ELSEVIER

Parallel Computing 28 (2002) 485–501

PARALLEL COMPUTING

www.elsevier.com/locate/parco

Performance analysis of adaptive wormhole routing in a two-dimensional torus

F. Quaglia [a,*], B. Ciciani [a], M. Colajanni [b]

[a] *Dip. di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy*
[b] *Dip. di Scienze dell'Ingegneria, Università di Modena, Via Campi, 41100 Modena, Italy*

Received 20 November 1999; received in revised form 3 May 2000; accepted 30 August 2001

Abstract

This paper presents an analytical evaluation of the performance of adaptive wormhole routing in a two-dimensional torus. Our analysis focuses on minimal and fully adaptive wormhole routing that allows a message to use any shortest path between source and destination. A validation of the analysis through simulation is presented to demonstrate the accuracy of the obtained results. Finally, we remark that no theoretical limitation prevents the extension of our analytical approach to the evaluation of the performance of adaptive wormhole routing in hypercubes or other symmetric topologies with wrap-around connections.

Keywords: Message passing; Multicomputers; Performance analysis; Torus topology; Wormhole

1. Introduction

Wormhole routing is the most adopted and efficient technique for communications in parallel machines [28]. This policy makes the latency insensitive to the diameter of the interconnection network in case of light traffic and allows the reduction of the channel buffer size. Even if commercial multicomputers employ wormhole combined with *deterministic* routing, several studies have demonstrated the usefulness of *adaptive* strategies [5,6,13,15,22]. Deterministic routing does not have the ability to

* Corresponding author.
E-mail addresses: quaglia@dis.uniroma1.it (F. Quaglia), ciciani@dis.uniroma1.it (B. Ciciani), colajanni@unimo.it (M. Colajanni).

0167-8191/02/$ - see front matter
PII: S0167-8191(01)00146-6

cope with dynamic network conditions such as faults and congestion because the path between source and destination is determined statically. These drawbacks are avoided by adaptive routing that, in case of channel unavailability, looks for alternative paths.

In this paper we propose an analytical approach to evaluate performance of minimal and fully adaptive wormhole. We present the model for a two-dimensional torus with bi-directional links. However, our analysis can be extended to other symmetric k-ary n-cubes with wrap-around connections (especially with low dimensional topologies which were demonstrated to achieve best performance [2,10]).

In the literature, most of the results about adaptive wormhole have been obtained through simulation studies (e.g., [6,15,16,20,22]), whereas analytical models are usually restricted to deterministic wormhole [1,7,8,10,12,19] or to different techniques such as *circuit-switching* [9] and *virtual-cut-through* [21,24]. To the best of our knowledge there are two papers related to our work [23,26], which present analytical evaluations for the case of adaptive wormhole routing.

Finally, we note that the dynamic choice of paths proper of adaptive routing makes an analytical treatment of the wormhole technique very difficult. Many equations of our performance model show mutual dependencies that prevent a closed form solution and motivate the recursive form equation for the mean latency time.

The remainder of the paper is organized as follows. In Section 2 we describe the operational features of minimal and fully adaptive wormhole routing in a two-dimensional torus. In Section 3 we present the model. In Section 4 we propose a solution for the estimation of the mean latency time. In Section 5 we validate the analytical results against time values obtained from simulations.

2. Routing techniques

Three main techniques transmit packets while building the path from the source to the destination processor: *store-and-forward*, *virtual-cut-through*, and *wormhole*. The store-and-forward technique gathers the entire message at each intermediate node before asking for another channel. Virtual-cut-through, proposed by Kermani and Kleinrock [18], aims at reducing the transmission time. It partitions a message in *flits* and implements a pipeline technique for the transmission: upon getting a channel, the *header flit* tries to get another channel while the *data flits* are transmitted through the already obtained channels. This strategy requires message buffering only in the case of channel unavailability.

Latest generation multicomputers adopt *wormhole* routing [3,25] which, in the absence of channel contention, behaves like virtual-cut-through. However, when contention occurs, wormhole does not gather all the flits into the buffer of the last reached node, but stores them in the flit buffers of the nodes along the already established path. A channel is released only after the last flit (*tail flit*) of the message is transmitted through it.

Unlike deterministic routing that, in the case of channel contention, blocks the message transmission, adaptive wormhole routing looks for alternative paths with-

out releasing the already obtained channels. Deadlock is avoided by using multiple *virtual channels* on each physical link and forcing a pre-defined allocation order of virtual channels to messages [3,17,25]. An adaptive policy that adheres to minimal path length is called *minimal*, whereas *non-minimal* policies, such as the turn model [15] and dimension reversal routing [11], allow a message to use non-shortest paths. Moreover, wormhole routing is called *fully adaptive* if a message is allowed to follow any path of the minimal (or non-minimal) class; it is called *partially adaptive* when a message can use only a subset of the paths of a class. In this paper, we refer to minimal and fully adaptive wormhole, such as *planar-adaptive routing* [6], which allows a message to use any shortest path from the source to the destination. In a two-dimensional torus, minimal fully adaptive wormhole routing works as follows:

- The header flit moves along the X dimension until it reaches the column of the destination processor. Then, it proceeds along the channels of the Y dimension. If there is no channel contention, the message is transmitted just as in dimension ordering routing [25].
- If a channel along X is busy, the header flit tries to use a channel along Y. Then the header tries again to use a channel along X until it reaches the column of the destination processor.
- If channels are busy along both dimensions, the header flit waits until one of them is released.

3. System model and basic assumptions

Each node (i, j) of the two-dimensional torus consists of a *processing element* $P_{i,j}$ that generates and receives messages, and a *router node* $N_{i,j}$ with a controller per each dimension. Nodes are connected through bi-directional full-duplex links. [1] The flit size (bits) is equal to the number of wires B of a link. Therefore, each flit is transmitted in a single cycle link time. This time represents the basic temporal unit of our analysis.

To obtain an analytically tractable model we assume that generation times of messages at each processor are independent and identically distributed in accordance with a Poisson process with rate $1/\tau_{\text{bit}}$ (bit/cycle). This assumption and the symmetry of torus topology guarantee that the network is *balanced* [18] that is, we can assume that channels are equally likely to be visited independently of the message destination distribution. In this paper, the analysis is presented under the hypothesis that the destination is an uniformly distributed random variable. However, there is no theoretical limitation that prevents the extension of the analysis to other distributions that provide regular traffic (this can be done by simply modifying the average path length). The analytical approach prevents the study of other realistic traffic patterns, such as those analyzed in [20].

[1] The terms *link* and *channel* are used interchangeably.

As in Dally's approach [10], we assume that the network has no virtual channels. Actually, each physical link of a two-dimensional torus must have at least three virtual channels to guarantee deadlock-freedom of adaptive wormhole transmissions [6,14]. This assumption introduces an approximation because contentions are solved at the channel boundary level instead of the flit buffer boundary. However, the validation of our model against a simulation model with four virtual channels shows that the analytical results are quite accurate. The effects of this assumption on performance arise only when the message generation rate is close to the network saturation point because, as expected, the model tends to anticipate saturation. Also, the

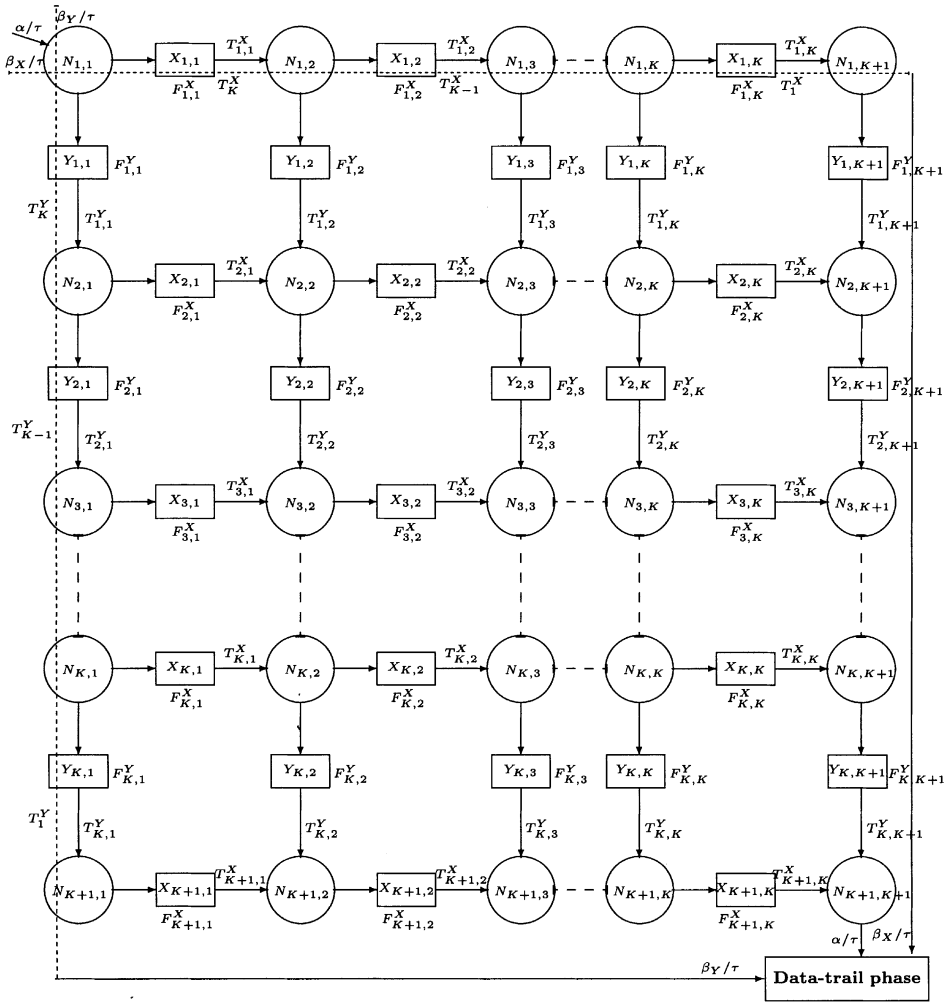

Fig. 1. Communication diagram for an average message.

analysis might be extended to include the effects of virtual channels using the approaches presented in [4,23,27] as a basis.

Our analysis sees the wormhole message transmission as consisting of two consecutive and separate phases: *path-hole* and *data-trail*. During the path-hole phase, the *header flit* selects the path from the source to the destination. All delays due to channel contentions are taken into account in this phase. The data-trail phase models the transmission of data flits along the channels of the selected path. Since we are assuming uniformly chosen destinations, the average length of the path along each dimension is $K = k/4$, where k is the number of nodes of each dimension. We define *average message* a message that travels on exactly K channels in each dimension.

Due to network symmetry and bi-directional links, it is possible to partition the torus into four quadrants (north–east, north–west, south–west, south–east), with the same number of nodes, such that the path of an average message belongs entirely to one quadrant. Our analysis focuses on the south–east quadrant. This quadrant is depicted in Fig. 1, where router nodes (i.e., circles) and channels (i.e., rectangular boxes) are represented through a two-dimensional array notation that identifies their position with respect to the average message's view that is, $N_{1,1}$ is the first router passed through by the average message, $X_{1,1}$ and $Y_{1,1}$ are the first horizontal and vertical channels available to that message, respectively. Note that Fig. 1 describes only the part of the network that would be crossed by the average message.

Since each message can choose among four directions, the flow considered in the analysis represents $1/4$ of the entire flow generated by each node. So, let $1/\tau_E = 1/(L\tau_{bit})$ be the message generation rate (messages/cycle) per each node, where $1/\tau_{bit}$ is the emission rate in bit/cycle, and L is the message average length. The average flow considered in the analysis has generation rate $\tau = 4\tau_E$.

In minimal and fully adaptive wormhole, only the messages that have to change both dimensions to reach the destination can use adaptive path selection. All the other messages must follow a deterministic path. Hence, we can identify three flow streams:

- the stream β^X denoting messages that use exclusively channels of the dimension X;
- the stream β^Y denoting messages that use exclusively channels of the dimension Y;
- the stream $\alpha = 1 - (\beta^X + \beta^Y)$ denoting messages that can use adaptive path selection along both dimensions.

As the traffic is uniform, the value of each stream can be estimated through the ratio between the number of nodes reachable by that stream and the total number of nodes:

$$\alpha = (k-1)^2/(k^2-1) = (k-1)/(k+1), \tag{1}$$

$$\beta^X = \beta^Y = (k-1)/(k^2-1) = 1/(k+1). \tag{2}$$

In Fig. 1 plain lines denote the paths of the α stream, horizontal dotted lines refer to the β^X stream, and vertical dotted lines refer to the β^Y stream. Time values and flow rates associated with a channel of the α stream are identified by the same labels

of the channel. For example, $F^X_{i,j}$ represents the flow rate of the α stream that uses $X_{i,j}$. Moreover, $T^X_{i,j}$ denotes the *residual transmission time* from $X_{i,j}$ that is, the mean time the header flit takes to get from $X_{i,j}$ to the destination. Analogous notation is used for a vertical channel $Y_{i,j}$.

A single parameter is sufficient for the identification of the residual transmission time of the streams β^X and β^Y, because they use channels along one dimension. For example, T^X_{K-j+1} refers to the channel $X_{1,j}$ which is used by β^X, and T^Y_{K-i+1} refers to the channel $Y_{i,1}$ which is used by β^Y. The sub-index denotes the number of router nodes the β stream has yet to cross.

4. The analysis

The evaluation of the mean latency time is carried out through a mean flow analysis. We proceed along the following steps. In Section 4.1 we determine the equations for all the flows in Fig. 1. Then in Section 4.2 we model each link as an M/G/1 queue with multiple classes so that we can give the equations for the mean waiting time a message experiences at each link. The solution of these equations requires an evaluation of the utilization time of each link that, in turn, depends on the residual transmission time which will be evaluated in Section 4.3. Finally, in Section 4.4 we estimate the probabilities of link contention and define the equation for the mean latency time.

4.1. Flow rates

When a message of the α stream reaches a node $N_{i,j}$, it attempts to continue along the dimension X. If the channel $X_{i,j}$ is not available (this happens with probability p_X), then the message tries to continue along the dimension Y. If the channel $Y_{i,j}$ is busy as well (this happens with probability p_Y), then the message waits until one channel is released. The evaluation of p_X and p_Y is postponed to Section 4.4.1. Now, we estimate the flow rates generated by the α stream using the *flow conservation law*. To this purpose, let us introduce the following theorem:

Theorem 1 (bifurcation rule). *Let F be the adaptive flow reaching a node, F^X and F^Y be the horizontal and vertical flows exiting from that node, respectively. We have $F^X = F(1-p_X)/(1-p_Xp_Y)$ and $F^Y = Fp_X(1-p_Y)/(1-p_Xp_Y)$.*

Proof. The flow exiting from a node consists of two components: F and H. H denotes the flow of messages previously queued due to channel unavailability along both dimensions. Therefore, $H = (F+H)p_Xp_Y$ and, after some algebra, $H = Fp_Xp_Y/(1-p_Xp_Y)$. Combining previous equations, we have

$$F + H = F/(1-p_Xp_Y). \tag{3}$$

F^X and F^Y are the fractions of $F+H$ that find the horizontal and vertical channel not busy, respectively. Hence, $F^X = (F+H)(1-p_X)$ and $F^Y = (F+H)p_X(1-p_Y)$. Therefore, from (3) we can write

$$F^X = F(1-p_X)/(1-p_Xp_Y) = Ff^X, \tag{4}$$

$$F^Y = Fp_X(1-p_Y)/(1-p_Xp_Y) = Ff^Y. \quad \square \tag{5}$$

By applying the *bifurcation rule* to the α stream we obtain the equations for all the flows of the diagram in Fig. 1. We partition the horizontal channels into six classes. Each of them is affected by the same type of adaptive flow: the first channel $X_{1,1}$; the first row $X_{1,j}$ but the first channel; the first column $X_{i,1}$ but the first and the last channels; the intermediate channels $X_{i,j}$ but the first row, the first column and the last row; the last channel of the first column $X_{K+1,1}$; the last row $X_{K+1,j}$ but $X_{K+1,1}$. Using Theorem 1 and Fig. 1, we obtain the flow rates associated with each class of channels

$$F^X_{1,1} = f^X\alpha/\tau, \tag{6}$$

$$F^X_{1,j} = F^X_{1,j-1}f^X = (f^X)^j\alpha/\tau \quad (j = 2,3,\ldots,K), \tag{7}$$

$$F^X_{i,1} = F^Y_{i-1,1}f^X = (f^Y)^{i-1}f^X\alpha/\tau \quad (i = 2,3,\ldots,K), \tag{8}$$

$$F^X_{i,j} = F^X_{i,j-1}f^X + F^Y_{i-1,j}f^X \quad (i,j = 2,3,\ldots,K), \tag{9}$$

$$F^X_{K+1,1} = F^Y_{K,1} = (f^Y)^K\alpha/\tau, \tag{10}$$

$$F^X_{K+1,j} = F^X_{K+1,j-1} + F^Y_{K,j} \quad (j = 2,3,\ldots,K). \tag{11}$$

Similarly, we partition the vertical channels into the following six classes: the first channel $Y_{1,1}$; the first row $Y_{1,j}$ but the first and last channels; the last channel $Y_{1,K+1}$ of the first row; the first column $Y_{i,1}$ but the first channel; the intermediate channels $Y_{i,j}$ but the first row, the first and the last columns; the last column $Y_{i,K+1}$ but the first channel. The flow rates associated with each class can be written as

$$F^Y_{1,1} = f^Y\alpha/\tau, \tag{12}$$

$$F^Y_{1,j} = F^X_{1,j-1}f^Y = (f^X)^{j-1}f^Y\alpha/\tau \quad (j = 2,3,\ldots,K), \tag{13}$$

$$F^Y_{1,K+1} = F^X_{1,K} = (f^X)^K\alpha/\tau, \tag{14}$$

$$F^Y_{i,1} = F^Y_{i-1,1}f^Y = (f^Y)^i\alpha/\tau \quad (i = 2,3,\ldots,K), \tag{15}$$

$$F^Y_{i,j} = F^X_{i,j-1}f^Y + F^Y_{i-1,j}f^Y \quad (i,j = 2,3,\ldots,K), \tag{16}$$

$$F^Y_{i,K+1} = F^X_{iK} + F^Y_{i-1,K+1} \quad (i = 2,3,\ldots,K). \tag{17}$$

There exist mutual dependencies among (6)–(17). Once estimated the probabilities of link contention (see Section 4.4.1), these equations can be solved analyzing the

flow of each channel starting from $X_{1,1}$, and then following the west–east and north–south directions.

4.2. Model of a link

We model each link as an M/G/1 queue with multiple classes of flow indexed $m = 1, 2, \ldots, M$. Assuming that class m messages arrive with Poisson rate λ_m and require a mean service time equal to $\overline{T}_m$, we have the following expression for the mean waiting time to get that channel [29, p. 276]:

$$\overline{W} = \sum_{m=1}^{M} \frac{\rho_m}{\rho} E[W_m] = \frac{1}{2(1-\rho)} \sum_{m=1}^{M} \lambda_m \left(\overline{T}_m^2 + \sigma_m^2 \right), \tag{18}$$

where σ_m^2 denotes the variance of $\overline{T}_m$, and $\rho = \sum_{m=1}^{M} \lambda_m \overline{T}_m$. The service time $\overline{T}_m$, corresponding to the link utilization time for the flow of class m, will be evaluated in Section 4.2.2.

4.2.1. Mean waiting time

The analysis is carried out focusing on messages that travel from $N_{1,1}$ to $N_{K+1,K+1}$ following the west–east and north–south directions that is, in the south–east quadrant. Hence, the waiting times of interest are W_{WE} and W_{NE} for the horizontal channels, and W_{NS} and W_{WS} for the vertical channels. In particular, we give a detailed estimation of W_{WE} for a generic horizontal channel $X_{i,j}$. The other waiting times are obtained through analogous considerations.

As shown by (18), to obtain W_{WE} we have to determine the M classes of flow that use $X_{i,j}$. Depending on the source and destination, that channel may represent the first horizontal channel requested by a message generated at $P_{i,j}$, or a channel of the last row of the diagram in Fig. 1 (that is, the message using it has no alternative path), or any horizontal channel in the middle of the same diagram. Looking at Fig.

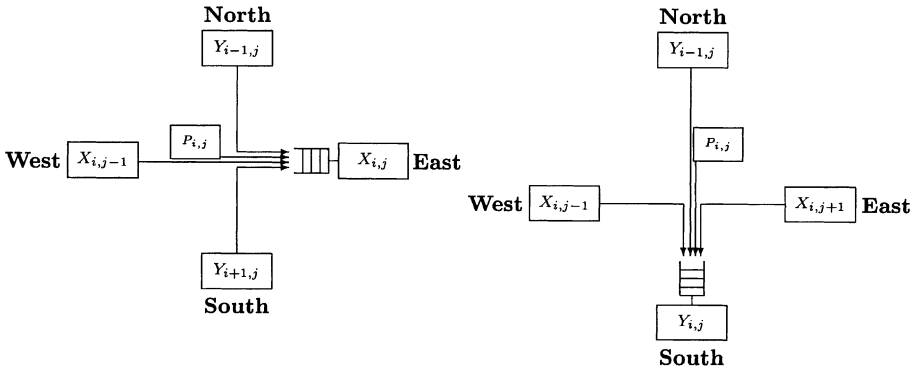

Fig. 2. Possible sources of contention when a west–east message asks for $X_{i,j}$ (left), and a north–south message asks for $Y_{i,j}$ (right).

2, we can distinguish $M = 3$ classes of flows that use $X_{i,j}$ in different ways and affect the waiting time W_{WE} with the following contributions:

1. $W_{\mathrm{WE}}^{[\mathrm{north\text{–}east}]}$ that is, the waiting time due to the flow that comes from north and, once reached $N_{i,j}$, follows the east direction. This flow consists of two streams:
   (a) The messages that have reached the row of the destination node and cannot proceed in an adaptive way. From the point of view of the average message, $X_{i,j}$ corresponds to the channels $X_{K+1,j}$ of the last row in Fig. 1. To estimate the contribution of this stream, we sum all its components except the last channel $X_{K+1,K}$. From (18) we have

$$W_{\mathrm{WE}}^{[\mathrm{north\text{–}east}]_1} = \frac{1}{2(1-\rho_{\mathrm{WE}})} \sum_{j=1}^{K} F_{K,j}^{Y} \left[(\overline{T}_{K+1,j}^{X})^2 + (\sigma_{K+1,j}^{X})^2 \right], \tag{19}$$

where $F_{K,j}^{Y}$ are in (15) and (16). To simplify the notation, we use the following abbreviation $(S_{i,j}^{X})^2 = [(\overline{T}_{i,j}^{X})^2 + (\sigma_{i,j}^{X})^2]$, and $(S_{i,j}^{Y})^2 = [(\overline{T}_{i,j}^{Y})^2 + (\sigma_{i,j}^{Y})^2]$.
   (b) The messages that use $X_{i,j}$ for an horizontal step and can still proceed in an adaptive way. From the point of view of the average message, $X_{i,j}$ represents any generic horizontal channel except those of the last row in Fig. 1 that is, any channel $X_{i,j}$ for $i, j = 1, 2, \ldots, K$. Hence, we have

$$W_{\mathrm{WE}}^{[\mathrm{north\text{–}east}]_2} = \frac{1}{2(1-\rho_{\mathrm{WE}})} \sum_{i=2}^{K} \sum_{j=1}^{K} f^{X} F_{i-1,j}^{Y} (S_{i,j}^{X})^2. \tag{20}$$

2. $W_{\mathrm{WE}}^{[\mathrm{south\text{–}east}]}$ that is, the waiting time due to the flow that comes from south and, once reached the node $N_{i,j}$, changes dimension following the east direction. This flow is symmetric to $W_{\mathrm{WE}}^{[\mathrm{north\text{–}east}]}$. The estimation of its two components $W_{\mathrm{WE}}^{[\mathrm{south\text{–}east}]_1}$ and $W_{\mathrm{WE}}^{[\mathrm{south\text{–}east}]_2}$ is quite analogous to (19) and (20). Therefore, in the evaluation of W_{WE}, $W_{\mathrm{WE}}^{[\mathrm{south\text{–}east}]}$ gives the same contribution as $W_{\mathrm{WE}}^{[\mathrm{north\text{–}east}]}$.
3. $W_{\mathrm{WE}}^{[P_{i,j}\text{–east}]}$ that is, the waiting time due to messages generated at $P_{i,j}$ and going towards east. From the point of view of the average message, the channel under consideration in Fig. 1 is $X_{1,1}$. As $X_{1,1}$ is used by any stream going towards east (in both south and north directions), we multiply by two the equations of the deterministic and adaptive streams. Specifically:
   (a) The messages that have to reach a destination which is in the same row of the sender processor (horizontal dotted line in Fig. 1) contribute as

$$W_{\mathrm{WE}}^{[P_{i,j}\text{–east}]_1} = \frac{\beta^{X}}{(1-\rho_{\mathrm{WE}})\tau} (S_{K}^{X})^2. \tag{21}$$

   (b) The messages that have to reach a destination which is in a different row and column (plain arrows in Fig. 1) contribute as

$$W_{\mathrm{WE}}^{[P_{i,j}\text{–east}]_2} = \frac{\alpha f^{X}}{(1-\rho_{\mathrm{WE}})\tau} (S_{1,1}^{X})^2. \tag{22}$$

Taking into account all previous contributions, we get

$$W_{\mathrm{WE}} = \frac{1}{1-\rho_{\mathrm{WE}}}\left[\sum_{j=1}^{K} F^{Y}_{K,j}(S^{X}_{K+1,j})^2 + \sum_{i=2}^{K}\sum_{j=1}^{K} f^{X}F^{Y}_{i-1,j}(S^{X}_{i,j})^2 + \frac{\beta^{X}}{\tau}(S^{X}_{K})^2 + \frac{\alpha f^{X}}{\tau}(S^{X}_{1,1})^2\right], \tag{23}$$

where $(S^{X}_{i,j})^2$ and $(S^{Y}_{i,j})^2$ are evaluated in Section 4.2.2, and the utilization rate is

$$\begin{aligned}\rho_{\mathrm{WE}} &= \rho^{[\mathrm{north\text{–}east}]}_{\mathrm{WE}} + \rho^{[\mathrm{south\text{–}east}]}_{\mathrm{WE}} + \rho^{[P_{i,j}\text{–east}]}_{\mathrm{WE}} \\ &= 2\left(\sum_{j=1}^{K} F^{Y}_{K,j}\overline{T}^{X}_{K+1,j} + \sum_{i=2}^{K}\sum_{j=1}^{K} f^{X}F^{Y}_{i-1,j}\overline{T}^{X}_{i,j} + \frac{\beta^{X}}{\tau}\overline{T}^{X}_{K-1} + \frac{\alpha f^{X}}{\tau}\overline{T}^{X}_{1,1}\right).\end{aligned} \tag{24}$$

The other waiting times can be estimated through the same approach used for W_{WE}. For the west–east direction in X (Fig. 2), we have

$$W_{\mathrm{NE}} = \frac{1}{1-\rho_{\mathrm{NE}}}\left[\sum_{j=1}^{K-1} F^{X}_{K+1,j}(S^{X}_{K+1,j+1})^2 + \sum_{i=1}^{K}\sum_{j=1}^{K-1} f^{X}F^{Y}_{i,j}(S^{X}_{i,j+1})^2 + \sum_{j=1}^{K-1}\frac{\beta^{X}}{\tau}(S^{X}_{j})^2 + \frac{\alpha f^{X}}{\tau}(S^{X}_{1,1})^2\right]. \tag{25}$$

For the north–south direction in the dimension Y, we have to evaluate W_{NS} and W_{WS}. The three classes of flows on $Y_{i,j}$ are shown in Fig. 2. We get

$$W_{\mathrm{NS}} = \frac{1}{1-\rho_{\mathrm{NS}}}\left[\sum_{i=1}^{K} F^{X}_{i,K}(S^{Y}_{i,K+1})^2 + \sum_{i=1}^{K}\sum_{j=1}^{K-1} f^{Y}F^{X}_{i,j}(S^{Y}_{i,j+1})^2 + \frac{\beta^{Y}}{\tau}(S^{Y}_{K})^2 + \frac{\alpha f^{Y}}{\tau}(S^{Y}_{1,1})^2\right], \tag{26}$$

$$W_{\mathrm{WS}} = \frac{1}{1-\rho_{\mathrm{WS}}}\left[\sum_{i=1}^{K-1} F^{Y}_{i,K+1}(S^{Y}_{i+1,K+1})^2 + \sum_{i=1}^{K-1}\sum_{j=1}^{K} f^{Y}F^{Y}_{i,j}(S^{X}_{i+1,j})^2 + \sum_{i=1}^{K}\frac{\beta^{Y}}{\tau}(S^{Y}_{i})^2 + \frac{\alpha f^{Y}}{\tau}(S^{Y}_{1,1})^2\right]. \tag{27}$$

4.2.2. Link utilization time

A message holds a link for a period that we call *link utilization time*. Its value is equal to the residual transmission time of the message minus the time for the already transmitted flits. Some simple considerations lead to the following equations that denote the link utilization time as a function of the link position in an average message transmission

$$\overline{T}^{X}_{j} = T^{X}_{j} - j \quad (j = 1,\ldots,K), \tag{28}$$

$$\overline{T}_i^Y = T_i^Y - i \quad (i = 1, \ldots, K), \tag{29}$$

$$\overline{T}_{i,j}^X = T_{i,j}^X - (2K - i - j + 2) \quad (i = 1, \ldots, K+1,\ \ j = 1, \ldots, K), \tag{30}$$

$$\overline{T}_{i,j}^Y = T_{i,j}^Y - (2K - i - j + 2) \quad (i = 1, \ldots, K,\ \ j = 1, \ldots, K+1). \tag{31}$$

The evaluation of the residual transmission times is presented in Section 4.3. The general distribution assumed for the link utilization time would require also the specification of the variance of the service time $\overline{T}_{i,j}$. We observe that in wormhole routing the link utilization time is equal to the mean time to transmit the entire message plus the mean waiting time to obtain the remaining links of the path. The former term has a known distribution (that chosen for the message length), whereas the latter term and the covariance between these terms are unknown. We assume null covariance and exponential distribution for the mean waiting time. These assumptions typically lead the analytical values to underestimate the real mean latency times unless for very low message generation rates, as shown by the model validation section.

4.3. Residual transmission time

The residual transmission time $T_{i,j}^X$ is the average delay the header of a message experiences while traveling from $X_{i,j}$ to the destination. Hence, T_{latency} represents the residual transmission time of a message from the source node. To evaluate residual transmission time values, we adopt a backward analysis that moves from the *data trail* node of the diagram in Fig. 1 back to the first line of horizontal and vertical channels.

The mean time T_{DT} to complete the *data trail* phase is a known parameter because it corresponds to the transmission time of all the flits of an average message. Therefore, we can write $T_{\text{DT}} = L/B$, where B denotes the number of wires per link, and L the average length of the message in flits.

For the adaptive messages, we distinguish four classes of horizontal channels: the last channel of the last row $X_{K+1,K}$; the last column $X_{i,K}$ but $X_{K+1,K}$; the last row $X_{K+1,j}$ but $X_{K+1,K}$; the other $X_{i,j}$. These classes have the following residual transmission times:

$$T_{K+1,K}^X = T_{\text{DT}} + 1, \tag{32}$$

$$T_{i,K}^X = W_{\text{WS}} + T_{i,K+1}^Y + 1 \quad (i = 1, 2, \ldots, K), \tag{33}$$

$$T_{K+1,j}^X = W_{\text{WE}} + T_{K+1,j+1}^X + 1 \quad (j = 1, 2, \ldots, K-1), \tag{34}$$

$$\begin{aligned} T_{i,j}^X = {} & (1 - p_X) T_{i,j+1}^X + p_X (1 - p_Y) T_{i,j+1}^Y \\ & + p_X p_Y \left[r(W_{\text{WE}} T_{i,j+1}^X) + (1 - r)(W_{\text{WS}} T_{i,j+1}^Y) \right] + 1 \\ & (i = 1, 2, \ldots, K;\ \ j = 1, 2, \ldots, K-1). \end{aligned} \tag{35}$$

The residual transmission times in (32)–(34) are immediately derived from Fig. 1. The additional unit term denotes the time to transmit one data flit through a link. The residual transmission time in (35) for the intermediate links is obtained by adding the time values corresponding to the following events multiplied by the probability of their occurrence: the message continues along the dimension X; the message continues along the dimension Y; the message has found both channels busy, and will continue when either one becomes free.

The equations for the residual transmission time of the vertical channels are obtained in a similar way. We distinguish four classes of vertical channels: the last channel of the last column $Y_{K,K+1}$; the last row $Y_{K,j}$ but $Y_{K,K+1}$; the last column $Y_{i,K+1}$ but $Y_{K,K+1}$; the other $Y_{i,j}$. Therefore, we get

$$T^Y_{K,K+1} = T_{\mathrm{DT}} + 1, \tag{36}$$

$$T^Y_{K,j} = W_{\mathrm{NE}} + T^X_{K+1,j} + 1 \quad (j = 1, 2, \ldots, K), \tag{37}$$

$$T^Y_{i,K+1} = W_{\mathrm{NS}} + T^Y_{i+1,K+1} + 1 \quad (i = 1, 2, \ldots, K-1), \tag{38}$$

$$\begin{aligned} T^Y_{i,j} = {} & (1 - p_X) T^X_{i+1,j} + p_X (1 - p_Y) T^Y_{i+1,j} + p_X p_Y \\ & \times \left[s(W_{\mathrm{NS}} T^X_{i+1,j}) + (1 - s)(W_{\mathrm{NE}} T^Y_{i+1,j}) \right] + 1 \\ & (i = 1, 2, \ldots, K-1; \;\; j = 1, 2, \ldots, K). \end{aligned} \tag{39}$$

The terms r and s in (35) and (39) are binary terms: if $W_{\mathrm{WS}} < W_{\mathrm{WE}}$, then $r = 0$, else $r = 1$; if $W_{\mathrm{NE}} < W_{\mathrm{NS}}$, then $s = 0$, else $s = 1$. The residual transmission time values associated with the β^X and β^Y streams are obtained analogously

$$T^X_1 = T^Y_1 = T_{\mathrm{DT}} + 1, \tag{40}$$

$$T^X_j = W_{\mathrm{WE}} + T^X_{j-1} + 1 \quad (j = 2, \ldots, K), \tag{41}$$

$$T^Y_i = W_{\mathrm{NS}} + T^Y_{i-1} + 1 \quad (i = 2, \ldots, K). \tag{42}$$

Eqs. (32)–(42) are in recursive form. They can be solved through a backward flow analysis starting from T_{DT}, and then following the south–north and west–east direction, alternating horizontal and vertical channels.

4.4. Evaluation of the latency time

4.4.1. Probability of link contention

The evaluation of the mean latency time requires an estimation of the probability of contention on the vertical and horizontal links, that is p_X and p_Y. These probabilities can be obtained as the sum of all flow rates and time values along horizontal and vertical links, respectively. In addition to the flows that travel following the west–east direction in X and the north–south direction in Y (as shown by Fig. 1), a horizontal channel is used also by the flows that follow the south–north and west–east direc-

tions. Analogously, a vertical channel is used also by the flows that follow the north–south and east–west directions. These contributions double the amount of messages on each link and motivate the multiplier terms two in the following equations:

$$p_X = 2\sum_{i=1}^{K+1}\sum_{j=1}^{K} F_{i,j}^X \overline{T}_{i,j}^X + \frac{2\beta^X}{\tau}\sum_{j=1}^{K}\overline{T}_j^X, \tag{43}$$

$$p_Y = 2\sum_{i=1}^{K}\sum_{j=1}^{K+1} F_{i,j}^Y \overline{T}_{i,j}^Y + \frac{2\beta^Y}{\tau}\sum_{i=1}^{K}\overline{T}_i^Y. \tag{44}$$

4.4.2. Mean latency time

We are now able to evaluate the mean latency time as weighted sum of the residual transmission time values experienced by α, β^X, β^Y. We get

$$T_{\text{latency}}(\tau_E) = \alpha T^\alpha + \beta^X\left(T_K^X + W_{WE} + W_{NE}\right) + \beta^Y\left(T_K^Y + W_{NS} + W_{WS}\right), \tag{45}$$

where

$$T^\alpha = (1-p_X)T_{1,1}^X + p_X(1-p_Y)T_{1,1}^Y + p_X p_Y \times \left[v(W_{WE} + W_{NE} + T_{1,1}^X) + (1-v)(W_{NS} + W_{WS} + T_{1,1}^Y)\right], \tag{46}$$

v is a binary term, that is if $(W_{WE} + W_{NE}) < (W_{NS} + W_{WS})$, then $v = 1$, else $v = 0$. The mutual dependency existing among some variables does not permit to achieve a closed formula for the mean latency time. However, a simple backward computation provides any performance value in few iterative steps.

5. Validation and performance

In this section we validate the analytical model through a discrete event simulator. The independent replications method was used to obtain confidence intervals at 95% level of confidence.

We report values related to four different network dimensions: 4×4, 8×8, 12×12 and 16×16. The average message length is 12 flits, hence we validate the model in the case of average message length greater than the average path length, average message length equal to the average path length, and average message length smaller than the average path length. The message generation is modeled as a Poisson process with τ_E time cycles per node, and the message destination is an uniformly distributed random variable.

Tables 1 and 2 show the analytical and simulated latency times for the transmission of a message. The results show that the latency evaluated through the model is a good approximation of that obtained through simulation. In particular, the error is under 6% for low and medium arrival rates and under 12% for high arrival rates that tend to saturate the network.

Table 1
Comparison of analytical and simulation results for 4×4 and 8×8 torus

Gen. rate per node	4×4			8×8		
	Simulation	Model	Error (%)	Simulation	Model	Error (%)
0.001	13.43	13.65	+1.6	15.55	15.73	+1.1
0.002	13.58	13.70	+0.9	15.96	15.92	−0.3
0.003	13.68	13.75	+0.5	16.27	16.11	−1.0
0.004	13.89	13.81	−0.6	16.81	16.31	−2.9
0.005	14.14	13.87	−1.9	17.10	16.51	−4.6
0.006	14.32	13.93	−2.7	17.66	16.72	−5.3
0.007	14.53	13.98	−3.8	18.15	16.94	−6.6
0.008	14.73	14.04	−4.7	18.65	17.18	−7.9
0.009	14.89	14.10	−5.3	19.14	17.42	−8.9
0.010	15.06	14.17	−5.9	19.52	17.69	−9.4
0.011	15.29	14.23	−6.9	20.12	17.97	−10.7
0.015	16.10	14.50	−9.9	22.18	19.39	−12.6

As an example of application, we use our model to perform some analytical comparisons between the performance of adaptive and deterministic wormhole. To this purpose, we use the model of deterministic routing presented in [7]. In the plots, the *message generation time* τ_E is denoted as mgt. Delay is measured in clock cycles. Recall that the main task of this paper is the presentation of the analytical model for wormhole routing and not a truly comparison of deterministic and adaptive performance. Therefore, the analytical comparison will be short.

In Fig. 3 we estimate the influence of the network dimension on the communication in the case of low, medium, and high traffic ($\tau_E = 300$, 500 and 1000, respectively). In particular, we plot the average blocking time per hop (that is, the waiting time to get a link) for an average message length of 12 flits. The advantages of adaptive routing become evident for medium-large traffic. Fig. 4 compares the mean latency time of deterministic and adaptive routing in a 12×12 torus with bi-directional links as a function of the message length, for three message generation

Table 2
Comparison of analytical and simulation results for 12×12 and 16×16 torus

Gen. rate per node	12×12			16×16		
	Simulation	Model	Error (%)	Simulation	Model	Error (%)
0.001	17.79	17.90	+0.6	20.07	20.05	+0.1
0.002	18.43	18.31	−0.6	20.99	20.65	−1.6
0.003	19.09	18.76	−1.6	21.85	21.37	−2.2
0.004	19.88	19.27	−3.0	22.82	22.27	−2.5
0.005	20.73	19.87	−4.1	23.99	23.47	−2.2
0.006	21.33	20.58	−3.5	25.06	25.34	+1.1
0.007	22.15	21.40	−3.4	26.27	29.28	+11.4
0.008	22.65	22.52	−0.6	–	–	–
0.009	23.25	24.20	+4.0	–	–	–

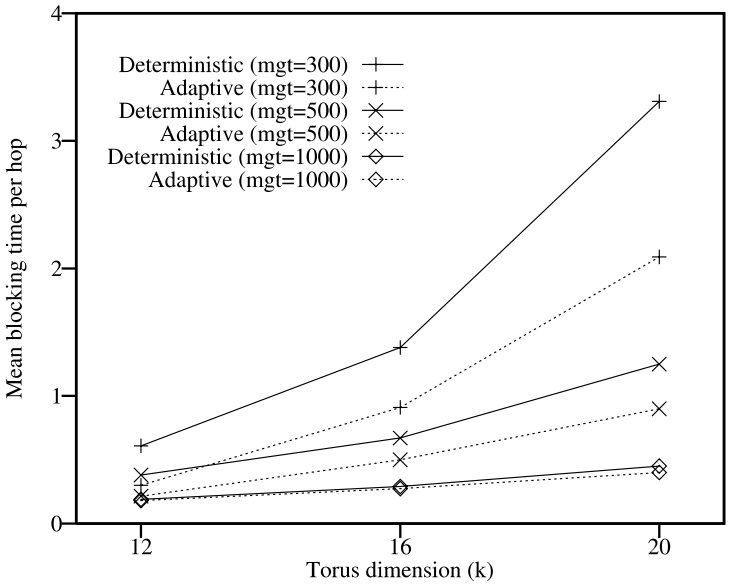

Fig. 3. Mean blocking time per hop for deterministic and adaptive routing as a function of the torus dimension $k \times k$.

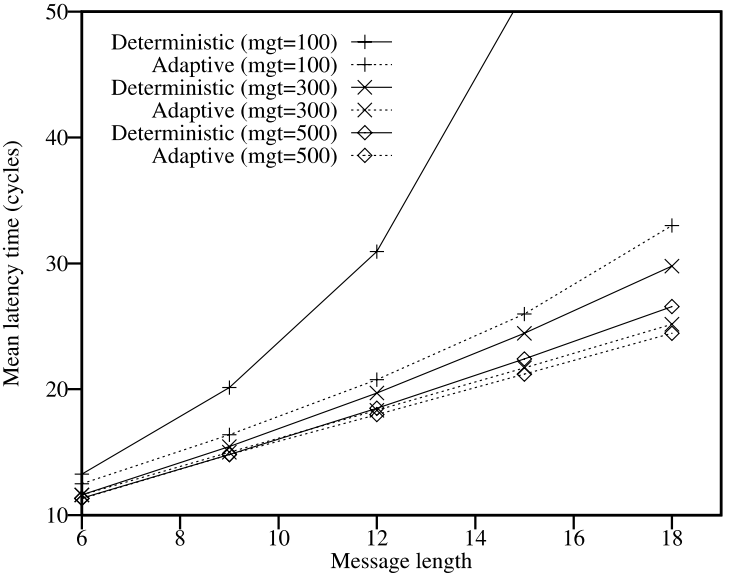

Fig. 4. Mean latency time for deterministic and adaptive routing as a function of the message length for three traffic conditions.

time values that is, $\tau_E = 300$, $\tau_E = 500$, and $\tau_E = 1000$. The adaptive technique applied to the torus topology achieves best performances for any considered message generation rate and message length. Moreover, it is important to observe that the torus topology favors the adaptive policies even for uniform traffic. Conversely, as demonstrated in [6], in a mesh without wrap-around connections the degree of adaptiveness is more limited, thereby leading to prefer deterministic to adaptive strategies in the case of uniform traffic.

6. Conclusions

In this paper we propose a modeling approach to evaluate the message latency time of minimal and fully adaptive wormhole routing in two-dimensional torus. The assumptions that render the model analytically tractable are commonly accepted in literature. Due to the complexity of the investigated routing strategy, a closed form solution for the latency time is impracticable. Our analysis achieves recursive formulas that give the average latency time in a few iterative steps. We validate our analysis through simulations that demonstrate the accuracy of the results. As a final point, we would like to remark that no theoretical limitation prevents the extension of our analysis to hypercubes and other symmetric topologies with wrap-around connections.

References

[1] V.S. Adve, M.K. Vernon, Performance analysis of mesh interconnection networks with deterministic routing, IEEE Transactions on Parallel and Distributed Systems 5 (3) (1994) 225–246.
[2] A. Agarwal, Limits on interconnection network performance, IEEE Transaction on Parallel and Distributed Systems 2 (4) (1991) 398–412.
[3] K.M. Al-Tawil, M. Abd-El-Barr, F. Ashraf, A survey and comparison of wormhole routing techniques in mesh networks, IEEE Network (March–April) (1997) 38–45.
[4] Y. Boura, C.R. Das, T.M. Jacob, A performance model for adaptive routing in hypercubes, in: Proceedings of International Workshop on Parallel Processing, 1994, pp. 11–16.
[5] M.-S. Chen, K.G. Shin, Adaptive fault tolerant routing in hypercube multicomputers, IEEE Transaction on Computers 39 (12) (1990) 1406–1416.
[6] A.A. Chien, J.H. Kim, Planar-adaptive routing: low-cost adaptive networks for multiprocessors, Journal of ACM 42 (1) (1995) 91–123.
[7] B. Ciciani, M. Colajanni, C. Paolucci, An accurate model for the performance analysis of deterministic wormhole routing, in: Proceedings of International Parallel Processing Symposium (IPPS'97), IEEE Computer Society, Sivler Spring, MD, 1997, pp. 353–359.
[8] B. Ciciani, M. Colajanni, C. Paolucci, Performance evaluation of deterministic wormhole routing in k-ary n-cubes, Parallel Computing 24 (1998) 2053–2075.
[9] M. Colajanni, B. Ciciani, S. Tucci, Performance analysis of circuit-switching interconnection networks with deterministic and adaptive routing, Performance Evaluation 34 (1998) 1–26.
[10] W.J. Dally, Performance analysis of k-ary n-cube interconnection networks, IEEE Transactions on Computers 39 (6) (1990) 775–785.
[11] W.J. Dally, H. Aoki, Deadlock-free adaptive routing in multicomputer networks using virtual channels, IEEE Transaction on Parallel and Distributed Systems 4 (4) (1993) 466–475.
[12] J.T. Draper, J. Gosh, A comprehensive analytical model for wormhole routing in multicomputer systems, Journal of Parallel and Distributed Computing 23 (2) (1994) 202–214.

[13] J. Duato, P. Lopez, Highly adaptive wormhole routing algorithms for N-dimensional torus, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 21 (1995) 87–104.
[14] J. Duato, A new theory of deadlock free adaptive routing in wormhole routing networks, IEEE Transactions on Parallel and Distributed Systems 4 (12) (1993) 1320–1331.
[15] C.J. Glass, L.M. Ni, The turn model for adaptive routing, Journal of ACM 41 (5) (1994) 874–902.
[16] 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 Transactions on Parallel and Distributed Systems 5 (12) (1994) 1233–1251.
[17] K. Hwang, Advanced Computer Architecture, McGraw Hill, New York, 1993.
[18] P. Kermani, L. Kleinrock, Virtual cut-through: a new computer communication switching technique, Computer Networks 3 (1979) 267–286.
[19] J. Kim, C.R. Das, Hypercube communication delay with wormhole routing, IEEE Transactions on Computers 43 (7) (1994) 806–814.
[20] A. Kumar, L.N. Bhuyan, Evaluating virtual channels for cache-coherent shared-memory multiprocessors, in: Proceedings of 1996 ACM International Conference on Supercomputing, 1996.
[21] A. Lagman, W.A. Najjar, S. Sur, P.K. Srimani, Evaluation of idealized adaptive routing on k-ary n-cubes, in: Proceedings of 1993 Symposium on Parallel and Distributed Processing, IEEE Computer Society, Silver Spring, MD, 1993, pp. 166–169.
[22] D. Linder, J.C. Harden, An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes, IEEE Transactions on Computers 40 (1) (1991) 2–12.
[23] S. Loucif, M. Ould-Khaoua, L.M. Mackenzie, Analysis of fully adaptive wormhole-routing in tori, Parallel Computing 25 (12) (1999) 1477–1487.
[24] W.A. Najjar, A. Lagman, S. Sur, P.K. Srimani, Modeling adaptive routing in k-ary n-cube networks, in: Proceedings of MASCOTS '94, 1994, pp. 120–125.
[25] L.M. Ni, P.K. McKinley, A survey of wormhole routing techniques in direct networks, IEEE Computer 26 (2) (1993) 62–76.
[26] M. Ould-Khaoua, An analytical model of Duato's fully-adaptive routing algorithm in k-ary n-cubes, in: Proceedings of 27th International Conference on Parallel Processing, 1998.
[27] M. Ould-Khaoua, A performance model for duato's adaptive routing algorithm in k-ary n-cubes, IEEE Transactions on Computers 48 (12) (1999) 1–8.
[28] H.J. Siegel, C.B. Stunkel, Trends in parallel machine interconnection networks, IEEE Computational Science and Engineering (Summer) (1996).
[29] H. Takagi, Queueing Analysis – A Foundation of Performance Evaluation, Elsevier, North-Holland, 1991.