Volume 2, No. 5, Sept-Oct 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Optimal Routing Algorithm in a Octagon-Cell Network

Sanjukta Mohanty*
Lecturer
Dept of Computer sc. & Applications
North Orissa University, Sriram Chandra Vihar
Takatpur, Baripada, Mayurbhanj,India
sanjuktamohanty2005@rediffmail.com

Dr Prafulla ku. Behera
Reader
Dept of Computer sc. & Application
Utkal University, Vani Vihar
Bhubaneswar, India
p_behera@hotmail.com

Abstract: In an Interconnected network topology, the source node first makes a connection with the destination node before sending a packet. The routing algorithm is the part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. In this paper an efficient and optimized routing algorithm for octagon-cell network topology is presented and we have shown the octagon-cell network by means of an undirected Graph G = (V,E), where V is the set of nodes in the graph (hosts in the network) & E is the set of edges in the graph (links in the network). Octagon-cell can be expandable in an incremental fashion due to its recursive structure. This paper also introduces the node degree, diameter, number of links, bisection width of the octagon-cell network.

Keywords: octagon_cell; interconnection topology; routing; routing algorithm; network services

I. INTRODUCTION

An interconnection network can be viewed as an undirected graph in which the vertices correspond to processors and edges correspond to the bi-directional communication links between processing elements [1]. Routing of data employs routing algorithms. The performance on the routing algorithms depends on the routing algorithms adopted.

Routing algorithm is a key factor which affects the efficiency of the communication network in parallel computing. The performance of routing algorithm is influenced by many aspects like bandwidth of link, buffer capacities used at the bottleneck links, type of routing algorithms used that is single path or multipath etc [3]. Parallel processing elements provide high processing power for solving problems in major application areas such as image processing and scientific computing. Some features in the network of parallel machines are highly desirable such as minimal communication cost, efficient routing and the capability of embedding topological structures such as ring, linear array, tree, and mesh. Among the wide variety of interconnection networks structures proposed for parallel computing systems is hypercube which received much attention due to attractive properties inherited in their topology [2].

The embeddability, symmetry, stronger resilience, and simple routing have made hypercube superior to many other multicomputer networks. But in a hypercube structure, the number of communication ports and channels per processor increases with the increase of the network size which is a disadvantage of the structure. Therefore this study will be carried out, which can help to develop a new interconnection network to meet the demands of highly parallel systems [2].

The performance of routing algorithm also depends on the transport protocol used that is TCP (Transmission Control Protocol) and UDP (User Datagram Protocol)

II. BACKGROUND OF ROUTING ALGORITHM

Today with the fast growth of Internet everybody wants to get connected to the Internet. Millions of people use the Internet for daily business all over the world. Today the Internet has become very complex and dynamic. Failures and changes occur at every step because of traffic flow from one part of network to another part. Routing is the process of selecting paths in a network to send traffic. Routing is an important aspect of network communication which affects the performance of any network , since other characteristics of the network like throughput, reliability and congestion depend directly on it.

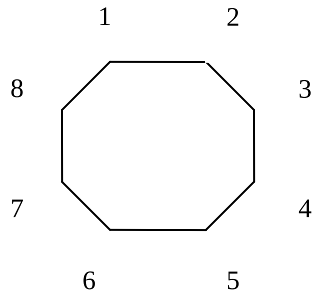
Parallel processing elements provide high processing power for solving problems in major application areas such as image processing and scientific computing [1,2,4,5,6,7]. The underlying communication network and the matching of the algorithm with the network structure strongly affect the performance of parallel machines. Hypercube is the interconnection network structure proposed for parallel computing system which received much attention due to the attractive properties inherited in their topology [2,8]. The embeddability, symmetry, regularity, strong resilience, and simple routing have made hypercube superior to many other multicomputer networks. But in a hypercube structure, the number of communication ports and channels per processor increases with the increase of network size which is a disadvantage of the structure [2]. In this paper we have described a new topological structure called octagon-cell network with optimal routing algorithm.

III. DESCRIPTION OF OCTAGON-CELL NETWORK

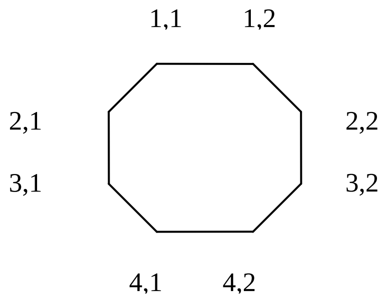
An interconnection network can be viewed as an undirected graph, in which vertices correspond to processors and edges correspond to the bidirectional communication links between processing elements [2].

In this paper we have proposed octagon-cell network. An octagon-cell has eight nodes. It has d levels numbered from

1 to d with depth d. Level 1 represents one octagon-cell. Level 2 represents eight octagon-cells surrounding the octagon-cell at level 1. Level 3 represents 16 octagon-cells surrounding the 8 octagon-cells at level 2 and so on.



Octagon-Cell level: 1



Addressing nodes in Octagon-Cell with level:1
(X,Y represents line no-X with node no-Y)

Figure 1

Each level I has N_i nodes, representing processing elements and interconnected in a ring structure. In a octagon-cell network, the number of nodes at level i is

$$N_i = 8(4i-3)$$

Now at level 1, $N_1 = 8$, since there is a single octagon-cell with 8 vertices. Level 2 introduces 8 octagon-cells. Therefore at level 2 the number of nodes

$N_2 = 8(4*2-3) = 8*5 = 40$, $N_3 = 8(4*3–3) = 8*9 = 72$

In octagon-cell the level (i+1) has 32 nodes in addition to corresponding nodes to those at level i. Therefore

$N_i = 8+(i-1)*32 = 8+32*i–32 = 32*i–24 = 8(4*i–3)$

The total number of nodes in a octagon-cell network is,

$$\mathbf{N} = \sum_{i=1}^{d} 8(4i–3) = 32\sum i -\sum 24 = 32\sum i–24\sum 1 = 32d (d+1)/2–24d = 16d^2– 8d$$

$$= 8d(2d-1)$$

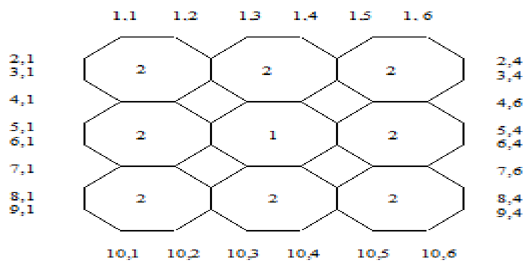
Or we can write N = 8i(2i-1),

Now N = $16d^2$–8d or $16d^2$ = N+8d or d^2 = N+8d/16 or d = 1/4 Sqrt(N+8d)

Therefore the total no of nodes at level 1 is N = 8(2*1-1) = 8

At level 2, N = 8(2*4-2) = 48

At level 3, N = 8(2*9-3) = 120 and so on.



Addressing nodes in Octagon-Cell with level:2

Here we have shown the addressing nodes of only 1^{st} and 10^{th} line but not all the lines, for Example 2,2 and 2,3 haven't been shown, because the figure will be more complex.

A. Diameter:

The diameter D of a network is defined as the maximum shortest path between any two nodes [2,9,10,11]. The path length is measured by the number of links traversed. The network diameter indicates the maximum number of distinct hops between any two hops between any two nodes. The network diameter should be as small as possible. It will not only reduce the traversing time for messages, but also minimize message density in the links of the network [2].

The diameter of octagon-cell is 4(2i-1)

At level 1 the diameter is 4(2i-1) = 4(2.1-1) = 4

At level 2 the diameter is 4(2i-1) = 4(2.2-1) = 12 and so on

The graph diameter verses number of nodes is drawn below.

Table I

Number of Nodes	*Diameter*
8	4
48	12
120	20
224	28
360	36

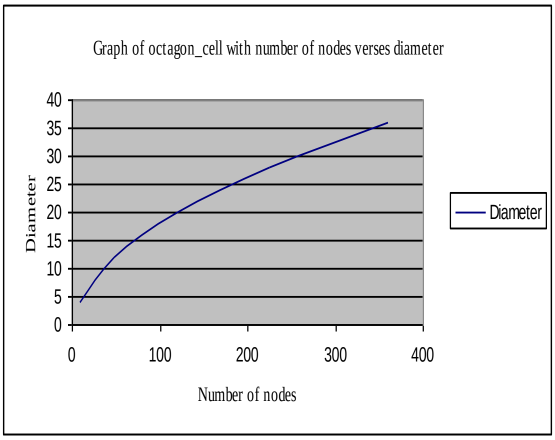


Figure 2

B. Node Degree:

The node degree on an interconnection network is defined as the maximum number of edges that a node can have in the network [2].

The node degree of octagon-cell with depth 1 is 2. If d > 1, then node degree remains constant, that is 3. The network topology which secures constant node degree is highly desirable. Constant node degree facilitates modularity in building blocks for scalable systems [2,9,10,11]. Therefore the node degree of octagon-cell is constant when d > 1.

The graph of octagon-cell with number of nodes verses node degree is shown below:

Table II

Number of Nodes	***Node Degree***
8	2
48	3
120	3
224	3

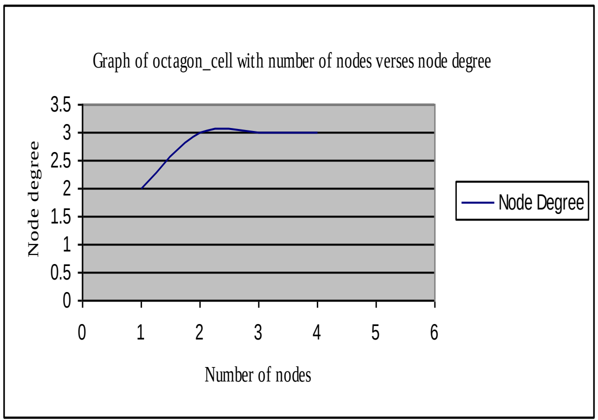


Figure 3

C. *Number of Links:*

The number of links at each level i is N_i = 8, 52, 100, 148….for levels 1, 2, 3, 4…..respectively.

The total number of links in octagon-cell are given by 8, 60, 160, 308….for levels 1, 2, 3, 4…..respectively.

The total number of links are given by the following formulas,

$N_1 = 8(2i^2 – 1)$, $N_2 = 8(2i^2 – 1) + 4$, $N_3 = 8(2i^2 – 1) + 24$, $N_4 = 8(2i^2 – 1) + 44$ ….for levels 1, 2, 3, 4…. respectively.

The graph of octagon-cell with number of nodes verses number of links is shown below:

Table III

Number of Nodes	***Number of links***
8	8
48	60
120	160
224	308

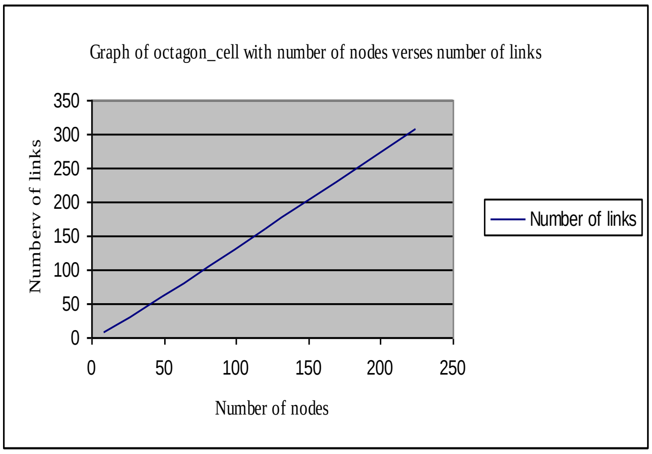


Figure 4

D. *Bisection Width:*

When a given network is cut into two equal halves the minimum number of edges along the cut is called the channel bisection width b [2,9,10,11]. The bisection width octagon-cell network is 2d.

IV. THE ROUTING ALGORITHM IN OCTAGON-CELL NETWORK

Routing is the function that allows information to be transmitted over a network from a source to a destination through a sequence of intermediate switching / buffering stations or nodes. Routing is necessary because in real systems not all nodes are directly connected. Routing algorithms can be classified as static or dynamic, and centralized or distributed. Centralised algorithms usually have scalability problems, or the inability of the network to recover in case of a failure in the central controlling station. Static routing assumes that network conditions are time-invariant, which is an unrealistic assumption in most of the cases. Adaptive routing schemes also have problems, including inconstistencies arising from node failures and potential oscillations that lead to circular paths and instability. Routing algorithms can also be classified as minimal or non-minimal. Minimal routing allows packets to follow only minimal cost paths, while non-minimal routing allows more flexibility in choosing the path by utilizing other heuristics [13,14,15].

Communication or data routing is the most fundamental function of interconnection networks. Data routing is the act of moving information across an interconnection network from a source to a destination [2,12].

In this paper we have developed optimal point-to-point routing algorithm for octagon-cell networks. An optimal routing algorithm is any network would route information or message from a source to a destination along a shortest path. The routing algorithm can be centralized or distributed. In centralized routing, a single processing element determines the shortest path from a given source node to a given destination node. Then the message is sent along that path. Finding the shortest path by a single central processing element is very slow and leads to a major bottleneck. If every source node takes the responsibility of computing the shortest path to the destination and sends that path along with the message to guide it through the intermediate nodes, the bottleneck problem is avoided, but traffic is increased and routing remains relatively slow [2].

In distributed routing however, all intermediate nodes on the shortest path cooperate to find the shortest path using destination address. Therefore each intermediate node needs only the destination address to determine which neighbor falls on the shortest path to a given destination. In the routing algorithm each intermediate node determines in constant time, which of its neighbors will receive the message.

The routing of message can be viewed as a sequence of changes made on the source address label to become the destination address label. These changes are done at every intermediate node on the path [2]. In octagon-cell network when the message is received by an intermediate node, it will consider itself as a new source.

We have developed a recursive routing algorithm for octagon-cell network. Due to its recursive structure routing can be done easily. In our algorithm we have used level numbering scheme. That is each node in octagon-cell is identified by a pair (X,Y). Where X denotes the line number in which the node exists and Y denotes serial number of the node in that line. A node with the address 1,1 is the 1st node in 1st line. A node with the address 1,2 is the 2nd node in 1st line and so on.

A. Notations used in the algorithm"

X_s Represents the address of source node in the horizontal direction / Line number of source node in horizontal direction.

Y_s Represents the address of source node in the vertical direction / Serial number of the source node in X_s line.

X_d Represents the address of destination node in the horizontal direction / Line number of destination node in horizontal direction.

Y_d Represents the address of destination node in the vertical direction / Serial number of destination node in X_d line.

X_s mod 3 is assigned by a value which is the remainder after dividing X_s by 3.

→ Arrows in our figures represent the destination point.

a. Darken circles in our figures represent the source point.

m Represents the line number of the octagon_cell network in horizontal direction.

&& Represents logical AND

|| Represents logical OR

Note: In the following algorithms, we have taken only integral parts of the quotients by neglecting the fractional parts where we have done division by using '/ ' (division symbol)

B. Case-1 Optimal Routing Algorithm for Horizontal Move for lines m where m mod 3 = 1 [Move from left to right , if ($X_s = X_d$ && $Y_s < Y_d$) and Move from right to left, if ($X_s = X_d$ && $Y_s > Y_d$)]

<u>Move (X_s,Y_s,X_d,Y_d)</u>
If ($y_s < y_d$)
If ($x_s = x_d$ && y_s is odd)
  Move (x_s, $y_s + 1$, x_d, y_d)
  Else
    If ($x_s = x_d$ && y_s is even)
      Move ($x_s + 1$, $y_s/2 + 1$, x_d, y_d)
      Else
        If ($x_s \neq x_d$)
          Move ($x_s - 1$, $2y_s - 1$, x_d, y_d)
Else
  Destination reached
Else
  If ($y_s > y_d$)
  If ($x_s = x_d$ && y_s is odd)
    Move (x_s+1, $y_s/2+1$, x_d, y_d)
    Else
      If ($x_s = x_d$ && y_s is even)
        Move (x_s, $y_s - 1$, x_d, y_d)
        Else
          If ($x_s \neq x_d$)
            Move (x_s-1, $2y_s-2$, x_d, y_d)
Else
  If ($y_s = y_d$)
    Go to Vertical Move
Else
  Destination reached

Example 1.1 Let (X_s,Y_s) = (1,3) and (X_d,Y_d) = (1,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

1,3 → 1,4 → 2,3 → 1,5 → 1,6 → 2,4 → 1,7 → 1,8

The shortest path length is 7

Example 1.2 Let (X_s,Y_s) = (4,8) and (X_d,Y_d) = (4,2) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

4,8 → 4,7 → 5,4 → 4,6 → 4,5 → 5,3 → 4,4 → 4,3 → 5,2 → 4,2

The shortest path length is 9

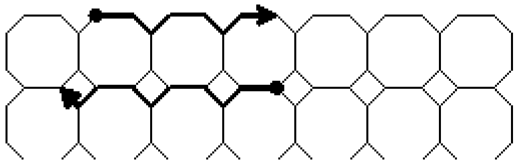


Figure - 5 [We have drawn only first 7 lines of the Octagon-Cell network of depth 4]

We have traced the optimal paths from the source node (1,3) to the destination node (1,8) & from the source node (4,8) to the destination node (4,2) shown by the deep lines.

C. Case-2 Optimal Routing Algorithm for Horizontal Move for lines m where m mod 3 ≠ 1.[Move from left to right, If ($X_s = X_d$ && $Y_s < Y_d$) and Move from right to left, if ($X_s = X_d$ && $Y_s > Y_d$)

<u>Move(X_s,Y_s, X_d,Y_d)</u>
If ($x_s = x_d$ && $y_s < y_d$ && x_s mod 3=2)
 If ($x_s = x_d$ && x_s mod 3=2)
  Move (x_s-1, $2y_s-1$,x_d, y_d)
   Else
    If ($x_s \neq x_d$ && y_s is odd)
      Move (x_s, $y_s + 1$, x_d, y_d)
       Else
        If ($x_s \neq x_d$ && y_s is even)
          Move ($x_s + 1$, $y_s/2+ 1$, x_d, y_d)
Else
  Destination reached
Else
  If ($x_s = x_d$ && $y_s < y_d$ && $x_s = 3n$)
  If ($x_s = x_d$ && $x_s = 3n$)
    Move (x_s+1, $2y_s-1$, x_d, y_d)
     Else
      If ($x_s \neq x_d$ && y_s is odd)
        Move (x_s, $y_s + 1$, x_d, y_d)
         Else
          If ($x_s \neq x_d$ && y_s is even)
            Move ($x_s - 1$, $y_s/2+ 1$, x_d, y_d)
Else
  Destination reached
Else
  If ($x_s = x_d$ && $y_s > y_d$ && x_s mod 3 = 2)
  If ($x_s = x_d$ && x_s mod 3 = 2)
    Move (x_s-1, $2y_s-2$, x_d, y_d)
     Else
      If ($x_s \neq x_d$ && y_s is even)
        Move (x_s, $y_s - 1$, x_d, y_d)
         Else
          If ($x_s \neq x_d$ && y_s is odd)
            Move ($x_s + 1$, $y_s/2+ 1$, x_d, y_d)
Else

Destination reached
Else
If ($x_s = x_d$ && $y_s > y_d$ && $x_s = 3n$)
If ($x_s = x_d$ && $x_s = 3n$)

Move (x_s+1, $2y_s-2$, x_d, y_d)
Else
If ($x_s \neq x_d$ && y_s is even)
Move (x_s, $y_s - 1$, x_d, y_d)
Else
If ($x_s \neq x_d$ && y_s is odd)
Move ($x_s - 1$, $y_s/2+ 1$, x_d, y_d)
Else
If ($y_s = y_d$)
Go to Vertical Move
Else
Destination reached

Example 2.1 Let $(X_s,Y_s) = (2,3)$ and $(X_d,Y_d) = (2,8)$ be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
2,3 → 1,5 → 1,6 → 2,4 → 1,7 → 1,8 → 2,5 → 1,9 → 1,10 → 2,6 → 1,11 → 1,12 → 2,7 → 1,13 → 1,14 → 2,8
The shortest path length is 15

Example 2.2 Let $(X_s,Y_s) = (5,9)$ and $(X_d,Y_d) = (5,6)$ be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
5,9 → 4,16 → 4,15 → 5,8 → 4,14 → 4,13 → 5,7 → 4,12 → 4,11 → 5,6
The shortest path length is 9

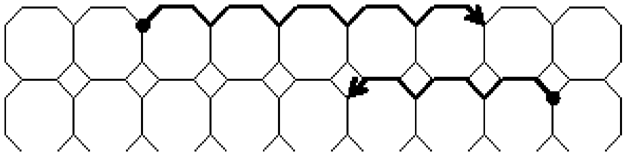


Figure – 6: [We have drawn only first seven lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (2,3) to the destination node (2,8) & from the source node (5,9) to the destination node (5,6) shown by the deep lines.

NOTE: In these algorithms first we check the main condition associated with If statement with the given source nodes and destination nodes. The calculated intermediate nodes x_s,y_s may violate the main condition. In example 2.1, given $y_s < y_d$. But one of the calculated intermediate node is 1,9 where $x_s = 1$, $y_s = 9$ and $y_s > y_d$. In this case the reader shouldn't jump to next group of algorithm for the condition $y_s > y_d$. In case 2 we have four group of algorithms with four conditions. The reader should calculate the intermediate nodes by using the same group algorithm.

D. Case-3 Optimal Routing Algorithm for Vertical Move for lines m = X_s where m mod 3 = 1.[Move from top to bottom, If ($X_s < X_d$ && $Y_s = Y_d$)]

Move (X_s, Y_s, X_d, Y_d)
If (y_d = 1 || 2) // Logical OR
For (i = x_s to x_d-1 in increasing order)
Move (x_s+1, y_s, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 ≠ 1 && y_s is odd)
Move(x_s, y_s+1, x_d, y_d)
Else
If ((x_s mod 3 = 1 && x_d mod 3 ≠ 1 && y_s is even && $x_s < x_d$) || (x_s mod 3 = 1 && x_d mod 3 = 1))
Move (x_s+1, $y_s/2+1$, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 ≠ 1 && y_s is even && $x_s > x_d$)
Move (x_s-1, $y_s/2+1$, x_d, y_d)
Else
If ((x_s mod 3 = 2 && x_d mod 3 ≠ 1 && $x_s \neq x_d$) || (x_s mod 3 = 2 && x_d mod 3 = 1))
Move (x_s+1, y_s, x_d, y_d)
Else
If ((x_s = 3n && x_d mod 3 ≠ 1) || (x_s = 3n && y_d is odd && x_d mod 3 = 1))
Move (x_s+1, $2y_s-1$, x_d, y_d)
Else
If (x_s = 3n && x_d mod 3 = 1 && y_d is even)
Move (x_s+1, $2y_s-2$, x_d, y_d)
Else
If ($x_s = x_d$)
Go to Horizontal Move
Else
Destination reached

Example 3.1 Let $(X_s,Y_s) = (1,10)$ and $(X_d,Y_d) = (9,10)$ be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
1,10 → 2,6 → 3,6 → 4,11 → 4,12 → 5,7 → 6,7 → 7,13 → 7,14 → 8,8 → 9,8 → 10,15 → 10,16 → 9,9 → 10,17 → 10,18 → 9,10
The shortest path length is 16

Example 3.2 Let $(X_s,Y_s) = (4,7)$ and $(X_d,Y_d) = (13,7)$ be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
4,7 → 5,4 → 6,4 → 7,7 → 8,4 → 9,4 → 10,7 → 11,4 → 12,4 → 13,7
The shortest path length is 9

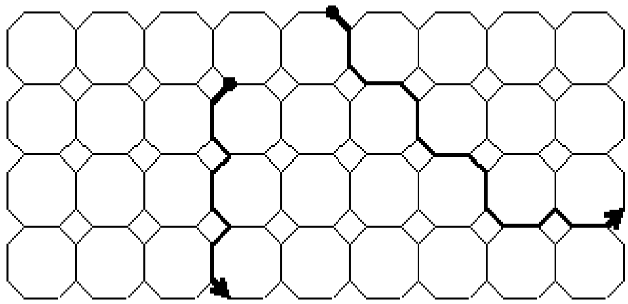


Figure – 7: [We have drawn only first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (1,10) to the destination node (9,10) and from source node (4,7) to the destination node (13,7) shown by the deep lines.

E. Case-4 Optimal Routing Algorithm for Vertical Move for lines m=X_s where m mod 3 = 1.[Move from bottom to top, If ($X_s > X_d$ && $Y_s = Y_d$)]

Move (X_s, Y_s, X_d, Y_d)
If (y_d = 1 || 2) // Logical OR
For (i = x_s to x_d-1 in decreasing order)
Move (x_s-1, y_s, x_d, y_d)
Else

If ((x_s mod 3 = 1 && x_d mod 3 = 1) || (x_s mod 3 = 1 && x_d mod 3 ≠ 1 && y_s is even && $x_s \neq x_d$))
Move(x_s-1, y_s/2+1, x_d, y_d)
Else
If ((x_s mod 3 = 1 && x_d mod 3 ≠ 1 && y_s is odd && $x_s \neq x_d$)
Move (x_s, y_s+1, x_d, y_d)
Else
If (x_s = 3n)
Move (x_s-1, y_s, x_d, y_d)
Else
If (x_s mod 3 = 2 && x_d mod 3 = 1 && y_d is even) || (x_s mod 3 = 2 && $y_s = y_d$)
Move (x_s-1, 2y_s-2, x_d, y_d)
Else
If (x_s mod 3 = 2 && x_d mod 3 ≠ 1 && $y_s \neq y_d$ && $x_s \neq x_d$)
Move (x_s-1, 2y_s-1, x_d, y_d)
Else
If (x_s mod 3 ≠ 1 && x_d mod 3 ≠ 1 && $y_s = y_d$)
Go to Vertical Move Bottom to Top for X_s mod 3 ≠ 1
Else
If (x_s mod 3 = 2 && x_d mod 3 = 1 && y_d is odd)
Move (x_s-1, 2y_s-1, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 ≠ 1 && $x_s = x_d$)
Go to Horizontal Move
Else
Destination reached

Example 4.1 Let (X_s,Y_s) = (10,5) and (X_d,Y_d) = (3,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
10,5 → 10,6 → 9,4 → 8,4 → 7,7 → 7,8 → 6,5 → 5,5 → 4,8 → 3,5
The shortest path length is 9

Example 4.2 Let (X_s,Y_s) = (13,15) and (X_d,Y_d) = (1,15) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
13,15 → 12,8 → 11,8 → 10,15 → 9,8 → 8,8 → 7,15 → 6,8 → 5,8 → 4,15 → 3,8 → 2,8 → 1,15
The shortest path length is 12

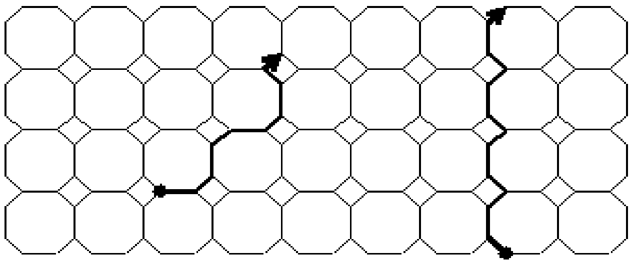


Figure – 8: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (10,5) to the destination node (3,5) and from source node (13,15) to the destination node (1,15) shown by the deep lines.

F. Case-5 Optimal Routing Algorithm for Vertical Move for lines m=X_s where X_s mod 3 ≠1.[Move from top to bottom, If ($X_s < X_d$ && $Y_s = Y_d$)]

Move (X_s, Y_s, X_d, Y_d)
If (y_d = 1 || 2) // Logical OR
For (i = x_s to x_d-1 in increasing order)
Move (x_s+1, y_s, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move(x_s+1, y_s, x_d, y_d)
Else
If (x_s = 3n)
Move (x_s+1, 2y_s-2, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 = 1 && y_s is even && $y_s \neq y_d$ && $x_s \neq x_d$)
Move (x_s, y_s-1, x_d, y_d)
Else
If ((x_s mod 3 = 1 && x_d mod 3 = 1 && y_s is odd && $y_s \neq y_d$ && $x_s \neq x_d$) || (x_s mod 3 = 1 && x_d mod 3 ≠ 1))
Move (x_s+1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 = 1 && $x_s = x_d$)
Go to Horizontal Move
Else
If (x_s mod 3 = 1 && x_d mod 3 = 1 && $y_s = y_d$)
Go to Vertical Move Top to Bottom for X_s mod 3 = 1
Else
Destination reached

Example 5.1 Let (X_s,Y_s) = (5,10) and (X_d,Y_d) = (11,10) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
5,10 → 6,10 → 7,18 → 8,10 → 9,10 → 10,18 → 11,10
The shortest path length is 6

Example 5.2 Let (X_s,Y_s) = (3,7) and (X_d,Y_d) = (11,7) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
3,7 → 4,12 → 5,7 → 6,7 → 7,12 →8,7 → 9,7 → 10,12 → 11,7
The shortest path length is 8

Example 5.3 Let (X_s,Y_s) = (5,6) and (X_d,Y_d) = (10,6) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
5,6 → 6,6 → 7,10 → 7,9 → 8,5 → 9,5 → 10,8 → 10,7 → 11,4 → 10,6
The shortest path length is 9

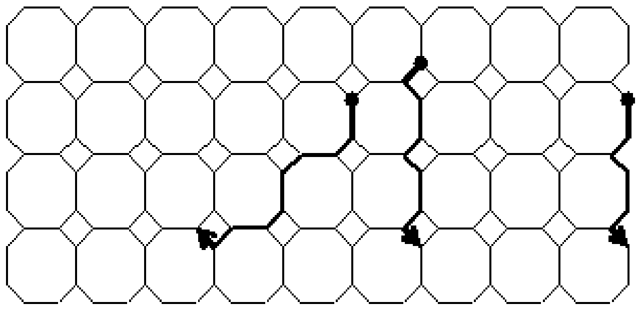


Figure – 9: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (5,10) to the destination node (11,10) and from source node (3,7) to the destination node (11,7) and also from the source node (5,6) to the destination node (10,6) shown by the deep lines.

G. Case-6 Optimal Routing Algorithm for Vertical move for lines m = X_s where m mod 3 ≠1.[Move from bottom to top, If ($X_s > X_d$ && $Y_s = Y_d$)]

Move (X_s, Y_s, X_d, Y_d)

If (y_d = 1 || 2) // Logical OR
For (i = x_s to x_d-1 in decreasing order)
Move (x_s-1, y_s, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move(x_s-1, 2y_s-2, x_d, y_d)
Else
If (x_s = 3n)
Move (x_s-1, y_s, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 = 1 && y_s is even && $y_s \neq y_d$ && $x_s \neq x_d$)
Move (x_s, y_s-1, x_d, y_d)
Else
If((x_s mod 3 = 1&& x_d mod 3 = 1 && y_s is odd && $y_s \neq y_d$ && $x_s \neq x_d$) || (x_s mod 3 = 1 && x_d mod 3 ≠ 1 && $x_s \neq x_d$))
Move (x_s-1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 1 && x_d mod 3 = 1 && $y_s = y_d$)
Go to Vertical Move Bottom to Top for X_s mod 3 = 1
Else
If ($x_s = x_d$)
Go to Horizontal Move
Else
Destination reached

Example 6.1 Let (X_s,Y_s) = (11,5) and (X_d,Y_d) = (4,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
11,5 → 10,8 → 10,7 → 9,4 → 8,4 → 7,6 → 7,5 → 6,3 → 5,3 → 4,5
The shortest path length is 9

Example 6.2 Let (X_s,Y_s) = (12,6) and (X_d,Y_d) = (4,6) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
12,6 → 11,6 → 10,10 → 10,9 → 9,5 → 8,5 → 7,8 → 7 ,7 → 6,4 → 5,4 → 4,6
The shortest path length is 10

Example 6.3 Let (X_s,Y_s) = (11,10) and (X_d,Y_d) = (5,10) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
11,10 → 10,18 → 9,10 → 8,10 → 7,18 → 6,10 → 5,10
The shortest path length is 6

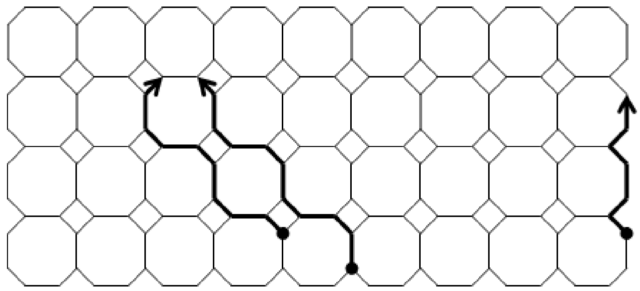


Figure – 10: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (20,5) to the destination node (13,5) and from source node (12,6) to the destination node (4,6) and from source node (11,10) to the destination node (5,10) shown by the deep lines.

H. Case-7 Optimal Routing Algorithm for $X_s \neq X_d$ && $Y_s \neq Y_d$ && $X_s < X_d$ && $Y_s < Y_d$

a. SUB CASE -7.1 (X_s mod 3 = 1 && X_d mod 3 = 1)

Move (X_s, Y_s, X_d, Y_d)

If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is odd)
Move(x_s, y_s+1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is even)
Move (x_s+1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s+1, y_s, x_d, y_d)
Else
If(x_s = 3n)
Move (x_s+1, 2y_s-1, x_d, y_d)
Else If ($x_s = x_d$)
Go to Horizontal Move
Else If ($y_s = y_d$)
Go to Vertical Move Top to Bottom
Else
Destination reached.

Example 7.1.1 Let (X_s,Y_s) = (1,11) and (X_d,Y_d) = (7,12) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
1,11 → 1,12 → 2,7 → 3,7 → 4,12 → 5,7 → 6,7 → 7,12
The shortest path length is 7

Example 7.1.2 Let (X_s,Y_s) = (4,6) and (X_d,Y_d) = (13,10) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
4,6 → 5,4 → 6,4 → 7,7 → 7,8 → 8,5 → 9,5 → 10,9 → 10,10 → 11,6 → 12,6 → 13,10
The shortest path length is 11

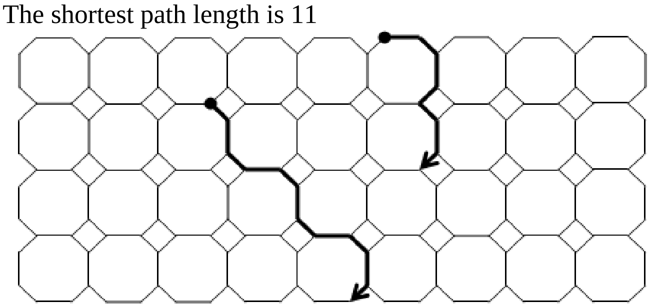


Figure – 11: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (1,11) to the destination node (7,12) and from source node (4,6) to the destination node (13,10) shown by the deep lines.

b. *SUBCASE -7.2 (X_s mod 3 = 1 && X_d mod 3 ≠ 1)*
Go to subcase-7.1

Example 7.2.1 Let (X_s,Y_s) = (7,3) and (X_d,Y_d) = (12,7) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
7,3 → 7,4 → 8,3 → 9,3 → 10,5 → 10,6 → 11,4 → 12,4 → 13,7 → 13,8 → 12,5 → 13,9 → 13,10 → 12,6 → 13,11 → 13,12 → 12,7
The shortest path length is 16
Example 7.2.2 Let (X_s,Y_s) = (1,1) and (X_d,Y_d) = (5,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
1,1 → 1,2 → 2,2 → 3,2 → 4,3 → 4,4 → 5,3 → 4,5 → 4,6 → 5,4 → 4,7 → 4,8 → 5,5
shortest path length is 12

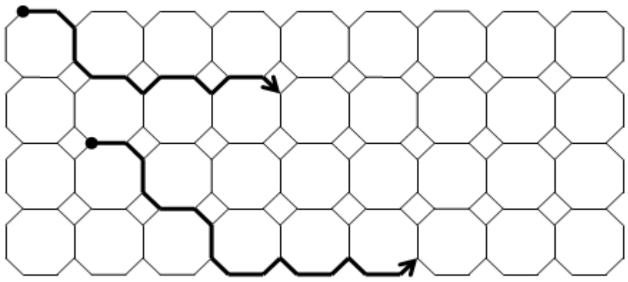


Figure – 12: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (7,3) to the destination node (12,7) and from source node (1,1) to the destination node (5,5) shown by the deep lines.

c. *SUBCASE -7.3 (X_s mod 3 ≠ 1 && X_d mod 3 = 1)*

If $(2y_s – 2) \geq y_d$ (Move to left)

<u>Move (X_s, Y_s, X_d, Y_d)</u>
If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is odd)
Move(x_s+1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is even)
Move (x_s,y_s-1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s+1, y_s, x_d, y_d)
Else
If(x_s = 3n)
Move (x_s+1, 2y_s-2, x_d, y_d)
Else If ($x_s = x_d$)
Go to Horizontal Move
Else If ($y_s = y_d$)
Go to Vertical Move Top top Bottom
Else
Destination reached.
Else If $(2y_s – 2) \leq y_d$ (Move to right)
Go to subcase-7.1
Example 7.3.1 Let (X_s,Y_s) = (3,7) and (X_d,Y_d) = (13,11) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
3,7 → 4,12 → 4,11 → 5,6 → 6,6 → 7,11 → 8,6 → 9,6 → 10,11 → 11,6 → 12,6 → 13,11
The shortest path length is 11
Example 7.3.2 Let (X_s,Y_s) = (2,4) and (X_d,Y_d) = (7,10) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
2,4 → 3,4 → 4,7 → 4,8 → 5,5 → 6,5 → 7,9 → 7,10
The shortest path length is 7

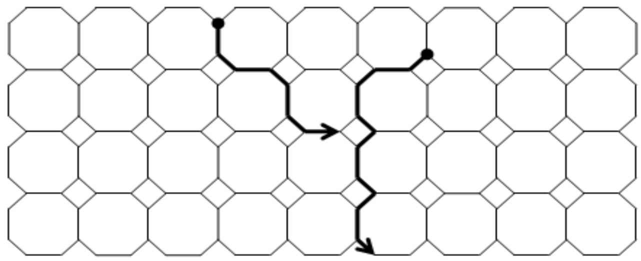


Figure – 13: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (3,7) to the destination node (13,11) and from source node (2,4) to the destination node (7,10) shown by the deep lines.

d. *SUBCASE -7.4 (X_s mod 3 ≠ 1 && X_d mod 3 ≠ 1)*

Go to subcase-7.1

Example 7.4.1 Let (X_s,Y_s) = (2,4) and (X_d,Y_d) = (8,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
2,4 → 3,4 → 4,7 → 4,8 → 5,5 → 6,5 → 7,9 → 7,10 → 8,6 → 7,11 → 7,12 → 8,7 → 7,13 → 7,14 → 8,8
The shortest path length is 14
Example 7.4.2 Let (X_s,Y_s) = (6,4) and (X_d,Y_d) = (14,7) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
6,4 → 7,7 → 7,8 → 8,5 → 9,5 → 10,9 → 10,10 → 11,6 → 12,6 → 13,11 → 13,12 → 14,7
The shortest path length is 11

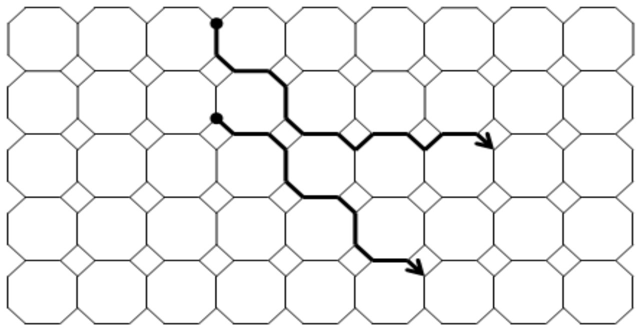


Figure – 14: [We have drawn the first 16 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (2,4) to the destination node (8,8) and from source node (6,4) to the destination node (14,7) shown by the deep lines.

I. Case-8 Optimal Routing Algorithm for $X_s \neq X_d$ && $Y_s \neq Y_d$ && $X_s < X_d$ && $Y_s > Y_d$

a. SUB CASE-8.1 (X_s mod 3 = 1 && X_d mod 3 = 1)

<u>Move (X_s, Y_s, X_d, Y_d)</u>
If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is odd)
Move(x_s+1, y_s/2+1, x_d, y_d)

Else
If (x_s mod 3 = 1&& y_s is even)
Move (x_s, y_s-1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s+1, y_s, x_d, y_d)
Else
If(x_s = 3n)
Move (x_s+1, 2y_s-2, x_d, y_d)
Else If (x_s = x_d)
Go to Horizontal Move
Else If (y_s = y_d)
Go to Vertical Move Top to Bottom
Else
Destination reached

Example 8.1.1 Let (X_s,Y_s) = (1,9) and (X_d,Y_d) = (7,3) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
1,9 → 2,5 → 3,5 → 4,8 → 4,7 → 5,4 → 6,4 → 7,6 → 7,5 → 8,3 → 7,4 → 7,3
The shortest path length is 11

Example 8.1.2 Let (X_s,Y_s) = (4,12) and (X_d,Y_d) = (13,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
4,12 → 4,11 → 5,6 → 6,6 → 7,10 → 7,9 → 8,5 → 9,5 → 10,8 → 11,5 → 12,5 → 13,8
The shortest path length is 11

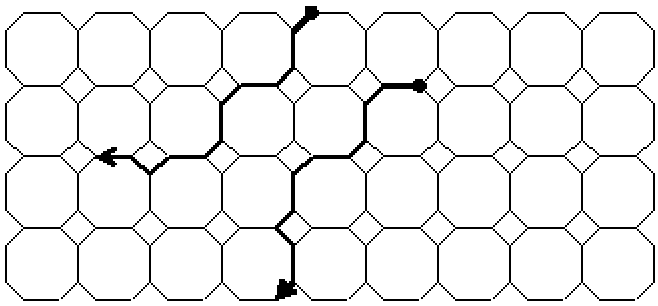


Figure – 15: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (1,9) to the destination node (7,3) and from source node (4,12) to the destination node (13,8) shown by the deep lines.

b. *SUB CASE-8.2 (X_s mod 3 = 1 && X_d mod 3 ≠ 1)*

If (2y_d -1) > y_s (Move left to right)
Move (X_s, Y_s, X_d, Y_d)
If (x_s ≠ x_d && y_s ≠ y_d)
If (x_s mod 3 = 1 && y_s is even)
Move(x_s+1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is odd)
Move (x_s, y_s+1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s+1, y_s, x_d, y_d)
Else
If(x_s = 3n)
Move (x_s+1, 2y_s-1, x_d, y_d)
Else If (x_s = x_d)
Go to Horizontal Move
Else If (y_s = y_d)
Go to Vertical Move Top to Bottom
Else
Destination reached
Else If (2y_d -1) ≤ y_s (Move right to left)
Go to subcase-8.1

Example 8.2.1 Let (X_s,Y_s) = (4,10) and (X_d,Y_d) = (8,7) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
4,10 → 5,6 → 6,6 → 7,11 → 7,12 → 8,7
The shortest path length is 5

Example 8.2.2 Let (X_s,Y_s) = (4,8) and (X_d,Y_d) = (8,3) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
4,8 → 4,7 → 5,4 → 6,4 → 7,6 →7,5 → 8,3
The shortest path length is 6

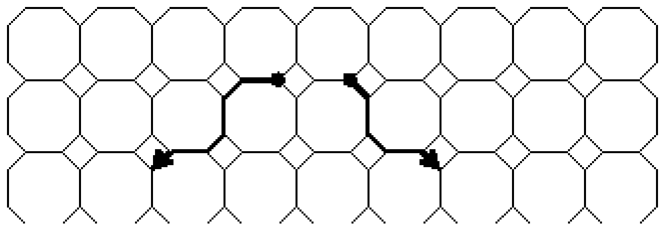


Figure – 16: [We have drawn the first 10 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (4,10) to the destination node (8,7) and from source node (4,8) to the destination node (8,3) shown by the deep lines.

c. *SUB CASE-8.3 (X_s mod 3 ≠ 1 && X_d mod 3 = 1)*

Go to subcase-8.1

Example 8.3.1 Let (X_s,Y_s) = (2,10) and (X_d,Y_d) = (10,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
2,10 → 3,10 → 4,18 → 4,17 → 5,9 → 6,9 → 7,16 → 7,15 → 8,8 → 9,8 → 10,14 → 10,13 → 11,7 → 10,12 → 10,11 → 11,6 → 10,10 → 10,9 → 11,5 → 10,8
The shortest path length is 19

Example 8.3.2 Let (X_s,Y_s) = (6,6) and (X_d,Y_d) = (7,3) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
6,6 → 7,10 → 7,9 → 8,5 → 7,8 → 7,7 → 8,4 → 7,6 → 7,5 → 8,3 → 7,4 → 7,3
The shortest path length is 11

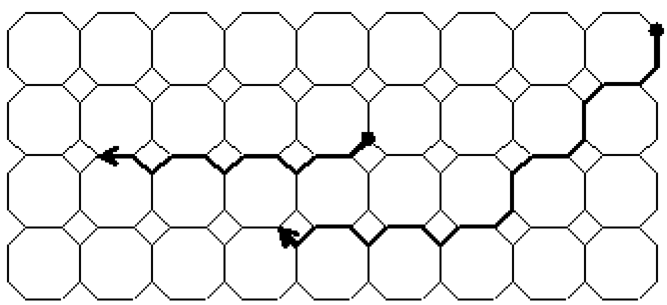


Figure -17: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (2,10) to the destination node (10,8) and from source node (6,6) to the destination node (7,3) shown by the deep lines.

d. SUB CASE-8.4 (X_s mod 3 ≠ 1 && X_d mod 3 ≠ 1)

Go to subcase-8.1

Example 8.4.1 Let (X_s,Y_s) = (2,8) and (X_d,Y_d) = (6,7) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

2,8 → 3,8 → 4,14 → 4,13 → 5,7 → 6,7

The shortest path length is 5

Example 8.4.2 Let (X_s,Y_s) = (2,6) and (X_d,Y_d) = (6,3) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

2,6 → 3,6 → 4,10 → 4,9 → 5,5 →6,5 → 7,8 → 7,7 → 6,4 → 7,6→7,5→6,3

The shortest path length is 11

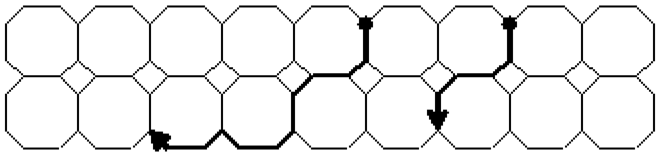


Figure – 18: [We have drawn the first 7 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (2,8) to the destination node (6,7) and from source node (2,6) to the destination node (6,3) shown by the deep lines.

J. Case-9 Optimal Routing Algorithm for $X_s \neq X_d$ && $Y_s \neq Y_d$ && $X_s > X_d$ && $Y_s < Y_d$

a. SUB CASE-9.1 (X_s mod 3 = 1 && X_d mod 3 = 1)

Move (X_s, Y_s, X_d, Y_d)

If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is odd)
Move(x_s, y_s+1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is even)
Move (x_s-1, y_s/2+1, x_d, y_d)
Else
If (x_s = 3n)
Move (x_s-1, y_s, x_d, y_d)
Else
If(x_s mod 3 = 2)
Move (x_s-1, 2y_s-1, x_d, y_d)
Else If ($x_s = x_d$)
Go to Horizontal Move
Else If ($y_s = y_d$)
Go to Vertical Move Bottom to Top
Else
Destination reached

Example 9.1.1 Let (X_s, Y_s) = (13,10) and (X_d, Y_d) = (1,16) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

13,10 → 12,6 → 11,6 → 10,11 → 10,12 → 9,7 → 8,7 → 7,13 → 7,14 → 6,8 → 5,8 → 4,15 → 4,16 → 3,9 → 2,9 → 1,16

The shortest path length is 15

Example 9.1.2 Let (X_s, Y_s) = (16,2) and (X_d, Y_d) = (4,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

16,2 → 15,2 → 14,2 → 13,3 → 13,4 → 12,3 → 11,3 → 10,5 → 10,6 → 9,4 → 8,4 → 7,7 → 7,8 → 6,5 → 5,5 → 4,8

The shortest path length is 15

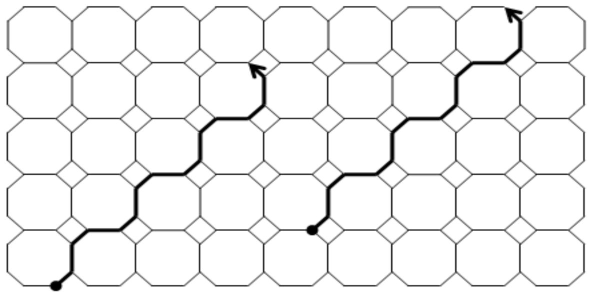


Figure – 19: [We have drawn the first 16 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (13,10) to the destination node (1,16) and from source node (16,2) to the destination node (4,8) shown by the deep lines.

b. SUB CASE-9.2 (X_s mod 3 = 1 && X_d mod 3 ≠ 1)

Go to subcase - 9.1

Example 9.2.1 Let (X_s,Y_s) = (7,3) and (X_d, Y_d) = (2,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

7,3 → 7,4 → 6,3 → 5,3 → 4,5 → 4,6 → 3,4 → 2,4 → 1,7 → 1,8 → 2,5 → 1,9 → 1,10 → 2,6 → 1,11 → 1,12 → 2,7 → 1,13 → 1,14 → 2,8

The shortest path length is 19

Example 9.2.2 Let (X_s,Y_s) = (7,5) and (X_d,Y_d) = (3,9) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:

7,5 → 7,6 → 6,4 → 5,4 → 4,7 → 4,8 → 3,5 → 4,9 → 4,10 → 3,6 → 4,11 → 4,12 → 3,7 → 4,13 → 4,14 → 3,8 → 4,15 → 4,16 → 3,9

The shortest path length is 18

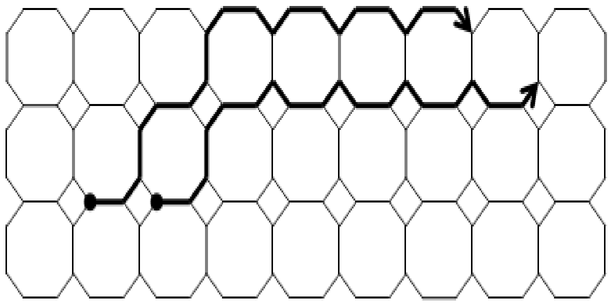


Figure – 20: [We have drawn the first 10 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (7,3) to the destination node (2,8) and from source node (7,5) to the destination node (3,9) shown by the deep lines.

c. SUB CASE-9.3 (X_s mod 3 ≠ 1 && X_d mod 3 =1)

If (2y_s -1) ≥ y_d (Move right to left)

Move (X_s, Y_s, X_d, Y_d)

If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is even)
Move(x_s, y_s-1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is odd)
Move (x_s-1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s-1,2y_s-2, x_d, y_d)
Else

If(x_s = 3n)
Move (x_s-1, y_s, x_d, y_d)
Else If ($x_s = x_d$)
Go to Horizontal Move
Else If ($y_s = y_d$)
Go to Vertical Move
Else
Destination reached
Else If ($2y_s$ -1) ≤ y_d (Move left to right)
Go to subcase-9.1

Example 9.3.1 Let (X_s,Y_s) = (14,10) and (X_d,Y_d) = (4,13) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
14,10 → 13,18 → 13,17 → 12,9 → 11,9 → 10,16 → 10,15 → 9,8 → 8,8 → 7,14 → 7,13 → 6,7 → 5,7 → 4,13
The shortest path length is 13

Example 9.3.2 Let (X_s,Y_s) = (15,3) and (X_d,Y_d) = (1,12) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
15,3 → 14,3 → 13,5 → 13,6 → 12,4 → 11,4 → 10,7 → 10,8 → 9,5 → 8,5 → 7,9→ 7,10 → 6,6 → 5,6 → 4,11 → 4,12 → 3,7 → 2,7 → 1,12
The shortest path length is 18

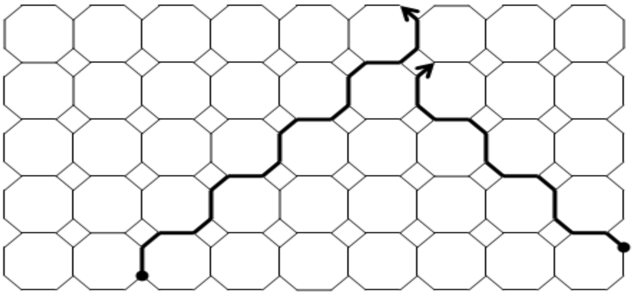


Figure – 21: [We have drawn the first 10 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (14,10) to the destination node (4,13) and from source node (15,3) to the destination node (1,12) shown by the deep lines.

d. *SUB CASE-9.4 (X_s mod 3 ≠ 1 && X_d mod 3 ≠ 1)*

Go to subcase - 9.1

Example 9.4.1 Let (X_s,Y_s) = (11,5) and (X_d,Y_d) = (3,6) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
11,5 → 10,9 → 10,10 → 9,6 → 8,6 → 7,10 → 6,6 → 5,6 → 4,10 → 3,6
The shortest path length is 9

Example 9.4.2 Let (X_s,Y_s) = (6,3) and (X_d,Y_d) = (2,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
6,3 → 5,3 → 4,5 → 4,6 → 3,4 → 2,4 → 1,7 → 1,8 → 2,5 → 1,9 → 1,10 → 2,6 → 1,11 → 1,12 → 2,7 → 1,13 → 1,14 → 2,8
The shortest path length is 17

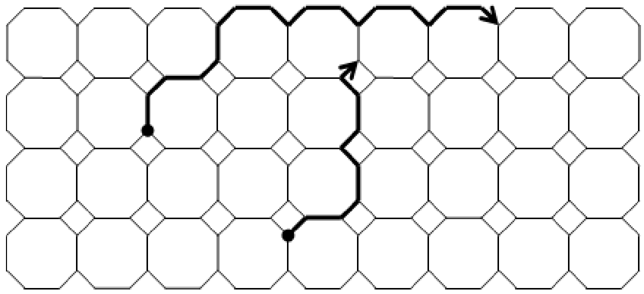


Figure – 22: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (11,5) to the destination node (3,6) and from source node (6,3) to the destination node (2,8) shown by the deep lines.

K. *Case-10 Optimal Routing Algorithm for $X_s \neq X_d$ && $Y_s \neq Y_d$ && $X_s > X_d$ && $Y_s > Y_d$*

a. *SUB CASE-10.1 (X_s mod 3 = 1 && X_d mod 3 = 1)*

Go to sub case 9.3 [Move right to left]

Example 10.1.1 Let (X_s,Y_s) = (10,16) and (X_d,Y_d) = (1,8) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
10,16 → 10,15 → 9,8 → 8,8 → 7,14 → 7,13 → 6,7 → 5,7 → 4,12 → 4,11 → 3,6 → 2,6 → 1,10 → 1,9 → 2,5 → 1,8
The shortest path length is 15

Example 10.1.2 Let (X_s,Y_s) = (13,5) and (X_d,Y_d) = (4,1) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
13,5 → 12,3 → 11,3 → 10,4 → 10,3 → 9,2 → 8,2 → 7,2 → 7,1 → 6,1 → 5,1 → 4,1
The shortest path length is 11

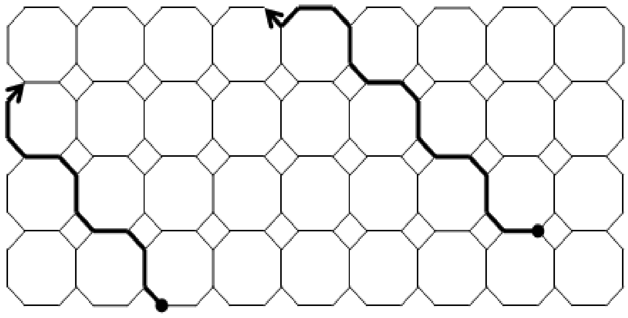


Figure – 23: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (10,16) to the destination node (1,8) and from source node (13,5) to the destination node (4,1) shown by the deep lines.

b. *SUB CASE-10.2 (X_s mod 3 = 1 && X_d mod 3 ≠ 1)*

If ($2y_d$ -1) > y_s (Move to right)
Move (X_s, Y_s, X_d, Y_d)
If ($x_s \neq x_d$ && $y_s \neq y_d$)
If (x_s mod 3 = 1 && y_s is odd)
Move(x_s, y_s+1, x_d, y_d)
Else
If (x_s mod 3 = 1&& y_s is even)
Move (x_s-1, y_s/2+1, x_d, y_d)
Else
If (x_s mod 3 = 2)
Move (x_s-1,$2y_s$-1, x_d, y_d)
Else

If(x_s = 3n)
Move (x_s-1, y_s, x_d, y_d)
Else If ($x_s = x_d$)
Go to Horizontal Move
Else If ($y_s = y_d$)
Go to Vertical Move
Else
Destination reached
If ($2y_d$-1) ≤ y_s [Move right to left]
Go to sub case 9.3

Example 10.2.1 Let (X_s,Y_s) = (10,16) and (X_d,Y_d) = (2,10) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
10,16 → 9,9 → 8,9 → 7,17 → 7,18 → 6,10 → 5,10 → 4,18 → 3,10 → 2,10
The shortest path length is 9

Example 10.2.2 Let (X_s,Y_s) = (7,5) and (X_d,Y_d) = (2,1) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
7,5 → 6,3 → 5,3 → 4,4 → 4,3 → 3,2 → 2,2 → 1,2 → 1,1 → 2,1
The shortest path length is 9

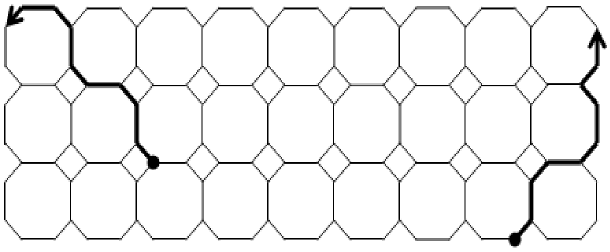


Figure – 24: [We have drawn the first 10 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (10,16) to the destination node (2,10) and from source node (7,5) to the destination node (2,1) shown by the deep lines.

c. *SUB CASE-10.3 (X_s mod 3 ≠ 1 && X_d mod 3 = 1)*

Go to sub case 9.3 [Move right to left]

Example 10.3.1 Let (X_s,Y_s) = (12,7) and (X_d,Y_d) = (7,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
12,7 → 11,7 → 10,12 → 10,11 → 9,6 → 8,6 → 7,10 → 7,9 → 8,5 → 7,8 → 7,7 → 8,4 → 7,6 → 7,5
The shortest path length is 13

Example 10.3.2 Let (X_s,Y_s) = (8,10) and (X_d,Y_d) = (4,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
8,10 → 7,18 → 7,17 → 6,9 → 5,9 → 4,16 → 4,15 → 5,8 → 4,14 → 4,13 → 5,7 → 4,12 → 4,11 → 5,6 → 4,10 → 4,9 → 5,5 → 4,8 → 4,7 → 5,4 → 4,6 → 4,5
The shortest path length is 21

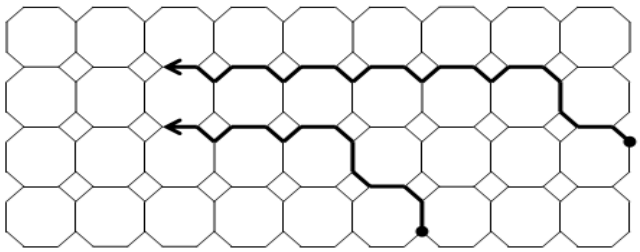


Figure – 25: [We have drawn the first 13 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (12,7) to the destination node (7,5) and from source node (8,10) to the destination node (4,5) shown by the deep lines.

d. *SUB CASE-10.4 (X_s mod 3 ≠ 1 && X_d mod 3 ≠ 1)*

Go to sub case -9.3 [Move right to left]

Example 10.4.1 Let (X_s, Y_s) = (8,7) and (X_d, Y_d) = (3,4) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
8,7 → 7,12 → 7,11 → 6,6 → 5,6 → 4,10 → 4,9 → 3,5 → 4,8 → 4,7 → 3,4
The shortest path length is 10

Example 10.4.2 Let (X_s,Y_s) = (6,8) and (X_d,Y_d) = (2,5) be the source and destination nodes respectively. To reach the destination the intermediate nodes for optimal paths are:
6,8 → 5,8 → 4,14 → 4,13 → 3,7 → 2,7 → 1,12 → 1,11 → 2,6 → 1,10 → 1,9 → 2,5
The shortest path length is 11

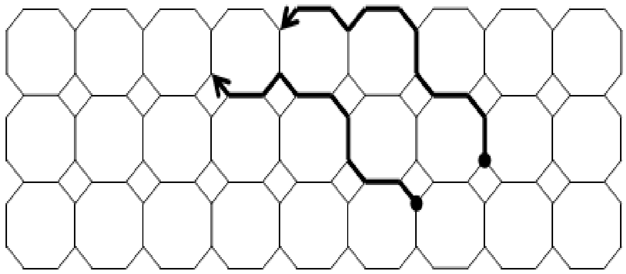


Figure – 26: [We have drawn the first 10 lines of the Octagon-Cell network of depth 5]

We have traced the optimal paths from the source node (8,7) to the destination node (3,4) and from source node (6,8) to the destination node (2,5) shown by the deep lines.

V. CONCLUSION

Communication or data routing is the most fundamental function of interconnection networks. Data routing is the act of moving information across an interconnection network from a source to a destination. In this paper a new topological structure called octagon-cell network has been developed and also we have derived a simpler and an efficient optimal point to point routing algorithm for this octagon-cell network.

VI. ACKNOWLEDGMENT

This study is based on the success of data routing which is the most fundamental function of interconnection networks. This optimal routing algorithm is the one of the most valuable researches to accomplish the various problems related to them.

VII. REFERENCES

[1]. Boxer, L. and Miller, R., 1998. "Dynamic Computational Geometry on Meshes and Hypercubes", Proceedings of the 1988 International Conference on Parallel Processing, st. Charles, lllinois, pp. 323-330.

[2]. Ahmad Sharieh, Mohammad Qatawneh, Wesam Almobaideen, Azzam Sleit, "Hex-Cell: Modeling,Topological Properties and Routing

Algorithm", European Journal of Scientific Research, ISSN 1450-216X Vol.22 No.2 (2008), pp. 457-468.

[3]. ER.Yashpaul Singh and A.Swarup, "Analysis of Equal Cost Adaptive Routing Algorithms using Connection-oriented and Connection less protocols", World Academy of Science,Engineering and Technology 51 2009,pp. 299-302.

[4]. Catherine Decayeux and David Seme, 2005. "3D Hexagonal Network: Modeling, Topological properties, Addressing scheme, and Optimal Routing Algorithm", IEEE Transaction on parallel and distributed Systems, Vol. 16, No. 9, pp 875-884.

[5]. Parhami, 2005. "Computer Architecture: From Microprocessors to Supercomputers", Oxford.

[6]. Qatawneh Mohammed, 2006. "Adaptive Fault Tolerant Routing algorithm for Tree-Hypercube Multicomputer', Journal of Computer Science 2 (2): pp 124-126.

[7]. Y Saad and M H Schultz, 1998. "Topological Properties of Hypercubes", IEEE Transaction Computers, Vol. 37, pp 867

[8]. Chan, T. F. and Saad, Y., 2002. "Multigrid Algorithms on the Multiprocessor", IEEE Transactions on Computers, Vol. C-35, No. 11, November, pp. 969-977.

[9]. Della Vecchia, G. and C. Sanges, 1995. "A Recursively Scalable Network for VLSI Implementation", Future Generation Computer Systems, pp.235-243

[10]. Kai Hwang, 1993. "Advanced Computer Architecture: Parallelism, Scalability, Programmability:, McGraw-Hill Book Co. International Edition.

[11]. Parhami, B., 1999. "Introduction to parallel Processing: Algorithms and Architectures", Plenum.

[12]. Ghose, K. and K.R. Desai, 1995. "Hierarchical Cubic Network", IEEE Transaction Parallel and Distributed Systems, Vol. 6, No. 4, pp. 427-435.

[13]. J. Bosh, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proceedings of 4th Annual AAM/IEEE International Conference On Mobile Computing and Networking , 1998.

[14]. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, Ant-based Load Balancing in telecommunications Networks, Adaptive Behavior, Vol. 5, pp. 169-207, 1997.

[15]. C. K. Toh, Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless D Hoc Networks, IEEE Communications Magazine, pp. 138-147, 2001