Embedding Binary Tree and Bus into Hex-Cell Interconnection Network

Mohammad Qatawneh[1]

Department of Computer Science, KASIT, University of Jordan,
P.O. Box 13047, Amman 11942, Jordan
[1]mohd.qat@ju.edu.jo

Abstract. The Hex-Cell network is an interconnection network that is recursively defined and has excellent properties for scalable distributed systems. Amongst the attractive features of the hex-cell network is the embedding capability of topological structures such as binary tree and bus. In this paper, we present algorithms for embedding a binary tree and bus topologies into hex-cell interconnection network with dilation and congestion of one for tree and bus.

[Mohammad Qatawneh. **Embedding Binary Tree and Bus into Hex-Cell Interconnection Network.** Journal of American Science. 2011;7(12):367-370] (ISSN: 1545-1003). http://www.americanscience.org

Key words: Embedding, Hex-Cell network, Dilation, Congestion, Tree.

1. Introduction

The distributed computing system's performance is greatly affected by the underlying Interconnection NETwork (I-NET**)** topology. Therefore, the choice of the I-NET topology is a critical issue that must be taken into consideration when designing distributed systems and developing distributed applications. Recently, many I-NETs and their properties have been studied in the literature [1, 6, 7, 10, 11]. One of the central issues in evaluating a network is to study the graph embedding problem [2, 3, 9, 10, 11], an. Graph embedding is a technique in I-NETs that maps a guest graph into another host graph (usually an I-NET).

Four criteria can be used to evaluate embedding results. The first criterion is dilation which can be defined as the maximum distance between any two nodes on the newly derived structure, measures the system communication time. The second is congestion**.** It is the key factor affecting the system delay and packet loss during data transmission. Expansion is the third criterion which reflects the efficiency of node utilization for an embedding scheme. The capability of fault tolerance denotes the reliability and robustness of the derived structure [4, 5, 7, 8].

The hex-cell network is emerging as one of the good candidate network architectures for parallel machines. This is due to the facts that the hex-cell has a simple recursive structure and that there are simple routing algorithms [1, 2]. Another important consideration in the choice of network architecture is its ability to accommodate different algorithms efficiently. The problem of efficiently implementing various algorithms on parallel architectures has traditionally been studied as the "logical mapping problem" [12, 13] so that the problem of implementation becomes one of embedding the "data-dependency graph" underlying an algorithm within the processor interconnection graph.

This paper proposes an algorithm for embedding a binary tree and another for a bus into hex-cell interconnection network. The main result is that the N-node hex-cell can emulate every N-node tree with a constant factor. In particular, the paper shows how to embed any tree node within an N-node hex-cell so that:

1- The mapping of tree nodes to hex-cell nodes is one-to-one (i.e., the max load of the embedding is one).

2-The dilation of the embedding is 1.

3- The congestion of the embedding is one.

The rest of this paper is organized as follows. Section 2 presents the definition of hex-cell network. Section 3 proposes the embedding algorithms for binary tree and bus into hex-cell network. Section 4 exhibits concluding remarks.

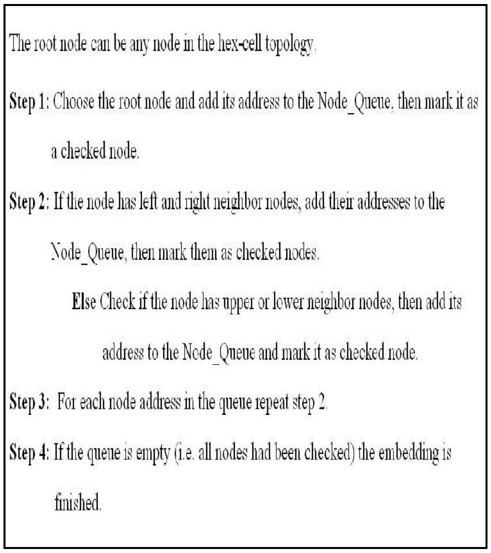
2. Definition of Hex-Cell Network

A hex-cell network with depth d is denoted by HC (d) and can be constructed by using units of hexagon cells, each of six nodes [1]. A hex-cell network with depth d has d levels numbered from 1 to d, where level 1 represents the innermost level corresponding to one hexagon cell. Level 2 corresponds to the six hexagon cells surrounding the hexagon at level 1. Level 3 corresponds to the 12 hexagon cells surrounding the six hexagons at level 2 as shown in Fig. 1 below. The levels of the HC (d) network are labeled from 1 to d. Each level i has Ni nodes, representing processing elements and interconnected in a ring structure.

In order to explain the following proposed algorithms, the level numbering scheme is used as follows. Each node in the HC is identified by a pair (X.Y), where X denotes the line number in which the hex-cell topology with two levels; i.e. HC (2) as shown in Fig. 2 is considered as a host graph for the proposed embedding algorithms.

3.1. Embedding Binary Tree into Hex-Cell Network

The tree topology is an important class of communication graphs since it is a common data structure. A binary tree is a tree in which the degree of every vertex is less than or equal to 3. Mapping tree structure into different networks has been widely studied [8, 10, 12]. The proposed embedding binary tree into hex-cell network is shown in Fig. 3. The algorithm shows that the binary tree can be embedded into hex-cell network with a dilation and congestion of 1.



```
The root node can be any node in the hex-cell topology.
Step 1: Choose the root node and add its address to the Node_Queue, then mark it as
        a checked node.
Step 2: If the node has left and right neighbor nodes, add their addresses to the
        Node_Queue, then mark them as checked nodes.
            Else Check if the node has upper or lower neighbor nodes, then add its
            address to the Node_Queue and mark it as checked node.
Step 3: For each node address in the queue repeat step 2.
Step 4: If the queue is empty (i.e. all nodes had been checked) the embedding is
        finished.
```

Figure 3: Embedding binary tree into hex-cell network

3.2 Discussions via examples

The following examples explain the embedding of tree into hex-cell network. Applying the algorithm in Fig. 3, let (3.1) be the root node. When executing the algorithm, step 2 will be applied, and this leads to nodes (3.2) and (2.1). The algorithm will repeat itself until the queue is empty. The final result of the algorithm is shown in Fig. 4. Fig. 5 displays the outcome of executing the algorithm starting with node 3.3.

node exists, and Y denotes the location of the node in the line as in figure 2. A node with the address 1.1 is the first node that exists at the line 1. The address 1.2 refers to the second node that exists at the line 1, and so on.

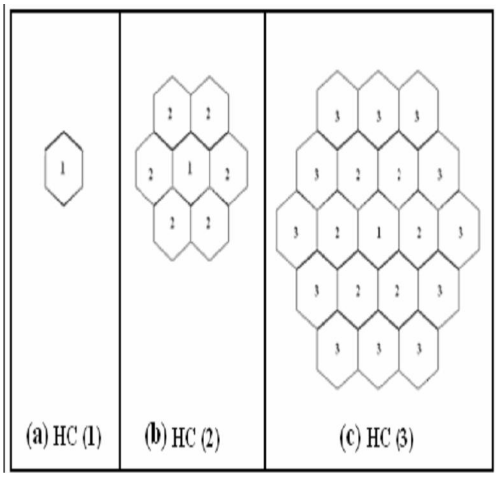


Figure 1: (a) HC (one level) (b) HC (two levels) (c) HC (three levels)

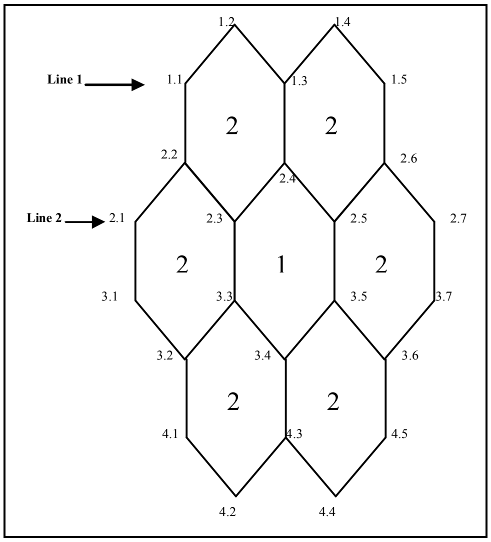


Figure 2: Addressing nodes in HC (2)

3. Proposed Embedding algorithms

The hex-cell embedding problem, a restricted version of the general mapping problem, is the problem of mapping a communication graph into hex-cell network. A hex-cell is an undirected graph H = (V_H, E_H) consisting of n = 6d2 vertices labeled by a pair (X.Y) as shown in fig. 2. An embedding f of a graph G = (V, E) into a hex-cell network H = (V_H,E_H) is a one-to-one function $f : V \rightarrow V_H$. For embedding f, the dilation cost of an edge {V, W} Є E is d(f(V),f(W)). For the hex-cell embedding problem, a

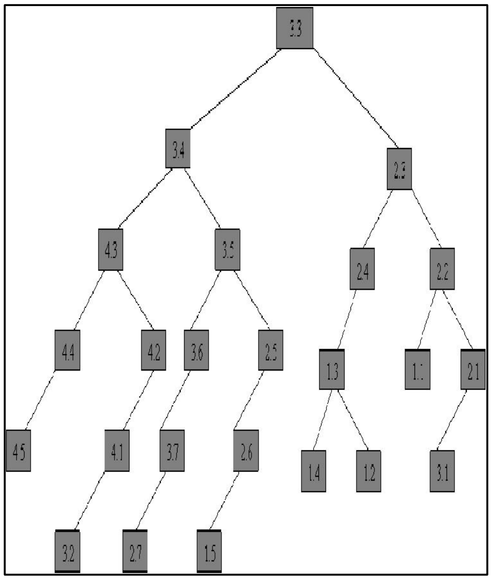


Figure 6: Embedding binary tree into hex-cell network

The following example explains the embedding of bus into hex-cell network. Let (2.1) be the address of the start node. By executing steps 2 and 3, we reach the node (2.2). Then, moving one inner level leads to the node (2.3). Step3 is repeated until node (2.4) has been reached as shown in Fig. 7.

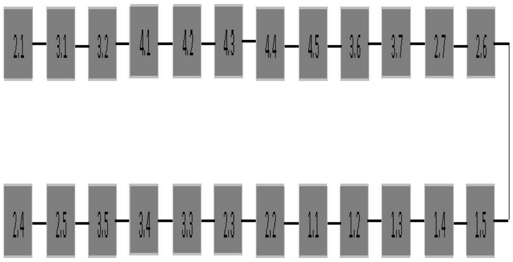


Figure 7: Embedding a bus topology into HC (2).

4. Conclusion

The algorithm shows that the binary tree can be embedded into hex-cell network with a dilation and congestion of 1.This paper proposed two algorithms for dynamic embeddings of binary tree, and bus topologies into hex-cell network with optimal dilation and congestion. The computations needed to embed these topologies are simple and the embedding can be computed in linear time. The proposed embedding schemes have the advantage of simple and straightforward implementations. The dilation and congestion is one for both proposed algorithms which positively reflects on the communication and delay costs.

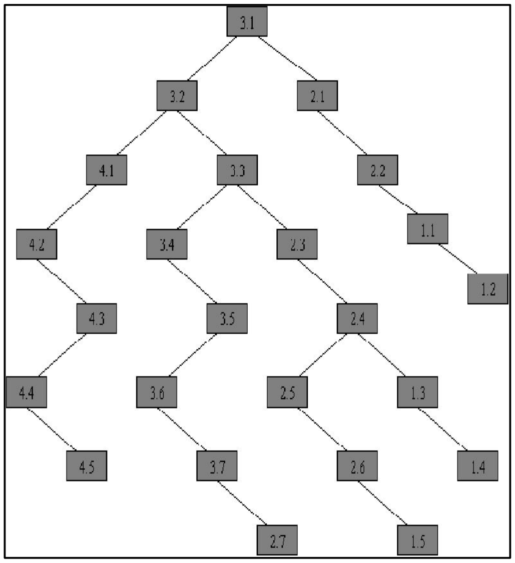
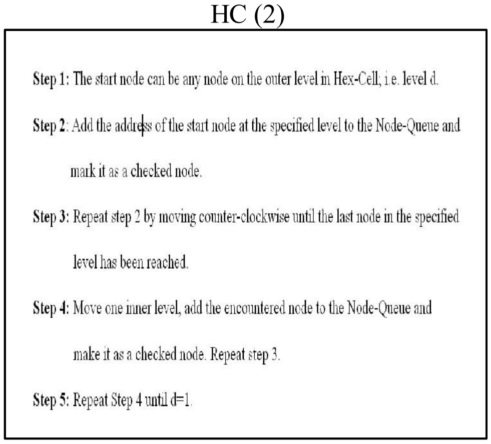


Figure 4: The result of embedding binary tree into HC (2)



Step 1: The start node can be any node on the outer level in Hex-Cell, i.e. level d.

Step 2: Add the address of the start node at the specified level to the Node-Queue and mark it as a checked node.

Step 3: Repeat step 2 by moving counter-clockwise until the last node in the specified level has been reached.

Step 4: Move one inner level, add the encountered node to the Node-Queue and make it as a checked node. Repeat step 3.

Step 5: Repeat Step 4 until d=1.

Figure 5: The result of embedding binary tree into HC (2)

3.3 Embedding Bus into Hex-Cell Network

The proposed algorithm for embedding bus topology into Hex-Cell network is shown in Fig. 6. The algorithm maps a bus topology at the outer level of Hex-Cell. Then, it maps the inner level until d=1 has been reached. The following example explains the embedding of a bus topology into hex-cell network. Applying the algorithm in Fig. 7, let (2.1) be the start node. When executing the algorithm, step 2 will be applied by moving court-clockwise until the last node in the specified level has been reached as shown in figure 8. The algorithm shows that the bus topology can be embedded into hex-cell network with a dilation and congestion of 1.

8- Baril J. and Vajnovzki V. Minimal change list for Lucas strings and some graph theoretic consequences, *Elsevier. Theoretical Computer Science 2005;* 346:189-199.

9- Ernastuti A., and Vajnovzki V. Embeddings of linear arrays, Rings and 2-d dimensions on extended Lucas cube networks, *Proceedings of the International Conference on Electrical Engineering and Informatics Institute Teknologi Bandung*, *Indonesia*, 2007; 17-19.

10-Younes M. A Study of Mapping Some Popular Schemes into the Hex-Cell, *Journal of Applied Sciences Research, 200;* 5(8): 974-977.

11- Tsai C. Linear array and ring embeddings in conditional faulty hypercubes, Theoretical Computer Science, 2004; 314: 431 – 443.

12- Berman F. and Snyder L. On mapping parallel algorithms into parallel architectures, J. Parallel Distrib. Computing, 1987; 4: 439-458.

13- Bokhari S. On the mapping problem, IEEE Trans. Comput., 1981; C-30: 207-214.

11/20/2011

References

1- Sharia A., Qatawneh M., Almobaideen W., and Sliet A. Hex-Cell: Modeling, Topological Properties and Routing Algorithm, *European Journal of Scientific Research* 2008; 22(2): 457-468.

2- Qatawneh M., Bdour H., Sabah S., Samhan R., Sliet A., Qatawneh J., and Al-Mobaideen W. An alternative Routing algorithm for hex-Cell network, Information: An international interdisciplinary Journal 2011; 14(10): 3499-3514.

3- Almobaideen W., Qatawneh M., Sleit A., and Salah I. Efficient mapping scheme of ring topology onto tree-hypercubes, *Journal of. Applied Sci*. 2007; 7: 2666-2670.

4- Qatawneh M. Embedding linear array onto tree-hypercube network, *European Journal of scientific Research, 200;,* 10(2): 72-76.

5- Yang M., Jimmy J, and Hsu L. Hamiltonian circuit and linear array embeddings in faulty *k*-ary *n*-cubes, Journal of Parallel Distrib. Comput 2007; 67: 362-368.

6- Tsai C. Linear array and ring embeddings in conditional faulty hypercubes*, Theoret. Comput. Sci*. 2004; 3: 431–443.

7- Day K. The conditional node connectivity of the *k*-ary *n*-cube, *Journal of Interconnection Networks 2004;* 5 (1): 13–26.