ELSEVIER

Parallel Computing 24 (1998) 2053–2075

PARALLEL
COMPUTING

Performance evaluation of deterministic wormhole routing in *k*-ary *n*-cubes

B. Ciciani [a,1], M. Colajanni [b,*], C. Paolucci [a]

[a] *Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, Rome 00198, Italy*

[b] *Dipartimento di Informatica, Sistemi e Produzione, Università di Roma "Tor Vergata", Via di Tor Vergata, Rome 00133, Italy*

Received 15 June 1997; received in revised form 27 April 1998

Abstract

We present a new analytical approach for the performance evaluation of deterministic wormhole routing in *k*-ary *n*-cubes. Our methodology achieves closed formulas for average time values through the analysis of network flows. The comparison with simulation models demonstrates that our methodology gives accurate results for both low and high traffic conditions. Another important quality is the flexibility of our approach. We demonstrate that it can be used to model *dimension-ordered*-routing in several *k*-ary *n*-cubes such as hypercubes, 3D symmetric and asymmetric tori, architectures with uni- and bi-directional channels. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Performance analysis; Message passing; Dimension-ordered routing; Torus networks; Wormhole routing

1. Introduction

Wormhole routing is an efficient switching technique which is adopted by a variety of parallel machines with distributed memory architectures, such as Intel Paragon, nCUBE-2, MIT J-machine, Cray T3D. Most multicomputers employ wormhole in combination with *deterministic* routing policies: the route between two nodes is determined in a static way and, in case of link conflict, the transmission waits until the link becomes free. Although the deterministic policies do not have the

[*] Corresponding author. E-mail: colajanni@uniroma2.it

[1] E-mail: ciciani@dis.uniromal.it

PII: S0167-8191(98)00094-5

ability to respond to dynamic network conditions, their implementation is quite simple. Conversely, *adaptive* wormhole routing requires more sophisticated hardware and more complex algorithms for preventing deadlocks and livelocks. An interesting survey of wormhole routing is contained in [14]. Commercial multicomputers adopted deterministic routing in three types of k-ary n-cubes: uni-directional hypercube, uni-directional and bi-directional torus, bi-directional mesh [13]. In uni-directional systems, there is only one link between adjacent nodes, while in a bi-directional system, each connection between two nodes consists of one link for each direction. For bi-directional networks, the wrap-around links between the edge nodes are not essential, while in uni-directional systems without wrap-around, a communication may require k-1 hops even between adjacent nodes.

The goal of this paper is to propose an approach that achieves accurate closed formulas for the mean latency time of *dimension-ordered* wormhole routing in k-ary n-cubes. The error is below 10% in most instances, and much less for low and average traffic conditions. An important contribution of our results is that the same approach is immediately applicable to any k-ary n-cube with wrap-around connections for $k < 5$. For higher values of k, it is impossible to construct a deadlock-free minimal deterministic routing algorithm [5]. We present the methodology for hypercubes, an *asymmetric* $(k_0 \times k_1 \times k_2)$ and *symmetric* $(k \times k \times k)$ torus with uni-directional and bi-directional links, respectively.

Performance analysis of wormhole routing in k-ary n-cubes topologies has been investigated by several authors. The majority of them adopted simulation models [2–4,7,8,12,15,16], while few results were obtained analytically [6,11,1]. Adve and Vernon [1] use approximate Mean Value Analysis to analyze the performance of wormhole routing in a k-ary n-cube, where the system is modeled as a *closed* queuing network. This choice makes their model realistic, although complex to the extent that it can be solved only through iterative algorithms. Dally [6] proposes a general model for k-ary n-cubes, that he adopts to demonstrate that low k-dimensional topologies provide best performance. The results of this paper are a very important contribution in this area, however the Dally model solves only symmetric topologies with uni-directional links. Kim and Das [11] specialize this model for the hypercube topology. They obtain accurate results, although limited to the hypercube that is a perfect symmetric and balanced topology. Our approach is more general than the models proposed in [6,11], because it is applicable to several topologies such as hypercubes, symmetric and asymmetric tori with both uni-directional and bi-directional links. The contribution of our model is outlined in Table 1.

The paper is organized as follows. In Section 2, we describe the operational features of the interconnection network and motivate the assumptions that make the model analytically tractable. In Section 3, we present the approach for the message latency time evaluation focusing on a 3D asymmetric torus with uni-directional links. In Sections 4 and 5, we demonstrate the flexibility of our approach by extending the analysis to a 3D symmetric torus with bi-directional links and a hypercube, respectively. In Section 6, we validate all models by comparing the analytical results against time values which are obtained from our simulations and other published results. In Section 7, we use our models to evaluate the performance

Table 1
Comparison of analytical results for deterministic wormhole routing

		Dally 90	Kim and Das 94	Adve and Vernon 94	Model
Topology	k-ary n-cubes (Uni-directional symmetric)	X		X	X
	k-ary n-cubes (Uni-directional asymmetric)				X
	k-ary n-cubes (Bi-directional)			X	X
	Hypercube		X		X
System	Closed			X	
	Open	X	X		X
Result	Closed formulas	X	X		X
	Iterative methods			X	

of wormhole routing in several network dimensions and traffic loads. Section 8 concludes the paper with some final remarks.

2. Network description

2.1. Deterministic wormhole routing

There are three main routing strategies that transmit data while building the path from the source to the destination node: *store-and-forward*, *virtual-cut-through*, and *wormhole*. The store-and-forward technique collects the entire message at each intermediate node of the path before requesting the next link. A first reduction of the transmission time was achieved by the *virtual-cut-through* proposed by Kermani and Kleinrock [10]. Unlike store-and-forward, this technique divides the message in small parts, namely *flits*, and implements a pipeline switching technique: once the *header flit* has obtained the first link, it requires the second link while the *data flits* are transmitted from the first to the adjacent node. This strategy has the advantage of limiting message buffering to the instances in which the requested link is not available. In these cases, all data flits are gathered in the buffer of the last node reached by the header.

Latest generation multicomputers use *wormhole* routing techniques. Wormhole routing has same pipeline behavior as virtual-cut-through in a contention-free network. However, in the case of conflict, the two techniques differ. In brief, the wormhole communication evolves through two concurrent activities: *path-hole* and *data-trail*. During the path-hole phase, the *header flit* establishes the path from the source to the destination node through the *dimension-ordered* algorithm. The header flit is routed in one dimension at a time; it proceeds to the next dimension only when arrived at the proper coordinate in each dimension [9]. This popular algorithm is adopted by many commercial machines. On hypercube topologies is known as

E-cube routing [17], while on 2D mesh/torus is called *XY* routing. This algorithm guarantees deadlock-free minimal routing for any $k < 5$. Since the path from the sender to the destination node is deterministic, in the case of conflict the flits are stored in special *flit buffers* belonging to the controllers of the links that are used by the communication (*blocking* phase). The data-trail begins when the header flit obtains the first link and causes the transmission of all data flits to flow through the links which are obtained by the header. A link is released when the last flit (*tail flit*) of the message is transmitted through it.

2.2. Assumptions

Each node of the interconnection network contains two main components: a *processing element*, and a *communication controller* consisting of a $(n+1) \times (n+1)$ crossbar switch and a controller for each dimension that guarantees a maximum of n simultaneous communications for each router. The processing element uses the remaining two links to send and receive its own messages. Our approach considers an architecture model which is fairly accurate for current machines:

(1.1) *The memory capacity of each node is unlimited.* This assumption permits not to consider the loss of messages and consequent re-transmission due to insufficient memory space.

(1.2) *The flit size* (*in bits*) *is equal to the number of wires W that compose a link.*

(1.3) *Each flit is transmitted in a single cycle link time.* This time represents the basic temporal unit for the model.

Moreover, we adopt the following assumptions which are commonly accepted in literature to obtain a model analytically tractable:

(2.1) *The communications arising from each node are independent and identically distributed in accordance with a Poisson process having generation rate* $1/\tau$ (*bit/cycle*) *per node.*

(2.2) *The network has single flit buffers per link.* Actually, each physical channel of the uni-directional torus has two single flit buffers. One for each multiplexed virtual channel which is required to guarantee deterministic deadlock-free routing [5]. Since the equations of our model, such as the Dally's approach [6], do not give a detailed description of the virtual channels, contentions are solved at *message boundary* level instead of *flit boundary*. Although this assumption makes the model values of the uni-directional torus an upper bound approximation of the actual latency time, it does not affect the accuracy of the analytical results that remain very close to simulations.

The assumption (2.1) and the network topology with wrap-around connections guarantee that the network is *balanced* [10], that is, we can assume that any link is equally likely to be visited independently of the chosen distribution for the destination nodes. In this paper, the methodology is presented under the hypothesis of a *uniform* message distribution. However, this is not a theoretical limitation of the model because, for any other distribution that imposes regular traffic on the network, we have only to modify the average message path length.

3. The model for torus with uni-directional links

In this section we consider *XYZ dimension-ordered* routing in a $k_0 \times k_1 \times k_2$ torus with uni-directional links. The proposed methodology allows us to evaluate the average message transmission time through a closed formula. Moreover, we demonstrate the flexibility of the model by analyzing the routing in 3D symmetric torus with bi-directional links in Section 4, and hypercube in Section 5.

3.1. Flow analysis

To achieve closed formulas for the latency time, we model the wormhole message transmission by distinguishing three phases: *path-hole phase without conflicts* (PH), *path-hole phase with blocking* (PHB), *data-trail phase* (DT). This separation does not correspond to the actual behavior of the routing algorithm in which data-trail and path-hole phase are partially concurrent. However, if we consider realistic message transmissions, we have that the number of flits is usually larger than the number of hops between source and destination node. In those cases, the three phase assumption causes very limited approximations on the mean time values because when the tail flit of a message enters the network, the header flit has already reached the destination, so that the tail flit is transmitted without suffering delays due to conflicts.

Fig. 1 shows the part of the 3D asymmetric torus with uni-directional links that an average message sees while it flows from the source to the destination node. We can merely describe the PH and DT phases as single nodes, while we have to detail the PHB phase because the contention is the main issue that we have to address in the performance analysis of any routing policy. In Fig. 1, the nodes of the PHB phase represent the links, while the arcs denote the paths among the links. It has to be noted that this figure does not model the entire torus architecture, but only the subset of nodes that might be visited by an average message.

Since for uniformly distributed routing the average length of the path in each dimension is $k_j/2$, the flow in each dimension is represented by a column of $k_j/2$ nodes, where j indicates the dimension (0, 1, 2). The node c_{ij} $(i = 1, \ldots, k_j/2)$ denotes that the communication uses a link of the jth dimension as the ith link of its transmission in that dimension. A message may use all links of a column or a subset of links of two or three columns. However, the important point is that a message can never require links of a lower dimension because we are considering *XYZ* routing. The labels on the arcs are the rates of the messages that flow through the network, while more arcs reaching the same node indicate a possible point of conflict. At the entrance of the 1st and 2nd dimension, Fig. 1 depicts a small horizontal barrier that logically collects the flows that require the first link of a new dimension. The probability that a message does not use any link of the jth dimension during its transmission is $p_j = 1/k_j$. This equation is simply obtained from the observation that in a 3D torus, the following relation is true for each dimension: e.g., $p_0 = (k_1k_2)/(k_0k_1k_2)$, if one considers the 0th dimension. To simplify the notation we use $\alpha = 1 - p_0$, $\beta = 1 - p_1$ and $\gamma = 1 - p_2$ to denote the probability of a message transmission through the links of the 0th, 1st and 2nd dimension, respectively.

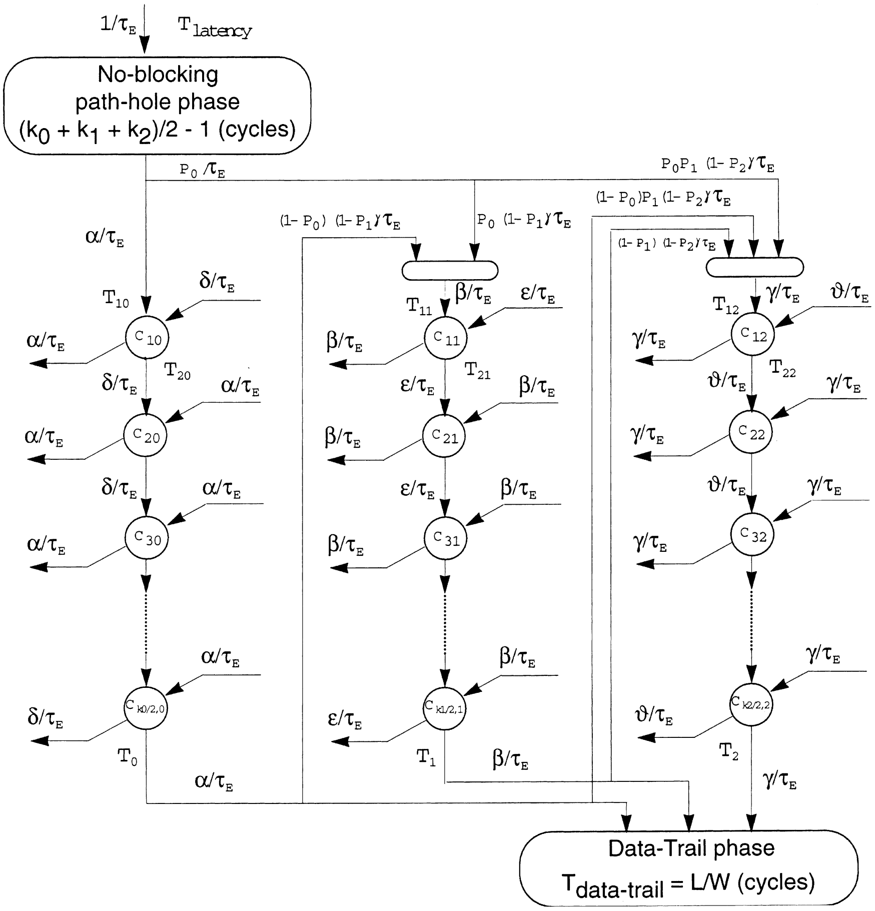

Fig. 1. Network's view of an average message transmission in a $k_0 \times k_1 \times k_2$ torus with uni-directional links.

The messages have an average length equal to L. Let $1/\tau_{\mathrm{E}} = L/\tau$ denote the message generation rate (in msg/cycle) per node, where $1/\tau$ is the emission rate in bit/cycle. Although the performance evaluation carried out using a flit/cycle generation rate would seem more intuitive, we adopt the bit/cycle basis to compare our analysis with Dally's results [6]. We distinguish the following important actions of a communication:

- *Path-hole phase without conflicts.* Once generated, a message is at the entry point of the flow diagram. Each data transmission is subject to an average delay of $(k_0 + k_1 + k_2)/2 - 1$ cycles. This value denotes the average time that the header flit requires to reach without conflicts the node that precedes the destination node.
- *Entrance in a dimension.* Once completed the PH phase, we can see the message flow as divided into three streams. A fraction $(1 - p_0)/\tau_{\mathrm{E}}$ enters the 0th dimension,

a fraction $p_0(1 - p_1)/\tau_E$ enters the 1st dimension without using links of the 0th dimension, while a fraction $p_0 p_1(1 - p_2)/\tau_E$ uses only links of the 2nd dimension. The messages entering a new dimension denote the *main flow* in that dimension. This flow may conflict against the *contention flow* that is formed by messages coming from other source nodes and using the first link of the main flow in that dimension. For example, the contention flow is denoted by δ/τ_E, ε/τ_E, ϑ/τ_E for the 0th, 1st and 2nd dimension, respectively. Theorem 1 gives a relation between main flow and contention flow for each dimension.

- *Flow among the links of a dimension*. This phase concerns the conflict analysis and is discussed in Section 3.2.
- *Exit from a dimension*. At the exit from 0th and 1st dimension, a part of the flow requires links of successive dimensions, while another part reaches the data-trail phase. Since we are considering dimension-ordered routing, the flow leaving the 0th dimension is divided into three parts: a fraction $(1 - p_0)(1 - p_1)/\tau_E$ enters the 1st dimension, a fraction $(1 - p_0)p_1(1 - p_2)/\tau_E$ enters the 2nd dimension, and a fraction $(1 - p_0)p_1 p_2/\tau_E$ reaches the data-trail phase. The flow from the 1st dimension is divided into two parts: a fraction $(1 - p_1)(1 - p_2)/\tau_E$ enters the 2nd dimension and a fraction $(1 - p_1)p_2/\tau_E$ reaches the data-trail phase. The flow from the 2nd dimension has no choice and goes entirely to the data-trail phase.
- *Data-trail phase*. This is the last phase of a message transmission. The header flit has reached the destination node, that is, the message does not need other links, and data flits only have to be transmitted. Therefore, the time to complete this phase is $T_{DT} = L/W$ cycles, where L is the average length of the message in flits and W is the number of wires per link.

3.2. Conflict analysis

We discuss the conflict analysis for the 2nd dimension, however the evaluation of the flows occurring in the other dimensions is analogous. The flow rates are evaluated by means of the *flow conservation law*, that is, in a stable system, all the flow getting into a node is equivalent to the flow leaving that node.

We distinguish three kinds of links: first c_{12}, internal c_{i2}, and last $c_{(k_2/2),2}$. At the conflict barrier, the 2nd dimension has an incoming flow which is given by the sum of the following three flows: the messages that use the links of the 0th and 2nd dimension, while skipping the 1st; the communications that pass at least through the links of the 1st and 2nd dimension using or not the 0th dimension; the messages that transit through the links of the 2nd dimension only. These flows are denoted by the equation $[(1-p_0)p_1(1-p_2) + (1-p_1)(1-p_2) + p_0 p_1(1-p_2)]/\tau_E$ that, after some simple algebra, becomes $(1 - p_2)/\tau_E = \gamma/\tau_E$. When the flow γ/τ_E requires the first link c_{12}, it conflicts against the flow ϑ/τ_E. This flow stands for the messages that have already obtained at least one link of the 2nd dimension and now require the link c_{12}.

Once the flow γ/τ_E has obtained the link c_{12}, a fraction of its communications joins the flow ϑ/τ_E which is directed towards the adjacent link c_{22}. The other fraction joins the flow γ/τ_E. This latter represents the communications that use c_{12} as last link and leave the k_2-ring of the 2nd dimension. The messages that do not use anymore links of the 2nd dimension enter the data-trail phase because they have reached the destination node between the links c_{12} and c_{22}. Conversely, the flows α/τ_E and β/τ_E that leave the k_0- and k_1-ring include both completed communications and the messages that continue their path through the links of other dimensions.

In the attempt of reaching the node c_{22}, the flow ϑ/τ_E conflicts against the messages that requires c_{22} as first link of the 2nd dimension. The analysis concerning the link c_{12} explains that this latter flow is equal to γ/τ_E. An analogous approach can be used to derive the flows at any node $c_{i2}, i = 2, \ldots, k_2/2 - 1$, while a different reasoning should be applied to $c_{k_2/2,2}$. The last node receives a flow γ/τ_E that leaves the 2nd dimension and reach the data-trail phase, and a flow ϑ/τ_E that requires other links of the 2nd dimension. The Theorem 1 determines an important relation between *main* and *contention* flows of a dimension: α/τ_E and δ/τ_E, β/τ_E and ε/τ_E, γ/τ_E and ϑ/τ_E.

Theorem 1. *Any uni-directional flow ξ that requires the first link of the jth dimension, consisting of k_j nodes, conflicts against a flow equal to $\xi k_j/2$.*

Proof. Let us consider a k_j-ring of one dimension of a $k_0 \times k_1 \times k_2$ cube with uni-directional links. Let us denote the nodes of this ring with $(k_j, 1), \ldots, (k_j, k_j)$. Moreover, let us suppose that the flow ξ enters the ring at the node 1, and continues towards the direction $(k_j, k_j), (k_j, k_j - 1), (k_j, k_j - 2), \ldots$ of the ring.

When the flow ξ uses the first link, it conflicts against other messages that have entered the ring at other nodes (different from 1), and now request the link (k_j, 1). We observe that each of these messages uses in average half of the links of the k_j-ring, that is, they cover $k_j/2$ hops. Hence, the (k_j, 1) link is actually requested only by the messages that have entered the jth dimension at one of the following $k_j/2$ nodes: $(k_j, 1), (k_j, 2), \ldots, (k_j, k_j/2)$. The proof comes from the observation that an average flow ξ is associated to each node of the ring. □

By applying the Theorem 1 to the nodes c_{20}, c_{21} and c_{22}, we have that $\delta = \alpha k_0/2$, $\varepsilon = \beta k_1/2$ and $\vartheta = \gamma k_2/2$, respectively.

3.3. Backward flow analysis

We adopt a *backward flow analysis* to evaluate each time value of the diagram in Fig. 1. This analysis is based on the assumption that each time value denotes the average time that has to be passed or, in other words, the time that a message *sees* from a given point of the network until the destination node. For example, any T_X label in Fig. 1 denotes the average time that a message will experience from that point until the completion of its transmission. Proceeding in such a way, we obtain a set of recursive equations that allow us to move from T_{DT}, that is, the time to complete the data-trail phase, back to a closed formula for the average latency time. Under the *backward flow analysis* hypothesis, T_{latency} is the time that a message sees as soon as it enters the network. Moreover, T_{DT} denotes the final phase that can simply be expressed by $T_{\mathrm{DT}} = L/W$, where W denotes the number of wires per link, and L the average length of the message in flits. In Fig. 1, T_j (for $j = 0, 1, 2$) is the time seen at the exit of dimension j, while $T_{i,j}$ (for $i = 1, \ldots, k_j/2, j = 0, 1, 2$) is the time that a message sees after it has obtained the link $c_{i-1,j}$.

The backward flow analysis is based on the law that correlates the time seen before a conflict point to the time seen after this barrier or simple link. To this purpose, let us consider in Fig. 2 a node C with two incoming flows a and b having inter-arrival times equal to τ_{a} and τ_{b}, respectively. In addition, let T_{a} and T_{b} denote the time seen after the collision point by the messages belonging to the former and latter flow.

We are interested to estimate for example the time T_{X} which is related to the flow a. Some observations about wormhole routing simplifies this analysis.

- Once a message has obtained the link C, it holds this link for a mean time equal to T_{a} and T_{b} depending on the flow. Therefore, T_{a} and T_{b} can be considered as the mean service time that the messages of class a and b experience at the server C, respectively.
- As we consider a wormhole transmission, we have not to include in the evaluation of T_{X} any waiting time due to the messages of the same flow. If a message belonging to the flow a reaches the link C, any other message of the flow a has already released this link, and is not anymore a source of conflict. Analogous consideration holds for the other flow. Therefore, either a message of the flow a finds the C free or verifies that is occupied by at most one message of the flow b. In the former instance, the message will see a time T_{a} only. Otherwise, it will experience an additional delay which is equal to the residual time of occupancy of the link C by the message b.

These considerations allow us to write the time T_{X} as follows:

$$T_{\mathrm{X}} = T_{\mathrm{a}} + (\mathrm{prob}\{C \text{ held by } b\})R_{\mathrm{b}} = T_{\mathrm{a}} + \rho_{\mathrm{b}}R_{\mathrm{b}}, \tag{1}$$

where $\rho_{\mathrm{b}} = T_{\mathrm{b}}/\tau_{\mathrm{b}}$ is the *utilization rate* of the server C due to messages of flow b, while R_{b} is the *residual service time* of message b. For a generic distribution of the service time, having mean T_{b} and variance σ_{b}^2, this latter component is given by [18]:

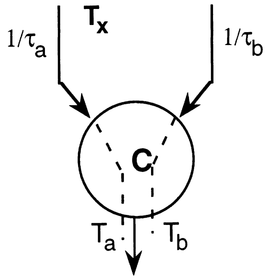

Fig. 2. Example of a link which is requested by two flows.

$$R_b = \frac{T_b}{2} + \frac{\sigma_b^2}{2T_b}. \tag{2}$$

In general, $R_b \geqslant T_b/2$ and the equal sign holds only for $\sigma_b^2 = 0$, that is, when the distribution of the service time is deterministic. However, under the hypothesis that the variance is not very high, Eq. (2) can be finely approximated by the former addend. We will see in Section 6 that this assumption leads to less accurate results only for very high traffic rates. Using $R_b = T_b/2$, Eq. (1) can be approximated by the so-called *conflict equation*:

$$T_X = T_a + \frac{T_b^2}{2\tau_b}. \tag{3}$$

The simplicity of this equation does not prevent us to model even multiple sources of conflicts that occur when a message uses for the first time links of the dimension one and two.

3.4. Estimation of the latency time

The mean latency time is evaluated through a recursive application of the conflict equation. The analysis starts from the node $c_{k_2/2,2}$, continues along the other nodes of the 2nd dimension, then passes to the nodes of the 1st and 0th dimension starting from $c_{k_1/2,1}$ and $c_{k_0/2,0}$, respectively. By applying Eq. (1) to the node $c_{k_2/2,2}$, we have $T_X = T_{k_2/2,2}$, $T_a = T_2, \tau_b = \tau_E/\gamma$, and $T_b = T_{2,2}$. The last equation is motivated by the observation that $T_{2,2}$ is the time seen by the flow γ/τ_E when it leaves the node $c_{1,2}$. We can denote the time seen from the last link of the 2nd dimension as

$$T_{k_2/2,2} = T_2 + \frac{T_{2,2}^2}{2(\tau_E/\gamma)} = T_2 + \frac{\gamma T_{2,2}^2}{2(\tau_E)}. \tag{4}$$

By applying Eqs. (3) and (4) to the node $c_{k_2/2-1,2}$, we can write $T_{k_2/2-1,2}$ as follows:

$$T_{k_2/2-1,2} = T_{k_2/2,2} + \frac{\gamma T_{2,2}^2}{2(\tau_E)} = T_2 + 2\frac{\gamma T_{2,2}^2}{2(\tau_E)}. \tag{5}$$

If we proceed in an analogous way for the other links of the 2nd dimension, we obtain recursively all time values of the 2nd dimension as a function of T_2. For example, the time seen after the node $c_{1,2}$ is

$$T_{2,2} = T_{k_2/2-(k_2/2-2),2} = T_2 + \left(\frac{k_2}{2} - 1\right)\frac{\gamma T_{2,2}^2}{2(\tau_E)}. \tag{6}$$

By recalling that $T_2 = T_{DT}$ and $T_{DT} = L/W$, we can solve the second order equation (6) with respect to $T_{2,2}$ as a function of known values that is,

$$T_{2,2} = \frac{1 - \sqrt{1 - (k_2 - 2)\gamma T_2/\tau_E}}{(k_2 - 2)\gamma/(2\tau_E)}, \tag{7}$$

where we chose the minus sign for consistency reason, that is, only in this way the following relation $\lim_{\tau_E \to \infty} T_{2,2} = T_2$ is true. If we apply Eq. (3) to the node c_{12}, we obtain:

$$T_{1,2} = T_{2,2} + \vartheta \frac{T_{2,2}^2}{2\tau_E}, \tag{8}$$

where $\vartheta = \gamma k_2/2$ on the basis of Theorem 1. It has to be noted that the flow ϑ/τ_E consists of several components that see different times. Therefore, using $T_{2,2}$ as time associated to this flow causes an upper bound approximation. However, we will see that this assumption does not affect much the accuracy of the model.

Once obtained $T_{1,2}$, we can evaluate T_1, that is, the time which a message sees at the exit from the 1st dimension. This time consists of two parts depending whether the flow β/τ_E uses or not links of the 2nd dimension. The latter component is simply given by $p_2 T_{DT}$, while the former can be evaluated through a simple extension of the *conflict equation* applied to three incoming flows instead of two. Hence, T_1 can be written as

$$\begin{aligned} T_1 &= p_2 T_{DT} + (1-p_2)\left[T_{1,2} + (1-p_0)p_1(1-p_2)\frac{T_{1,2}^2}{2\tau_E} + p_0 p_1 (1-p_2)\frac{T_{1,2}^2}{2\tau_E}\right] \\ &= p_2 T_{DT} + (1-p_2)\left[T_{1,2} + p_1(1-p_2)\frac{T_{1,2}^2}{2\tau_E}\right]. \end{aligned} \tag{9}$$

By proceeding analogously to the analysis carried out for the 2nd dimension in Eqs. (2)–(6), we can write $T_{2,1}$ and $T_{1,1}$ as follows:

$$T_{2,1} = \frac{1 - \sqrt{1 - (k_1 - 2)\beta T_1/\tau_E}}{(k_1 - 2)\beta/(2\tau_E)}, \tag{10}$$

$$T_{1,1} = T_{2,1} + \varepsilon \frac{T_{2,1}^2}{2\tau_E}, \tag{11}$$

where $\varepsilon = \beta k_1/2$ as stated by Theorem 1. A similar analysis can be applied to obtain the parameters relative to the 0th dimension,

$$\begin{aligned} T_0 = p_1 p_2 T_{DT} &+ p_1(1-p_2)\left[T_{1,2} + p_0 p_1 (1-p_2)\frac{T_{1,2}^2}{2\tau_E} + (1-p_1)(1-p_2)\frac{T_{1,2}^2}{2\tau_E}\right] \\ &+ (1-p_1)\left[T_{1,1} + p_0(1-p_1)\frac{T_{1,1}^2}{2\tau_E}\right], \end{aligned} \tag{12}$$

$$T_{2,0} = \frac{1 - \sqrt{1 - (k_0 - 2)\alpha T_2/\tau_E}}{(k_0 - 2)\alpha/(2\tau_E)}, \tag{13}$$

$$T_{1,0} = T_{2,0} + \delta \frac{T_{2,0}^2}{2\tau_E} = T_{2,0} + \alpha \frac{k_0 T_{2,0}^2}{4\tau_E}. \tag{14}$$

At this point, we are able to obtain the closed form equation for the average latency time as sum of the time for completing the path-hole phase without conflicts (that is, the first addend between brackets) plus the three times that a message may see at the exit from this phase:

$$T_{\text{latency}} = \left(\frac{k_0+k_1+k_2}{2}-1\right) + (1-p_0)T_{1,0} + p_0(1-p_1)$$
$$\left[T_{1,1} + (1-p_0)(1-p_1)\frac{T_{1,1}^2}{2\tau_{\mathrm{E}}}\right] + p_0p_1(1-p_2)$$
$$\left[T_{1,2} + (1-p_0)p_1(1-p_2)\frac{T_{1,2}^2}{2\tau_{\mathrm{E}}} + (1-p_0)(1-p_1)\frac{T_{1,2}^2}{2\tau_{\mathrm{E}}}\right]. \tag{15}$$

4. The model for torus with bi-directional links

The approach presented in Section 3 is highly flexible. We demonstrate how it can be extended to other architectures by analyzing the performance of dimension-ordered routing in a 3D symmetric torus with bi-directional links. The same methodology can be easily applied even to other torus and hypercube topologies. Because of space limitations, we only point out the main differences between the model for bi-directional links and that for asymmetric torus with uni-directional channels. The basic approach and demonstrations are analogous to those of Section 3, while the main differences concern:

- The average path length in a symmetric *k*-ary 3-cube with bi-directional channels and uniform traffic is equal to 3*k*/4 because the direction which leads to the shortest path is selected. Therefore, the *PH phase* causes a delay equal to $(3k/4) - 1$ cycles. This value denotes the time required by the header flit to reach without conflicts the node that precedes the destination. Fig. 3 shows the part of the torus network that an average message may use traveling along the links of the three dimensions in one of the two directions.
- Since we are considering a 3D network with bi-directional links, the messages leaving a processor may choose among eight paths. For example, by denoting with *l* and *r* the left and right direction on each dimension, we have 0*l*–1*l*–2*l*, 0*l*–1*l*–2*r*, 0*l*–1*r*–2*l*, 0*l*–1*r*–2*r*, 0*r*–1*l*–2*l*, 0*r*–1*l*–2*r*, 0*r*–1*r*–2*l*, 0*r*–1*r*–2*r*. Hence, each path represents only 1/8 of the entire flow which is generated by a processor.
- The flow entering the three dimensions is now denoted by the rates $\alpha = (1-p)/8\tau_{\mathrm{E}}, \beta = (1-p)^2/8\tau_{\mathrm{E}} + p(1-p)/8\tau_{\mathrm{E}} = \alpha$, and $\gamma = p^2(1-p)/8\tau_{\mathrm{E}} + (1-p)^2/8\tau_{\mathrm{E}} + p(1-p)^2/8\tau_{\mathrm{E}} = \alpha = \beta$, where $p = 1/k$ because of the symmetric topology consideration.
- When each of these flows requires the first link of a new dimension, it conflicts against two flows. Let us enter into details by focusing on the 2nd dimension. The flow γ conflicts against the 3γ and $k\gamma$ flows.

The flow 3γ is a fraction of the flow 4γ that is generated by the same processor. It has used the links of the 0th and 1st dimension in an opposite direction with respect to the flow γ, while it follows the same direction of the flow γ for what concerns the 2nd dimension. For example, when the flow of the 0*l*–1*l*–2*l* path enters the second dimension, it conflicts against the flow of the paths 0*l*–1*r*–2*l*, 0*r*–1*l*–2*l*, 0*r*–1*r*–2*l*.

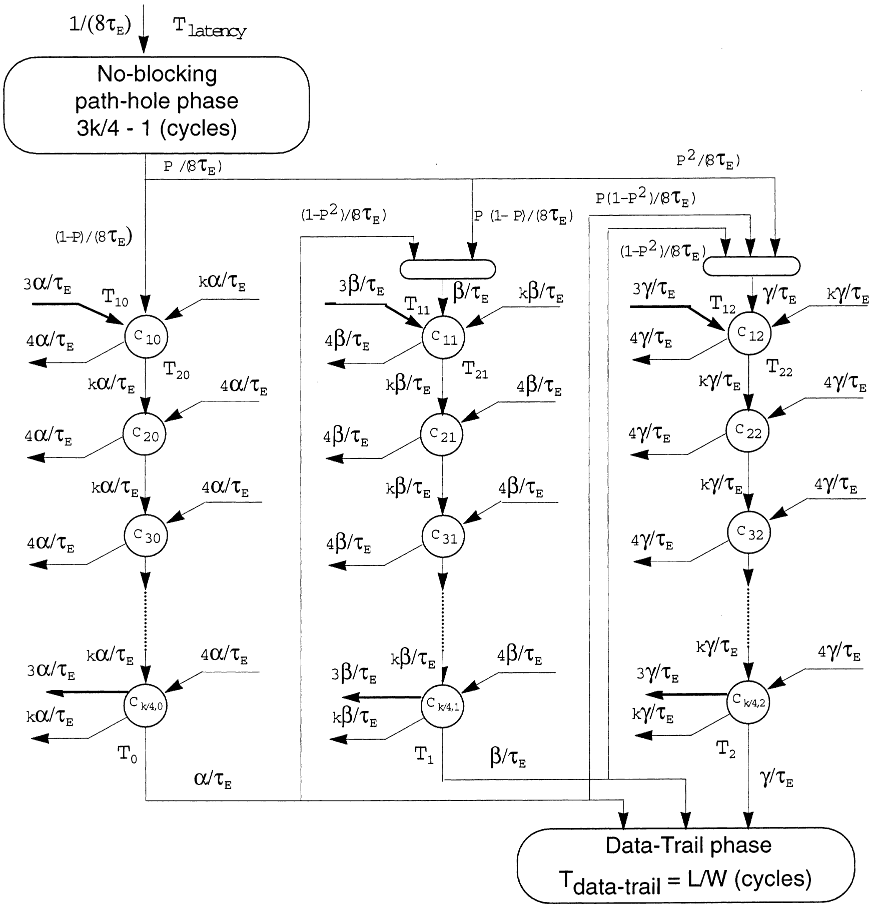

Fig. 3. Network's view of an average message transmission in a k-ary 3-cube, that is, a 3D symmetric torus, with bi-directional links.

The flow $k\gamma$ denotes the messages that are generated by other processors. Analogously to the flow $\vartheta = \gamma k/2$ of the uni-directional network, the flow $k\gamma$ denotes the communications that are in the k-ring (because they have already obtained at least another link in this dimension) and now require the link c_{12}. Theorem 2 estimates this flow as a function of the incoming flow 4γ.

Theorem 2. *Any flow $\Phi + 3\Phi$ that uses the first link of a new dimension, consisting of k nodes, conflicts against a flow equal to $(k/4)4\Phi = k\Phi$.*

The proof is similar to that of Theorem 1, and is omitted for space limitations. Theorem 2 motivates the entire set of flows which are shown in Fig. 3.

Following an approach analogous to that presented in Section 3, we can evaluate any average time that characterizes the wormhole communications in a bi-directional network. We use the same hypothesis about the time values, that is, each time does not denote the completion time but the time that a communication sees from a certain point until the destination. This allows us to use the same *backward flow analysis* that from T_{DT} arrives to estimate T_{latency} in closed form.

We modify Eqs. (7)–(14) of Section 3 starting from the link of the 2nd dimension. Therefore, we have

$$T_{2,2} = T_{k/4-(k/4-2),2} = T_2 + \left(\frac{k}{4} - 1\right)\frac{2\gamma T_{2,2}^2}{\tau_E}, \tag{16}$$

$$T_{2,2} = \frac{1 - \sqrt{1 - 2(k-4)\gamma T_2/\tau_{\mathrm{E}}}}{(k-4)\gamma/\tau_{\mathrm{E}}}, \tag{17}$$

$$T_{1,2} = T_{2,2} + 3\gamma\frac{T_{2,2}^2}{2\tau_{\mathrm{E}}} + k\gamma\frac{T_{2,2}^2}{2\tau_{\mathrm{E}}}, \tag{18}$$

while for the 1st dimension, we have

$$\begin{aligned} T_1 &= pT_{\mathrm{DT}} + (1-p)\left[T_{1,2} + p(1-p)^2\frac{T_{1,2}^2}{16\tau_{\mathrm{E}}} + p^2(1-p)\frac{T_{1,2}^2}{16\tau_{\mathrm{E}}}\right] \\ &= pT_{\mathrm{DT}} + (1-p)T_{1,2} + p(1-p)^2\frac{T_{1,2}^2}{16\tau_{\mathrm{E}}}, \end{aligned} \tag{19}$$

$$T_{2,1} = \frac{1 - \sqrt{1 - 2(k-4)\beta T_1/\tau_{\mathrm{E}}}}{(k-4)\beta/\tau_{\mathrm{E}}}, \tag{20}$$

$$T_{1,1} = T_{2,1} + 3\beta\frac{T_{2,1}^2}{2\tau_{\mathrm{E}}} + k\beta\frac{T_{2,1}^2}{2\tau_{\mathrm{E}}}, \tag{21}$$

and for the 0th dimension,

$$\begin{aligned} T_0 = p^2 T_{\mathrm{DT}} &+ p(1-p)\left[T_{12} + \frac{T_{12}^2}{16\tau_{\mathrm{E}}} + \left(p^2 - p^3 + (1-p)^2\right)\right] \\ &+ (1-p)\left[T_{11} + p(1-p)\frac{T_{11}^2}{16\tau_{\mathrm{E}}}\right], \end{aligned} \tag{22}$$

$$T_{2,0} = \frac{1 - \sqrt{1 - 2(k-4)\alpha T_0/\tau_{\mathrm{E}}}}{(k-4)\alpha/\tau_{\mathrm{E}}}, \tag{23}$$

$$T_{1,0} = T_{2,0} + 3\alpha\frac{T_{2,0}^2}{2\tau_{\mathrm{E}}} + k\alpha\frac{T_{2,0}^2}{2\tau_{\mathrm{E}}}. \tag{24}$$

We can now write the closed form equation for the average latency time in a *k*-ary 3-cube with bi-directional links as:

$$T_{\text{latency}} = \left(\frac{3k}{4} - 1\right) + (1-p)T_{1,0} + p(1-p)\left[T_{1,1} + (1-p)^2 \frac{T_{1,1}^2}{16\tau_E}\right] + p^2(1-p)\left[T_{1,2} + p(1-p)^2 \frac{T_{1,2}^2}{16\tau_E} + (1-p)^2 \frac{T_{1,2}^2}{16\tau_E}\right]. \quad (25)$$

5. The model for hypercube

The model for deterministic wormhole routing in a hypercube is for certain aspects a simplification of the model for the 3D torus. The hypercube is, in fact, a perfect symmetric topology in any dimension, that is, for any n of the 2-ary n-cube. As routing policy, we consider the E-cube algorithm [17] which is the *dimension-ordered* policy applied to hypercube topologies. Since the hypercube has more links than a torus and the E-cube algorithm allows each message to use only a small subset of the links, in a hypercube we have less collisions than in a torus. The example of a 2^3 hypercube in Fig. 4 clarifies this point. Let us consider the messages generated by the processor 000. Once the destination has been chosen, a message can follow a single path. The route is determined by complementing a bit for each new node that has to be reached. For instance, starting from the first bit (we assume the rightmost), a message m_1 from the 000 to the 111 node follows the path 000–001–011–111. It is important to analyze what happens through this route by referring to Fig. 4. The message m_1 does not experience any conflict in acquiring the first link 000–001 because messages that could use that link (e.g., from node 100 to 001), actually follow alternative paths (e.g., 100–101–001). When the message m_1 tries to acquire the second link, it may conflict against the messages that go from 001 to 011 only. Analogously, when m_1 tries to obtain the third link, it conflicts against the messages that go from 001 to 111 and from 011 to 111.

The network view of an average communication in a 2^n hypercube is shown in Fig. 5. As expected, this diagram is highly simplified with respect to the torus diagrams of Figs. 1 and 3. In Fig. 5, the nodes represents a link for each dimension because each message can use only one link in each dimension. The arcs denote the

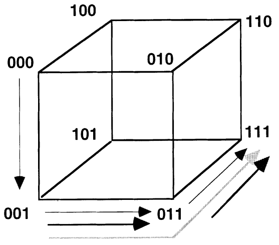

Fig. 4. Link conflicts in a 2^3 hypercube with deterministic routing (E-cube).

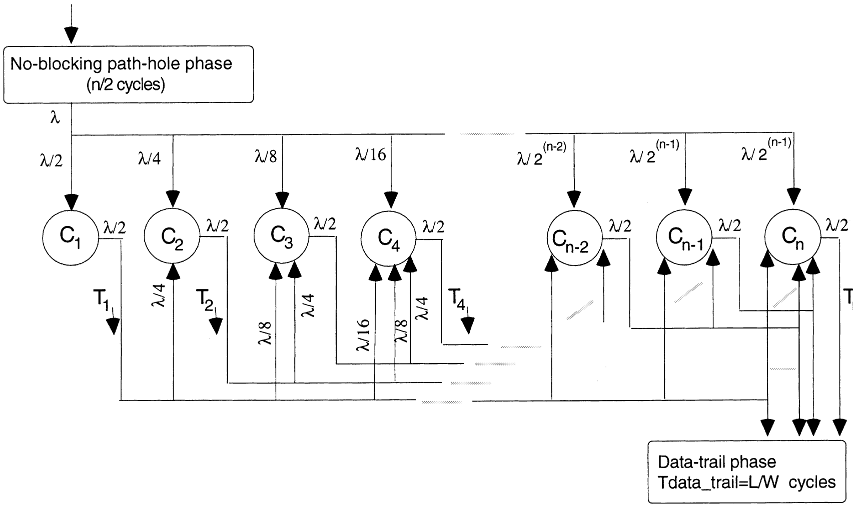

Fig. 5. Network's view of an average message transmission in a 2-ary n-cube (that is, a 2^n hypercube) with uni-directional links.

paths of the flows which are generated by one processor. The label on each arc indicates the value of the flow in that point. We use $\lambda = 1/\tau$. This diagram can be evaluated by following the same approach described in Sections 3 and 4. Since we assume that the destination nodes are uniformly distributed and the routing algorithm is E-cube, the flow generated by a processor uses a link of the first dimension with probability $\frac{1}{2}$ (this is the probability that a message has to complement the rightmost bit), and continues through other dimensions with the same probability of complementing a bit in that dimension. For example, if we observe the left part of the diagram in Fig. 5, when the messages exit from C_1, half of the flow needs to use a link of the second dimension, while the remaining half continues through other dimensions. This is recursively repeated until the last dimension, that is, the n-th bit. It should be noted that this diagram gives another view of the same link conflicts discussed in Fig. 4. For instance, the link C_3 is required by three kind of flows: (1) the messages that uses C_3 as the first link (for example, if C_3 is the link 011–111 in Fig. 4, they correspond to the messages that are generated by the node 011 and go to 111); (2) the messages that uses C_3 as the second link, that is, those generated by the node 001 that go to 111; (3) the messages that uses C_3 as the third link, that is, those generated by the node 000 that go to 111.

Therefore, for the analysis of the flows in Fig. 5 we can apply the same *conflict equation* (3) discussed in Section 3. Moreover, we adopt the backward flow analysis and the assumption that each time value of the diagram denotes the time seen from that point until the end of the data transmission. In particular, we have that $T_n = (1/2^0)T_{\mathrm{DT}} = T_{\mathrm{DT}} = L/W$, while for evaluating the other time values we repeatedly apply the conflict equation. Therefore, taking into account that $T_{\mathrm{a}} = T_n$ while T_{b} is the sum of all the conflicts that affect the node C_n, we can write

$$\begin{aligned} T_{n-1} = {} & (1/2^1)T_n + [T_n + (1/2^n + 1/2^{n-1} + 1/2^{n-1} + \cdots + 1/2^3)T_n^2/(2\tau)]/2 \\ = {} & (1/2)T_n + (1 + 1/2^{n-2})T_n^2/(2^4\tau) + T_n/2. \end{aligned} \tag{26}$$

Proceeding in an analogous way, we can obtain all T_i values for $i = n-2, \ldots, 1$. The generic formula is:

$$T_i = T_\Sigma + T_n^2/(2^{2n+1-i}\tau) + T_n/2^{2n-i}, \tag{27}$$

where

$$T_\Sigma = \sum_{j=i+2}^{n} T_{i+1}^2/(2^4\tau) + T_{i+1}/2 + \left(1 + \frac{1}{2} + \cdots + \frac{1}{2^{j-i-1}}\right)T_j^2 \Bigg/ (2^{j-i+3}\tau) + T_j/2^{j-i}.$$

The sum of all flows reaching each link but the last is equal to $\lambda/2 = 1/(2\tau)$. The most important result is that each time value can be expressed as a function of parameters which are previously evaluated. Hence, we can write the following closed formula for the average latency time in a hypercube with wormhole routing:

$$\begin{aligned} T_{\text{latency}} = {} & [(1/2)T_1] + \left[(1/2^2)T_2 + T_2^2/(2^3\tau)\right] + \cdots + \left[(1/2^{n-1})(T_{n-1} + (1 + 1/2 \right. \\ & \left. + \cdots + 1/2^{n-3})T_{n-1}^2/(2^3\tau))\right] + \left[(1/2^{n-1})(T_{n-1} + (1 + 1/2 \right. \\ & \left. + \cdots + 1/2^{n-2})T_n^2/(2^3\tau))\right] + n/2. \end{aligned} \tag{28}$$

6. Validation of the models

In this section we validate the analytical models with uni-directional and bi-directional links, and use them to obtain some interesting performance parameters. We compare the results of the former model against the analytical and simulation results presented by Dally in [6].

Table 2 shows the average latency time as a function of the traffic $1/\tau$ (expressed in bit/cycle) in a 16-ary 3-cube. Our analytical values are obtained by setting $k_0 = k_1 = k_2 = 16$ and $\tau_E = \tau/L$ in Eq. (15). These results are very close to the simulation values: the differences are below 5% for most of the points and reach 10% only for very high traffic rates. Moreover, Table 2 shows that our approach improves the good results achieved by the Dally analytical model.

Table 2
Comparison of estimated message latency times with analytical and simulation results proposed by Dally in [6] (uni-directional 16 × 16 × 16 torus, single-flit buffers, $L = 200$ bits, $W = 8$, average message length equal to 25 flits)

$1/\tau$	Model	% Error	Dally's simulation [6]	Dally's analysis [6]	% Error
0.05	52	+1.9	51	52	+1.9
0.10	56	+1.8	55	57	+3.6
0.15	63	+3.2	61	66	+8.2
0.20	73	+4.2	70	78	+11.4
0.25	92	+9.5	84	90	+7.1
0.29	133	−10.1	148	125	−15.5

As expected, the approximations introduced in our model cause an error that is almost always positive. Only for very high traffic levels, the error increases and becomes negative. This is due to the approximation in Eq. (3): in a congested networks the variance of the waiting times augments, and the residual time is typically higher than the $T/2$ value which we used in this *conflict equation*. Therefore, we believe that the accuracy of our model can be even improved by approximating the residual time through a more precise parameter that takes into account even the network utilization. However, this study is beyond the scope of this paper.

To validate the analytical model for the 3D torus with bi-directional links, we utilized our discrete event simulator because this model was not yet solved analytically. The simulation results were obtained by means of the Independent Replications Method implemented in Simula language, with confidence intervals at a 95% level of confidence.

Table 3 reports the analytical and simulation latency times for wormhole routing in a 3D torus with 216 nodes, that is, 6 × 6 × 6. The message generation is modeled as a Poisson process with $1/\tau_E$ rate (in msg/cycle) per node. The destination node is uniformly chosen, while the message length is exponentially distributed with mean equal to 12 flits. Table 3 demonstrates the accuracy of the analytical approach for any message transmission rates, even when the network traffic is close to saturation. The errors are under 5% in most of the instances. We have to consider a message generation rate close to 0.04 per node to experience an error that overcomes 10%.

To validate the model for the hypercube, we can again use the simulation results presented by Dally in [6]. Table 4 shows the average latency time as a function of the traffic $1/\tau$ (bit/cycle) in a 2-ary 10-cube. The analytical values, which are obtained by setting $\tau_E = \tau/L$ in Eq. (28), are close to simulation values. The differences overcome 10% only for very high traffic rates. Analytical results of Dally are very similar to ours. It has to be noted that for $1/\tau = 0.5$ the network is saturated. The approximations introduced in our model always cause a negative error. This means that in a hypercube topology the residual time is usually higher than the $T/2$ value which we used as approximation of the *conflict equation* (3).

Table 3
Comparison of estimated message latency time with simulation (bi-directional 6 × 6 × 6 torus, single-flit buffers, $L = 96$ bits, $W = 8$)

$1/\tau_E$	Model	Simulation	% Error
0.001	15.66	15.77	−0.7
0.002	15.88	16.02	−0.9
0.005	16.57	16.87	−1.8
0.010	17.90	18.42	−2.9
0.016	20.33	21.16	−4.1
0.02	21.69	23.16	−6.8
0.04	35.64	40.06	−12.4

Table 4
Comparison of estimated message latency times with Dally's simulation [6] (uni-directional 2^{10} hypercube, single-flit buffers, $L = 200$ bits, $W = 1$)

$1/\tau$	Model	Simulation [6]	% Error
0.05	212	214	−1.2
0.10	220	224	−1.8
0.20	237	246	−3.7
0.30	257	272	−5.6
0.35	268	292	−8.2
0.40	280	312	−10.4
0.45	293	342	−14.5

7. Performance results

We can use the analytical models to evaluate the performance of deterministic wormhole routing in various network topologies and scenarios. The values of this section are obtained by considering $1/\tau_E$ as the message generation rate.

Fig. 6 shows the sensitivity of the latency time to the distribution of the links in each of the three dimensions of a torus with 10^3 nodes and uni-directional links. The symmetric topologies achieve best performance and this occurs even in the (not shown) case of bi-directional links. On the other hand, if we exclude unrealistic

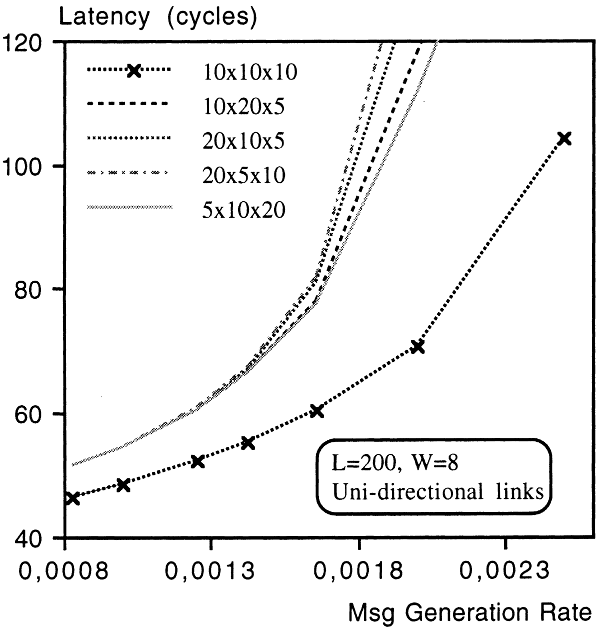

Fig. 6. Comparison of asymmetric and symmetric topologies (3D torus with 1000 nodes and uni-directional links).

unbalanced topologies such as 2 × 5 × 100 or 4 × 32 × 32, we do not see big differences among the performance of asymmetric networks. In the uni-directional instance, the *dimension-ordered* routing performs slightly better for the topologies that adopt an increasing ordering of their dimensions, that is $k_0 \leqslant k_1 \leqslant k_2$. The results achieved for the 5 × 10 × 20 topology were confirmed even by experiments relative to other dimensions. Conversely, for network with bi-directional channels, the advantages of an ordering of the dimensions are less evident. The decreasing ordering (32 × 16 × 8) performs often as the best, however other experiments for different message lengths indicate that even the increasing dimension-ordered may be preferable.

Fig. 7 shows the performance of a 3D symmetric torus with uni-directional and bi-directional links as a function of the message generation rate for three message lengths ($L = 96$, 200, 400, and $W = 8$). The advantages of the topologies with bi-directional links are especially evident for medium-large messages. Fig. 8 shows analogous results for a uni-directional hypercube with the same number of nodes as the torus of Fig. 7. A comparison between these figures points out that, for large architectures and small-medium messages, the latency time of wormhole routing in a hypercube is comparable to that in a 3D torus with bi-directional channels.

Figs. 9 and 10 estimate the influence of the network dimension on the communication performance in 3D torus and hypercube topology, respectively. This analysis is carried out in the case of low, medium, and high traffic conditions, that is,

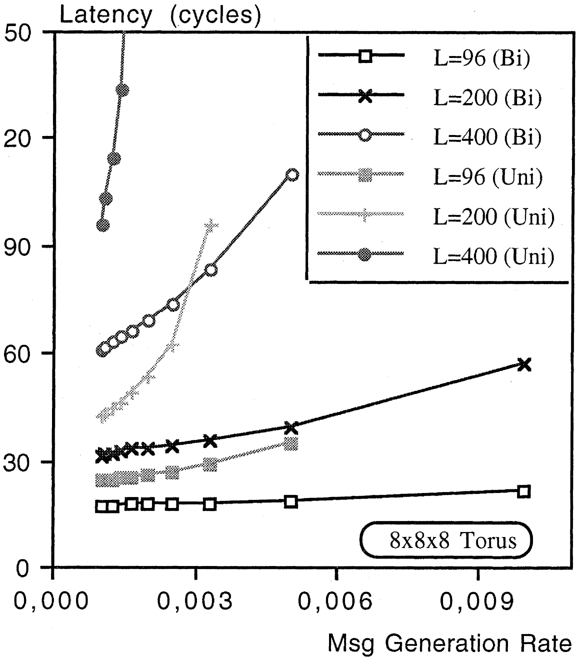

Fig. 7. Performance of various message lengths in uni- and bi-directional 3D tori.

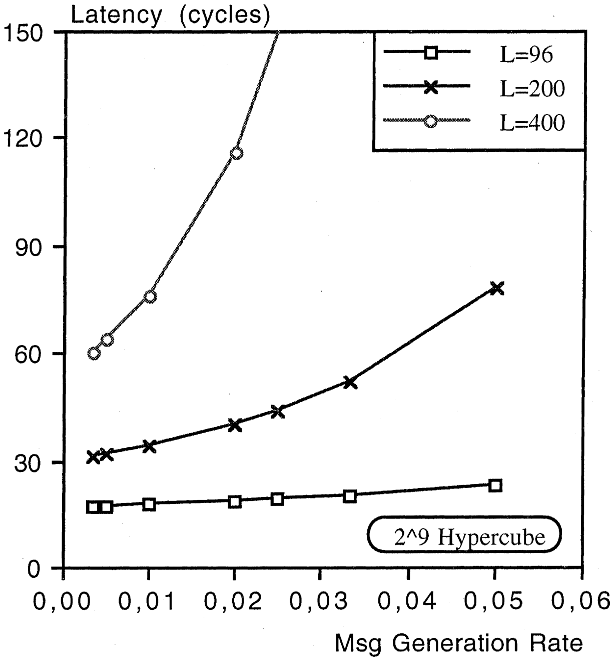

Fig. 8. Performance of various message lengths in uni-directional hypercube.

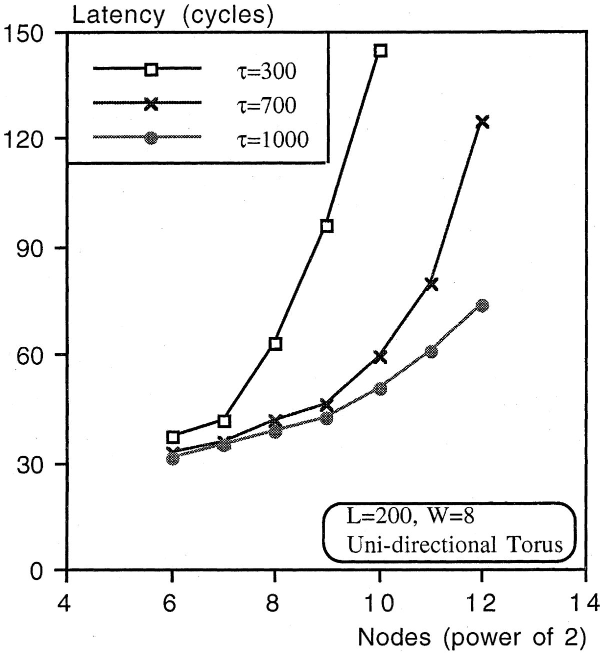

Fig. 9. Performance of wormhole routing in 3D tori with different dimensions for three traffic conditions.

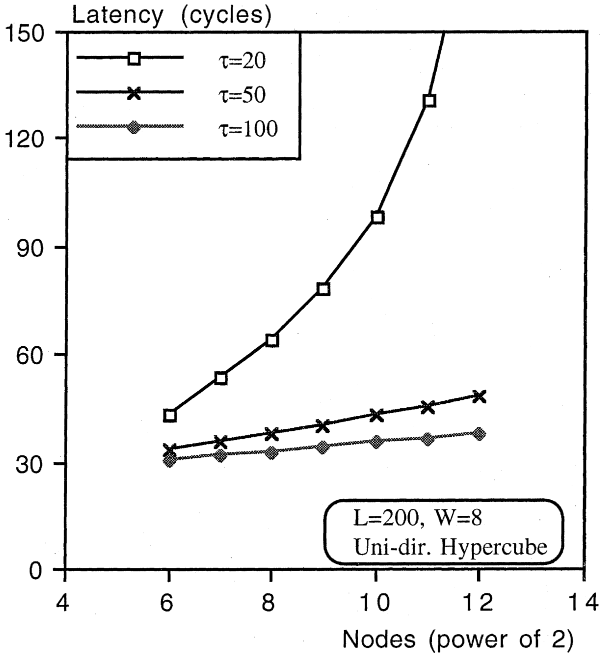

Fig. 10. Performance of wormhole routing in hypercube with different dimensions for three traffic conditions.

$\tau_E = 1000$, 700 and 300 for the torus, and $\tau_E = 100$, 50 and 20 for the hypercube. It has to be observed that in a 3D torus with uni-directional links a message generation rate of $\tau_E = 300$ per node is acceptable only for networks containing less than 300 processors, while in a hypercube a much higher rate, such as $\tau_E = 50$ per node, does not saturate the network even when the number of processors is very large.

8. Conclusions

We propose a new approach for modeling deterministic (*dimension-ordered*) wormhole routing in symmetric and asymmetric k-ary n-cubes. The methodology is based on the *backward flow analysis* and can be considered a unifying approach to achieve closed formulas for the mean latency time in any k-ary n-cube with wrap-around connections and uni-directional or bi-directional links.

In this paper we apply this methodology to the hypercube and two topologies that were not yet solved through closed analytical formulas: an asymmetric $k_0 \times k_1 \times k_2$ torus with uni-directional links, and a symmetric k-ary 3-cube with bi-directional links. Moreover, the same approach can be applied to evaluate the performance of wormhole routing in a wide set of topologies such as hypercubes, symmetric and asymmetric tori, with both uni-directional and bi-directional links.

The assumptions which we used to render the model analytically tractable are generally accepted in literature. Moreover, our approach has been validated by means of simulations that demonstrated the accuracy of the analytical results. As side contribution of this paper, we also verified that our model is slightly more accurate than Dally's model. Moreover, preliminary results of our research demonstrate that the flexibility of this approach is a fine basis for analyzing even the performance of adaptive wormhole routing.

References

[1] V.S. Adve, M.K. Vernon, Performance analysis of mesh interconnection networks with deterministic routing, IEEE Trans. Parallel and Distributed Systems 5 (3) (1994) 225–246.

[2] R.V. Boppana, S. Chalasani, A comparison of adaptive wormhole routing algorithms, in: Proceedings of 20th Annual International Symposium on Computer Architecture, San Diego, May 1993.

[3] A.A. Chien, J.H. Kim, Planar-adaptive routing: low-cost adaptive networks for multiprocessors, J. ACM 42 (1) (1995) 91–123.

[4] M. Colajanni, A. Dell'Arte, B. Ciciani, Communication switching techniques and link-conflict resolution strategies: A comparison analysis, in: Proceedings of EUROSIM'95, Vienna, September 1995.

[5] W.J. Dally, C.L. Seitz, Deadlock-free message routing in multicomputer networks, IEEE Trans. Comput. C-36 (5) (1987) 547–553.

[6] W.J. Dally, Performance analysis of *k*-ary *n*-cube interconnection networks, IEEE Trans. Comput. 39 (6) (1990) 775–785.

[7] L. Gravano, G.D. Pifarré, P.E. Berman, J.L.C. Sanz, Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks, IEEE Trans. Parallel and Distributed Systems 5 (12) (1994) 1233–1251.

[8] D.C. Grunwald, D.A. Reed, Networks for parallel processors: measurements and prognostications, in: Proceedings of 3rd Conference on Hypercube Concurrent Computers and Applications, New York, January 1988, pp. 610–619.

[9] K. Hwang, Advanced Computer Architecture, McGraw-Hill, 1993.

[10] P. Kermani, L. Kleinrock, Virtual cut-through: A new computer communication switching technique, Computer Networks 3 (1979) 267–286.

[11] J. Kim, C.R. Das, Hypercube communication delay with wormhole routing, IEEE Trans. Comput. 43 (7) (1994) 806–814.

[12] C.A. Lamanna, W.H. Shaw, A performance study of the hypercube parallel processor architecture, Simulation 53 (3) (1991) 185–196.

[13] D. Linder, J.C. Harden, An adaptive and fault tolerant wormhole routing strategy for *k*-ary *n*-cubes, IEEE Trans. Comput. 40 (1) (1991) 2–12.

[14] L.M. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, IEEE Comput. 26 (2) (1993) 62–76.

[15] S. Ramany, D. Eager, The interaction between virtual link flow control and adaptive routing in wormhole networks, in: Proceedings of ACM International Conference on Supercomputing, Manchester, UK, July 1994, pp. 136–145.

[16] D.S. Reeves, E.F. Gehringer, A. Chandiramani, Adaptive routing and deadlock recovery: A simulation study, in: Proceedings of 4th Hypercube Conference, 1989, pp. 331–337.

[17] H. Sullivan, T.R. Bashkow, A large scale, homogeneous, fully distributed parallel machine, in: Proceedings of 4th Symposium on Computer Architecture, March 1977, pp. 105–124.

[18] K.S. Trivedi, Probability and Statistics with Reliability, Queueing and Computer Science Applications, Prentice-Hall, Englewood Cliffs, NJ, 1982.