Strathprints Institutional Repository

A General Analytical Model of Adaptive Wormhole Routing in *k*-Ary *n*-Cube Interconnection Networks

A. Khonsari, M. Ould-Khaoua, J. Ferguson*,

Department of Computing Science
University of Glasgow, Glasgow G12 8QQ, UK.
E-mail: {ak, mohamed}@dcs.gla.ac.uk

**Department of Computer and Information Sciences*
University of Strathclyde, Glasgow, G1 1XH, UK,
Email: John.Ferguson@cis.strath.ac.uk

Abstract

Several analytical models of fully adaptive routing have recently been proposed for k-ary n-cubes and hypercube networks under the uniform traffic pattern. Although, hypercube is a special case of k-ary n-cubes topology, the modeling approach for hypercube is more accurate than k-ary n-cubes due to its simpler structure. This paper proposes a general analytical model to predict message latency in wormhole-routed k-ary n-cubes with fully adaptive routing that uses a similar modeling approach to hypercube. The analysis focuses Duato's fully adaptive routing algorithm [12], which is widely accepted as the most general algorithm for achieving adaptivity in wormhole-routed networks while allowing for an efficient router implementation. The proposed model is general enough that it can be used for hypercube and other fully adaptive routing algorithms.

1. Introduction

It is widely recognised that one of the critical components of a multicomputer is the interconnection network used to connect the processing elements together. Most current multicomputers [6, 15, 22, 24, 25] employ *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 (where *k*=2) and the torus (where *n*=2). The former has been employed in multicomputers such as the N-Cube [22] and iPSC/2 [25] while the latter has been adopted in machines like the J-machine [24], CRAY T3E [6] and CRAY T3D [15].

Modern parallel routers significantly reduce average latency by using wormhole switching [7]. Wormhole is a switching strategy that divides each packet in elementary units called *flits*, each of a few bytes for transmission and flow control, and advances each flit as soon as it arrives at a node. The *header* flit (containing routing information) governs the route and the remaining data flits follow it in a pipelined fashion. If a channel transmits the header of a message, it must transmit all the remaining flits of the same message before transmitting flits of another message. Once the header is blocked, the data flits are blocked in-situ. Wormhole is attractive because it reduces the latency of message delivery compared to store and forward and requires only a few flit buffers per node. Network throughput of wormhole routed networks can be increased by organizing the flit buffers associated with each physical channel into several virtual channels [9]. These virtual channels are allocated independently to different packets and compete with each other for the physical bandwidth. This decoupling allows active messages to pass blocked messages using network bandwidth that would otherwise be wasted.

Most interconnection networks including *k*-ary *n*-cubes provide multiple physical paths for routing a message between two given nodes. This introduces the problem of choosing a route between many alternatives. Many practical multicomputers [15, 24] have adopted *deterministic routing* where messages with the same source and destination addresses always take the same network path. This form of routing has been popular because it requires a simple deadlock-avoidance algorithm, resulting in a simple router implementation. However, messages cannot use alternative paths to avoid congested channels, and thus reduce their latency. *Fully-adaptive* routing has often been suggested to overcome this limitation by enabling messages to explore all available paths. Several authors like Duato [12], Lin *et al* [20], and Su and Shin [29] have proposed fully-adaptive routing algorithms, which can achieve deadlock-freedom with a minimal requirement for virtual channels, allowing for an efficient router implementation.

Analytical models of deterministic routing in common wormhole-routed networks including the *k*-ary *n*-cube have been widely reported in the literature [2, 4, 5,

ISBN: 1-56555-269-5

11, 14, 17]. Several researchers have recently proposed analytical models of fully-adaptive routing under the uniform traffic pattern [4, 26, 28]. For instance, Boura *et al* [4] have proposed a model of fully-adaptive routing in the hypercube. The authors in [26, 28] have described recently models for the high-radix k-ary n-cubes.

The most difficult part in developing any analytical model of adaptive routing is the computation of the probability of message blocking at a given router due to the number of combinations that have to be considered when enumerating the number of paths that a message may have used to reach its current position in the network. The problem is further exacerbated when the network dimensionality increases as the number of alternative paths increases. The model in [28] computes the exact expressions for the probability of message blocking at a given router by considering all the possible paths that enable a message to cross from its source to its current position in the network. However, the model is very time consuming due to recursive calculations of message blocking for each node in each iteration of message latency calculation. This paper proposes an alternative analytical model for computing the mean message latency in k-ary n-cubes with fully-adaptive routing. The derivation of the model is similar to the hypercube model presented in [4] and is general that can be used for k-ary n-cubes and hypercubes.

As in previous similar studies [4, 26, 28], the present analysis uses Duato's fully adaptive routing algorithm [12]. This form of routing is widely accepted as one of the most general fully-adaptive routing algorithm for wormhole-routed networks, leading to an efficient router implementation. The Cray T3E [6] and the reliable router [10] are two examples of recent practical systems that have adopted Duato's routing algorithm. However, the modelling approach can be easily adopted by other fully-adaptive routing algorithms [e.g. 3, 18, 21].

The rest of the paper is organised as follows. Section 2 reviews some definitions and background that will be useful for the subsequent sections. Section 3 present the analytical model and finally, section 5 concludes this study.

2. Preliminaries

The unidirectional k-ary n-cube, where k is referred to as the *radix* and n as the *dimension*, has $N=k^n$ nodes, arranged in n dimensions, with k nodes per dimension. Each node can be identified by an n-digit radix k address ($a_1, a_2, \ldots, a_n$).. The i^{th} digit of the address vector, a_i, represents the node position in the i^{th} dimension. Node with address ($a_1, a_2, \ldots, a_n$) is linke to node ($b_1, b_2, \ldots, b_n$) if and only if there exists i, $(1 \le i \le n)$, such that $a_i = (b_i + 1) \bmod k$ and $a_j = b_j$ for $1 \le j \le n$; $i \ne j$. Thus, each node is connected to a neighbouring node in each dimension.

Each node consists of a processing element (PE) and switching element (SE) or route. The PE contains a processor and some local memory. The router has $(n+1)$ input and $(n+1)$ output channels. A node is connected to its neighboring nodes through n inputs and n output channels in a unidirectional k-ary n-cube. The remaining channels are used by the PE to inject/eject messages to/from the network 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 [7]. The router contains flit buffers for any incoming virtual channel. An $(n+1)V$-way crossbar switch direct message flits from any input virtual channel to any output virtual channel. Such a switch can simultaneously connect multiple input to multiple output virtual channels while there is no conflicts.

Deadlock-free fully-adaptive routing algorithms that require only one extra virtual channel compared to deterministic routing have been discussed in [12, 13, 29] of which Duato's fully-adaptive routing algorithm is most known and widely used in studies and practice as it provide the maximum adaptivity with the minimum number of virtual channels.

Duato's algorithm [12] 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. The virtual channels of class b define a complete deadlock-free virtual sub-network, which acts like a "drain" for the virtual sub-networks built from virtual channels belonging to class a. In k-ary n-cubes, Duato's algorithm requires at least three virtual channels per physical channel to ensure deadlock-freedom where the class a contains one virtual channel and class b owns two virtual channels. When there are more than three virtual channels, network performance is maximised when the extra virtual channels are added to class a [12, 13]. Thus, with V virtual channels per physical channel, the best performance is achieved when class a and b contain V-2 and 2 virtual channels respectively. When the network is a hypercube (k=2), however, arrangement of virtual channels will be different. In this case Duato's algorithm requires at least one virtual channel in class b and all the remainder virtual channels to be included in class a virtual channels.

3. Analysis

The model uses assumptions that are widely used in the literature [1, 2, 4, 5, 8, 9, 11, 14, 17, 26, 28].

a) Nodes generate traffic independently of each other, and which follows a Poisson process with a mean rate of λ messages per cycle.
b) The arrival process at a given channel is approximated by an independent Poisson process.
c) Message destinations are uniformly distributed across network nodes.
d) Message length is fixed and equal to *M* flits, each of which is transmitted in one cycle from one router to the next.
e) The local queue at the injection channel in the source node has infinite capacity. Moreover, messages are transferred to the local PE as soon as they arrive at their destinations through the ejection channel.
f) *V* virtual channels are used per physical channel. Class *a* contains $(V-2)$ virtual channels, that are crossed adaptively. On the other hand, class *b* contains two virtual channels that are crossed deterministically. Let the virtual channels belonging to class *a* and *b* be called the adaptive and deterministic virtual channels respectively. When there is more than one adaptive virtual channel available a message chooses one at random. To simplify the model derivation no distinction is made between the deterministic and adaptive virtual channels when computing virtual channels occupancy probabilities [4, 26, 28].

The model computes the mean message latency as follows. First, the mean network latency, $\overline{S}$, that is the time to cross the network is determined. Then, the mean waiting time seen by a message in the source node, $\overline{W_s}$, is evaluated. Finally, to model the effects of virtual channels multiplexing, the mean message latency is 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

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

The average number of hops that a message makes across the network, $\overline{d}$, is given by

$$\overline{d} = \sum_{i=1}^{d_{\max}} i p_i \tag{2}$$

where p_i is the probability that a newly-generated message makes i hops to reach its destination and $d_{\max}$ is the maximum distance that a message may traverse to reach its destination (also called network *diameter*) and are given by

$$d_{\max} = n(k-1) \tag{3}$$

To compute p_i let us refer to the following result from the combinatorial theory [23, 27, 30].

Proposition 1: *The number of ways to distribute r like objects into m different cells, such that no cell contains less than q objects and not more than* $q+k-1$ *objects is the coefficient of* x^{r-qm} *in the expansion of the polynomial*

$$F(x) = (1-x^k)^m (1-x)^{-m} = (1+x+x^2+.....+x^{k-1})^m$$

Let us refer to the coefficient of x^{r-qm} as $N_q^{q+k-1}(r,m)$.

In [30], the expression of $N_q^{q+k-1}(r,m)$ is given by

$$N_q^{q+k-1}(r,m) = \sum_{l=0}^{m} (-1)^l \binom{m}{l} \binom{r-mq-lk+m-1}{m-1} \tag{4}$$

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 which are *i* hops away from a given node in the *k*-ary *n*-cube as

$$n_i = N_0^{k-1}(i,n) = \sum_{l=0}^{n} (-1)^l \binom{n}{l} \binom{i-lk+n-1}{n-1} \tag{5}$$

Hence, recalling that a node can not send a message to itself, p_i can be written as

$$p_i = \frac{n_i}{N-1} \tag{6}$$

with N being the number of nodes in the network ($N = k^n$).

Fully-adaptive routing allows a message to use any available channel that brings it closer to its destination resulting in an evenly distributed traffic rate on all network channels. A router in the *k*-ary *n*-cube has *n* output channels and the PE generates, on average, λ messages in a cycle. Since each message travels, on average, $\overline{d}$ hops to cross the network the rate of messages received by each channel, λ_c, can be written as [2]

$$\lambda_c = \frac{\lambda \overline{d}}{n} \tag{7}$$

The network latency for a message consists of two parts: one is the delay due to the actual message transmission time i.e. $M+i$, and the other is due to blocking in the network. The network latency of an *i*-hop message, S_i, can therefore be written as

$$S_i = M + i + \sum_{h=1}^{i} B_{h,i} \quad (8)$$

where M is the message length and $B_{h,i}$ is the mean blocking time seen by an *i*-hop message at the h^{th}-hop channel $(1 \le h \le i)$ in its journey. Averaging over all the possible nodes destined made by a regular message yields the mean network latency for regular messages as

$$\overline{S} = \sum_{i=1}^{d_{max}} p_i S_i \quad (9)$$

Calculation of the message blocking ($B_{h,i}$)

A message is blocked at a given 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. When blocking occurs a message has to wait for a deterministic virtual channel at the lowest dimension [12]. Note that under the uniform traffic pattern and due to the symmetry of the *k*-ary *n*-cube topology, adaptive routing results in an evenly distributed traffic rate on all network channels. Furthermore, a message sees the same mean waiting time and mean service time across all channels regardless of their positions in the network. However, the message sees a different probability of blocking at each channel as the number of alternative paths, that can be selected, changes from one channel to the next. The probability of blocking depends on the number of output links, and thus on the virtual channels that a message can use at its next hop.

Consider a message that has to cross *i* hops to reach its destination. Suppose that this *i*-hop message has reached the h^{th}-hop channel $(1 \le h < i)$ along its path. Let $P_{block_{h,i}}$ and w denote the probability of blocking of an *i*-hop message in its h^{th}-hop channel and the mean waiting time when blocking occurs, respectively. The mean blocking time, $B_{h,i}$, is given by

$$B_{h,i} = \sum_{h=1}^{i} P_{block_{h,i}} w \quad (10)$$

The probability of blocking $P_{block_{h,i}}$ is computed as follows. The number of alternate routes that an *i*-hop message can select when it reaches channel *h*, to advance towards its destination depends on the number of dimensions it has already passed. Let $\varphi_{h,i}$ be the number of dimensions that an *i*-hop message still has to visit when crossing channel h ($\varphi_{h,i}$ are determined below). Hence, the probability $P_{block_{h,i}}$ can be calculated as

$$P_{block_{h,i}} = P_{a\&d} P_a^{\varphi_{h,i}-1} \quad (11)$$

with P_a being the probability that all adaptive virtual channels of a physical channel are busy and $P_{a\&d}$ being the probability that all adaptive and deterministic virtual channels of a physical channel are busy. To compute P_a three cases should be considered, and are as follows [26, 28].

a) V virtual channels are busy which means all adaptive virtual channels are busy as well.

b) (V-1) virtual channels are busy. The number of combinations where (V-1) out of V virtual channels are busy is $\binom{V}{V-1}$ of which only two combinations result in all adaptive virtual channels being busy.

c) (V-2) virtual channels are busy. The number of combinations where (V-2) out of V virtual channels are busy is $\binom{V}{V-2}$ of which only one combination results in all adaptive virtual channels being busy.

Similarly, to compute $P_{a\&d}$ two cases should be considered, and these are the following.

a) V virtual channels are busy, that is all adaptive, and deterministic virtual channels are busy.

b) (V-1) virtual channels are busy. In this case, only two combinations out of $\binom{V}{V-1}$ result in all adaptive and deterministic virtual channels being busy.

Let P_v $(0 \le v \le V)$ represent the probability that v virtual channels at a physical are busy. Taking into account the different cases mentioned above, P_a and $P_{a\&d}$ are given in terms of P_v by [26, 28]

$$P_a = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} + \frac{P_{V-2}}{\binom{V}{V-2}} \quad (12)$$

$$P_{a\&d} = P_V + \frac{2P_{V-1}}{\binom{V}{V-1}} \quad (13)$$

Calculation of the average number of channels that an i-hop message can select at channel h ($\varphi_{h,i}$)

Let $S = s_1 s_2 \cdots s_n$ be the source node and $D = d_1 d_2 \cdots d_n$ denotes a destination. Let us define the set $I_x = \{i_{x_l}\}, (1 \le l \le n)$, $(1 \le x \le n_i)$, where each element denotes the number of hops that the message makes along each dimension l when it traverses the network from the source node to the destination node, that is $(s_l + i_{x_l}) \bmod k = d_l$. The index x in i_{x_l} represents each of the n_i nodes that are *i*-hop away from the source node, in which $i_x = \sum_{l=1}^{n} i_{x_l}$.

For the following discussion let set $I = \{i_1, i_2, \cdots, i_n\}$ denotes one of the n_i, *i*-hop messages and let m_I and $Z = \{z_1, z_2, \cdots, z_{m_I}\}$ denote the number of non-zero and the set of non-zero elements in this set, respectively. The number of ways to distribute h $(1 \le h \le i)$ hops over m_I dimensions such that the number of hops made in each dimension l $(1 \le l \le m_I)$ be at most the *l*-th element of the set *I,* that is i_l, can be calculated the following result from combinatorial theory [23, 30].

Proposition 2: *The number of ways to distribute r indistinguishable objects into m distinguishable cells, such that no cell contains less than* q_l *objects and not more than* $(q_l + k_l - 1)$ $1 \le l \le m$ *objects is given by the coefficient of* x^r *in the following product.*

$$f(x) = \prod_{l=1}^{m} (x^{q_l} + x^{q_l+1} + \ldots + x^{q_l+k_l-1}) \tag{14}$$

Let us refer to the coefficient of x^r as $N_{Q,K}^{m}(r)$, $Q = (q_1, q_2, .., q_m)$, $K = (k_1, k_2, .., k_m)$. In [16], an expression for $N_{Q,K}^{m}(r)$ is calculated that is given by

$$N_{Q,K}^{m}(r) = \sum_{j=0}^{2^m-1} (-1)^{\sum_{l=1}^{m} j_l} \binom{m + r - \sum_{l=1}^{m} q_l - 1 - \sum_{l=1}^{m} j_l k_l}{m-1}$$

$$Q = (q_1, .., q_m), K = (k_1, .., k_m) \tag{15}$$

If the hops made by the *i*-hop message at channel h are treated as like objects and the visited dimensions, m_I, as different cells, the number of ways to distribute h hops over m_I dimensions, $n_{i,h}$, can be written as

$$n_{i,h} = N_{0,Z}^{m_I}(h) \qquad Z = \{z_1, z_2, \cdots, z_{m_I}\} \tag{16}$$

In this way h hops are distributed over m_I dimensions such that the number of hops made at each dimension l be at most the *l*-th, $1 \le l \le n$, element of the set *Z*, that is z_l. The probability that a message has entirely crossed one dimension on its h^{th}-hop is given by

$$Pass_h^1(i) = \frac{\sum_{l=1}^{m'_I(1)} N_{0,Z'(l)}^{m'_I(1)}(h - z_l)}{N_{0,Z}^{m_I}(h)} \tag{17}$$

where $m'_I(1) = m_I - 1$ and $Z'(l) = \{z_1 - 1, z_2 - 1, \cdots, z_{l-1} - 1, z_{l+1} - 1, \cdots, z_{m'_I(l)} - 1\}$. Similarly, the probability that a message has entirely crossed two dimensions on its h^{th} hop can be expressed as

$$Pass_h^2(i) = \frac{\sum_{l_1}^{m'_I(2)} \sum_{l_2=l_1+1}^{m'_I(2)} N_{0,Z'(l_1,l_2)}^{m'_I(2)}(h - z_{l_1} - z_{l_2})}{N_{0,Z}^{m_I}(h)} \tag{18}$$

where $m'_I(2) = m_I - 2$ and $Z'(l_1, l_2) = \{z_1 - 1, \cdots, z_{l_1-1} - 1, z_{l_1+1} - 1, \cdots, z_{l_2-1} - 1, z_{l_2+1} - 1, \cdots, z_{m'_I(2)} - 1\}$

More generally, the probability that a message has entirely crossed *y* dimensions can be written as

$$Pass_h^y(i) = \frac{\sum_{l_1=1}^{m'_I(y)} \sum_{l_2=l_1+1}^{m'_I(y)} \cdots \sum_{l_y=l_{y-1}+1}^{m'_I(y)} N_{0,Z'(l_1,l_2,\cdots,l_y)}^{m'_I(y)}(h - \sum_{i=1}^{y} z_{l_i})}{N_{0,Z}^{m_I}(h)} \tag{19}$$

where $m'_I(y) = m_I - y$ and $Z'(l_1, l_2, \cdots, l_y) = \{z'_1, z'_2, \cdots, z'_{m'_I(y)}\}$ is the non-zero elements in the following set

$$I'(l_1, l_2, \cdots, l_y) = \{I'_1, I'_2, \cdots, I'_n\}$$

$$I'_i = \begin{cases} 0 & i = l_1 \ \ or \ \ i = l_2 \ \ or \ \cdots \ or \ \ i = l_y \\ I_i - 1 & otherwise \end{cases} \tag{20}$$

Considering all of the *i*-hop messages and using the above equation, the probability of passing *y* dimensions for an *i*-hop on its h^{th} hop can be expressed as

$$P_h^y(i) = \sum_{x=1}^{n_i} Pass_h^y(i_x) \tag{21}$$

The number of channels, and thus the number of virtual channels, that a message can select at a given hop depends on the number of dimensions still to be visited. When a message arrives at channel i it has already made

$(i-1)$ hops and has crossed, say, y dimensions, $(1 \le y \le n)$. At its next hop the message can use (n-y) channels at the remaining (n-y) dimensions still to be visited. Averaging all of the possible cases yields the number of channels, $\varphi_{h,i}$, that the message can select when crossing channel h, $(1 \le i \le d_{\max}), (1 \le h \le i)$, as

$$\varphi_{h,i} = \sum_{y=0}^{n} (n-y) P_h^y (i) \tag{22}$$

Calculation of the mean waiting time at a channel (w), local queue ($\overline{W}_s$)

To determine the mean waiting time, w, to acquire a virtual channel a physical channel is treated as an M/G/1 queue with a mean waiting time of [19]

$$w = \frac{\rho \overline{S}(1 + C_{\overline{S}}^2)}{2(1-\rho)} \tag{23}$$

$$\rho = \lambda_c \overline{S} \tag{24}$$

$$C_{\overline{S}}^2 = \frac{\sigma_{\overline{S}}^2}{\overline{S}^2} \tag{25}$$

where λ_c is the traffic rate on the channel given by equation 7, $\overline{S}$ is its service time calculated in equation 9, and $\sigma_{\overline{S}}^2$ is the variance of the service time distribution. Since the minimum service time at a channel is equal to the message length, M, following a suggestion proposed in [11], the variance of the service time distribution can be approximated as $\sigma_{\overline{S}}^2 = (\overline{S} - M)^2$. Hence, the mean waiting time becomes

$$w = \frac{\lambda_c \overline{S}^2 \left(1 + \frac{(\overline{S}-M)^2}{\overline{S}^2}\right)}{2(1-\lambda_c \overline{S})} \tag{26}$$

Similarly, modelling the local queue in the source node as an M/G/1 queue, with the mean arrival rate and service time $\overline{S}$ with an approximated variance $\left(\overline{S} - M\right)^2$ yields the mean waiting time seen by a message at source node as [19]

$$W_s = \frac{\frac{\lambda}{V} \overline{S}^2 \left(1 + \frac{\left(\overline{S}-M\right)^2}{\overline{S}^2}\right)}{2(1 - \frac{\lambda}{V} \overline{S})} \tag{27}$$

Calculation of the average degree of virtual channels multiplexing ($\overline{V}$):

The probability, P_v, that v adaptive virtual channels are busy at a physical channel can be determined using a Markovian model. State π_v $(0 \le v \le V)$ corresponds to v virtual channels being busy. The transition rate out of state π_v to state π_{v+1} is the traffic rate λ_c (given by equation 7) while the rate out of state π_v to state π_{v-1} is $\frac{1}{\overline{S}}$ ($\overline{S}$ is given by equation 9). The transition rates out of state π_V are reduced by λ_c to account for the arrival of messages while a channel is in this state. The steady-state solutions of the Markovian model yeild the probability P_v $(1 \le v \le V)$ as [9]

$$P_v = \begin{cases} \frac{1}{\sum_{i=0}^{V} Q_i} & v = 0 \\ P_0 Q_v & 1 \le v \le V \end{cases} \tag{28}$$

$$Q_v = \begin{cases} (\lambda_c \overline{S})^v & 0 \le v \le V-1 \\ \frac{(\lambda_c \overline{S})^V}{1-\lambda_c \overline{S}} & v = V \end{cases} \tag{29}$$

When multiple virtual channels are used per physical channel they share the bandwidth in a time-multiplexed manner. The average degree of multiplexing of virtual channels, that takes place at a given physical channel, can be estimated by [9].

$$\overline{V} = \frac{\sum_{v=1}^{V} v^2 P_v}{\sum_{v=1}^{V} v P_v} \tag{30}$$

The above equations reveal that there are several inter-dependencies between the different variables of the model. For instance, Equations 8, 9 and 10 reveal that $\overline{S}$ is a function of w while equation 26 shows that w is a function of $\overline{S}$. Given that closed-form solutions to such inter-dependencies are very difficult to determine the different variables of the model are computed using iterative techniques for solving equations.

In Fig. 1, we have compared the proposed model to the accurate model proposed in [28]and the model proposed in [26], which referred as general, complex and average in the figure respectively for two different network, namely the 8-ary 3-cube and 10-ary 5-cube with message lengths M=32 and 64 flits and V=3 and 5 virtual channels per physical channel. As can be seen in the figure, the proposed model in this paper is almost matching to the other models. However, it is slightly overestimating the mean message latency which causes an earlier saturation compared to accurate model proposed in [28]. Although the model is slightly less accurate than the model in [28] under heavy traffic loads and near the saturation region, its generality and simple applicability to other routing algorithms makes it an attractive tool for studying performance metrics of k-ary n-cubes and hypercubes under different working conditions. Moreover, in Fig. 1, we have compared the proposed model to the hypercube model proposed in [4], which referred as bora, and general in the figure respectively for the 2-ary 3-cube with message lengths M=32 and 64 flits and V=3 virtual channels per physical channel. As can be seen in the figure, the proposed model in this paper exactly matches to the Bora model. However, it is worth mentioning that equations 12 and 13 in the hypercube is given by [4].

5. Conclusion

This paper has described an analytical model to compute the mean message latency in wormhole-routed *k*-ary *n*-cubes and hypercube with Duato's fully-adaptive routing algorithm. The proposed model achieves a good degree of accuracy under different operating conditions. Furthermore, it manages to achieve this good degree of accuracy wile maintaining generality, ease of applicability and efficient execution time, making it a practical evaluation tool that can be used to gain insight into the performance behavior of fully-adaptive routing in wormhole-routed *k*-ary *n*-cubes.

Our next objective is to develop analytical models for other common network topologies for multicomputers, e.g., n-dimensional meshes, which are variations of k-ary n-cubes without wrap-around connections. Developing a model for meshes is more complicated than for k-ary n-cubes because traffic rates and service times have to be computed at each network channel as these differ from one channel to the next due to the inherent asymmetry of these topologies.

References

[1] S. Abraham, K. Padmanabhan, Performance of the direct binary *n*-cube networks for multiprocessors, *IEEE Trans. Computers* Vol. 37, No. 7, pp. 1000-1011, 1989.

[2] A. Agarwal, Limits on interconnection network performance, *IEEE Trans. Parallel & Distributed Systems* (2)4, pp. 398-412, 1991.

[3] K. V. Anjan, T.M. Pinkston and J. Duato, Generalised theory for deadlock-free adaptive wormhole routing and its application to DISHA concurrent, *Proc. 10th Int. Parallel Processing Symposium*, IEEE Computer Society Press, Honululu, pp. 815-821, 1996.

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

[5] B. Ciciani, M. Colajanni, C. Paolucci, An accurate model for the performance analysis of deterministic wormhole routing, *Proc. 11th Int. Parallel Processing Symposium*, pp. 353-359, 1997.

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

[7] W.J. Dally, C.L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks, *IEEE Trans. Computers* C36(5), pp. 547-553, 1987.

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

[9] W.J. Dally, Virtual channel flow control, *IEEE Trans. Parallel & Distributed Systems* 3(2), pp. 194-205, 1992.

[10] W.J. Dally, H. Aoki, Deadlock-free adaptive routing in multicomputers using virtual channels, *IEEE Trans. Parallel & Distributed Systems* 4(4), pp. 466-475, 1993.

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

[12] J. Duato, A New theory of deadlock-free adaptive routing in wormhole routing networks, *IEEE Trans. Parallel & Distributed Systems* 4(12), pp. 320-1331, 1993.

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

[14] R. Greenberg, L. Guan, Modelling and comparison of wormhole routed mesh and torus networks, *Proc. 9th IASTED Int. Conf. Parallel & Distributed Computing and Systems*, 1997.

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

[16] A. Khonsari, H. Sarbazi-Azad: On the Problem of Distributing Objects in Cells with Different Capacities, *Proc. International Conference on Algorithms Theory, Discrete Mathematics, Systems and Control,* December 17-21, 2001, Cairns, Queensland, Australia, also in V.V. Kluev, N.E. Mastorakis (Editors), Topics in Applied and Theoretical Mathematics and Computer Science, WSEAS Press, ISBN: 960-8052-47-6, pp.155-157, 2001.

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

[18] J. Kim, A. Chien and Z. Liu, Compressionless routing: A framework for adaptive and fault- tolerant

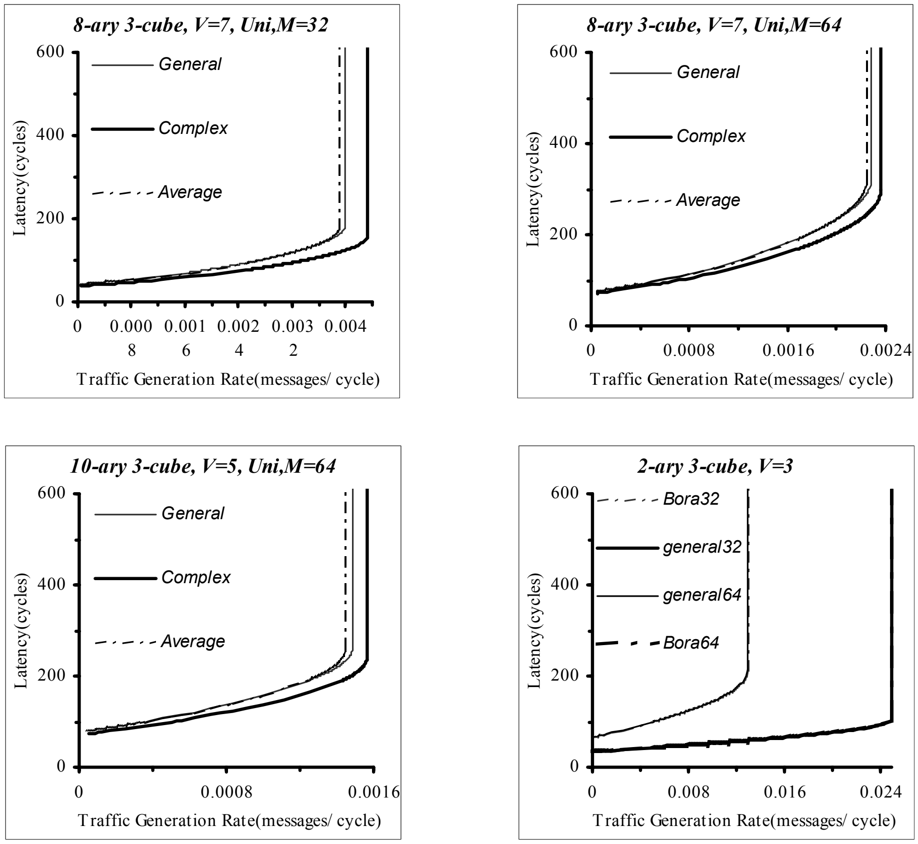

Fig. 1: Average message Latency predicted by the model against simulation results for an 8-ary 3-cube and 10-ary 3-cube and 3 dimensional hypercube with message length *M*=32, 64 virtual channel number *V*=3, *V*=5 and *V*=7 for 4 different models.

routing, *IEEE Trans. Parallel & Distributed Systems* 8(3), pp. 229-244, 1997.

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

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

[21] J. M. Martinez, P. Lopez, J. Duato, T. M. Pinkston, Software-based deadlock recovery techniques for true fully adaptive routing in wormhole networks, *Proc. Int. Conf. Parallel Processing (ICPP'97)*, pp. 182-189, 1997.

[22] N-Cube Systems, *N-cube Handbook*, N-Cube, 1986.

[23] R. Nelson, Probability, Stochastic processes, and queuing theory: The mathematics of Computer Performance Modeling, Springer Verlag, 1995.

[24] M. Noakes, W.J. Dally, System design of the J-machine, *Proc. Advanced Research in VLSI* MIT Press, pp. 179-192, 1990.

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

[26] M. Ould-Khaoua, An analytical model of Duato's adaptive routing algorithm, *Proc. Int. Conf. Parallel Processing*, pp. 106-113, 1998.

[27] J. Riorda, An introduction to combinatorial analysis, John Wiley & Sons, 1958.

[28] H. Sarbazi-Azad, M. Ould-Khaoua, L.M. Mackenzie, An accurate analytical model of adaptive wormhole routing in *k*-ary *n*-cube interconnection networks, *Performance Evaluation*, Vol. 43, No. 2-3, pp. 165-179, 2001.

[29] C. Su, K.G. Shin, Adaptive deadlock-free routing in multicomputers using one extra channel, *Proc. Int. Conf. Parallel Processing*, pp. 175-182, 1993.

[30] W.A. Whiteworth, Choice and Chance, Cambridge University Press, 1901.