J Supercomput (2014) 69:121–138
DOI 10.1007/s11227-014-1130-0

Some properties and algorithms for the hyper-torus network

Jong-Seok Kim · Sung Won Kim · Ke Qiu · Hyeong-Ok Lee

Published online: 27 February 2014

Abstract The hyper-torus network based on a three-dimensional hypercube was introduced recently. The hyper-torus $QT(m, n)$ performs better than mesh type networks with a similar number of nodes in terms of the network cost. In this paper, we prove that if n is even, the bisection width of $QT(m, n)$ is $6n$, whereas it is $6n + 2$ if it is odd. Second, we show that $QT(m, n)$ contains a Hamiltonian cycle. In addition, its one-to-all and all-to-all broadcasting algorithms are introduced. All of these broadcasting algorithms are asymptotically optimal.

Keywords Interconnection network · Hyper-torus · Bisection width · Hamiltonian cycle · Broadcasting

1 Introduction

To enhance the performance of supercomputers within a large-scale parallel computing system, many researchers have focused on areas, such as CPU development, intercon-

J.-S. Kim
Department of Computer Science, University of Rochester, Rochester, NY 14627, USA

S. W. Kim
Department of Information and Communication Engineering,
Yeungnam University, Gyeongsan, Gyeongbuk 712-749, South Korea

K. Qiu
Department of Computer science, Brock University, Saint Catharines, ON L2S 3A1, Canada

H.-O. Lee (✉)
Department of Computer Education, Sunchon National University,
Sunchon, Chonnam 540-742, South Korea
e-mail: oklee@scnu.ac.kr; oklee@sunchon.ac.kr

nection network, data transfer methodology and algorithm development. To maximize the performance of supercomputers, the underlying interconnection network plays a pivotal role. Many topologies have been introduced as interconnection networks for parallel computing systems, such as torus, hypercube and star graph.

An interconnection network consists of a set of processors, local memory and communication links for data transmission between processors. Such an interconnection network can be modeled as a graph $G = (V, E)$. Each processor, P_i, is a component of V, which is a set of nodes, and two processors, P_i and P_j, are connected via a communication link (P_i, P_j). When modeling an interconnection network as a graph, a processor is considered as a node of V and a communication link is considered as an edge of E. The number of nodes adjacent to node, P_i, is the degree of the node.

Based on their topologies, most interconnection networks can be roughly classified into three families: mesh family, hypercube family, and star graph family [1]. The mesh family includes the torus [2], honeycomb mesh [3], diagonal mesh [4] and hexagonal mesh [5]. The hypercube family includes cube-connected cycles, folded hypercubes, twisted cubes, butterfly, etc. [6,7]. The star family includes the star graph, and transposition graphs, etc. [1]. The mesh family, which is commonly used in the VLSI circuit design area, is widely used and commercialized as various systems as it is easy to extend its connection network. A low-dimensional mesh is easy to design and useful in terms of algorithms; therefore, it is commonly used as a connection network for parallel computers.

Recently, the hyper-torus network, which is based on the hypercube and whose degree is fixed at 4, was introduced in [8]. The hyper-torus network performs better than the mesh family of networks with a similar number of nodes in terms of the network cost (defined as the product of its degree and diameter [8]). Comparisons among the hyper-torus and other mesh/torus type of networks were provided in [8]. In addition to having a smaller network cost, the hyper-torus is highly scalable, symmetric, and has a small diameter [8].

One of the metrics for evaluating interconnection networks is the bisection width (to be defined later). It is an important parameter for processor communications [9]. It is also related to edge congestion in routing algorithms [10]. The decision version of the problem of finding the bisection width is known to be NP-complete [11,12]. Ki et al. [8] analyzed the bisection width of the hyper-torus $QT(m, n)$. However, it only considered the case when n is even and did not prove that the analyzed bisection width is a minimum. This paper will prove that when m is even, the bisection width of $QT(m, n)$ is $6n$ whereas it is $6n + 2$ if it is odd. Also, we will show that $QT(m, n)$ contains a Hamiltonian cycle. In addition, one-to-all and all-to-all broadcasting algorithms of $QT(m, n)$ will be introduced, and their broadcasting times will be analyzed.

This paper is organized as follows. Section 2 gives the definition of $QT(m, n)$ and its fundamental characteristics. In Sect. 3, we analyze the bisection width of $QT(m, n)$ and Sect. 4 shows that $QT(m, n)$ contains a Hamiltonian cycle. Section 5 introduces one-to-all and all-to-all broadcasting algorithms and analyzes their running times. Section 6 concludes the paper.

2 Preliminaries

The hypercube is one of the well-known interconnection networks and it has excellent characteristics. The three-dimensional hypercube Q_3 is a network that consists of 8 nodes and 12 edges; its diameter and degree are 3 and it is both node symmetric and edge symmetric.

Figure 1 shows a 3-dimensional hypercube Q_3.

We denote the set of nodes for hypercube Q_3 as V and its set of edges as E. The definition of Q_3 is as follows where % represents mod operation and $Z_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ a cyclic group of order 8 under addition modulo 8. In other words, two nodes i and j, $0 \leq i, j \leq 7$, are connected if their corresponding binary representations differ in exactly one of the three positions.

Definition 1 $V = \{z | z \in Z_8\}$

$$E = \{(z_1, z_2) | z_1 - z_2 \equiv 1 \text{or}\, z_1 - z_2 \equiv 2 \text{or}\, z_1 - z_2 \equiv 4 (\%8)\}$$

A recent study has proposed a hyper-torus network [8]. Basically, this network combines an m by n torus with a 3-dimensional hypercube Q_3 and is denoted as $QT(m, n)$. This notation is first used in [8] that proposed the network. Apparently, Q represents the hypercube while T represents the torus, and $QT(m, n)$ contains mn basic modules each of which is a 3-dimensional hypercube Q_3. Ki et al. [8] analyzed the diameter and the bisection width of the hyper-torus network and proposed a simple routing algorithm. The basic module address of the hyper-torus $QT(m, n)$ $(m, n \geq 2)$ is represented as (x, y) and its node address is (x, y, z). x represents X axis coordinate of the basic module, y is Y axis coordinate of the basic module and z is the node address in the basic module Q_3. We will represent the node set of $QT(m, n)$ as V_{qt} and its edge set as E_{qt}. Therefore, the definition of the hyper-torus $QT(m, n)$ is as follows:

Definition 2 $V_{qt} = \{(x, y, z) 0 \leq x < m, 0 \leq y < n, z \in Z_8\}$

The edges of $QT(m, n)$ consist of internal edges, E_{in}, and external edges, E_{ex}. An internal edge is an edge that connects the nodes in the same basic module Q_3. An

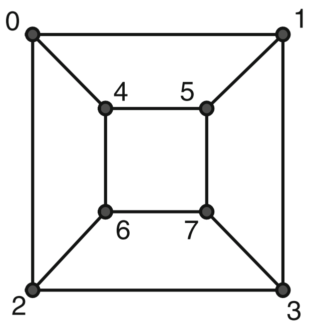


Fig. 1 3-Dimensional hypercube Q_3

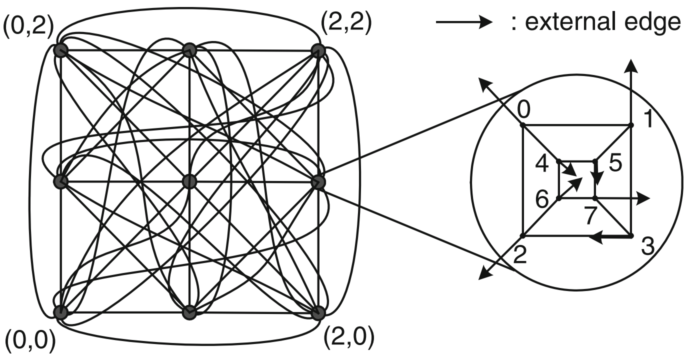


Fig. 2 $QT(3,3)$

external edge is an edge that links nodes that are within the different basic modules. The definition of an external edge can be expressed as:

Definition 3 $E_{ex} = \{$
vertical edge: $((x, y, 1), (x, (y+1)\%n, 5)), ((x, y, 5), (x, (y-1+n)\%n, 1))$
horizontal edge: $((x, y, 7), ((x+1)\%m, y, 3)), ((x, y, 3), ((x-1+m)\%m, y, 7))$
oblique line edge: $((x, y, 6), ((x+1)\%m, (y+1)\%n, 2)), ((x, y, 2), ((x-1+m)\%m, (y-1+n)\%n, 6))$
reverse oblique line edge: $((x, y, 0), ((x-1+m)\%m, (y+1)\%n, 4)), ((x, y, 4), ((x+1)\%m, (y-1+n)\%n, 0))\}$

Figure 2 shows $QT(3,3)$ with Q_3 as a basic module. In Figs. 3, 4 and 5, the oblique line edges and reverse oblique line edges are omitted.

It was shown in [8] that for $QT(m,n)$, its diameter is $2\max(\lfloor m/2\rfloor, \lfloor n/2\rfloor) + 4$ which is $O(m+n)$.

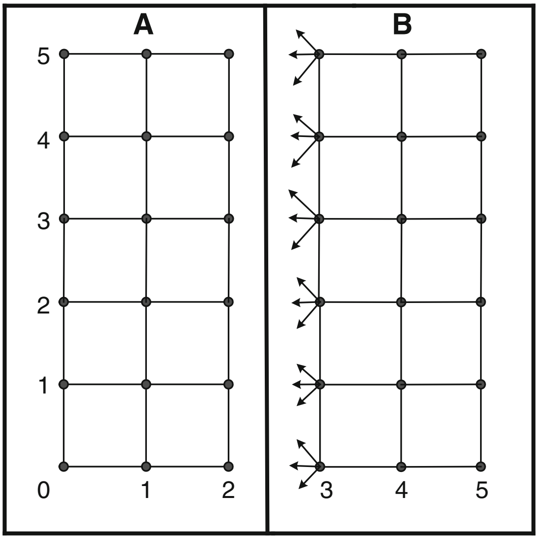
3 The bisection width of $QT(m, n)$

The bisection width is the minimum number of edges that need to be removed to segregate one connected network into two networks [9,13–16]. These two segregated networks would have the same number of nodes or have one node difference with each other. A smaller bisection width is less desirable than a larger bisection width, because a small number of edge faults can disconnect the network. Although Ki et al. [8] studied the bisection width when n is even, they did not prove that the result is a minimum. Therefore, in this section we analyze the bisection width of $QT(m,n)$. Although the bisection width of $QT(m,n)$ should be studied under both conditions, $m \geq n$ and $n \geq m$, this section only considers the $m \geq n$ condition because both results from the two conditions are similar.

Theorem 1 *The bisection width of $QT(m,n)$ is $6n$ when m is even and $6n+2$ when m is odd.*

Proof We will prove two cases based on the condition of m; when m is odd and even.

Fig. 3 Bisection width of $QT(6, 6)$



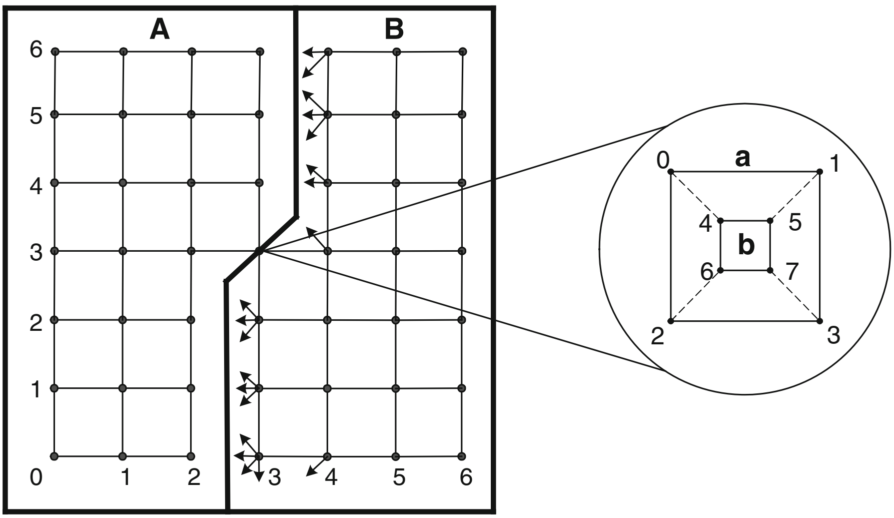


Fig. 4 Bisection width of $QT(7, 7)$

1. When m is even:
   When removing the external edges that connect the basic modules $(0, i)$ and $(m - 1, i)$ and when removing external edges that connect basic modules $(\frac{m}{2} - 1, i)$ and $(\frac{m}{2}, i)$, the original network is broken into two networks with the same number of nodes $(0 \leq i \leq n - 1)$. The number of external edges that connect one basic module in partition B and its adjacent basic module in partition A is 3 (oblique line, horizontal, reverse oblique line edge). Because the number of such i's is n, the number of external edges that connect the basic modules $(0, i)$ and $(m - 1, i)$ is $3n$ and the number of external edges that connect basic modules $(\frac{m}{2} - 1, i)$ and $(\frac{m}{2}, i)$

is $3n$. Therefore, the bisection width that segregates the original network into two networks of the same size is $6n$. Figure 3 presents the bisection width of $QT(6, 6)$. The $3n$ external edges that connect the basic modules $(0, i)$ and $(m - 1, i)$ and should be removed $(0 \le i \le n - 1)$ are disregarded, and the remaining external edges that should be removed are presented as arrows in Fig. 3. Consequently, the bisection width of $QT(6, 6)$ is 36.

2. When m is odd: the following two cases will be proven based on the condition of n; when n is odd and even.

   (a) When n is odd:
   As the number of basic modules could be any odd number, the original network can be divided into two groups by eliminating four edges that consist of the basic module $\left(\lfloor\frac{m}{2}\rfloor, \lfloor\frac{m}{2}\rfloor\right)$. Let a be a group that includes nodes 0, 1, 2 and 4, and b a group that contains nodes 4, 5, 6 and 7. This will be applied to the remaining basic modules, except $\left(\lfloor\frac{m}{2}\rfloor, \lfloor\frac{n}{2}\rfloor\right)$, in networks A and B as follows. When $i > \lfloor\frac{n}{2}\rfloor$, the basic module (j, i) is contained in network A and the basic module (k, i) is taken in network B $\left(0 \le j \le \lfloor\frac{m}{2}\rfloor, \lfloor\frac{m}{2}\rfloor < k < m\right)$. When $i \le \lfloor\frac{n}{2}\rfloor$, the basic module (j', i) is contained in network A and the basic module (k', i) is going to network B $\left(0 \le j < \lfloor\frac{m}{2}\rfloor, \lfloor\frac{m}{2}\rfloor \le k < m\right)$. Group a is included in network A and group b is included in network B. As the number of edges that are to be removed to divide $\left(\lfloor\frac{m}{2}\rfloor, \lfloor\frac{n}{2}\rfloor\right)$ into groups a and b is 4 (edges that connect 0 and 4, 1 and 5, 2 and 6 and 3 and 7), the number of edges to be removed within $\left(\lfloor\frac{m}{2}\rfloor, \lfloor\frac{n}{2}\rfloor\right)$ is 4. The number of edges to be removed within the module $\left(\lfloor\frac{m}{2}\rfloor + 1, n - 1\right)$, which is contained in network B, is 2, and the number of edges that need to be removed within module $\left(\lfloor\frac{m}{2}\rfloor + 1, \lfloor\frac{n}{2}\rfloor + 1\right)$ is 2. The number of edges to be removed within module $\left(\lfloor\frac{m}{2}\rfloor + 1, \lfloor\frac{n}{2}\rfloor\right)$ is 1 and the number of edges to be removed within module $\left(\lfloor\frac{m}{2}\rfloor + 1, 0\right)$ is also 1. Within module $\left(\lfloor\frac{m}{2}\rfloor, 0\right)$, the number of edges to be removed is 4 and the number of edges to be removed within the rest basic modules is 3. Therefore, in this case, bisection width is $6n + 2$. Figure 4 shows the bisection width of $QT(7, 7)$. The $3n$ external edges to-be-removed that connect the basic modules $(0, i)$ and $(m - 1, i)$ are not presented in Fig. 4 $(0 \le i \le n - 1)$. The should-be-removed edges that divide the network into groups a and b are presented as dotted lines and other external edges that are to be removed are presented as arrows. Figure 4 shows that the bisection width of $QT(7, 7)$ is 44.

   (b) When n is even:
   The network $QT(m, n)$ is divided into two networks as follows. When $i \ge \frac{n}{2}$, the basic module (j, i) and (k, i) is contained in network A and B, respectively, $\left(0 \le j \le \frac{m}{2}, \frac{m}{2} < k < n\right)$. When $i < \frac{n}{2}$, the basic module (j', i) and (k', i) is included in network A and B, respectively, $\left(0 \le j' \le \frac{m}{2}, \frac{m}{2} < k' < n\right)$. The number of the to-be-removed edges within module $\left(\frac{m}{2} + 1, n - 1\right)$ in network B is 2 and the number of to-be-removed edges within $\left(\frac{m}{2} + 1, \frac{n}{2}\right)$ is 2. The number of edges to be removed within $\left(\frac{m}{2} + 1, \frac{n}{2} - 1\right)$ is 1 and the number of edges to be removed within $\left(\frac{m}{2} + 1, 0\right)$ is 1. The number of to-be-removed

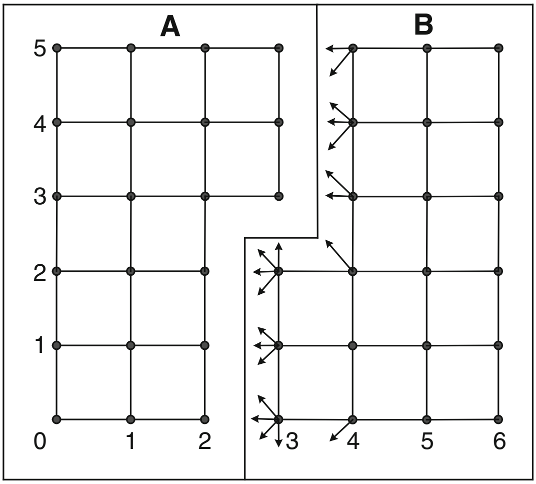


Fig. 5 Bisection width of $QT(7, 6)$

edges within $\left(\frac{m}{2}, 0\right)$ is 4 and the number of to-be-removed edges within the other basic modules that connect networks A and B is 3. Therefore, in this case, the bisection width is $6n + 2$. Figure 5 shows the bisection width of $QT(7, 6)$. The $3n$ external edges that connect basic modules $(0, i)$ and $(m - 1, i)$ are not presented $(0 \leq i \leq n - 1)$ and other external edges that should be removed are presented as arrows. Figure 5 shows that the bisection width of $QT(7, 6)$ is 38.

Now, we will show that the bisection width just obtained is the minimum.

For Case 1, the easiest way to divide network $QT(m, n)$ into networks A and B, which contain the same number of nodes, is to divide $QT(m, n)$ using the proposed bisection width from Case 1, where m and n are even. In this case, the number of removed edges per basic module is 3. If the number of removed edges within any basic module is less than 3, $QT(m, n)$ is not divided into networks A and B but is connected as a single network from Fig. 3. This shows that the minimum number of edges that need to be removed per basic module to divide $QT(m, n)$ into two networks, A and B, is 3. For that reason, the bisection width that is to divide $QT(m, n)$ into two networks is $6n$, as proposed in Case 1.

For Case 2(b), the easiest way to divide $QT(m, n)$ into two networks, A and B, which include the same number of nodes, is to divide the network using the method that Case 2(b) proposed (n is even and m is odd). As shown in Fig. 5, although the number of should-be-removed edges is 3 per module when dividing the modules at Y axis level, the number of should-be-removed edges increases by 1 for some modules in the intersection column when dividing the modules as a grid format. The modules that are divided as grid format are called grid modules. The external edges of the grid modules that need to remove one additional edge in Fig. 5 are ((3, 2, 1), (3, 3, 5)) or ((3, 5, 1), (3, 0, 5)). Therefore, the bisection width decreases with decreasing number of grid modules along the partition axis. The number of grid modules along the partition axis that needs to divide $QT(m, n)$ into two networks is always even. Because

m is odd in Case 2(b), at least two grid modules are needed to divide $QT(m,n)$. As the number of should-be-removed external edges that are connected at Y axis level is $6n$ and one more external edge within two grid modules should be deleted, the proposed bisection width, $6n + 2$, is the minimum. The methodology for proving that Case 2(a) is the minimum is similar to the case when n is even and m is odd.

Now, we will consider a method to remove the edges that connect unit nodes to divide $QT(m,n)$. In Fig. 3, assume that node $u = (2, 3, z_1)$ in network A is contained in network B and node $v = (3, 1, z_2)$ in network B is in network A. The number of edges that need to be removed will be determined using two cases that are dependent on the connection states of node u's external edge. First, consider the case that node u is connected to network B via external edge. In this case, three internal edges that connect node u and the basic module, in which node u is contained, need to be removed. Then, the two external edges that connect network B and the basic module containing node u should be deleted. Second, consider the case that node u is connected to network A via an external edge. In this case, it is important to remove the three internal edges that connect node u and the basic module containing node u. Then, the three external edges that connect network B and the basic module containing node u should be deleted. Because the bisection width will gain a minimum value when node u is connected to network B via an external edge and node v is connected to network A via external edge, bisection with in this case is $3n + 3(n - 2) + 5 + 5 = 6n + 4$, which is larger than the proposed bisection width in this theorem.

To divide a connection network, $QT(m,n)$, into two distinct connection networks, A and B, the edge that connects the two unit nodes within a single basic module is removed. In Fig. 3, assume that nodes $u = (2, 3, z_1)$ and $u' = (2, 3, z_1')$ within connection network A are also within connection network B and nodes $v = (3, 1, z_2)$ and $v' = (3, 1, z_2')$ within connection network B are also located in connection network A. The number of edges that need to be removed will vary according to the connection states between nodes u and u' and the states between the two nodes, u and u', and connection network B. The case that the number of removable edges is a minimum is that nodes u and u' are adjacent and these two nodes are connected to connection network B and its external edge. Therefore, the bisection width is $3n+3(n-2)+5+5 = 6n+4$, which shows that it is larger than the bisection width obtained in this theorem.

Consider removing the edge that connects three unit nodes within a single basic module to divide a connection network $QT(m,n)$ into connection networks A and B. Again, as shown in Fig. 3, assume that nodes $u = (2, 3, z_1)$, $u' = (2, 3, z_1')$ and $u^* = (2, 3, z_1^*)$ within connection network A are included in connection network B and node $v = (3, 1, z_2)$ within connection network B is located in connection network A. Then, the number of edges that should be removed will vary according to the connection states among nodes u, u' and u^* and the connection states among these three nodes and connection network B. The condition that minimizes the edges that should be removed is that nodes u, u' and u^* are adjacent to each other and these nodes are connected to connection network B and the external edge. Therefore, its bisection width, $3n + 3(n - 2) + 5 + 5 = 6n + 4$, is larger than the proposed bisection width in this theorem.

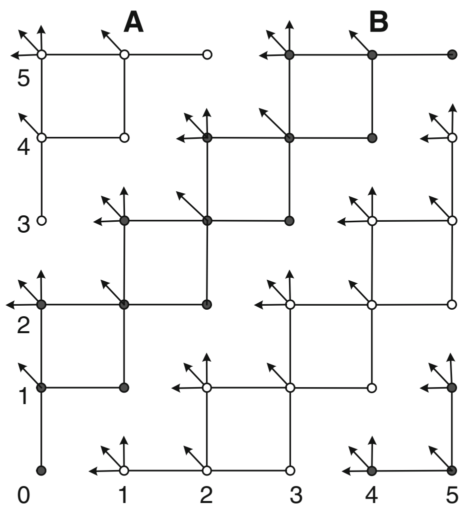


Fig. 6 Bisection width of $QT(6, 6)$ using diagonal partition

A method to remove the edge that connects four nodes within a basic module to divide the connection network $QT(m, n)$ into connection A and B is to make the number of edges, which are to be removed using the proposed method in Case 2(a), a minimum. Therefore, its bisection width, i.e. $3n + 3(n - 2) + 5 + 5 = 6n + 4$, is larger than that in this theorem.

According to the cases shown thus far, it was verified that the bisection width will be minimized when the number of unit nodes, which are divided when dividing a connection network and the number of grid modules, are smaller.

Now, consider the other partition problem. The partition such as Fig. 3 is called vertical partition, the partition included grid module such as Fig. 5 is called grid partition, and the partition such as Fig. 6 is called diagonal partition. In Fig. 6, white node is the node in group A and black node is the node in group B, and external edges that are to be removed are presented as arrows.

We proved that the bisection width of $QT(m, n)$ using grid partition is larger than that using vertical partition at the above proof. And we can show that the bisection width of $QT(m, n)$ using diagonal partition is larger than that using vertical partition through a comparison between Figs. 3 and 6.

Therefore, the proposed bisection width is the minimum. □

4 Hamiltonian cycle of $QT(m, n)$

A Hamiltonian path in a connection network is a path that passes all the nodes within the network only once. A Hamiltonian cycle is a cycle that traverses all the nodes precisely once [17,18]. If there is a Hamiltonian path or Hamiltonian cycle in a connection network, the network can be a pipeline that makes parallel processing easy because it becomes a linear array or ring. Q_3 is a connection network that has a Hamiltonian cycle. In Property 1, several Hamiltonian paths within Q_3 will be shown.

Table 1 Hamiltonian cycle algorithm in $QT(m, n)$

$(0, n-1) - (1, n-1) - \cdots - (m-2, n-1) - (m-1, n-1)$
$-(0, n-2) - (1, n-2) - \cdots - (m-2, n-2) - (m-1, n-2)$
$-(0, n-3) - (1, n-3) - \cdots - (m-2, n-3) - (m-1, n-3)$
$- \cdots - (0, 1) - (1, 1) - \cdots - (m-2, 1) - (m-1, 1)$
$-(0, 0) - (1, 0) - \cdots - (m-2, 0) - (m-1, 0) - (0, n-1)$

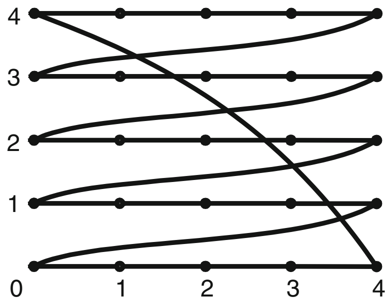


Fig. 7 Hamiltonian cycle in $QT(5, 5)$

Property 1 Some Hamiltonian paths in Q_3 are as follows:
Case 1) $0 \longrightarrow 1 \longrightarrow 3 \longrightarrow 2 \longrightarrow 6 \longrightarrow 4 \longrightarrow 5 \longrightarrow 7$
Case 2) $3 \longrightarrow 2 \longrightarrow 0 \longrightarrow 1 \longrightarrow 5 \longrightarrow 4 \longrightarrow 6 \longrightarrow 7$
Case 3) $3 \longrightarrow 2 \longrightarrow 0 \longrightarrow 1 \longrightarrow 5 \longrightarrow 7 \longrightarrow 6 \longrightarrow 4$

As the Hamiltonian paths in Q_3 in Property 1 are shown, the following will show that there is a Hamiltonian cycle among the basic modules of $QT(m, n)$. The Hamiltonian cycle construction in $QT(m, n)$ is given in Table 1.

In Property 1, establishing a Hamiltonian path for module $(0, p)$ depends on case 1) and for module $(m-1, p)$ on case 3) $(0 \le p \le n-1)$. The remaining modules depend on case 2) and the paths for modules $(m-1, q)$ and $(0, q')$ are established by the reversing oblique line edge $(0 \le q \le n-1, q' = q-1$, if $q = 0$ then $q' = n-1)$. Therefore, $QT(m, n)$ contains a Hamiltonian cycle from the proposed Hamiltonian cycle construction:

Theorem 2 *$QT(m, n)$ is Hamiltonian.*

Figure 7 shows a Hamiltonian cycle in $QT(5, 5)$.

5 Broadcasting in $QT(m, n)$

Broadcasting is a basic data communication method for interconnection networks [19–24]. There are two types of broadcasting: one-to-all broadcasting and all-to-all broadcasting. One-to-all type is to transmit a message from one node to all other nodes and all-to-all type is to transmit a message from all the nodes that have messages to other nodes. Single-port telecommunication is to transmit messages from one node that contains the messages to one adjacent node only and all-port telecommunication is to transmit messages from one node to all adjacent nodes within a time of unit. The former is called single link available (SLA) and the latter is multiple link available (MLA). Since the diameter of the $QT(m, n)$ is $O(m+n)$, any broadcasting algorithm, whether on the single-port or all-port model, has a lower bound of $\Omega(m+n)$. Broadcasting in $QT(m, n)$ can be done by the two cases—whether $m \geq n$ or $n \leq m$; In this section, we only consider the case when $m \geq n$ since the algorithm for the other case is similar. The broadcasting time is measured in terms of the total number of hops (links) required.

5.1 One-to-all broadcasting in $QT(m, n)$

One-to-all broadcasting for Q_3, the basic module of $QT(m, n)$, and its broadcasting time were proved in [21].

Lemma 1 *One-to-all broadcasting time for 3-dimensional hypercube, Q_3, is 3 in both SLA and MLA.*

With the one-to-all broadcasting and its time for Q_3, the basic module of $QT(m, n)$, proposed in [21,22], we will propose one-to-all broadcasting for $QT(m, n)$. $QT(m, n)$ is node symmetric [8]. Therefore, we can assume that M_0 is $(\lfloor \frac{x}{2} \rfloor, \lfloor \frac{y}{2} \rfloor)$ in one-to-all broadcasting algorithm. One-to-all broadcasting algorithm contains conditions as follows where

M_0: basic module that source nodes are within
M_f: basic module where $x = 0$ or $x = m - 1$ or $y = 0$ or $y = n - 1$
M_j: basic module that received messages (except M_0 and M_f)
$u_i = (x_0, y_0, z_1)$: internal nodes of M_0 $(0 \leq z_1 \leq 7, 0 \leq i \leq 7)$
$v_i = (x, y, z_1)$: internal nodes of M_j $(0 \leq z_1 \leq 7, 0 \leq i \leq 7)$
w_0: a node that is connected via internal node v_1 of M_j and external edge(s) by Condition 2
$\longrightarrow$: message transfer from internal edge
$\Longrightarrow$: message transfer from external edge

Condition 1 *all the internal nodes of M_0 will receive messages in the following order, then execute 1st step of one-to-all broadcasting algorithm.*

1. $u_0 = (x, y, z_1) \longrightarrow u_1 = (x, y, (z_1 + 4)\%8)$
2. $u_0 = (x, y, z_1) \longrightarrow u_2 = (x, y, (z_1 + 1)\%8)$
   $u_1 = (x, y, (z_1 + 4)\%8) \longrightarrow u_3 = (x, y, (z_1 + 5)\%8)$
3. $u_0 = (x, y, z_1) \longrightarrow u_4 = (x, y, (z_1 + 2)\%8)$
   $u_1 = (x, y, (z_1 + 4)\%8) \longrightarrow u_5 = (x, y, (z_1 + 6)\%8)$
   $u_2 = (x, y, (z_1 + 1)\%8) \longrightarrow u_6 = (x, y, (z_1 + 3)\%8)$
   $u_3 = (x, y, (z_1 + 5)\%8) \longrightarrow u_7 = (x, y, (z_1 + 7)\%8)$

Condition 2 *Except M_0 and M_f, modules that receive messages using one-to-all broadcasting algorithm will transfer messages to all the nodes within the module in the following order.*

Table 2 One-to-all broadcasting algorithm of $QT(m, n)$

[Step 1] the source module transfers messages to eight adjacent nodes using external edge(s)
[Step 2] modules that receive messages via vertical (horizontal, oblique line, reverse oblique line) edges will transfer messages to adjacent modules that do not have messages via vertical (horizontal, oblique line, reverse oblique line) edges
[Step 3] modules that receive messages via oblique line (reverse oblique line) edges will send messages to adjacent modules that have no messages from vertical or horizontal edges
[Step 4] Until the messages reach to all the modules in $QT(m, n)$, repeat Steps 2 and 3

1. $v_0 = (x, y, z_1) \longrightarrow v_1 = (x, y, (z_1 + 4)\%8)$
2. $v_1 = (x, y, (z_1 + 4)\%8) \Longrightarrow w_0 = (x', y', (z_1 + 4)\%8)$
$v_0 = (x, y, z_1) \longrightarrow v_2 = (x, y, (z_1 + 1)\%8)$
3. $v_0 = (x, y, z_1) \longrightarrow v_3 = (x, y, (z_1 + 2)\%8)$
$v_1 = (x, y, (z_1 + 4)\%8) \longrightarrow v_4 = (x, y, (z_1 + 6)\%8)$
$v_2 = (x, y, (z_1 + 1)\%8) \longrightarrow v_5 = (x, y, (z_1 + 5)\%8)$
4. $v_2 = (x, y, (z_1 + 1)\%8) \longrightarrow v_6 = (x, y, (z_1 + 3)\%8)$
$v_5 = (x, y, (z_1 - 3)\%8) \longrightarrow v_7 = (x, y, (z_1 + 7)\%8)$

Condition 3 *Modules that receive messages via oblique line or reverse oblique line edges will transfer messages to adjacent modules that have no message from vertical or horizontal edges after executing Condition 2. However, when there are no adjacent modules to transfer from oblique line or reverse oblique line edges among module M_f, it sends messages to all the nodes within the module as in Condition 1, then sends messages to adjacent modules that do not have messages from vertical or horizontal edges. Modules (1, 15) and (15, 1) in Fig. 10 are in this case.*

Condition 4 *After executing one-to-all broadcasting algorithm, all the nodes within M_f will get messages by Condition 1.*

Table 2 shows one-to-all broadcasting algorithm of $QT(m, n)$.

We will calculate the broadcasting time for the one-to-all broadcasting algorithm using SLA.

Case 1) broadcasting time for modules that receive messages via vertical (horizontal, oblique line, reverse oblique line) edges only:
From Lemma 1, broadcasting time for the inside of the source module is 3. From the source module to the destination module, the maximum broadcasting time via vertical (horizontal, oblique line, reverse oblique line) edges is $\lfloor \frac{m}{2} \rfloor - 1$. For the broadcasting from the source module to the destination module, the maximum broadcasting time via internal edges of each module is $\lfloor \frac{m}{2} \rfloor - 2$. From Lemma 1, broadcasting time within the destination module is 3. Therefore, the total broadcasting time is $3 + \lfloor \frac{m}{2} \rfloor - 1 + \lfloor \frac{m}{2} \rfloor - 2 + 3 = 2\lfloor \frac{m}{2} \rfloor + 3$.

Case 2) broadcasting time of modules that receive messages using oblique line (reverse oblique line) edges and vertical edges:
Broadcasting time within the source module is 3 from Lemma 1. The maximum broadcasting time that transfers messages from the source module to the destination

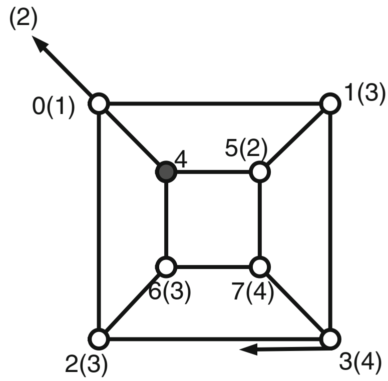


Fig. 8 Message transfer of the basic module at step 3 using SLA

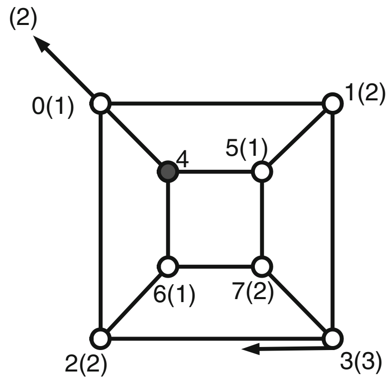


Fig. 9 Message transfer of the basic module at Step 3 using MLA

module using oblique line (reverse oblique line) edges and vertical edges is $\lfloor \frac{m}{2} \rfloor - 1$. The maximum broadcasting time from the source module to the destination module via internal edges of each module is $\lfloor \frac{m}{2} \rfloor + 1$. From Lemma 1, broadcasting time within the destination module is 3. Therefore, broadcasting time of this case is $3 + \lfloor \frac{m}{2} \rfloor - 1 + \lfloor \frac{m}{2} \rfloor + 1 + 3 = 2\lfloor \frac{m}{2} \rfloor + 6$.

The reason that the broadcasting time of Case 2) is increased by 3 compared to that of Case 1) is shown from Fig. 8. Figure 8 shows modules that receive messages via oblique line edges, then transfer the messages to adjacent modules via vertical edges. As shown in Fig. 8, to send messages to adjacent modules via vertical edges, Case 2) needs to utilize internal edges three times more than Case 1). In Fig. 8, (number) illustrates broadcasting time of the inside of the module and • depicts the source node that contains messages from the inside of the module. $\longleftarrow$ represents external edges.

One-to-all broadcasting algorithm using MLA utilizes the same algorithm for the one-to-all broadcasting algorithm using SLA. Broadcasting time using MLA in Step 3 is smaller than that using SLA by 1, as shown in Figs. 8 and 9. Therefore, broadcasting time from one-to-all broadcasting algorithm using MLA is $2\lfloor \frac{m}{2} \rfloor + 5$.

Theorem 3 *Broadcasting time of $QT(m, n)$ from one-to-all broadcasting algorithm using SLA is $2\lfloor \frac{m}{2} \rfloor + 6$ and that of $QT(m, n)$ from one-to-all broadcasting algorithm using MLA is $2\lfloor \frac{m}{2} \rfloor + 5$.*

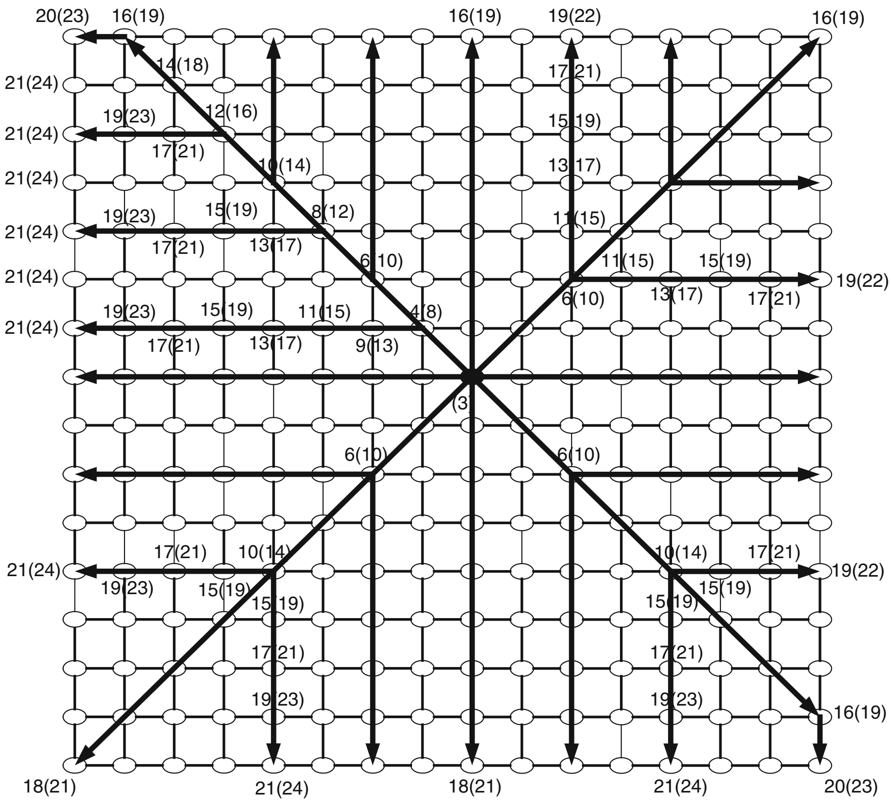


Fig. 10 One-to-all broadcasting of $QT(16, 16)$ using SLA

Figure 10 shows one-to-all broadcasting mechanism of $QT(16, 16)$ using SLA. We do not present oblique line edges or reverse oblique line edges in this Figure. Numbers in Fig. 10 represent broadcasting time for the messages to arrive at its destined cluster and (number) depicts time that all the nodes in the cluster get the messages. And the arrows illustrate processes for message transfer.

5.2 All-to-all broadcasting of $QT(m, n)$

Lemma 2 *All-to-all broadcasting time of 3-dimensional hypercube Q_3 using SLA is 5 and the broadcasting time of Q_3 using MLA is 3.*

Proof In Figure 4, we can see that there are two cycles that are represented as a and b in Q_3. Broadcasting mechanism using SLA is as follows. In parallel, both 4-cycles a and b can do an all-to-all broadcasting in three time units. Now, each node on cycle a has four messages from all four nodes on the cycle and the same can be said for nodes on cycle b. In another step, all nodes on cycle b send their four messages to their corresponding nodes on cycle a. In one more step, all nodes on cycle a send their four

Table 3 All-to-all broadcasting algorithm of $QT(m, n)$ using SLA

[Step 1] Send messages of all nodes within each module to all the other nodes within the same basic module that composes $QT(m, n)$
[Step 2] Using internal edges and vertical edges, send messages to all the basic modules on all the columns in $QT(m, n)$:
$(x, y, 7) - ((x+1)\%m, y, 3) - ((x+1)\%m, y, 7) - ((x+2)\%m, y, 3)$
$-((x+2)\%m, y, 7) - \cdots ((x-1+m)\%m, y, 3) - ((x-1+m)\%m, y, 7)$
[Step 3] Send messages from $(x, y, 7)$ and $(x, y, 3)$ to $(x, y, 1)$ and $(x, y, 5)$
[Step 4] Using internal edges and horizontal edges, send messages to all the basic modules on all the rows in $QT(m, n)$:
$(x, y, 1) - (x, (y+1)\%n, 5) - (x, (y+1)\%n, 1) - (x, (y+2)\%n, 5)$
$-(x, (y+2)\%n, 1) - \cdots - (x, (y-1+n)\%n, 5) - (x, (y-1+n)\%n, 1)$
[Step 5] Send messages of $(x, y, 1)$ and $(x, y, 5)$ within each basic module to all the other nodes within the same basic module that composes $QT(m, n)$

messages to their corresponding nodes on cycle b so that now, all nodes in Q_3 have eight messages and the total time is 5.

Broadcasting mechanism using MLA is as follows. In the first step of broadcasting, all the nodes in a and b cycles send messages to all the adjacent nodes. After this step is repeated once, all the nodes within a and b cycles will have messages from all the nodes within each cycle. In the second step, all the nodes in a cycle and all the nodes in b cycle that are linked via edges will send and get messages each other. Therefore, all-to-all broadcasting time in Q_3 using MLA is 3. □

Table 3 shows all-to-all broadcasting algorithm of $QT(m, n)$ using SLA ($0 \leq x \leq m-1, 0 \leq y \leq n-1$). In this algorithm, if $x = 0$ then $x - 1 = m - 1$ and $y = 0$ then $y - 1 = n - 1$.

All-to-all broadcasting time using SLA is as follows. [Step 1] is a process that sends message of a node within each module to all the other nodes within the same basic module that composes $QT(m, n)$ and its broadcasting time is 5. [Step 2] is a process that sends messages of $(x, y, 7)$ and $(x, y, 3)$ that are nodes of the basic module located in all the columns within $QT(m, n)$ and its broadcasting time is $2m - 2$. [Step 3] is a process that sends messages from $(x, y, 7)$ and $(x, y, 3)$ to $(x, y, 1)$ and $(x, y, 5)$ and its broadcasting time is 1. [Step 4] is a process that sends messages of $(x, y, 1)$ and $(x, y, 5)$ that are located in all the rows within $QT(m, n)$ and its broadcasting time is $2n - 2$. [Step 5] is a process that sends the messages of $(x, y, 1)$ and $(x, y, 5)$ within each basic module to all the other nodes within the same basic module and its broadcasting time is 3. Therefore, all-to-all broadcasting time using SLA is $2m+2n+5$.

Theorem 4 *Broadcasting time of $QT(m, n)$ from all-to-all broadcasting algorithm using SLA is $2m + 2n + 5$.*

Table 4 shows all-to-all broadcasting algorithm of $QT(m, n)$ using MLA ($0 \leq x \leq m-1, 0 \leq y \leq n-1$). In this algorithm, if $x = 0$ then $x - 1 = m - 1$ and $y = 0$ then $y - 1 = n - 1$.

Table 4 All-to-all broadcasting algorithm of $QT(m, n)$ using MLA

[Step 1] Send messages of all nodes within each module to all the other nodes within the same basic module that composes $QT(m, n)$

[Step 2] Using internal edges and horizontal edges, send messages to all the basic modules on all the columns in $QT(m, n)$:

1) $m =$ even:

$(x, y, 7) - ((x+1)\%m, y, 3) - ((x+1)\%m, y, 7) - ((x+2)\%m, y, 3)$
$-((x+2)\%m, y, 7) - \cdots - ((x+\frac{m}{2})\%m, y, 3)$

and

$(x, y, 3) - ((x-1+m)\%m, y, 7) - ((x-1+m)\%m, y, 3) - ((x-2+m)\%m, y, 7)$
$-((x-2+m)\%m, y, 3) - \cdots - ((x+\frac{m}{2})\%m, y, 7)$

2) $m =$ odd:

$(x, y, 7) - ((x+1)\%m, y, 3) - ((x+1)\%m, y, 7) - ((x+2)\%m, y, 3)$
$-((x+2)\%m, y, 7) - \cdots - ((x+\lfloor\frac{m}{2}\rfloor)\%m, y, 3) - ((x+\lfloor\frac{m}{2}\rfloor)\%m, y, 7)$

and

$(x, y, 3) - ((x-1+m)\%m, y, 7) - ((x-1+m)\%m, y, 3) - ((x-2+m)\%m, y, 7)$
$-((x-2+m)\%m, y, 3) - \cdots - ((x+\lceil\frac{m}{2}\rceil)\%m, y, 7) - ((x+\lceil\frac{m}{2}\rceil)\%m, y, 3)$

[Step 3] Send messages from $(x, y, 7)$ and $(x, y, 3)$ to $(x, y, 1)$ and $(x, y, 5)$

[Step 4] Using internal edges and vertical edges, send messages to all the basic modules on all the rows in $QT(m, n)$:

1) $n =$ even:

$(x, y, 1) - (x, (y+1)\%n, 5) - (x, (y+1)\%n, 1) - (x, (y+2)\%n, 5)$
$-(x, (y+2)\%n, 1) - \cdots - (x, (y+\frac{n}{2})\%n, 5)$

and

$(x, y, 5) - (x, (y-1+n)\%n, 1) - (x, (x, (y-1+n)\%n, 5) - (x, (x, (y-2+n)\%n, 1)$
$-(x, (x, (y-2+n)\%n, 5) - \cdots - (x, (y+\frac{n}{2})\%n, 1).$

2) $m =$ odd:

$(x, y, 1) - (x, (y+1)\%n, 5) - (x, (y+1)\%n, 1) - (x, (y+2)\%n, 5)-$
$(x, (y+2)\%n, 1) - \cdots - (x, (y+\lfloor\frac{n}{2}\rfloor)\%n, 5) - (x, (y+\lfloor\frac{n}{2}\rfloor)\%n, 1)$

and

$(x, y, 5) - (x, (y-1+n)\%n, 1) - (x, (y-1+n)\%n, 5) - (x, (y-2+n)\%n, 1)$
$-(x, (y-2+n)\%n, 5) - \cdots - (x, (y+\lceil\frac{n}{2}\rceil)\%n, 1) - (x, (y+\lceil\frac{n}{2}\rceil)\%n, 5)$

[Step 5] Send messages of $(x, y, 1)$ and $(x, y, 5)$ within each basic module to all the other nodes within the same basic module that composes $QT(m, n)$

All-to-all broadcasting time using MLA is as follows. [Step 1] is a process that sends message of a node within each module to all the other nodes within the same basic module that composes $QT(m, n)$ and its broadcasting time is 3. [Step 2] is a process that sends messages of $(x, y, 7)$ and $(x, y, 3)$ that are nodes of the basic module located in all the columns within $QT(m, n)$ and its broadcasting time is $m-1$. [Step 3] is a process that sends messages from $(x, y, 7)$ and $(x, y, 3)$ to $(x, y, 1)$ and $(x, y, 5)$ and its broadcasting time is 1. [Step 4] is a process that sends messages of $(x, y, 1)$

and $(x, y, 5)$ that are located in all the rows within $QT(m, n)$ and its broadcasting time is $n-1$. [Step 5] is a process that sends the messages of $(x, y, 1)$ and $(x, y, 5)$ within each basic module to all the other nodes within the same basic module and its broadcasting time is 2. Therefore, all-to-all broadcasting time using MLA is $m+n+4$.

Theorem 5 *Broadcasting time of $QT(m, n)$ from all-to-all broadcasting algorithm using MLA is $m+n+4$.*

In view of the $\Omega(m+n)$ lower bound, all of our broadcasting algorithms are asymptotically optimal.

6 Conclusion

This paper reported an analysis of $QT(m, n)$, which is an interconnection network that performs better than a diagonal mesh, whose degree is a constant, torus, honeycomb mesh, honeycomb torus and hexagonal torus in terms of the network cost and diameter. First, we proved that the bisection width of $QT(m, n)$ is $6n$ when m is even and $6n+2$ when m is odd. In addition, we proposed an algorithm to construct a Hamiltonian cycle in $QT(m, n)$. We also suggested one-to-all and all-to-all broadcasting algorithms for $QT(m, n)$ and proved that broadcasting time of $QT(m, n)$ by one-to-all broadcasting algorithm using SLA is $2\lfloor \frac{m}{2} \rfloor + 6$ and that of $QT(m, n)$ by one-to-all broadcasting algorithm using MLA is $2\lfloor \frac{m}{2} \rfloor + 5$. And we proved that broadcasting time of $QT(m, n)$ by all-to-all broadcasting algorithm using SLA is $2m+2n+5$ and that of $QT(m, n)$ by all-to-all broadcasting algorithm using MLA is $m+n+4$. From the analysis, $QT(m, n)$ network is one that fits the large-scale system for parallel processing.

Future research will be needed to consider embedding $QT(m, n)$ to the existing mesh family of networks. An optimal routing algorithm, fault tolerant routing algorithm or parallel routing algorithm that allows the shortest path length will be valuable topics for future research.

Acknowledgments We thank the reviewers for their comments and suggestions which have substantially improved our presentation. This research was supported by Basic Science research program through the National research Foundation of KOREA (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A4A01014439).

References

1. Seo JH, Lee HO, Jang MS (2008) Petersen-torus networks for multicomputer systems. In: Proceedings of the 4th international conference on networked computing and advanced, information management, pp 567–571
2. Dally W, Seitz C (1986) The torus routing chip. Distrib Comput 1:187–196
3. Stojmenovic I (1997) Honeycomb network: topological properties and communication algorithms. IEEE Trans Parallel Distrib Syst 8(10):1036–1042
4. Tang KW, Padubidri SA (1994) Diagonal and toroidal mesh networks. IEEE Trans Comput 43(7):815–826
5. Chen MS, Shin KG (1990) Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. IEEE Trans Comput 39(1):10–18
6. Vaidya AS, Rao PS, Shankar SR (1993) A class of hypercube-like networks, Proceedings of the 5th IEEE symposium on parallel and distributed processing, Dallas, Dec, pp 800–803

7. Yeh CH, Varvarigos EA, Parhami B (1999) Efficient VLSI layouts of hypercubic networks. In: Proceedings of the 7th symposium on the frontiers of massively parallel computation, Feb., pp 98–105
8. Ki WS, Lee HO, Oh JC (2009) The new torus network design based on 3-dimensional hypercube. In: Proceedings of the 11th international conference on advanced communication technology, pp 615–620
9. Bornstein CF, Litman A, Maggs BM, Sitaraman RK, Yatzkar T (2001) On the bisection width and expansion of butterfly networks. Theory Comput Syst 34:491–518
10. Hsu Lh, Lin CK (2008) Graph theory and interconnection networks. CRC Press, Boca Raton
11. Díaz J, Serna MJ, Wormald MC (2007) Bounds on the bisection width for random d-regular graphs. Theo Comput Sci 382:120–130
12. Garey MR, Johnson DS (1979) Computers and intractability, a guide to the theory of NP-completeness, Freeman, San Francisco
13. Monien B, Preis R (2006) Upper bounds on the bisection width of 3- and 4-regular graphs. J Discrete Algorithms 4:475–498
14. Stacho L, Vrt'o I (1998) Bisection width of transposition graphs. Discrete Appl Math 84(1–3):221–235
15. Tang KW, Kamoua R (2007) An upper bound for the bisection width of a diagonal mesh. IEEE Trans Comput 56(3):429–431
16. Zdeborová L, Boettcher S (2010) A conjecture on the maximum cut and bisection width in random regular graphs. J Stat Mech Theory Exp 2010:1–12
17. Rahman MS, Kaykobad M (2005) On Hamiltonian cycles and Hamiltonian paths. Inf Process Lett 94(1):37–41
18. Wang D (2001) Embedding Hamiltonian cycles into folded hypercubes with faulty links. J Parallel Distrib Comput 61(4):545–564
19. Albader B, Bose B, Flahive M (2012) Efficient communication algorithms in hexagonal mesh interconnection networks. IEEE Trans Parallel Distrib Syst 23(1):69–77
20. Farah RN, Othman M (2014) Broadcasting communication in high degree modified chordal rings networks. Appl Math Inf Sci 8(1):229–233
21. Ho CT, Johnsson L (1986) Distributed routing algorithms for broadcasting and personalized communication in hypercubes. In: Proceedings of international conference on Parallel Processing, pp 640–648
22. Johnson SL, Ho CT (1989) Optimal broadcasting and personalized communication in hypercubes. IEEE Trans Comput 38(9):1249–1268
23. Stewart IA (2014) Interconnection networks of degree three obtained by pruning two-dimensional tori. IEEE Trans Comput (to appear)
24. Thomson A, Zhou S (2014) Frobenius circulant graphs of valency six, Eisenstein–Jacobi networks, and hexagonal meshes. Eur J Comb 38:61–78