis for wormhole routing with arbitrary size messages and finite buffers under application-specific traffic patterns. The model proposed in [6] is based on a G/G/1 queuing model. It estimates performance metrics, such as the average latency and router blocking time, and can be conveniently used for optimization purposes to find appropriate design parameters and obtain quick performance estimates. Reference [7] analyzed how multiple traffic streams originating at the source could be combined into a single traffic stream without any adverse impact on the statistical characteristic
of the destination traffic. The model was derived from the observations of the statistical behaviors of received streams at destinations in a single-source multiple-destinations scenario. Reference [8] presented an analytic formalism that enables powerful modeling of discrete event (DE) systems. There are some other works introduce latency models using the queuing theorem[9],[10].

3. NoC Average Latency

The objective of the NoC power budgeting is to optimize the average latency under a power envelop, which corresponds to minimizing the overall system running time. Lightweight latency models can be derived from the queue model, which will be described in this section. Our goal is to find a quantitative relationship between the network latency and the router frequencies.

In this section, we will first discuss how to model a single router by using the queuing theorem. A router is considered as a set of first-in first-out (FIFO) buffers connected by a crossbar switch. Fig. 1 shows a model of a router for the mesh network, Ch0 is used to connect local IPs, Ch1 to Ch4 are used to connect to the routers nearby. Secondly, we will discuss how to calculate the average latency.

In this work, we assume that a router has \( p \) channels with buffers and the network uses wormhole control and is under deterministic routing algorithms. The model we proposed has been constructed on the M/G/1 queue.

![Router model as a collection of FIFO buffers with five input channels.](image)

3.1 Waiting Time in Each Router

First, we discuss the single queue, the waiting time of the \( i \)th arrival packet which includes the waiting time in the queue and the service time is

\[
W_i = \sum_{k=1}^{N_i} X_k + R_i
\]

where \( X_k \) is the service time of the \( k \)th packet, including the time to traverse the router and link, and \( X_1 \) can be considered to be proportional to the router frequency \( f_i \). \( R_i \) is the residual service time, which is the time needed to finish the serve of a packet that is being served now, from the aspect of the \( i \)th incoming packet.

\[
\]  

where \( E[W_i] \) means the expected waiting time of the \( i \)th packet, \( E[X_i] \) means the expected service time of the \( k \)th packet, \( N_i \) is the length of the queue when the \( i \)th packet is coming, \( N \) is the average length of queue, \( \mu \) is the serve rate of the server, and \( R \) is the residual service time.

Here we have used the “Poisson arrivals see time averages” (PASTA) property and the independent service time property, from Little’s formula:

\[
W = R + \lambda W/\mu
\]

where \( W \) is the average wait time and \( \lambda \) is the arrival rate of channel.

So we get

\[
W = R/(1 - \rho)
\]

where \( \rho = \lambda/\mu \) and

\[
R = \frac{\lambda E[X^2]}{2} + \frac{(1 - \rho)E[V^2]}{2E[V]}
\]

where \( V \) is the vacation time and \( X \) is the service. The serve of a packet can begin only at the start of a cycle. If the queue is empty at the start of a cycle, the server will not work for this duration of the cycle (vacation). So, the average wait time \( W \) is

\[
W = \frac{\lambda E[X^2]}{2(1 - \lambda/\mu)} + \frac{E[V^2]}{2E[V]}. \quad (6)
\]

Following the description and the assumption above, we can calculate the \( E[V] \) and \( E[V^2] \) easily.

We model the router as a set of FIFO buffers connected with a crossbar (see Fig. 1), if there are more than one input channel want to transmit packets to the same output channel at the same time, an arbitration will take place. So \( W_i \) is composed of following four components:

1) The service time that the packets have already waited in the same buffer;
2) The time that packets have waited in the other buffers of the same router and been served before the incoming packet;
3) The residual service time, which is the time needed to finish the serve of a packet that is being served now, from the aspect of the incoming packet (\( R \));
4) The service time of this packet.

So the average waiting time of channel \( i \) is

\[
\]
where \( \lambda_i \) is the arrival rate of channel \( i \), and \( c_{i,j} \) is the confliction rate between input channel \( i \) and input channel \( j \). If \( i=j \), \( c_{i,j} = 0 \); if \( i \neq j \), then

\[
c_{i,j} = \sum_{0 \leq k \leq p-1} \left( \frac{\lambda_j}{\lambda_k} \right) \left( \frac{\lambda_{j,k}}{\lambda_j} \right)
\]

(8)

where \( \lambda_{j,k} \) is the arrival rate at input channel \( i \) which is routed toward output channel \( k \). And \( \lambda_{i,j} \) representing the packet rate generated by IP\(_i\) and routed to IP\(_j\) can be expressed as

\[
\lambda_{i,j} = \sum_{s \in \mathbb{S}} \left( \lambda^{i,s} \times R \left( S \rightarrow D, IC_i \rightarrow OC_j \right) \right)
\]

(9)

where \( IC_i \) is the \( i \)th input channel, \( OC_j \) is the \( j \)th output channel, \( s \) is the source IP, and \( D \) is destination IP. And \( R \left( S \rightarrow D, IC_i \rightarrow OC_j \right) \) equals 1 if a packet from router \( S \) to router \( D \) passes from \( IC_i \) to \( OC_j \), otherwise, it equals 0.

### 3.2 Average Delay of Network

In Fig. 2, a flow is generated in IP\(_s\) and its destination is IP\(_d\). The packet traverses \( R_s \), \( R_m \), and \( R_d \) before reaches IP\(_d\). The latency of the header flit of this packet is the sum of the time waiting in the input Ch\(0\) of \( R_s \), the time waiting in the input Ch\(1\) of \( R_m \), and the time waiting in the Ch\(4\) of \( R_d \), which is,

\[
L^{s \rightarrow d}_h = W^0_s + W^1_m + W^4_d
\]

(10)

and the latency of this packet is,

\[
L^{s \rightarrow d} = L^{s \rightarrow d}_h + \left[ Z/W \right]
\]

(11)

where \( Z \) is the size of this packet and \( W \) is the bandwidth of the network, and \( \left[ Z/W \right] \) is the body flits traverse time. So, the latency of the network is

\[
L = \sum_s \sum_d P^{s \rightarrow d} L^{s \rightarrow d} = \sum_s \sum_d P^{s \rightarrow d} \left( \sum_{i \in \Omega} W_i + \left[ Z/W \right] \right)
\]

\[
= \sum_s \sum_d P^{s \rightarrow d} \times \left\{ \left( \lambda_i + \sum_{0 \leq j \leq p-1} c_{i,j} \lambda_j \right)E \left[ X_j^2 \right] + \frac{E \left[ V_j^2 \right]}{2E[V]} \right\} + \left[ Z/W \right]
\]

(12)

where \( P^{s \rightarrow d} \) is the probability of a packet is generated in IP\(_s\) and is delivered to IP\(_d\), \( Y_k \) is the service time of packet served by router \( k \), \( \Omega \) is the set of input channels by which the flow is passing, and \( \Phi \) is the set of routers where the flow will passing.

![Fig. 2. Two-hops flow from IP\(_s\) (source) to IP\(_d\) (destination).](image)

### 3.3 Latency Model with Dynamic Frequency Scaling

Assume that each router can operate on any of the \( m \) frequency levels, the router frequency \( f_i \in \{ \tau_1, \tau_2, \ldots, \tau_m \} \), where \( \tau_m \) stands for the frequency of the full speed and \( \tau_1 = \tau_m/m, \tau_2 = 2\tau_m/m, \ldots \). So if the router \( n \) works at the frequency \( \tau_i \), the current service time of the router is \( X^+_n = mX/\tau_i \) and \( E \left[ (V^+_n)^2 \right]/2E[V^2] = mE[V^2]/2E[V] \); if the router \( n \) works at \( \tau_2 \), the current service time of the router is \( X^+_n = mX/2 \) and \( E \left[ (V^+_n)^2 \right]/2E[V^2] = mE[V^2]/4E[V] \) and so on. As routers are working at different frequencies, and the bandwidth of the wire which connects them is also different, so the time to transmit the packet \( \left[ Z/W \right] \) is determined by the router(s) who has/have the least bandwidth \( W_j \), so the latency \( L \) is

\[
L = \sum_s \sum_d P^{s \rightarrow d} \left\{ \sum_{i \in \Omega} \left( \lambda_i + \sum_{0 \leq j \leq p-1} c_{i,j} \lambda_j \right)E \left[ (mX/l_k)^2 \right] + \frac{E \left[ V_j^2 \right]}{2E[V]} \right\} + \left[ Z/W \right]
\]

(13)

where \( l_k \) is the frequency level of router \( k \).

### 4. Experiment Results

In this Section, we will verify the accuracy of the proposed model. The model has been validated on a simulator where the injected packets of each router was generated by a generator written in C++, the injection rate of each router is independent distributed, and the routing algorithm behavior works at the flit level. Throughout the experiments, we assume that each input buffer slot can hold a 64-bit flit. If there is no contention and the router is working at the full speed, the service time is 5 cycles for
the header flit and the transition time for the body flit is 1 cycle. In these experiments, the vacation time follows the uniform distribution, so

\[ E[V] = \int_{0}^{1} x \, dx = \frac{1}{2} \]

and

\[ E[V^2] = \int_{0}^{1} x^2 \, dx = \frac{1}{3} \]

There are 4 batches and each batch includes 100000 up to 1000000 packets depending on the traffic injection rate, the network size, and the frequency of routers. From these batches, we can see the accuracy of the latency model. We consider a 6×6 mesh and an 8×8 mesh in the experiment, respectively, and the frequency of some routers has been changed. Fig. 3 and Fig. 4 are the results for the 6×6 mesh. Fig. 3 shows the result for the mesh where all the routers are working at the full speed. Fig. 4 shows the result for the mesh where the frequencies of some routers have been changed: 6 routers are working at level 9, 6 routers are working at level 8, and 6 routers are working at level 7, and the rest routers are working at level 10. We can see from Fig. 3 and Fig. 4 that in the 6×6 mesh the analytical model is very accurate. The experiment results show that the accuracy of this model is more than 90% when the average injection rate is under 0.045 packet per cycle per router, and is not precisely enough when the injection rate is beyond 0.05 packet per cycle per router.

Fig. 5 and Fig. 6 are the results for the 8×8 mesh. For the mesh to Fig. 6, 6 routers are working at level 9, 6 routers are working at level 9, 6 routers are working at level 7, and the rest routers are working at level 10. It can be seen from these figures that in the 8×8 mesh the analytical model is very accurate. The experiment results show that the accuracy of this model is more than 90% when the average injection rate is under 0.027 per router, and is not precisely enough when the injection rate is beyond 0.03 per router.

5. Conclusions

In this paper, we propose a latency analytical model that can show us the relationship between the NoC average latency and frequency of routers, which is based on the M/G/1 queueing model. The experiment results
demonstrate the accuracy and effectiveness of the proposed model. The accuracy more than 90% can be achieved when the injection rate is less than 0.045 (6×6 mesh) or 0.027 (8×8 mesh). By using this model, we can make a tradeoff between the power consumption and the latency of NoC. It can help us to find the least latency and optimized parameters of NoC with a constraint on power, and vice versa. And the higher requests on power consumption efficiency and latency will show the utility value and importance of this model in NoC design. In the future, we will do some research on power management by using this model.

References


Sheng-Nan Li was born in Heilongjiang Province, China in 1990. He received the B.S. degree in computer science from the Mudanjiao Normal University of China in 2013. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, Harbin Institute of Technology (HIT). His research interests include fault tolerant computing and parallel computing.

Wen-Ming Pan was born in Guangdong, China in 1982. He received the B.S. and M.S. degrees in electronic engineering from Jinan University, Guangzhou in 2004 and 2007, respectively. He works as a research assistant with Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences. He has been engaged in the FPGA design research for years. His research interests include networks-on-chip and parallel computing.