

SHORTEST PATH ROUTING IN EXCHANGED FOLDED HYPERCUBE

K.Bhavani¹, P.Venkat Reddy², Sudarson Jena³, V. Kamakshi Prasad⁴

¹Research Scholar, GITAM University, Hyderabad

²Research Scholar, JNTU, Hyderabad

³GITAM University, Hyderabad

⁴JNTUH, Hyderabad

Abstract

Routing plays a vital role in transmission of data from one node to another in interconnection networks and also used to determine the performance of the network . In this paper the shortest path routing for Exchanged Folded Hypercube is proposed whose diameter is $s+t$ when $s \leq 2$ and $t \leq 2$.we have also provided the conditions for determining the path from node u to node v in graph $EFH(s,t)$ which determines a path of length $s+t$. We have also shown the broadcasting communication from the source node to different destination nodes by using improved exchanged tree which is a spanning tree and having the properties of binary tree.

Keywords- Exchanged Folded Hypercube, Hypercube, Routing Algorithm, Interconnection of Networks.

1. INTRODUCTION

Now a days Parallel and Distributed systems plays a very important role in interconnection of the networks[2] because it is necessary for one processor to communicate with other processor when large number of processors execute the program to solve a problem. The topology will determine how the systems are interconnected[3]. The architecture is represented as a Graph $G(V,E)$ where the set of V are the nodes and E is the set of edges which represent the communication links between the nodes. The Graph $G(V,E)$ is said to be connected if there exists a path of length L from one node to another[7].

The Hypercube is popular network topology that is used in interconnection of the networks due to it's high reliability and it's embedding properties, strong connectivity and regularity[1]. There are 2^n nodes in the Hypercube which are connected if and only if there is a one bit difference between the two nodes. There are many other topologies like exchanged hypercube(EH)[8] and exchanged folded hypercube(EFH) are developed from the Hypercube. The exchanged hypercube is obtained by removing some of the edges of hypercube.

The Routing algorithm is used to determine the path for transmission between two different nodes. There are three types of routing algorithms.

- i). Deterministic routing algorithm
- ii).Adaptive routing algorithm
- iii).Randomized routing algorithm

In deterministic routing algorithm the route from the source to the destination is predefined and all the flits should follow the same path. In adaptive routing algorithm we will consider the traffic or the status of the channel between two nodes such that the congestion can be avoided and data is transmitted. In randomized routing algorithm the path for the

routing is determined randomly and we can use the random generator to determine the route between two nodes.

In message passing networks the designing of routing algorithms for communications plays an important role[6].The routing algorithm will determine the transmission of data from one node to another. While transmission it is necessary to select the best path . The selection of the best path can be done by the shortest path routing algorithm. The shortest path routing algorithm is used to determine the path between the source and the destination whose diameter is minimum. Yang Li proposed a routing algorithm whose diameter is $\lceil s/2 \rceil + \lceil t/2 \rceil + 2$ when $s \geq 2$ and $t \geq 2$. In this paper shortest path routing algorithm for the $EFH(s,t)$ is determined whose diameter is $s+t$ for $s \leq 2$ and $t \leq 2$.

The EFH is obtained from EH which is obtained from n -cube. The total number of nodes in EH is 2^{s+t+1} and the total number of edges are $(s+t+2)2^{s+t+1}$. The exchanged hypercube with $s=1$ and $t=2$ is as shown in Fig 1.

Table 1- Comparison of Hypercube , Exchanged folded Hypercube, Exchanged hypercube

Network	Edges	Degree of the node	Diameter
hypercube	$n2^{n-1}$	n	N
exchanged hypercube(s,t)	$(s+t+2)2^{s+t+1}$	$s+1$ or $t+1$	$s+t+1$
exchanged folded hypercube(s,t)	$(s+t+4)2^{s+t+1}$	$s+2$ or $t+2$	$\lceil s/2 \rceil + \lceil t/2 \rceil + 2$ or $s+t$

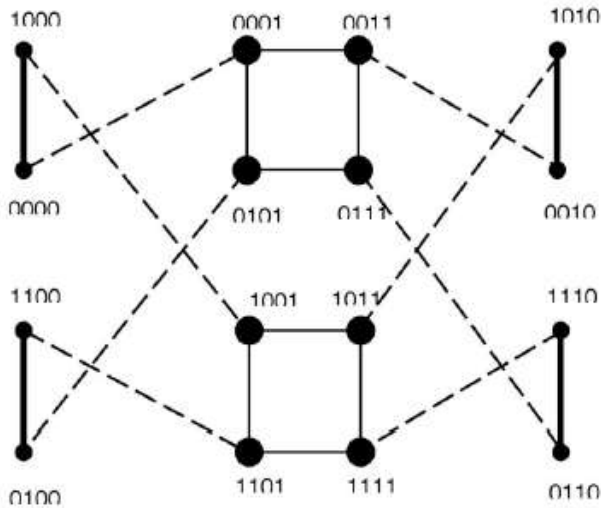


Fig1. Exchanged Hypercube EH(1,2)

The rest of the paper is organized as follows . The structure of the exchanged folded hypercube and the connection between the adjacent nodes are determined in Section II. The cases that are considered in routing algorithm are discussed in section III. The broadcasting communication which is represented in spanning trees is discussed in section IV followed by conclusion.

2. EXCHANGED FOLDED HYPERCUBE

The exchanged folded hypercube [10] is determined from exchanged hypercube which is in turn obtained from the hypercube one of the most popular topologies in parallel and distributed systems. The total number of nodes in exchanged folded hypercube is 2^{s+t+1} . The total number of edges are $(3s+3t+4) 2^{s+t+1}$ links less when compared to the folded hypercube[5]. The diameter of the EFH is half of the diameter of EH and also the cost factor is less when compared with EH. The degree of the EFH is $s+2$ if the node address ends with 0 or $t+2$ if the node address ends with 1. The degree of the node is defined as the total number of links connected to the node.

Let us consider the graph $G(V,E)$ as an undirected graph in which the V is the set of vertices and E is the set of edges. The vertices are labeled with binary strings. The exchanged folded hypercube is represented as $EFH(s,t)$ where $s \geq 1$ and $t \geq 1$. The exchanged folded hypercube of $EFH(1,2)$ is as shown in Fig 2 where $s=1$ and $t=2$. The total number of vertices are 2^{s+t+1} and the total number of edges of the EFH are $(s+t+4) \times 2^{s+t+1}$.

Lemma 1: The nodes in $EFH(s,t)$ is 2^{s+t+1} and the total edges in $EFH(s,t)$ are $(s+t+4) \times 2^{s+t+1}$.

Proof: As discussed the nodes in exchanged folded hypercube are 2^{s+t+1} . There are four different types of edges in $EFH(s,t)$ the total number of edges of E_1 are $(s+t+1) \times 2^{s+t}$ and E_2 edges are $2^s \times t \times 2^{t-1} = t \times 2^{s+t-1}$ and E_3 edges are $2^t \times s \times 2^{s-1} = s \times 2^{s+t-1}$ and E_4 is 2^{s+t} .

So the total number of edges in the graph $EFH(s,t)$ are

$$(s+t+4) \times 2^{s+t+1} .$$

There are four different types of the edges which connect two different vertices based on the conditions mentioned below.

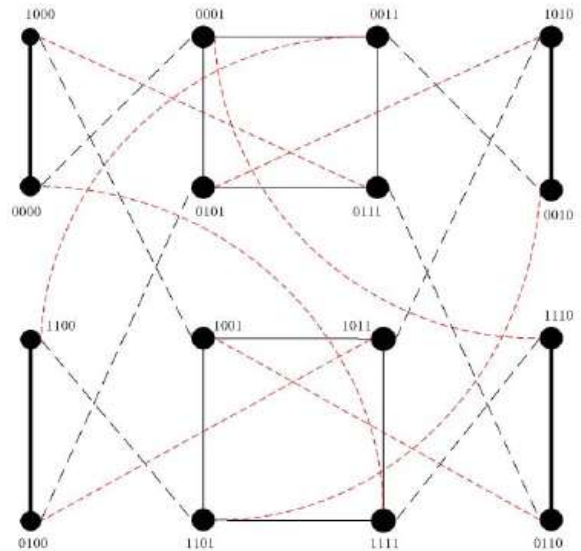


Fig 2. The Exchanged Folded Hypercube EFH(1,2)

1. E_1 : Dark lines represent that $V_1[0]= V_2[0]=0, V_1[t:1]= V_2[t:1], H(V_1[s+t:t+1], V_2[s+t:t+1])=1$.
2. E_2 : Solid lines represent that $V_1[0]= V_2[0]=1, V_1 [s+t:t+1]= V_2[s+t:t+1], H(V_1[t:1], V_2[t:1])=1$.
3. E_3 :Black dashed lines represent $V_1 \oplus V_2=1$
4. E_4 :Red dashed line represents $V_1 \oplus V_2=s+t+1$.

The E_4 edge can be determined as the complementary edge.

The \oplus denotes that the operation performed is an EX-OR operation.

$H(u,v)$ will denote the hamming distance between the nodes u and v .

Hamming distance is the number of bits that both the u and v differ[1].

The complement of the node address can be obtained by interchanging 1's to 0's and 0's to 1's.

Theorem 1: The degree of the $EFH(s,t)$ is $s+2$ if the node address ends with 0 and $t+2$ if the node address ends with 1.

Proof : As we know that the $EFH(s,t)$ is obtained by adding some edges to $EH(s,t)$ degree of the exchanged hypercube is $s+1$ if the node address ends with 0 and $t+1$ [8] if the node address ends with 1 only the E_4 edge is added to each node the degree is incremented by 1. So we will obtain the degree $s+2$ in $EFH(s,t)$ if the node address ends with 0 and $t+2$ if node address ends with 1.

3. SHORTEST PATH ROUTING IN EFH

Routing is the process of selecting best paths in a network. The term routing also meant forwarding network traffic among networks. A data packet is typically forwarded from one node to another through the networks that constitute the internetwork until it reaches its destination node. The transmission of data in the network will effect the performance of the network. The smaller the diameter the less complexity in transmission of data. The distance between the vertices u and v represented as $d(u, v)$ is the shortest path of length l from u to the vertex v . If the source and destination are not connected based on the conditions specified above we will use the intermediate nodes for transmission which will be determined by the algorithm described below. If there are no possible paths from the existing node then we can traverse to the previous node and can determine the intermediate node by using depth first search technique[4].

Definition 1: The diameter of $G(V,E)$ denoted by D_G is given by $D_G = \max\{d(u,v): u,v \in V\}$

The shortest distance between the two nodes in the network is given by D_G .

The deterministic routing algorithm discussed in this paper will determine the shortest path from one node to another which deals with the comparison of most significant bits between two nodes. There are four different cases that we consider in routing from node u to the node v .

Algorithm:

Input: If $u_{n-1} u_{n-2} \dots u_0$ be the source and $v_{n-1} v_{n-2} \dots v_0$ be the destination.

Output : The shortest path from u to v

Case 1:

```

Step 1: if ( $u_{n-1} = v_{n-1} = 0$ )
{
  if( $H(u,v)=1$  or  $H(u,v)=2$ )
  {
    if( $u_0=0$ )
    {
       $u_0=1$ ;
    }
    else check for the different bit and modify it.
  }
  if( $H(u, v) > 2$ )
  {
     $u_{n-1}=1$ 
  }
}

```

```

Step2: for each intermediate node check  $H(i,v)$  where  $i$ 
is the intermediate node.
if( $H(i,v)=1$  or  $H(i,v)=2$ )
{
  check for the difference bit and modify it
}

```

```

Step 3: check for the connections by the conditions
specified if available include it in the path or change the
next bit in the intermediate node if we have reached  $u_0$  then
visit the previous node and modify the bits.

```

```

Step 4: If( $H(i, v) > 2$ )
take the complement of the intermediate node
go to step 3

```

```

Step 5: check if the node is the destination if it is not the
destination continue the process until the destination is
reached.

```

Case 2:

```

Step 1: if ( $u_{n-1} = v_{n-1} = 1$ )
{
  if( $H(u,v)=1$  or  $H(u,v)=2$ )
  {
    if( $u_0=0$ )
    {
       $u_0=1$ ;
    }
    else check for the different bit and modify it.
  }
  if( $H(u, v) > 2$ )
  {
     $u_{n-1}=0$ 
  }
}

```

```

Step2: for each intermediate node check  $H(i,v)$  where  $i$ 
is the intermediate node.
if( $H(i,v)=1$  or  $H(i,v)=2$ )
{
  check for the difference bit and modify it
}

```

```

Step 3: check for the connections by the conditions
specified if available include it in the path or
change the next bit in the intermediate node if
we have reached  $u_0$  then visit the previous node
and modify the bits.

```

```

Step 4: if( $H(i, v) > 2$ )
take the complement of the intermediate node
go to step 3

```

```

Step 5: continue the process until the destination is
reached.

```

Case 3:

```

Step 1: if ( $u_{n-1} = 0$  and  $v_{n-1} = 1$ )

```

```

Step 2: if ( $u_{n-1} = u_0 = 0$ )
If( $H(u, v) > 2$ )
take the complement and check for the different bits of the
intermediate node and change it.
go to step 3

```

Step 3: check for the connections by the conditions specified if available include it in the path or change the next bit in the intermediate node if we have reached u_0 then visit the previous node and modify the bits.

Step 4: else if($H(u, v)=2$) and $u_0 = v_0$
take the complement of the node check for the different bits of the intermediate node and change it.
go to step 3

step 5: else if($H(u, v)=2$) and $u_0 \neq v_0$
 $u_{n-1}=1$
go to step 3

step 6: if($H(u, v)=1$)
 $u_{n-1}=1$
go to step 3

step 7: else if ($u_{n-1} \neq u_0$)
if($H(u,v)=1$ or $H(u,v)=2$)
 $u_0=0$;

step 8: for each intermediate node check $H(i,v)$ where i is the intermediate node.
if($H(i,v)=1$ or $H(i,v)=2$)
{
check for the difference bit and modify it
}
goto step3.

Step 9: if($H(i, v)>2$)
take the complement of the intermediate node
go to step 3

Step 10: if($H(u, v)>2$) and $u_0 = v_0$
take the complement of the source node
go to step 8 and 9
if($H(u, v)>2$) and $u_0 \neq v_0$

Check for the 0's in v and place 1's in the positions of source if already present check for 1's and place 0's

goto step 3.

Case 4:

Step 1: if ($u_{n-1} = 1$ and $v_{n-1}=0$)

Step 2: if ($u_{n-1} \neq u_0$)
if($H(u, v)>2$)
take the complement check for the different bits of the intermediate node and change it.
go to step 3

Step 3: check for the connections by the conditions specified if available include it in the path or change the next bit in the intermediate node if we have reached u_0 then visit the previous node and modify the bits.

Step 4: else if($H(u, v)=2$) and $u_0 = v_0$
take the complement check for the different bits of the intermediate node and change it.
go to step 3

Step 5: else if($H(u, v)=2$) and $u_0 \neq v_0$
 $u_{n-1}=0$
go to step 3

Step 6: if($H(u, v)=1$)
 $u_{n-1}=0$
go to step 3

Step 7: else if ($u_{n-1} = u_0$)
if($H(u,v)=1$ or $H(u,v)=2$)
 $u_0=0$;

Step 8: for each intermediate node check $H(i,v)$ where i is the intermediate node.
if($H(i,v)=1$ or $H(i,v)=2$)
{
check for the difference bit and modify it
}
goto step3.

Step 9: if($H(i, v)>2$)
take the complement of the intermediate node
go to step 3

Step 10: if($H(u, v)>2$) and $u_0 = v_0$
take the complement of the intermediate node
go to step 8 and 9
if($H(u, v)>2$) and $u_0 \neq v_0$

Check for the 0's in v and place 1's in the positions of source if already present check for 1's and place 0's
goto step 3.

Suppose if we want to transmit the data from 0001 to 0110 then as $u_{n-1} = v_{n-1} = 0$ we will consider case 1. The hamming distance between the two nodes is 3 as it is greater than 2 we will make $u_{n-1}=1$. So the node obtained is 1001 so now we have to check for the connections as there are no connections we have to modify the next bit and this is continued the intermediate node that will be obtained is 1110 as the hamming distance between the two nodes is 1 we have to check for the difference bit and the difference bit is at u_{n-1} we will modify the bit from 1 to 0 so the node obtained is 0110 which is the destination so the path that is obtained is 0001-1110-0110.

Broadcasting Communication

The broadcasting communication is the transmission of data from one source to multiple destinations in the network. The broadcasting of data from source to the destination is represented by the spanning trees. The construction of the spanning tree for communication is done by Improved exchanged tree(IET) proposed by K.Li,Y.Mu,G.Min[9].

Theorem 2: The spanning tree of EFH(s,t) is given by IET(s,t)

Proof : The spanning tree consists of the vertices and edges where the vertices are the nodes and the edges are the communication links between the nodes so the nodes and links in spanning tree are the subset of the graph EFH(s,t).so the IET(s,t) is subset of EFH(s,t).

4. EVALUATION OF COMMUNICATION IN EFH

The broadcasting is done from one node to more than one destination. The below figure 3 and figure 4 shows the spanning trees which determine the shortest paths from the source node to the different destination nodes and the intermediate nodes that are used for transmission of data to the destination.

The source in the below fig 3 is the 0000 and the different destination nodes are 0010,0110,1010

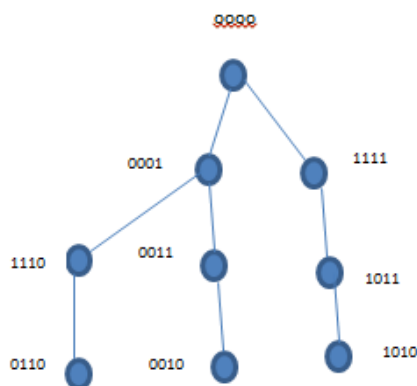


Fig 3.The improved exchanged tree IET(1,2)

The source in the below fig 4 is the 1011 and the destinations are 0000,0101,0011

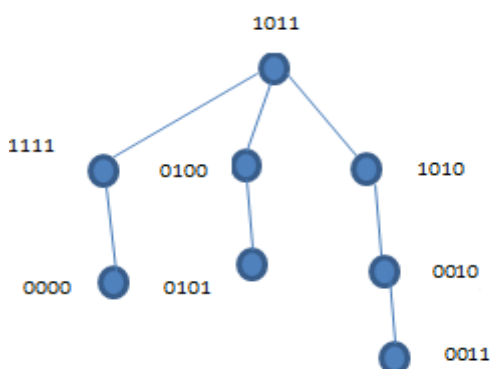


Fig 4.The improved exchanged tree IET(1,2)

5. CONCLUSION

Exchanged folded hypercube is a variant of hypercube and its maintaining the properties of hypercube is discussed. The properties of exchanged hypercube and the hypercube and exchanged folded hypercube are discussed. This paper determined the shortest path routing algorithm for transmission of data from source node u to the destination node v whose diameter is s+t when $s \leq 2$ and $t \leq 2$ in EFH(s,t). The broadcasting communication from source to different destinations is also determined.

REFERENCES

- [1] Y. Saad and M. H. Schultz , "Topological properties of hypercubes" , IEEE Trans. Comput. , vol. C-37 , pp.867 -872 , 1988
- [2] L.N. Bhuyan, Q. Yang, and D.P. Agrawal, "Performance of Multiprocessor Interconnection Networks," Computer, vol.22,no. 2, pp. 25-37, Feb. 1989.
- [3] R.L. Sharma, Network Topology Optimization—The Art and Science of Network Design. Van Nostrand Reinhold, 1990.
- [4] M. S. Chen, K. G. Shin, "Depth-first Search Approach for Fault-tolerant Routing in Hypercube Multicomputers", IEEE Trans. on Paral. Distrib. Syst. 1, 2:152-159, 1990.
- [5] A. El-Amawy and S. Lati , "Properties and performance of folded hypercubes," IEEE Trans. Parallel Distrib. Syst., vol. 2, no. 1, pp. 31–42, Jan. 1991.
- [6] J. Duato, S. Yalamanchili, and L. Ni. Interconnection networks : an engineering approach. IEEE computer society press, 1997.
- [7] D.B. West , Introduction to Graph Theory, september, 2000.
- [8] P. K. K. Loh, W. J. Hsu, and Y. Pan, "The Exchanged Hypercube," IEEE Trans. Parallel Distrib. Syst., vol. 16, no. 9, pp. 866–874, Sep. 2005.
- [9] K. Li, Y. Mu, K. Li, and G. Min, "Exchanged crossed cube: A novel interconnection network for parallel computation," Parallel and Distributed Systems, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2012.
- [10] Yang Li, Heng Qi, Zhiyang Li and Kequi Li " The Exchanged Folded Hypercube" IEEE Inter.Conf. on High Perf. Comp. and Communication. pp 45-50, 2013.