Analysis of Circuit Switching for the Torus Interconnect Networks with Hot-Spot Traffic*

F. Safaei[1,3], A. Khonsari[2,1], M. Fathy[3], M. Ould-Khaoua[4]

[1] IPM School of Computer Science, Tehran, Iran
[2] Dept. of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[3] Dept. of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[4] Dept. of Computing Science, University of Glasgow, UK
{safaei, ak}@ipm.ir, {f_safaei, mahfathy}@iust.ac.ir, mohamed@dcs.gla.ac.uk

Abstract

Several analytical models have recently been proposed for Circuit-Switched interconnect networks under the uniform traffic pattern. However, there has been hardly any model reported yet that deals with other important non-uniform traffic patterns, such as hot-spots. This paper presents a new mathematical model to capture the mean message latency in the torus interconnect network with Circuit Switching in the presence of hot-spot. Simulation experiments demonstrate close agreement between the observed network behavior and those obtained by the analytical model.

Keywords: Parallel computers, Interconnect networks, Torus, Circuit switching, Hot-spot traffic, Queuing theory, and Performance evaluation.

1. Introduction

Large-scale parallel computers, Multiprocessors System-on-Chip (MP-SoCs), multicomputers, cluster computers and peer-to-peer networks are potential candidates for providing very high computational power. The systems are usually organized as a number of nodes where each node has its own local processor, memory, and other supporting devices. Such networks may accept a message from any processing node, and deliver it to any other processing node. Interconnection network design greatly affects both system cost and performance [1, 2].

The communication latency of networks depends on several factors including topology, switching, and routing. The topology of a network defines how the nodes are interconnected and is generally modeled as a graph in which the vertices represent the nodes and the edges denote the channels. The torus (also known as *k*-ary 2-cube) has become a widely accepted interconnection network due to its desirable and powerful topological properties. Examples of experimental and commercial systems based on the torus include Cray T3D [3], Cray T3E [4].

In most parallel computer systems, a message enters the network from a source node and is switched or routed towards its destination through a series of intermediate nodes. In Circuit Switching (CS), a dedicated path is established between source and destination before the data transfer initiates. Once the data transfer is initiated, the message is never blocked. As the channels creating the path are reserved exclusively, buffering of data is not required. On the other hand, establishing the path requires significant overhead during the data-transmission phase; all channels are reserved for the entire duration of message transfer. The most notable advantage of CS is its ability to provide messages with an agreed-upon *Quality of Services* (QoS), e.g., guaranteed latency, once a connection has been established. This feature makes circuit-switched networks suitable for supporting simultaneous (data, voice and image) communications across parallel computers, distributed computers, and telecommunication systems. Circuit-switched networks accomplish simultaneous communications by means of disjoint paths of electrical links and switches. Recently there has been renewed interest in CS because of the ease of building very high capacity circuit switches. In [5] it is shown that the core of the network, where access links limit the maximum flow rate, and where high capacity is needed most, there is a little or no difference in performance between CS and PS or WS. Given that CS can be built to have higher capacity than other well-known switching methods, this suggests that CS warrants further investigations.

* This research was in part supported by a grant from I.P.M. (No. CS1384-3-01).

Routing algorithms for large-scale parallel computers, MP-SoCs, multicomputers and cluster computers are generally classified as being either *deterministic* or *adaptive*. Deterministic routing is simple and easily implemented, with minimal overhead. Adaptive routing, on the other hand, improves both the performance and fault-tolerance of a communication network and, more importantly, allows for further flexibility at the cost of additional complexity in the algorithm and its implementation.

Several models analyzing CS have been proposed in the literature over the recent years [6-10]. However, the performance properties of CS have not been thoroughly investigated in the presence of non-uniform traffic patterns. A non-uniform traffic that has attracted much attention is the hot-spot model which leads to extreme network congestion resulting in serious performance degradation due to the tree saturation phenomenon in the network. This paper proposes a new analytical model to compute message latency for CS in the torus networks under hot-spot traffic. The model achieves a good degree of accuracy which is evident by the results gathered from simulation experiments to validate the proposed model.

The rest of the correspondence is organized as follows. In Section 2, we introduce some definitions and give some preliminaries. In Section 3, we give a brief overview of the assumptions used in this paper. Moreover, we describe the proposed analytical model in this section. In Section 4, we compare the delays predicted analytically with those obtained through simulation experiments. Finally, Section 5 summarizes our findings and concludes the paper.

2. Two-dimensional torus network and its router structure

The 2-D torus, consists of $N=k^2$ processors arranged along the points of a 2-D space that have integer coordinates. Along each dimension, there are k processors with identities (x_1, x_2), $x_i=0, 1, \ldots, k-1$, where x_1 represents the row position and x_2 indicates the column position of the node. Two processors (x_1, x_2) and (y_1, y_2) are connected by a (bi-directional) link if and only if $x_1=(x_2+1) \bmod k$ or $y_1=(y_2+1) \bmod k$. Thus, each node is connected to two neighbouring nodes in each dimension and consists of a processing element (PE) and a router. The PE to inject/eject messages to/from network uses the remaining channels, respectively. Messages generated by the PE are transferred to the router through the injection channel. Messages at the destination are transferred to the local PE through the ejection channel. Each physical channel is associated with some, say *V, virtual channels*. A virtual channel has its own flit queue, but shares the bandwidth of the physical channel with other virtual channels in a time-multiplexed fashion [11].

3. Analytic performance modeling

In this Section we present an analytical performance modeling for the torus network using CS in the presence of hot-spot traffic.

3.1 Assumptions

The proposed model is built on the basis of the following assumptions which are widely used in the similar studies [6-12].

1. The traffic model is based on Pfister and Norton approach [13] and is used to generate hot-spot traffic pattern. In their method, each generated message has a finite probability α of being directed to the hot-spot node and probability $(1-\alpha)$ of being directed to other network nodes. We usually refer to these types of messages as *hot-spot* and *regular*, respectively.
2. Each processor generates messages independently, which follows a Poisson process with a mean rate of λ_g including regular and hot-spot fractions, $\alpha\lambda_g$ and $(1-\alpha\lambda_g)$, respectively.
3. The arrival process at a given communication network is approximated by an independent Poisson process. Therefore, the rate of process arrival at a communication network can be calculated using *Jackson's queuing networks* formula [12].
4. The destination of each message would be any node in the network with uniform distribution.
5. Message length is fixed at M flits, each of which requires one cycle to cross from one node to the next.
6. V virtual channels $(V\geq 1)$ are used per physical channel. When there is more than one virtual channel available that bring a message closer to its destination, one is chosen at random.

3.2 The proposed analytical model

The average message latency is composed of the average network latency, $\overline{S}$, that is the average time to cross the network, and the average waiting time seen by a message at the source node, $\overline{W}_s$. However, to capture the effects of virtual channels multiplexing, the average

message latency has to be scaled by a factor, $\bar{V}$, representing the average degree of virtual channels multiplexing that takes place at a given physical channel. Therefore, we can write the average message latency [14]

$$Latency = (\bar{S} + \bar{W}_s)\bar{V} \quad (1)$$

The regular and hot-spot messages see different network latencies as they pass different number of channels to reach their destinations. Let $\bar{S}$ denotes the average network latency seen by a message i.e., a message that needs to cross from source to destination. If $\bar{S}_r$ and $\bar{S}_\alpha$ denote the average network latency for regular and hot spot messages, respectively, the average network latency taking into account both types of messages is given by

$$\bar{S} = (1-\alpha)\bar{S}_r + \alpha\bar{S}_\alpha \quad (2)$$

The average number of hops that a regular message visits along a given dimension and across the network, $\bar{k}$, $\bar{d}$ respectively, are given by Agarwal [15]

$$\bar{k} \approx k/4, \ \bar{d} = 2\bar{k} \quad (3)$$

Fully adaptive routing allows a regular message to use any channel that brings it closer to its destination, resulting in an evenly distributed regular traffic rate on all network channels. A router in the torus has 2 output channels and the PE generates, on average, (1- $\alpha\lambda_g$) regular messages in a cycle. Since each regular message travels, on average, $\bar{d}$ hops to cross the network, the rate of regular messages received by each channel, λ_r, can be expressed as

$$\lambda_r = (1-\alpha)\lambda_g\bar{d}/2 = (1-\alpha)\lambda_g\bar{C}_r/4 \quad (4)$$

Where, $\bar{C}_r$ (calculated below) is the average time needed to setup a path for a regular *r*-hop message header. The number of nodes, which are *r*-hop away from a given node in a $k \times k$ torus, is given by [10]

$$N_r = \begin{cases} r+1 & r < k \\ 2k-r-1 & k \le r \le 2k-2 \\ 0 & r > 2k-2 \end{cases} \quad (5)$$

By Eq. (5) we find that, the number of all channels, which located *j* hops away from the hot-spot node, is $2N_{j-1}$. Therefore, the number of source nodes for which one of these $2N_{j-1}$ channels can act as intermediate channel to reach the hot-spot node is given by

$$N - \sum_{r=0}^{j-1} N_r = \sum_{r=j}^{2(k-1)} N_r \quad (6)$$

Given that each of the *N* nodes generates, on average, $\alpha\lambda_g$ hot-spot messages in a cycle, the rate of hot-spot traffic, λ_{α_j}, received by a channel located *j* hops away from hot-spot node is simply given by

$$\lambda_{\alpha_j} = \alpha\lambda_g \sum_{r=j}^{2(k-1)} N_r \Big/ \left(2N_{j-1}\right) \quad (7)$$

To determine the total input traffic rate for the network, we calculate the traffic rate on the channel is located *j* hops from the hot-spot node and add this value to the rate of messages arriving at the channel consists of the traffic rate of regular messages. Therefore, the overall traffic rate is computed as

$$\lambda_j = \lambda_r + \lambda_{\alpha_j} \quad (8)$$

In CS, the network latency for a regular message consists of two parts: one the time to setup a path and other, the delay due to the actual message transmission time. Thus, the network latency of an *r*-hop regular message with message length *M* can be written as

$$\bar{S}_r = M + r + \bar{C}_r \quad (9)$$

Also, the latency seen by a hot-spot message, which is *j* hops a way from the hot-spot node is

$$\bar{S}_{\alpha_j} = M + j + \bar{C}_\alpha \quad (10)$$

Where, *M* is the message length and $\bar{C}_\alpha$ is the average time needed to setup a path for a hot-spot message header (given below). Note that in Eqs. (9) and (10) the terms *r* and *j* both accounts for *r* and *j* cycles that are required to send the acknowledgement flit back to the source node.

Since adaptive routing distributes regular traffic evenly across the network channels, the average service time seen by a regular message is the same across all channels. When a regular (or hot-spot) message reaches a channel that is *j* hops away from the hot-spot node, the mean service time at the channel, considering both regular and hot-spot message with their appropriate weights, can be written as

$$\bar{S}_j = \left(\lambda_r/\lambda_j\right)\bar{S}_r + \left(\lambda_{\alpha_j}/\lambda_j\right)\bar{S}_{\alpha_j} \quad (11)$$

Moreover, the probability of being *j* hops away from a given node as destination is

$$p_{\alpha_j} = N_j/(N-1) \quad (12)$$

Consequently, the average network latency seen by a hot-spot message can be written as

$$\bar{S}_\alpha = \sum_{j=1}^{2(k-1)} p_{\alpha_j}\bar{S}_{\alpha_j} \quad (13)$$

In order to compute the average path set up, $\bar{C}$, we employ a Markov chain (depicted by Fig. 1) to model the header behaviour to cross the network [16].

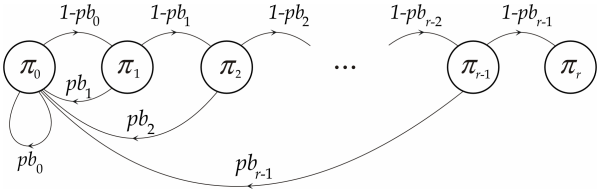

Fig. 1: The Markov chain diagram for calculating the average path setup time.

Each state in Fig. 1 represents the current location (i.e. node) of the header along its network path. States π_0 and π_r denote that the header is at the source and destination nodes, respectively. State π_j ($1 \le j \le r$-1) corresponds the case where the header is at intermediate node that is j-hops away from the source; or r-j+1 hops remain to reach to the hot-spot node. A transition out of state π_j to π_{j+1} implies that the header succeeds in acquiring virtual channel and brings it one hop closer to its destination. Also, each transition from state π_j to state π_0 means that the header has encountered blocking and has to backtrack to the source node. The transition rate is the probability, pb_j, that the header is blocked at the intermediate node corresponding to state π_j. Therefore, 1-pb_j denotes the transition probability of advancing across the reserved path. In what follows, we calculate the expected duration to reach state π_r starting from state π_0 corresponds to the average time for the header to reserve a path from the source to destination both for normal and hot-spot cases. This time can be computed using the first step analysis method applied to Markov chains [16].

Let $\bar{C}_j$ be the average time interval to reach the state π_r originating from state π_j. $\bar{C}_j$ is always finite [16], and $\bar{C}_{j+1}$ denotes the header at state π_j succeeds in acquiring a virtual channel and it can proceeds to state π_{j+1}. On the other hand, when the header encounters situation of blocking, it backtracks to the source node corresponding to state π_0 and the residual expected duration would be $\bar{C}_0$. It is assumed that the header needs one cycle to move from one node to another. The above argument reveals that the average time, $\bar{C}_j$, satisfies the following equation

$$\bar{C}_j = \begin{cases} (1-pb_j)(\bar{C}_{j+1}+1) + pb_j(\bar{C}_0 + j) & 0 \le j \le r-1 \\ 0 & j = r \end{cases} \quad (14)$$

Solving the above equation yields the expected time, $\bar{C}_0$, for the header to reach the destination originating from the source node. Once the header reaches its destination, an acknowledgment flit is transmitted back to the source through the reserved path. Therefore, the average time to setup a path for an r-hop regular message can be written as

$$\bar{C}_r = \bar{C}_0 + r \quad (15)$$

Similarly, the time is needed to setup a reserved path for hot-spot message that is j-hops away ($1 \le j \le 2(k-1)$) from the hot-spot node is given by

$$\bar{C}_{\alpha j} = \bar{C}_0 + j \quad (16)$$

To compute the probability of blocking we use the method described in [6]. If P_V denotes the probability of V virtual channels at a given physical channel are busy, the probability pb_j, that the header is blocked is given by

$$pb_j = \sum_{t=0}^{1} \psi_j^t (P_V)^{2-t} \quad (17)$$

Where ψ_j^t is the probability that the header has entirely crossed t dimensions along on its j-hop path. The derivation of probability that a message header has crossed all the channels of one dimension has been derived in [6]. We recollect briefly here the main equations for the calculation of ψ_j^t. The probability that there remains only one dimension to cross a message j-hops away from its destination, P_{φ_j}, can be written as

$$P_{\varphi_j} = \begin{cases} 2/(j+1) & \bar{k} \le r-j < r \\ 0 & 0 \le r-j < \bar{k} \end{cases} \quad (18)$$

Consequently, the probability that the header has entirely crossed t dimensions along on its j-hop path is given by

$$\psi_j^t = \begin{cases} 1-P_{\varphi_j} & t = 0 \\ P_{\varphi_j} & t = 1 \end{cases} \quad (19)$$

To determine the average time, $\bar{W}_s$, that a message sees in the source node before entering into the network, the injection channel is treated as an M/G/1 queue with a mean time waiting of [12]

$$\bar{W}_s = \rho \bar{S}(1 + C_{\bar{S}}^2)/(2(1-\rho)) \quad (20)$$

$$\rho = \lambda_g \bar{S} \quad (21)$$

$$C_{\bar{S}}^2 = \sigma_{\bar{S}}^2 / \bar{S}^2 \quad (22)$$

Where λ_g is the traffic rate on the network, $\bar{S}$ is the average service time, and $\sigma_{\bar{S}}^2$ is the variance of the service distribution. A message in the source node can enter the network through any of the V virtual channels. A regular message originating from a given source node that is j hops away from the hot-spot node sees a network latency of $\bar{S}_r$ (given by Eq. (9)), whereas a

hot-spot message sees a latency of $\bar{S}_\alpha$ (given by Eq. (13)) to reach the hot-spot node. So, the average latency taking into accounts, both regular and hot-spot messages is simply calculated by Eq. (2). Modelling the local queue in the source node that is *j*-hops away from the hot-spot node as an M/G/1 queue, and the average arrival rate on each virtual channel is λ_g/V and service time, $\bar{S}$, with an approximated variance $(\bar{S}-M-3\bar{d}+1)^2$ [17] yields the mean waiting time as

$$\bar{W}_s = \frac{\left(\lambda_g/V\right)\bar{S}^2\left(1+\left(\bar{S}-M-3\bar{d}+1\right)^2/\bar{S}^2\right)}{2\left(1-\left(\lambda_g/V\right)\bar{S}\right)} \qquad (23)$$

The probability, P_{v_j}, $(0 \le j \le V)$, that *v* virtual channels are busy at the physical channel that is *j* hops away from the hot-spot node, can be determined using a Markovian model (details of the model can be found in [18]). State ξ_{v_j} corresponds to *v* virtual channels being occupied. The transition rate out of state ξ_{v_j} to $\xi_{(v+1)_j}$ be λ_j, where λ_j is the traffic rate (given by Eq. (8)), while the transition rate out of ξ_{v_j} to $\xi_{(v-1)_j}$ be $1/\bar{S}_j$ ($\bar{S}_j$ given by Eq. (11)). In the steady state, the model yields the following probabilities [18]

$$Q_{0j} = 1 \qquad (24)$$

$$Q_{vj} = Q_{(v-1)j}\lambda_j\bar{S}_j \qquad (1 \le v \le V-1) \qquad (25)$$

$$Q_{Vj} = Q_{(V-1)j}\lambda_j/\left(1/\bar{S}_j - \lambda_j\right) \qquad (26)$$

$$P_{0j} = \left(\sum_{l=0}^{V} Q_{lj}\right)^{-1} \qquad (27)$$

$$P_{vj} = P_{(v-1)j}\lambda_j\bar{S}_j \qquad (1 \le v \le V-1) \qquad (28)$$

$$P_{Vj} = P_{(V-1)j}\lambda_j/\left(1/\bar{S}_j - \lambda_j\right) \qquad (29)$$

The average degree of virtual channels multiplexing located *j*-hops away from hot-spot can be found to be [18]

$$\bar{V}_j = \sum_{v=1}^{V} v^2 P_{v_j} \Big/ \sum_{v=1}^{V} v P_{v_j} \qquad (30)$$

And the average multiplexing rate through the network is given by

$$\bar{V} = \sum_{j=1}^{2(k-1)} p_{\alpha_j}\bar{V}_j \qquad (31)$$

Examining the above equations of the analytical model reveals that it is very difficult to give close-form solutions to the various variables of the model. Therefore, these equations are solved iteratively [12].

4. Model validation

In order to validate the proposed model and justify the applied approximations, the analytical model was simulated. For each simulation experiment, statistics were gathered for a total number of 100,000 messages. Statistic gathering was inhibited for the first 10,000 messages to avoid distortions due to the start-up transient. The results of simulation and analysis for an 8-ary 2-cube (N=64) with message length M=32 and 64 flits, hot-spot traffic fractions α=0.02, 0.3 and V=2, 6 virtual channels per physical channel are depicted in Fig. 2.

The figures reveal that the analytical model predicts the mean message latency with a good degree of accuracy in all regions. However, some discrepancies around the saturation point are apparent. This is a result of the approximations made when constructing the analytical model, e.g. the approximation used to estimate the variance of the service time distribution at a channel. This approximation greatly simplifies the model by avoiding the computation of the exact distribution of the message service time at a given channel.

5. Conclusions

Recently there has been renewed interest in Circuit Switching (CS) as an efficient switching method for supporting simplicity, reliability, availability, and Quality of Service (QoS) in peer-to-peer networks due to preserving both communication performance and fault-tolerant demands in such systems. Analytical models of fully adaptive routing have recently been proposed for CS in torus under uniform traffic. However, there has not been any analytical model of CS with non-uniform traffic such as hot-spot. This paper has presented a novel analytical model for the performance evaluation of CS in the torus under hot-spot traffic when fully adaptive routing and virtual channels flow control are used. Our next objective is to develop our modeling approach to consider the behavior of CS in the presence of faulty components.

References

[1] W.J. Dally and B. Towles, *Principles and practices of interconnection networks*, Morgan Kaufman Publishers, 2004.

[2] P. Mohapatra, Wormhole Routing Techniques for Directly Connected Multicomputer Systems, ACM Computing Surveys, Vol. 30, No. 3, September 1998.

[3] R.E. Kessler, J.L. Scwarzmeier, CRAY T3D: A new dimension for Cray Research, *in Proceedings COPMPCON*, pp. 176-182, spring 1993.

[4] Cray Research Inc., The Cray T3E scalable parallel processing system, on Cray's web page at http://www.cray.com/PUBLIC/product- info/T3E.

[5] P. Molinero-Fernandez, N. McKeown, The Performance of Circuit Switching in the Internet, *Journal of Optical Networking*, Vol. 2, pp. 82-96, 2003.

[6] G. Min, M. Ould-Khaoua, H. Sdazi-Azad, Communication Delay in Circuit-Switched Interconnection Networks, *IPCCC 2001*, pp. 51-56, 2001.

[7] L. Chlamtac, A. Ganz, M.G. Kienzle, A performance model of a connection-oriented hypercube interconnection system, Performance Evaluation, Vol. 25, No. 2, pp. 151-167, 1996.

[8] M. Colajanni, B. Ciciani, S. Tucci, Performance analysis of circuit-switching interconnection networks with deterministic and adaptive routing, *Performance Evaluation*, Vol. 34, No. 1, pp. 1-26, 1998.

[9] V. Sharma, EA. Varvarigos, Circuit switching with input queuing: an analysis for the *d*-dimensional wraparound mesh and the hypercube, *IEEE Trans. Parallel & Distributed systems*, Vol. 8, No. 4, pp.349-366, 1997.

[10] H. Sarbazi-Azad, L. M. Mackenzie , M. Ould-Khaoua, Hot Spot Analysis in Wormhole-routed Tori, *IPCCC 2000, Conference proceeding of the IEEE International*, pp.337-343, 2000.

[11] W.J. Dally, Performance analysis of k-ary n-cubes interconnection networks, *IEEE Trans. Computers,* Vol. 39, No.6, pp. 775-785, 1990.

[12] L. Kleinrock, *Queueing Systems*, Vol. 1, John Wiley, New York, 1975.

[13] G.J. Pfister, V.A. Norton, Hot spot contention and combining in multistage interconnection networks, *IEEE Trans. Computers,* Vol. 34, No. 10, pp. 943-948, 1985.

[14] M. Ould-Khaoua, A Performance model for Duato's adaptive routing algorithm in *k*-ary *n*-cubes, *IEEE Trans. Computers*, Vol. 48, No 12, pp. 1-8, 1999.

[15] A. Agarwal, Limits on interconnection network performance, *IEEE Trans. Parallel & Distributed Systems*, Vol. 2, No. 4, pp. 398-412, 1991.

[16] W. Feller, *An introduction to probability theory and its applications*, Vol. 1, John Wiley & Sons, New York, 1967.

[17] J. Draper, J. Ghosh, A comprehensive analytical model for wormhole routing in multicomputers systems, Journal of Parallel and Distributed Computing (JPDC), Vol. 32, pp. 202–214, 1994.

[18] W.J. Dally, Virtual channel flow control, *IEEE Trans. Parallel & Distributed Systems,* Vol. 3, No. 2, pp. 194-205, 1992.

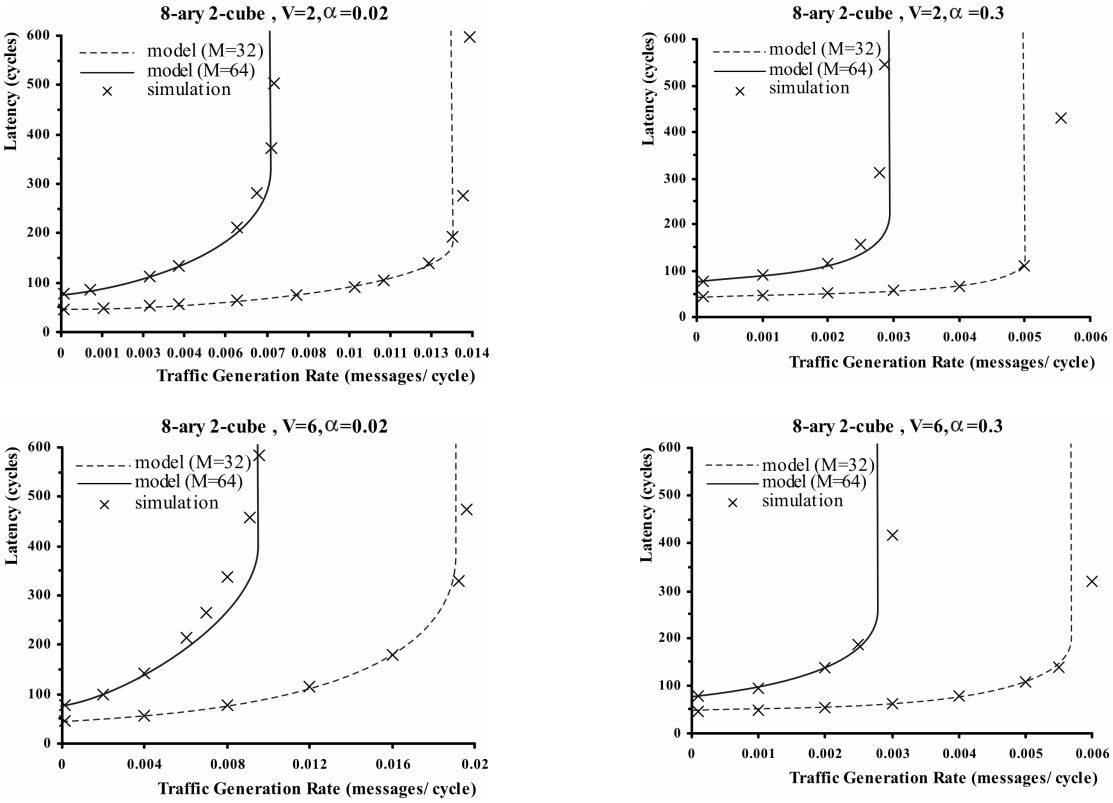

Fig. 2: Average message latency calculated by analytical model against those obtained from simulation for an 8×8 torus with *M*=32 and 64 flits, *V*=2, 6, and hot-spot traffic fractions α=0.02 and 0.3.