Establishing Performance Evaluation Model with Queuing Theory in NoC

Tao He*[1], Fucai Chen[1], Tao Qin[2]
[1]National Digital Switching System Engineering Technological Research Center
Zhengzhou 450002, Henan, China
[2]National Computer Network Emergency Response Technical Team Coordination Center
Beijing 100029, China
*Corresponding author, e-mail: hetaotao1980@gmail.com

Abstract

The transmission delay is an important index of the system performance of NoC (Network on Chip). Although the method of simulation-based can get accurate transmission delay, the simulation requires time consuming and a large number of test vectors. Especially in the study of some algorithms, such as mapping algorithms, buffer allocation algorithm, the simulation method is not applicable. To solve the above problems, the paper uses queuing theory to study the establishment of a NoC delay model. The model considers the limited buffer of routing unit and virtual channel technology for network transmission delay and proposes solving algorithm of the delay model based on reverse deduction method. The simulation shows that this model used to analyze the application-specific data transmission delay has smaller average error (reduced by 8%) and higher evaluation efficiency (increased by more than 30 times), which provide designers an efficient method of performance evaluation.

Keywords: *Network on Chip, Delay Model, Queuing Theory*

1. Introduction

With the development of deep sub-micron technology, more and more processor cores are integrated on a chip and conventional bus architecture cannot satisfy the communication requirements of chip gradually. To solve this problem, the Network on Chip was proposed [1, 2, 3, 4]. In NoC, the entire chip is divided into a plurality of interconnected modules which are connected together in network constituted by routing nodes. Communication is no longer in bus, but the data encapsulated in the packets transmits in the network. Performance parameters of NoC including delay, jitter, etc. The packet delay is the time between being injected into the network and received by the destination node. The changes of network status will make the transmission delay of each packet different, which cause the data transmission jitter. With real-time services emerging, not only real-time but also the smaller jitter is required for audio and video services to playback smoothly. In a word, modeling for network delay plays a significant role in the performance evaluation system.

Currently, the methods of NoC performance evaluation mainly are divided into two types, simulation-based and non-simulation-based method. Researchers use the macro network performance analysis tools to evaluate the performance of NoC. The tools have variable protocol, so the network parameters, links and topology can freely change. But it fit with the specific hardware lower in On-chip environment. Some Researchers use the emulator based FPGA to integrate more function blocks on FPGA and VHDL to descript routing unit on chip. The paper [5] evaluates the latency, throughput and leakage power of NoC based on the RTL model with VHDL language. The paper focuses on the physical performance of the network but don't involve verification or evaluation for the application level. The paper [6] proposes a performance modeling and simulation approach to explore the application-platform design space at transaction level. Several examples of application are provided but they are not related to the topology or size of the network. The evaluation tools above are realized at different levels of abstraction. Some use RTL model for cycle accurate simulation and some use transaction-level model for high simulation speed or simple programming model. Therefore it is highly unlikely that they can verify applicable NoC designs and have adaptability in physical structures

Received February 24, 2013; Revised June 30, 2013; Accepted July 14, 2013

as well. To analog performance analysis, some research organizations use high-level language and object-oriented approach to modularize network components on chip, e.g. gpNoCSim [7]. The gpNoCSim is a JAVA realization of NoC universal simulator, which mainly includes the modules of controller, network manager, network, switch, router, node, node traffic and etc. In [8], it presents a comparatively complete NoC modeling and simulation scheme and its development language is SystemC which is one of system-level design languages. Therefore the modeling and simulation are on the abstraction of high-level and can hardly reflect specific hardware characteristics.

In non-simulation-based way, queuing theory has been widely used in network performance analysis. In [9], delay model assumes that the service time of each switching node obeys exponential distribution, which does not in the practical application of the system. This result leads to the analytical error of the model in the heavy traffic load. In [10], the researchers establish the model of transmission delay in NoC, and discuss how to allocate virtual channels. In [11], the routing unit model based wormhole switching technology is proposed and delay model is applied to the mapping algorithm. But the delay models proposed in [10, 11] do not consider the limited cache of routing unit. In [12, 13], the authors get the delay model based on queuing theory in universal routing structure. [14] used queuing theory, proposes a method similar to the shared bus waiting time modeling to estimate NoC waiting time, but the computational complexity is high. In [15], it indirectly optimizes latency and energy consumption by optimizing distribution of link load, but determining the optimization coefficients of the objective function shows the lack of theoretical property. In [16], concepts of average edge delay and critical path are proposed, without sufficient consideration for the delay factors. In [17], author uses M/G/1/N queuing model to analyze the effects that finite buffer has on the network performance, but this model does not support NoC using virtual channels.

Although the method of simulation-based can get accurate transmission delay, the simulation requires time consuming and a large number of test vectors. Especially in the study of some algorithms, such as mapping algorithms, buffer allocation algorithm, the simulation method is not applicable. To solve the problems above, the paper uses queuing theory to study the establishment and algorithms of the network-on-chip delay model and then proofs the model's validity by comparing the prediction results with the simulation results.

2. NoC Delay Analysis Using Queuing Theory

Data transmission delay of NoC is widely used to evaluate the performance of network communications architecture. The mathematical model of delay provides designers with a rapid way of performance evaluation, which will guide the design of the communication structure.

2.1. Related Queuing Theory

Queuing theory is the discipline that studies the theory and application of queuing phenomenon. The queuing system consists of some important factors, e.g. input distribution, arrival distribution, queuing method, service distribution. Typical distributions include Poisson distribution, erlang distribution, exponential distribution and etc. The Poisson distribution is a most important type in the queuing system assumptions. Without this assumption, most of the mathematical analyses of queuing are impossible.

2.2. M/G/1-based Delay Model

In this paper, NoC delay analysis model based on the following assumptions and these assumptions are commonly used. In the network, sending the data packet of each node is independent of each other, and it is a Poisson process. Network uses XY routing algorithm. The length of the packet is M flits and the input buffer width of routing unit is the bit width of a flit. Each virtual channel input buffer length of the routing unit is F ($1 < F < M$) flits. The buffer on the direction of the local input of the routing unit is infinitely large, and the capacity of the unit processing the arriving packets is fast enough.

For a lot of applications running on the NoC platform, they usually have a plurality of data streams and on NoC, the average delay is the mean value of these data streams delay. Average delay of NoC can be expressed as T_N in equation (1),

$$T_N = \frac{1}{m}\sum_{i=1}^{m} T_{Li} = \frac{1}{m}\sum_{i=1}^{m}(T_{Ri} + W_{Si}) = \frac{1}{m}(T_b \times d + \sum_{j=1}^{d} B_{x_j,y_j,dir_j} + T_b \times (M-1) + W_{Si}) \qquad (1)$$

In equation (1), m is the number of data streams; T_{Li} is the transmission delay of the packet in a data stream in NoC; T_b is time of a flit passing a routing unit without blocking; B_{x_j,y_j,dir_j} is the input buffer blocking delay of head flit at the direction dir of the j^{th} routing along data stream. (x, y) is the coordinate of routing unit and dir indicates the direction, e.g. E, W, S, N; d is the number of hops for the data packet from the source node to the destination node; W_{Si} is the waiting time for flit head in the source node. As shown in equation (1), the key of calculating the average delay is to get the values of $B_{x,y,dir}$ and W_{Si}. This section will describe the algorithm of above two key parameters. $B_{x,y,dir}$ will be calculated by the following equation (2).

$$B_{x,y,dir} = \theta_{x,y,dir}\,\omega_{x,y,dir} = Bi \qquad (2)$$

In equation(2), $\theta_{x,y,dir}$ is the probability of the packet blocked in the dir direction of buffer in routing unit (x, y); $\omega_{x,y,dir}$ is the required average waiting time of data packet due to blocking in the buffer.

Average waiting time $\omega_{x,y,dir}$ will be gotten from M/G/1 queuing model.

$$\omega_{x,y,dir} = \frac{\lambda_{x,y,dir}}{2(1-\lambda_{x,y,dir}Ti_{x,y,dir})}\left[1+\frac{(Ti_{x,y,dir} - M \times T_b)^2}{Ti_{x,y,dir}{}^2}\right] \qquad (3)$$

In equation (3), $Ti_{x,y,dir}$ is the average service time of the packet. In order to calculate $\theta_{x,y,dir}$, the forward probability matrix F is the first step.

$$F = \begin{pmatrix} 0 & f_{NE} & f_{NS} & f_{NW} & f_{NL} \\ f_{EN} & 0 & f_{ES} & f_{EW} & f_{EL} \\ f_{SN} & f_{SE} & 0 & f_{SW} & f_{SL} \\ f_{WN} & f_{WE} & f_{WS} & 0 & f_{WL} \\ f_{LN} & f_{LE} & f_{LS} & f_{LW} & 0 \end{pmatrix} \qquad (4)$$

Forward probability f_{ij} is the probability of packet injected into the routing unit from i direction and leaving from j direction, which can be expressed as,

$$f_{ij} = \frac{\lambda_{ij}}{\sum_{\forall j}\lambda_{ij}}, where\ i, j = N, E, S, W \qquad (5)$$

λ_{ij} is data packet arrival rate in dir direction of input channel.

$$\lambda_{ij} = \sum_{\forall k,l}\sum_{\forall k',l'} \alpha_{k,l}\, d_{(k,l)(k',l')} R(k,l,k',l',x,y,dir) \qquad (6)$$

$\alpha_{k,l}$ is packet injection rate of node (k,l); $d_{(k,l)(k',l')}$ is the probability of packets sent to node (k',l') from node (k,l); $R(k,l,k',l',x,y,dir)$ is routing function. When the data packets sent from source node (k,l) to destination node $(k^{'},l^{'})$ use dir direction of the input channel in routing unit (x,y), $R(k,l,k^{'},l^{'},x,y,dir)$ is evaluated 1 or is evaluated 0. This paper uses XY dimension-order routing, so it is convenient to get data packet arrival rate of each of the input direction in routing unit according to equation(6). Takes $dir = E$ for example,

$$\theta_{x,y,E} = f_{EN}P_{x,y+1,S} + f_{ES}P_{x,y-1,N} + f_{EW}P_{x-1,y,E} + f_{EL}\theta_{x,y,E\to L} \tag{7}$$

In above equation, $P_{x,y,dir}$ is the probability that the buffer of node (x,y) is occupied, which will be introduced in the following. $\theta_{x,y,E\to L}$ is blocking probability of data packet that forwarded from the input buffer to the local port.

$$\theta_{x,y,dir\to L} = \sum_{\forall dir^{'},dir^{'}\neq dir} \lambda_{dir^{'}\to L} \Big/ \sum_{\forall dir^{'}} \lambda_{dir^{'}\to L} \tag{8}$$

When blocking occurs, the local output port is occupied by the other direction input buffers.

In NoC with wormhole exchange, a data packet is dispersed into multiple links and when the data packet header is injected into a link, it will occupy the link till the last flit leaves. A packet of occupancy of the link is shown in Figure 1.

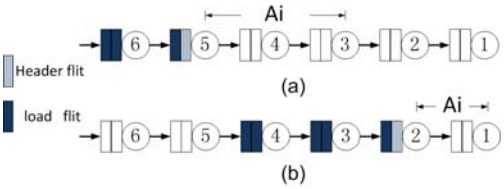


Figure 1. Wormhole routing transmission example

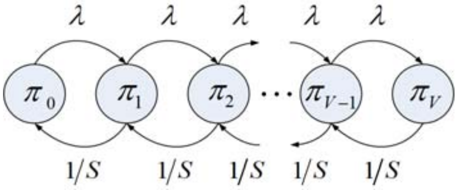


Figure 2. State transition of v occupied virtual channels

In Figure 1, data packets are transmitted to the routing unit 1 from routing unit 6 and injected into the destination node in routing unit 1. Assume the packet length is 6 flits. In (a), when the entire packet leaves the routing unit 5, header flit has yet reached the destination node, so only three downstream routing units of routing unit 5 affect the average service time of the packets; In (b), when part of packet still stay in routing unit 4, header flit has reached the destination node, so all the downstream routing units will affect its service time. Thus, we can get average service time $Ti_{x,y,dir}$ of the routing unit i which is deducted reversely along the transmission path from the destination node.

$$Ti_{x,y,dir} = \begin{cases} M \times T_O + \sum\limits_{j=i-Ai}^{i-1} Bj & i = 1,2,3,\cdots \\ M \times T_O & i = 0 \end{cases} \tag{9}$$

T_O is service time of the destination node network interface. Ai is the number of downstream routing unit affecting average service time.

$$Ai = \begin{cases} \lceil M / F \rceil & i \geq 1 + \lceil M / F \rceil \\ i - 1 & \text{else} \end{cases} \tag{10}$$

Wormhole routing in NoC, without the use of virtual channel technology, will increase packet delay greatly because of HOL blocking. In this paper, Markov chain is used to analyze occupied probability of v virtual channels, which is shown in Figure 2. In Figure 2, state π_v corresponds to that v virtual channels are occupied. State transition probability of π_v to π_{v+1} is data arrival rate λ and π_v to π_{v-1} is $1/S$. S is average service time. According to the Markov model, the occupied probability of v virtual channels is P_v.

$$P_v = (\lambda S)^v \frac{1-\lambda S}{1-(\lambda S)^{V+1}} \qquad 0 \le v \le V \tag{11}$$

Using dynamic virtual channel allocation mechanism, $P_{x,y,dir}$ is the probability that all virtual channels are occupied. That is equation (12).

$$P_{x,y,dir} = P_v = (\lambda_{x,y,dir} Ti_{x,y,dir})^V \frac{1-\lambda_{x,y,dir} Ti_{x,y,dir}}{1-(\lambda_{x,y,dir} Ti_{x,y,dir})^{V+1}} \tag{12}$$

V is the number of all virtual channels. Virtual channel mechanism of routing in paper is shown in Figure 3.

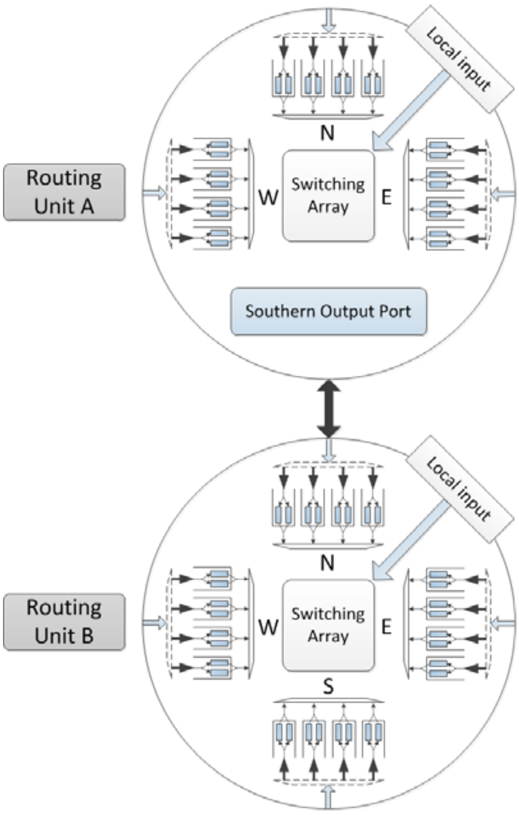
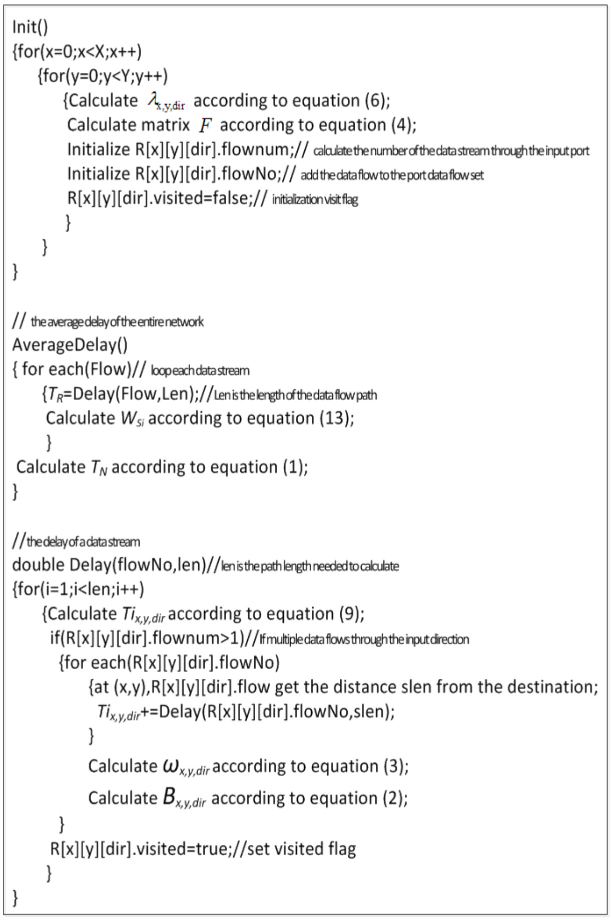


Figure 3. Virtual channel allocation mechanism



```
Init()
{for(x=0;x<X;x++)
   {for(y=0;y<Y;y++)
      {Calculate $\lambda_{x,y,dir}$ according to equation (6);
       Calculate matrix $F$ according to equation (4);
       Initialize R[x][y][dir].flownum;// calculate the number of the data stream through the input port
       Initialize R[x][y][dir].flowNo;// add the data flow to the port data flow set
       R[x][y][dir].visited=false;// initialization visit flag
      }
   }
}

// the average delay of the entire network
AverageDelay()
{ for each(Flow)// loop each data stream
     {$T_R$=Delay(Flow,Len);//Len is the length of the data flow path
      Calculate $W_{Si}$ according to equation (13);
      }
 Calculate $T_N$ according to equation (1);
}

//the delay of a data stream
double Delay(flowNo,len)//len is the path length needed to calculate
{for(i=1;i<len;i++)
     {Calculate $Ti_{x,y,dir}$ according to equation (9);
      if(R[x][y][dir].flownum>1)//If multiple data flows through the input direction
        {for each(R[x][y][dir].flowNo)
            {at (x,y),R[x][y][dir].flow get the distance slen from the destination;
             $Ti_{x,y,dir}$+=Delay(R[x][y][dir].flowNo,slen);
            }
            Calculate $\omega_{x,y,dir}$ according to equation (3);
            Calculate $B_{x,y,dir}$ according to equation (2);
        }
      R[x][y][dir].visited=true;//set visited flag
     }
}
```

Figure 4. Network average delay algorithm pseudocode

To the north input port of routing unit B, only the data of routing unit A in direction East, West, North, and local output buffer will be input. Allocate four virtual channels of B to the possible arrival of four input directions respectively. When the west input port of routing unit A sends a written request to the north input port buffer of routing unit B, if corresponding virtual channels of north input port is occupied, the request cannot get a response.

Calculating the average waiting time in source node is similar to the average waiting time in network on chip. Taking the network interface buffer channel of source node as an M/G/1 queue model, equation (13) is shown. In equation (13), T_R is network delay and $\lambda_{x,y,L}$ is arrival rate of data packet which is input from the source node to the network.

$$W_{Si} = \frac{\lambda_{x,y,L}}{2(1-\lambda_{x,y,L}T_R)}\left[1+\frac{(T_R - M \times T_b)^2}{{T_R}^2}\right] \quad (13)$$

2.3. Delay Model Algorithm

In this paper, the reverse deduction algorithm is used to solve delay model and following is the basic idea. Starting with the destination node, calculates the delay of the reverse deduction routing units along transmission path and gets average transmission delay of data packets on chip in the end. The algorithm pseudocode of calculating network average delay is shown in Figure 4.

Data structure R records the injection rate $\lambda_{x,y,dir}$ of an input port, forwarding probability matrix F, visits flag $visited$, average waiting time $\omega_{x,y,dir}$, blocking delay $B_{x,y,dir}$, data stream number $flowNo$, the amount of data streams $flownum$. $R[x][y][dir]$ represents the dir direction input port of the routing unit (x,y). Data structure $Flow$ records coordinates and injection rates of source node and destination node. $init()$ is to initialize $R[x][y][dir]$. $AverageDelay()$ is to calculate the average delay of network, which calls $Delay()$ first. In $AverageDelay()$, len is the length of the entire path between source node and destination node. Starting with the destination node, $Delay()$ calculates routing units delay by reverse deduction along transmission path. If other data streams flow through a routing unit input port, it has to calculate the average service time of them on this node by recursive invocation $Delay()$. len here is the length of the path from the destination node to the routing unit which is calculating.

3. Delay Model Validation

In order to verify the effectiveness of the delay model, this paper uses the existing RTL level model in the laboratory as simulation platform. Parameter settings are $M = 8$ and $T_b = 4$ which means routing unit transmitting a flit requiring four clock cycles. Topology is a two-dimensional mesh and grid size is 4, which is a 4×4 2D mesh. The model uses XY fixed routing algorithm.

Simulation experiments need to set space distribution and time distribution of traffic. In space distribution, two typical traffics are selected, uniform traffic and hotspot traffic. Each resource node sends data to the other nodes with the same probability $1/15$ in uniform traffic; in hotspot traffic, one or several nodes are usually set up as the hot and other nodes send packets to the hot spots with higher probability. Paper selects node $(2,2)$ as the hot and the probability of transmission data to it is twice that of the other nodes. In time distribution, data injection rate obeys Poisson distribution, which is adjusted with parameter $\lambda(packet/cycle/node)$. In each simulation experiment, there are about 100,000 packets sent to the destination node, wherein the first 10,000 data packets are not collected. This is to avoid the collection of the packets delay when the network is instable.

Experiment uses Intel Core i7 3.5GHz CPU and win8 operating system. The consuming time of simulation and delay model analysis are 113s and 3.6s respectively. The speed increased by more than 30 times. Comparing the results of simulation and delay model analysis under uniform traffic can get Figure 5.

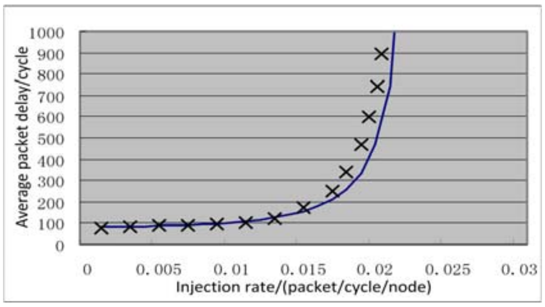


Figure 5. Comparing the results of simulation and analysis under uniform traffic

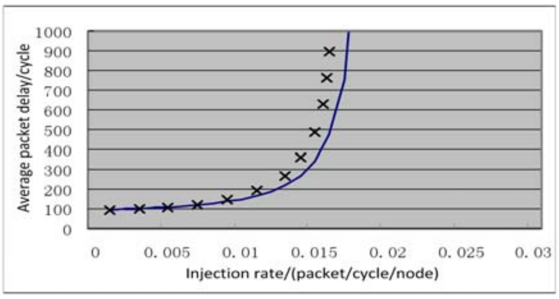


Figure 6. Comparing the results of simulation and analysis under hotspot traffic

The ordinate represents the transmission delay and the abscissa indicates the injection rate. It can be seen from Figure 5 that the results of delay model and simulation are closer at a low injection rate. When the injection rate reaches a certain value, the network congestion occurs and the average delay increases rapidly with bigger result deviation. Comparison result under single hot spot traffic is shown in Figure 6.

In Figure 6, the similar conclusions in Figure 5 also are obtained. In the analysis of the input channel blocking probability, this paper only considers the probability of the head buffer blocking flit header without the middle buffer blocking data flits in order to simplify the model, which makes bigger error. Overall, the results of the analysis of the delay model reflect the average transmission delay variations, and the average error is less than 8% compared with simulation data.

Delay model provides designers with a performance evaluation method, which can guide the design of NoC communication components and reduce hardware cost. Papers uses delay model to analyze routing unit performance in different buffer depth, packet length and virtual channel allocation mechanism. Figure 7 and Figure 8 show the specific results.

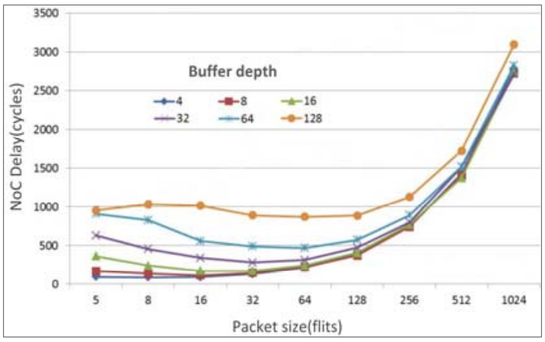


Figure 7. NoC delay for different buffer depth and packet sizes

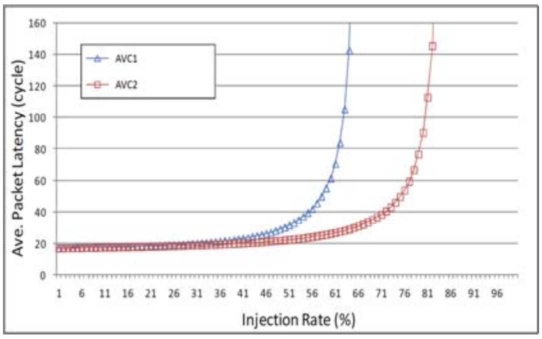


Figure 8. Virtual channel allocation mechanism for performance

Figure 7 presents the NoC latency obtained from experiments with different packet sizes (5-flit to 1024-flit) simulating varied buffer depth (4-flit to 128-flit length). It is observable a low NoC latency with smaller packet sizes such as 5-flit until 16-flit, even for 32-flit packet size, combined with buffer depth of 4-flit up to 32-flit. From 64-flit buffer depth there is a substantial increase and variability on the NoC latency. It happens due to large buffers enabling to insert more packets, which are concurring for the same paths in the NoC. As a consequence, the NoC latency is increased. In order to ensure performance of the system as much as possible within conditions of limited system resources, model can use the characteristic of imbalance traffic to allocate different amount of buffer to the input channel of each routing unit. Allocate more buffers to the input channels having heavy traffic and load. Buffer allocation algorithm can be gotten on the basis of network delay model.

This paper analyses the impact of two distribution mechanism to delay performance in the case of 4 virtual channels. In Figure 8, AVC1 represents static virtual channel allocation mechanism of routing unit and AVC2 represents a dynamic virtual channel allocation mechanism. Obviously, the performance of the latter is better than the former. Although dynamic allocation mechanism is more complex, the performance is significantly improved in the case of the same buffer depth. Therefore, the dynamic allocation mechanism is an important direction of improvement of the routing unit.

4. Conclusion

In the early stage of system design, assessment and decision-making selection to the delay of scheme can effectively shortens the design cycle and improves the efficiency of the design and development of high-performance NoC. This paper focuses on NoC delay model based on communication components in RTL level and establishes mathematical analysis model for NoC data transmission delay. This model considers routing unit limited buffer and virtual channel technology to the influence on network transmission delay. In addition, the paper presents the algorithm of delay model based on reverse deduction method. Compared with the simulation results of accuracy clock, the average analytical error is about 8% and the assessment speed increases by more than 30 times. Considering the versatility of delay model, the next step of this paper is to study the delay models with packet injection rate obeying adaptive distribution. In current delay model, the packet injection rate obeys Poisson distribution, while adaptive distribution is more suitable for NoC environment.

Acknowledgements

This work was supported by Main project of National 863 Program “2009AA012201” and the key scientific and technological projects of Science and Technology Commission of Shanghai Municipality “08dz501600”.

References

[1] Benini L, De Micheli G. Networks on chips: A new SoC paradigm. *Computers*. 2002; 35(1): 70-78.

[2] Murali S, Meloni P, Angiolini F. *Designing application-specific networks on chips with floorplan information*. Proceedings of IEEE/ACM International Conference on Computer-Aided Design. San Jose. 2006; 355-362.

[3] Jeang Yuan L, Hung Chung W, Chiang Chuen M. *A methodology based on maximal-profit spanning tree for designing application specific networks on chip*. Proceedings of First International Conference on Innovative Computing, Information and Control. Beijing. 2006; 2: 18-21.

[4] Andreas H, Maarten W, Arno M. *Applying dataflow analysis to dimension buffers for guaranteed performance in networks on chip.* Proceedings of Second ACM/IEEE International Symposium on NoCS. Newcastle upon Tyne. 2008; 211 – 212.

[5] Banerjee N, Vellanki P, Chaha KS. *A power and performance model for Network-on-Chip architectures*. Proceedings of DATE2004. Paris. 2004; 1250-1255.

[6] Kreku J, Hoppari M, Kestilä T. *Application-platform performance modeling and evaluation, Specification, Verification and Design Languages.* Proceedings of Forum on FDL 2008. Stuttgart. 2008; 43-48.

[7] Hemayet, Hossain. *GPNOCSIM-A General purpose simulator for network-on-chip*. Proceedings of International Conference on Information and Communication Technology 2007. Cairo. 2007; 254-257.

[8] Y Li, L Liang. A NoC Modeling and Simulating Method with SystemC. *Microelectronic & Computer*. 2010; 78-82.

[9] Dally, WJ. Performance analysis of k-ary n-cube interconnection networks. *IEEE Transactions on Computers*. 1990; 39(6): 775-785.

[10] Huang TC, Ogras UY, Marculescu R. *Virtual channels planning for networks-on-chip*. Proceedings of International Symposium on Quality Electronic Design. San Jose. 2007; 879-884.

[11] Ogras UY, Marculescu R. *Analytical router modeling for networks-on-chip performance analysis*. Proceedings of Design, Automation and Test in Europe Conference. Nice. 2007; 1096-1101.

[12] Sepulveda J, Strum M, Wang JC. *A multi-objective approach for multi-application NoC mapping, Circuits and Systems (LASCAS).* Proceedings of 2011 IEEE Second Latin American Symposium on. Bogota. 2011; 1-4.

[13] Marculescu R, Ogras U Y, Li-Shiuan P, Jerger N E, Hoskote Y. Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2009; 28(1): 3-21.
[14] Zhou W B, Zhang Y, Mao Z G. An application specific NoC mapping for optimized delay. *DTIS 2006*. La Marsa. 2006; 184-188.
[15] Shengguang Y, Li L, Minglun G, Yuang Z. An Energy- and Delay-aware Mapping Method of NoC. *Journal of Electronics*. 2008; 13(5): 937-942.
[16] Marculescu R, Chou CL. *Contention-aware application mapping for network on chip communication architecture*. Proceedings of IEEE International Conference on Computer Design. Lake Tahoe. 2008; 164-169.
[17] Mingche L, Zhiying W, Kui D. A Novel Performance Analysis Approach for Network on Chip Based on Analytical Router Modeling. *Journal of Computer-Aided Design & Computer Graphics*. 2009; 3(21): 339-345.