The Chained-Cubic Tree Interconnection Network

Malak Abdullah[1], Emad Abuelrub[2], and Basel Mahafzah[3]
[1]Department of Computer Science, Jordan University of Science and Technology, Jordan
[2]Department of Computer Science, Zarqa Private University, Jordan
[3]Department of Computer Science, The University of Jordan, Jordan

Abstract: *The core of a parallel processing system is the interconnection network by which the system's processors are linked. Due to the great role played by the interconnection network's topology in improving the parallel processing system's performance, various topologies have been proposed in the literature. This paper proposes a new interconnection network topology, referred to as the chained-cubic tree, in which chains of hypercubes are arranged in a tree structure. The major topological properties of the proposed topology have been investigated, including its diameter, degree, connectivity, bisection width, size, cost, and hamiltonicity. A comparative study is then conducted between the proposed CCT and other interconnection networks' topologies, including tree and hypercube in order to evaluate the rank occupied by CCT among other well-known topologies in terms of various performance and cost metrics. The concluding results proved that the CCT topology overcomes the shortcomings of its progenitors, tree and hypercube, while keeping most of its appealing properties.*

Keywords: *Chained-cubic tree, hypercube, tree, interconnection network, and topology.*

Received October 15, 2009; accepted November 5, 2009

1. Introduction

The past two decades have witnessed a revolution in high performance scientific computing programs, real time applications, and other technologies in various fields that are pressing request for ever-fast computers and concurrent computations, which may not be implemented without parallel processing techniques. Therefore, parallel processing is considered as one of the hot topics in the computing era.

Based on the fact that interconnection networks form the heart of parallel processing and due to their major role in parallel systems' performance, many researchers have been motivated to present, evaluate, analyze, and improve many interconnection network topologies [1, 2, 4, 6, 7, 8, 12, 21]. For example, Harwood and Shen [7] proposed a new family of extremal interconnection networks, and they analyzed their topological properties from cost, degree, diameter, and bisection width. Interconnection networks have been defined as a form of connectivity between their components to facilitate the routing of data between the parallel entities for communication and cooperating. The performance of any parallel machine is directly coupled with the degree of concurrency supplied by the interconnection network. To obtain high performance, parallel entities must be able to communicate with each other concurrently capturing a topology structure. The word topology comes from the Greek word topos meaning place and logos which means study. It is a description of any kind of locality in terms of its layout. In interconnection networks, a topology is a schematic description of the arrangement of a network (its geometry), including its nodes and connecting lines, which is considered as a branch of the mathematical graph theory. The topology of an interconnection network is considered to be its most important feature and the major determiner of performance. Building a good topology structure for an interconnection network has always been a finicky step since high concurrency may yield to expensive cost. Therefore, a wide variety of network topologies have been used in interconnection networks trying to trade off cost and scalability with performance.

The hypercube is one of the most known interconnection networks due to its attractive properties. However, because of its major drawback in terms of increasing degree for massive parallel processing [1, 8, 9, 15], many researchers proposed many alternatives of the hypercube to get over this problem. A study of the existing literature reveals that none of the known interconnection networks can be claimed to outperform all the others with respect to all the properties and metrics such as diameter, degree, bisection width, and connectivity. However, many researchers who tried to solve the degree problem of the hypercube found themselves to be involved with other problems such as high diameter, complexity design, and high cost. Therefore, many of the hypercube's alternatives and variations were not applied in the real world, but some compromises have been made in order to make it practicable. On the other hand, the tree topology cannot be considered as a practical topology, because of its low bisection width and the absence of parallel paths [8, 9]. However, it has a good degree and diameter properties with basic routing algorithms. Researchers who did investigate the tree topology found many promising alternatives. At any rate, this area is still open for more improvements on quality and performance.

Seeking a good variation of hypercubes and trees, that preserves their attractive properties and reduces their drawbacks, led to propose the Chained-Cubic Tree (CCT) network as a combination of chained hypercube and tree networks. It is instigated by the good qualities it exhibits over its constituent networks. It makes a fair compromise between them by supporting and upgrading their advantages, reducing or eliminating foremost drawbacks with acceptable cost and appealing properties.

The remainder of this paper is organized as follows. Section 2 surveys briefly some of the related work in this area. In section 3, some fundamental concepts are presented. The topological structure of the proposed interconnection network is presented in section 4. Section 5 discusses the topological properties of CCT. A comparative study with other interconnection networks is conducted in section 6. Section 7 concludes the paper and suggests some future works.

2. Related Work

A wide variety of network topologies has been used in interconnection networks to obtain high performance in a parallel system. In one direction, many researchers investigated pure topologies such as ring, mesh, hypercube, tree, and star networks. These pure topologies have gained widespread acceptance in parallel computing due to many of their attractive properties [4, 6, 8, 9, 15]. In another direction, researchers studied the combination or modification of these pure topologies, with a great deal of research that has been directed towards studying their topological properties and communication capabilities [1, 2, 8, 12, 16, 18, 19, 20, 21, 22]. Since this work proposes a new interconnection network that is related to hypercube and tree topologies, it will be helpful to mention briefly some of these topologies that are generated based on the hypercube and tree topologies.

The hypercube network is one of the popular pure topologies that was a subject of a wide range of studies and has been applied in many of the combined topologies [9]. It has many attractive properties such as low diameter, symmetry, regularity, high communication bandwidth, maximum fault-tolerance, and simple routing strategies [6, 8, 9]. Regarding to its major shortcoming in terms of increasing degree for massive parallel processing, many researchers presented variations of the hypercube or combined it with other topologies to alleviate this shortcoming. Efe *et al.* [5] introduced a new version of the hypercube with some changes made on its linking procedure. This new interconnection network has been known as the crossed hypercube network. It has been indicated that it preserves the attractive properties of the hypercube and more importantly reduces the diameter by a factor of two. Preparata and Vuillemin [14] proposed the cube-connected cycles network, which is a combination of the hypercube and the ring, produced by replacing each node of the d-dimensional hypercube with a ring of *d* nodes to eliminate the hypercube's degree drawback and to preserve the hypercube's appealing properties. Abuelrub [1] proposed the hyper-mesh interconnection network, which eliminates the major drawbacks of hypercube and mesh interconnection networks by combining them as a hypercube of two dimensional meshes. This topology has reduced the hypercube's degree and has decreased the diameter of the mesh. Loh *et al.* [11] has proposed the exchanged hypercube, which is obtained by systematically removing links from a binary hypercube. It maintains several desirable properties of the binary hypercube and reduces interconnection complexity by scaling upward with lower edge cost than the n-cube.

Many researchers tried to eliminate the drawbacks of trees in order to gain from the great appealing advantages of this pure topology. Leiserson [10] proposed fat-trees, which improve the bisection width of traditional trees. The fat-tree is an indirect interconnection network based on a complete binary tree. Unlike traditional trees in computer science, fat-trees resemble real trees because they get thicker near the root. Al-Omari [13] presented the tree-hypercube interconnection network, which consists of a full binary tree with extra links between the same tree level nodes to construct a hypercube, whose dimension is equal to the current tree's level. This hyper interconnection network has the parallel path property that could not be found in the pure tree interconnection network. Leighton [9] had proposed the mesh of trees, which is a hybrid interconnection network based on two dimensional meshes and trees. It has a small diameter relative to meshes, and a large bisection width relative to trees that supports the parallel paths property. Also, it is known as the fastest network when considered solely in terms of speed [3].

3. Preliminaries

A graph G denoted as a pair (V, E), is a finite nonempty set V of elements called vertices, together with a set E of two element subsets of V called edges. Given vertices v_1 and v_2 in a graph, the edge between them may be written as (v_1, v_2). A graph is called directed if the edges have a direction, and undirected if the edges have no implied direction [8, 9]. The d-dimensional hypercube interconnection network, denoted by Q_d, is an undirected graph consisting of $N=2^d$ nodes, labeled from 0 to 2^d-1 binary string, and has $d2^{d-1}$ edges, such that there is an edge between any two nodes if and only if the binary representations of their labels differ in precisely one bit. Therefore, it has two nodes along each dimension. Nodes that are connected via one edge have labels that are different in one bit position, nodes that are distant by two edges are different in two bit positions, and so on [8, 9, 15].

A tree is an undirected connected graph of at least two nodes with no cycles. A binary tree structure is considered as one of the inexpensive ways to connect n nodes. A full binary tree with n nodes has $n-1$ edges [9]. Each node has a connection to its parent node, plus two connections to its left and right children, with the exception of the root, which has no parent, and the leaves, which have no children. Taking into account that a tree topology provides parent-child connections between nodes, two nodes may communicate in this topology by finding a path ascending from the source to a common parent node and then descending to the destination, which will issue a bottleneck towards the root. In a binary tree interconnection network, the depth of the root is 0 and the depth of any other node is one plus the depth of its parent. A complete binary tree is a binary tree in which all the leaves are at the same depth. There are 2^p nodes at depth p of a complete binary tree. The height of a binary tree is the maximum of the depths of its leaves. A complete binary tree with height h, denoted by T_h, has $2^{h+1}-1$ nodes [8, 9].

4. Topological Structure

A new interconnection network is proposed based on a tree of height h, T_h, and a hypercube of dimension d, Q_d, topologies, referred to as the Chained-Cubic Tree, *CCT (h, d)*.

Definition 1: the chained-cubic tree interconnection network is an undirected graph, which is constructed by replacing the $2^{h+1}-1$ super-nodes of a tree T_h by hypercubes Q_d and connecting the sibling hypercubes in the same tree level with each other via extra cascading links.

It attempts to eliminate or reduce the progenitor's drawbacks with benefiting from their advantages. *CCT* is constructed by replacing the $2^{h+1}-1$ super-nodes of the tree by hypercubes of size 2^d nodes and connects the sibling hypercubes in the same tree level with each other via extra links. Figure 1 shows *CCT (1, 2)* as an example.

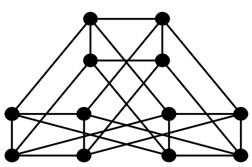

Figure 1. Chained-cubic tree *(1, 2)*.

The topological structure of the *CCT (h, d)* interconnection network can be built as follows:

1. Build a tree T_h and a hypercube Q_d. Figure 2 shows tree T_1 and hypercube Q_3, respectively.

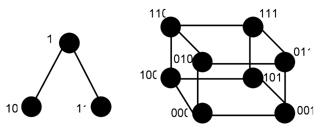

Figure 2. Tree T_1 and hypercube Q_3.

2. Replace each tree node with a 2^d nodes hypercube Q_d, as shown in Figure 3.

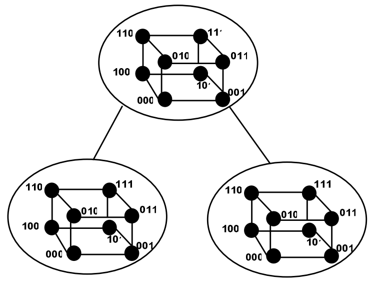

Figure 3. Replacing tree nodes with hypercubes.

3. Replace the parent-child tree links by connecting every node in the hypercube parent node with its corresponding hypercube child node. Figure 4 illustrates the process of constructing vertical parent-child connections (parent hypercube–child hypercube). For example, Figure 4 shows, in bold lines, how to connect parent hypercube's node 6 (110) with its two children hypercubes' node 6 (110).

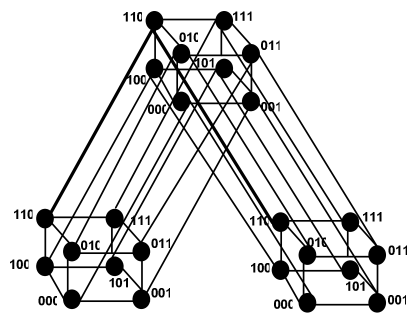

Figure 4. Vertical parent-child links for *CCT (1, 3)* for nodes 110.

At each level of the tree, add additional links to the tree hypercube nodes to connect each one of the hypercubes with each other from right and left of the same tree level to construct a chain of hypercubes in each level of the tree. These connections are denoted as the horizontal cascading links of *CCT*. Connecting two hypercubes in the same level can be applied by connecting the nodes that differ in the prefix bit. For example, Figure 5 shows, in bold lines, how to connect first hypercube's node 3 (011) with the second hypercube's node 7 (111) and vice versa.

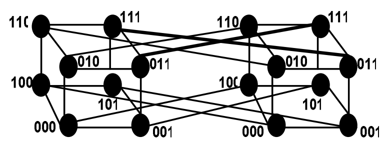

Figure 5. Horizontal cascading links for *CCT (h, 3)*.

Figure 6 shows a *CCT (1, 3)* as an example with soft vertical parent-child links and bold horizontal cascading links. Note that, to scale up the design, add a new tree level of chained hypercubes. Then connect this level with the last level of the old tree in vertical parent-child connections to produce *CCT (h+1, d*). Note that there is no need to upgrade the size of the hypercube, which could change all the nodes of degree *e* to nodes with degree *e+1*.

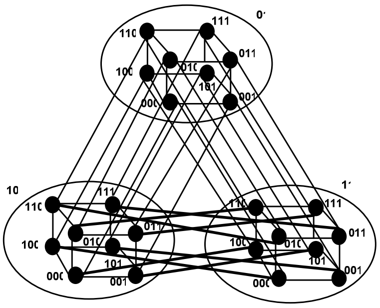

Figure 6. A *CCT (1, 3)* with soft vertical parent-child links and bold horizontal cascading links.

Now, after the building procedure is made, we need a labeling process for the nodes in the *CCT(h, d)*.

Definition 2: each node *v* in the *CCT(h, d)*, where *h* referred to the height of T_h and *d* referred to the dimension of Q_d, is labeled by a pair *<t-label(v), q-label(v)>*, where *t-label(v)* and *q-label(v)* are called the tree binary label and the hypercube binary label, respectively, in which *v* is located in the tree node that is labeled by *t-label* and in the hypercube nodes that is labeled by *q-label.* Figure 7 shows the labeling procedure for *CCT (1, 3)* which is constructed from the previous construction steps and resulted by Figure 6. For example, the node with binary representation 110 of the hypercube that lays on the root of the tree with binary representation 01 as shown in Figure 6, has been labeled in Figure 7 as <01, 110> as bold text

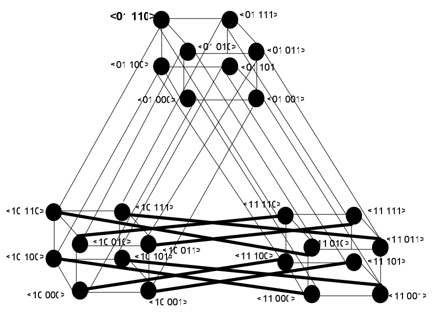

Figure 7. Labeling *CCT(1, 3).*

Definition 3: for two binary hypercube labels *q-label* (v_1) and *q-label* (v_2), where v_1 and v_2 are two nodes in *CCT,* these labels are called *prefixed-differ* hypercube labels if and only if they are differ in the prefix bit (left most). We can refer to two prefixed-differ hypercube labels by prefixed-differ*(q-label* (v_1)*, q-label* (v_2)*).*

Definition 4: for each two nodes v_1 *<t-label* (v_1)*, q-label* (v_1)*>* and v_2 *<t-label* (v_2)*, q-label* (v_2)*>* in *CCT(h, d),* v_1 is the parent of v_2 if and only if *q-label (v1)= q-label (v2)* and $\lfloor$ *(t-label(v2)/2)* $\rfloor$ *= t-label(v1)* (note that *q-label*(v_1)*= q-label*(v_2) and $\lfloor$ *log(t-label*(v_2)*)* $\rfloor$ *=* $\lfloor$ *log(t-label*(v_1)*)* $\rfloor$ *+ 1).*

Definition 5: two nodes v_1 *<t-label* (v_1)*, q-label* (v_1)*>* and v_2 *<t-label* (v_2)*, q-label* (v_2)*>* in *CCT(h, d)* are connected by an edge if and only if exactly one of the following conditions holds:

1. v_1 is a parent of v_2.
2. v_2 is a parent of v_1.
3. v_1 and v_2 are on the same tree level and *t-label* (v_1)*= t-label* (v_2)*-1* and *prefixed-differ (q-label* (v_1)*, q-label* (v_1)*).*
4. v_1 and v_2 are on the same tree level and *t-label* (v_1)*= t-label* (v_2)*+1* and *prefixed-differ (q-label* (v_1)*, q-label* (v_2)*).*

In Definition 5, the first two cases are connected via vertical parent-child links, while the other two cases are connected via the horizontal cascading links.

The vertical parent-child links are considered as a result of the cross product of tree and hypercubes, while, the horizontal cascading links are added to this structure as extra links to provide it with many appealing properties such as hamiltonicity. The next section describes the properties of this new interconnection network.

5. Topological Properties

In order to evaluate and understand the role and the behavior of this new network structure, it is helpful to answer the following two questions. What are the topological properties of this architecture? Does it provide noticeable improvements over its constituent topologies and other topologies?

There are various criteria used to characterize the performance and the cost of interconnection networks. We use the following basic properties to characterize the *CCT (h, d)* interconnection network, where *h* refers to the height of the tree, T_h, and *d* refers to the dimension of the hypercube, Q_d.

1. Diameter: the maximum distance between any two nodes in the network.

Theorem 1: the diameter of *CCT (h, d)* is *2h+d-1*

Proof: since the diameter of a tree T_h is *2h*, and the diameter of the hypercube Q_d is *d*.

Therefore, the maximum distance between any two nodes in *CCT (h, d)* will go along the tree architecture in *2h* diameter, and then pass through the hypercube in *d* diameter. Therefore, the diameter of *CCT (h, d)* will be the summation of the tree and hypercube diameters minus one since we can skip the root step. Therefore, the diameter of *CCT (h, d)* is *2h+d-1.* Figure 8 shows the maximum distance between two nodes v_1 and v_2 in *CCT.*

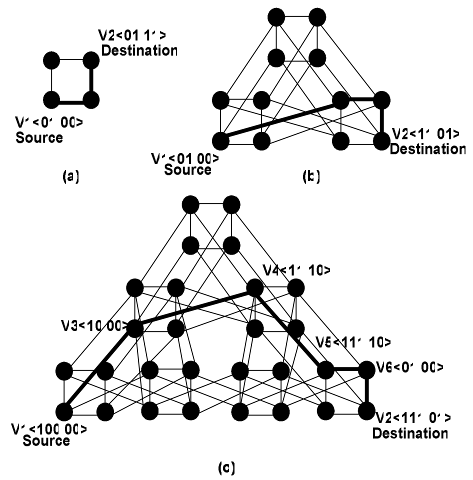

Figure 8. The maximum distance (diameter) between two nodes *(v1, v2)* in *CCT*: (a) *CCT (0, 2);* (b) *CCT(1, 2);* (c) *CCT(2, 2).*

2. Degree: the maximum number of links that are connected with a node.

Theorem 2: the degree of *CCT (h, d)* is *d+5.*

Proof: the degree of the hypercube Q_d is *d*, and the degree of the tree T_h is at most 3, see Figure (9). Also, the extra links for horizontal connections in the *CCT* will cost each node at most 2 links. Therefore, the degree will be *d*+3+2= *d*+5. Figure (10) shows the number of links of an internal node in *CCT (3, 3)* is equal to 8.

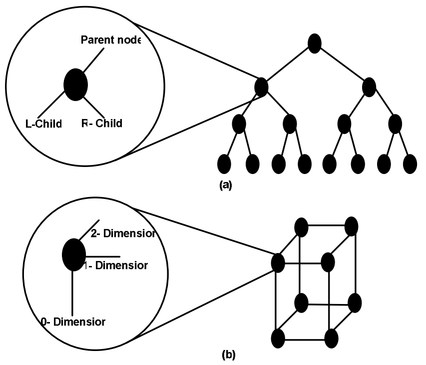

Figure 9. The vertex degree for the nodes in: (a) Tree T_3, (b) Hypercube. Q.

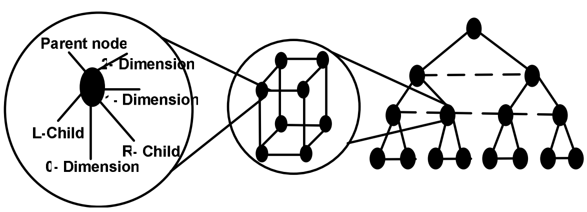

Figure 10. The vertex degree for an internal node in *CCT (3, 3).*

3. Connectivity: a measure of the multiplicity of paths between any two nodes.

Theorem 3: the connectivity of *CCT (h, d)* is at least *d+2* and at most *d+5*, and it is *d* if and only if *h=0.*

Proof: the connectivity is measured based on the node's neighbors. Therefore, all the neighbors' cases in the *CCT* topology must be considered:

1. A hypercube node in the root of a tree topology will be connected to *d* nodes of its own hypercube's group and with two child nodes. So, the number of neighbors will be *d+2.*
2. The internal node in the *CCT (h, d)* will be connected, at most, to *d* nodes of its own hypercube's group and with one parent node, two sibling nodes, and two child nodes. Therefore, the number of neighbors will be at most *d+5.*
3. A leaf hypercube node will be connected to *d* nodes of its own hypercube group, one parent node, and one or two sibling nodes. So, the number of neighbors will be at least *d+2* and at most *d+3* in this case.
4. If the height of the tree is zero, *CCT* has only one hypercube, then any node will be connected to only *d* nodes. Therefore, the number of neighbors will be *d*.

The connectivity of the *CCT* architecture depends on the cases that have the least number of neighbors; which are the first, third, and fourth cases. Therefore, the connectivity of *CCT (h, d)* is at least *d* if *h= 0*; otherwise, the connectivity will be at least *d+2.*

5. Bisection width: The minimum number of communication links that must be removed to partition the network into at most two equal halves.

Theorem 4: the bisection width of *CCT (h, d)* is $2^d (h+1.5)$.

Proof: the *CCT* architecture resembles the tree architecture in having two sub trees, and then we need to remove all the connections between these sub trees and divide the root (a hypercube) into two equal partitions. To divide the tree into two sub trees, we have to remove $h2^d$ horizontal ascendant links and 2^d vertical links that connect the root with the two halves. To divide the root hypercube we need to remove 2^{d-1} links. So, the summation will generate the bisection width of *CCT*, which is $h2^d + 2^d + 2^{d-1} = (h+1)\, 2^d + 2^{d-1} = 2^d\, (h+1.5)$.

6. Size: The number of nodes in the network.

Theorem 5: the size *of CCT (h, d)* is 2^{h+d+1}-2^d.

Proof: the size of the hypercube, Q_d, is 2^d nodes, and the size of the tree, T_h, is 2^{h+1}-*1*. The size of *CCT* will be generated by multiplying the size of the hypercube with the size of the tree and the result will be 2^{h+d+1}-2^d.

7. Cost: the number of communication links required to build the network.

Theorem 6: the cost or the number of links of *CCT (h, d)* is $2^{h+d}(d+4)-2^d(d/2+h+4)$.

Proof: the number of hypercubes in *CCT is* $(2^{h+1}-1)$. The cost of each Q_d hypercube is $d2^d/2$. The number of parent hypercubes is (2^h-1). Each parent hypercube is connected with two child hypercubes, where the number of links is equal to (2^{d+1}). If we trace the tree from left to right, we will have $(2^{h+1}-h-2)2^d$ horizontal connection links. So, the overall number of links is:

$(2^{h+1}-1)\ d2^{d-1}+\ (2^h-1)(2^{d+1})+(2^{h+1}-h-2)2^d=\ 2^{h+d}(d+4)-2^d(d/2+h+4)$. It is obvious that, this proposed topology guarantees a high bisection width and connectivity, with low diameter compared to tree topology and low degree compared to hypercube topology.

8. Hamiltonicity: a path which is a sequence of adjacent vertices is a Hamiltonian path if its vertices are distinct and they span on the vertices. A cycle which is a path with at least three vertices such that the first vertex is the last one is Hamiltonian cycle if it traverses every vertex of the graph exactly once [20].

As mentioned before, *CCT* is constructed from arranging chains of hypercubes in a tree structure. However, one property of hypercubes is that they are Hamiltonian [17, 18, 19, 20]. Sun *et al.* [18] showed the Hamiltonian laceability of faulty hypercubes. Also, Sun *et al.* [19] showed the mutually independent Hamiltonian paths and cycles in hypercubes. Moreover, Stewart [17] presented distributed algorithms for building Hamiltonian cycles in k-ary n-cubes and hypercubes with faulty links. However, next we will show that *CCT* is Hamiltonian.

Theorem 7: *CCT* is a Hamiltonian interconnection network, which means there exist a cycle that can cover all the nodes only once and ends with the same starting node.

Proof: to be Hamiltonian is so important, since this will facilitate the process of reformulating the interconnection network to another well-known interconnection networks (such as ring or tree). Since the nodes of the tree are replaced by hypercubes, and we know that the hypercube is Hamiltonian (by following the gray code procedure for visiting nodes) [6, 8], and each internal hypercube in any level of the *CCT* is connected by right and left to its sibling hypercubes. Each level of the *CCT* is also connected to two levels above and below of it. Then, there exist a path to visit all the nodes without repeating any node in this path except the starting node.

The process of constructing a Hamiltonian circuit of *CCT(h, d)* with starting node $v_0 = v_1$ <*t-label (v1), q-label (v1)*> which is also the ending node is as follows:

1. If v_1 is a node in a hypercube with unvisited nodes, then v_1 will be replaced with the next node of following the gray code procedure of *q-label* (The Hamiltonian circuit of hypercube with ignoring the last step of returning to the starting node).
2. Else, If v_1 has unvisited right child via the horizontal cascading links, and the *t-label* for both nodes end with 1, then visit the right child of v_1, and v_1 is replaced by this new visited node.
3. Else, if v_1 has unvisited left sibling node that both share the same direct parent hypercube, then visit it via the horizontal cascading link that connect them, and replace v_1 with the new visited node.
4. Else, if v_1 has unvisited parent node and the *t-label* of both nodes end with 0 then visit it via the vertical parent-child links, (Special case: if the v_1 =010 then visit the parent node). Replace v_1 with the new visited node.
5. Else, if v_1 has been connected from left with unvisited node then visit it via the horizontal cascading links, and replace v_1 with the new visited node.
6. Repeat the previous steps until all the nodes are visited and the next node is the starting node v_0.

Figure 11 shows the Hamiltonian circuit of *CCT(3, 3),* the starting node is <0110, 000> where each node in the tree corresponds to a hypercube Q_3.

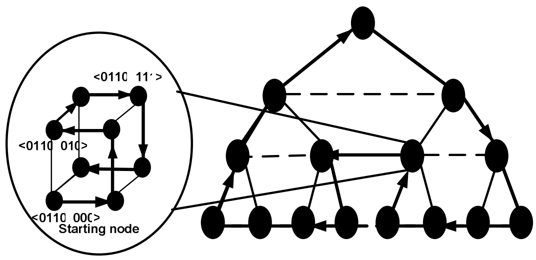

Figure 11. The hamiltonicity of *CCT(3, 3)* with starting node <0110, 000>.

6. Comparison and Evaluation

It is essentially impossible to fairly compare interconnection networks, simply because there are too many parameters and topological properties. The most suitable way for evaluating a new topology is to conduct a comparative study between the topological properties of both; the proposed topology and the other topologies that are familiar by their appealing topological properties. In this paper, it has been chosen to compare *CCT* with the following topologies: a hypercube Q_d of dimension *d,* a tree T_h of height *h*, a 2-dimensional mesh *M(r, r),* and a mesh of trees of *r by r* mesh and tree of height *h, MOT(r, h).* These are chosen according to their preference, strong properties, and the good qualities they exhibit over other topologies. Table 1 shows their topological properties.

In order to evaluate the proposed interconnection network, each of the proposed network's topological properties need to be evaluated and measured for growing sizes of *CCT.* This allows us to compare the proposed *CCT* interconnection network and determine its position among other interconnection networks based on the evaluated topological properties. This will give us the ability to judge if it is better or worse than its progenitors; tree and hypercube, and other interconnection networks, as shown in Figures (12-17). Figure 12 shows these topologies' scaling near to the preferred sizes. It is shown that the mesh and mesh of trees are the best scaling topologies. Figure 13 shows the diameter of each of the above mentioned topologies with their sizes growing up. It is obvious from Figure 13 that hypercube and *CCT* topologies have the first and second best two diameters, respectively. It means that their speeds are high compared with others. It can also be noticed that the mesh topology is considered as an outlier for its high diameter. Figure 14 shows a comparison between the above mentioned topologies according to the degree property. Tree, mesh, and *MOT* topologies have a constant degree. This means scaling up the system can be done without replacing the old processors with new ones. It can be observed that the hypercube topology has the highest degree. This is costly when the network needs to be scaled up. *CCT* has also a bad degree, but it could not be considered as a shortcoming since scaling up the system can be made by adding a new level to the tree. So, there is no need to replace the processors.

Figure 15 shows a comparative study based on the bisection width property, where high bisection width is more desirable. It is obvious that the hypercube topology has the highest bisection width. The *CCT* topology has the second highest bisection width, which is responsible for increasing its reliability. The rest of the topologies' bisection width values are low, especially, the tree topology. Figure 16 illustrates the connectivity comparisons between interconnection networks, where *CCT* has the second highest connectivity. Figure 17 shows the cost property for the above mentioned topologies. Many researchers neglect the cost of the interconnection network since the goal is to gain speed by trading off between performance and cost. Also, the cost is ignored based on the noticeable fact that reveals the continuous declining of the computer components' cost. The tree is considered as the cheapest network. The hypercube has a high performance beside its high cost, while the *CCT* topology has a moderate cost. As it has been mentioned, it may not be considered as a disadvantage for this proposed topology since it has a good performance in return.

7. Conclusions and Future Work

Huge research efforts have been directed towards studying interconnection networks due to the significant role they play in parallel processing systems' performance. So, many topologies have been proposed in the last few decades. An observable fact is that high performance may be accompanied by design and cost complexities. None of these topologies, until now, can be claimed to outperform all the others. The hypercube topology is considered as one of the strongest topologies, but because of its high degree which forms a major drawback when massive parallelism is applied, many researchers worked to find alternatives for this topology while preserving the same advantages. Tree topology cannot be considered as a practical topology, because of its connectivity and bisection width problems, although it has good degree and diameter properties with basic routing algorithms.

The *CCT* topology is introduced to make a compromise between both topologies. It enhances the constituent networks, the tree and the hypercube networks, by keeping the diameter within the moderate range of a good diameter, increasing the connectivity and the bisection width for trees to be near the hypercube, and decreasing the degree of the nodes corresponding to the hypercubes. However, it may have its own drawbacks such the physical constraints to build it, but preliminary investigations show that the new proposed *CCT* topology exhibits the good properties of its constituent networks and eliminates their disadvantages.

As a future work, more studies can be made on the *CCT* topological properties. A good problem would be to investigate its communication and computation operations.

Table 1. Topological properties of some interconnection networks.

Topology/ Properties	Size	Diameter	Degree	Connectivity	Bisection Width	Cost
Hypercube Q_d	2^d	D	d	d	2^{d-1}	$d2^{d-1}$
Tree T_h	$2^{h+1}-1$	2h	3	1	1	$2^{h+1}-2$
Mesh M(r, r)	r^2	2r-2	4	2	R	$2r^2-2r$
Mesh of Trees MOT(r, h)	$3r^2-2r$	4 log(r)	3	2	R	$4r^2-4r$
Chained-Cubic Tree CCT(h,d)	$2^{h+d+1}-2^d$	$2h+d-1$	$d+5$	at least $d+2$ at most $d+5$	$2^d(h+1.5)$	$2^{h+d}(d+4)-2^d(d/2+h+4)$

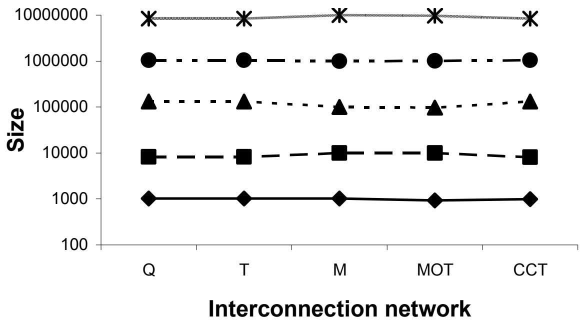

Figure 12. Interconnection networks expansion incremental differences.

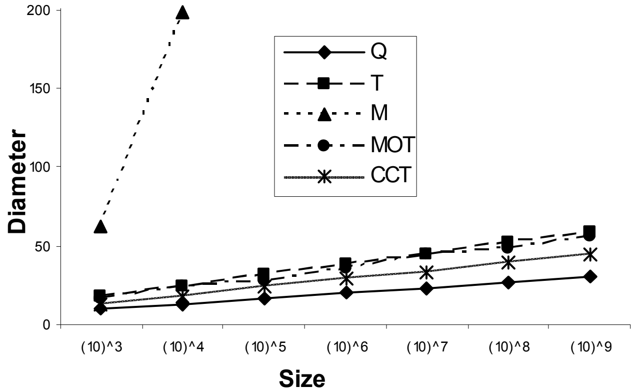

Figure 13. Diameter comparison for interconnection networks of different sizes.

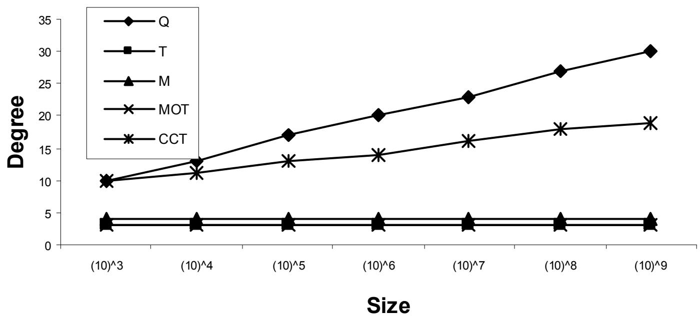

Figure 14. Degree comparison for interconnection networks of different sizes.

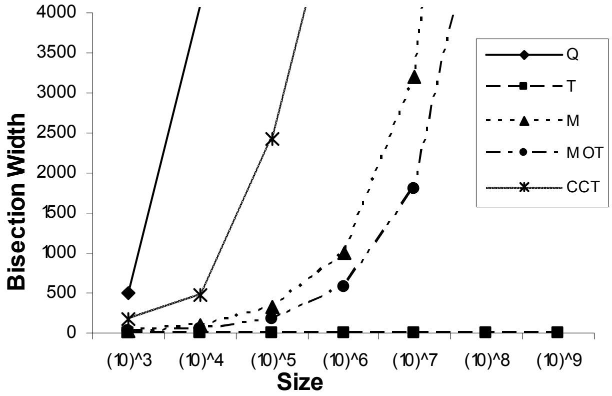

Figure 15. Bisection width comparison for interconnection networks of different sizes.

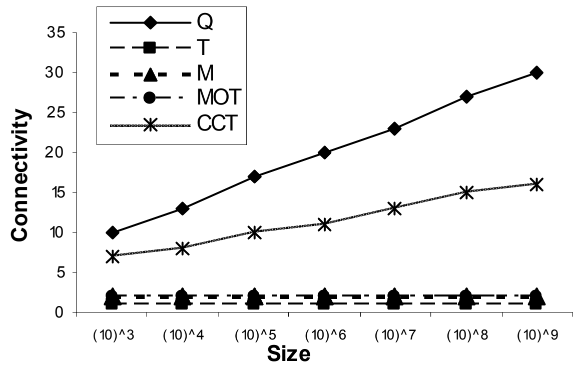

Figure 16. Connectivity comparison for interconnection networks of different sizes.

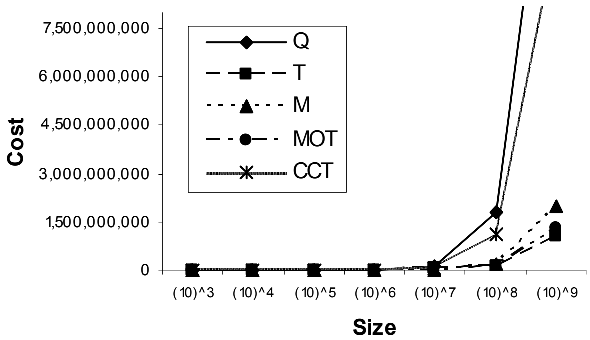

Figure 17. Cost comparison for interconnection networks of different sizes.

References

[1] Abuelrub E., "On Hyper-Mesh Multicomputers," *Computer Journal of the Institute of Mathematics and Computer Sciences*, vol. 12, no. 2, pp. 63-81, 2002.

[2] Awwad A., Al-ayyoub A., and Ould-Khaoua M., "On the Topological Properties of the Arrangement- Star Network," *Computer Journal of Systems architecture*, vol. 48, no. 11-12, pp. 325-336, 2003.

[3] Chen W., Chen G., and Hsu D., "Combinatorial Properties of Mesh of Trees," *in Proceedings of the 5th International Symposium on Parallel Architectures, Algorithms, and Networks,* USA, pp. 7-9, 2000.

[4] Day K. and Tripathi A., "A Comparative Study of Topological Properties of Hypercubes and Star Graph," *Computer Journal of IEEE Transactions on Computers*, vol. 5, no. 1, pp. 31-38, 1994.

[5] Efe K., "The Crossed Cube Architecture for Parallel Computation," *Computer Journal of IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 5, pp. 513-524, 1992.

[6] Grama A., Gupta A., Karypis G., and Kumar V., *Introduction to Parallel Computing*, IBM T.J. Watson Research Center, Yorktown Heights, USA, 2003.

[7] Harwood A. and Shen H., "A New Family of Extremal Interconnection Networks," *Computer Journal of Interconnection Networks*, vol. 2, no. 4, pp. 421-444, 2001.

[8] Jordan H. and Alaghband G., *Fundamentals of Parallel Processing*, New Jersey, 2003.

[9] Leighton F., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees,and Hypercubes*, Morgan Kaufmann Publishers, California, 1992.

[10] Leiserson C., "Fat Trees: Universal Networks for Hardware Efficient Supercomputing," *in Proceedings of the International Conference of Parallel Processing*, pp. 393-402, 1985.

[11] Loh P., Hsu W., and Pan Y., "The Exchanged Hypercube," *Computer Journal of IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 9, pp. 866-874, 2005.

[12] Loucif S., Ould-Khaouaa M., Al-Ayyoubb A., "Hypermeshes: Implementation and Performance," *Computer Journal of Systems Architecture*, vol. 48, no. 1-3, pp. 37-47, 2002.

[13] Omari M., "Tree-Hypercube: Hierarchical, Partitionable Multiprocessor Interconnection Network," *Doctoral Dissertation*, Illinois Institute, USA, 1992.

[14] Preparata F. and Vuillemin J., "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Computer Journal of Communications of the ACM*, vol. 24, no. 7, pp. 300-310, 1981.

[15] Saad Y. and Schultz M., "Topological Properties of Hypercubes," *Computer Journal of IEEE Transactions on Computers*, vol. 37, no. 7, 1988.

[16] Salazar F. and Barker J., "Hamming Hypermeshes: High Performance Interconnection Networks for Pin-out Limited Systems," *Computer Journal of Performance Evaluation*, vol. 63, no. 8, pp. 759-775, 2006.

[17] Stewart I., "Distributed Algorithms for Building Hamiltonian Cycles in K-ary N-cubic and Hypercubes with Faulty Links," *Computer Journal of Interconnection Networks*, vol. 8, no. 3, pp. 253-284, 2007.

[18] Sun C., Hung C., Huang H., Hsu L., and Jou Y., "Hamiltonian Laceability of Faulty Hypercubes," *Computer Journal of Interconnection Networks*, vol. 8, no. 2, pp. 133-145, 2007.

[19] Sun C., Lin C, Huang H., Hsu L., "Mutually Independent Hamiltonian Paths and Cycles in Hypercubes," *Computer Journal of Interconnection Networks*, vol. 7, no. 2, pp. 235-255, 2006.

[20] Tsai C., Tan J., Liang T., and Hsu L., "Fault-Tolerant Hamiltonian Laceability of Hypercubes," *Information Processing Letters*, pp. 301-306, 2002.

[21] Zheng S., Cong B., and Bettayeb S., "The Star-Hypercube Hybrid Interconnection Networks," *in Proceedings of the ISCA International Conference on Computer Application in Design*, Simulation, and Analysis, pp. 98-101, USA, 1993.

[22] Zhu Q., Xu J., Hou X., and Xu M., "On Reliability of the Folded Hypercubes," *Information Sciences*, vol. 177, no. 8, pp.1782-1788, 2007.

Malak Abdullah is an instructor of Computer Science Department at Jordan University of Science and Technology (JUST), Irbid, Jordan. She received her BSc degree in computer science and information technology from JUST, in 2004. Then, she worked at Zarqa Private University with IAJIT and ACIT team in 2005. After that, she obtained her MSc degree in computer science from JUST in 2007. Her research interests include parallel processing, network, artificial intelligent, and data mining.

Emad Abuelrub received his Bachelor degrees in computer engineering and computing information systems from Oklahoma State University, USA, in 1984 and 1985, respectively. He then joined the Alabama A&M University, USA, where he obtained his MS degree in computer science in 1987. He completed his PhD degree in computer science from Louisiana State University, USA, in 1993. Currently, he is an associate professor of computer science and dean of the Faculty of Science and Information Technology at Zarqa Private University, Jordan. He is serving as the Secretary General of the Colleges of Computing and Information Technology (CCIS) at the Association of Arab Universities, Editor-in-Chief of the International Arab Journal of Information Technology (IAJIT), and Secretary General of the International Arab Conference on Information Technology (ACIT). He is a specialist in accreditation and quality assurance of higher education and a consultant for many regional institutes. He has authored or co-authored over 60 technical papers and 20 books. His areas of interest include parallel and distributed systems, interconnection networks, fault-tolerance computing, parallel algorithms, parallel architectures, e-technology, and quality assurance in higher education. He is a member of the IEEE, ACM, IAENG, and JEA.

Basel Mahafzah is an assistant professor of computer science at the University of Jordan, Jordan. He received his BSc degree in computer science in 1991 from Mu'tah University, Jordan. He also earned from the University of Alabama in Huntsville, USA, a BSE degree in computer engineering. Furthermore, he obtained his MS degree in computer science and PhD degree in computer engineering from the same University, in 1994 and 1999, respectively. During his graduate studies he obtained a fellowship from Jordan University of Science and Technology. After he obtained his PhD and before joining the University of Jordan, he joined the Department of Computer Science at Jordan University of Science and Technology, where he held several positions; assistant dean, vice dean, and chief information officer at King Abdullah University Hospital. His research interests include performance evaluation, parallel and distributed computing, interconnection networks, artificial intelligence, data mining, software testing, and e-learning. He received more than one million US dollars in research and projects grants. Moreover, Dr. Mahafzah supervised Master students and developed several graduate and undergraduate programs in various fields of information technology. His experience in teaching extends to ten years.