Bisection (Band)Width of Product Networks with Application to Data Centers*

Jordi Arjona Aroca[1,2] and Antonio Fernández Anta[1]

[1] Institute IMDEA Networks, Madrid, Spain
[2] Universidad Carlos III de Madrid, Madrid, Spain

Abstract. The bisection width of interconnection networks has always been important in parallel computing, since it bounds the amount of information that can be moved from one side of a network to another, i.e., the bisection bandwidth. The problem of finding the exact bisection width of the multidimensional torus was posed by Leighton and has remained open for 20 years. In this paper we provide the exact value of the bisection width of the torus, as well as of several d-dimensional classical parallel topologies that can be obtained by the application of the Cartesian product of graphs. To do so, we first provide two general results that allow to obtain upper and lower bounds on the bisection width of a product graph as a function of some properties of its factor graphs. We also apply these results to obtain bounds for the bisection bandwidth of a d-dimensional BCube network, a recently proposed topology for data centers.

Keywords: Bisection bandwidth, bisection width, torus, BCube, product graphs, complete binary trees, extended trees, mesh-connected trees.

1 Introduction

The bisection width and the bisection bandwidth of interconnection networks have always been two important parameters of a network. The first one reflects the smallest number of links which have to be removed to split the network in two equal parts, while the second one bounds the amount of data that can be moved between these parts. In general, both values are derivable one from the other, which is the reason why most previous work has been devoted to only one of then (in particular, the bisection width).

The bisection width has been a typical goodness parameter to evaluate and compare interconnection networks for parallel architectures [14, 7, 5]. This interest has been transferred to the Network-On-Chip topologies, as the natural successors of the parallel architectures of the 90's [13, 15, 22, 19]. The bisection (band)width is also nowadays being used as a reference parameter on the analysis of the latest topologies that are being deployed in data centers. This can be seen in recent papers which propose new topologies, like BCube [11] or DCell [12]. The bisection (band)width is used to compare these new topologies with classical topologies, like grids, tori, and hypercubes, or with other datacenter topologies, like trees and fat trees.

* This research was supported in part by the Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TEC2011-29688-C02-01, and National Natural Science Foundation of China grant 61020106002.

Finding the exact value of the bisection width is hard in general. Computing it has proven to be challenging even for very simple families of graphs. For instance, the problem of finding the exact bisection width of the multidimensional torus was posed by Leighton [14, Problem 1.281] and has remained open for 20 years. One general family of interconnection networks, of which the torus is a subfamily, is the family of product networks. The topology of these networks is obtained by combining factor graphs with the Cartesian product operator. This technique allows to build large networks from the smaller factor networks. Many popular interconnection networks are instances of product networks, like the grid and the hypercube. In this paper we derive techniques to bound the bisection width of product networks, and apply these techniques to obtain the bisection width of some product network families.

Related Work To our knowledge, Youssef [20, 21] was among the first to explore the properties of product networks as a family. He presented the idea of working with product networks as a divide-and-conquer problem, obtaining important properties of a product network in terms of the properties of its factor graphs.

The bisection width of arrays and tori was explored by Dally [6] and Leighton [14] in the early 90s, presenting exact results for these networks when the number of nodes per dimension was even. The case when there are odd number of nodes per dimension was left open. Rolim et al. [18] gave the exact values for the bisection width of 2 and 3-dimensional grids and tori, but left open the question for longer number of dimensions.

For the special case in which all the factors are isomorphic, Efe and Fernández [9] provided a lower bound on the bisection width of a product graph as a function of a new parameter of a factor network they defined, the maximal congestion. Nakano [16] presented the exact value of the bisection width for the Cartesian product of isomorphic paths and cliques (i.e., square grids and Hamming graphs). If the factor graphs have k nodes, he proved that the d-dimensional square grid has bisection width k^{d-1} when k is even, and $\frac{(k^d-1)}{(k-1)}$ when k is odd. Similarly, the square Hamming graph has bisection width k^{d+1} when k is even, and $(k+1)\frac{(k^d-1)}{4}$ when k is odd. The exact bisection width of the d-dimensional square grid was found independently by Efe and Feng [8].

For the present paper it is very relevant the work of Azizoglu and Egecioglu. In [2] and [4] they studied the relationship between the isoperimetric number and the bisection width of different product networks. In the former paper, they find the exact value of the bisection width of the cylinders (products of paths and rings) with even number of nodes in its largest dimension. In the latter reference they found the exact bisection width of the grid $A^{(d)}_{k_1,k_2,\ldots,k_d}$, with k_i nodes along dimension i, and where $k_1 \geq k_2 \geq \ldots \geq k_d$. The value of this bisection width is $BW(A^{(d)}_{k_1,k_2,\ldots,k_d}) = \sum_{i=1}^{\alpha} C_i$, where α is the smallest index for which k_i is even ($\alpha = d$ if no index is even), and $C_i = \prod_{j=i+1}^{d} k_j$. Since this value will appear frequently, we will use the following notation throughout the rest of the paper, $\Psi(\alpha) = \sum_{i=1}^{\alpha} C_i = \sum_{i=1}^{\alpha} \prod_{j=i+1}^{d} k_j$.

Contributions In this paper we present two theorems that allow to derive lower and upper bounds on the bisection width of a product network as a function of some simple parameters of its factor graphs. Then, we apply these results to obtain the exact value of the bisection width for several families of product networks. The families presented

are of interest because they have been proposed as interconnection networks for parallel architectures, but their bisection width has never been derived exactly.

One of the most interesting contribution of this paper is the exact value of the bisection width of the torus, since, as mentioned before, this problem has been open for almost 20 years. We find here that the exact value of the bisection width of a d-dimensional torus $T^{(d)}_{k_1,k_2,\ldots,k_d}$, that has k_i nodes along dimension i, and where $k_1 \geq k_2 \geq \ldots \geq k_d$, is exactly twice the bisection width of the grid of similar dimensions $A^{(d)}_{k_1,k_2,\ldots,k_d}$. I.e., $BW(T^{(d)}_{k_1,k_2,\ldots,k_d}) = 2\Psi(\alpha) = 2\sum_{i=1}^{\alpha} C_i$, where α is the smallest index for which k_i is even ($\alpha = d$ if no index is even), and $C_i = \prod_{j=i+1}^{d} k_j$. In addition to the result for the torus, we provide the exact value for the bisection width of products of complete binary trees (CBT) of any size (mesh-connected trees [10]), products of extended CBT (which are CBT with the leaves connected with a path [10]), products of CBT and paths, and products of extended CBT and rings. To obtain the bisection *bandwidth* of these networks, we assume that every edge removed by the bisection width is in fact a duplex link with bandwidth of T in each direction. This directly implies that for any of these networks G, the bisection bandwidth is computed as $BBW(G) = 2T \cdot BW(G)$.

The general upper and lower bound results are also used to derive bounds on the bisection bandwidth of a topology proposed for datacenters, the BCube. A BCube is the Cartesian product of factors networks formed by k nodes connected via a k-port switch (where the switch is not considered to be a node). An essential difference of this topology from the previous one is that edges do not connect nodes directly, and the direct relation between bisection width and bisection bandwidth does not hold anymore. In networks with switches like this one, the switching capacity s of the switch comes into play as well. Since the bisection bandwidth is the parameter of interest in datacenters, we derive bounds on its value for two cases: when the bottleneck for the bisection bandwidth is at the links (Model A), and when it is at the switches (Model B).

Table 1 summarizes the results derived for the bisection bandwidth obtained for the different parallel topologies and for BCube. As can be seen, for the former the values

Table 1. Bisection bandwidth of different product networks

Product graph	**Factor graphs**		$\beta(G)$	$CC(G)$	**Bisection bandwidth**
Torus	Ring		$1/8$	2	$4T \cdot \Psi(\alpha)$
Product of extended CBT	XTs		$1/8$	2	$4T \cdot \Psi(\alpha)$
Product of extended CBT & rings	Rings & XTs		$1/8$	2	$4T \cdot \Psi(\alpha)$
Mesh connected trees	CBT		$1/4$	1	$2T \cdot \Psi(\alpha)$
Product of CBT and paths	Paths & CBTs		$1/4$	1	$2T \cdot \Psi(\alpha)$
BCube	Model A	even	$\frac{k-1}{k^2}$	$\frac{k}{2}$	$2T\frac{k^{d+1}}{4(k-1)} \leq BBW(BCA^{(d)}_k) \leq 2T\frac{k^d}{2}$
		odd	$\frac{1}{k+1}$	$\frac{k-1}{2}$	$2T\frac{k+1}{4}\frac{k^d-1}{k-1} \leq BBW(BCA^{(d)}_k) \leq 2T\frac{k^d-1}{2}$
	Model B	even	$\frac{k-1}{2k}$	1	$s\frac{k^d}{2(k-1)} \leq BBW(BCB^{(d)}_k) \leq s\frac{k^d-1}{k-1}$
		odd	$\frac{k}{2(k+1)}$	1	$s\frac{k+1}{2k}\frac{k^d-1}{k-1} \leq BBW(BCB^{(d)}_k) \leq s\frac{k^d-1}{k-1}$

obtained are exact, while for the latter the upper and lower bounds found do no match exactly. However, they differ by less than a factor of two.

The rest of the paper is organized as follows. Section 2 presents some basic definitions used in the rest of sections. In Section 3 we provide the general results to derive bounds on the bisection bandwidth of product networks. Section 4 and Section 5 present our results for the bisection bandwidth of some classical parallel topologies. Bounds on the bisection bandwidth of the BCube network are presented in Section 6.

Due to space limitations, some proofs have been omitted. They can be found in [1].

2 Definitions

Graphs and Bisections Given a graph[3] G, we denote its sets of vertices and edges as $V(G)$ and $E(G)$, respectively. In some cases, when it is clear from the context, only V or E will be used, omitting the graph G. Unless otherwise stated, the graphs considered are undirected.

Given a graph G with n nodes, we use $S(G)$ to denote a subset of $V(G)$ such that $|S(G)| \leq \frac{n}{2}$. We also use $\partial^G S(G)$ to denote the set of edges connecting $S(G)$ and $V(G) \setminus S(G)$. Formally, $\partial^G S(G) = \{(u,v) \in E(G) : u \in S(G), v \in G \setminus S(G)\}$. The graph G may be omitted from this notation when it is clear from the context.

The main object of this work is to calculate the bisection width and bisection bandwidth of different product networks. The *bisection width* of an n-node graph G, denoted by $BW(G)$, is the smallest number of edges that have to be removed from G to partition it in two halves. Formally, $BW(G) = \min_{S:|S|=\lfloor \frac{n}{2} \rfloor} |\partial^G S|$. The *bisection bandwidth* of a network G, denoted by $BBW(G)$, is the minimal amount of traffic which can be transferred between any two halves of the network when its links are transmitting at full speed. As mentioned above, unless otherwise stated we assume that all the links in a network G are duplex and have the same capacity T in each direction. Then, we can generally assume that the relation between the bisection bandwidth and the bisection width is $BBW(G) = 2T \cdot BW(G)$.

Factor and Product Graphs We define first the Cartesian product of graphs.

Definition 1. *The* d-dimensional Cartesian product *of graphs* $G_1, G_2, ..., G_d$*, denoted by* $G_1 \times G_2 \times \cdots \times G_d$*, is the graph with vertex set* $V(G_1) \times V(G_2) \times \cdots \times V(G_d)$*, in which vertices* $(u_1, ..., u_i, ..., u_d)$ *and* $(v_1, ..., v_i, ..., v_d)$ *are adjacent if and only if* $(u_i, v_i) \in E(G_i)$ *and* $u_j = v_j$ *for all* $j \neq i$*.*

The graphs $G_1, G_2, ..., G_d$ are called the *factors* of $G_1 \times G_2 \times \cdots \times G_d$. Observe that $G_1 \times G_2 \times \cdots \times G_d$ contains $\prod_{j \neq i} |V(G_j)|$ disjoint copies of G_i, which form dimension i.

We define now some of the basic factor graphs that will be considered. The *path* of k vertices, denoted by P_k, is a graph such that $V(P_k) = \{0, 1, \ldots, k-1\}$ and where $E(P_k) = \{(i, i+1) : i \in [0, k-2]\}$. The *complete graph* (a.k.a. the clique) of k vertices, denoted by K_k, is a graph such that $V(K_k) = \{0, 1, \ldots, k-1\}$ and where $E(K_k) = \{(i,j) : (j \neq i) \wedge (i, j \in V(K_k))\}$. The *r-complete graph* of k vertices denoted by rK_k, is a graph such that $V(rK_k) = \{0, 1, \ldots, k-1\}$ and where $E(rK_k)$

[3] Unless otherwise stated we will use the terms graph and network indistinctly.

is a multiset such that each pair of vertices $i, j \in V(rK_k)$ is connected with r parallel edges. (i.e., each $e \in E(rK_k)$ has multiplicity r).

Using these and other graphs as factors, we will define, across the text, different d-dimensional Cartesian product graphs. For convenience, for these graphs we will use the general notation $G^{(d)}_{k_1,\ldots,k_d}$, where G is the name of the graph, the superscript (d) means that it is a d-dimensional graph, and $k_1, \ldots k_d$ are the number of vertices in each dimension. (Superscript and subscripts may be omitted when clear from the context.) It will always hold that $k_1 \geq k_2 \geq \ldots \geq k_d$, i.e., the factor graphs are sorted by decreasing number of vertices. We will often use n to denote the number of nodes a the graph $G^{(d)}_{k_1,\ldots,k_d}$, i.e., $n = k_1 k_2 \cdots k_d$, and we will always use α to denote the index of the lowest dimension with an even number of vertices (if there is no such dimension, $\alpha = d$, where d is the index of the lowest dimension).

According to this notation we will present different d-dimensional product graphs as follows. The *d-dimensional array*, denoted by $A^{(d)}_{k_1,\ldots,k_d}$, is the Cartesian product of d paths of $k_1, \ldots, k_d$ vertices, respectively. I.e., $A^{(d)}_{k_1,\ldots,k_d} = P_{k_1} \times P_{k_2} \times \cdots \times P_{k_d}$. The *d-dimensional r-Hamming graph*, denoted by $rH^{(d)}_{k_1,\ldots,k_d}$, is the Cartesian product of d r-complete graphs of $k_1, \ldots, k_d$ nodes, respectively. I.e., $rH^{(d)}_{k_1,\ldots,k_d} = rK_{k_1} \times rK_{k_2} \times \cdots \times rK_{k_d}$. Observe that the *Hamming graph* [3] is the particular case of the r-Hamming graph, with $r = 1$. For brevity, we use $H^{(d)}_{k_1,\ldots,k_d}$ instead of $1H^{(d)}_{k_1,\ldots,k_d}$, to denote the Hamming graph.

Boundaries and Partitions We define now the dimension-normalized boundary [4]. Let $G^{(d)}_{k_1,\ldots,k_d}$ be a d-dimensional product graph and $S(G)$ a subset of $V(G)$. Then, the *dimension-normalized boundary* of $S(G)$, denoted by $B_G(S)$, is defined as $B_G(S) = \frac{|\partial^G_1 S|}{\sigma_1} + \frac{|\partial^G_2 S|}{\sigma_2} + \ldots + \frac{|\partial^G_d S|}{\sigma_d}$, where, for each $i \in [1, d]$, ∂^G_i is ∂^G applied to the dimension i of G and $\sigma_i = k_i^2 - (k_i \bmod 2)$. Observe that for $rH^{(d)}_{k_1,\ldots,k_d}$, any subset S of nodes, and any dimension i, it holds that $|\partial^{rH}_i S| = r \cdot |\partial^H_i S|$. Hence, $B_{rH}(S) = \frac{|\partial^{rH}_1 S|}{\sigma_1} + \cdots + \frac{|\partial^{rH}_d S|}{\sigma_d} = r(\frac{|\partial^H_1 S|}{\sigma_1} + \cdots + \frac{|\partial^H_d S|}{\sigma_d}) = r \cdot B_H(S)$.

Let us define now the lexicographic-order. Consider graph $H^{(d)}_{k_1,\ldots,k_d}$, we say that vertex $x = (x_1, x_2, \ldots, x_d)$ precedes vertex $y = (y_1, y_2, \ldots, y_d)$ in *lexicographic-order* if there exists an index $i \in [1, d]$ such that $x_i < y_i$ and $x_j = y_j$ for all $j < i$. Azizoğlu and Eğecioğlu [3] proved the following result.

Theorem 1 ([3]). *Let S be any subset of $V(H)$ and $\bar{S}$ the set of first $|S|$ vertices of H in lexicographic-order[4], then $B_H(\bar{S}) \leq B_H(S)$.*

3 Bounds on the Bisection Width of Product Graphs

In this section we present general bounds on the bisection width of product graphs as well as presenting two important parameters, the normalized congestion and the central cut, which are used to obtain them. These bounds will be used in the upcoming sections to find the bisection width of several instances of product graphs.

[4] Observe that we have reversed the ordering of dimensions with respect to the original theorem from Azizoğlu and Eğecioğlu.

Lower Bound We start by defining the normalized congestion of a graph. Let G be a graph with n nodes. Then, an *embedding* of graph rK_n onto G is a mapping of the edges of rK_n into paths in G. We define the *congestion of G with multiplicity r*, denoted by $m_r(G)$, as the minimum (over all such embeddings) of the maximum number of embedded paths that contain an edge from G. To formally define this concept, we first define the congestion of an edge $e \in E(G)$ under the embedding M_r of rK_n onto G, denoted by $c_{M_r}(e)$, as $c_{M_r}(e) = |\{e' \in E(rK_n) : e \in M_r(e')\}|$. (Observe that $M_r(e') \subseteq E(G)$ is a path in G.) Then, the congestion $m_r(G)$ is $m_r(G) = \min_{M_r \in \mathcal{E}} \max_{e \in E(G)}\{c_{M_r}(e)\}$, where $\mathcal{E}$ is the set of all possible embeddings of rK_n onto G. Then, we define the normalized congestion with multiplicity r of G as $\beta_r(G) = \frac{m_r(G)}{\sigma_n}$. We proceed to extend Theorem 1 to r-Hamming graphs.

Theorem 2. *Consider a d-dimensional r-Hamming graph $rH^{(d)}$. Let S be any vertex subset of $V(rH^{(d)})$ and $\bar{S}$ the set of first $|S|$ vertices of $rH^{(d)}$ in lexicographic order, then $B_{rH}(\bar{S}) \leq B_{rH}(S)$.*

Proof. We prove the theorem by contradiction. Assume that there is a set of vertices $X \neq \bar{S}$ such that $|X| = |\bar{S}|$ and $B_{rH}(\bar{S}) > B_{rH}(X)$. Then, applying the fact that $|\partial_i^{rH} S| = r \cdot |\partial_i^H S|$ to both X and $\bar{S}$, we obtain that $B_H(\bar{S}) = \frac{B_{rH}(\bar{S})}{r} > \frac{B_{rH}(X)}{r} = B_H(X)$, which contradicts Theorem 1 and proves the theorem.

Then, from the definition of $B_H(\bar{S})$, we obtain the following.

Theorem 3. *Let $G = G_1 \times \ldots \times G_d$, where $|V(G_i)| = k_i$ and $k_1 \geq k_2 \geq \ldots \geq k_d$. Let $\beta_r(G_i)$ be the normalized congestion with multiplicity r of G_i (for any r), for all $i \in [1, d]$. Consider any subset $S \subset V(G)$ and the subset $\bar{S}$ which contains the first $|S|$ vertices of G, in lexicographic order. Then, $B_{rH}(\bar{S}) \leq \sum_{i=1}^{d} \beta_r(G_i)|\partial_i^G S|$*

Corollary 1. *Let G and $\beta_r(G_i)$ be defined as in Theorem 3. Consider any subset $S \subset V(G)$ such that $|S| = \lfloor \frac{|V(G)|}{2} \rfloor$. Then $\frac{r}{4}\Psi(\alpha) \leq \sum_{i=1}^{d} \beta_r(G_i)|\partial_i^G S|$. When $\beta_r(G_i) = \beta$ for all $i \in [1, d]$, this implies $\frac{r}{4\beta}\Psi(\alpha) \leq BW(G)$.*

Upper Bound Having proved the lower bound on the bisection width, we follow with the upper bound. We define first the central cut of a graph G. Consider a graph G with n nodes, and a partition of $V(G)$ into three sets S^-, S^+, and S, such that $|S^-| = |S^+| = \lfloor \frac{n}{2} \rfloor$ (observe that if n is even then $S = \emptyset$, otherwise $|S| = 1$). Then, the *central cut of G*, denoted by $CC(G)$, is

$$\min_{\{S^-, S^+, S\}} \max\{|\partial^G S^-|, |\partial^G S^+|\}.$$

Observe that, for even n, the central cut is the bisection width. Now, we use the definition of central cut in the following theorem.

Theorem 4. *Let $G = G_1 \times \ldots \times G_d$. Then, $BW(G) \leq \max_i \{CC(G_i)\} \cdot \Psi(\alpha)$.*

(a) The 4-vertex path and clique

(b) The 5-vertex path and clique

Fig. 1. Paths and their possible cuts

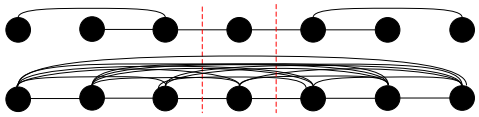


Fig. 2. The 7-vertex complete binary tree and the 7-vertex clique, with their possible cuts

4 Bisection Width of Products of Paths and CBT

In this section we will obtain the bisection bandwidth of product graphs which result from the Cartesian product of paths and Complete Binary Trees (CBT). We will present, first, the different factor graphs we are using and the product graphs we are bisecting, then, we will compute the congestion and central cut of these factor graphs and, finally, calculate the bisection width of these product graphs.

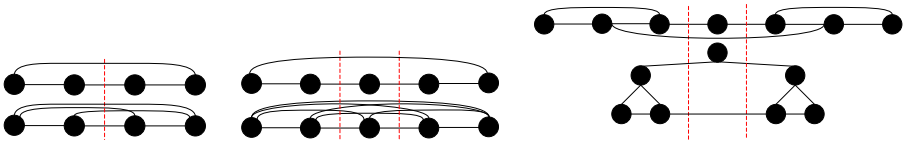
Factor and Product Graphs Paths were defined in Section 2. The *complete binary tree* of k vertices, denoted by CBT_k, is a graph such that $V(CBT_k) = \{1, 2, \ldots, k\}$, with $k = 2^j - 1$ (j is the number of levels of the tree), and where $E(CBT_k) = \{(i, j) : ((j = 2i) \vee (j = 2i + 1)) \wedge (i \in [1, 2^{j-1} - 1])\}$. Combining these factor graphs through the Cartesian product, we obtain the product networks that we define below. A *d-dimensional mesh-connected trees and paths*, denoted by $MCTP^{(d)}_{k_1,k_2,\ldots,k_d}$, is the Cartesian product of d graphs of $k_1, k_2, \ldots, k_d$ vertices, respectively, where each factor graph is a complete binary tree or a path. I.e., $MCTP^{(d)}_{k_1,k_2,\ldots,k_d} = G_{k_1} \times G_{k_2} \times \cdots \times G_{k_d}$, where either $G_{k_i} = CBT_{k_i}$ or $G_{k_i} = P_{k_i}$. We also define the *d-dimensional mesh-connected trees* [10], denoted by $MCT^{(d)}_{k_1,k_2,\ldots,k_d}$ as the graph $MCTP^{(d)}_{k_1,k_2,\ldots,k_d}$ in which all the factor graphs are complete binary trees. (Observe that the array is also the special case of $MCTP^{(d)}_{k_1,k_2,\ldots,k_d}$ in which all the factor graphs are paths.)

Congestion and Central Cut of Paths and CBT The bisection widths of the aforementioned product graphs can be calculated using the bounds defined in Section 3. To do so, we need to compute first the values of the normalized congestion and central cut of their factor graphs, that is, of a path and of a CBT.

The value of the congestion of a CBT is exactly the same as the congestion of a path with an odd number of nodes. CBT share withe paths the property of having only one possible routing between two nodes. As can be seen in Figures 1 and 2, the possible cuts are similar. We can show that the normalized congestion of both paths or CBTs is exactly $\beta_r(P_k) = \beta_r(CBT_k) = \frac{r}{4}$.

The value of the central cut of both the path and CBT can also be easily deduced from Figures 1 and 2, being $CC(P_k) = CC(CBT_k) = 1$.

Bounds on the Bisection Width of Products of CBTs and Paths We can compute now the bisection width of a product of CBTs and paths from the congestion and the central cut of the possible factor graphs, directly applying the results of Section 3.



(a) The 4-vertex ring and clique

(b) The 5-vertex ring and clique

(c) Central cut on a extended complete binary tree

Fig. 3. Rings and extended complete binary tree possible cuts

Theorem 5. *The bisection width of a d-dimensional mesh-connected trees and paths $MCTP^{(d)}_{k_1,k_2,\ldots,k_d}$ is $\Psi(\alpha)$. Hence, the bisection width of the d-dimensional mesh-connected trees $MCT^{(d)}_{k_1,k_2,\ldots,k_d}$ is $BW(MCT^{(d)}) = \Psi(d)$.*

5 Products of Rings and Extended Trees

In this section we will obtain a result for the bisection bandwidth of the product graphs which result from the Cartesian product of rings and extended complete binary trees.

Factor and Product Graphs The *ring* of k vertices, denoted by R_k, is a graph such that $V(R_k) = \{0, 1, \ldots, k-1\}$ and where $E(R_k) = \{(i, (i+1) \mod k) : i \in V(R_k)\}$. The *extended complete binary tree* (a.k.a. XT) of k vertices, denoted by X_k, is a complete binary tree in which the leaves are connected as a path. More formally, $V(X_k) = V(CBT_k)$ and $E(X_k) = E(CBT_k) \cup \{(i, i+1) : i \in [2^{j-1}, 2^j - 2]\}$. Combining these graphs as factor graphs in a Cartesian product, we can obtain the three different kinds of product graphs. A *d-dimensional mesh-connected extended trees and rings*, denoted by $MCXR^{(d)}_{k_1,k_2,\ldots,k_d}$, is the Cartesian product of d graphs of $k_1, k_2, \ldots, k_d$ vertices, respectively, where each factor graph is an XT or a ring. I.e., $MCXR^{(d)}_{k_1,k_2,\ldots,k_d} = G_{k_1} \times G_{k_2} \times \cdots \times G_{k_d}$, where either $G_{k_i} = X_{k_i}$ or $G_{k_i} = R_{k_i}$. The *d-dimensional torus*, denoted by $T^{(d)}_{k_1,k_2,\ldots,k_d}$, is the Cartesian product of d rings of $k_1, k_2, \ldots, k_d$ vertices, respectively. I.e., $T^{(d)}_{k_1,k_2,\ldots,k_d} = R_{k_1} \times R_{k_2} \times \cdots \times R_{k_d}$. And, as happened in Section 4 with $MCT^{(d)}$, we also define the *d-dimensional mesh-connected extended trees*, denoted by $MCX^{(d)}_{k_1,k_2,\ldots,k_d}$, a special case of $MCXR^{(d)}_{k_1,k_2,\ldots,k_d}$ in which all factor graphs are XT. (The torus is the special case of $MCXR^{(d)}_{k_1,k_2,\ldots,k_d}$ in which all factor graphs are rings.)

Congestion and Central Cut of Rings and XT The congestion and central cut of both a ring and an XT are needed to apply the bounds obtained in Section 3. Similarly to what happened with paths and CBTs, the congestion of rings and XT is the same. The extended complete binary tree X_k has a Hamiltonian cycle [10], so we can find a ring R_k contained onto it. Consequently, the congestion of an XT and a ring with the same number of nodes will be the same. It can be shown that both normalized congestions with multiplicity $r = 2$ is $\beta_2(R_k) = \beta_2(X_k) = 1/4$. Due to these similarities, central cuts of both graphs are also going to be the same, as can be easily observed from Figures 3(a), 3(b) and 3(c), $CC(R_k) = CC(X_k) = 2$.

Bounds on the Bisection Width of Products of XT and Rings With the normalized congestion and central cut of the different factor graphs, we can obtain the bisection width of products of XT and rings.

Theorem 6. *The bisection width of a d-dimensional mesh-connected extended trees and rings $MCXR^{(d)}_{k_1,k_2,\ldots,k_d}$ is $2\Psi(\alpha)$. Hence, the bisection width of the d-dimensional torus $T^{(d)}$ is $BW(T^{(d)}) = 2\Psi(\alpha)$ and the bisection width of the d-dimensional mesh-connected extended trees $MCX^{(d)}$ is $BW(MCX^{(d)}) = 2\Psi(d)$.*

6 BCube

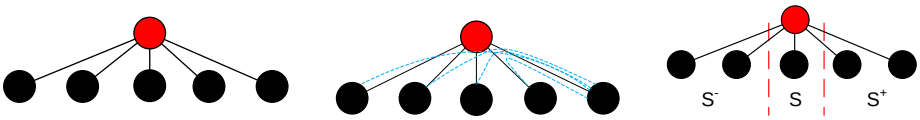
We devote this section to obtain bounds on the bisection width of a d-dimensional BCube [11]. BCube is different from the topologies considered in the previous sections because it is obtained as the combination of basic networks formed by a collection of k nodes (servers) connected by a switch. These factor networks are combined into multidimensional networks in the same way product graphs are obtained from their factor graphs. This allows us to study the BCube as an special instance of a product network. The d-dimensional BCube can be obtained as the d dimensional product of one-dimensional BCube networks, each one of k nodes.

Factor and Product Graphs We first define a *Switched Star network* and how a d-dimensional BCube network is built from it. A *Switched Star network* of k nodes, denoted by SS_k, is composed of k nodes connected to a k-ports switch. It can be seen as a complete graph K_k where all the edges have been replaced by a switch. Combining d copies of this network as factor networks of the Cartesian product, we obtain a d-dimensional BCube. Hence, a *d-dimensional BCube*, denoted by $BC^{(d)}_k$, is the Cartesian product of d SS_k (the switches are not considered nodes for the Cartesian product). I.e., $BC^{(d)}_k = SS_k \times SS_k \times \cdots \times SS_k$. $BC^{(d)}_k$ can also be seen as a d-dimensional homogeneous array where all the edges in each path have been removed and replaced by a switch where two nodes $(u_1, ..., u_i, ..., u_d)$ and $(v_1, ..., v_i, ..., v_d)$ are connected to the same switch if and only if $(u_i \neq v_i)$ and $u_j = v_j$ for all $j \neq i$.

The main reason for obtaining the bisection width of a d-dimensional BCube is to be able to bound its bisection bandwidth. However, as the d-dimensional BCube is not a typical graph, the bisection width can have different forms depending on where the communication bottleneck is located in a BCube network. We present two possible models for SS_k. The first one, Model A or *star-like model,* denoted by SSA_k, consists of k nodes connected one-to-one to a virtual node which represents the switch. The second one, Model B or *hyperlink model,* denoted by SSB_k, consists of k nodes connected by a hyperlink[5]. While the two presented models are logically equivalent to a complete graph, they have a different behavior from the traffic point of view. We show this with two simple examples.

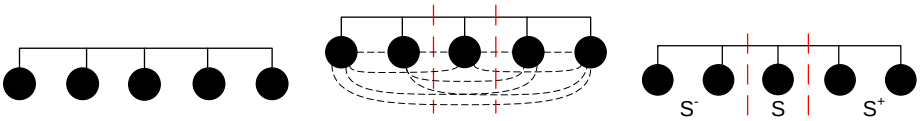
Let us consider that we have a SS_3 where the links have a speed of 100 Mbps while the switch can switch at 1 Gbps. Under these conditions, the links become the bottleneck of the network and, even when the switches would be able to provide a bisection bandwidth of 1 Gbps, the effective bisection bandwidth is only of 200 Mbps in both directions. Consider the opposite situation now, where the BCube switch only supports 500 Mbps of internal traffic, while the links transmit at 1 Gbps. In this case, the switches are the bottleneck of the network and the bisection bandwidth is only 500 Mbps, although the links would be able to support up to 2 Gbps.

[5] This model is quite similar to the one proposed by Pan in [17].



(a) A 5-node star-like SS (b) Congestion of a 5-node star-like SS (c) Central cut of a 5-node star-like SS

Fig. 4. Model A of a 5-node switched star SSA_5 and its congestion and central cut



(a) A 5-node hyperlink SS (b) Congestion of a 5-node hyperlink SS (c) Central cut of a 5-node hyperlink SS

Fig. 5. Model B of a 5-node switched star SSB_5 and its congestion and central cut

The first example illustrates an scenario where we would bisect the network by removing the links that connect the servers to the switches, which corresponds to Model A. On the other hand, what we find in the second example is a typical scenario for Model B, where we would do better by removing entire switches when bisecting the network. In particular, being s the switching capacity of a switch, and T the traffic supported by a link, we will choose Model A when $s \geq \lfloor \frac{k}{2} \rfloor \cdot 2T$ and Model B when $s \leq 2T$. (Note that this does not cover the whole spectrum of possible values of s, T, and k.)

Congestion and Central Cut of BCube We will compute now the congestion and central cut of both models in order to be able to calculate the respective lower and upper bounds. We start by the congestion and central cut of Model A. If we set $r = 1$, the congestion of every link of the star is easily found[6] to be $m_r(SSA_k) = k - 1$ as shown in Figure 4(b). The central cut, which is also trivial, can be found in Figure 4(c). Both will depend on whether the number of nodes k is even or odd.

Lemma 1. *The normalized congestion of SSA_k is $\beta_r(SSA_k) = \frac{k-1}{k^2-b}$, and the central cut is $CC(SSA_k) = \frac{k-b}{2}$, where $b = k \bmod 2$.*

Having computed the congestion and the central cut for Model A, we will compute them now for Model B. If we set $r = 1$ there will be only one edge to be removed, the congestion of the graph will be total amount of edges of its equivalent K_k, i. e., $m_r(SSB_k) = \frac{k(k-1)}{2}$. The central cut is also easily computed, as there is only one hyperlink. Both $m_r(SSB_k)$ and $CC(SSB_k)$ are shown in Figure 5.

Lemma 2. *The normalized congestion of SSB_k is $\beta_r(SSB_k) = \frac{k-1}{2(k^2-b)}$, where $b = k \bmod 2$, and the central cut is $CC(SSB_k) = 1$.*

Bounds on the Bisection Width of BCube Having computed the congestion and central cut of both models, we can calculate the lower and upper bounds on the bisection width of each one of them. We will start by the lower and upper bounds on the bisection width of Model A and, then, we will calculate both bounds for Model B. We first present the following lemma for the lower bound on the bisection width of a Model A BCube.

[6] Note that in the computation of the congestion, the switch is not considered a node of the graph.

Lemma 3. *The bisection width of a Model A d-dimensional BCube, $BCA_k^{(d)}$, is lower bounded by $\frac{k^{d+1}}{4(k-1)}$ if k is even, and by $\frac{k+1}{4}\frac{k^d-1}{k-1}$ if k is odd.*

After presenting the lower bound on the bisection width of a Model A d-dimensional BCube, we follow with the upper bound.

Lemma 4. *The bisection width of a Model A d-dimensional BCube, $BCA_k^{(d)}$, is upper bounded by $\frac{k^d}{2}$ if k is even, and by $\frac{k^d-1}{2}$ if k is odd.*

Now, from the combination of Lemma 3 and Lemma 4 we can state Theorem 7:

Theorem 7. *The value of the bisection width of a Model A d-dimensional BCube, $BCA_k^{(d)}$, is in the interval $[\frac{k^{d+1}}{4(k-1)}, \frac{k^d}{2}]$ if k is even, and in the interval $[\frac{k+1}{4}\frac{k^d-1}{k-1}, \frac{k^d-1}{2}]$ if k is odd.*

Corollary 2. *The bisection bandwidth of a Model A d-dimensional BCube satisfies,*

$$BBW(BCA_k^{(d)}) \in \begin{cases} [2T\frac{k^{d+1}}{4(k-1)}, 2T\frac{k^d}{2}] & \text{if } k \text{ is even} \\ [2T\frac{k+1}{4}\frac{k^d-1}{k-1}, 2T\frac{k^d-1}{2}] & \text{if } k \text{ is odd.} \end{cases}$$

Let us calculate now the bounds of a Model B d-dimensional BCube. As we did with Model A, we present the following two lemmas for both the lower and upper bounds.

Lemma 5. *The bisection width of a Model B d-dimensional BCube, $BCB_k^{(d)}$, is lower bounded by $\frac{k^d}{2(k-1)}$ if k is even, and by $\frac{k+1}{2k}\frac{k^d-1}{k-1}$ if k is odd.*

Lemma 6. *The bisection width of a Model B d-dimensional BCube, $BCB_k^{(d)}$, is upper bounded by $\frac{k^d-1}{k-1}$.*

Combining the previous lemmas we can state the following theorem.

Theorem 8. *The value of the bisection width of a Model B d-dimensional BCube, $BCB_k^{(d)}$, is in the interval $[\frac{k^d}{2(k-1)}, \frac{1-k^d}{1-k}]$ if k is even, and in the interval $[\frac{k+1}{2k}\frac{k^d-1}{k-1}, \frac{k^d-1}{k-1}]$ if k is odd.*

Corollary 3. *The bisection bandwidth of a Model B d-dimensional BCube satisfies,*

$$BBW(BCB_k^{(d)}) \in \begin{cases} [s\frac{k^d}{2(k-1)}, s\frac{1-k^d}{1-k}] & \text{if } k \text{ is even} \\ [s\frac{k+1}{2k}\frac{k^d-1}{k-1}, s\frac{k^d-1}{k-1}] & \text{if } k \text{ is odd.} \end{cases}$$

References

1. Arjona Aroca, J., Fernández Anta, A.: Bisection (Band)Width of Product Networks with Application to Data Centers. ArXiv e-prints, CoRR abs/1202.6291 (Feb 2012)
2. Azizoğlu, M.C., Eğecioğlu, Ö.: The isoperimetric number and the bisection width of generalized cylinders. Electronic Notes in Discrete Mathematics 11, 53–62 (2002)
3. Azizoğlu, M.C., Eğecioğlu, Ö.: Extremal sets minimizing dimension-normalized boundary in hamming graphs. SIAM J. Discrete Math. 17(2), 219–236 (2003)

4. Azizoğlu, M.C., Eğecioğlu, Ö.: The bisection width and the isoperimetric number of arrays. Discrete Applied Mathematics 138(1-2), 3–12 (2004)
5. Dally, W., Towles, B.: Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003)
6. Dally, W.J.: Performance analysis of k-ary n-cube interconnection networks. IEEE Trans. Computers 39(6), 775–785 (1990)
7. Duato, J., Yalamanchili, S., Lionel, N.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)
8. Efe, K., Feng, G.L.: A proof for bisection width of grids. World Academy of Science, Engineering and Technology 27(31), 172 – 177 (2007)
9. Efe, K., Fernández, A.: Products of networks with logarithmic diameter and fixed degree. IEEE Trans. Parallel Distrib. Syst. 6(9), 963–975 (1995)
10. Efe, K., Fernández, A.: Mesh-connected trees: A bridge between grids and meshes of trees. IEEE Trans. Parallel Distrib. Syst. 7(12), 1281–1291 (1996)
11. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., Lu, S.: Bcube: a high performance, server-centric network architecture for modular data centers. In: SIGCOMM. pp. 63–74. ACM (2009)
12. Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y., Lu, S.: Dcell: a scalable and fault-tolerant network structure for data centers. In: SIGCOMM. pp. 75–86. ACM (2008)
13. Jayasimha, D.N., Zafar, B., Hoskote, Y.: On chip interconnection networks why they are different and how to compare them. Intel (2006)
14. Leighton, F.T.: Introduction to parallel algorithms and architectures: array, trees, hypercubes. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992)
15. Mirza-Aghatabar, M., Koohi, S., Hessabi, S., Pedram, M.: An empirical investigation of mesh and torus noc topologies under different routing algorithms and traffic models. In: 10th Euromicro DSD. pp. 19–26. IEEE Computer Society, Washington, DC, USA (2007)
16. Nakano, K.: Linear layout of generalized hypercubes. Int. J. Found. Comput. Sci. 14(1), 137–156 (2003)
17. Pan, Y., Zheng, S.Q., Li, K., Shen, H.: An improved generalization of mesh-connected computers with multiple buses. IEEE Trans. Parallel Distrib. Syst. 12, 293–305 (March 2001)
18. Rolim, J.D.P., Sýkora, O., Vrto, I.: Optimal cutwidths and bisection widths of 2- and 3-dimensional meshes. In: WG. LNCS, vol. 1017, pp. 252–264. Springer (1995)
19. Salminen, E., Kulmala, A., H, T.D.: Survey of network-on-chip proposals. Simulation (March), 1–13 (2008)
20. Youssef, A.: Cartesian product networks. In: ICPP (1). pp. 684–685 (1991)
21. Youssef, A.: Design and analysis of product networks. In: Frontiers'95. pp. 521–. IEEE Computer Society, Washington, DC, USA (1995)
22. Zydek, D., Selvaraj, H.: Fast and efficient processor allocation algorithm for torus-based chip multiprocessors. Comput. Electr. Eng. 37, 91–105 (January 2011)