A Performance Model for Hypercube Based NoC with Fully Adaptive Routing Algorithm

Jin Liu[1], Xiaofeng Wang[1], Hongmin Ren[1], Jin Wang[2], Jeong-Uk Kim[3]

[1] College of Information Engineering, Shanghai Maritime University, Shanghai 200135, China
[2] Computer and Software School, Nanjing University of Information Science & Technology, Nanjing 210044, China
[3] Department of Energy Grid, Sangmyung University, Seoul 110-743, Korea
{jinliu, xfwang, hmren}@shmtu.edu.cn；wangjin@nuist.edu.cn；jukim@sum.ac.kr

Abstract. This paper presents a performance model for predicting average message latency under uniformly distributed traffic in a hypercube based network-on-chip (NoC). Unlike previous works, the model obtains service rate for incoming traffic at a particular channel of a node by calculating reverse service rate provided by downstream nodes, and has simple closed-form calculation to produce accurate analytic results when the system is operating in stable state. The validity of the proposed model is shown by comparing results calculated out by the model with results obtained through simulations with different network configurations and traffic loads.

Keywords: NoC; Model; Hypercube; Adaptive Routing.

1 Introduction

With the rapid development of VLSI, number of components that are interacting to compute a solution in SoC keeps increasing. Having dedicated connections between any given modules could be extremely complex as the number of modules increases. A viable alternative could be an interconnection network within the chip. There are a number of network designs with different topology for network on chip reported in the literatures. [1-4] As parallel architectures become larger and faster, packaging and physical constrains are assuming more important roles, and hypercube-based NoCs as shown in Fig.1, have symmetry, regularity and can reduce network's bisection width, thus they are receiving increasing attentions. NoCs employ hierarchical architecture which is similar as usual computing network, the switching and routing technology are vital for system performance. Wormhole switching [5] and virtual channel mechanism have been widely used in practical network on chip [6] to get pipelined data transmission to reduce average message latency, make better utilization of physical channel while using minimal local buffer and avoiding deadlock. Routing algorithms can be implemented as either deterministic or adaptive. Deterministic routing protocols choose the path based on the message's source and destination. When using deterministic routing, a packet will be delayed if any channel along the path is busy

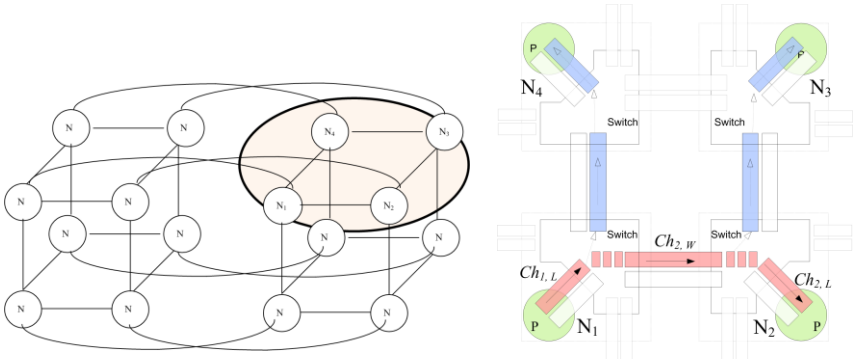
with other packets, or even worse, if a channel along the path is faulty then the packet cannot be delivered. Nevertheless, deterministic routing still has been widely used due to its simplicity [6]. And its analytical model has been widely reported in the literature [7-9]. Adaptive routing protocols which provide alternative paths for communicating nodes have been proposed to make more efficient use of bandwidth and to improve fault tolerance of interconnection network. Several adaptive routing algorithms have been proposed, showing that message blocking can be considerably reduced, thus strongly improving network throughput. Among them, routing algorithms based on Duato's design methodology [10] are extensively used. These routing algorithms split each physical channel into two virtual channel sets, the adaptive and the deterministic channels. When the paths of adaptive channels are blocked, a message uses an escape channel at the congested node. If there is any free adaptive channel available at subsequent nodes, the message can go back to the adaptive channels. Adaptive routing algorithms can be further categorized to progressive and backtracking algorithms. Progressive routing algorithms move the message header forward by reserving a new channel. In our analytical model, we assume the routing algorithm is progressive, no backtracking is allowed. Several analytical models for wormhole switching network using fully adaptive routing protocols were reported in the literature, but their calculation process is complicated and the presented results only hold in relatively small network state region[11-13],.

Unlike previous works, this paper presents a novel performance analysis model for hypercube based network on chip with wormhole routing, virtual channels and fully adaptive routing. The model has relatively simple calculation process and yields satisfactory predictions in the network's steady state region. The rest of this paper has been organized as follows. In Section 2, we present the model in detail. In Section 3, validation of the proposed model is made by comparing its prediction with simulation results. Some concluding remarks are included in Section 4.

2 The Performance Model

2.1 System Description

A hypercube network as shown in Fig.1 (A) is used to illustrate our model. Each node consists of a processing element (PE) and a switch. PE is responsible for generating messages and consuming messages from other nodes. Each switch has 5 input and output channels. PE is connected to switch by local injection/ejection channel. A node is connected to 4 adjacent neighboring nodes by bi-directional network channels. A message which is generated from a node's PE will first be transmitted to the switch by local injection channel. Then it will be routed toward the destination. At the destination node, the message is transmitted to PE through local ejection channel. Thus a message has to travel through at least 3 links from the source to destination. For instance, as shown in Fig. 1 (B), the message generated in node N1 has to traverse channels Ch1,L, Ch2,W and Ch2,L to arrive at its destination PE in N2.



A. The hypercube-based NoC B. Four communicating nodes

Fig. 1. A 16 nodes hypercube based NoC system

The model is also based on the following commonly accepted assumptions.[7-13].

1) Each node generates traffic independently with traffic following a Poisson process on a mean rate of M_{gen} messages/cycle.
2) Message destinations are uniformly distributed across the network nodes. Although in an actual application, if node A sends a message to node B it's highly possible that B will send back a message to A.
3) Messages are of constant length L flits. A message is long enough so that its data flits span from source to destination nodes.
4) Each flit requires one cycle to be transmitted from one node to the next over physical link between them. Two cycles are needed for a flit to cross a node, i.e. from an input buffer to an output port in absence of blocking.
5) The local queue at the injection channel in the source node has infinite capacity. Messages at the destination node are transferred to the local PE one at a time through the ejection channel.
6) A set of virtual channels are multiplexed across a physical channel. Virtual channels share physical channel's buffer and each has its own single flit space.
7) Physical channels between any two adjacent nodes are duplex. More than two virtual channels are used for each direction of a physical channel. If there are adaptive virtual channels available, a message can use a random one.

2.2 Calculation of Average Message Latency

There are a few notations used in the derivation of our model, a brief summary of them can be found in the Appendix. The average message latency T_{msg} comprises of the message transmission delay across the network channel t_w, the intra-router delay t_s, [5] the average contention delay W_q at the network channels and the average delay W_{ej} at nodes' local ejection channel. It can be computed as follows:

$$T_{msg} = (D+L)\cdot T_{tr} + (N_q - 1)\cdot W_q + W_{ej} \quad (1)$$

Where T_{tr} is given by max(t_s, t_w). It demonstrates the nature of pipelined flits transmission of wormhole switching in absence of contention. $(D+L)T_{tr}$ denotes the mean time that a message's header flit and data flits need to travel from source to destination. $(N_q\text{-}1)W_q$ shows the waiting time that header flit experienced at the $N_q\text{-}1$ channels of intermediate nodes. As the minimum link number that a message travels is 3, the average hops that messages take, D can be obtained by:

$$D = \sum_{i=3}^{k} p_i \cdot i \quad (2)$$

Where k is the diameter of the network, and p_i denotes the probability that a message's travel path is i links long. Under the uniform traffic pattern, the average traffic arrival rate λ for each channel is determined by the message generating rate M_{gen}, average routing hops D and output channels number of each node o.[4]

$$\lambda = \frac{M_{gen} \cdot D}{o} \quad (4)$$

In order to receive service from a link, the message's header flit will acquire a virtual channel. A VC keeps serving the message until all the data flits flow across this node. When the traffic rate is light, there is no congestion; the service time S and service rate SR at each channel can be defined respectively as follows:

$$S = t_s \cdot (L+1) \quad (5)$$

$$SR = \frac{1}{S} \quad (6)$$

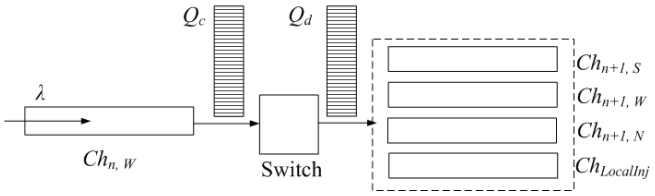


Fig. 2. Queuing model for an input channel

As traffic rate keeps increasing, congestion appears in the network and waiting queues build up at bottleneck links. In this case, the service that one channel can provide to the incoming messages is not only determined by its own service capacity, but also by the blocking state of its immediate downstream channels. In a hypercube network, the traffic arrival rates for input channels of a node are equal to each other due to its symmetry. Without loss of generality, suppose the node that we analyze is at nth hop of a messages routing path. We analyze queuing system model of channel $Ch_{n,W}$ at west input port. The waiting queue $Q_{n,W}$ at channel $Ch_{n,w}$ is treated as two distinct queues Q_c and Q_d as shown in Fig. 2. Q_c is the result of contentions at $Ch_{n,w}$ which is determined by the traffic arrival rate λ and the router self's service rate SR. Q_d is due to the contention that a message experiences when it's to be accepted by a downstream input channel. An $M/M/m$ queuing system is used to model Q_c.

The probability that an arrival message will find v virtual channels are busy and will be forced to wait in queue can be obtained by the following equation: [14]

$$P_Q^{'} = \frac{p_0 \cdot (v \cdot \rho)^v}{v!(1-\rho)} \tag{7}$$

Where p_0 is given by:

$$p_0 = \left(\sum_{n=0}^{v-1} \frac{(v \cdot \rho)^n}{n!} + \frac{(v \cdot \rho)^v}{v!(1-\rho)} \right)^{-1} \tag{8}$$

For Q_d, the service rate, μ_r, is the service rate that offered by all the immediate downstream channels to $Ch_{n,w}$. To obtain μ_r, we can further divide Q_d to four queues, each of which is associated with an immediate downstream channel in one of the possible downstream directions, *i.e*. east, south, north in this case and local injection, as shown in Fig. 4. Consider the waiting queue at west downstream channel $Ch_{n+1,W}$. we can get average waiting time for a message by: [14]

$$W_{q_d} = \frac{P_Q^{'} \cdot \rho}{\lambda \cdot (1-\rho)} \tag{9}$$

Thus the average time that a message spends at the queuing system of $Ch_{n+1,W}$ is $W_{qd} + S$. In addition, there are v virtual channels in $Ch_{n+1,W}$ and $Ch_{n+1,W}$ has o possible inputs(including local injection channel), therefore we can get μ_r^w as :

$$\mu_r^w = \frac{v}{o \cdot (S + W_{qd})} \tag{10}$$

Hence, we get μ_r as:

$$\mu_r = p_w \mu_r^w + p_s \mu_r^S + p_N \mu_r^N + p_L \mu_r^L \tag{11}$$

Due to symmetry of hypercube, we know that $p_W=p_S=p_N$, so $\mu_r=3\ p_W \mu_r^w + P_L \mu_r^L$. Using Little's Theorem, the average number of messages in the whole queuing system at $Ch_{n,W}$ can be derived as:

$$N = \lambda \cdot T = \frac{\lambda}{\mu} + \frac{\lambda \cdot P_Q}{v \cdot \mu - \lambda} \tag{12}$$

Also by Little's Theorem, the average number of messages in Q_c and Q_d are:

$$N_d = \frac{\lambda}{\mu_r - \lambda} \tag{13}$$

$$N_c = \frac{\lambda}{SR} + \frac{\lambda \cdot P_Q}{v \cdot SR - \lambda} \tag{14}$$

As $Q_{n,E}$ comprises of Q_c and Q_d, combine Eq. (12), (13) and (14), we get:

$$\frac{\lambda}{\mu} + \frac{\lambda \cdot P_Q}{v \cdot \mu - \lambda} = \frac{\lambda}{SR} + \frac{\lambda \cdot P_Q}{v \cdot SR - \lambda} + \frac{\lambda}{\mu_r - \lambda} \tag{15}$$

Based on Eq. (7) and (8), replace P_Q in Eq. (15), solve the resulting nonlinear equation, we can get service rate μ at channel $Ch_{n,W}$. Then, by Little's Theorem again,

we can obtain the average time W_q that a message has to wait in queuing system at the intermediate nodes of a message's traveling path:

$$W_q = \frac{P_Q \cdot \rho}{\lambda \cdot (1-\rho)} \tag{16}$$

Finally, as we mentioned that the node's local ejection channel is treated as independent *M/M/m* queuing system; the average waiting time on ejection node channel can be obtained by:

$$W_{ej} = \frac{P_{Q_{ej}} \cdot \rho_{ej}}{(\lambda + \lambda') \cdot (1-\rho_{ej})}, \quad \rho_{ej} = \frac{\lambda + \lambda'}{v \cdot SR} \tag{17}$$

At this point, the message delay T_{msg} defined in Eq. (1) can be calculated out as all the unknown variables at the right hand side are all obtained.

3 Validation of the Model

A number of simulation experiments have been carried out to validate the proposed model. The simulator we used is based on flexsim1.2 [7]. Due to space limitation, results presented here are average message latency obtained from both model and simulations on 64 and 256 nodes hypercubes. In both calculations and simulations, 4 VCs multiplexing a PC and the message size is set to 32 flits. As shown in Fig. 3 and 4, the simulation results and predictions of our model match well from a very light usage of a network channel to about 50% average channels utilization after which point the network gets into saturation state.

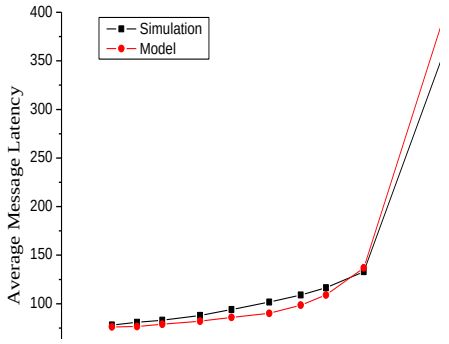


Fig. 3. Model and Simulation results in 64 nodes hypercube network

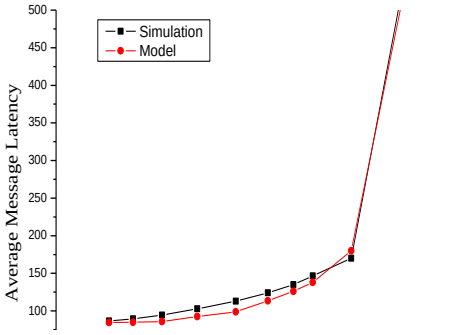


Fig. 4. Model and Simulation results in 256 nodes hypercube network

4 Conclusions and Future Work

We have presented an analytical model for predicting average message latency and average link waiting time in a hypercube network which employs wormhole switching, virtual channel and adaptive routing. Unlike previously proposed models, this model is based on general queuing theory and probability analysis. It can be concluded that our model provides an effective and practical evaluation tool. In the future work, since this model obtains message waiting time at each queuing system at channels of a routing path, we will adapt it for networks with other topologies and take different traffic types into consideration.

Appendix

Notations

$Ch_{n,\,dir}$ - Physical channel in *dir* direction of a message's nth hop node.
dir - Direction. Each nodes has ports on 5 directions, i.e. East, South, West, North and Local
D - Average path length for all the delivered messages
L - Average message length
M_{gen} - Average message generation rate at each node
N_q - Average number of nodes along the path
o - In/output ports number of a node (not including local channel port), in our network, it's 4.
P_Q' - The blocking probability for a message at a physical channel without contention.
P_Q - The blocking probability for a message at a physical channel
S - The service time for a message at a channel w/o contention.
SR - The service rate for a message at a channel w/o contention
T_{msg} - The mean latency for all the delivered messages
T_s - Routing (Switching) delay across a node
T_w - Propagation delay across the physical channel
v - The number of virtual channels multiplexing a transmission direction of physical channel
W_q - Average waiting time for a message at each intermediate nodes
W_{ej} - Mean waiting time for a message on ejection node channel
λ - The average message arrival rate at a channel
μ - The service rate for a message at a physical channel
μ_r^{dir} - The service rate that a channel observes from the immediate downstream channel on dir direction.
μ_r - The service rate that a channel observes from all its immediate downstream channels at nodes on the path.
μ_{rE} - The service rate that a channel observes from all its immediate downstream channels at destination node.

Acknowledgements

This work was supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Knowledge Economy (MKE, Korea). Professor Jeong-Uk Kim is the corresponding author.

References

1. P. Guerrier and A. Greiner, “A Generic Architecture for On-Chip Packet-Switched Interconnections,” Proc. Design and Test in Europe (DATE), pp. 250-256, Mar. 2000.
2. S. Kumar et al., “A Network on Chip Architecture and Design Methodology,” Proc. Int’l Symp. VLSI (ISVLSI), pp. 117-124, 2002.
3. W.J. Dally and B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks,” Proc. Design Automation Conf. (DAC), pp. 683-689, 2001.
4. F. Karim et al., “An Interconnect Architecture for Networking Systems on Chips,” IEEE Micro, vol. 22, no. 5, pp. 36-45, Sept./Oct.2002.
5. J.Duato, S. Yalmanchili and L. Ni, Interconnection Networks, IEEE Computer Society, 1997
6. P.P. Pande. et al. "Performance Evaluation and Design Trade-offs for MP-SoC Interconnect Architectures”, IEEE Trans. Computers, vol. 54, no. 8, Aug 2005, 1025-1040.
7. J.T. Draper and J. Ghosh, “A Comprehensive Analytical Model for Wormhole Routing in Multicomputer Systems,” J. Parallel and Distributed Computing, vol. 32, pp. 202-214, 1994.
8. R. Greenberg and L. Guan, “Modeling and Comparison of Wormhole Routed Mesh and Torus Networks,” Proc. Ninth IASTED Int'l Conf. Parallel and Distributed Computing and Systems, 1997.
9. W.J. Guan, W.K. Tsai, and D. Blough, “An Analytical Model for Wormhole Routing in Multicomputer Interconnection Networks”, Proc. Int'l Conf. Parallel Processing, pp. 650-654, 1993.
10. J. Duato, “A New Theory of Deadlock-Free Adaptive Routing in Wormhole Routing Networks,” IEEE trans. Parallel and Distributed Systems, vol. 4, no. 12, pp. 1,320-1,331, Dec. 1993.
11. Y. Boura, C.R. Das, and T.M. Jacob, “A Performance Model for Adaptive Routing in Hypercubes,” Proc. Int'l Workshop Parallel Processing, pp. 11-16, Dec. 1994.
12. M. Ould-Khaoua, “A Performance Model for Duato's Fully Adaptive Routing Algorithm in k-Ary n-Cubes”, IEEE Transactions on Computers, v.48 n.12, p.1297-1304, Dec 1999.
13. Mohamed K.Ould, Hamid A.Sarbazi, Analytical Model of Adaptive Wormhole Routing in Hypercubes in the Presence of Hot Spot Traffic, IEEE Tran. on parallel and distributed systems,vol.12, no.3, Mar 2001
14. D. Bertsekas and R. Gallager, Data Networks, second edition. Prentice-Hall, 1992.