An Analytical Model of Fully-Adaptive Wormhole-Routed *k*-Ary *n*-Cubes in the Presence of Hot Spot Traffic

H. Sarbazi-Azad
Dept of Computing Science
University of Glasgow
Glasgow, G12 8QQ, U.K.

M. Ould-Khaoua
Dept of Computer Science
University of Strathclyde
Glasgow, G1 1HX, U.K.

L. M. Mackenzie
Dept of Computing Science
University of Glasgow
Glasgow, G12 8QQ, U.K.

Abstract

Several analytical models of fully-adaptive routing have recently been proposed for wormhole-routed k-ary n-cubes 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. As a result, most studies have resorted to simulation when evaluating the performance merits of adaptive routing. This paper describes the first analytical model of fully-adaptive wormhole routing in k-ary n-cubes in the presence of hot spot traffic. Results from simulation show close agreement with those predicted by the model.

1. Introduction

Most current multicomputers employ wormhole-routed *k*-ary *n*-cubes for low-latency and high-bandwidth inter-processor communication. The two most popular instances of *k*-ary *n*-cubes are the hypercube and torus. The former has been used in early systems, e.g. the Cosmic Cube [17] while the latter has been adopted in recent machines like the J-machine [14], and CRAY T3D [9].

Existing multicomputers [9, 13, 14] have widely adopted *deterministic routing* due to its simplicity and ease of implementation. However, messages cannot use alternative paths to avoid congested channels. *Fully-adaptive* routing overcomes the limitations of deterministic routing by enabling messages to explore all available paths. Several authors have proposed fully-adaptive routing algorithms that require a minimal number of virtual channels to ensure deadlock-freedom [8]. For example, the routing algorithm proposed by Duato [7] divides the virtual channels into two classes: *a* and *b*. At each routing step, a message visits adaptively any available virtual channel from class *a*. If all the virtual channels belonging to class *a* are busy, it visits a virtual channel from class *b* using deterministic routing.

Analytical models of deterministic routing in wormhole-routed *k*-ary *n*-cubes have been widely reported in the literature [1, 3, 4, 6]. Several researchers have recently proposed analytical models of fully-adaptive routing under the uniform traffic pattern [2, 12, 15]. Recent studies [8] have revealed that the performance advantages of adaptive routing over deterministic routing are more noticeable when traffic is non-uniform due, for example, to the presence of hot spots in the network [16]. To our best knowledge, no study has been reported in literature for modelling hot spots in wormhole-routed networks, and consequently most studies have resorted to simulation to evaluate the performance benefits of adaptive routing in the presence of hot spot traffic [8]. In an effort to fill this gap, this paper proposes the first analytical model for computing the mean message latency in *k*-ary *n*-cubes with fully-adaptive wormhole routing in the presence of hot spot traffic. The model is developed for Duato's fully-adaptive routing algorithm [7].

2. The Analytical Model

Details of the router structure used in the analysis can be found in [15]. The model uses assumptions that are widely used in the literature [1-6, 10, 15].

a) A generated message has a finite probability h of being directed to the hot spot node, and probability $(1-h)$ of being directed to the other nodes. Let us refer to these two types of messages as hot spot and regular messages respectively.

b) Nodes generate traffic independently of each other, and which follows a Poisson process with a mean rate of λ messages/cycle consisting of regular and hot spot portions of $h\lambda$ and $(1-h)\lambda$ respectively.

c) Message length is *M* flits, each of which is transmitted in one cycle from one node to the next.

d) The local queue in the source node has infinite capacity. Moreover, messages are transferred to the local node as soon as they arrive at their destinations.

e) *V* virtual channels are used per physical channel. In

Duato's algorithm [7, 8], class *a* contains $(V-2)$ virtual channels, which are crossed adaptively, and class *b* contains two virtual channel, which are crossed deterministically.

The mean message latency is composed of the mean network latency, $\overline{S}$, that is the time to cross the network, and the mean waiting time seen by a message in the source node, $\overline{W_s}$. However, to capture the effects of virtual channels multiplexing, the mean message latency has to be scaled by a factor, $\overline{V}$, representing the average degree of virtual channels multiplexing, that takes place at a given physical channel. Therefore, the mean message latency can be written as

$$Latency = (\overline{S} + \overline{W_s})\overline{V} \quad (1)$$

The regular and hot spot messages see different network latencies as they cross different number of channels to reach their destinations. If $\overline{S_r}$ and $\overline{S_h}$ denote the mean network latency for regular and hot spot messages respectively, the mean network latency taking into account both types of messages is given by

$$\overline{S} = (1-h)\overline{S_r} + h\overline{S_h} \quad (2)$$

The average number of hops that a regular message makes per dimension and in the network are given by [1]

$$\overline{k} = (k-1)/2 \quad (3)$$

$$\overline{d} = n\overline{k} \quad (4)$$

Fully-adaptive routing allows a message to use any channel that brings it closer to its destination, resulting in an even traffic rate of regular messages on network channels. A router in the *k*-ary *n*-cube has *n* output channels and the node generates, on average, $(1-h)\lambda$ regular messages in a cycle. Since each regular message travels, on average, $\overline{d}$ hops to cross the network, the rate of regular messages received by each channel, λ_r, can be written as [1]

$$\lambda_r = (1-h)\lambda\overline{d}/n \quad (5)$$

The network latency for a message consists of two parts: one is the delay due to the message transmission time, and the other is due to the blocking time in the network. As we shall see, the mean blocking time experienced by a regular message at a channel is the same across all the channels along its path. Since a regular message makes, on average, $\overline{d}$ hops to reach its destination, the mean network latency, $\overline{S_r}$, of a regular message can therefore be written as

$$\overline{S_r} = M + \overline{d} + \overline{d}B_r \quad (6)$$

where M is the message length and B_r is the mean blocking time seen by a regular message at a channel.

The hot spot traffic is not uniformly distributed across the network channels as channels located nearer to the hot spot node receive higher traffic rates than those further away. Consider a channel that is *j* hops away from the hot spot node, and let p_{h_j} be the probability that a hot spot message uses this channel to reach its destination, which is the hot spot node. Given that each of the *N* nodes generates, on average, $h\lambda$ hot spot messages in a cycle, the rate of hot spot traffic received by the channel, λ_{h_j}, is simply given by

$$\lambda_{h_j} = hN\lambda p_{h_j} \quad (7)$$

To compute p_{h_j} let us refer to the following result from the combinatorial theory.

Proposition [30]**:** *The number of ways to distribute r like objects into m different cells, such that no cell contains less than p objects and not more than* $p+q-1$ *objects is the coefficient of* x^{r-pm} *in the expansion of the polynomial* $(1-x^q)^m(1-x)^{-m} = (1+x+x^2+.....+x^{q-1})^m$.

Let us refer to the coefficient of x^{r-pm} as $N_p^{p+q-1}(r,m)$. In [30], its expression is given by

$$N_p^{p+q-1}(r,m) = \sum_{l=0}^{m}(-1)^l \binom{m}{l}\binom{r-mp-ql+m-1}{m-1} \quad (8)$$

If the hops made by a message are treated as like objects and the visited dimensions as different cells the above proposition can be used to compute the number of nodes that are *i* hops away from a given node in the *k*-ary *n*-cube. Given that an *i*-hop message may make from 0 up to (*k*-1) hops along each of the *n* dimensions, the number of nodes that are *i* hops away from a given node is simply

$$n_i = N_0^{k-1}(i,n) = \sum_{l=0}^{n}(-1)^l \binom{n}{l}\binom{i-lk+n-1}{n-1} \quad (9)$$

The probability, P_{h_j}, that a message has used during its network journey a particular channel, located *j* hops away from the hot spot node, can be derived as follows. Consider the set *J* of all the channels located *j* hops away from the hot spot node. Equation (9) gives the number of elements in the set *J* as

$$\|J\| = n_{j-1} \cdot n \quad (10)$$

Recalling that a *k*-ary *n*-cube has *N* nodes, the number of source nodes for which an element of *J* can act as intermediate channel to reach the hot spot node is $k^n - \sum_{l=0}^{j-1} n_l$. Therefore, p_{h_j} can be written as

$$p_{h_j} = \frac{N - \sum_{l=0}^{j-1} n_l}{N\|J\|} = \frac{\sum_{l=j}^{n(k-1)} n_l}{N\|J\|} \quad (11)$$

A hot spot message encounters different blocking times at different channels due to the non-uniform traffic rates on network channels caused by the hot spot traffic. A hot spot message may visit 1, 2, …, or $n(k-1)$ channels to reach the hot spot node. All these cases have to be taken into account when computing the mean network latency for hot spot messages. The network latency seen by a *j*-hop hot spot message is given by

$$S_{h_j} = M + j + \sum_{m=1}^{j} B_{h_{m,j}} \tag{12}$$

where $B_{h_{m,j}}, 1 \le j \le k(n-1),$ is the mean blocking time of the *j*-hop hot spot message at its *m*-th hop channel. Let θ_j be the probability that a node is *j* hops away from a given node (the hot spot node in our specific case). The number of nodes that are *j* hops away from a given node can be obtained using equation (9). Dividing the result over the total number of nodes yields θ_j as

$$\theta_j = n_j /(N-1) \tag{13}$$

Hence, the mean network latency seen by a hot spot message, $\overline{S_h}$, can be expressed by

$$\overline{S_h} = \sum_{j=1}^{n(k-1)} \theta_j S_{h_j} \tag{14}$$

A regular or hot spot message is blocked at a channel when all the adaptive virtual channels at the remaining dimensions to be visited and also the deterministic virtual channels at the lowest dimension still to be visited are busy. When blocking occurs a message has to wait for a deterministic virtual channel at the lowest dimension [7]. The mean waiting time depends on the location of the current channel relatively to the hot spot node as the traffic rate changes from one channel to the next.

A regular message makes, on average, $\bar{d}$ hops to cross the network. Suppose that the message has reached the m^{th}-hop $(1 \le m \le \bar{d})$ channel along its path. This channel can be between 1 and $n(k\text{-}1)$ hops away from the hot spot node. Let $\varphi_{j,l}$ denote the probability of blocking for a message which makes, on average, *l* hops per dimension- in case of a regular message *l* equals to $\bar{k}$ - when the current channel is *j* hops $(1 \le j \le n(k-1))$ away from the hot spot node. Moreover, let w_j denote the mean waiting time when blocking occurs at the channel. Since there are $n \cdot n_{j-1}$ channels (equation 10) that are *j* hops away from the hot spot node out of the total number of channels in the network *nN*, the mean blocking time, B_r, for a regular message can be written as

$$B_r = \sum_{j=1}^{n(k-1)} \frac{n_{j-1}}{N} \varphi_{j,\bar{k}} w_j \tag{15}$$

Since a regular message makes, on average, $\bar{k}$ hops per dimension, the probability of termination, $P_{t,\bar{k}}$, which is the probability that a message has crossed all the channels of a given dimension when it reaches a given router, is therefore given by [1]

$$P_{t,\bar{k}} = 1/\bar{k} \tag{16}$$

Hence, the probability that a message still has to visit, say, β out of the *n* dimensions, $\pi_{\beta,\bar{k}}$, can be written as

$$\pi_{\beta,\bar{k}} = \binom{n}{\beta} \left(1 - P_{t,\bar{k}}\right)^{\beta} P_{t,\bar{k}}^{n-\beta} \tag{17}$$

When β dimensions are still to be visited, a regular message can select any one of the (*V*-2) adaptive virtual channels belonging to the (β-1) dimensions still to be visited adaptively. It can also use one of the two deterministic channels and any one of the (*V*-2) adaptive virtual channels at the lowest dimension to be visited according to deterministic routing [7]. Using equation 17 we can express $\varphi_{j,\bar{k}}$ as

$$\varphi_{j,\bar{k}} = \sum_{\beta=1}^{n} \pi_{\beta,\bar{k}} p_{a_j}^{\beta-1} p_{d_j} \tag{18}$$

with p_{a_j} being the probability that all adaptive virtual channels at a physical channel located *j* hops away from the hot spot node are busy and p_{d_j} being the probability that all adaptive and deterministic virtual channels are busy. Adapting the same approach used in [15], p_{a_j} and p_{d_j} can be expressed as

$$p_{a_j} = p_{V_j} + \frac{2\,p_{(V-1)_j}}{\binom{V}{V-1}} + \frac{p_{(V-2)_j}}{\binom{V}{V-2}} \tag{19}$$

$$p_{d_j} = p_{V_j} + \frac{2\,p_{(V-1)_j}}{\binom{V}{V-1}} \tag{20}$$

where p_{v_j} represent the probability that *v* virtual channels are busy at a physical channel located *j* hops away from the hot spot node (p_{v_j} is determined below).

To determine the mean waiting time, w_j, to acquire a virtual channel a physical channel is treated as an M/G/1 queue with a mean waiting time of [11]

$$w_j = \frac{\rho_j S_j (1+C_{S_j}^2)}{2(1-\rho_j)} \quad (21)$$

$$\rho_j = \lambda_j S_j \quad (22)$$

$$C_{S_j}^2 = \sigma_{S_j}^2 / S_j^2 \quad (23)$$

where λ_j is the traffic rate on the channel located *j* hops away from the source node, S_j is its service time, and $\sigma_{S_j}^2$ is the variance of the service time distribution. The rate of messages arriving at the channel is composed of regular and hot spot messages, and is equal to

$$\lambda_j = \lambda_r + \lambda_{h_j} \quad (24)$$

Let us assume for a moment that there is no hot spot traffic present in the network. Since adaptive routing distributes regular traffic evenly across the network channels, the mean service time for regular messages is the same across all channels and is approximately equal to the mean network latency, $\overline{S_r}$ [2, 15]. The presence of hot spot traffic, however, causes the service time to vary from one channel to another due to the non-uniformity of traffic rates on the channels. When a message reaches a channel that is *j* hops away form the hot spot node, the mean service time considering both regular and hot spot message can be written as

$$S_j = (\lambda_r / \lambda_j)\overline{S_r} + (\lambda_{h_j} / \lambda_j)\overline{S_{h_j}} \quad (25)$$

where $\overline{S_{h_j}}$ is the mean latency seen by a hot spot message to cross from a channel located *j* hop away from the hot spot node to the hot spot node itself. The expression of $\overline{S_{h_j}}$ is given by

$$\overline{S_{h_j}} = M + \sum_{i=j}^{n(k-1)} \left(\xi_{j,i} \sum_{l=1}^{j-1} B_{h_{l+i-j,i}} \right) \quad (26)$$

with $\xi_{j,i}$ being the ratio of the number of nodes which are *j* hops away from the hot spot node to the number of nodes that are i, $j \le i \le n(k-1)$, hops away from the hot spot node. Using equation (9) we can write $\xi_{j,i}$ as

$$\xi_{j,i} = n_i / \sum_{m=j}^{n(k-1)} n_m \quad (27)$$

Since the minimum service time at a channel is equal to the message length, *M*, following a suggestion of [6], the variance of the service time distribution can be approximated as

$$\sigma_{S_j}^2 = (S_j - M)^2 \quad (28)$$

As a result, the mean waiting time becomes

$$w_j = \frac{\lambda_j S_j^2}{2(1-\lambda_j S_j)} \left[1 + \frac{(S_j - M)^2}{S_j^2} \right] \quad (29)$$

The possible number of hops made along a given dimension is between 0 and $(k-1)$. Equation (9) can be used to determine the number of ways to distribute the hops made by the hot spot message along the *n* dimensions. However, instead of considering all the possible combinations when distributing the hops among the *n* dimensions, we consider only the case where the message makes an equal number of hops per dimension. This greatly simplifies the computation of the mean blocking time especially when *n* and *k* are large due to the large number of combinations that has to be considered. Using this approximation, the calculation of the blocking time for regular messages can be adapted for hot spot messages. When the message reaches the *m*-th hop channel, it is (*j*-*m*+1) hops away from the hot spot node. The mean blocking time is given by

$$B_{h_{m,j}} = \varphi_{j-m+1,\, j/n} w_{j-m+1} \quad (30)$$

The new expressions for the probability of blocking, $\varphi_{j-m+1,\, j/n}$, and mean waiting time, w_{j-m+1}, for hot spot messages can be obtained by simply substituting *j* by (*j*-*m*+1) and $\bar{k}$ by j/n in the above equations (16)-(29).

A regular message originating from a source node that is *j* hops away from the hot spot node sees a network latency of $\overline{S_r}$ (equation 6), whereas a hot spot message sees a latency of S_{h_j} (equation 12) to reach the hot spot node. Therefore, the mean network latency for a message that originates at a source node that is located *j* hops away from the hot spot node, S_{s_j}, taking into account both regular and hot spot messages with their appropriate weights is simply given by

$$S_{s_j} = (1-h)\overline{S_r} + hS_{h_j} \quad (31)$$

Modelling the local queue in the source node, located *j* hops away from hot spot node, as an M/G/1 queue with the mean arrival rate λ/V and service time S_{s_j} with an approximated variance $(S_{s_j} - M)^2$ yields the mean waiting time as

$$W_{s_j} = \frac{\frac{\lambda}{V} S_{s_j}^2}{2(1-\frac{\lambda}{V} S_{s_j})} \left[1 + \frac{(S_{s_j} - M)^2}{S_{s_j}^2} \right] \quad (32)$$

Averaging over all possible values of *j* gives the mean waiting time in a source node as

$$\overline{W_s} = \sum_{j=1}^{n(k-1)} \theta_j W_{s_j} \quad (33)$$

The probability, p_{v_j}, that v virtual channels are busy at a physical channel located *j* hops away from the hot spot node, can be determined using a Markovian model, yielding the following steady-state solution (see [5, 15] for details)

$$Q_{v_j} = \begin{cases} \lambda_j{}^v S_j{}^v & 0 \le v < V \\ \lambda_j{}^V S_j{}^V / (1 - \lambda_j S_j) & v = V \end{cases} \quad (34)$$

$$P_{v_j} = \begin{cases} 1 / \sum_{i=0}^{V} Q_{i_j} & v = 0 \\ P_{0_j} Q_{v_j} & 1 \le v \le V \end{cases} \quad (35)$$

Averaging over all the possible values of *j* gives the average degree of multiplexing of virtual channels at a given physical channel as [5]

$$\overline{V} = \sum_{j=1}^{n(k-1)} \theta_j \frac{\sum_{v=1}^{V} v^2 P_{v_j}}{\sum_{v=1}^{V} v P_{v_j}} \quad (36)$$

Examining the above equations reveals that there are several inter-dependencies between the different variables of the model. For instance, Equation (25) and (26) reveal that S_j is a function of $B_{h_{m,j}}$ while equations (29) and (30) show that $B_{h_{m,j}}$ is a function of S_j. Given that closed-form solutions to such inter-dependencies are very difficult to be determined the different variables of the model are computed using iterative techniques for solving equations.

3. Model Validation

Figure 1 depicts latency results predicted by the above model plotted against those provided by the simulator for the following cases only: networks size $N = 8^2$ and $N = 8^3$ nodes; message length M=32 and 64 flits; number of virtual channels V=3 and 5; fractions of hot spot traffic are $h = 0.08$, 0.16 and 0.24. The results reveal that the analytical model predicts the mean message latency with a reasonable degree of accuracy when the network is in the steady state regions, that is when it has not reached the saturation point. However, there are discrepancies in the results provided by the model and simulation when the network is under heavy traffic and approaches the saturation point. This is due to the approximations that have been made in the analysis to ease the model development (e.g. equation 28). Nevertheless, it can be concluded that the model produces latency results with a good degree of accuracy in the regions of interests and its simplicity makes it a practical evaluation tool to study the performance of fully-adaptive routing in *k*-ary *n*-cubes in the presence of hot spot traffic.

4. Conclusions

Several analytical models of fully-adaptive routing have recently been proposed for wormhole-routed *k*-ary *n*-cubes under the uniform traffic pattern. This paper has presented the first analytical model to compute the mean message latency in the presence of hot spot traffic in wormhole-routed *k*-ary *n*-cubes with fully-adaptive routing algorithm. Simulation experiments have revealed that the analytical model produces latency results that are in a good agreement with those produced through simulation.

References

[1] A. Agarwal, Limits on interconnection network performance, *IEEE TPDS* (2)4, pp. 398-412, 1991.

[2] Y. Boura, C.R. Das, T.M. Jacob, A performance model for adaptive routing in hypercubes, *Proc. Int. Workshop Parallel Processing*, pp. 11-16, Dec. 1994.

[3] B. Ciciani, M. Colajanni, C. Paolucci , An accurate model for the performance analysis of deterministic wormhole routing, *Proc. IPSS'97*, pp. 353-359, 1997.

[4] W.J. Dally, Performance analysis of *k*-ary *n*-cubes interconnection networks, *IEEE TC* 39(6),1990.

[5] W.J. Dally, Virtual channel flow control, *IEEE TPDS* 3(2), pp. 194-205, 1992.

[6] J.T. Draper, J. Ghosh, A comprehensive analytical model for wormhole routing in multicomputer systems, *JPDC* 32, pp. 202-214, 1994.

[7] J. Duato, A new theory of deadlock-free adaptive routing in wormhole routing networks, *IEEE TPDS* 4(12), pp. 320-1331, 1993.

[8] J. Duato, S. Yalamanchili, L. Ni, Interconnection networks: An engineering approach, IEEE Computer Soc.Press, 1997.

[9] R.E. Kessler, J.L. Schwarzmeier, CRAY T3D: A new dimension for Cray Research, *in CompCon*, 1993, 176-182.

[10] J. Kim, C.R. Das, Hypercube communication delay with wormhole routing, *IEEE TC* C-43(7), pp. 806-814, 1994.

[11] L. Kleinrock, Queueing Systems Vol. 1, John Wiley, 1975.

[12] S. Loucif, M. Ould-Khaoua, L.M. Mackenzie, Analysis of fully-adaptive wormhole routing in tori, *Parallel Computing* 25(12), pp. 1477-1487, 1999.

[13] N-Cube Systems, N-cube Handbook, N-Cube, 1986.

[14] M. Noakes, W.J. Dally, System design of the J-machine, *Proc. Advanced.Research in VLSI*, MIT Press, 1990.

[15] M. Ould-Khaoua, An analytical model of Duato's adaptive routing algorithm, *IEEE TC* 48(12), pp. 1-8, 1999.

[16] G.J. Pfister, V.A. Norton, Hot spot contention and combining in multistage interconnection networks, *IEEE TC* 34(10), pp. 943-948, 1985.

[17] C.L. Seitz, The Cosmic Cube, *CACM* 28, pp. 22-33, 1985.

[18] W.A. Whiteworth, Choice and Chance, Cambridge Univ. Press, 1901.

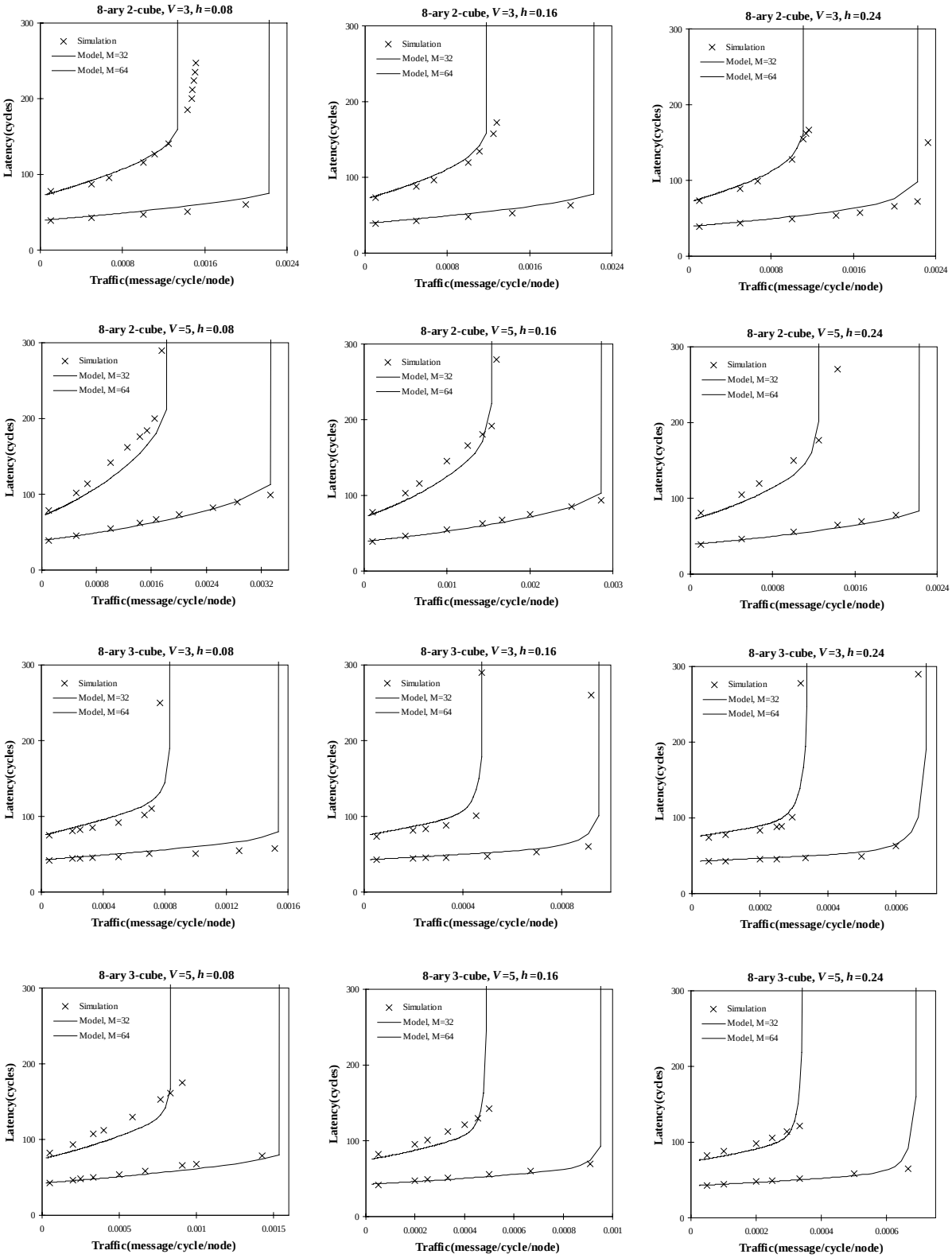

Fig. 1- Latency predicted by the model and simulation in the 8-ary 2-cube and 8-ary 3-cube. Message length *M*=32 and 64, hot spot portion *h*=0.08, 0.16 and 0.24, and number of virtual channels *V*=3 and 5.