Analytic Modeling of Channel Traffic in *n*-Cubes

Hamid Sarbazi-Azad, Hamid Mahini, Ahmad Patooghy

IPM School of Computer Science, and Sharif University of Technology,
Tehran, Iran.
azad@{ipm.ir/sharif.edu}

Abstract. Many studies have shown that the imbalance of network channel traffic is of critical effect on the overall performance of multicomputer systems. In this paper, we analytically model the traffic rate crossing the network channels of a hypercube network under different working conditions. The effect of different parameters on the shaping of non-uniformity and traffic imbalance over network channels, are considered and analytical models for each case are proposed.

1 Introduction

The *routing algorithm* indicates the next channel to be used at each intermediate node. That channel may be selected from a set of possible choices and according to the size of this set, the routing algorithm may be divided into three categories: *deterministic, partially adaptive, and fully adaptive* [6, 16].

Deterministic routing assigns a single path to each source and destination node (the size of the mentioned set is one in this category) resulting in a simple router implementation. However, under deterministic routing, a message cannot use alternative paths to avoid congested channels along its route and therefore the low network performance is inevitable. The XY and e-cube routing algorithms are the most known routing algorithm of this category in meshes and hypercubes [5].

Fully adaptive routing has been suggested to overcome this limitation by enabling messages to explore all available paths (the above mentioned set has its maximum possible size) and consequently offers the potential for making better use of network resources. But these algorithms imply more router complexity for deadlock-freedom. An example of a fully adaptive routing algorithm is Duato's [7] routing algorithm. Hop-based routing [3] and Linder-Harden's [13] algorithm are also adaptive routing algorithms proposed for the mesh and hypercube networks.

Partially adaptive routing algorithms try to combine the advantages of the two other categories to produce a routing with limited adaptivity and establish a balance between performance and router complexity. They allow selecting a path from a subset of all possible paths. In fact, these algorithms limit the size of the set of possible choices. Turn model based algorithms and planar adaptive routing algorithm are the most important partially adaptive algorithms for the mesh and hypercube networks [8].

The performance of the network is mainly determined by the three characteristics of interconnection networks mentioned above (topology, switching method, and routing algorithm). However, traffic distribution of the workload is another important factor in determining the overall performance of the system. Although this fac-

tor is not considered a network characteristic and is determined by the applications being executed on the machine, it has great impact on performance. The three abovementioned factors are also influential on the traffic pattern [16].

Numerous studies have shown that with load balanced channel traffic greater network performance can be expected. This infers that overall performance is also affected by traffic pattern [14]. In this paper, we analytically model the effect of some important factors that cause imbalance in the channel traffic rates. This study focuses on the hypercube network for the sake of presentation and derives some analytic models to predict traffic in the network. In particular we model the effect of destination address distribution and routing algorithm on network channel traffic.

2 Preliminaries

This section describes the network and node structure in a hypercube along with e-cube [6] deterministic routing algorithm with virtual channels.

A n-dimensional hypercube can be modeled as a graph H_n (V, E), with the node set V (H_n) and edge set E (H_n), where $|V|=2^n$ and $|E|=n2^n$ nodes. The 2^n nodes are distinctly addressed by n-bit binary numbers, with values from 0 to 2^n -1. Each node has link at n dimensions, ranging from 1 (lowest dimension) to n (highest dimension), connecting to n neighbours. An edge connecting nodes $X= x_n x_{n-1} \ldots x_1$ and $Y= y_n y_{n-1} \ldots y_1$ is said to be at dimension j or to the jth dimensional edge if their binary addresses $x_n x_{n-1} \ldots x_1$ and $y_n y_{n-1} \ldots y_1$ differ at bit position j only, i.e. $x_j \neq y_j$. An edge in H_n can also be represented by an n-character string with one hyphen (-) and n-1 binary symbols {0, 1}. For example in a H_4, the string 00-1 denotes the edge connecting nodes 0001 and 0011.

Each node consists of a processing element (PE) and router. The PE contains a processor and some local memory. The router has (n+1) input and (n+1) output channels. A node is connected to its neighbouring nodes through n inputs and n output channels. The remaining channels are used by the PE to inject/eject messages to/from the network respectively. Messages generated by the PE are transferred to the router through the injection channel. Messages at the destination are transferred to the local PE through the ejection channel. The router contains flit buffers for incoming virtual channels. The input and output channels are connected by a (n+1)-way crossbar switch which can simultaneously connect multiple input to multiple output channels in the absence of channel contention [2].

Many routing algorithms have been proposed for hypercubes. Given a message with source and destination addresses $x= x_n x_{n-1} \ldots x_1$ and $y= y_n y_{n-1} \ldots y_1$ in an n-dimensional hypercube, the current router (the router which message is in) chooses the next channel to be taken by the message as follows. If the current router address $C= c_n c_{n-1} \ldots c_1$, the router calculate bit pattern $B = C \oplus y$ (bit-wise exclusive-or of C and y) the set of channels which can be used to take the message closer to the destination node, are those positions in the pattern B which are "1" [6].

Choosing one of them in a fixed manner (say the first least significant "1"), the routing algorithm will be *deterministic* and referred to as e-cube routing [1]. Choosing a subset of this set will result in partially adaptive routing. For example, p-cube [8] divides this set into two subsets: one set including those positions that are "0" in C and "1" in y, and the other set including those positions that are "1" in C and "0" in y. According to e-cube routing, a message in its first step can take any of channels belonging to the first set and when all channels in the first set are all passed, it can choose any of the channels in the second set. Choosing any channel from the main

set in any order can result in a fully adaptive routing algorithm. Linder-Harden's, Duato's, and hop-based routing algorithms are examples of fully adaptive routing algorithms. However, making the algorithm deadlock free requires some careful consideration in order to avoid cyclic dependencies which may result in deadlock. We usually use virtual channels to develop such deadlock avoidance treatments.

3 Traffic Pattern and Load Distribution

Uniform Traffic Pattern Analysis: With uniform network traffic, all network nodes (except the source node) can be the destination of a message with equal probability. In this case, the number of nodes which are i hops away from a given node in an n-dimensional hypercube is $\binom{n}{i}$. Given that a uniform message can be destined to the other network nodes with equal probability, the probability that a uniform message generated at a given source node makes i hops to reach its destination, P_{u_i}, can be given by [1, 4]

$$P_{u_i} = \frac{\binom{n}{i}}{N-1}. \tag{1}$$

where N is the number of nodes in the network $N=2^n$. The average number of hops that a uniform message makes across the network is given by [16]

$$d_u = \sum_{i=1}^{n} i P_{u_i}. \tag{2}$$

Now, we can estimate the message arrival rate over each channel as [15]

$$\lambda_{channel} = \frac{N \lambda d_u}{nN} = \frac{\lambda d_u}{n}. \tag{3}$$

Note that the above rate is valid if the routing algorithm can distribute the traffic evenly over network channels. The e-cube deterministic routing and the fully adaptive routing algorithms shown in Figure 1 can balance the traffic over network channels [1, 9, 10, 11, 12] while p-cube partially adaptive routing algorithm can not [6] as will be shown in the next section.

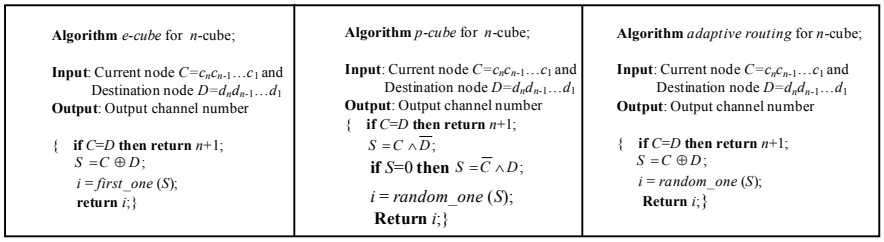


Fig. 1. Different routing algorithms in the nD hypercube; a) Deterministic, b) Partially adaptive, c) Fully adaptive.

The model used here to create non-uniform traffic pattern has been widely used in the literature. According to this model a node can partly create non-uniform traffic. To this end, we consider each node to generate non-uniform traffic with a probability

of x, where $0 \le x \le 1$, and creates messages with uniform destination distribution with a probability 1-x. Thus, x = 0 corresponds to a uniform traffic while x=1 indicates a complete non-uniform traffic defined by the non-uniform traffic generator used [16].

In what follows, we consider traffic analysis for three different popular traffic patterns used in the literature: *hotspot*, *bit-reversal*, and *matrix-transpose*.

Hotspot traffic pattern: According to the hotspot traffic distribution a node can create a message to the hotspot node with rate h [14]. It is easy to see that a node may create an i-hop hotspot node with the probability of

$$P_{h_i} = \frac{\binom{n}{i}}{N-1}. \tag{4}$$

Thus, the average number of hops that a hotspot message may traverse in the network is equal to that for the uniform traffic. Therefore, hotspot traffic does not change the average distance a message (hotspot or uniform) may take to reach its destination. However, it can make unbalanced traffic over network channel as the channels closer to the hotspot node will receive more messages than others.

Consider a channel that is j hops, $1 \le j \le n$, away from the hot-spot node. The probability that a hot-spot message has used this channel during its network journey can be derived as follows. Consider the set J of all the channels located j hops away from the hotspot node. The number of source nodes for which an element of J can act as an intermediate channel to reach the hotspot node is $N - \sum_{k=0}^{j-1}\binom{n}{k}$. Since there are $(n-j+1)\binom{n}{j-1}$ such intermediate channels (or elements in the set J), the hotspot message arrival rate at a channel located j hops away from the hotspot node is given be

$$\lambda_{h,channel_j} = \frac{\lambda h\left[N - \sum_{k=0}^{j-1}\binom{n}{k}\right]}{(n-j+1)\binom{n}{j-1}}. \tag{5}$$

The overall message arrival rate including hotspot and uniform messages can be then given by

$$\lambda_{channel_j} = (1-h)\lambda_{channel} + \lambda_{h,channel_j}. \tag{6}$$

where $\lambda_{channel}$ is given by equation (3).

Bit-reversal traffic pattern: Many applications, such as FFT, may produce bit-reversal non-uniform traffic pattern in the network [6, 16]. According to this traffic pattern, a source node $x_1x_2\cdots x_n$ sends message to node $B(x_1x_2\cdots x_n) = x_nx_{n-1}\cdots x_1$. Let us now calculate the probability, P_{β_i}, that a generated bit-reversal message makes i hops to cross the network.

Let $x = x_1x_2\cdots x_n$ be the source address and $B(x) = x'_1\,x'_2\cdots x'_n$ be the destination address for the bit-reversal message. When n is even every single bit difference be-

tween the first half parts of the source and destination addresses, $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$, results in another bit difference in the next halves, $x_{n/2} x_{n/2+1} \cdots x_n$ and $x'_{n/2} x'_{n/2+1} \cdots x'_n$. Thus, the probability that a bit-reversal message makes *i* hops, where *i* is odd, is zero. Let us calculate the number of possible ways that the source and destination nodes of the bit-reversal message are located *i* hops away from each other (*i*= 0, 2, 4, …, *n*). To do so, we can simply consider the first half of the address patterns of the source and destination nodes and calculate the number of possible combinations that $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$ are different in j bit positions for *j*= 0, 1, 2,…, *n*/2. A bit in the address $x_1 x_2 \cdots x_{n/2}$ with its corresponding bit in the address $x'_1 x'_2 \cdots x'_{n/2}$ make up four combinations, 00, 01, 10 and 11. In two combinations, 00 and 11, those two bits are equal while in the other two combinations, 01 and 10, they are different. Therefore, the number of possible combinations that $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$ are different in exactly *j* bits is $\binom{n/2}{j} 2^{n/2-j} 2^j$ (*j*= 0, 1, 2, …, *n*/2), and consequently the number of possible combinations that $x_1 x_2 \cdots x_n$ and $x'_1 x'_2 \cdots x'_n$ are different in exactly *i* bits (*i*= 0, 2, 4, …, *n*) is given by

$$n_{\beta_i} = \binom{\frac{n}{2}}{\frac{i}{2}} 2^{\frac{n}{2}-\frac{i}{2}} 2^{\frac{i}{2}} = 2^{\frac{n}{2}} \binom{\frac{n}{2}}{\frac{i}{2}} \quad (i = 0, 2, 4, \ldots, n). \tag{7}$$

When *n* is odd we can apply the above derivation for the bit patterns $x_1 x_2 \cdots x_{(n-1)/2} x_{(n+1)/2+1} \cdots x_n$ and $x'_1 x'_2 \cdots x'_{(n-1)/2} x'_{(n+1)/2+1} \cdots x'_n$. Note that the bit $x_{(n+1)/2}$ in *x* is equal to the bit $x'_{(n+1)/2}$ in $B(x)$. Therefore, the number of combinations where the bit patterns *x* and $B(x)$ are different in *i* bits (given by equation 4) is doubled considering the two possible values 0 and 1 for $x_{(n+1)/2}$ when *n* is odd. Combining these two cases (odd and even *n*) will result in a general equation for the number of possible combination that *x* and $\mathrm{B}(x)$ are different in exactly *i* bits (*i*= 0, 1,…, *n*) as

$$n_{\beta_i} = \begin{cases} \binom{\lfloor \frac{n}{2} \rfloor}{\frac{i}{2}} 2^{\lfloor \frac{n}{2} \rfloor + (n \bmod 2)} & \text{if i is even} \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

Thus, the probability that a bit-reversal message makes *i* hops to reach its destination can be written as

$$P_{\beta_i} = \frac{n_{\beta_i}}{N - n_{\beta_0}} = \begin{cases} \dfrac{\binom{\lfloor \frac{n}{2} \rfloor}{\frac{i}{2}}}{2^{\lfloor \frac{n}{2} \rfloor - (n \bmod 2)}}, & \text{if } i \text{ is even}. \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The average number of hops that a bit-reversal message makes across the network is given by

$$d_{bit-reversed} = \sum_{i=1}^{n} i P_{\beta_i}. \tag{10}$$

Using a similar model to combine hotspot and uniform traffic, we can assume that the generation rate of bit-reversal messages at a node is β and thus the rate for generating uniform messages is 1-β. Therefore, the average distance that a message takes in the network can be given as

$$d_\beta = \beta d_{bit-reversed} + (1-\beta) d_u . \tag{11}$$

Matrix-transpose traffic pattern: Many applications, such as matrix problems and signal processing, may produce matrix-transpose non-uniform traffic pattern in the network [6, 16]. According to this traffic pattern a source node $x_1 x_2 \cdots x_n$ sends messages to node $B(x_1 x_2 \cdots x_n) = x_{n/2} \cdots x_1 x_n \cdots x_{n/2+1}$ (when n is even) or $x_{(n-1)/2} \cdots x_1 x_n \cdots x_{(n-1)/2+1}$ (when n is odd). Let us now calculate the probability, P_{β_i}, that a generated bit-reversal message makes i hops to cross the network.

Let us now calculate the probability, P_{m_i}, that a newly-generated matrix-transpose message makes i hops to cross the network. Examining the address patterns generated by matrix-transpose permutations reveals that this probability has to be calculated in different ways for odd and even values of n (the dimensionality of the hypercube). Let $x = x_1 x_2 \cdots x_n$ be the source address and $M(x) = x'_1 x'_2 \cdots x'_n$ be the destination address for a matrix-transpose message. When n is even, every single bit difference between the first $n/2$ bit positions of the source and destination addresses, $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$, results in another bit difference in the remaining $n/2$ bit positions of addresses, $x_{n/2+1} x_{n/2+2} \cdots x_n$ and $x'_{n/2+1} x'_{n/2+2} \cdots x'_n$. Therefore, the probability that a matrix-transpose message makes i hops is zero when i is odd. Let us determine the number of possible cases where the source and destination of a matrix-transpose message are located i hops away from each other (i= 0, 2, 4, …, n). This can be done by simply considering only the first $n/2$ bit positions in the source and destination addresses, and thus enumerating the number of combinations where $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$ are different in exactly j bit positions (j= 0, 1, 2, …, $n/2$). Any bit in the address pattern $x_1 x_2 \cdots x_{n/2}$ with the corresponding bit in the pattern $x'_1 x'_2 \cdots x'_{n/2}$ make up four combinations, which are 00, 01, 10 and 11. In two combinations, 00 and 11, those two bits are equal while in the other two combinations, 01 and 10, they are different.

Therefore, the number of possible combinations that result in the patterns $x_1 x_2 \cdots x_{n/2}$ and $x'_1 x'_2 \cdots x'_{n/2}$ being different in exactly j bits is $\binom{n/2}{j} 2^{n/2-j} 2^j$ (j= 0, 1, 2, …, $n/2$). So, the number of possible combinations where $x_1 x_2 \cdots x_n$ and $x'_1 x'_2 \cdots x'_n$ are different in exactly i bits (i= 0, 2, 4, …, n) is given by

$$n_{m_i,even} = \binom{\frac{n}{2}}{\frac{i}{2}} 2^{\frac{n}{2}-\frac{i}{2}} 2^{\frac{i}{2}} = 2^{\frac{n}{2}} \binom{\frac{n}{2}}{\frac{i}{2}}, \quad i = 0, 2, 4, \ldots, n. \tag{12}$$

Consider the case where n is odd. Examining the address patterns of the source

$x_1x_2\cdots x_n$ and destination $x'_1\,x'_2\cdots x'_n$ generated by the matrix-transpose permutation shows that finding the number of combinations where these address patterns are different in exactly i bits (i= 0, 1, 2, …, n) is equivalent to the problem of finding the number of ways that i bits can be placed on a "fictive" circle such that no two adjacent bits on the circle be equal. It is can easily be checked that when i is odd there is no way to place i bits on the circle where no two adjacent bits are equal. When i is even, two configurations meet the desired condition. These i bits can be selected from n bits in $\binom{n}{i}$ different combinations resulting in a total of

$$n_{m_i,odd} = 2\binom{n}{i}, \quad i = 0, 2, 4, \ldots, n\text{-}1. \tag{13}$$

combinations where the address patterns $x_1x_2\cdots x_n$ and $x'_1\,x'_2\cdots x'_n$ are different exactly in i bits. Combining equations (4) and (5) gives a general equation for the number of possible combinations that result in the address patterns $x_1x_2\cdots x_n$ and $x'_1\,x'_2\cdots x'_n$ being different in exactly i bits (i= 0, 1,…, n) as

$$n_{m_i} = \begin{cases} 0 & \text{if } i \text{ is odd} \\ 2^{\frac{n}{2}}\binom{\frac{n}{2}}{\frac{i}{2}} & \text{if } (i \text{ is even})\text{and}(\ n \text{ is even}) \\ 2\binom{n}{i} & \text{if } (i \text{ is even})\text{and}(\ n \text{ is odd}) \end{cases}. \tag{14}$$

Thus, the probability that a matrix-transpose message makes i hops to reach its destination can be written as

$$P_{m_i} = \frac{n_{m_i}}{N - n_{m_0}} = \begin{cases} 0, & \text{if } i \text{ is odd} \\ \dfrac{\binom{\frac{n}{2}}{\frac{i}{2}}}{2^{\frac{n}{2}} - 1}, & \text{if } (i \text{ is even})\text{and}(\ n \text{ is even}) \\ \dfrac{\binom{n}{i}}{2^{n-1} - 1}, & \text{if } (i \text{ is even})\text{and}(\ n \text{ is odd}) \end{cases}. \tag{15}$$

The average number of hops that a matrix-transpose message makes across the network is given by

$$d_{matrix-transpose} = \sum_{i=1}^{n} iP_{m_i}. \tag{16}$$

Again assuming that the generation rate of bit-reversal messages at a node is m and thus the rate for generating uniform messages is 1-m. Therefore, the average distance that a message takes in the network can be given as

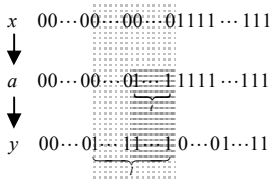
$$d_m = md_{matrix-transpose} + (1-m)d_u. \tag{17}$$

4 The effect routing algorithm

In this section, we show that even with uniform traffic pattern a routing algorithm may result in unbalanced traffic rate over network channels. With uniform traffic and the use of e-cube or fully adaptive routing the message arrival rate over network channels is balanced.

Let us now analyze the traffic rate over network channels when p-cube partially adaptive routing is used. The way in which this algorithm partitions channels and uses them in two steps, causes traffic load to become heavier in some corners of the network.

Assume the source node x and destination node y as shown below. Let us now calculate the probability that a message passes node a when going from x to y. It is easy to see that the number of combinations to have t bits of l zero bits equal to one is given by $\binom{l}{t}$. Out of these combinations only one pattern corresponds to node a as shown below.

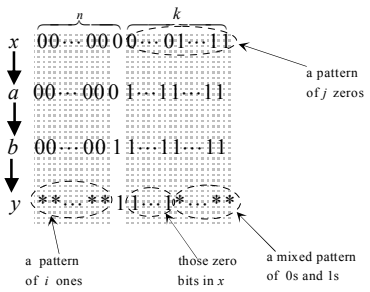


Thus, the probability to pass node a when traversing the network from node x to node y is given by

$$P_{x \xrightarrow{a} y} = \frac{1}{\binom{l}{t}}. \tag{18}$$

Assuming a hypercube, H_n, with p-cube routing, we now calculate the traffic arrival rate, $\lambda_{<a,b>}$, over channel $< a, b >$, where $a = \underbrace{00\cdots0}_{n-m}\underbrace{1\,1\cdots1}_{m}$, and $b = \underbrace{00\cdots0}_{n-m-1}\underbrace{1\,1\cdots1}_{m+1}$.

Note that all possible source and destination nodes, x and y, for which a and b are passed have the address form shown below



So, if x and y can be a source and destination node with a possible path between them including channel <a, b>, the k-th bit of x must be 0 while it is 1 in y. Note that bits 1, 2, …, k-1 in x can not be 1, otherwise channel <a, b> can not be passed. This is because if bit t, $t < k$, in x be 1, then the t-th bit in y can not be 1 as this does not result in passing channel <a, b>; also the t-th bit in y can not be 0, since we are still in the first phase of routing (note that still k-th bit has not changed) and again channel <a, b> can not be passed.

Also note that none of the two corresponding bits in x and y in the last m-bit part

can be 0 at the same time, otherwise channel <a, b> can not be passed.

Considering the above conditions we can see that x and y have the following attributes:

- i bits in the first (k-1)-bit part of y are equal to 1,
- j bits in the last m-bit part of x equals 0 while they are 1 in y,
- the remaining m-j bits of the last m-bit part are either 1 in x and 0 in y, or 1 in both x and y.

It is easy to see that the number of cases for which x and y fulfill the above conditions is given by $\binom{k}{i}\binom{m-1}{j}2^{m-j-1}$. The probability that a message from x to y passes node a along its path is given by equation (18) as $1/\binom{i+j+1}{j}$, and the probability that it passes channel <a, b> after node a is equal to $1/(i+1)$. Thus, the probability that a message from x to y passes channel <a, b> can be given by

$$P_{x \xrightarrow{<a,b>} y} = \frac{1}{\binom{i+j+1}{j}(i+1)}. \tag{19}$$

Recalling that the message generation rate of each node, e.g. node x, is λ, and that a message can be destined to any other $N = 2^{n+m} - 1$ nodes, and summing up all the possible combination of i's and j's (in above discussion), we can calculate the message arrival rate over channel <a, b> as

$$\lambda_{<a,b>} = \frac{\lambda}{2^n - 1} \sum_{i=0}^{n-m-1} \sum_{j=0}^{m} \frac{\binom{n-1}{i}\ \binom{m}{j}\ 2^{m-j}}{(i+j+1)\ \binom{i+j}{i}}. \tag{20}$$

Using a similar analysis, we can derive an expression to calculate the traffic arrival rate on channel $< a, b >$, where $a = \underbrace{00\cdots0}_{n-m}\underbrace{11\cdots1}_{m}$, and $b = \underbrace{00\cdots0}_{n-m+1}\underbrace{11\cdots1}_{m-1}$, as

$$\lambda_{<a,b>} = \frac{\lambda}{2^n - 1} \sum_{i=0}^{n-m} \sum_{j=0}^{m-1} \frac{\binom{n}{i}\ \binom{m-1}{j}\ 2^{m-j-1}}{(i+j+1)\ \binom{i+j}{i}}. \tag{21}$$

5 Discussions

In this section, we utilize the analytical expressions derived in previous sections. The network configuration used for our analysis is a 10-dimensional hypercube, for the sake of our presentation. Nodes generate messages using a Poisson model with an average message generation of $\lambda = 0.01$. However, the conclusions made for each case are valid for other network configurations and message generation rates at each node.

Figure 2 shows the message arrival rate over different channels of a 10-D hypercube for different hotspot traffic portions h = 0 (defining a pure uniform traffic), 0.2, 0.4, 0.6, 0.8 and 1.0 (corresponding to pure hotspot traffic). The obvious non-uniformity in message arrival rate over network channels (as can be seen in the figure) may result in significant performance degradation compared to the uniform traffic. As shown in this figure, the closer the channel is to the hotspot node, the higher the message arrival rate is. However, the arrival rate over the channel located

one hop away from the hotspot node is extremely higher than the rest of channels (those located some hops away from the hotspot node). In addition, the message arrival rates for channels located 3 hops away, or more, from the hotspot node are of little difference.

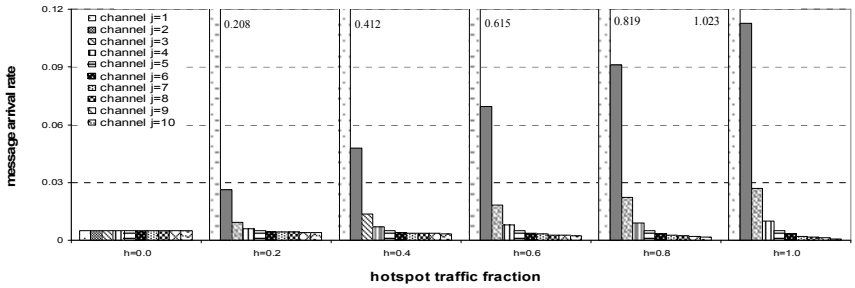


Fig. 2. The message arrival rate over different channels in a 10-cube for different hotspot rates.

Figure 3 shows the average number of hops that a message takes in different sized hypercubes under different bit-reversal traffic portions $\beta = 0$ (as the representative of pure uniform traffic pattern), 0.2, 0.4, 0.6, 0.8, 1.0 (as the pure bit-reversal traffic) when $\lambda = 0.01$. It can be seen in this figure that the average path length decreases in proportion with the increase in the bit-reversal traffic ratio β. The average path length, however, increases when network size (n) increases for different bit-reversal traffic portions β.

In Figure 4, the average number of hops that a message takes in different sized hypercubes under different matrix-transpose traffic portions $m = 0$ (as the representative of pure uniform traffic pattern), 0.2, 0.4, 0.6, 0.8, 1.0 (defining a pure matrix-transpose traffic) is shown when $\lambda = 0.01$. It can be seen in the figure, the average path length decreases proportionally when the matrix-transpose traffic portion increases. The interesting thing about the effect of matrix-transpose traffic pattern is the effect of network dimensionality.

It can be seen from the figure than for low matrix-transpose rates the average path length increases when network size increase. The opposite, however, is observed for high matrix-transpose traffic rates where the average path length decreases when network dimensionality increases. This is different from that observed under the bit-reversal traffic pattern.

Figure 5 shows the message arrival rate over network channels in the 10-D hypercube when $\lambda = 0.01$ and P-cube routing algorithm is used for uniform traffic pattern. As can be seen in the figure, although the traffic pattern used is uniform, the message arrival rates over different channels are very different. For the sake of comparison, the white bar shows the rate when e-cube or full-adaptive routing algorithms are used. In that case, the traffic rate over network channels is distributed evenly. The address pattern of nodes indicating the channel) is small and it gradually increases when the channel weight increases. This continues until the channel Hamming weight is approximately 8 (or 9) after which there is a little decrease again on channel arrival rate. Note that each bar represents the message arrival rate over a group of channels with equal Hamming weight.

6 Conclusions

The performance of an interconnection network mainly depends on the distribution of channel traffic. In most studies, it is shown when the traffic is evenly distributed over network channels that performance increases. There are many sources of network channel load imbalance, including network topology (which is solely dependent on the definition of the network), message destination address traffic pattern, and the routing algorithm employed.

In this paper, we have analytically studied the characteristics of network channel load and have derived some mathematical expression for predicting the traffic arrival rate over different network channels. We considered the effect of different well-known traffic patterns including uniform, hotspot, matrix-transpose, and bit-reversal traffic patterns on channel traffic rate. In addition, we have analyzed p-cube routing (a partially adaptive routing algorithm in hypercubes) and have shown that even under uniform traffic, the traffic load on different channels may vary.

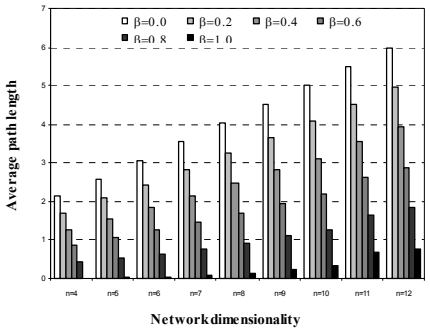


Fig. 3. The average path length messages take in a 10-cube for different bit-reversal traffic rates.

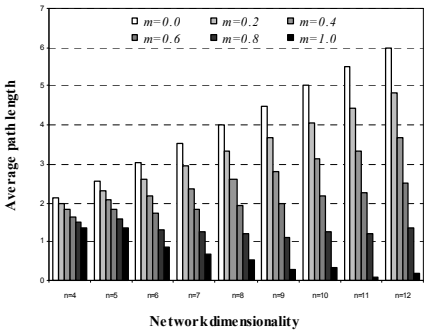


Fig. 4. The average path length messages take in a 10-cube for different matrix-transpose traffic rates.

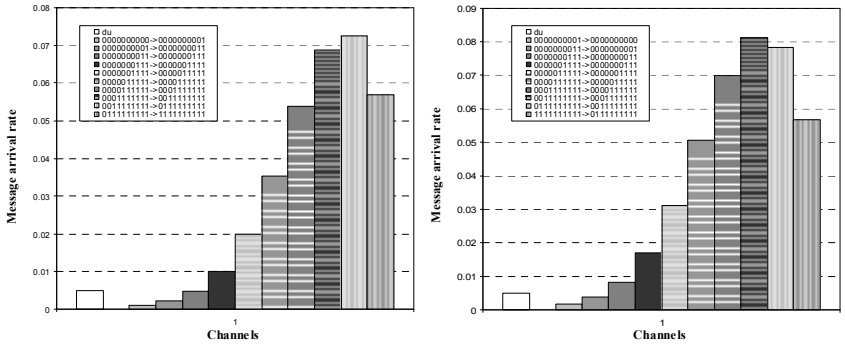


Fig. 5. The message arrival rate on different channels in a 10-cube using p-cube routing; a) negative channels, b) positive channels.

References

1. Abraham, S., Padmanabhan, K.,: Performance of the direct binary n-cube networks for multiprocessors. IEEE Trans. Computers, 37(7), 1989, pp. 1000-1011.
2. Boura, Y.: Design and analysis of routing schemes and routers for wormhole-routed mesh architectures, Ph.D. Thesis, Dept of Computer Sci. & Eng., Penn. State Univ., 1995.
3. Boppana, R.V., Chalasani, S.: A framework for designing deadlock-free wormhole routing algorithms. IEEE Trans. on Parallel and Distributed Systems, 7(2), 1996, pp. 169-183.
4. Boura, Y., Das, C.R., Jacob, T.M.: A performance model for adaptive routing in hypercubes. Proc. Int. Workshop Parallel Processing, pp. 11-16, Dec. 1994.
5. Dandamudi, S.: Hierarchical interconnection networks for multicomputer systems. Ph.D. Dissertation, Computer Science, Univ. Saskatchewan, Saskatoon, 1988.
6. Duato, J., Yalamanchili, S., Ni, L.: Interconnection networks: An engineering approach. IEEE Computer Society Press, 1997.
7. Duato, J.: A new theory of deadlock-free adaptive routing in wormhole networks. IEEE Trans. Parallel Distributed Systems, Vol. 4, No. 12, 1993, pp. 1320-1331.
8. Glass, C.J., Ni, L.M.: The turn model for adaptive routing. Proc. of 19th Int. Symposium Computer Architecture, 1992, pp. 278-287.
9. Guan, W.J., Tsai, W.K., Blough, D.: An analytical model for wormhole routing in multicomputer interconnection networks. Proc. Int. Conference Parallel Processing, 1993, pp. 650-654.
10. Hady, F.T., Menezes, B.L.: The performance of crossbar-based binary hypercubes. IEEE Trans. Computers, 44(10), 1995, pp. 1208-1215.
11. Horng, M., Kleinrock, L.: On the performance of a deadlock-free routing algorithm for boolean n-cube interconnection networks with finite buffers. Proc. Int. Conference Parallel Processing, 1991, pp. III-228-235.
12. Kim, J., Das, C.R.: Hypercube communication delay with wormhole routing. IEEE Trans. Computers, 43(7), 1994, pp. 806-814.
13. Linder, D.H., Harden, J.C.: An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. IEEE Trans. on Computers, 40(1,) 1991, pp. 2-12.
14. Pfister, G.J., Norton, V.A.: Hot-spot contention and combining in multistage interconnection networks. IEEE Trans. Computers, 34(10), pp. 943-948, 1985.
15. Sarbazi-Azad, H.: A mathematical model of deterministic wormhole routing in hypercube multicomputers using virtual channels. Journal of Applied Mathematical Modeling, 27(12), 2003, pp. 943-953.
16. Sarbazi-Azad, H.: Performance analysis of multicomputer interconnection networks. PhD Thesis, Glasgow University, Glasgow, UK, 2002.