University of Castilla–La Mancha

A publication of

Department of Computing Systems

Building $3D$ torus using low–profile expansion cards*

by

F.J. Andújar, J.A. Villar, F.J. Alfaro, J.L. Sánchez, J. Duato

Technical Report **#DIAB-11-02-3** February 2011

(*) This work has been jointly supported by the Spanish MEC, MICINN and European Commision FEDER funds under grants "Consolider Ingenio–2010 CSD2006-00046" and "TIN2009–14475–C04", respectively; and by Junta de Comunidades de Castilla–La Mancha under grants "PCC08–0078–9856", "POII10–0289–3724" and Beca Predoctoral de Investigación "07/096".

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS
ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE CASTILLA–LA MANCHA
CAMPUS UNIVERSITARIO s/n
02071, ALBACETE, SPAIN
Phone +34.967.599200, Fax +34.967.599224

Building $3D$ torus using low–profile expansion cards

Francisco J. Andújar, Juan A. Villar, Francisco J. Alfaro, José L. Sánchez
Dpto. de Sistemas Informáticos
Escuela Superior de Ingeniería Informática
Universidad de Castilla–La Mancha
02071 – Albacete, España
{fandujar, juanan, falfaro, jsanchez}@dsi.uclm.es

José Duato
Dpto. de Ingeniería de Sistemas y Computadores
Camino de Vera, s/n
Universidad Politécnica de Valencia
46022 – Valencia, España
jduato@gap.upv.es

February 2011

Contents

1 Abstract

Torus is a subclass of direct topologies that was defined to support n dimensions in theory. Although recently a supercomputer has been built on a network with six dimensions, the most common case is that only implementing three dimensions. In the market, there are low–profile communication expansion cards that have a reduced number of ports that is not enough to build tori of a certain number of dimensions. In this report, we will deal with four–port expansion cards. By means of one of these cards per node, a 2D torus topology could be built, but not a 3D torus topology. However, two of these cards could be used per node to build a 3D torus topology. In this case, two ports are used to interconnect both cards each other, and the other six ports to connect to six neighbors in the 3D torus. Theoretically, there are several ways of assigning the dimension and direction of the ports. This report presents a detailed study of the possible port configurations, and under specific network conditions the best of them is obtained.

2 Introduction

Nowadays, large supercomputers and clusters dominate the supercomputing market. These high node computing systems usually use high performance interconnection networks. The network topology plays a major role in determining the overall system performance. There are many factors that may affect the choice of an appropriate network topology. However, fat–tree [Lei85] and torus [DYN03] are the preferred topologies for indirect and direct networks, respectively, for these systems.

Torus topology belongs to the n–cube k–ary family that consists of n dimensions with k nodes in each dimension, with a total of k^n nodes. In particular, a 3D torus is a 3–cube k–ary topology. This topology has low radix and diameter, allowing an easy implementation and reducing the latency of the communications. It is also important that the scalability cost is linear. Moreover, 3D torus supports several routing algorithms that increase path diversity so that the fault tolerance and load balance become feasible. Additionally, the topology maps very well several well–known traffic patterns generated by current scientific-purpose applications. Specifically, applications that use 3D mathematical models fall inside this category. The 3D torus topology is one of the most common topologies used in the largest supercomputers in the Top500 list [DMS95]. Some examples are Cray's XT and XE families and IBM's Blue Gene family. There are three Cray supercomputers (Jaguar [Inc09], Cielo [VRB$^+$11], and Hopper [Inc11]) in the top 10 of the current Top500 list (November 2011). In previous Top500 lists, some Blue Gene supercomputers were also at the top (Blue Gene/L [AAA$^+$02], Blue Gene/P [IBM08]).

To obtain a 3D torus, six ports (or links) per node are needed, two for each dimension. In the market, there are low–profile expansion cards that have a variety of characteristics. Usually these low–profile expansion cards are incorporated in each node of a cluster. As today it is also usual that each node of the cluster is 1U (1.75

inches) tall, manufactures provide low–profile expansion cards with few communication ports. In general, nD tori can be built using several low–expansion cards per node if the number of ports of these cards is less than $2n$. In these cases, there are multiple ways of obtaining the $2n$ ports of every node from the ports of the internal cards.

We have focused on cards with a reduced number of ports, which can only be used to build a few network topologies. For example, it is possible to build 2D torus with four–port cards, but not a 3D torus. For applications using multidimensional models (e.g., 3D models), the logical model has to be mapped into a 2D physical network causing a reduction in the performance of the system. However, if two of these cards are used on each node the ports could be used in such a way that any node will be able to connect with its neighbors in a 3D torus. As shown in Figure 1, two ports would be used to interconnect the two cards, and the remaining ports for inter–node communication. We will call this new topology 3D Twin torus or just 3DT torus.

The important issue for the 3DT torus topology is that, in order to reduce the latency, we need to avoid as much as possible the paths, which pass through the node, use the two cards. If we have success, the cost of the communication would be noticeably reduced. The six ports of the node are split in two groups, and every group is assigned to one of the cards, as shown in Figure 1. The ports of the 3DT torus topology have an assigned dimension (i.e., X, Y, Z) and direction (i.e., positive or negative), which have to be established in the network deployment. There are several ways of assigning the dimension and direction to the ports. We refer to each of them as port configuration or simply configuration. However, to reduce the communication overhead we must try to make the traffic uses only one of the cards. Every configuration has a different performance level, even though uniform traffic pattern is used, and therefore we have to study the behavior of every port configuration in order to choose the best one.

Moreover, it is probable that *deadlock* appears in the 3DT torus network. This problem occurs because the link interconnecting the two cards can be used for any message in the network, independently of the dimension that the message is crossing. This causes new cycles in the 3DT torus network that do not appear in a 3D torus made directly with 6-port cards.

This report presents a detailed study of the behavior of these configurations and determines which is the best of them. Since the study is dependent on several factors, we limit the scope of this study considering a particular deterministic routing algorithm and traffic pattern. From these initial hypotheses and using the notation introduced in Section 3.1, all the possible configurations of the topology, which we called 3DT (**3D** **T**win) torus, are shown in Section 3.3. Section 5.1 describes informally the procedure that drives to the final solution, whereas Section 5.2 covers the whole study in a more formal way. Previously, in Section 4 we present a more simple study only considering topological aspects, and compare the proposed 3DT torus topology with a 2D torus topology with the same number of processing elements.

After the theoretical study, Section 6 presents the routing algorithm used in 3DT torus and the developed techniques to avoid deadlock. Once the routing algorithm is defined, in Section 7 we evaluate the different node configurations and compare the performance of a 2D torus and 3DT torus using a simulator. Finally, Section 8 presents the conclusions and the future work of this study.

3 3DT torus topology model

In this section we define the 3DT torus topology and show all the possible configurations of its basic communication hardware. Previously, we introduce the notation to be used in the rest of the document.

3.1 Notation

The following notation is used below:

- k: number of nodes in every dimension of the 3D torus. The same number of nodes in each dimension is assumed.
- $\langle x, y, z\rangle$: node identifier, $0 \leq x, y, z < k$.
- X^-, X^+: ports corresponding to dimension X.
- Y^-, Y^+: ports corresponding to dimension Y.
- Z^-, Z^+: ports corresponding to dimension Z.
- $\mathcal{P}$: set of ports of a node, $\mathcal{P} = \{X^-, X^+, Y^-, Y^+, Z^-, Z^+\}$.
- P: port of a node, $P \in \mathcal{P}$.
- $PE0$, $PE1$: processing elements of a node.
- $N_s^P(\langle x, y, z\rangle)$: set of nodes that send messages to node $\langle x, y, z\rangle$ and reach it through the port P.
- $N_d^P(\langle x, y, z\rangle)$: set of nodes to which the node $\langle x, y, z\rangle$ sends messages from its port P.
- $D_s^P(\langle x, y, z\rangle)$: cardinal of the set $N_s^P(\langle x, y, z\rangle)$.
- $D_d^P(\langle x, y, z\rangle)$: cardinal of the set $N_d^P(\langle x, y, z\rangle)$.
- $R_{P\rightarrow P'}(\langle x, y, z\rangle)$: number of paths that pass through the node $\langle x, y, z\rangle$ from input port P to output port P'. If both P and P' belong to the same dimension, sometimes we refer to the addition of $R_{P\rightarrow P'}(\langle x, y, z\rangle)$ and $R_{P'\rightarrow P}(\langle x, y, z\rangle)$ by an expression, using only the letter that identifies the dimension, hiding the sign of the direction in the dimension, and substituting the double arrow symbol by a single arrow ($R_{X\leftrightarrow X}(\langle x, y, z\rangle)$, $R_{Y\leftrightarrow Y}(\langle x, y, z\rangle)$ and $R_{Z\leftrightarrow Z}(\langle x, y, z\rangle)$).

- $[a, b]^n$: interval that defines a set of values between 0 and $n - 1$. Definition 5.1 shows more precisely this concept.
- D, d_{avg}: Diameter and average distance of a network, respectively.

Although in some specific cases we have simplified the notation usually used, it does not imply loss of accuracy.

3.2 3DT torus topology definition

A 3D Twin torus, or just 3DT torus, is a 3–cube k–ary (3D torus) topology, with $k \in \mathbb{N}^*$ and $k \geq 2$. Each node in this topology is a *virtual node*[1] consisting basically of the following main components:

- Hardware for communications: it consists of two four–port cards, offering a total of eight ports. Two of these ports (one of each card) are used to interconnect both cards to each other, and the six remaining ports are used to connect the node to the three dimensions in the 3D torus (two ports for each dimension).
- Computing hardware: each internal four–port card is connected to a processing element, and so there are two processing elements in each virtual node. Therefore, there are a total of $2k^3$ processing elements in the network.

Fig. 1 shows a network fragment and a detail of a node. In short, every node has two processing elements (*PE0* and *PE1*) and two communication cards (*Card0* and *Card1*).

3.3 3DT torus topology configuration

As the six available ports on each virtual node belong to two different communication cards, there are several alternatives to use the card ports to connect the node with its neighbors, i.e. there are several ways of assigning ports to dimensions (Fig. 1 only shows one of possible resultant port configurations).

The number of different configurations is the number of ways to combine three ports from a set of six. Hence,

$$\binom{6}{3} = \frac{6!}{3!\ 3!} = 20$$

[1] In this point, we use this term for better explaining how a node is formed in this topology. However, in most of the paper we will use the term *node* to refer it.

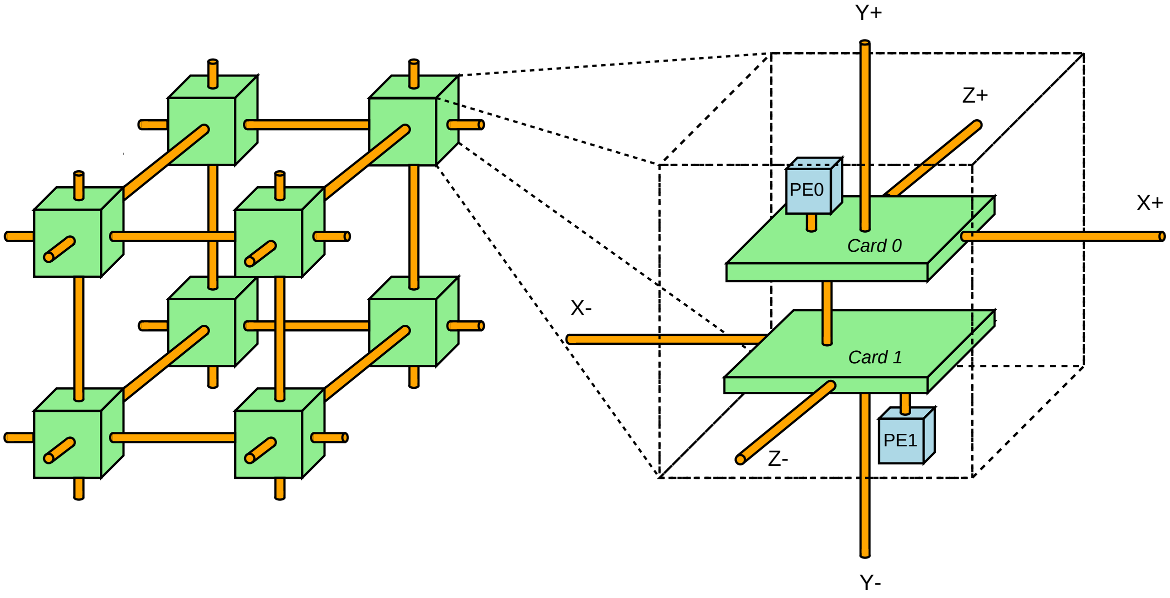


Figure 1: Fragment of a 3DT torus and detail of the communications hardware circuit, based on two 4-port cards.

Because of the symmetry between both cards, the 20 configurations may actually be reduced to only 10 different configurations, which are shown in Table 1. In this table, the different cases have been labelled from A to J, and the columns Card0 and Card1 show the dimension and direction of the three ports of each card.

Table 1: All different port configurations for the 3DT torus topology.

Case	Card 0	Card 1
A	$\{X^+, Y^+, Z^+\}$	$\{X^-, Y^-, Z^-\}$
B	$\{X^+, Y^+, Z^-\}$	$\{X^-, Y^-, Z^+\}$
C	$\{X^+, Y^+, Y^-\}$	$\{X^-, Z^+, Z^-\}$
D	$\{X^+, Y^+, X^-\}$	$\{Y^-, Z^+, Z^-\}$
E	$\{X^+, Y^-, Z^+\}$	$\{X^-, Y^+, Z^-\}$
F	$\{X^+, Y^-, Z^-\}$	$\{X^-, Y^+, Z^+\}$
G	$\{X^+, Y^-, X^-\}$	$\{Y^+, Z^+, Z^-\}$
H	$\{X^+, Z^+, Z^-\}$	$\{X^-, Y^+, Y^-\}$
I	$\{X^+, Z^+, X^-\}$	$\{Y^+, Y^-, Z^-\}$
J	$\{X^+, Z^-, X^-\}$	$\{Y^+, Y^-, Z^+\}$

Some of these configurations have similar behavior, but their performance may vary depending on the conditions taken into account when they are analyzed. In the following sections we present two studies in order to evaluate and compare the behavior of all the configurations shown in Table 1. In the first case, the study is only conducted from a topological viewpoint. To do this, we will consider two parameters characterizing a topology: diameter and average distance. The second study also considers a routing algorithm and a traffic pattern. In this case, a much more detailed analysis is required.

4 Analysis of the 3DT torus only considering the topology

In order to have a first view of the behavior of the configurations, we can use some topological parameters like the diameter and the average distance. These parameters will allow us to compare the different configurations. Moreover, the values of these parameters will be compared with the diameter and average distance of a 2D torus network with the same number of processing elements and based on the same 4-port cards. In what follows, this 2D torus topology will be referred to *equivalent 2D torus topology.*

To perform this comparative study, the following considerations[2] will be taken into account:

- To simplify the calculations, $k = 2^w$ has been considered, with $w > 0$. Therefore, there are $2 \times (2^w)^3 = 2^{3w+1}$ nodes in the network. If we consider k odd, the final conclusions of this study are the same, but the formal study is easier if k is even.
- Regarding the equivalent 2D torus topology:
  - If w is odd, $3w+1$ is even and it is possible to build a 2D torus with the same number of nodes in each dimension. Specifically, the equivalent topology is a $2^{\frac{3w+1}{2}} \times 2^{\frac{3w+1}{2}}$ torus.
  - If w is even, $3w + 1$ is odd and the number of nodes in each dimension will be different. In this case, the equivalent topology is a $2^{\frac{3w}{2}} \times 2^{\frac{3w}{2}+1}$ torus.
- To calculate the diameter and the average distance of the 3DT and 2D tori we will use the expressions deduced and included in the Appendix A of this document.
- There are several minimal paths between two PEs that are separated by the maximum distance. In this paper, we only explain the most simple and intuitive ways to obtain these paths, although there are more possibilities.
- The average distance of 3DT torus has been obtained by simulation, because obtaining this expression analytically is very complex. The simulator models the topology explained in Section 3.2. From a source PE, the simulator sends a message to every possible destination.

[2] These considerations are only taken into account for this first study.

Once a message has been injected in the network, the message is replicated when crosses a internal card and it is sent to each port card, except the source port. The messages stop replication when arrive to destination PE, the distance exceeds the diameter or distance is greater than the double of distance between source and destination node in a 3D torus topology made with 6–ports cards. In this way, the simulator can get the minimal path between each pair of nodes and calculate the average distance of the network.

Among the ten different configurations shown in Table 1, in terms of diameter and average distance, really there are only two different configuration types: A first class in which the configurations have the two ports of each dimension located on different internal cards; and a second configuration class where the two ports of only one dimension are separated into different internal cards. Configurations A, B, E and F correspond to the first case, and configurations C, D, G, H, I and J correspond to the second one. We will refer to the first set of configurations as *SC0* and we will use *SC1* for referring to the other set of configurations. However, all the configurations belonging to both sets have the same diameter and the difference in average distance between the configurations in the two sets is negligible. Specifically, these values are:

$$\begin{aligned} D &= 2^{w+1} \\ d_{avg} &\approx 2^{w} \end{aligned}$$

These results and the values obtained for the equivalent 2D torus topology are included in Table 2. In the following, we explain how the diameter is obtained for *SC0* and *SC1* sets. We use A and D as representative configurations of the sets *SC0* and *SC1*, respectively. However, the process can be applied to the other configurations of the sets. In both cases, we consider the distance between the PEs in the nodes $\langle 0,0,0\rangle$ and $\langle k/2,k/2,k/2\rangle$ to calculate the diameter of the 3DT torus.

Topology	Dimensions	D	d_{av}
3DT torus	$2^w \times 2^w \times 2^w$	2^{w+1}	2^w
2D torus (w even)	$2^{\frac{3w}{2}} \times 2^{\frac{3w}{2}+1}$	$3 \times 2^{\frac{3w}{2}-1}$	$3 \times 2^{\frac{3w}{2}-2}$
2D torus (w odd)	$2^{\frac{3w+1}{2}} \times 2^{\frac{3w+1}{2}}$	$2^{\frac{3w+1}{2}}$	$2^{\frac{3w+1}{2}-1}$

Table 2: Diameter and average distance of the 3DT torus topology and its equivalent 2D torus topology.

Diameter of SC0 set: As discussed above, there are many minimal paths between the PEs further away in the network. We present two ways to obtain these minimal paths using configurations in *SC0* set.

- **Zigzagging across two dimension and crossing the ring of remaining dimension.** We indicate in a schematic way how the diameter is obtained:

a) First, we choose the dimension that will be crossed in last place.

b) Next, we consider a 2D plane composed of the other two dimensions. A path between any pair of PEs separated by the largest distance can be established zigzagging through these dimensions without crossing any internal link. Thus, the distance is the same as in a 2D torus, i.e., $2\,(k/2) = k$ links.

c) Finally, the path goes across the last dimension. Since each dimension is a $2k$-ring[3], the path crosses k links to reach the furthest PE.

Therefore, the diameter of the network is $D = 2k$.

Example 4.1 *Given a 3DT torus with $k = 4$, the diameter is calculated considering PEs in the nodes $\langle 0,0,0\rangle$ and $\langle 2,2,2\rangle$. Following the previous steps, we have:*

a) Dimension Z is chosen as the last one to be crossed.

b) Now, we consider the plane XY (Fig. 2). Two paths are drawn, each one considers a different source communication card or PE. As shown in the figure, minimal paths from node $\langle 0,0\rangle$ to node $\langle 2,2\rangle$ can be established without crossing any internal link.

c) Finally, the path goes across the dimension Z. To move in this dimension it is necessary to cross the internal link of the half of the nodes in the Z-ring.

The path between the PEs further away in the network goes across a diagonal connecting them. A possible diagonal between the nodes $\langle 0,0,0\rangle$ and $\langle k/2,k/2,k/2\rangle$ will cross the node $\langle 1,1,1\rangle$, the node $\langle 2,2,2\rangle$, the node $\langle 3,3,3\rangle$, and so on until the node $\langle k/2,k/2,k/2\rangle$.

As the furthest node in the network is at the same distance in all the directions, we must choose a diagonal that allows to reach the next node in the diagonal without crossing any internal link. Once in this node, the path crosses the internal link to continue moving along the diagonal.

Fig. 3 shows some intermediate nodes of the path. Starting from $PE0$, a diagonal would cross the dimension X in the negative direction, the dimension Y in the positive direction, and finally, the dimension Z in the negative direction. The path must cross 4 links (3 external and one internal links) to reach the next node in the diagonal. As $k/2$ nodes have to be crossed along the diagonal, the diameter is:

$$D = 4 \times \frac{k}{2} = 2k$$

[3] Note that in the case of configurations in the set *SC0* the internal link must be used when crossing a node.

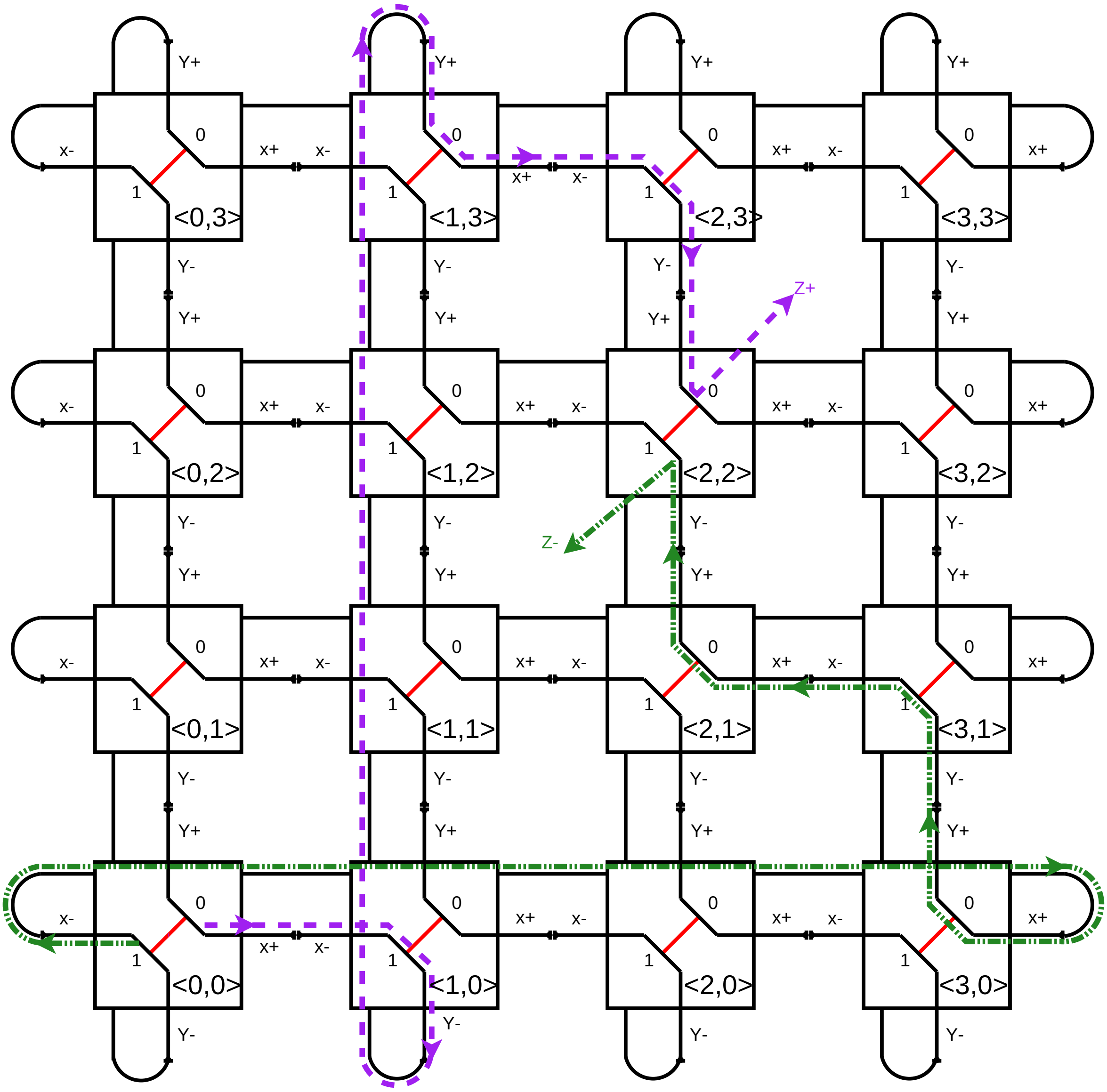


Figure 2: Plane XY of a 3DT torus, ($k = 4$, configuration A) and paths between the furthest PEs in the plane.

Diameter of SC1 set: In this case, we can obtain the diameter in the simplest way attending to the following steps:

a) The path starts crossing the dimension whose two ports are connected to the same card as the source PE. Thus, the path crosses completely that dimension without using any internal link. In total, the path crosses $k/2$ external links.

b) Next, the path crosses the dimension whose ports are not connected to the same internal card. As the destination PE is at the same distance in both directions, we choose the port connected to the current internal card, avoiding the use of the internal link. Thus, the path crosses $k/2$ external links and $k/2 - 1$ internal links.

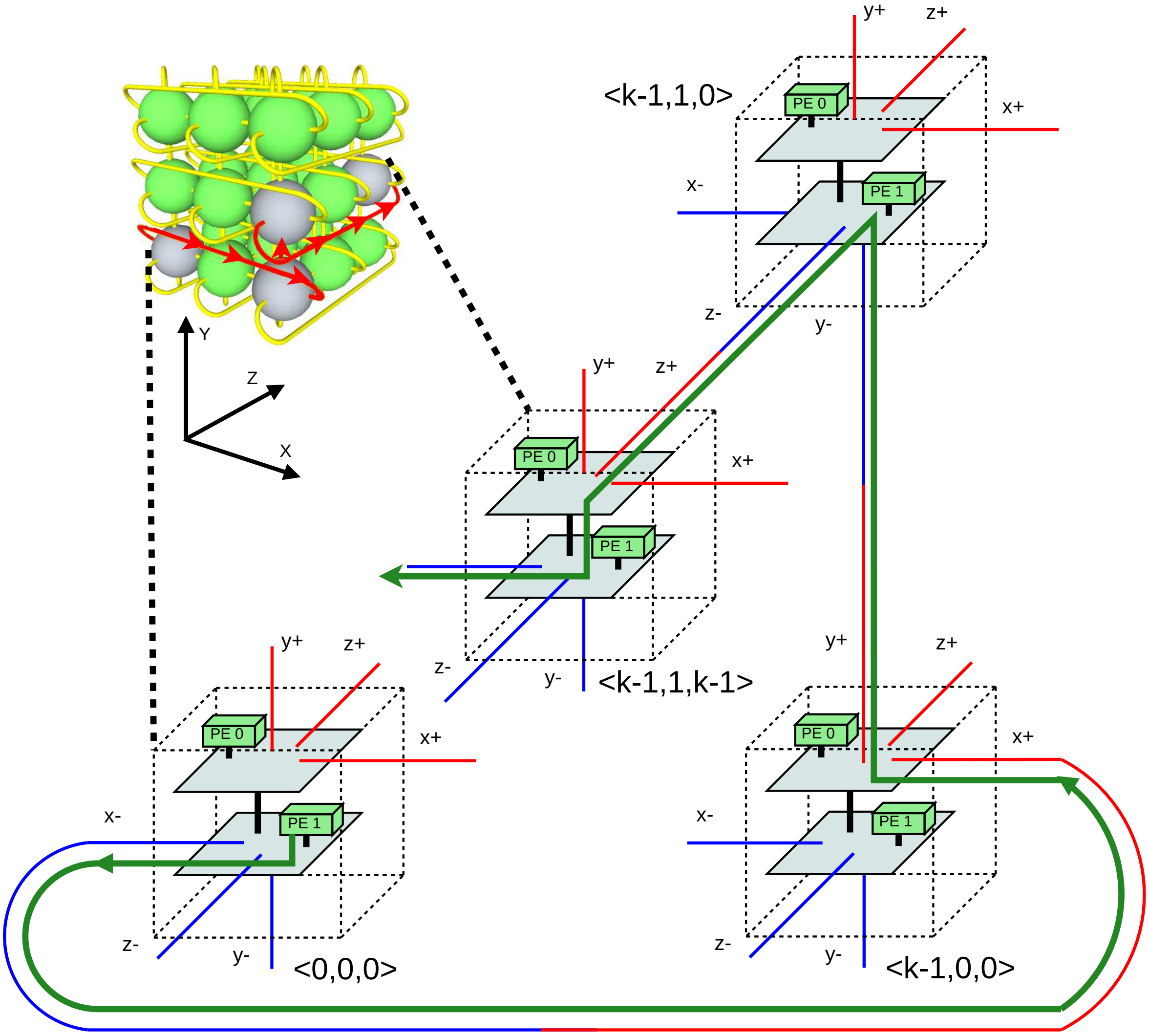


Figure 3: A possible diagonal path for configuration A.

c) After crossing the second dimension, the path is on an internal card connected to the ports of the last dimension. The path crosses this dimension using $k/2$ external links.

d) When the destination node is reached, it is necessary to cross the internal link to arrive at the PE located in the other card because it is the furthest PE from the source PE.

Thus, the diameter of the network is :

$$D = \frac{k}{2} + \frac{k}{2} + \left(\frac{k}{2} - 1\right) + \frac{k}{2} + 1 = 2k$$

Fig. 4 shows two paths of this type for the configuration D. The path originating at *PE0* crosses the dimensions in order $X - Y - Z$, whereas the path originating at *PE1* crosses the dimensions in order $Z - Y - X$.

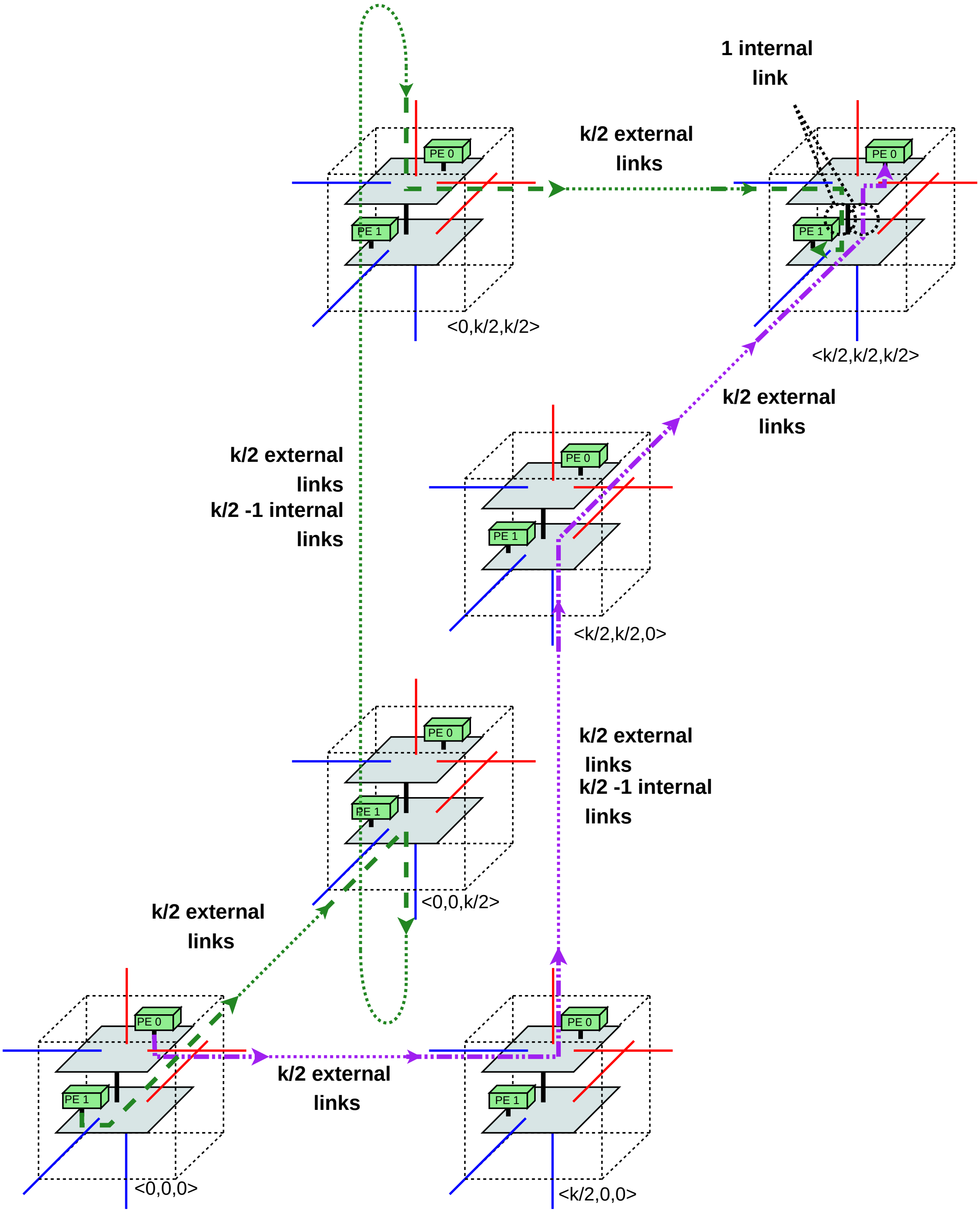


Figure 4: Paths between furthest PEs for the configuration D.

In summary, the configurations in both sets, *SC0* and *SC1*, have the same diameter, i.e., $2k$. If $k = 2^w$, then the diameter is, in both cases:

$$D = 2\times 2^w = 2^{w+1}$$

Torus 2D / Torus 3DT comparative: Finally, we compare D and d_{avg} values for the 3DT torus topology with those obtained for the equivalent 2D torus topology. Since in all cases $D = 2\times d_{avg}$, we only compare one of these parameters. Specifically, diameter was chosen for this study. Let us assume the diameter in the 2D torus is greater than in the 3DT torus.

- If w is odd:

$$\begin{aligned} D_{2D} &> D_{3DT} \\ 2^{\frac{3w+1}{2}} &> 2^{w+1} \\ \frac{3w+1}{2} &> w+1 \\ 3w+1 &> 2w+2 \\ w &> 1 \end{aligned}$$

- If w is even:

$$\begin{aligned} D_{2D} &> D_{3DT} \\ 3\times 2^{\frac{3w}{2}-1} &> 2^{w+1} \\ \log_2 3 + \frac{3w}{2} - 1 &> w+1 \\ \frac{3w}{2} &> w+2-\log_2 3 \\ 3w &> 2w+4-2\log_2 3 \\ w &> 4-2\log_2 3 \approx 0.830 \end{aligned}$$

That is, for $w \geq 2$ ($k \geq 4$, 64 or more *PEs*), whether w is even or odd, the 3DT torus topology with two internal cards per node has smaller diameter and average distance than the 2D torus topology with one internal card per node.

5 Analysis of the 3DT torus also considering routing and traffic

In this second study we introduce the routing mechanism and the network workload.

- Routing algorithm is deterministic, dimension order routing (DOR) [DYN03]. In the case of k is even and when the shortest distance between a source–destination pair of nodes is the same through positive and negative directions, then the positive link is chosen.
- The traffic pattern is uniform. Based on the statistical characteristics of the traffic pattern, in this case the network load can be expressed either in terms of the number of paths or the number of messages because the conclusions are the same. Therefore, we develop the study in terms of paths instead messages.

5.1 Informal Description

This section briefly describes in an informal way the methodology used to obtain the optimal ports configuration of the communication cards included in any node of the 3DT torus. Note that we consider the best configuration that minimizes the number of paths that go across a node by using the two cards (i.e. paths using the internal link that connects the two cards).

The number of paths that pass through the node $\langle x, y, z\rangle$, $0 \leq x, y, z < k$, can be calculated by the Cartesian product of all the possible source-destination node pairs whose paths include such a node. For example, the number of paths that pass through the node $\langle x, y, z\rangle$ using the port X^- as input port and the port Y^+ as output port, is the result of multiplying the number of nodes that send messages to node $\langle x, y, z\rangle$ reaching it through the port X^- and the number of nodes to which the node $\langle x, y, z\rangle$ sends messages from its port Y^+. Hence, using the notation introduced in Section 3.1:

$$R_{X^- \to Y^+}(\langle x, y, z\rangle) = D_s^{X^-}(\langle x, y, z\rangle) \times D_d^{Y^+}(\langle x, y, z\rangle)$$

As we will shown below, this expression can be applied only when the input port and the output port do not belong to the same dimension. Otherwise, it is necessary to apply a more complex procedure to calculate the correct number of paths.

In any case, if the paths using the ports of a node and the configuration of the two cards are both known, then the number of paths that pass through the internal link that connects the two cards can be calculated. Thus, the basic procedure consists in determining all the possible configurations and selecting that which minimizes the number of paths that pass through the two cards.

Note that routing will be considered at node level, but not at processing element level. If routing at processing element level is considered, the injection of messages from the processing elements would imply to multiply by two the number of paths passing through the internal link. However, the final result would be the same because all the expressions would be multiplied by the same factor.

For a given node, only paths passing through it will be considered, and paths with origin or destination at that node will be not taken into account. Under uniform traffic, each processing element sends and receives the same amount of messages by each port. Thus, messages from *PE0* or messages from *PE1* must use the internal link to get any port P. In any case, the number of messages using the internal link will be the same whatever the chosen configuration. Therefore, paths starting or finishing in that node will not affect the results obtained for a given configuration.n.

Summing up, the methodology we will use consists of several steps:

A. To obtain the sets $N_s^P(\langle x, y, z\rangle)$ and $N_d^P(\langle x, y, z\rangle)$, where $0 \leq x, y, z < k$ and $P \in \mathcal{P}$.

B. To obtain their cardinals $D_s^P(\langle x, y, z\rangle)$ and $D_d^P(\langle x, y, z\rangle)$, where $0 \leq x, y, z < k$ and $P \in \mathcal{P}$.

C. For each of the ten configurations in Table 1, to calculate the number of paths $R_{P \to P'}(\langle x, y, z\rangle)$ that cross the node $\langle x, y, z\rangle$ using the link that interconnects the two cards, where $0 \leq x, y, z < k$, $P \neq P'$ and $P, P' \in \mathcal{P}$.

D. Finally, to search for the optimal configurations.

Note that in all calculations involving the above steps, we will distinguish the case of odd k and even k because the results are different for both cases.

5.2 Formal Study

This section develops a detailed study that has been conducted to determine the best way to use the ports of the two communication cards, whose union can establish connections with neighboring nodes to form a 3DT torus. The study is presented according to the steps indicated in Section 5.1.

5.2.1 Useful definitions

Firstly, we include a definition to simplify some parts of the study.

Definition 5.1 *The notation $[a,b]^n$, with $b = a+m$; $a, b \in \mathbb{Z}$ and $n, m \in \mathbb{N}$, defines a set of nodes whose members are $\{(a)\ mod\ n, (a+1)\ mod\ n, \ldots, (a+m-1)\ mod\ n, (a+m)\ mod\ n\}$*

We define the operation module, $a\ mod\ k$, where $a \in \mathbb{Z}$ and $k \geq 2$, as the remainder of integer division: $a\ mod\ k = (a+k)\ mod\ k$. As an example, $(-3)\ mod\ 7 = 4$ and $(-1)\ mod\ 7 = 6$.

Proposition 5.1 *The cardinal of the set that is defined by the interval $[a,b]^n$ is $b-a+1$.*

Proof: By definition, we know the interval $[a,b]^n$ defines a set with the following values: $a, a+1, a+2, \ldots, a+m-1, a+m$. Because $b = a+m \Rightarrow m = b-a$ and the cardinal of the set defined by the interval $[a,b]^n$ is $m+1 = b-a+1$. □

5.2.2 Sets N_s^P and N_d^P for the node $\langle x, y, z\rangle$

Based on any 3D torus topology and the DOR routing algorithm, it is easy to determine the nodes that belong to the sets N_s^P and N_d^P. Next, we indicate the members of these sets using set terminology. We distinguish between odd and even k values. Thus, if k is odd the number of reachable nodes from a specific node that are located in the same dimension is $\frac{k-1}{2}$ regardless the direction. However, if k is even the result changes in function of the direction because the distance between some nodes is the same in both directions. The load of the links depends on the direction that is chosen in those cases. As mentioned above, in this study we take always the positive direction.

Definition 5.2 *Let $N_s^{X^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port X^- $(0 \leq x, y, z < k)$, whose members are defined as follows:*

$$N_s^{X^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ :\ x' \in \begin{cases} \left[x - \frac{k-1}{2}, x-1\right]^k & \textit{if } k \textit{ is odd} \\ \left[x - \frac{k}{2}, x-1\right]^k & \textit{if } k \textit{ is even} \end{cases},\ y' = y,\ z' = z\}$$

Definition 5.3 *Let $N_s^{X^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port X^+ $(0 \leq x, y, z < k)$, whose members are defined as follows:*

$$N_s^{X^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ :\ x' \in \begin{cases} \left[x+1, x + \frac{k-1}{2}\right]^k & \textit{if } k \textit{ is odd} \\ \left[x+1, x + (\frac{k}{2} - 1)\right]^k & \textit{if } k \textit{ is even} \end{cases},\ y' = y,\ z' = z\}$$

Definition 5.4 *Let $N_s^{Y^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Y^- ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_s^{Y^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x' < k,\ y' \in \begin{cases} \left[y - \frac{k-1}{2}, y-1\right]^k & \text{if } k \text{ is odd} \\ \left[y - \frac{k}{2}, y-1\right]^k & \text{if } k \text{ is even} \end{cases},\ z' = z\}$$

Definition 5.5 *Let $N_s^{Y^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Y^+ ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_s^{Y^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x' < k,\ y' \in \begin{cases} \left[y+1, y + \frac{k-1}{2}\right]^k & \text{if } k \text{ is odd} \\ \left[y+1, y + (\frac{k}{2} - 1)\right]^k & \text{if } k \text{ is even} \end{cases},\ z' = z\}$$

Definition 5.6 *Let $N_s^{Z^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Z^- ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_s^{Z^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x',y' < k,\ z' \in \begin{cases} \left[z - \frac{k-1}{2}, z-1\right]^k & \text{if } k \text{ is odd} \\ \left[z - \frac{k}{2}, z-1\right]^k & \text{if } k \text{ is even} \end{cases}\}$$

Definition 5.7 *Let $N_s^{Z^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Z^+ ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_s^{Z^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle\ : 0 \le x',y' < k, z' \in \begin{cases} \left[z+1, z + \frac{k-1}{2}\right]^k & \text{if } k \text{ is odd} \\ \left[z+1, z + (\frac{k}{2} - 1)\right]^k & \text{if } k \text{ is even} \end{cases}\}$$

Definition 5.8 *Let $N_d^{X^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port X^- ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{X^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' \in \begin{cases} \left[x - \frac{k-1}{2}, x-1\right]^k & \text{if } k \text{ is odd} \\ \left[x - (\frac{k}{2} - 1), x-1\right]^k & \text{if } k \text{ is even} \end{cases},\ 0 \le y',z' < k\}$$

Definition 5.9 *Let $N_d^{X^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port X^+ ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{X^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x' \in \begin{cases} \left[x+1, x + \frac{k-1}{2}\right]^k & \text{if } k \text{ is odd} \\ \left[x+1, x + \frac{k}{2}\right]^k & \text{if } k \text{ is even} \end{cases},\ 0 \le y',z' < k\}$$

Definition 5.10 *Let $N_d^{Y^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Y^- ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{Y^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x'=x,\ y' \in \begin{cases} \left[y-\frac{k-1}{2}, y-1\right]^k & \textit{if } k \textit{ is odd} \\ \left[y-(\frac{k}{2}-1), y-1\right]^k & \textit{if } k \textit{ is even} \end{cases},\ 0 \le z' < k\}$$

Definition 5.11 *Let $N_d^{Y^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Y^+ ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{Y^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x'=x,\ y' \in \begin{cases} \left[y+1, y+\frac{k-1}{2}\right]^k & \textit{if } k \textit{ is odd} \\ \left[y+1, y+\frac{k}{2}\right]^k & \textit{if } k \textit{ is even} \end{cases},\ 0 \le z' < k\}$$

Definition 5.12 *Let $N_d^{Z^-}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Z^- ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{Z^-}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x'=x,\ y'=y,\ z' \in \begin{cases} \left[z-\frac{k-1}{2}, z-1\right]^k & \textit{if } k \textit{ is odd} \\ \left[z-(\frac{k}{2}-1), z-1\right]^k & \textit{if } k \textit{ is even} \end{cases}\}$$

Definition 5.13 *Let $N_d^{Z^+}(\langle x,y,z\rangle)$ be a set of nodes that send messages to node $\langle x,y,z\rangle$ by using its port Z^+ ($0 \le x,y,z < k$), whose members are defined as follows:*

$$N_d^{Z^+}(\langle x,y,z\rangle) = \{\ \langle x',y',z'\rangle :\ x'=x,\ y'=y,\ z' \in \begin{cases} \left[z+1, z+\frac{k-1}{2}\right]^k & \textit{if } k \textit{ is odd} \\ \left[z+1, z+\frac{k}{2}\right]^k & \textit{if } k \textit{ is even} \end{cases}\}$$

5.2.3 D_s^P and D_d^P values for the node $\langle x,y,z\rangle$

Applying Proposition 5.1 to the sets defined in the Section 5.2.2, the value of D_s^P and D_d^P for the node $\langle x,y,z\rangle$ can be obtained. Remember that these values are the cardinals of the sets N_s^P and N_d^P. Tables 3 and 4 include the values of D_s^P and D_d^P, respectively.

Table 3: Number of nodes that send messages to node $\langle x, y, z \rangle$.

$$D_s^{X-}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2} & \text{if } k \text{ is odd} \\[2ex] \dfrac{k}{2} & \text{if } k \text{ is even} \end{cases}$$

$$D_s^{X+}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2} & \text{if } k \text{ is odd} \\[2ex] \dfrac{k}{2} - 1 & \text{if } k \text{ is even} \end{cases}$$

$$D_s^{Y-}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{if } k \text{ is odd} \\[2ex] \dfrac{k^2}{2} & \text{if } k \text{ is even} \end{cases}$$

$$D_s^{Y+}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{if } k \text{ is odd} \\[2ex] \left(\dfrac{k}{2} - 1\right)k & \text{if } k \text{ is even} \end{cases}$$

$$D_s^{Z-}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{if } k \text{ is odd} \\[2ex] \dfrac{k^3}{2} & \text{if } k \text{ is even} \end{cases}$$

$$D_s^{Z+}(\langle x, y, z \rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{if } k \text{ is odd} \\[2ex] \left(\dfrac{k}{2} - 1\right)k^2 & \text{if } k \text{ is even} \end{cases}$$

Table 4: Number of nodes to which the node $\langle x, y, z\rangle$ sends messages.

$$D_d^{X^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{if } k \text{ is odd} \\ \left(\dfrac{k}{2}-1\right)k^2 & \text{if } k \text{ is even} \end{cases}$$

$$D_d^{X^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k^2 & \text{if } k \text{ is odd} \\ \dfrac{k^3}{2} & \text{if } k \text{ is even} \end{cases}$$

$$D_d^{Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{if } k \text{ is odd} \\ \left(\dfrac{k}{2}-1\right)k & \text{if } k \text{ is even} \end{cases}$$

$$D_d^{Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2}k & \text{if } k \text{ is odd} \\ \dfrac{k^2}{2} & \text{if } k \text{ is even} \end{cases}$$

$$D_d^{Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{if } k \text{ is odd} \\ \dfrac{k}{2}-1 & \text{if } k \text{ is even} \end{cases}$$

$$D_d^{Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{k-1}{2} & \text{if } k \text{ is odd} \\ \dfrac{k}{2} & \text{if } k \text{ is even} \end{cases}$$

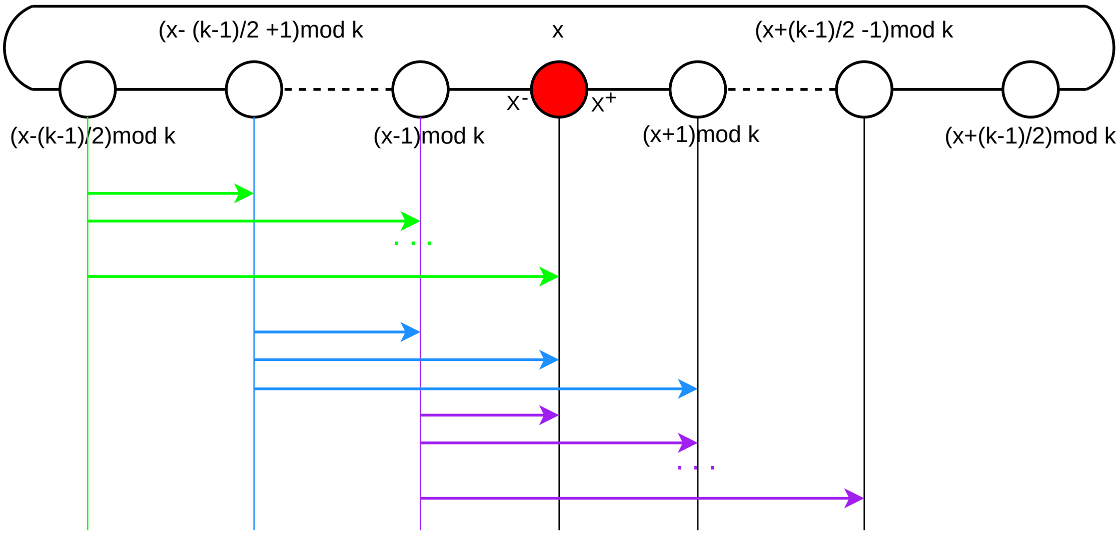


Figure 5: Paths that pass through a node in the dimension X.

5.2.4 Paths that pass through the node $\langle x, y, z\rangle$

In this section we calculate the number of paths that pass through the node $\langle x, y, z\rangle$ for each input–output pair of ports. To obtain this value we do a simple multiplication as we noted in Section 5.1, using the expressions obtained in Section 5.2.3. However, in some cases, we must take into account other considerations. So, there are two situations to be considered:

- When the input and output ports belong to the same dimension (Proposition 5.2).
- When the input and output ports belong to different dimensions (Proposition 5.3).

Proposition 5.2 *Given a node $\langle x, y, z\rangle$, the paths that cross this node from the input port P to the output port P', where $P \neq P'$ and they both belong to the same dimension, that is, $P, P' \in \{X^+, X^-\}$ or $P, P' \in \{Y^+, Y^-\}$ or $P, P' \in \{Z^+, Z^-\}$ are:*

$$R_{X\leftrightarrow X}(\langle x, y, z\rangle) = R_{Y\leftrightarrow Y}(\langle x, y, z\rangle) = R_{Z\leftrightarrow Z}(\langle x, y, z\rangle) = \begin{cases} \dfrac{(k-1)(k-3)}{4}k^2 & \textit{if } k \textit{ is odd} \\ \\ \dfrac{(k-2)^2}{4}k^2 & \textit{if } k \textit{ is even} \end{cases}$$

Proof: We firstly demonstrate the property when the paths are bound to the X dimension. A similar procedure for demonstrating the property for Y and Z dimensions will be performed.

Figure 5 shows a subset of the 3D torus nodes, which belong to the X dimension. For simplicity, we use only the digits corresponding to the X dimension to identify each node.

In the following we are going to distinguish the both cases: k is odd and k is even.

k is odd

Taking into account uniquely the nodes in the X dimension, that is, considering the sub–path that belongs to the X dimension, we can see in Figure 5 that

- there is no path from the node $(x - \frac{k-1}{2}) mod\ k$ that goes across x from the input port X^- to the output port X^+.
- there is only one path from the node $(x - \frac{k-1}{2} + 1) mod\ k$ that passes through the node x from the input port X^- to the output port X^+.
- there are two paths from the node $(x - \frac{k-1}{2} + 2) mod\ k$ that pass through the node x from the input port X^- to the output port X^+.
- ...
- there are $\frac{k-1}{2} - 1$ paths from the node $(x - 1) mod\ k$ that go across the node x from the input port X^- to the output port X^+.

Thus, there is a total number of $\sum_{i=1}^{\frac{k-1}{2}-1} i$ paths that pass through the node x from the input port X^- to the output port X^+. After assuming that these paths have the source and destination nodes in the same dimension, X, and as k is odd, then there is the same number of paths when the input port is X^+ and the output port is X^-. However, the paths that pass through a node in the X dimension could also change to the other two dimensions. Therefore, because the destination node could be any of the k^2 nodes which can be reached from Y and Z dimensions.

Hence, the total number of paths passing through the node $\langle x, y, z \rangle$, entering and leaving by any pair of ports in the X dimension is given by

$$\begin{aligned} R_{X \leftrightarrow X}(\langle x,y,z\rangle) &= R_{X^- \to X^+}(\langle x,y,z\rangle) + R_{X^+ \to X^-}(\langle x,y,z\rangle) = 2 \sum_{i=1}^{\frac{k-1}{2}-1} i k^2 = \\ &= 2 \frac{(\frac{k-1}{2} - 1)(\frac{k-1}{2})}{2} k^2 = \frac{k-3}{2} \frac{k-1}{2} k^2 = \frac{(k-1)(k-3)}{4} k^2 \end{aligned}$$

k is even

By similar reasoning, the number of paths that pass through the node $\langle x, y, z\rangle$ by using any pair of ports in the dimension X when k is even, could be obtained. However, the number of paths passing through the node $\langle x, y, z\rangle$ from the input port X^- to the output port X^+ is not equal to the number of paths crossing from the input port X^+ to the output port X^-.

Thus, if k is even, we obtain the following expression:

$$R_{X\leftrightarrow X}(\langle x,y,z\rangle) = R_{X^-\rightarrow X^+}(\langle x,y,z\rangle) + R_{X^+\rightarrow X^-}(\langle x,y,z\rangle) = \sum_{i=1}^{\frac{k}{2}-1} ik^2 + \sum_{i=1}^{\frac{k}{2}-2} ik^2$$

Therefore, the number of paths that pass through the node $\langle x, y, z\rangle$ by using whatever pair of ports in the X dimension is given by:

$$\begin{aligned} R_{X\leftrightarrow X}(\langle x,y,z\rangle) &= \left(\sum_{i=1}^{\frac{k}{2}-1} i + \sum_{i=1}^{\frac{k}{2}-2} i\right) k^2 = \left(2\sum_{i=1}^{\frac{k}{2}-2} i + \left(\frac{k}{2}-1\right)\right) k^2 = \\ &= \left(2\frac{\left(\frac{k}{2}-1\right)\left(\frac{k}{2}-2\right)}{2} + \left(\frac{k}{2}-1\right)\right) k^2 \\ &= \left(\frac{k}{2}-1\right)\left(\frac{k}{2}-2+1\right) k^2 = \left(\frac{k}{2}-1\right)^2 k^2 = \frac{(k-2)^2}{4} k^2 \end{aligned}$$

As we stated, the dimensions Y and Z have a similar treatment. For the paths traveling along the dimension Y, the calculations are performed in the same way as for the dimension X. Moreover, given a path passing through the node $\langle x, y, z\rangle$ in the dimension Y, its source node can be any of the nodes in the dimension X that can send messages to the node $\langle x, y, z\rangle$. On the other hand, the path can continue to k different nodes passing through the dimension Z.

Regarding Z dimension, if the path is finally traveling along the Z dimension, its source node can be one of k^2 nodes in the X and Y dimensions. Summing up, $R_{X\leftrightarrow X}(\langle x,y,z\rangle) = R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) = R_{Z\leftrightarrow Z}(\langle x,y,z\rangle)$. □

Proposition 5.3 *The number of paths that pass through the node $\langle x, y, z\rangle$ from the input port P to the output port P', where $P, P' \in \mathcal{P}$, $P \neq P'$ and they are located in different dimension is*

$$R_{P\rightarrow P'}(\langle x,y,z\rangle) = \begin{cases} 0 & \textit{if the direction } P \rightarrow P' \textit{ is not permitted} \\ D_s^P(\langle x,y,z\rangle) \times D_d^{P'}(\langle x,y,z\rangle) & \textit{if the direction } P \rightarrow P' \textit{ is permitted} \end{cases}$$

Proof: If the routing algorithm does not permit the transition $P \rightarrow P'$, then none of the paths will use P' after using P. Thus, no path passes through the node $\langle x, y, z\rangle$ from input port P to output port P'.

For permitted $P \to P'$ transitions, the source node of the path that passes through the node $\langle x, y, z\rangle$ from the input port P is any of the nodes that could reach the node $\langle x, y, z\rangle$ by the input port P, that is, any member of the set $N_s^P(\langle x, y, z\rangle)$. Similarly, the destination node of the path leaving the node $\langle x, y, z\rangle$ by the output port P', could be any of the nodes which are reachable from the node $\langle x, y, z\rangle$ through P', that is, any of the members of the set $N_d^{P'}(\langle x, y, z\rangle)$. Therefore, the total number of paths that go across the node $\langle x, y, z\rangle$ from the input port P to the output port P', is obtained by the following multiplication:

$$\mathrm{card}(N_s^P(\langle x, y, z\rangle)) \times \mathrm{card}(N_d^{P'}(\langle x, y, z\rangle)) = D_s^P(\langle x, y, z\rangle) \times D_d^{P'}(\langle x, y, z\rangle)$$

□

Applying Proposition 5.3 to any situation that involves a change of dimension while the path is passing through the node $\langle x, y, z\rangle$, the number of paths that pass through such a node for those situations can be calculated. Table 5 and Table 6 have a summary of them.

5.2.5 Evaluation of the port configurations

For a given configuration, and after obtaining the number of paths that pass through the node $\langle x, y, z\rangle$ considering all the pairs of ports, it is possible to calculate how many paths go across the internal link. A path uses the internal link when it goes across a node using an input port P and an output port P' if one of these ports belongs to one card and the other one belongs to the other card.

For instance, configuration A (TABLE 1) assigns the three ports of one card to the links (X^+, Y^+, Z^+), and the other card gets obviously the links (X^-, Y^-, Z^-), as it is shown in Fig. 1. In this configuration a path uses the two cards of a node for passing through it if the pair of ports used by the path is one of the followings (it should be noted that the first port is the input port and the second one is the output port):

$$(X^+, X^-)\ (X^+, Y^-)\ (X^+, Z^-)\ (Y^+, Y^-)\ (Y^+, Z^-)\ (Z^+, Z^-)$$

$$(X^-, X^+)\ (X^-, Y^+)\ (X^-, Z^+)\ (Y^-, Y^+)\ (Y^-, Z^+)\ (Z^-, Z^+)$$

Other combinations, which are allowed by the routing algorithm, involve passing through the node using an input port and an output port, both belonging to the same card. These configurations will not be considered because they do not produce communication overhead for the purpose of this study. Such pairs of ports for the configuration A are:

$$(X^+, Y^+)\ (X^+, Z^+)\ (Y^+, Z^+)\ (X^-, Y^-)\ (X^-, Z^-)\ (Y^-, Z^-)$$

In the following, we indicate for each configuration how to obtain the number of paths that pass through the node $\langle x, y, z\rangle$ by using the internal link that connects the two cards of the node. The final result for each configuration is shown avoiding the specific calculations.

Table 5: Number of paths that pass through the node $\langle x, y, z\rangle$.

$$R_{X^-\to Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\[2ex] \dfrac{(k-2)k^2}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^-\to Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\[2ex] \dfrac{k^3}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^+\to Y^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\[2ex] \dfrac{(k-2)^2}{4}k & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^+\to Y^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\[2ex] \dfrac{(k-2)k^2}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^-\to Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{if } k \text{ is odd} \\[2ex] \dfrac{(k-2)k}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^-\to Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{if } k \text{ is odd} \\[2ex] \dfrac{k^2}{4} & \text{if } k \text{ is even} \end{cases}$$

Table 6: (Cont.) Number of paths that pass through the node $\langle x, y, z\rangle$.

$$R_{X^+\rightarrow Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{if } k \text{ is odd} \\ \dfrac{(k-2)^2}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{X^+\rightarrow Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4} & \text{if } k \text{ is odd} \\ \dfrac{(k-2)k}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{Y^-\rightarrow Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\ \dfrac{(k-2)k^2}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{Y^-\rightarrow Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\ \dfrac{k^3}{4} & \text{if } k \text{ is even} \end{cases}$$

$$R_{Y^+\rightarrow Z^-}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\ \dfrac{(k-2)^2}{4}k & \text{if } k \text{ is even} \end{cases}$$

$$R_{Y^+\rightarrow Z^+}(\langle x,y,z\rangle) = \begin{cases} \dfrac{(k-1)^2}{4}k & \text{if } k \text{ is odd} \\ \dfrac{(k-2)k^2}{4} & \text{if } k \text{ is even} \end{cases}$$

- Case A: $\{X^+, Y^+, Z^+\} \mid \{X^-, Y^-, Z^-\}$

$$\begin{aligned}
R_A(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(3k^4 - 8k^3 + 6k^2 + 4k + 4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case B: $\{X^+, Y^+, Z^-\} \mid \{X^-, Y^-, Z^+\}$

$$\begin{aligned}
R_B(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + R_{X^+\to Z^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(3k^4 - 8k^3 + 6k^2) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case C: $\{X^+, Y^+, Y^-\} \mid \{X^-, Z^+, Z^-\}$

$$\begin{aligned}
R_C(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{X^-\to Y^-}(\langle x,y,z\rangle)) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Z^-}(\langle x,y,z\rangle) + R_{X^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{Y^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 + 2k^3 - 4k^2 - 2k + 4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case D: $\{X^+, Y^+, X^-\} \mid \{Y^-, Z^+, Z^-\}$

$$\begin{aligned}
R_D(\langle x,y,z\rangle) &= R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^-}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 - k^2 - 4k + 4) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 - 4k^2 + 4)/4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case E: $\{X^+, Y^-, Z^+\} \mid \{X^-, Y^+, Z^-\}$

$$\begin{aligned}
R_E(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Y^+}(\langle x,y,z\rangle) + R_{X^-\to Y^-}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^+\to Z^+}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(3k^4 - 8k^3 + 6k^2 - 4k + 4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case F: $\{X^+, Y^-, Z^-\} \mid \{X^-, Y^+, Z^+\}$

$$\begin{aligned}
R_F(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{Y\leftrightarrow Y}(\langle x,y,z\rangle) + R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Y^+}(\langle x,y,z\rangle) + R_{X^-\to Y^-}(\langle x,y,z\rangle) + R_{X^+\to Z^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(3k^4 - 8k^3 + 6k^2) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case G: $\{X^+, Y^-, X^-\} \mid \{Y^+, Z^+, Z^-\}$

$$\begin{aligned}
R_G(\langle x,y,z\rangle) &= R_{YY\leftrightarrow YY}(\langle x,y,z\rangle) + R_{X^+\to Y^+}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{X^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 - k^2 - 4k + 4) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 + 4k^2 - 8k + 4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case H: $\{X^+, Z^+, Z^-\} \mid \{X^-, Y^+, Y^-\}$

$$\begin{aligned}
R_H(\langle x,y,z\rangle) &= R_{X\leftrightarrow X}(\langle x,y,z\rangle) + R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^+\to Y^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{Y^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 + 2k^3 - 8k^2 + 6k) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case I: $\{X^+, Z^+, X^-\} \mid \{Y^+, Y^-, Z^-\}$

$$\begin{aligned}
R_I(\langle x,y,z\rangle) &= R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^+\to Y^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + R_{X^+\to Z^-}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^-}(\langle x,y,z\rangle) + R_{Y^+\to Z^+}(\langle x,y,z\rangle) + R_{Y^-\to Z^+}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 + 2k^3 - 4k^2 - 2k + 4) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

- Case J: $\{X^+, Z^-, X^-\} \mid \{Y^+, Y^-, Z^+\}$

$$\begin{aligned}
R_J(\langle x,y,z\rangle) &= R_{Z\leftrightarrow Z}(\langle x,y,z\rangle) + R_{X^+\to Y^-}(\langle x,y,z\rangle) + R_{X^+\to Y^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Y^-}(\langle x,y,z\rangle) + R_{X^-\to Y^+}(\langle x,y,z\rangle) + R_{X^+\to Z^+}(\langle x,y,z\rangle) + \\
&+ R_{X^-\to Z^+}(\langle x,y,z\rangle) + R_{Y^+\to Z^-}(\langle x,y,z\rangle) + R_{Y^-\to Z^-}(\langle x,y,z\rangle) = \\
&= \cdots = \begin{cases} \frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2) & \text{if } k \text{ is odd} \\ \frac{1}{4}(k^4 + 2k^3 - 8k^2 + 6k) & \text{if } k \text{ is even} \end{cases}
\end{aligned}$$

Tables 7 and 8 summarize the previous results when k is odd and even, respectively.

Table 7: Number of paths passing through a node using its two cards (k is odd).

Case	Number of paths that pass through the two cards
A, B, E, F	$\frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2)$
C, H, I, J	$\frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2)$
D, G	$\frac{1}{4}(k^4 - k^2 - 4k + 4)$

Table 8: Number of paths passing through a node using its two cards (k is even).

Case	Number of paths that pass through the two cards
A	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2 + 4k + 4)$
B, F	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2)$
C, I	$\frac{1}{4}(k^4 + 2k^3 - 4k^2 - 2k + 4)$
D	$\frac{1}{4}(k^4 - 4k^2 + 4)$
E	$\frac{1}{4}(3k^4 - 8k^3 + 6k^2 - 4k + 4)$
G	$\frac{1}{4}(k^4 + 4k^2 - 8k + 4)$
H, J	$\frac{1}{4}(k^4 + 2k^3 - 8k^2 + 6k)$

5.2.6 Analysis of the results

Figures 6 and 7 show a graphical representation of the expressions in tables 7 and 8. At a glance, we can see the differences between the port configurations and even guess which ones are the best configurations. However, it is recommended to perform a formal analysis to obtain fully accurate conclusions.

In the following we discuss this analysis for both cases: k is odd and k is even.

k is odd

If we observe Table 7 and Figure 6, it is possible to deduce that when k is odd the configurations D and G are the optimal configurations. To check it, we compare the expression of the configurations D and G with the expressions of the remaining configurations. We consider separately the case $k = 3$ because of its obviousness.

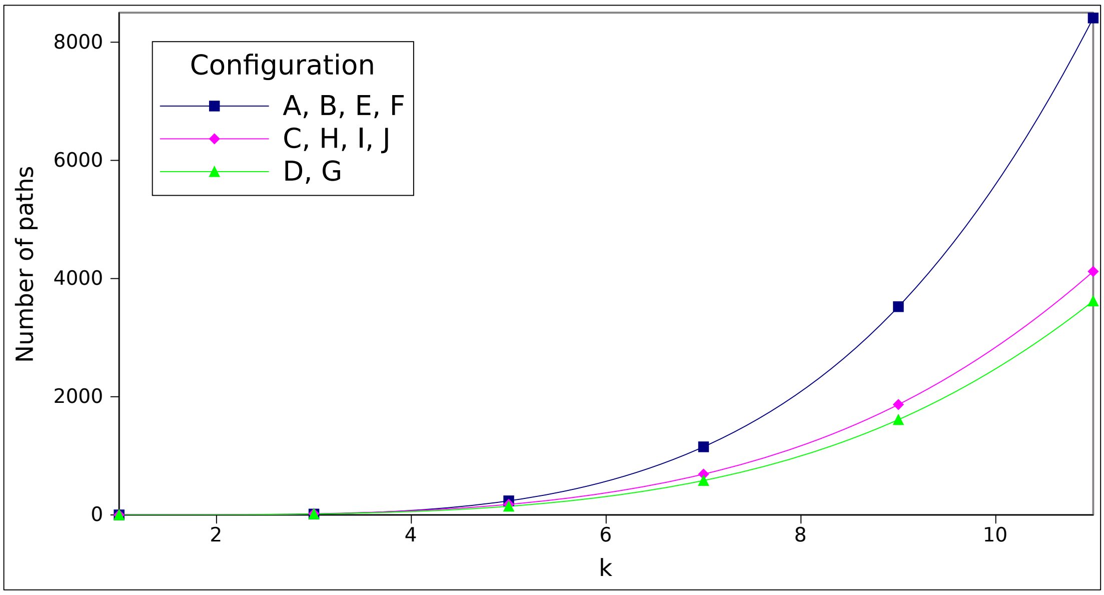


Figure 6: Number of paths that pass through a node through its two cards, for a few odd values of k.

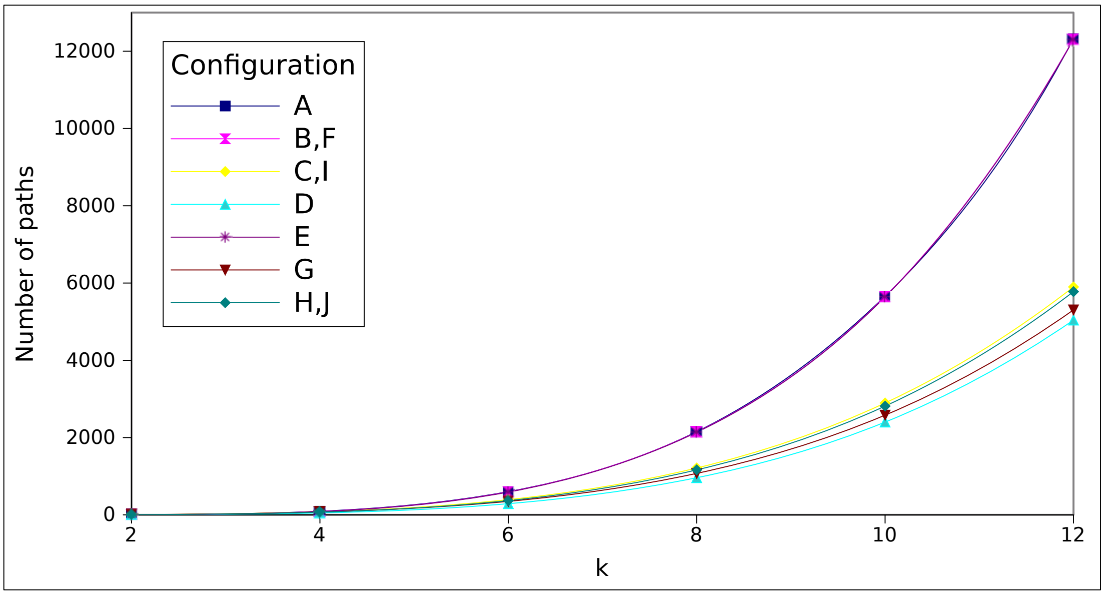


Figure 7: Number of paths that pass through a node through its two cards, for a few even values of k.

$\boxed{k = 3}$

If $k = 3$, the optimal configurations are A, B, E, and F. A simple substitution for each case is enough to check it.

$\boxed{k \geq 5}$

- $R_{ABEF}(\langle x, y, z\rangle) > R_{GD}(\langle x, y, z\rangle)$

$$\begin{aligned}
R_{ABEF}(\langle x, y, z\rangle) &> R_{GD}(\langle x, y, z\rangle) \\
R_{ABEF}(\langle x, y, z\rangle) - R_{GD}(\langle x, y, z\rangle) &> 0 \\
\frac{1}{4}(3k^4 - 8k^3 + 3k^2 + 2) - \frac{1}{4}(k^4 - k^2 - 4k + 4) &> 0 \\
\frac{1}{4}(2k^4 + 8k^3 - 4k^2 - 4k + 2) &> 0 \\
2(k-1)(k^3 - 3k^2 - k + 1) &> 0 \\
2(k-1)(k - r_1)(k - r_2)(k - r_3) &> 0
\end{aligned}$$

where r_1, r_2 and r_3 are constants and smaller than 4.

It should be noted that the inequality is true for $k \geq 5$, therefore $R_{ABEF}(\langle x, y, z\rangle) > R_{GD}(\langle x, y, z\rangle)$ is satisfied.

- $R_{CHIJ}(\langle x, y, z\rangle) > R_{GD}(\langle x, y, z\rangle)$

$$\begin{aligned}
R_{CHIJ}(\langle x, y, z\rangle) &> R_{GD}(\langle x, y, z\rangle) \\
R_{CHIJ}(\langle x, y, z\rangle) - R_{GD}(\langle x, y, z\rangle) &> 0 \\
\frac{1}{4}(k^4 + 2k^3 - 7k^2 + 2k + 2) - \frac{1}{4}(k^4 - k^2 - 4k + 4) &> 0 \\
\frac{1}{4}(2k^3 - 6k^2 + 6k - 2) &> 0 \\
2(k-1)^3 &> 0
\end{aligned}$$

which is true because $k \geq 5$, and therefore $R_{CHIJ}(\langle x, y, z\rangle) > R_{GD}(\langle x, y, z\rangle)$ is also true for $k \geq 5$.

k is even

Similarly, from Figure 7 we suspect that the optimal configuration in this case is D, and we will also check it by comparing the mathematical expressions.

- $R_A(\langle x,y,z\rangle) > R_D(\langle x,y,z\rangle)$

$$\begin{aligned}
R_A(\langle x,y,z\rangle) &> R_D(\langle x,y,z\rangle)\\
R_A(\langle x,y,z\rangle) - R_D(\langle x,y,z\rangle) &> 0\\
\frac{1}{4}(3k^4-8k^3+6k^2+4k+4) - \frac{1}{4}(k^4-4k^2+4) &> 0\\
\frac{1}{4}(2k^4-8k^3+10k^2+4k) &> 0\\
2k(k^3-4k^2+5k+2) &> 0
\end{aligned}$$

  The polynomial k^3-4k^2+5k+2 has only one root among the real numbers, $k=-0.315$. As $k>-0.315$ then it takes positive values. If $k \geq 2$, then the inequality is true. So, $R_A(\langle x,y,z\rangle) > R_D(\langle x,y,z\rangle)$ is true for $k \geq 2$.

- $R_{BF}(\langle x,y,z\rangle) > R_D(\langle x,y,z\rangle)$.

$$\begin{aligned}
R_{BF}(\langle x,y,z\rangle) &> R_D(\langle x,y,z\rangle)\\
R_{BF}(\langle x,y,z\rangle) - R_D(\langle x,y,z\rangle) &> 0\\
\frac{1}{4}(3k^4-8k^3+6k^2) - \frac{1}{4}(k^4-4k^2+4) &> 0\\
\frac{1}{4}(2k^4-8k^3+10k^2-4) &> 0\\
2(k-1)(k^3-3k^2+2k+2) &> 0
\end{aligned}$$

  The polynomial k^3-3k^2+2k+2 has only one root among the real numbers, $k=-0.521$. As $k>-0.521$ then it takes positive values. Also $(k-1)$ takes positive values for $k \geq 2$. Therefore, $R_{BF}(\langle x,y,z\rangle) > R_D(\langle x,y,z\rangle)$ is true for $k \geq 2$.

- $R_{CI}(\langle x,y,z\rangle) > R_D(\langle x,y,z\rangle)$.

$$\begin{aligned}
R_{CI}(\langle x,y,z\rangle) &> R_D(\langle x,y,z\rangle)\\
R_{CI}(\langle x,y,z\rangle) - R_D(\langle x,y,z\rangle) &> 0\\
\frac{1}{4}(k^4+2k^3-4k^2-2k+4) - \frac{1}{4}(k^4-4k^2+4) &> 0\\
\frac{1}{4}(2k^3-2k) &> 0\\
2k(k-1)(k+1) &> 0
\end{aligned}$$

  is true because $k \geq 2$. Therefore, $R_{CI}(\langle x,y,z\rangle) \geq R_D(\langle x,y,z\rangle)$ is true for $k \geq 2$.

- $R_E(\langle x, y, z\rangle) \geq R_D(\langle x, y, z\rangle)$.

$$\begin{aligned}
R_E(\langle x, y, z\rangle) &\geq R_D(\langle x, y, z\rangle) \\
R_E(\langle x, y, z\rangle) - R_D(\langle x, y, z\rangle) &\geq 0 \\
\frac{1}{4}(3k^4 - 8k^3 + 6k^2 - 4k + 4) - \frac{1}{4}(k^4 - 4k^2 + 4) &\geq 0 \\
\frac{1}{4}(2k^4 - 8k^3 + 10k^2 - 4k) &\geq 0 \\
2k(k-2)(k-1)^2 &\geq 0
\end{aligned}$$

  If $k = 2$ then $2k(k-2)(k-1)^2$ is equal to zero, so configuration E is equivalent to the D. If $k > 2$ then the roots of the polynomial are positive, so the result of the polynomial is always greater than zero. Therefore, $R_E(\langle x, y, z\rangle) \geq R_D(\langle x, y, z\rangle)$ is true for $k \geq 2$.

- $R_G(\langle x, y, z\rangle) > R_D(\langle x, y, z\rangle)$ for $k \geq 2$.

$$\begin{aligned}
R_G(\langle x, y, z\rangle) &> R_D(\langle x, y, z\rangle) \\
R_G(\langle x, y, z\rangle) - R_D(\langle x, y, z\rangle) &> 0 \\
\frac{1}{4}(k^4 + 4k^2 - 8k + 4) - \frac{1}{4}(k^4 - 4k^2 + 4) &> 0 \\
\frac{1}{4}(8k^2 - 8k) &> 0 \\
8k(k-1) &> 0
\end{aligned}$$

  is true because $k \geq 2$. Therefore, $R_G(\langle x, y, z\rangle) > R_D(\langle x, y, z\rangle)$ is true for $k \geq 2$.

- $R_{HJ}(\langle x, y, z\rangle) > R_D(\langle x, y, z\rangle)$ for $k \geq 2$.

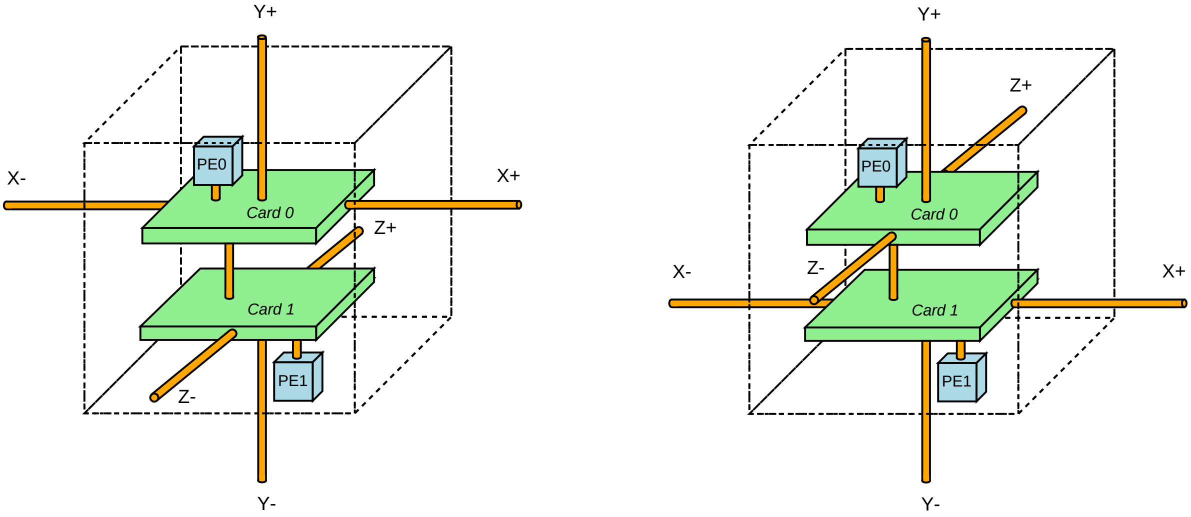
$$\begin{aligned}
R_{HJ}(\langle x, y, z\rangle) &> R_D(\langle x, y, z\rangle) \\
R_{HJ}(\langle x, y, z\rangle) - R_D(\langle x, y, z\rangle) &> 0 \\
\frac{1}{4}(k^4 + 2k^3 - 8k^2 + 6k) - \frac{1}{4}(k^4 - 4k^2 + 4) &> 0 \\
\frac{1}{4}(2k^3 - 4k^2 + 6k - 4) &> 0 \\
2(k-1)(k^2 - k + 2) &> 0
\end{aligned}$$

  For $k \geq 2$, the inequality has two positive solutions among the real numbers. The polynomial $k^2 - k + 2$ has no roots among the real numbers, however it takes always positive values for all $\mathbb{R}$. Therefore, $R_{HJ}(\langle x, y, z\rangle) > R_D(\langle x, y, z\rangle)$ is true for $k \geq 2$.

Thus, the configuration D is the best one when k is even. It imposes the minimum number of paths for the internal link that interconnects the two cards. Note that for $k = 2$, the configurations E and D are equivalent.

It should be reminded that we took the positive direction when the distance between the source and destination nodes is the same by the two directions. If such a initial hypothesis changed, then this study would remain exactly the same, but the optimal configuration would be G instead of D.

Finally, in summary, Fig. 8 shows the two configurations that offer the highest performance level, considering both even k and odd k values.



(a) (X^+, Y^+, X^-)–(Y^-, Z^+, Z^-) for odd and even k values.

(b) (X^+, Y^-, X^-)–(Y^+, Z^+, Z^-) for odd k values.

Figure 8: The optimal configurations.

6 Routing in 3DT torus

The routing algorithm is the mechanism that determines the path selected by a message to reach its destination. In many cases, some situations can difficult the routing, like *deadlock*, *livelock* or *starvation*. Specifically, the *deadlock* is an inherent problem in k-ary n-cubes. This problem is even more important in a 3DT torus because the internal link is shared by all dimensions of the 3DT torus. The techniques used in the common k-ary n-cubes need to be modify for avoiding the *deadlock* in a 3DT torus.

In this section a study of *deadlock* in 3DT torus is included. In first place, we present the *DOR* routing algorithm (**D**imension **O**rder **R**outing) adapted for 3DT topology in Section 6.1. Then, we study the cycles that appear in this topology (Section 6.2) and finally, we explain how remove these cycles and *deadlock* in Section 6.3.

6.1 *DOR* routing algorithm adapted for 3DT torus topology

DOR routing is commonly used in a k-ary n-cube because it is a very simple routing algorithm. Basically, a message is routed by the n dimensions following an ascending (or descending) strict order. If a node is identified by a n-tuple $\langle x_{n-1}, x_{n-1}, \ldots, x_1, x_0 \rangle$, a message needing to use all the dimensions is first routed through dimension 0, after that it is routed through dimension 1, and so on until reaching the dimension $n - 1$.

For a 3DT torus topology, each PE needs an identifier composed of three digits, one digit for each dimension (X, Y and Z), and another digit to identify the PE inside the node. First, a message crosses the three dimensions as needed. In this case, if the message has reached the destination node, it checks if the message is destined to the current PE or the neighbor PE, routing the message to the NIC or the internal link, respectively. Finally, we check if the output link belongs to the current card. Otherwise, the selected output port will be the internal link.

In Algorithm 1 we can see the pseudo-code of the DOR routing algorithm adapted for the 3DT torus. The function *ringDirection()* (Algorithm 2) is used for DOR routing to determine the output direction in any ring.

6.2 Analysing the of cycles in 3DT torus topologies

Most of the deterministic routing algorithms base their deadlock-freedom on the channel dependency graph. A routing algorithm is deadlock free if there are no cycles in its channel dependency graph [DS87]. However, when the number of nodes in the network increases, the channel dependency graph size increases in proportion of the number of channels, making difficult the elimination of cycles. Therefore, we have studied the reason why these cycles appear in the 3DT torus topology to act consequently and remove *deadlock* in the topology.

6.2.1 Types of traffic in the internal link

Unfortunately, in 3DT torus topologies new cycles appear on the network that are not present in the traditional 3D torus topology. This is due to the use of the internal link, which can be used by a message regardless of the dimension where it is traveling. In some cases, the message uses the internal link as a part of a ring. In other cases, the internal link is used for making a change between dimensions. Figure 9 shows the ring of all the dimensions using the configuration D[4]. In this case, we can see how the internal link is part of the Y-dimension ring.

[4] From now on, in the examples configuration D will be used because it is optimal if $k \geq 4$ and so the 3DT torus topology gets advantage over the 2D torus.

Algorithm 1 *DOR* routing algorithm for a 3DT torus.

```
Require: current node $\langle x_c, y_c, z_c, ep_c \rangle$, destination node $\langle x_d, y_d, z_d, ep_d \rangle$
Return: output port $p$
if $x_d \neq x_c$ then
    $p = ringDirection(x_c, x_d)$
else if $y_d \neq y_c$ then
    $p = ringDirection(y_c, y_d)$
else if $z_d \neq z_c$ then
    $p = ringDirection(z_c, z_d)$
else if $ep_d \neq ep_c$ then
    $p = internal\_link$
else
    $p = NIC$
end if
if $p \in LINKS(ep_c)$ then
    return $p$
else
    return $internal\_link$
end if
```

Algorithm 2 *ringDirection()* function.

```
Require: current digit $d\_cur$, destination digit $d\_des$
Return: output port $(D^+,\ D^-)$ //The letter D can be any dimension (X, Y or Z)
$aux = (d\_des - d\_cur) mod\ k$
if $aux > k/2$ then
    $aux = aux - k$
end if
if $aux \geq 0$ then
    return $D^+$
else
    return $D^-$
end if
```

Specifically, if we analyses the use of the internal link, we can distinguish 3 cases (Figure 10), depending on the destination of the message after using the internal link:

1.- The message uses the internal link to be injected in a d-dimension and the internal link does not belong to the d-dimension ring. In Figure 10 we can see how a message that arrives from a link of the X-dimension or the Y^+ link, must use the internal link to be injected in the Z-dimension (red dotted line).

2.- The message uses the internal link as a part of the d-dimension ring. The message can cross the d-dimension before using the internal channel or can be injected from another dimension. In Figure 10 we can see a message that arrives from the X-dimension, as well as a message that crosses the Y-dimension, must use the internal link to exit the node from the Y^- port (yellow dotted line).

3.- The destination of the message is the PE connected to the other card in the node. In this case, the message can arrive at the node from any link of the current card (blue dotted line).

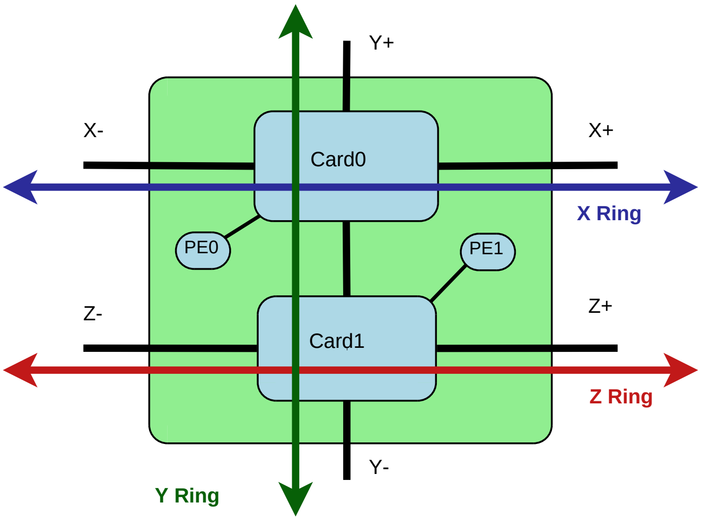


Figure 9: Rings corresponding to each dimension in any node in the network.

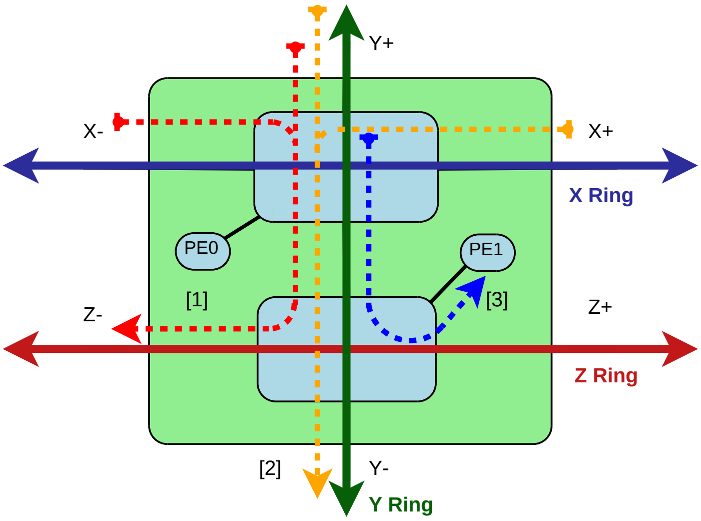


Figure 10: Possible uses of the internal link.

6.2.2 Types of cycles in 3DT torus topologies

From the previous study of possible uses of the internal link, we have identified two possible types of cycles in which the internal channel is involved:

A.- Several messages use the internal link as part of the ring of d-dimension. Without mechanisms to avoid *deadlock*, it can appear in any ring of a k-ary n-cube. This type of cycle is caused by the traffic of type 2. Example 6.1 shows in more detail a situation in which a *deadlock* appears due to this reason.

B.- Several messages use along their paths several internal links to be injected in a new dimension and also to reach the destination PE. This type of cycle appears due to the type of traffic 1 and 3. Example 6.2 shows in more detail a situation in which a *deadlock* appears due to this reason.

Example 6.1 *Given a 3DT torus network, with two nodes in Y-dimension and the nodes using the configuration D, and considering that:*

- *$PE0$ in the $\langle x, 0, z\rangle$ node sends a message to $PE0$ of the $\langle x, 1, z\rangle$ node, and vice versa.*
- *$PE1$ in the $\langle x, 0, z\rangle$ node sends a message to $PE1$ of the $\langle x, 1, z\rangle$ node, and vice versa.*

there exist cycles and deadlock can appear in the network. In Figure 11 we can see graphically this situation.

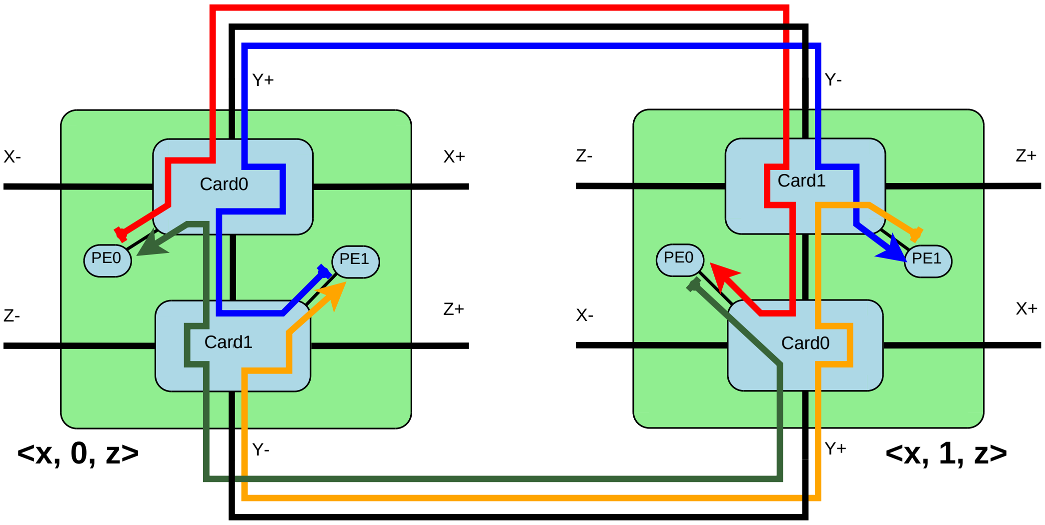


Figure 11: Possible deadlock due to the use of the internal link as a part of the Y-dimension ring.

Example 6.2 *Given a 3DT torus network of any size and nodes using configuration D, and considering that:*

- *$PE1$ in the $\langle x, y, z \rangle$ node sends a message to $PE0$ in the $\langle x+1, y, z+1 \rangle$ node.*
- *$PE1$ in the $\langle x+1, y, z+1 \rangle$ node sends a message to $PE0$ in the $\langle x, y, z \rangle$ node.*

there exist cycles and deadlock can appear in the network. In Figure 12 we can see graphically this situation.

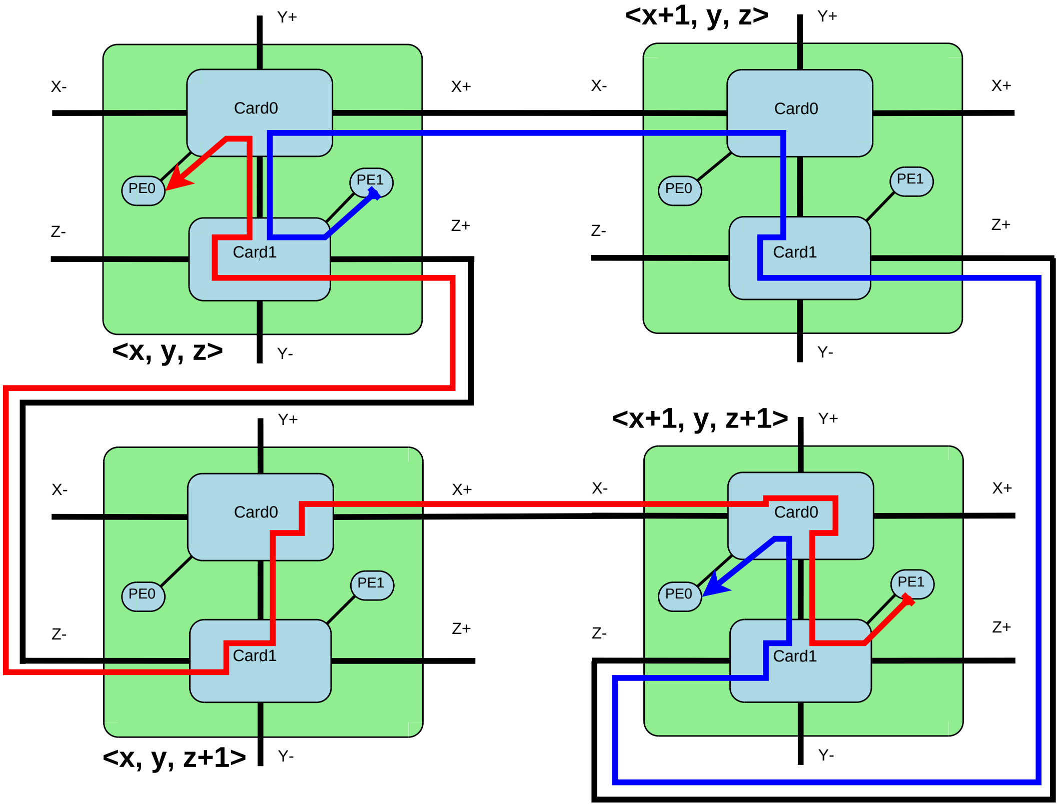


Figure 12: Possible deadlock due to the use of the internal link to change between dimensions and to reach the destination PE.

6.3 Deadlock-avoidance in 3DT torus topologies

Once we know how the cycles are produced, now we proceed to their elimination. Two of the commonly used techniques to avoid *deadlock* in k-ary n-cubes are the use of virtual channels [DS87] and the bubble flow control mechanism [CBGV97]. Considering only the external links of the nodes we can use these techniques in the same way that are used in traditional tori, but it is necessary to modify them for avoiding deadlock. Sections 6.3.1 and 6.3.2 present the modifications realized to the DOR routing algorithm to avoid deadlock, using virtual channels and the bubble flow control mechanism, respectively.

6.3.1 Virtual Channels

As mentioned above, new cycles appear in the network because all the dimensions use the internal link. To break these cycles, it is necessary to multiplex the internal link using virtual channels and treat the data stream of each dimension separately. It is also necessary to route messages destined to the neighbor PE using an exclusive virtual channel, to avoid the cycles of type B. Thus, the nodes require the following virtual channels:

- One virtual channel for the messages using an output port and internal link belonging to different dimensions (traffic type 1). In the modified algorithm for the configuration D we choose the virtual channel 0 for this traffic type.
- Two virtual channels for the messages using an output port and internal link belonging to the same dimension (traffic type 2). To choose which virtual channel is used for routing the message, we use the same criteria as in the external channel. If the current dimension digit of the destination node is greater than the digit of the current node, the first virtual channel is chosen. In other case, the second virtual channel is selected. In the modified algorithm for the configuration D we choose the virtual channels 1 and 2 for this traffic type.
- One virtual channel for the messages whose destination is the neighbor PE. The virtual channel 3 has been chosen for this traffic type in the modified algorithm.

Using these virtual channels, there are no cycles in the channel dependency graph and thus the network is deadlock-free. The total number of virtual channels depends on the node configuration. If the links of all dimensions are separated, it is necessary to use $3\times 2 + 1 = 7$ virtual channels (configurations A, B, E and F), while the rest of configurations only need $1 + 1\times 2 + 1 = 4$ virtual channels (configurations C, D, G, H, I and J).

Algorithm 3 and 4 show the modifications realized in the routing algorithm and *ringDirection()* function, respectively, whereas Figure 13 (a) shows graphically the use of virtual channels depending on the type of traffic, always considering the configuration D.

6.3.2 The Bubble flow control mechanism

In this case, it is also necessary to multiplex the traffic in the internal link, although fewer channels are required to ensure the deadlock freedom. Therefore, the internal link requires the following virtual channels:

- One virtual channel for the messages using an output port and the internal link belonging to different dimensions (traffic type 1). In this case, the access to the internal channel is not considered as a change of dimension, so to apply the bubble is not necessary. The change of dimension is realized when the message

Algorithm 3 Modifications in the routing algorithm to avoid deadlock using virtual channels and and configuration D.

```
Require: current node $\langle x_c, y_c, z_c, ep_c \rangle$, destination node $\langle x_d, y_d, z_d, ep_d \rangle$
Return: output port $p$, virtual channel $vc$
if $x_d \neq x_c$ then
    $p = ringDirection(x_c, x_d)$
else if $y_d \neq y_c$ then
    $p = ringDirection(y_c, y_d)$
else if $z_d \neq z_c$ then
    $p = ringDirection(z_c, z_d)$
else if $ep_d \neq ep_c$ then
    $p = internal\_link$
    $vc = 3$        // type 3.
else
    $p = NIC$
end if
if $p \notin LINKS(ep_c)$ then
    if $p \neq Y^+$ and $p \neq Y^-$ then
        $vc = 0$        // type 1.
    else if $vc = Up\_Links$ then
        $vc = 1$        // type 2.
    else
        $vc = 2$        // type 2.
    end if
    $p = internal\_link$
end if
```

Algorithm 4 Modifications in the *ringDirection()* function to avoid deadlock using virtual channels and and configuration D.

```
Require: current digit $d\_cur$, destination digit $d\_des$
Return: output port $p$, virtual channel $vc$
$aux = (d\_des - d\_cur) mod\ k$
if $aux > k/2$ then
    $aux = aux - k$
end if
if $aux \geq 0$ then
    $p = D^+$
else
    $p = D^-$
end if
if $d\_des > d\_cur$ then
    $vc = Up\_Links$
else
    $vc = Low\_Links$
end if
```

is injected into an external port from internal link. In the modified algorithm for the configuration D we choose the virtual channel 0 for this traffic type.

- One virtual channel for the message using an output port and internal link belong to the same dimension (traffic type 2). In this case, the internal link is considered as a part of the ring dimension, so it is necessary to apply the bubble if the message is injected from another dimension. In the modified algorithm for configuration D we choose the virtual channel 1 for this traffic type.

- One virtual channel for the messages whose destination is the neighbor PE. The bubble is not necessary because the next destination of the message is the NIC and the bubble only reduces the performance of the link. The virtual channel 2 has been chosen for this traffic type in the modified algorithm.

This mechanism ensures the deadlock freedom in the network. If the links of all dimensions are separated, it is necessary to use $3 \times 1 + 1 = 4$ virtual channels (configurations A, B, E and F), while the rest of configurations only need $1+1+1 = 3$ virtual channels (configurations$C$, D, G, H, I and J).

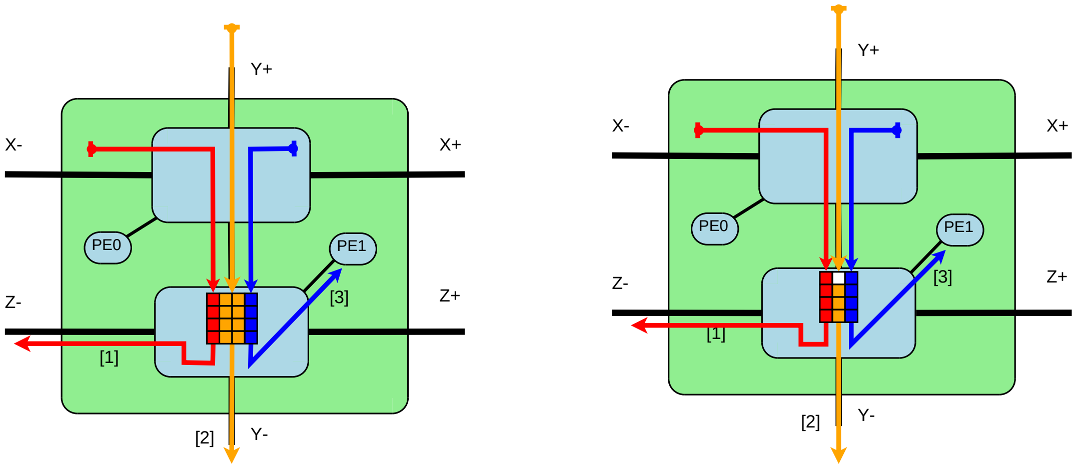
Algorithm 6 and 5 show the modifications realized in the routing algorithm and *ringDirection()* function, respectively, whereas Figure 13 (b) shows graphically the use of virtual channels depending on the type of traffic, always considering the configuration D.

Algorithm 5 Modifications in the *ringDirection()* function to avoid deadlock using the bubble flow control and the configuration D.

```
Require: current digit $d\_cur$, destination digit $d\_des$, input port $p_{in}$
Return: output port $p_{out}$, activating bubble $bub$
aux = $(d\_des - d\_cur) mod\ k$
if $aux > k/2$ then
    $aux = aux - k$
end if
if $aux \geq 0$ then
    $p_{out} = D^+$
else
    $p_{out} = D^-$
end if
if $p_{in} = D^+$ o $p_{in} = D^-$ then
    $bub =$ false
else if ($p_{out} = Y^+$ o $p_{out} = Y^-$) and $p_{in} = internal\_link$ then
    $bub =$ false
else
    $bub =$ true
end if
```

Algorithm 6 Modifications in the routing algorithm to avoid deadlock using the bubble flow control and the configuration D.

```
Require: current node $\langle x_c, y_c, z_c, ep_c \rangle$, destination node $\langle x_d, y_d, z_d, ep_d \rangle$
Return: output port $p$, virtual channel $vc$
if $x_d \neq x_c$ then
    $p = ringDirection(x_c, x_d)$
else if $y_d \neq y_c$ then
    $p = ringDirection(y_c, y_d)$
else if $z_d \neq z_c$ then
    $p = ringDirection(z_c, z_d)$
else if $ep_d \neq ep_c$ then
    $p = internal\_link$
    $vc = 2$          // type 3.
    $bub =$ false
else
    $p = NIC$
    $bub =$ false
end if
if $p \notin LINKS(ep_c)$ then
    if $p \neq Y^+$ and $p \neq Y^-$ then
        $vc = 0$          // type 1.
        $bub =$ false
    else
        $vc = 1$          // type 2.
    end if
    $p = internal\_link$
end if
```



(a) Virtual Channels

(b) Bubble flow control

Figure 13: Solutions to eliminate *deadlock* using the configuration D.

7 Performance evaluation of the topology

In this section we evaluate the 3DT torus topology. Two kinds of results are included: On the one hand, we compare all the node configurations, not only for corroborating the theoretical results of the study presented in previous sections, but mainly to evaluate the performance differences among all the port configurations. On the other hand, we also compare the performance of the 3DT torus network with the performance of the equivalent 2D torus network.

The evaluation of the topologies has been realized using a network simulator. In Section 7.1 we describe the features of the simulator used, while Section 7.2 includes the set of metrics used for performance evaluation. Finally, we show the experiments and the results obtained for the evaluation of different node configurations and the comparison between the 2D and the 3DT torus in Sections 7.3 and 7.4, respectively.

7.1 System Model

As mentioned above, the evaluation of the port configurations of the 3DT torus topology has been performed by simulation. To model the different topologies, we have used a 5-port switch (4 ports for interconnecting switches and one port for connecting the switch with its PE), whose features are the same, regardless of the modeled topology. Depending on the topology and solution chosen to eliminate the *deadlock*, the only differences between switches of different networks are the routing algorithm, the organization of the virtual channels in the physical channels and if the flow bubble mechanism is implemented or not.

Specifically, we have implemented an IQ (*Input Queued*) switch [KH98, MIM+97]. In this type of switches, there are only buffers in the input ports (Figure 14). These buffers are used if a packet cannot be routed to the corresponding output port.

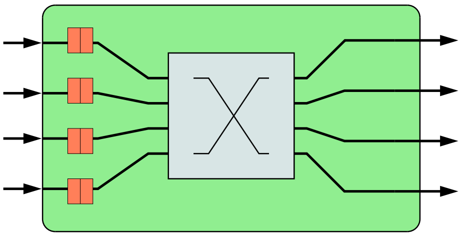


Figure 14: Scheme of an IQ switch.

The chosen switching technique for simulations is *virtual cut–through*, because it is commonly used in the supercomputer area. Since PEs are not integrated in the switch, the switch area destined to storage is less-restrictive than in other computation systems. Furthermore, this switching technique is less sensitive to the network diameter than other techniques, making it ideal for networks with a high number of nodes.

For the arbiter unit, we have modeled a two stages round-robin arbiter: in a first step, the arbiter chooses an input port with packets that can pass through the crossbar; then the arbiter chooses a virtual channel of this port (if it is necessary).

In the 2D torus, we have implemented the DOR routing algorithm, whereas in the 3DT torus we have used the modified DOR algorithm described in Section 6. In all the cases, the flow control is credit-based, but the use of the bubble flow control mechanism depends on the solution chosen to eliminate the deadlock.

In all the cases, the size of physical channels is the same. In all the experiments, the size of physical channels is 128 flits, and the size of the packets is 4 flits. The number and size of virtual channels depend on the topology and the solution used to remove the deadlock:

- **Bubble flow control**:
  - *2D torus topology*:
    - 1 link to the PE with a single virtual channel of 128 flits.
    - 4 interconnection links with a single virtual channel of 128 flits.
  - *3DT torus topology*:
    - 1 link to the PE with a single virtual channel of 128 flits.
    - 3 interconnection links with a single virtual channel of 128 flits.
    - 1 interconnection link (internal link) with 4 virtual channels[5] of 32 flits.
- **Virtual Channels**:
  - *2D torus topology*:
    - 1 link to the PE with 4 virtual channels of 32 flits.
    - 4 interconnection links with 4 virtual channels of 32 flits.
  - *3DT torus topology (configurations C, D, G, H, I and J)*:
    - 1 link to the PE with 4 virtual channels of 32 flits.
    - 3 interconnection links with 4 virtual channels[6] of 32 flits.
    - 1 interconnection link (internal link) with 4 virtual channels of 32 flits.
  - *3DT torus topology (configurations A, B, E and F)*:
    - 1 link to the PE with 4 virtual channels of 32 flits.
    - 3 interconnection links with 4 virtual channels of 32 flits.
    - 1 interconnection link (internal link) with 8 virtual channels[7] of 16 flits.

[5] Although it is only necessary to use 3 virtual channels, we use 4 virtual channels to make the implementation easier. For the traffic type 1 we have used two virtual channels instead of just one, choosing the destination virtual channel on the output port of the dimension in function of the output port.

[6] Although only two virtual channels are needed, four have been chosen to make the implementation easier.

[7] We have chosen eight virtual channels instead of seven to facilitate the implementation. The extra virtual channel is used for the traffic destined to the neighbor PE.

Finally, we have modeled an uniform traffic pattern. In a network with N processing elements, the destination of a message generated in a PE can be any of the remaining $N - 1$ PEs, with equal probability.

7.2 Metrics for the performance evaluation

For the performance evaluation, we have considered the following metrics:

- **Average throughput**. Measured in packets/cycle, it indicates the productivity of the network, i.e., the amount of information that the network can deliver per unit time.
- **Average network latency**. This value represents the average of the delays produced by the transmission of packets on the network measured in cycles. The latency of a message measures the number of cycles that elapse since the message is injected into a network switch from the NIC connected to the source PE until it is received by the NIC connected to the destination PE.
- **Average end-to-end latency**. The end-to-end latency measures the number of cycles that elapse since a message is generated by the NIC associated with the source PE until it is received at the NIC associated with the destination PE. This value represents the average end-to-end latency in the network.

7.3 Evaluation of different 3DT torus configurations

In this section we study, for a 3DT torus, the influence of the node configuration on the network performance. In first place, we describe the realized experiments, followed by the presentation of obtained results. Finally, we present a short analysis of these results.

7.3.1 Experiments

To evaluate the performance of different configurations, we have performed a set of tests, varying in each case the size of the network, the node configuration and the solution used to avoid *deadlock*. Each test consists of 30 experiments and the results presented in the next section are the average value of those 30 experiments. We take into account the next consideration to make the tests:

- We study the following topologies:
  - 3DT 4×4×2 torus, 64 PEs[8].

[8] Each node in the network has 2 PEs.

  - 3DT 4×4×4 torus, 128 *PEs*.
  - 3DT 5×5×5 torus, 250 *PEs*.
  - 3DT 6×6×6 torus, 432 *PEs*.

- Deadlock-avoidance mechanism:
  - Bubble flow control.
  - Virtual channel.

- We have tested the 10 node configurations studied (*A*-*J*).

7.3.2 Results

In this section we present graphically the results obtained in the experiments. Figure 15 shows, for a 3DT 4×4×2 torus, how productivity, network latency and end to end latency evolve when increasing the injection rate of the *PEs* in the network, for each configuration. The results of 128, 250 and 432 *PEs* are shown in Figures 16, 17 and 18, respectively.

7.3.3 Analysis of the results

Considering the presented results, the configuration D generally obtains the best performance. Only in the 64-*PEs* torus there are others configurations that behave similarly to the configuration D. Even so, as the size of the network is increased, the performance differences between the configuration D and the rest of configurations become more significant. These results are entirely consistent with the results of the theoretical study presented in Section 5.

For example, whereas in the 128-*PEs* torus with the bubble flow control the throughput of configuration D is 10% $\sim$ 23%[9] (depending on the configuration) higher than the rest of configurations, in the 250-*PEs* torus the throughput is 15% $\sim$ 33% higher. The same happens with the network and the end-to-end latency: these are reduced 5% $\sim$ 20% and 20% $\sim$ 40%, respectively, in 128-*PEs* torus, while the decrease in the 250-*PEs* torus is 12% $\sim$ 28% and 30% $\sim$ 50%.

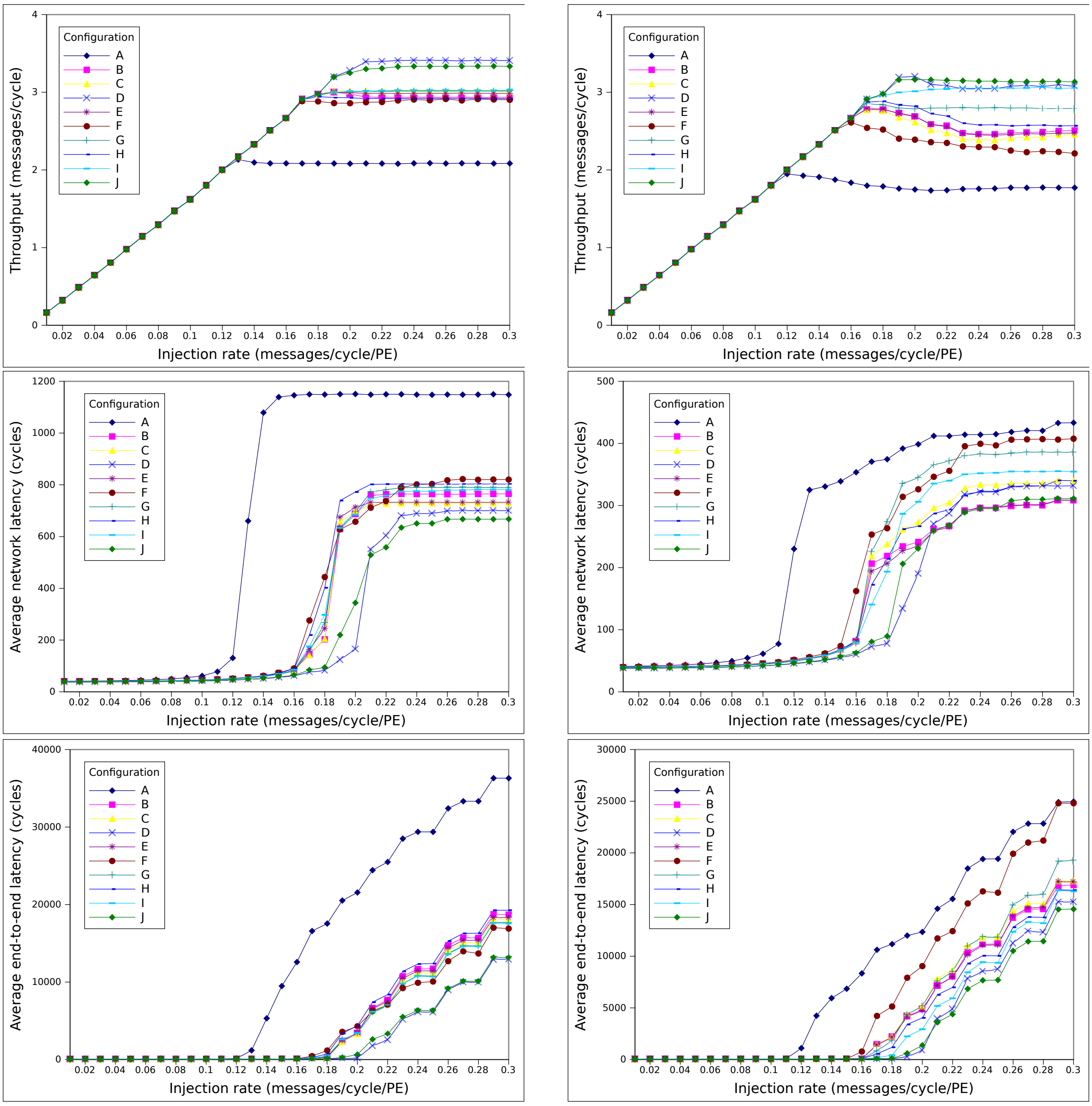
Note that in the 250-*PEs* torus ($k = 5$), the configuration G obtains similar performance that configuration D. Again, this is consistent with the results of the theoretical study, as both configurations are optimal if k is odd. We can see how the rest of configurations are grouped in two sets with similar performance, as expected.

In the 3DT torus using virtual channels to avoid *deadlock*, the throughput is reduced when network reaches the saturation point. This decrease is higher when the network size increases. This problem is due to network congestion and is typical in

[9] The expression $a\% \sim b\%$ means that the percentages range from $a\%$ to $b\%$. In this case, the throughput ranges 10% to 23%.

these architectures [Bol92, Izu06]. The performance degradation causes the throughput of 250-PE torus with node configuration D or G is the same that the throughput in torus using the configuration C, H, I and J. Even so, the network and end-to-end latencies are still 10% lower. In 432-PE torus, all the configurations are affected by the negative impact of congestion, but configuration D and G are the most resistant to performance degradation. This problem does not appear in the switches with the bubble flow mechanism implemented.

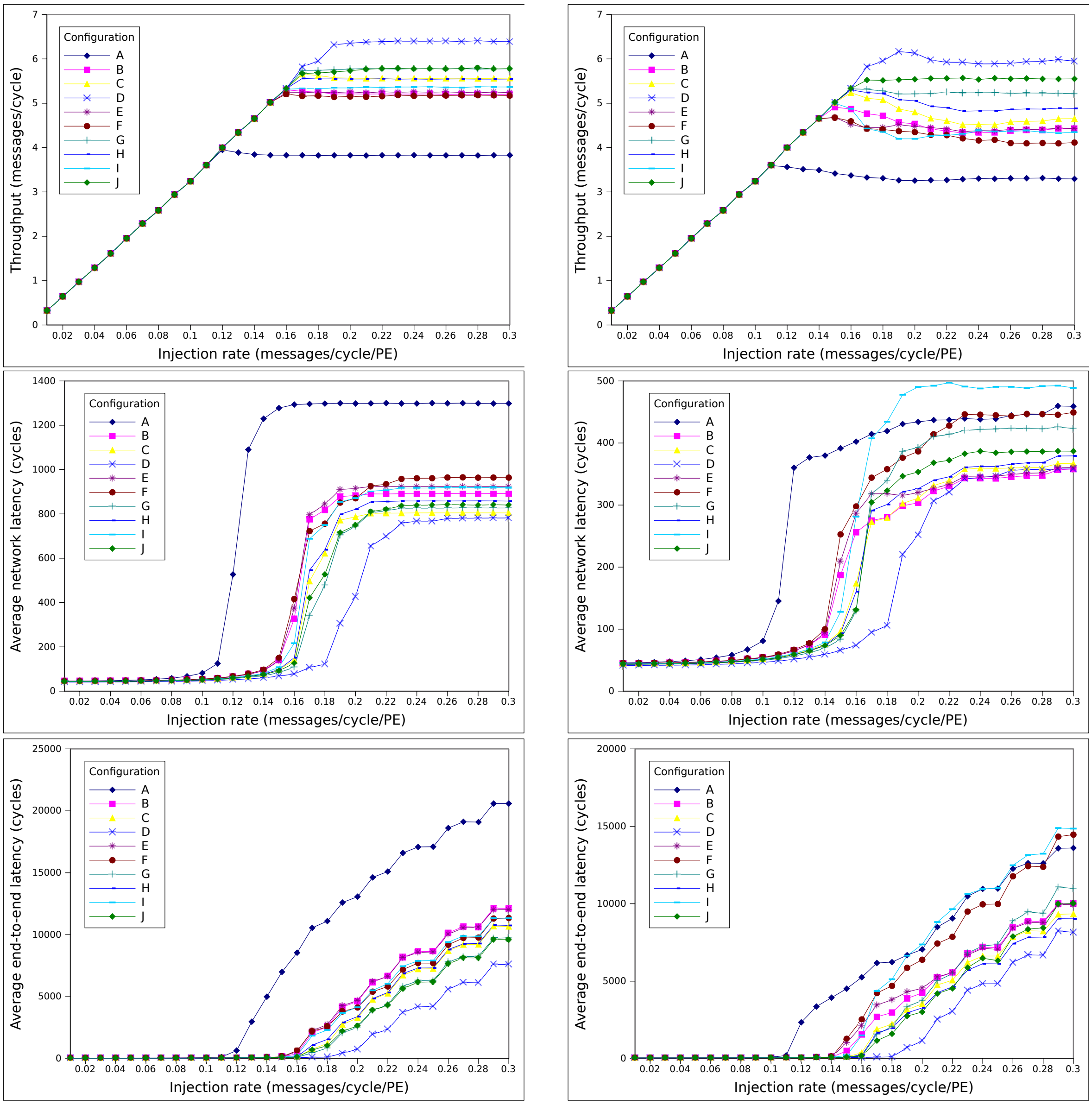
Taking into account the previous results, which coincide with the results of the theoretical study, for clarity reasons in the next tests of the 3DT torus topology we only use the node configuration D.



(a) Bubble Flow Control

(b) Virtual Channel

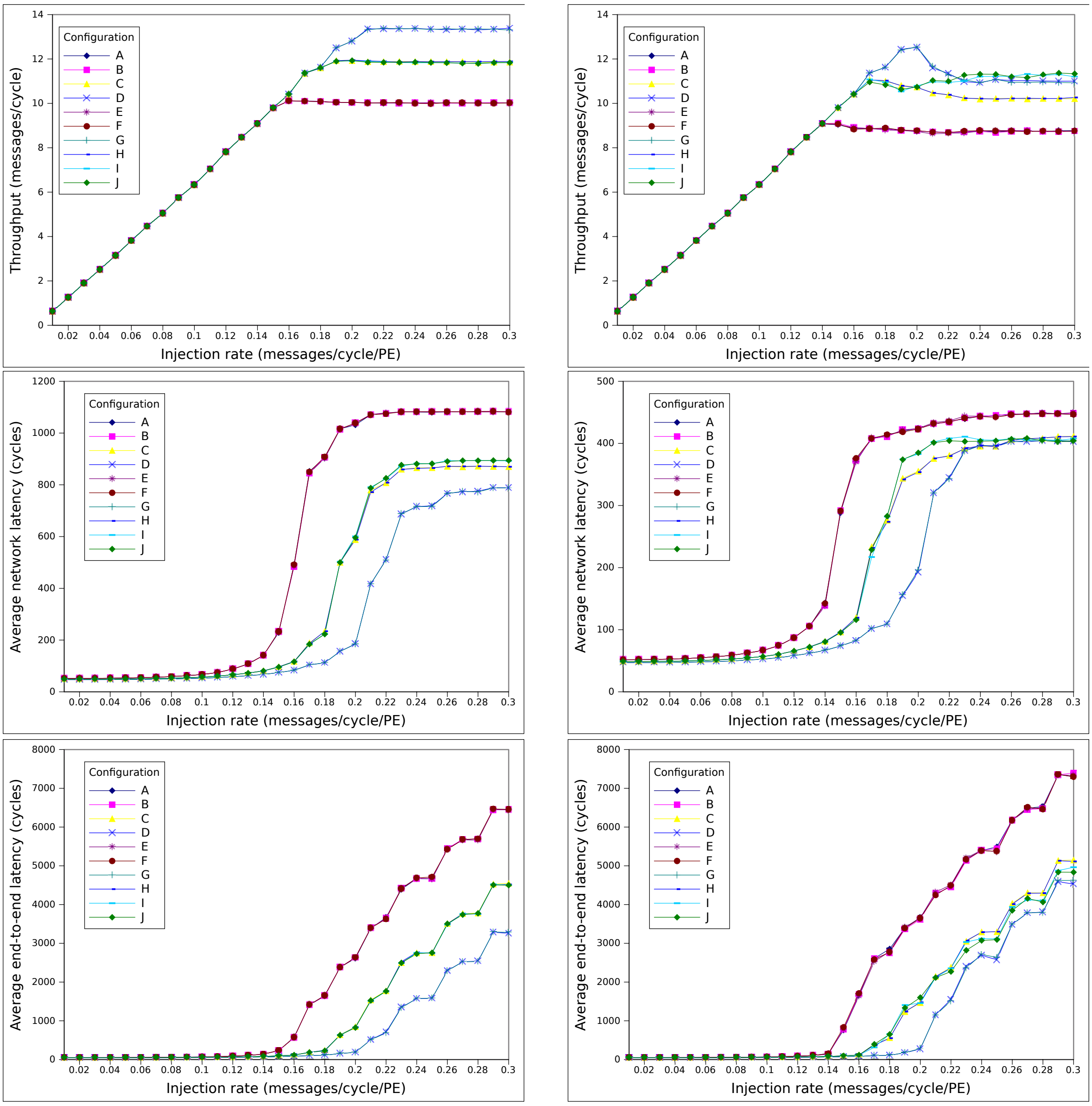
Figure 15: Network performance obtained for all the configurations of a 3DT 4×4×2 torus (64 PEs).



(a) Bubble Flow Control

(b) Virtual Channel

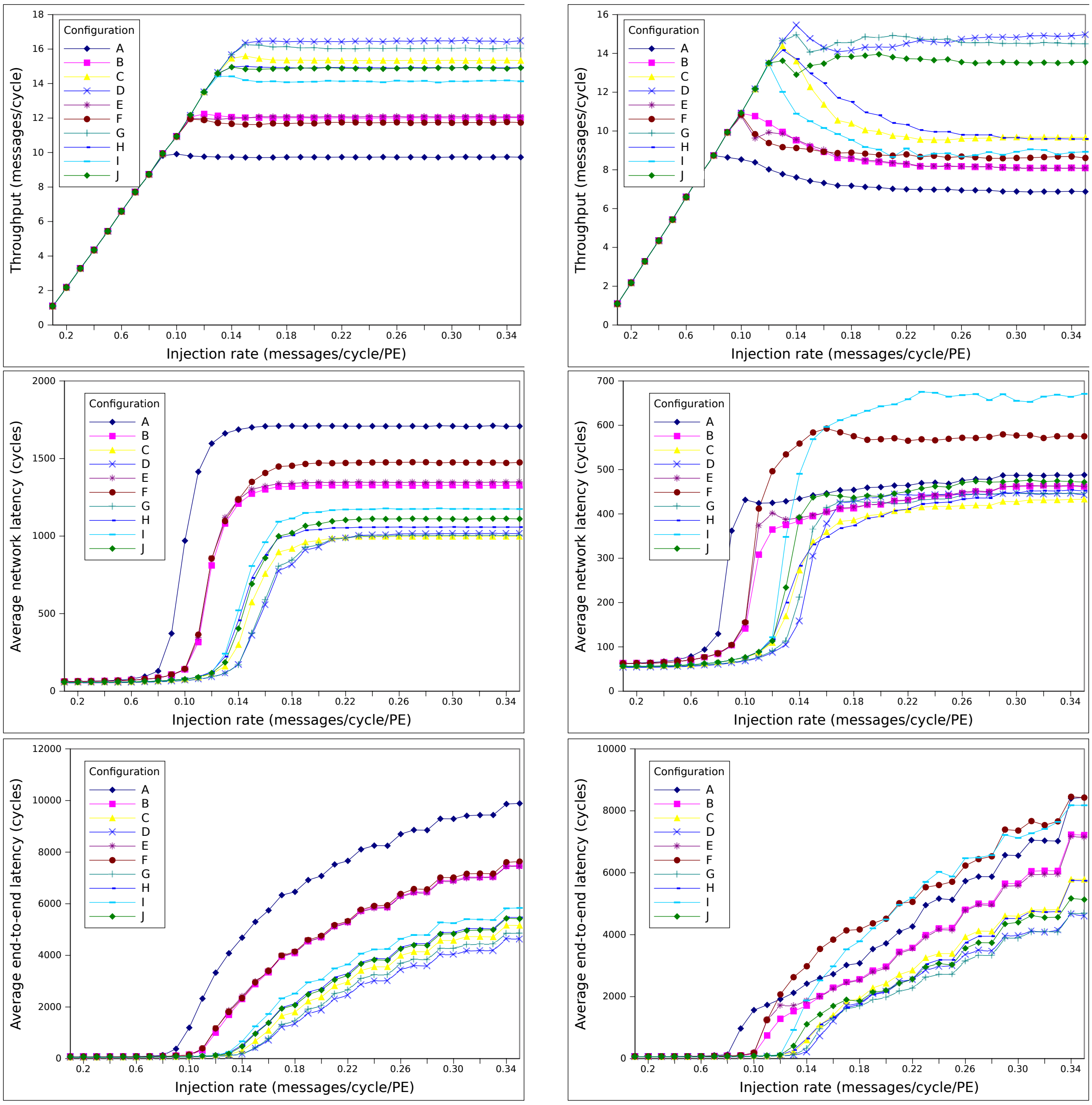
Figure 16: Network performance obtained for all the configurations of a 3DT 4×4×4 torus (128 *PEs*).



(a) Bubble Flow Control

(b) Virtual Channel

Figure 17: Network performance obtained for all the configurations of a 3DT 5×5×5 torus (250 PEs).



(a) Bubble Flow Control

(b) Virtual Channel

Figure 18: Network performance obtained for all the configurations of a 3DT 6×6×6 torus (432 PEs).

7.4 Comparison of the 3DT torus with the 2D torus

In this section we compare the performance of 2D and 3DT torus topologies, for networks with the same number of *PEs*. In first place, we describe the realized experiments and present the obtained results, including a short analysis of these results.

7.4.1 Experiments

Just as in the node configuration evaluation, we have realized a set of tests consisting 30 different experiments for each test. We have taken into account the next considerations to make the tests:

- The following topologies are compared:
  - 64 *PEs* network:
    - 2D 8×8 torus.
    - 3DT 4×4×2 torus.
  - 128 *PEs* network:
    - 2D 16×8 torus.
    - 3DT 4×4×4 torus.
  - 256 (250) *PEs* network:
    - 2D 16×16 torus.
    - 3DT 8×4×4 torus.
    - 3DT 5×5×5 (250 *PEs*) torus[10].
  - 432 (441) *PEs* network:
    - 2D 24×18 torus.
    - 2D 21×21 (441 *PEs*) torus[11].
    - 3DT 8×4×4 torus.
  - 1024 *PEs* network:
    - 2D 32×32 torus.
    - 3DT 8×8×8 torus.
- All 3DT tori use the configuration *D*.
- Deadlock-avoidance mechanism:
  - Bubble flow control.
  - Virtual channel.

[10] This network size was chosen because it allows to build a 3DT torus with the same number of nodes in all dimensions and the number of nodes is closer to 256 nodes.

[11] This network size allows to build a 2D torus with the same number of nodes in all dimensions and the number of nodes is closer to 432 nodes.

7.4.2 Results

In this section we present graphically the results obtained in the experiments. Figure 19 shows the obtained results for the network with 64 PEs (left) and 128 PEs (right), whereas Figure 20 shows the results of 256 PEs (left) and 432 PEs. Finally, Figure 21 shows the results of 1024 PEs torus. (right) networks.

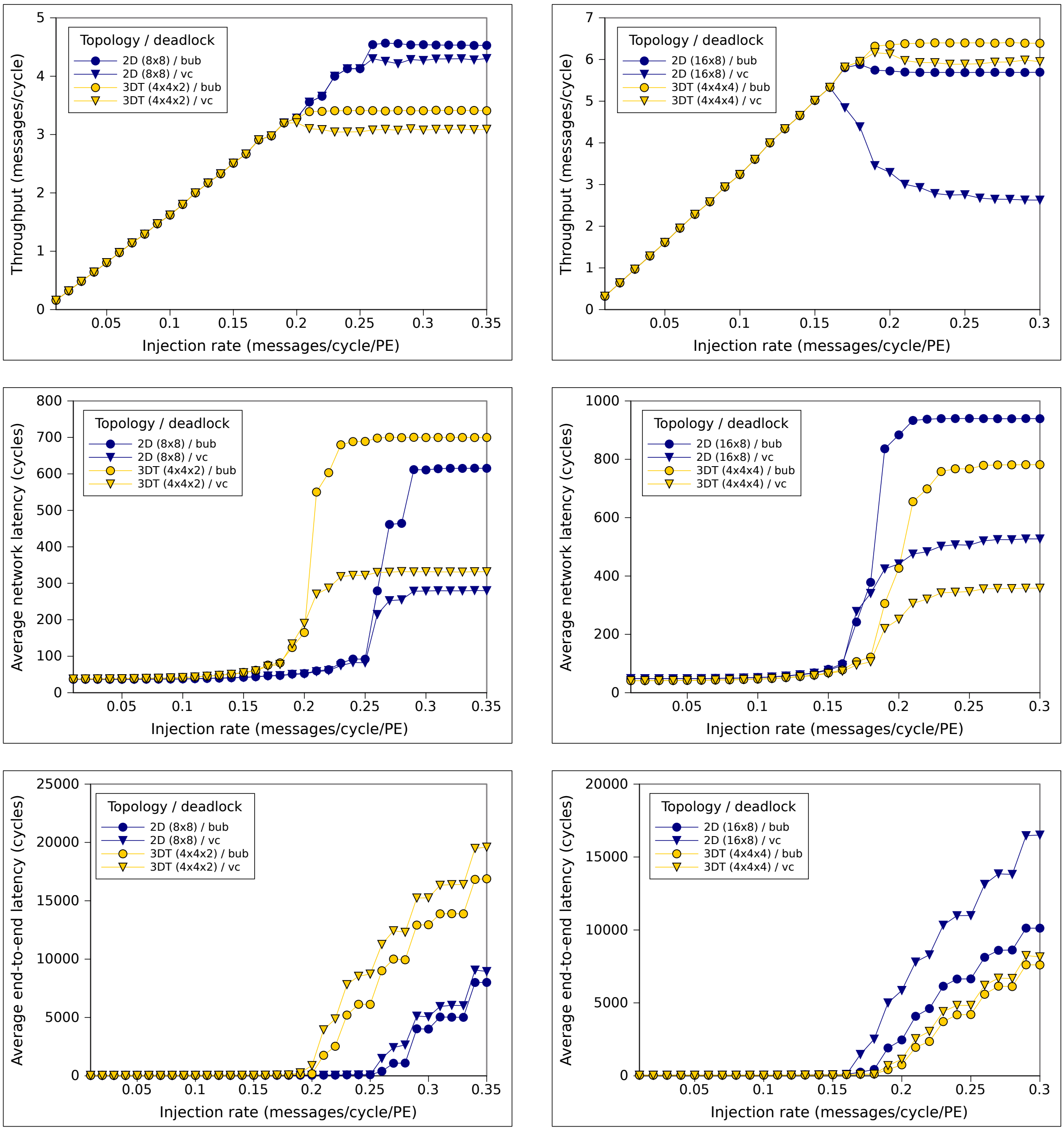


Figure 19: Network performance obtained for 2D and 3DT torus with 64 and 128 PEs.

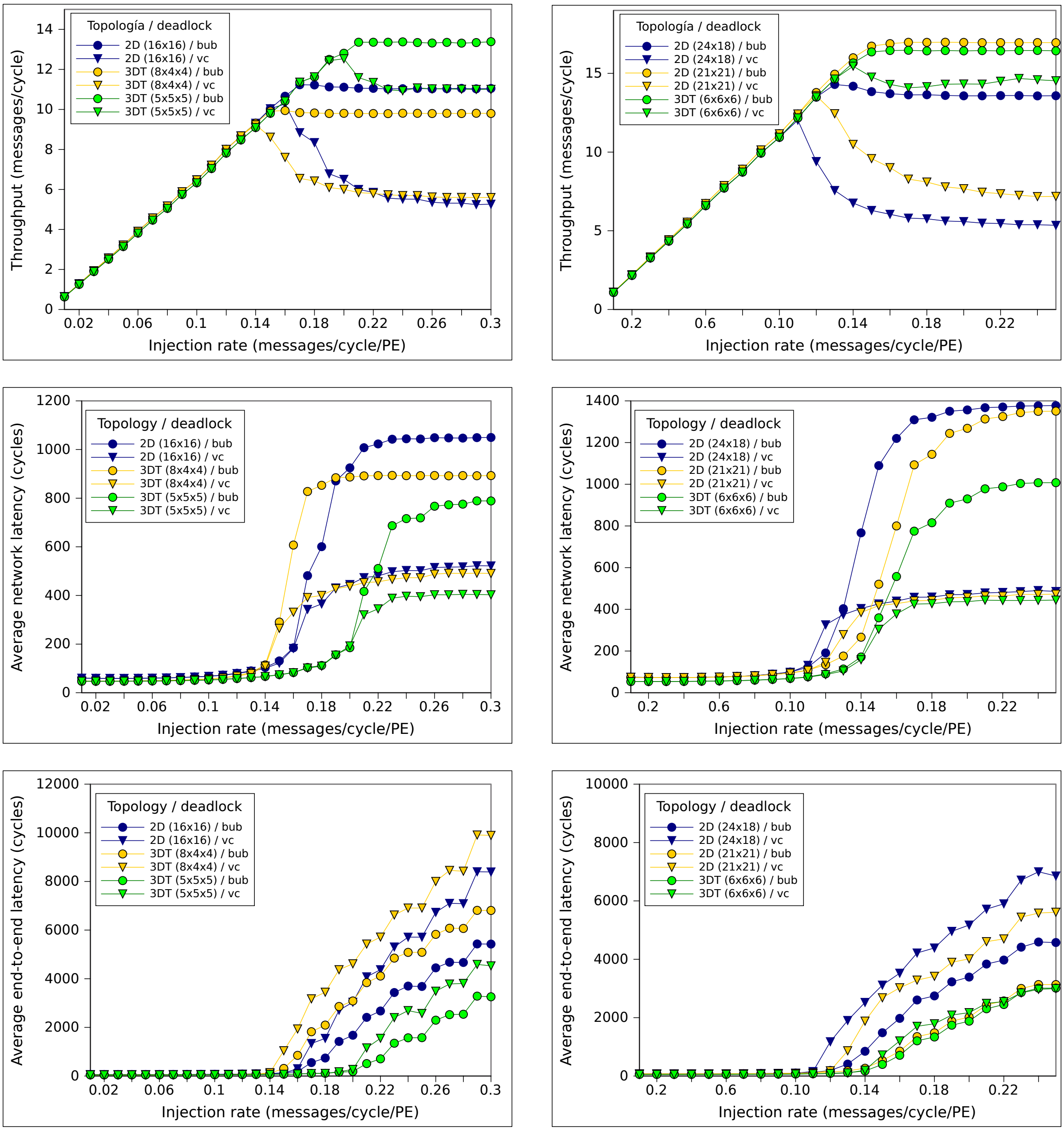


Figure 20: Network performance obtained for 2D and 3DT torus with 256 and 432 PEs.

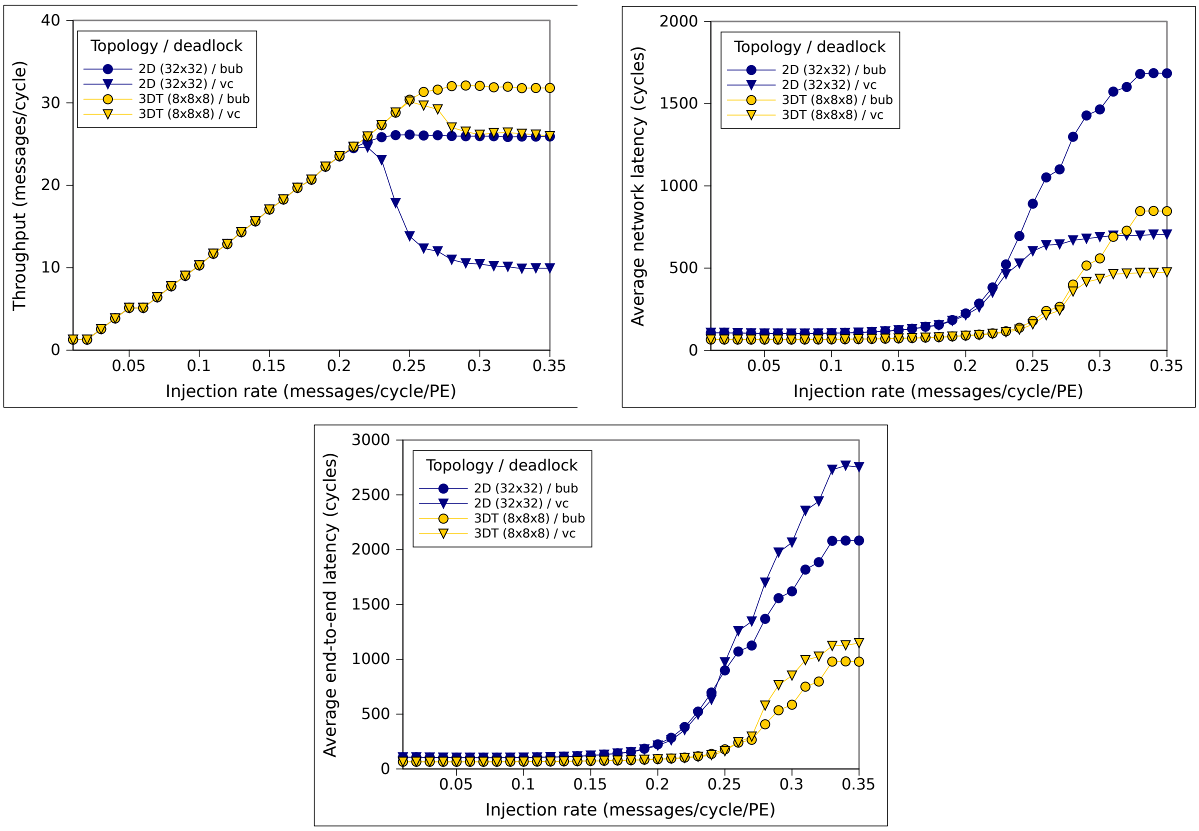


Figure 21: Network performance obtained for 2D and 3DT torus with 1024 PEs.

7.4.3 Analysis of the results

As expected after the results of the theoretical study, the 3DT torus obtains better performance than 2D torus in networks with more than 128 processing elements ($k \geq 4$). Only the performance of the 64-PEs 2D torus is higher than the equivalent 3DT torus. Moreover, the 8x4x4 3DT torus obtains worse results than its equivalent 2D torus, but this happens because the number of nodes is not the same in all dimensions. However, the 5×5×5 3DT torus, with the same size in the three dimensions, obtains a higher throughput and smaller network and end-to-end latency than the 2D torus whose number of nodes is approximately the same.

The 6x6x6 3DT torus also obtains worse network throughput than the equivalent 21×21 torus. This is due to the number of nodes is not exactly the same. However the throughput per node in the 21×21 torus is almost the same as the throughput per node in the 6×6×6 3DT torus. Although the network throughput is similar in both cases, the network latency is 35% higher in the 2D torus.

Regarding the decrease of performance experimented in the topologies using virtual channels to avoid *deadlock*, we can see how the decrease is higher in the 2D torus topology. Despite using the same switch in both cases, the 3DT torus topology is more resistant to the negative effect of congestion. Whereas the throughput decrease in 2D torus is 50% $\sim$ 60%, it is only 10% $\sim$ 20% in the 3DT torus.

As a result, there are large differences between the throughput obtained in the 2D torus and the 3DT torus. The 2D torus only reaches 50% (256 PEs), 47% (432 PEs) and 40% (1024 PEs) of the throughput obtained by its equivalent 3DT torus. These differences are shorter if the topology uses the bubble flow control mechanism, the 3DT torus only gets 15% more throughput than its equivalent 2D torus.

Finally, the graphs show the network latency is smaller if the network using virtual channels (up to 50% less), but using the bubble flow control mechanism the throughput increases 20% and the end-to-end latency is decreased, although the differences between the end-to-end latency become insignificant when the size of the network increases. As discussed above, the congestion decreases the throughput in the network using virtual channels, which obtain worse throughput and end-to-end latency than the network using the bubble flow control mechanism.

8 Conclusions and Future Work

8.1 Conclusions

After analyzing the results obtained in Section 7, the most relevant conclusions are the following:

- As expected from the result of theoretical study, **the node configuration D (and G if k is even) for the 3DT torus obtains the best performance** in networks larger than 128 PEs, although, due to the decrease of performance when the switches use virtual channels to avoid deadlock, there are other configurations that obtain similar performance than the configuration D. In any case, if the switches use the bubble flow control mechanism, the configuration D gets the highest performance.

- **The 3DT torus topology obtains a higher performance than a 2D torus topology** with the same or similar size, larger than 128 PEs, and using the same switch in both networks. Besides, the differences between the network performance become greater when the network size increases.

- Using the bubble flow control mechanism, the network obtains a higher throughput and a lower end-to-end latency, although the network latency increases respect the network built with switches using virtual channels to avoid *deadlock*.

- If switches use virtual channels for avoiding *deadlock*, **the 3DT torus topology is more resistant to performance degradation caused by congestion** than its equivalent 2D torus with the same number of PEs.

8.2 Future work

The work that we have presented in this document can be expanded in several ways. In the following, we present the research lines that could be followed in the future:

- For modeling the traffic load, we have only used a uniform traffic pattern. It is very interesting to study the performance of the 3DT torus and its configuration using other traffic patterns, and especially, real traffic loads, generated by scientific applications that perform their calculations on a 3D environment.

- It is also interesting to study how the different implementations of the 3DT torus influence on other aspects of the network, as the switch area or the power consumption.

- Another interesting idea is the development of adaptive routing algorithms that would take into account the internal structure of the node and try to minimize the use of the internal link.

- Extend the theoretical and simulation study presented here, but now for n-dimensional torus. Using cards with 6 ports or more and using the same approach as explained in this work, it is possible to build topologies with more dimensions.

References

[AAA$^+$02] N.R. Adiga, G. Almasi, Y. Aridor, R. Barik, D. Beece, R. Bellofatto, G. Bhanot, R. Bickford, M. Blumrich, and A. A. Bright. An overview of the blue gene/l supercomputer. In *Supercomputing 2002 Technical Papers*, 2002.

[Bol92] Kevin Bolding. Non-uniformities introduced by virtual channel deadlock prevention. Technical report, 1992.

[CBGV97] C. Carrion, R. Beivide, J.A. Gregorio, and F. Vallejo. A flow control mechanism to avoid message deadlock in k-ary n-cube networks. In *High-Performance Computing, 1997. Proceedings. Fourth International Conference on*, pages 322–329, dec 1997.

[DMS95] J. J. Dongarra, H. W. Meuer, and E. Strohmaier. TOP500 supercomputer sites. *j-SUPERCOMPUTER*, 11(2–3):133–163, June 1995.

[DS87] W.J. Dally and C.L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Computers, IEEE Transactions on*, C-36(5):547 –553, may 1987.

[DYN03] José Duato, Sudhakar Yalamanchili, and Lionel Ni. *Interconnection networks. An engineering approach.* Morgan Kaufmann Publishers Inc., 2003.

[IBM08] IBM Blue Gene Team. Overview of the ibm blue gene/p project. *IBM Journal of Research and Development*, 52(1/2), 2008.

[Inc09] Cray Inc. Jaguar supercomputer. Cray XT5-HE. `http://www.nccs.gov/jaguar/`, 2009.

[Inc11] Cray Inc. Hopper supercomputer. Cray XT5-He. `http://www.nersc.gov/users/computational-systems/hopper/`, 2011.

[Izu06] Cruz Izu. Throughput fairness in k-ary n-cube networks. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 137–145, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[KH98] M. Karol and M. Hluchyj. Queuing in high-performance packet-switching. *IEEE Journal on Selected Areas*, 1:1587–1597, 1998.

[Lei85] C. E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, 1985.

[MIM$^+$97] N. McKeown, M. Izzard, A. Mekkittikul, W. Ellersick, and M. Horowitz. The tiny tera: A packet switch core. *IEEE Micro*, 17:27–33, 1997.

[VRB$^+$11] Courtenay Vaughan, Mahesh Rajan, Richard Barrett, Doug Doerfler, and Kevin Pedretti. Investigating the impact of the Cielo Cray XE6 architecture on scientific application codes. In *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, IPDPSW'11, pages 1831–1837, Washington, DC, USA, 2011. IEEE Computer Society.

A Diameter and average distance of an n-dimensional torus

We include in this appendix the calculation of the diameter and the average distance for an n-dimensional torus. The case where the number of nodes in each dimension is even has a special interest because of the initial hypotheses considered in this technical report (Section 4).

Given an n-dimensional torus, with $k_0 \times k_1 \times \ldots \times k_{n-1}$ nodes, k_i nodes in dimension i, $0 \leq i < n$, we compute the expressions that determine its diameter (D) and its average distance (d_{avg}).

The largest, minimal path over all pairs of nodes is that crossing all dimensions, and traveling the greatest distance in each dimension. In an n-dimensional torus, every dimension is a bidirectional ring, so the diameter of a torus is:

$$D = \left\lfloor \frac{k_0}{2} \right\rfloor + \left\lfloor \frac{k_1}{2} \right\rfloor + \ ... \ + \left\lfloor \frac{k_{n-1}}{2} \right\rfloor$$

if all k_i are even, we have

$$D = \frac{k_0}{2} + \frac{k_1}{2} + \ ... \ + \frac{k_{n-1}}{2}$$

and if $k_i = k$, $0 \leq i < n$, then

$$D = n\frac{k}{2}$$

Similarly, the average distance will be the addition of the average distance in each dimension of the torus. As each dimension of the torus is a bidirectional ring of k_i nodes, and considering the special case where k_i is even, for each node there are:

- One node at 0 hops (the node itself).
- Two nodes at 1 hop.

  ...
- Two nodes at $k_i/2 - 1$ hops.
- One node at $k_i/2$ hops.

So, the average distance in the ring is:

$$d_{avg\_ring} = \frac{1}{k_i}\left((2 \times 1) + (2 \times 2) + \cdots + (2 \times \left(\frac{k_i}{2} - 1\right)) + \frac{k_i}{2}\right)$$

$$d_{avg\_ring} = \frac{\frac{k_i}{2} + 2\sum_{i=1}^{\frac{k_i}{2}-1} i}{k_i} = \frac{\frac{k_i}{2} + 2\frac{(\frac{k_i}{2}-1)\frac{k_i}{2}}{2}}{k_i} = \frac{\frac{k_i}{2} + \frac{k_i^2}{4} - \frac{k_i}{2}}{k_i} = \frac{\frac{k_i^2}{4}}{k_i} = \frac{k_i}{4}$$

and the average distance in the torus is:

$$d_{avg} = \frac{k_0}{4} + \frac{k_1}{4} + \ ... \ + \frac{k_{n-1}}{4}$$

and if $k_i = k$, $0 \leq i < n$, then

$$d_{avg} = n\frac{k}{4}$$