An Accurate Combinatorial Model for Performance Prediction of Deterministic Wormhole Routing in Torus Multicomputer Systems

H. H. Najaf-abadi[1], H. Sarbazi-azad[2,1]

[1] School of Computer Science, IPM, Tehran, Iran.
[2] Computer Engineering Dept., Sharif Univ. of Technology, Tehran, Iran.
{h_hashemi, azad}@ipm.ir, azad@sharif.edu

Abstract

Although several analytical models have been proposed in the literature for different interconnection networks with deterministic routing, very few of them have considered the effects of virtual channel multiplexing on network performance. This paper proposes a new analytical model to compute message latency in a general n-dimensional torus network with an arbitrary number of virtual channels per physical channel. Unlike the previous models proposed for toroidal-based networks, this model uses a combinatorial approach to consider all different possible cases for the source-destination pairs, thus resulting in an accurate prediction. The results obtained from simulation experiments confirm that the proposed model exhibits a high degree of accuracy for various network sizes, under different operating conditions, compared to a similar model proposed very recently [16], which considers virtual channel utilization in the k-ary n-cube network.

1. Introduction

Topology, routing algorithm and switching method are the most important factors determining the performance of an interconnection network. Practical multicomputers have widely employed torus networks for low latency high-bandwidth inter-processor communication [9].

Owning to its low buffer size, *wormhole switching* has been widely employed in multicomputers. Another advantage of wormhole routing is that, in the absence of blocking, message latency is almost independent of the distance between source and destination. In this switching technique, messages are broken into *flits*, each of a few bytes, for transmission and flow control. The *header* flit, containing routing information, is used to govern routing and the remaining data flits follow in a pipelined fashion. If the header is blocked, the other flits are blocked in situ. The advantage of this technique is that it reduces the impact of message distance on the latency under light traffic. Yet, as network traffic increases, messages may experience large delays to cross the network due to the chain of blocked channels [15]. To overcome this, the flit buffers associated with a given physical channel are organised into several virtual channels [7], each representing a “logical” channel with its own buffer and flow control logic. Virtual channels are allocated independently to different messages and compete with each other for the physical bandwidth. This decoupling allows messages to bypass each other in the event of blocking, using network bandwidth that would otherwise be wasted.

Routing algorithms establish the path between the source and destination of a massage. Routing can be deterministic or adaptive. With adaptive routing, the path taken by a message is affected by the traffic on network channels. In deterministic routing, messages with the same source and destination always traverse the same path. This form of routing results in a simpler router implementation [10] and has been used in many practical multicomputers.

Simulation is an approach to evaluate the performance of an interconnection network for a specific configuration. But, depending on the complexity of the interconnection network and resources available, this technique may be too time-consuming to perform. Another approach is utilization of an analytical model of the system. An appropriate analytical model can predict the performance of a specific interconnection network structure in a fraction of the time simulation would take. Thus, it is justified to be in pursuit of accurate analytical models for the performance of different network topologies.

Analytical models of networks base on wormhole switching and deterministic routing have been reported in the past [1-4, 6, 8, 11, 12, 14, 17]. There have however been few models reported in the literature that have considered the performance of such networks with any number of virtual channels per physical channel. Of these models, only [16] captures the effect of virtual channel multiplexing on dimension-order routing for any number of virtual channels per physical channel. The model proposed by Draper and Ghosh [8] considers only the use of a minimum requirement of virtual channels (2 virtual channels) to ensure deadlock freedom according to the methodology proposed in [5], and cannot deal with any arbitrary number of virtual channels. When the number of virtual channels is large (> 2), however, the effect of virtual channels on network performance cannot be ignored since this can cause the analytical model to produce inaccurate predictions of message latency, especially when the network operates under heavy traffic loads. This is because the multiplexing of virtual channels increases the latency seen by an individual message inside the network as virtual channels share the bandwidth of the physical channel in a multiplexed manner. The model, proposed very recently in [16], uses a different approach and has the main advantage of being simpler to derive than the existing models including Draper & Ghosh's model [8]. Moreover, the model can support both unidirectional and bidirectional k-ary n-cubes with any number of virtual channels. However, the accuracy of the model is its main drawback especially near the high traffic region.

In this paper, a new *combinatorial* performance model is proposed for dimension-order routing in which all the potential source-destination node pairs of messages are considred. Thus, the proposed model, while keeping all the advantages of the model proposed in [16], is highly improved in the accuracy of saturation point prediction.

2. The analytical model

In what follows, we first outline the assumptions made in the analysis. The model is discussed in the context of the unidirectional torus for the sake of presentation. Only a few simple modifications are required to adapt it for the bidirectional case.

2.1. Assumptions

The model is based on the following assumptions, which are widely used in the literature [1-4, 6-8, 11-14, 16].

a) The network is an n-D torus with radix k_1 for dimension 1, k_2 for dimension 2, and so on.
b) Nodes generate traffic independently of each other, and follow a Poisson process, with a mean rate of λ_g messages/cycle. Furthermore, message destinations are uniformly distributed across the network nodes.
c) Message length is fixed (M flits). Each flit is transmitted in t_c cycles from one router to the next.
d) Messages are transferred to the local PE through the ejection channel once they arrive at their destination.
e) L virtual channels, ($L \geq 2$), per physical channel are used.

For deadlock free routing, a restricted virtual channel allocation scheme, based on Duato's methodology [9] in the context of deterministic routing, is enforced. In this scheme the virtual channels of a given physical channel are split into two sets: $VC_1 = \{v_3, v_4, \dots, v_L\}$ and $VC_2 = \{v_1, v_2\}$. A message at node address $C=c_1c_2 \dots c_n$ and destined to node $D = d_1d_2 \dots d_n$, can choose any of the L-2 virtual channels in VC_1 of dimension i, the next dimension to be traversed. If all these virtual channels are busy, the message crosses v_1 when $c_i<d_i$; otherwise it crosses v_2 [5]. Adopting the same terminology as in [9], the virtual channels v_1 and v_2 represent "escape channels". Since this algorithm is a restricted form of Duato's methodology, it is deadlock free.

2.2. Model description

A generated message in an n-dimensional torus traverses m hops (where $1 \le m \le \sum_{i=0}^{n-1}(k_i - 1)$) to reach its destination. The destination node of an m-hop message can be any of $C_m^{n,(k_{n-1}-1,k_{n-2}-1,\dots,k_0-1)}$ nodes that are m hops away from the source node, where $C_m^{n,(k_{n-1},k_{n-2},\dots,k_0)}$ is the number of different m-combinations of n distinguishable types of items with each type i consisting of K_i identical items. This expression can be recursively calculated as:

$$C_m^{d,(k_{d-1},k_{d-2},\dots,k_0)} = \begin{cases} 1, & \text{if } m = 0 \\ 0, & \text{if } m < 0 \text{ or } d < 0 \\ \sum_{i=0}^{k_{d-1}} C_{m-i}^{d-1,(k_{d-2},k_{d-3},\dots,k_0)}, & \text{otherwise} \end{cases}$$

The average number of dimensions that a message traverses before reaching its destination is equal to:

$$h = \sum_{m=1}^{(k_0-1)+(k_1-1)\dots+(k_{n-1}-1)} mP_m$$

in which P_m is the probability of the length of the path of a message being equal to m hops. In a unidirectional torus with a uniform traffic pattern we have:

$$P_m = C_m^{n,(k_{n-1}-1,k_{n-2}-1,\dots,k_0-1)} \Big/ (k_0k_1 \dots k_{n-1} - 1)$$

in which the numerator expression corresponds to the number of nodes at a distance of m from the source node and the denominator is the number of nodes of the network other than the source node, i.e. potential destinations.

Considering that messages travel an average of h hops before reaching their destination, the average rate at which messages enter nodes of the network is equal to h times the message generation rate. On the other hand, since with unidirectional routing, the n output channels of each node are equally utilized, the arrival rate of messages to any network channel, denoted λ_c, is equal to $h\lambda_g/n$.

Average message latency is defined as the average amount of time for a message to reach its destination node and all its flits to be ejected out of the network. We consider the average message latency to be a measure representative of the performance of the parallel processing system. Computation of this parameter calls for the calculation of three main factors. One is the *effective network latency*. Another is the average degree of virtual channel multiplexing and the last factor is the average waiting time at the source node. The *average network latency* is defined as the sum of the average message transfer time and average blocking delay at different dimensions (excluding the effect of multiplexing). The effective network delay is then computed by inflating the average network latency by the average virtual channel multiplexing degree. The naming of factors in the following description complies closely to that of [13].

When a message needs to traverse one of the hops of dimension d_i, $0 \le d_i \le n-1$, it is delayed an average amount of time, W_{d_i}, before acquiring a virtual channel. A message is actually blocked only when all the virtual channels of its current hop are busy. The probability of l virtual channels of a hop in dimension d_i being busy is denoted by $P_{d_i,l}$. Considering the scheme used for virtual channel allocation, the probability of a message being blocked at a hop of dimension d_i (called the "blocking probability") is given by [16]: $P_{B_i} = P_{d_i,L} + P_{d_i,L-1}/L$

in which two cases are considered. The first expression, $P_{d_i,L}$, corresponds to the case where all the virtual channels are busy and the second, to where all but v_1 or v_2 (the one not corresponding to the direction of the message), are busy. If a message is blocked at a hop, the message is delayed by as much time as it takes all the flits of a blocking message to finish traversing that hop. If none of the messages occupying the virtual channels terminate after traversing that hop, the blocked message will additionally be delayed by the average waiting time encountered by a blocking message in the rest of its path to its destination.

Let P_{t,d_i} be the probability that a message will terminate after traversing a hop in dimension d_i. Since messages traverse an average of $k_i/2$ hops in dimension d_i, the probability of a message terminating after traversing a specific hop of dimension d_i is equal to $P_{t,d_i}/(k_i/2)$. Therefore, the probability of a message being blocked at a hop of dimension d_i and none of the blocking messages terminating after traversing that hop, can be denoted as:

$$P_{d_i} = \frac{P_{t,d_i}}{k_i/2}(1-\frac{P_{t,d_i}}{k_i/2})^{L-1}\frac{P_{d_i,L}}{L} + (1-\frac{P_{t,d_i}}{k_i/2})^{L} P_{d_i,L} + (1-\frac{P_{t,d_i}}{k_i/2})^{L-1}\frac{P_{d_i,L-1}}{L}$$

in which three cases are considered, each corresponding to one of the products. Enumerated from left to right, the first product corresponds to the case when L virtual channels are busy, but the message allocating one of them (v_1 or v_2, such that it can not be traversed by the blocked message) does terminate in the following node. The second and third cases correspond respectively to when L and L-1 virtual channels are busy and none of the messages allocating these virtual channels terminate in the following node.

$\widehat{W}_{d_i}$, the average waiting time of a blocked message to acquire a virtual channel at a hop of dimension d_i, when it considered that no other message is blocked at that hop, can be obtained as the product of the aggregate of the average waiting time of blocking messages in each of the dimensions of the remainder of their paths, and the conditional probability that none of the blocking messages terminate after traversing the channel, given that the channel is already known to be blocked (resulting in P_{d_i}/P_{B_i}), plus the length of a message times the channel cycle time, t_c (to account for the time it takes for the flits of a message to be transmitted over a single channel). This is expressed in the following equation:

$$\widehat{W}_{d_i} = \frac{P_{d_i}}{P_{B_i}} \sum_{d_j=d_i}^{n-1} \sum_{s=1}^{k_{d_j}-1} W_{d_j} P_{d_i,s|d_j} + Mt_c$$

in which $P_{d_i,s|d_j}$ is the probability of a message traversing s hops of dimension d_j given that it has already traversed a hop in dimension d_i. W_{d_i} is the average blocking time at a hop of dimension d_i when it is considered that other messages may be blocked at the same hop. If freed virtual channels are granted to waiting messages on a first-come-first-serve basis (which is usually the case), W_{d_i} can be calculated as $W_{d_i} = \hat{W}_{d_i} \widetilde{N}_{d_i,waiting}$, in which, $\widetilde{N}_{d_i,waiting}$ is the average number of waiting messages at a hop of dimension d_i. Therefore, the meaning of this is that, the actual average blocking time at a channel of dimension d_i is equal to the average waiting time of a blocked message to acquire a virtual channel at that hop when considering that no other message is blocked at that hop, times the average number of blocked messages at the channel.

The *network latency* of a message is defined as the time it takes all the flits of a message to cross the network, reach the destination node, and be ejected through the ejection channel. The average network latency of messages that traverse their first hop in dimension d_i, excluding the blocking delay of the first hop, denoted D_{d_i}, is defined as the sum of the average blocking delay that messages face at the other hops of d_i and dimensions higher than d_i (calculated as a weighted average of the waiting time in higher dimensions), the transfer time of all the flits of a message over a channel (Mt_c) and the waiting time at the ejection channel ($W_{ejection}$). Expressed mathematically:

$$D_{d_i} = \sum_{d_j=d_i}^{n-1} \sum_{s=1}^{k_{d_j}-1} W_{d_j} P_{d_j,s|d_{i_{first}}} + Mt_c + W_{ejection}$$

where $P_{d_j,s|d_{i_{first}}}$ is the probability of a message traversing at least s hops in dimension d_j given that it has traversed its first hop in dimension d_i.

The probability of a message reaching its destination after using one of the hops of dimension d_i, on the presumption that it does take a hop in that dimension, is expressed as:

$$P_{t,d_i} = \frac{\displaystyle\sum_{m=0}^{k'_{d_i}-1} P_{m+1} \frac{C_m^{d_i+1,\left(k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}{C_{m+1}^{n,\left(k_{n-1}-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}}{\displaystyle\sum_{m=0}^{k'_{n-1}-1} P_{m+1} \frac{C_m^{n-1,\left(k_{d_{n-1}}-1,\ldots,k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}{C_{m+1}^{n-1,\left(k_{n-1}-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}}$$

in which $k'_{d_i} = \sum_{i=0}^{d_i}\left(k_i-1\right)$

The numerator is the probability of a message traversing a hop in dimension d_i, but not traversing any hops in dimensions higher than d_i, and the denominator corresponds to the probability of a message traversing a hop of dimension d_i (in the first place).

When a message is to traverse a hop in dimension d_i, it may have already traversed m hops (where $0 \le m \le k'_{d_i} - 1$). The number of different paths that such a message may have taken to reach this specific hop is therefore equal to $C_m^{d_i+1,\left(k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}$, i.e. the number of different m-combinations of d_i +1 distinguishable types of items with each type j, for $0 \le j \le d_i\text{-}1$, consisting of $k_j\text{-}1$ identical items and type d_i consisting of k_{d_i}-1-1 items. But the total number of paths that an $m+1$ hop message may take is equal to $C_{m+1}^{n,\left(k_{n-1}-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}$, i.e. the number of different m-combinations of d_i +1 distinguishable types of items with each type j consisting of $k_j\text{-}1$ identical items. The probability that the last hop to be traversed by an $m+1$ hop message is in dimension d_i, is equal to the first combination divided by the second. Therefore, the probability of a message traversing its last hop in dimension d_i can be determined as a weighted average,

$$\sum_{m=0}^{k'_{d_i}-1} P_{m+1} \frac{C_m^{d_i,\left(k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}{C_{m+1}^{n-1,\left(k_{n-1}-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}.$$

Following the same approach, a message that is to traverse a hop in dimension d_i may use any of the $C_m^{n-1,\left(k_{d_n}-1,\ldots,k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}$ different paths to reach this hop. Hence, the probability that a message traverses a hop in dimension d_i can also be determined as a weighted average such as

$$\sum_{m=0}^{k'_{n-1}-1} P_{m+1} \frac{C_m^{n-1,\left(k_{d_n}-1,\ldots,k_{d_i}-1-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}{C_{m+1}^{n-1,\left(k_{n-1}-1,\ldots k_{d_1}-1,k_{d_0}-1\right)}}.$$

The later weighted average divided by the former will obviously result in P_{t,d_i}. In as similar manner, the probability of a message traversing at least s hops of dimension d_i, with the presumption that it has traversed at least one hop in dimension d_i (where $d_j > d_i$), is given by:

$$P_{d_j,s|d_i} = \frac{\displaystyle\sum_{m=0}^{k'_{n-1}-s-1} P_{m+1+s} \frac{C_m^{n-1,\left(k_{n-1}-1,\ldots,k_{d_j}-1-s,\ldots,k_{d_i}-1-1,\ldots,k_0-1\right)}}{C_{m+1+s}^{n-1,\left(k_{n-1}-1,\ldots,k_1-1,k_0-1\right)}}}{\displaystyle\sum_{m=0}^{k'_{n-1}-1} P_{m+1} \frac{C_m^{n,\left(k_{n-1}-1,\ldots,k_{d_i}-1-1,\ldots,k_0-1\right)}}{C_{m+1}^{n-1,\left(k_{n-1}-1,\ldots,k_0-1\right)}}}$$

In this equation, the numerator is the probability of a message traversing at least one hop in dimension d_i and at least s hops in dimension d_j, and the denominator is the probability of a message traversing at least one hop in dimension d_i (as the presumption).

In as similar manner, the probability of a message traversing at least s hops in dimension d_j, with the presumption that it has traversed its first hop in dimension d_i (where $d_j > d_i$) is given by

$$P_{d_j,s|d_{i,first}} = \frac{\displaystyle\sum_{m=0}^{k'_{n-1}-k'_{d_i-1}-s-1} P_{m+1+s} \frac{C_m^{n-d_i-1,\left(k_{n-1}-1,\ldots,k_{d_j}-1-s,\ldots,k_{d_i}-1-1\right)}}{C_{m+1+s}^{n-1,\left(k_{n-1}-1,\ldots k_1-1,k_0-1\right)}}}{\displaystyle\sum_{m=0}^{k'_{n-1}-k'_{d_i-1}-1} P_{m+1} \frac{C_m^{n-d_i-1,\left(k_{n-1}-1,\ldots k_{d_i-1}-1,k_{d_i}-1-1\right)}}{C_{m+1}^{n-1,\left(k_{n-1}-1,\ldots k_1-1,k_0-1\right)}}}$$

In this equation, the numerator is the probability of a message taking its first hop in dimension d_i and traversing at least a number of S hops in dimension d_j. The denominator is the probability of a message taking its first hop in dimension d_i (the presumption)

The *service time* of a channel is defined as the time it takes for all the flits of a message to be transmitted over the channel. The value of $P_{d_i,j}$ can be determined following the same approach taken in [16], using a Markovian model of the allocation of virtual channels. The Markovian model results in the following steady state probability (derivation explained in [16]), in which the service time of a channel has been approximated as the network latency of that channel:

$$P_{d_i,j} = \begin{cases} (1-\lambda_c D_{d_i})(\lambda_c D_{d_i})^l, & \text{if } 0 \le j < L \\ (\lambda_c D_{d_i})^l, & \text{if } j = L \end{cases}$$

The average number of waiting messages at a hop in dimension d_i can also be determined in a similar manner. First the probability of there being j messages waiting for a hop in d_i is calculated (according to the Markovian model mentioned above) as

$$P_{d_i,j,waiting} = \begin{cases} (1-\lambda_c D_{d_i})(\lambda_c D_{d_i})^l, & \text{if } 0 \le j < (2d_i-1)L \\ (\lambda_c D_{d_i})^l, & \text{if } j = (2d_i-1)L \end{cases}$$

where ($2d_i$ -1)L is the maximum number of messages that may be waiting to acquire a hop in dimension d_i. Verification of this fact is straightforward when it is established that only messages entering a node from dimension d_i or lower dimensions, may need to acquire a virtual channel of a hop in that dimension.

$\widetilde{N}_{d_i,waiting}$ can now be calculated as the weighted average of the number of waiting messages (the average number of messages that need to traverse a hop in d_i) as:

$$\widetilde{N}_{d_i,waiting} = \sum_{j=L}^{(2d_i-1)L} j.P_{d_i,j,waiting}$$

In the steady state, the rate of messages that exit the network through ejection channels is equal to the injection rate of messages, which is equal to the generation rate λ_g. Utilization of the ejection channel (in each node) is therefore equal to $M\lambda_g$. Given that messages are of fixed length, there is no variance in service time. Using an M/G/1 queueing model (as explained in [16]), we can calculate the waiting time at an ejection channel as:

$$W_{ejection} = M^2 \lambda_g / 2(1 - M\lambda_g)$$

The probability that a message uses a hop in dimension d_i as its first rout can be calculated as a weighted average:

$$\sum_{m=1}^{k'_{n-1}-k'_{d_i-1}} P_m \frac{C_{m-1}^{n-d_i-1,\left(k_{n-1}-1,\ldots,k_{d_i}-1-1\right)}}{C_m^{n-1,\left(k_{n-1}-1,k_{n-2}-1,\ldots,k_0-1\right)}}$$

Since this probability is dependent on d_i, a weighted average is more appropriate to determine the average total network latency of messages, denoted $\bar{S}$. Thus, considering that the total network latency of messages that take their first hop in dimension d_i is equal to $(D_{d_i} + W_{d_i})$:

$$\bar{S} = \sum_{d_i=0}^{n-1} \sum_{m=1}^{k'_{n-1}-k'_{d_i-1}} P_m.(D_{d_i} + W_{d_i}) \frac{C_{m-1}^{n-d_i-1,\left(k_{n-1}-1,k_{n-2}-1,\ldots,k_{d_i}-1-1\right)}}{C_m^{n-1,\left(k_{n-1}-1,k_{n-2}-1,\ldots,k_0-1\right)}}$$

In virtual channel flow control, multiple virtual channels share the bandwidth of a physical channel. Hence, the average service time of a message should be inflated by the amount of multiplexing that takes place across the different dimensions in order to obtain the effective average service time. The average degree of virtual channel multiplexing at dimension d_i is given by [16]:

$$\bar{l}_{d_i} = \sum_{l=0}^{L} l^2 P_{d_i,l} \Big/ \sum_{l=0}^{L} l P_{d_i,l}$$

Therefore, the average degree of multiplexing in the network becomes

$$\bar{l} = \frac{1}{k_0 + k_1 + \ldots + k_{n-1}} \sum_{d_i=0}^{n-1} k_{d_i} \bar{l}_{d_i}$$

Hence, as explained before, the effective average network delay is equal to $\bar{S}\bar{l}$.

A message originating from a given source node sees a network latency of $\bar{S}$. Modeling the local queue in the source node as an M/G/1 queue, with the mean arrival rate of λ_g/L and a service time of $\bar{S}$ with an approximated variance of $(\bar{S}-M)^2$, yields the mean waiting time seen by a message at the source node as

$$\overline{W}_s = \frac{(\lambda_g / L)\bar{S}^2 (1 + (\bar{S} - M)^2 / \bar{S}^2)}{2(1 - (\lambda_g / L)\bar{S})}$$

Finally, the average message latency of the network, $\bar{T}$, is obtained as the summation of the effective average network delay $(\bar{S}\bar{l})$, the average waiting time at the source node $(\overline{W}_s)$, and the average time for the last flit of a message to reach its destination $(h\bar{l})$, i.e. $\bar{T} = \bar{S}\bar{l} + \overline{W}_s + h\bar{l}$.

The complexity of the model reduces considerably for k-ary n-cube networks (a torus with $k_0=k_1=...=k_2$), and for the hypercube (a k-ary n-cube with $k=2$) the model reduces even more and becomes similar to the model presented in [13]. Derivation of the model for these two cases has been presented in [18].

3. Validation of the model

The analytical model has been validated through a discrete-event simulator that mimics the behaviour of the described routing algorithms in the network at the flit level. In each simulation experiment, a minimum of 120000 messages are delivered. Statistics gathering was inhibited for the first 10000 messages to avoid distortions due to the initial startup conditions. The simulator uses the same assumptions as the analysis, and some of these assumptions are detailed here with a view to making the network operation clearer. The network cycle time is defined as the transmitsion time of a single flit from one router to the next. Messages are generated at each node according to a Poisson process with a mean inter-arrival rate of λ_g messages/cycle. Message length is fixed at M flits. Destination nodes are determined using a uniform random number generator. The mean message latency is defined as the mean amount of time from the generation of a message until the last data flit reaches the local PE at the destination node. The other measures include the mean network latency, the time taken to cross the network, and the mean queuing time at the source node, the time spent at the local queue before entering the first network channel.

Numerous validation experiments have been performed for several combinations of network sizes, message lengths, and number of virtual channels to validate the model. However, for the sake of specific illustration, Fig. 1 depicts latency results predicted by the model plotted against those provided by the simulator for a 16x16 torus ($N=256$ nodes) and 8x8x8 torus (N=512), for different message lengths, M=32, 64 and 100 flits. Moreover, the number of virtual channels per physical channel was set to L=3, and 5. The horizontal axis in the figures represents the traffic rate at which a node injects messages into the network in a cycle. The vertical axis shows the mean message latency in crossing from source to destination, including waiting time at source and destination. The figure reveals that the model predicts the mean message latency with a good degree of accuracy when the network operates in the steady-state regions.

Figure 2 shows the accuracy of the proposed model (compared to the one given in [16]), for a 16x16 torus, L=5 and M=32. It is observed that the saturation point of the model presented here is closer to the saturation point of the simulation results. For the sake of brevity, only one scenario has been shown. However, in all considered cases, the model proposed here has been observed to predict the average message latency in the network with a higher degree of accuracy compared to the one proposed in [16].

4. Conclusions

Analytical models of torus-based networks with deterministic routing have widely been reported in the literature.

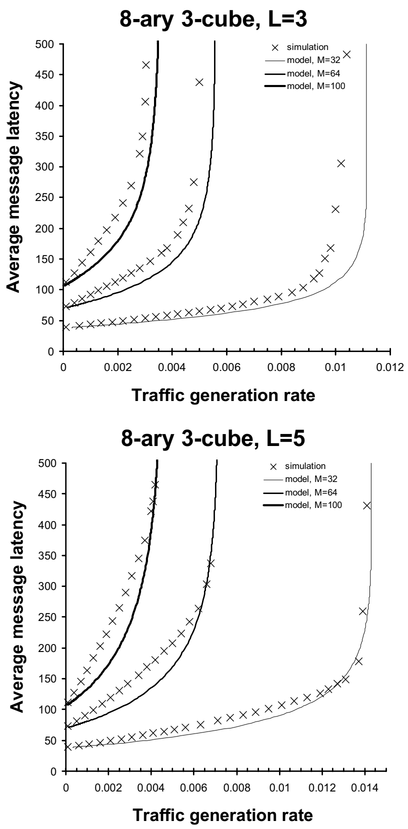

Figure 1. Average message latency predicted by the model against simulation results for *L*=3 and 5 virtual channels, and *M*=32, 64 and 100 flits in two different networks.

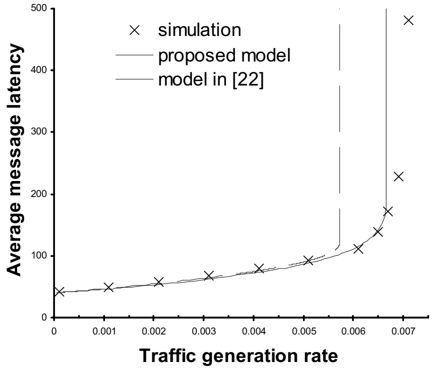

Figure 2. Comparing of prediction accuracy between the proposed model here and the one in [16] for a 16X16 torus with L=5 virtual channels and M=32 flits.

However, most of these models have not considered the effect of arbitrary numbers of virtual channels on network performance and those that have, are of frail accuracy near the saturation point. This paper has proposed a new combinatorial model that captures the effect of virtual channel multiplexing on message latency in *n*-D torus interconnection networks. The model is based on assumptions widely used in similar studies. Simulation experiments have revealed that the model predicts message latency with a reasonably high degree of accuracy compared to the most-recently proposed model in [16] having similar capabilities.

5. References

[1] V.S. Adve and M.K. Vernon, "Performance Analysis of Mesh Interconnection Network with Deterministic Routing," *IEEE Trans. Parallel and Distributed Systems*, 5 (3), 1994, 225-246.

[2] A. Agrawal, "Limits on interconnection network performance," *IEEE Trans. Parallel and Distr. Systems*, 2 (1991), 398-412.

[3] J.R. Anderson and S. Abraham, "Performance-based constraints for Multi-dimensional networks," *IEEE Trans. Parallel and Distributed Systems*, 11 (1), 2000, 21-35.

[4] B. Ciciani, M. Colajanni, and C. Paolucci, "Performance evaluation of deterministic wormhole routing in k-ary n-cubes," *Parallel Computing*, 24, 1998, 2053-2075.

[5] W.J. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Computers*, 36 (5), 1987, 547-553.

[6] W.J. Dally, "Performance analysis of k-ary n-cubes intercomnection networks," *IEEE Trans. Comp.*, 39 (6) (1990), 775-785.

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

[8] J.T. Draper and J. Ghosh, "A comprehensive analytical model for wormhole routing in multicomputer systems," *Journal Parallel and Distributed Computing*, 23, 1994, 202-214.

[9] J. Duato, S. Yalamanchili, and L. Ni, Interconnection networks: An engineering approach, IEEE Computer Society Press, Los Alamitos, CA, 2002.

[10] J. Duato, "Why commercial multicomputers do not use adaptive routing," *IEEE Technical Committee on Computer Architecture Newsletter*, 1994, 20-22.

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

[12] W.J. Guan, W.K. Tsai, and D. Blough, "An analytical model for wormhole routing in multicomputer interconnection networks", *Proc. Int. Conference on Parallel Processing*, 1993, pp. 650-654.

[13] Y. Boura, C.R. Das, "Modelling virtual channel flow control in *n*-dimensional hypercubes," *Proc. International Symposium on High Performance Computer Architecture*, 1995, 166-175.

[14] J. Kim and C.R. Das, "Hypercube communication delay with wormhole routing," *IEEE Trans. Comp.*, 43 (7), 1994, 806-814.

[15] L.M. Ni and K. McKinley, "A survey of wormhole routing techniques in direct networks," *IEEE Comp.*, 26, 1993, 62-76.

[16] H. Sarbazi-Azad, A. Khonsari, M. Ould-Khaoua, "Analysis of k-ary n-cubes with dimension-ordered routing," *Future Generation Computer Systems*, 19 (4), 2003, 493-502.

[17] H. Sarbazi-azad, "A Mathematical model of deterministic wormhole routing in hypercube multicomputers using virtual channels", *Journal of applied mathematical modeling,* 27, 2003, pp. 943-953.

[18] H. H. Najaf-abadi, H. Sarbazi-azad, "A performance model of deterministic wormhole routing in torus networks", *Technical report, IPM school of Computer science*, Tehran, 2003.