.

An Analytical Model of Adaptive Wormhole Routing with Time-out

A. Khonsari, H. Sarbazi-Azad, M. Ould-Khaoua

Department of Computing Science,
University of Glasgow,
Glasgow, G12 8QQ, U.K.
{ak, hsa, mohamed}@dcs.gla.ac.uk

Abstract

Several studies have shown that the performance advantages of adaptive routing over deterministic routing are reduced when the traffic contains strong degree of communication locality. This paper proposes a new analytical model of an adaptive routing algorithm proposed by Duato in [Dua94]. The main feature of this algorithm is the use of a time-out selection function for assigning virtual channels. This has the advantage of reducing virtual channels multiplexing to improve the network performance, especially, in the presence of communication locality. Simulation experiments reveal that the proposed model predicts message latency with a good degree of accuracy under different traffic conditions.

1 Introduction

The success of large-scale multicomputers is highly dependent on the efficiency of their underlying interconnection networks. The hypercube has been one of the most popular networks for practical multicomputers due to its desirable properties, including regularity, symmetry, low diameter and high connectivity. The iPSC/860 [Van94], iPSC/2 [Nug88] and SGI Origin 2000 [Lau97] are examples of commercial systems that are based on the hypercube.

Existing multicomputers [And97, Fil97, Lau97, Van94] have widely used wormhole routing [Dal90]. This is due to its low buffering requirement, and more importantly it makes latency independent of the message distance under light traffic loads. In wormhole routing, a message is divided into flits (a few bytes each) that form the smallest unit of information on which transfer and flow control is performed. The header flit establishes a path as it advances through the network, and the remaining data flits follow it in a pipelined fashion. When the header is blocked, the data flits remain spread across several routers, preventing other messages from using the channels that they occupy.

Since wormhole routing relies on a blocking mechanism for flow control, deadlock can occur because of cyclic dependencies over network resources (i.e., channels or buffers) during message routing. The provision of deadlock-free routing in wormhole-routed networks has been a major issue over the past decade, e.g. [Dua93, Dua97, Lin93, Su93]. Deadlock-avoidance is generally ensured by dividing each physical channel into several *virtual channels* and imposing certain restrictions on the way messages visit the 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 manner [Dal92]. A typical example of a deadlock-free routing widely used in practice is deterministic routing where messages visit dimensions in a predefined order. Deterministic routing has been popular in multicomputers [And97, Fil97, N-C90, Nug88, Van94] because of its minimal hardware requirements in term of virtual channels, allowing the design of simple and fast routers [Dua97]. However, this form of routing cannot efficiently handle non-uniform traffic patterns, as messages cannot avoid congested network regions. To overcome this, many adaptive routing algorithms have been proposed in the literature [Dua93, Dua94, Dua97, Lin93, Su93], where all available paths between two source and destination nodes may be taken to reduce message latency.

The authors in [Dua93, Lin93, Su93] have proposed adaptive routing algorithms, which can achieve deadlock freedom using a minimal number of virtual channels. Their proposed algorithms require only one extra virtual channel per physical channel, compared to deterministic routing, allowing for an efficient router implementation. For instance, Duato's algorithm [Dua93] divides the virtual channels associated with each physical channel 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. Duato's algorithm requires at least two virtual channels per physical channel to ensure deadlock-freedom in the hypercube (i.e., class *a* and *b* contains one virtual channel each). When there are more than two virtual channels, network performance is maximised when the extra virtual channels are added to class *a* [Dua93, Dua97]. Thus, when V virtual channels are used per physical channel in the hypercube, the best performance is achieved when class *a* and *b* contains (V-1) and one virtual channels, respectively. The first multicomputers and routers that use adaptive routing have recently been reported. The Cray T3E [And97] and Reliable Router [Dal94] are two examples of practical systems that have adopted Duato's adaptive routing algorithm.

Duato [Dua93, Dua97] has performed an extensive evaluation of his adaptive routing algorithm on the hypercube. The results of his studies have revealed that good performance levels are achieved compared to deterministic routing when the traffic was uniform. However, adaptive routing performed poorly when the traffic exhibited strong communication locality. Duato [Dua94] has attributed this performance degradation in the presence of communication locality to the overhead associated with virtual channels multiplexing. When traffic is local, messages make a few hops to reach their destinations, and therefore cannot take full advantage of adaptive routing to avoid congested channels since they can select among a few number of channels at an intermediate router. Moreover, the multiplexing of virtual channels causes an increase in message latency since messages are forced to share the bandwidth of the physical channels.

To reduce the effects of virtual channels multiplexing, Duato [Dua94] has introduced to his original adaptive routing algorithm described in [Dua93] a time-out mechanism when selecting a particular class of virtual channels. When a message is blocked upon reaching a given router, it waits for a fixed time period for one of the virtual channels belonging to class *a* that brings it closer to its destination. If it does not succeed, it waits for a virtual channel belonging to class *b*. Using the time-out mechanism as suggested in [Dua94] reduces the overhead due to virtual channels multiplexing. Simulation experiments reported in [Dua94] have confirmed that the new resulting adaptive routing algorithm leads to good performance improvement, especially when traffic contains a strong degree of communication locality (please see [Dua94] for more details on the algorithm and the simulation study).

Analytical models are cost-effective and versatile tools for evaluating network performance under different working conditions. Their significant advantage over simulation is that the analytical models can be used to obtain performance results for large systems, which are infeasible by simulation due to the excessive computation demands on conventional computers. Analytical models for deterministic routing in wormhole-routed networks have been widely reported in the literature [Aga91, Cic97, Dal90, Dra94, Kim94]. Although many adaptive routing algorithms have been proposed for wormhole-routed networks, it is necessary to have clear understanding of the factors that affect their potential performance before they can be widely adopted in commercial multicomputers. Except from the models suggested in [Bou94, Oul99] for the simple version of Duato's algorithm (i.e., with no time-out mechanism) [Dua93], there has not been any model proposed in the literature for any other adaptive routing algorithms, e.g. that of [Dua94, Lin93, Su93]. As a result, most existing studies [Lin93, Su93], including Duato's study of the time-out mechanism [Dua94], have resorted to software simulation to evaluate the performance merits of adaptive routing algorithms.

This paper presents a new analytical model to calculate the message latency in wormhole-routed hypercube with Duato's adaptive routing algorithm, augmented with the time-out mechanism [Dua94]. The derivation of the model is discussed for the hypercube. The main reason for choosing the hypercube is that all Duato's studies [Dua93, Dua94] have been conducted on this network. One of the important features of the proposed model is the use of results from queueing systems with impatient customers to capture the effects of the time-out mechanism on network performance. The validity of the model is demonstrated by comparing analytical results with those obtained through simulation experiments. The remainder of the paper is organised as follows. Section 2 describes the analytical model. Section 3 validates the model through simulation experiments. Finally, Section 4 concludes this study.

2 The Analytical Model

In this section, we describe the node organisation in the hypercube, assumptions, and communication model used in the analysis. We then present the analytical model.

2.1 Node Organisation

An *n*-dimensional hypercube consists of 2^n nodes, each addressed by a n-bit binary number from 0 to $2^n - 1$. Each node has exactly n neighbours. Two nodes $v = v_0 v_1 ... v_{i-1} v_i v_{i+1} ... v_{n-1}$ and $v' = v'_0 v'_1 ... v'_{i-1} v'_i v'_{i+1} ... v'_{n-1}$, $v_i, v'_i \in \{0,1\}$, are connected if and only if there is an i such that $v_i = v'_i \pm 1$ and $v_j = v'_j$ for all $j \neq i$. Each node consists of a processing element (PE) and a router. The PE contains a processor and some local memory. The router has $(n+1)$ input and $(n+1)$ output channels. Each node is connected to its n neighbouring nodes through n input and n output channels. The remaining input and output channels are used by the PE to inject/eject, respectively, messages to/from the network. The router contains flit buffers for each virtual channel. The input and output channels are connected by a $(n+1)V$-way crossbar switch (V is the number of virtual channels per physical channel) that is capable of simultaneously connecting multiple input channels to multiple output channels when there is no contention.

2.2 Assumptions

The model is based on the following assumptions, which are commonly accepted in the literature [Aga91, Bou94, Cic97, Dal90, Dal92, Dra94, Kim94, Oul99].

a) Nodes generate traffic independently of each other, and which follows a Poisson process with a mean rate of λ_g messages/cycle.
b) The message length is exponentially distributed with a mean length of M flits, each requiring one-cycle transmission time from one router to the next.
c) The local queue at the injection channel in the source node has infinite capacity. Moreover, messages are transferred to the local PE through the ejection channel as soon as they arrive at their destinations.
d) V $(V \geq 2)$ virtual channels are used per physical channel. In Duato's routing algorithm [Dua93, Dua94], class a contains $(V-1)$ virtual channels, which are crossed adaptively, and class b contains one virtual channels, which is crossed deterministically (e.g. in an increasing order of dimensions). Let the virtual channels belonging to class a and b be called the adaptive and deterministic virtual channels respectively. At a given routing step, a message chooses randomly one of the available adaptive virtual channels at one of the physical channels, if available, that brings it closer to its destination. There is no distinction between virtual channels when computing the different probabilities in order to simplify the analysis [Bou94, Oul99].
e) When a message finds all the required adaptive virtual channels busy upon reaching an intermediate router, it can wait up to τ cycles (the fixed time-out period) for one of the adaptive virtual channel to become free again. Otherwise, it suffers time-out, and as a result the message has to wait for the deterministic virtual channel corresponding to the lowest dimension still to be crossed according to deterministic routing [Dua93]. It is assumed that the probability of time-out at a given channel is independent of the subsequent channels.

2.3 The Communication Model

Even though the proposed model can deal with different traffic patterns (e.g., uniform and non-uniform), the present discussion focuses on the case where the traffic contains communication locality; where the likelihood of communication to different nodes decreases with distance. Because Duato's study [Dua94] has revealed that the performance advantages of adding a time-out mechanism are more noticeable when traffic contains a strong degree of communication locality.

The traffic pattern affects mainly the mean message distance, $\bar{d}$, which is the expected number of hops that a message makes to reach its destination. The mean message distance is generally given by

$$\bar{d} = \sum_{i=1}^{n} i p_i \tag{1}$$

where p_i is the probability of a message crossing i channels to reach its destination in a n-dimensional hypercube. Different choices of p_i lead to different distributions for message destinations, and consequently to different mean message distances. The following analysis uses the decreasing probability routing distribution defined in [Ree87] as a model of communication locality (it is worth noting that our modelling approach presented here can be equally apply to the other models discussed in [Ree87]). In this model, the probability, p_i, of sending a message to a particular destination node i hops away decreases with the distance i. For an n-dimensional hypercube, the probabilities p_i $(1 \le i \le n)$ can be defined as

$$p_i = \theta(\alpha, n)\alpha^i \tag{2}$$

where α is between 0 and 1 leading to varying degrees of communication locality. As α approaches zero the degree of locality increases, while as α approaches 1, the traffic become more uniform. The factor $\theta(\alpha, n)$ is a normalising constant, and is chosen such that

$$\theta(\alpha, n)\sum_{i=1}^{n}\alpha^i = 1 \tag{3}$$

From the above equation, we can easily determine $\theta(\alpha, n)$. Substituting the expressions of $\theta(\alpha, n)$ in equations 1 and 2 yields the probabilities p_i and the mean message distance, $\bar{d}$, as

$$p_i = \frac{(\alpha - 1)\alpha^{i-1}}{\alpha^n - 1} \tag{4}$$

$$\bar{d} = \frac{(n\alpha - n - 1)\alpha^n + 1}{(\alpha - 1)(\alpha^n - 1)} \tag{5}$$

2.4 Outline of the Model

The mean message latency is composed of the mean network latency, $\bar{S}$, that is the time to cross the network, and the mean waiting time seen by a message in the source node, $\bar{w}_s$. However, since virtual channels share the bandwidth of the physical channels, the analytical model needs to capture the multiplexing effects on message latency. Before showing how the model can include this effect, let us first calculate the two components of the message latency: $\bar{S}$ and $\bar{w}_s$.

Adaptive routing provides multiple paths that a message can take to cross the network. The number of these alternative paths depends on the number of the remaining hops to reach the destination node. Therefore, the model determines the mean network latency S_i $(1 \le i \le n)$ for an i-hop message, i.e., a message that needs to make i hops to cross from source to destination, and then determines the overall mean network latency, $\bar{S}$, by averaging over all the possible values of i. S_i is composed of the actual message transmission time, and the blocking times to acquire the required virtual channels along the message path. Therefore, we can write

$$S_i = (M + i) + \sum_{j=1}^{i} B_{i,j} \tag{6}$$

where M is the message length and $B_{i,j}$ is the mean blocking time experienced by an i-hop message at the j^{th} channel $(1 \le j \le i)$. Since the probability of generating an i-

hop message in the hypercube is p_i, averaging over all the possible hops made by a message, we obtain $\bar{S}$ as

$$\bar{S} = \sum_{i=1}^{n} p_i S_i \tag{7}$$

Let us now show how to compute the mean blocking time, B_j, seen by the i-hop message at the j^{th} channel along its path. When the message reaches the j^{th} channel, it has $(i-j+1)$ remaining channels to cross to reach its destination. To make the next hop, the message can use any of $(V-1)(i-j+1)$ adaptive virtual channels and one deterministic virtual channel. Therefore, the message sees a different probability of blocking at each hop as the number of alternative virtual channels that it can select changes from one hop to another. However, due to the fact that the traffic rates are equal across the network channels, and due to the symmetry of the hypercube, a message sees the same mean waiting time across all the visited channels [Oul99]. A message is blocked at a channel when all the adaptive virtual channels of the remaining dimensions to be visited and also the deterministic virtual channels of the lowest dimension still to be visited are busy. If P_l denotes the probability that l virtual channels are busy, the probability of blocking at adaptive and deterministic channel are respectively approximated by [Bou94]

$$P_{a_{i,j}} = \left(P_V + \frac{P_{V-1}}{\binom{V}{V-1}} \right)^{(i-j+1)} \tag{8}$$

$$P_d = P_V \tag{9}$$

The probability P_l $(0 \le l \le V)$ that l virtual channels at a given physical channel are busy can be determined using a Markovian model (details of the model can found in [Dal92, Oul99]). In the steady state, the model yields the following probabilities

$$P_l = \begin{cases} 1/\sum_{l=0}^{V} q_l & l = 0 \\ P_{l-1}\lambda_c \bar{S} & 0 < l < V \\ P_{l-1}\dfrac{\lambda_c}{1/\bar{S} - \lambda_c} & l = V \end{cases},$$

$$q_l = \begin{cases} 1 & l = 0 \\ q_{l-1}\lambda_c \bar{S} & 0 < l < V \\ q_{l-1}\dfrac{\lambda_c}{1/\bar{S} - \lambda_c} & l = V \end{cases} \tag{10}$$

The set of messages waiting for the adaptive virtual channels and the set of those waiting for the deterministic virtual channel form two separate queues at a given physical channel. We first calculate the mean waiting time at each adaptive channel, $\overline{W_a}$. To capture the effects of time-out when determining the mean waiting time, $\overline{W_a}$, and the probability of time-out, P_t, we use theoretical results of queueing systems with *impatient* customers and *deterministic* impatience time [Dal65, Tij86]; customers that do not receive service within a fixed time period, τ , leave the system. We have found that using M/M/1 queues with deterministic impatient time, suggested in [Tij86], enable us to obtain a simple and practical model that exhibits a reasonable degree of accuracy in predicting message latency (as will be shown later in Section 3). To derive the mean waiting time and probability of time-out the mean arrival rate and service time at a channel have to be determined first.

Adaptive routing allows a message to cross channels in any order that brings it closer to its destination, resulting in an equal and balanced traffic load on all network channels. A message crosses, on average, $\bar{d}$ hops to reach its destination. Since a router has n output channels and the local node generates, on average, λ_g messages in a cycle the rate of messages received by each channel, λ_c , is given by

$$\lambda_c = \lambda_g \frac{\bar{d}}{n} \tag{11}$$

Due to the symmetry of the hypercube topology and uniformity of traffic on network channels, the mean service time seen by a message at each channel is the same across all network channels [Oul99] and can be approximated by the mean network latency, $\bar{S}$ (equation 7). Treating the adaptive virtual channels as an M/M/1 queue of impatient customers with deterministic time-out, and with the mean arrival rate λ_c and service time $\bar{S}$, yields the mean waiting time and probability of suffering time-out seen by a message at an adaptive virtual channel as (see [Tij86] for more details)

$$\bar{w}_a = \frac{\dfrac{\lambda_c \bar{S}^2}{(1-\lambda_c \bar{S})} - \left(\dfrac{\bar{S}}{1-\lambda_c \bar{S}} + \lambda_c \bar{S}\tau \right) e^{\frac{-(1-\lambda_c \bar{S})\tau}{\bar{S}_a}} - P_t \tau}{(1-P_t)\left(1-\lambda_c^2 \bar{S}^2 e^{\frac{-(1-\lambda_c \bar{S})\tau}{\bar{S}_a}} \right)} \tag{12}$$

$$P_t = \frac{(1-\lambda_c \bar{S})\lambda_c \bar{S} e^{\frac{-(1-\lambda_c \bar{S})\tau}{\bar{S}_a}}}{1-\lambda_c^2 \bar{S}^2 e^{\frac{-(1-\lambda_c \bar{S})\tau}{\bar{S}_a}}} \tag{13}$$

In the event of blocking a message has to wait for one of the adaptive virtual channels for τ cycles to become free again. If it cannot get access within the time-out period, it suffers time-out, and has to wait for the deterministic virtual channel at the lowest dimension [Dua94]. Assume that the probability of time-out at each physical channel is P_t. Since a message crosses, on average, $\bar{d}$ routers (or dimension), the mean probability of suffering time-out at adaptive channels of a given router, P_{tr} , is computed by averaging over the probability of time-out of a message that still has to cross i $(1 \le i \le \bar{d})$ dimensions to reach to its destination and is given by

$$P_{tr} = \frac{P_t \sum_{i=1}^{\bar{d}} \left(P_V + \frac{P_{V-1}}{\binom{V}{V-1}} \right)^i}{\bar{d}} \quad (14)$$

Messages that suffer time-out at each adaptive channel use the deterministic virtual channel at the lowest dimension. Therefore the rate of messages that use the deterministic virtual channel at the lowest dimension is calculated by $P_{tr}\lambda_c$. Modelling the deterministic channel as an M/M/1 queue, with a mean arrival rate $P_{tr}\lambda_c$ and service time $\bar{S}$, yields the mean waiting time as [Kle75]

$$\bar{w}_d = \frac{P_{tr}\lambda_c \bar{S}}{\frac{1}{\bar{S}}\left(1 - P_{tr}\lambda_c \bar{S}\right)} \quad (15)$$

Now, averaging over the waiting times seen by a message at one of the adaptive virtual channels (given equation 12) and at the deterministic virtual channel (equation 15), yields the mean blocking time at a physical channel as

$$B_{i,j} = (1 - P_{tr}) P_{a_{i,j}} \bar{w}_a + P_{tr} \, (\tau + P_d \, \bar{w}_d \,) \quad (16)$$

The effects of queueing that occur in the source node must also be included. A message in the source node can enter the network through any of the V virtual channels. Moreover, the message sees a mean service time $\bar{S}$ (equation 7) to cross the network. Modelling the local queue in the source node as an M/M/1 queue with a mean arrival rate λ_g / V and mean service time $\bar{S}$ yields the mean waiting time, $\bar{w}_s$, experienced by a message in the source node as [Kle75]

$$\bar{w}_s = \frac{\frac{\lambda_g}{V} \bar{S}}{\frac{1}{\bar{S}}\left(1 - \frac{\lambda_g}{V} \bar{S}\right)} \quad (17)$$

Having determined the mean network latency, $\bar{S}$ (given by equation 7), and the mean waiting time in the source node, $\bar{w}_s$ (given by equation 17), we need now to include the effects of virtual channels multiplexing in order to complete the development of our model. Since the virtual channels are time-multiplexed to share the bandwidth of the physical channel, there is interaction between the virtual channels. This can be easily achieved, as discussed in [Bou94, Oul99], by scaling the mean waiting time seen by a message at a given queue by a factor, $\bar{V}_V$, representing the average degree of V virtual channels multiplexing, that takes place at a physical channel. The factor $\bar{V}_V$ can be estimated using the following formula [Dal92]

$$\bar{V}_V = \frac{\sum_{l=0}^{V} l^2 P_l}{\sum_{l=0}^{V} l P_l} \quad (18)$$

where P_l $(0 \le l \le V)$ is the probability that l virtual channels at a given physical channel are busy (and is computed using equation 10).

In the event of occurring timeout with probability, P_{tr}, the bandwidth of a physical channel is multiplexed among $(V-1)$ adaptive and one deterministic virtual channels (i.e. V virtual channels). Otherwise, the bandwidth is multiplexed only among $(V-1)$ adaptive virtual channels. If $\overline{V}_{V-1}$ denotes the average degree of multiplexing of $(V-1)$ virtual channels, Combining equations 7, 17, and 18 yields the mean message latency as [Bou94, Nug88].

$$Latency = \overline{w}_s \overline{V}_V + \overline{S}(\overline{V}_V P_{tr} + \overline{V}_{V-1}(1-P_{tr})) \qquad (19)$$

3 Validation of the Model

The above model has been validated through a discrete-event simulator that performs a time-step simulation of the network operations at the flit level. Each simulation experiment was run until the network reached its steady state, that is, until a further increase in simulated network cycles does not change the collected statistics appreciably. Extensive validation experiments have been performed for several combinations of network sizes, message lengths, and virtual channels, and the general conclusions have been found to be consistent across all the cases considered. However, for the sake of specific illustrations, results predicted by the above model plotted against those provided by the simulator for the following cases only (it is worth noting that most of the values selected for the following parameters have also been used in Duato's study [Dua94]).

- Network size is N= 2^{10} nodes.
- Message length *M*=32 and 256 flits.
- Time-out period (τ) is equal to the message length; it is equal to the time required to transmit a message across a physical channel (τ =32 and 256 cycles).
- Number of virtual channels per physical channel *V*=2 and 3.
- Two traffic patterns that exhibit communication locality are considered. In the first pattern (referred to as "communication pattern 1"), the decreasing probability routing distribution is defined such that $p_1 = 0.90$, $p_2 = 0.10$, and $p_i = 0$ $(3 \le i \le 10)$ while in the second pattern ("communication pattern 2"), $p_1 = 0.70$, $p_2 = 0.20$, and $p_i = 0.0125$ $(3 \le i \le 10)$.

Fig. 1 depicts the mean message latency results predicted by the proposed model plotted against those provided by the simulator as a function of the injected traffic. The horizontal axis in the figure represents the traffic rate (λ_g) while the vertical axis shows the mean message latency. The figure reveals that the analytical model predicts the mean message latency with a good degree of accuracy when the network is operating under light and moderate traffic. However, there are some differences between the model and simulation as the network approaches the saturation point in the heavy traffic region. This is due to the approximations that have been made to ease the development of the model.

4 Conclusions

This paper has presented an analytical model to compute the mean message latency in wormhole-routed hypercubes with the adaptive routing algorithm proposed by Duato in [Dua94]. The algorithm improves network performance when traffic exhibits strong communication locality by using a time-out mechanism for selecting a particular class of virtual channels during message routing. The proposed analytical model is based on assumptions widely used in similar studies. Validation experiments have revealed that the latency results predicted by the model are in good agreement with those provided by the simulation model. Our next objective is to extend our proposed modelling approach to other common multicomputer networks, such as high-radix *k*-ary *n*-cubes and meshes.

References

[Aga91] A. Agarwal, Limits on interconnection network performance, *IEEE TPDS*, vol. 2, pp.398-412, 1991.

[And97] E. Anderson, J. Brooks, C. Grassl, S. Scott, Performance of the Cray T3E multiprocessor, *Proc. Supercomputing Conference*, 1997.

[Bou94] Y. Boura, C.R. Das, T.M. Jacob, A performance model for adaptive routing in hypercubes, *Proc. Int. Workshop Parallel Processing*, Bangalore, India, pp. 11-16, 1994.

[Cic97] B. Ciciani, M. Colajanni and C. Paolucci, An accurate model for the performance analysis of deterministic wormhole routing, *Proc. Int. Parallel Processing Symposium*, pp. 353-359, 1997.

[Dal65] D.J. Daley, General customer impatience in the queue GI/G/1, *J. of Applied Probability*, vol. 2, pp. 186-205, 1965.

[Dal90] W.J. Dally, Performance analysis of *k*-ary *n*-cubes interconnection networks, *IEEE TC*, 39(6), pp. 775-785, 1990.

[Dal92] W.J. Dally, Virtual channel flow control, *IEEE TPDS*, vol. 3, no. 2, pp. 194-205, 1992.

[Dal94] W.J. Dally *et al*, The reliable router: A reliable and high-performance communication substrate for parallel computers, *Proc. 1st workshop Parallel Computers, Routing & communication*, in K. Bolding and L. Snyder (Eds.), LNCS, Springer-Verlag, pp. 241-255, 1994.

[Dra94] J.T. Draper and J. Ghosh, A comprehensive analytical model for wormhole routing in multicomputer systems, *J. Parallel & Distributed Computing*, vol. 32, pp. 202-214, 1994.

[Dua93] J. Duato, A new theory of deadlock-free adaptive routing in wormhole routing networks, *IEEE Trans. Parallel & Distributed Systems* 4(12), pp. 1320-1331, 1993.

[Dua94] J. Duato, Improving the efficiency of virtual channels with time-dependent selection functions, *Future Generation Computer Systems* 10(10), pp. 45-58, 1994.

[Dua97] J. Duato, S. Yalamanchili and L. Ni, *Interconnection Networks: An Engineering Approach* (IEEE CS Press, 1997).

[Fil97] M. Fillo, S.W. Keckler, W.J. Dally, N.P. Carter, A. Chang, Y. Gurevich, W.S. Lee, The M-Machine multicomputer, *Int. Journal of Parallel Programming*, 25(3), pp. 183-212, 1997.

[Hu95] P. C. Hu, L. Kleinrock, A queueing model for wormhole routing with time-out, *Proc. 4th Int. Conf. on Computer Communications and Networks*, pp. 584-593, 1995.

[Kim94] J. Kim and C.R. Das, Hypercube communication delay with wormhole routing, *IEEE TC* 43(7), pp. 806-814, 1994.

[Kle75] L. Kleinrock, *Queueing Systems: Theory,* vol.1, John Wiley & Sons, New York, 1975.

[Lin93] X. Lin, P.K. Mckinley and L.M. Lin, The message flow model for routing in wormhole-routed networks, *Proc. International Conference on Parallel Processing*, pp. 294-297, 1993.

[Lau97] J. Laudon, D. Lenoski, The SGI Origin: a ccNUMA highly scalable server, *Proc. ACM/IEEE 24th Int. Symp. Computer Architecture*, pp. 241-251, 1997.

[N-C90] N-Cube Company, *NCUBE-2 Processor Manual*, 1990.

[Ni93] L. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, *IEEE Computer* 26(2), pp. 62-76, 1993.

[Nug88] S.F. Nugent, The iPSC/2 direct-connect communication technology, *Proc. Conf. Hypercube Concurrent Computers & Applications* (1), pp. 51-60, 1988.

[Oul99] M. Ould-Khaoua, A performance model for Duato's fully adaptive routing algorithm in *k*-ary *n*-cubes, *IEEE Trans. Computers* 48(12), pp. 1-8, 1999.

[Ree87] D.A. Reed, R.M. Fujitomo, *Multicomputer networks: Message based parallel processing,* MIT Press, 1987.

[Su93] C. Su and K.G. Shin, Adaptive deadlock-free routing in multicomputers using one extra channel, *Proc. International Conference on Parallel Processing*, pp. 175-182, 1993.

[Tij86] H.C. Tijms, *Stochastic modelling and analysis: A computational approach* [page 32] (J. Wiley, 1986).

[Van94] B. Vanvoorst, S. Seidel, E. Barscz, Workload of an iPSC/860, *Proc.Scalable High-Performance Computing Conf.*, pp. 221-228, 1994.

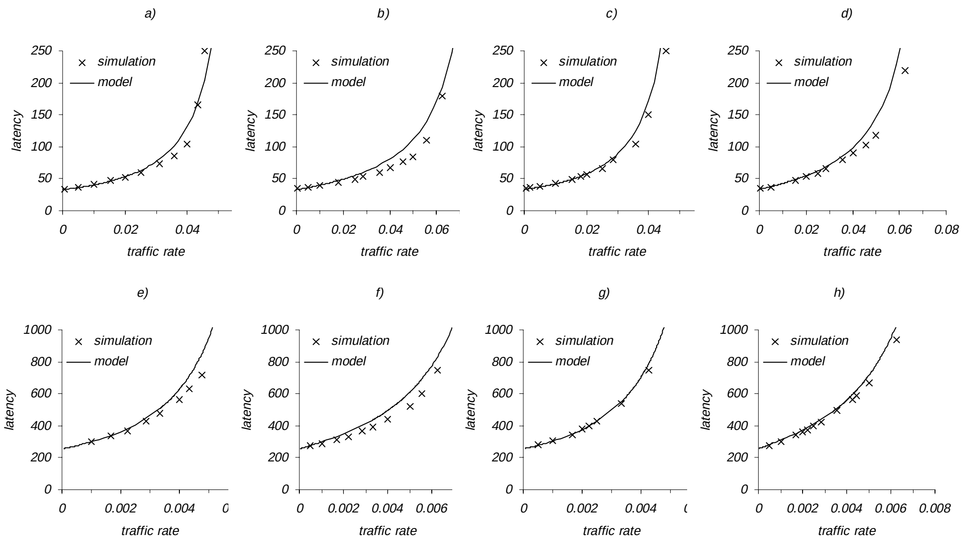

Fig. 1. Latency predicted by the model and simulation in the hypercube. a) Communication Pattern 1, $M = \tau = 32$, *V*=2, b) Communication Pattern 1, $M = \tau = 32$, *V*=3, c) Communication Pattern 2, $M = \tau = 32$, *V*=2, d) Communication Pattern 2, $M = \tau = 32$, *V*=3, e) Communication Pattern 1, $M = \tau = 256$, *V*=2, f) Communication Pattern 1, $M = \tau = 256$, *V*=3, g) Communication Pattern 2, $M = \tau = 256$, *V*=2, h) Communication Pattern 2, $M = \tau = 256$, *V*=3.