

DISTRIBUTED SYSTEMS LABORATORY

ASSIGNMENT 1,2,3

NAME:SUDHANSU SEKHAR SWAIN

ROLL:118CS0689

LAB 1

AIM OF THE EXP.: Design a network topology (Star, Ring, Tree) with arbitrary number of nodes. Determine the adjacency matrix, list and degree of nodes.

PROCEDURE(PSEUDOCODE):

1. Declare two variables numNodes (integer type) and topologyType (enum type) and take them as input.
from the user
2. If the topologyType "STAR" is selected,
 - 2.1. For each i from 1 to n-1
 - i. Add an edge between node i and node 0.
3. If the topologyType "RING" is selected,
 - 3.1. For each i from 0 to n-1
 - i. Add an edge between node i and node $((i \% n + 1) \% n)$
 - ii. Add an edge between node i and node $((i \% n - 1) \% n + n) \% n$
4. If the topologyType "TREE" is selected,
 - 4.1. For i from 0 to n-1
 - i. If $i * 2 + 1 < n$, add an edge between node i and node $i * 2 + 1$
 - ii. If $i * 2 + 2 < n$, add an edge between node i and node $i * 2 + 2$
5. Output the Adjacency List, Adjacency Matrix and Degree for each node.
6. END

RESULTS: The screenshots of user inputs/outputs is presented below

```
C:\Users\KANHU\Downloads\Q1.exe
Enter the number of nodes: 6
Enter the topology type: 0(RING),1(STAR),2(TREE) : 0
The adjacency list for the given input is
0 : 1 5
1 : 2 0
2 : 3 1
3 : 4 2
4 : 5 3
5 : 0 4

The adjacency matrix for the given input is
0 1 0 0 0 1
1 0 1 0 0 0
0 1 0 1 0 0
0 0 1 0 1 0
0 0 0 1 0 1
1 0 0 0 1 0

The degrees of the nodes for the given input is
0 : 2
1 : 2
2 : 2
3 : 2
4 : 2
5 : 2

-----
Process exited after 10.99 seconds with return value 0
Press any key to continue . . .

C:\Users\KANHU\Downloads\Q1.exe
Enter the number of nodes: 6
Enter the topology type: 0(RING),1(STAR),2(TREE) : 1
The adjacency list for the given input is
0 : 1 2 3 4 5
1 : 0
2 : 0
3 : 0
4 : 0
5 : 0

The adjacency matrix for the given input is
0 1 1 1 1 1
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0
1 0 0 0 0 0

The degrees of the nodes for the given input is
0 : 5
1 : 1
2 : 1
3 : 1
4 : 1
5 : 1

-----
Process exited after 6.343 seconds with return value 0
Press any key to continue . . .

C:\Users\KANHU\Downloads\Q1.exe
Enter the number of nodes: 6
Enter the topology type: 0(RING),1(STAR),2(TREE) : 2
The adjacency list for the given input is
0 : 1 2
1 : 0 3 4
2 : 0 5
3 : 1
4 : 1
5 : 2

The adjacency matrix for the given input is
0 1 1 0 0 0
1 0 0 1 1 0
1 0 0 0 0 1
0 1 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0

The degrees of the nodes for the given input is
0 : 2
1 : 3
2 : 2
3 : 1
4 : 1
5 : 1

-----
Process exited after 7.875 seconds with return value 0
Press any key to continue . . .
```

CONCLUSION:

The Adjacency Lists, Adjacency Matrices and The Degrees of Nodes have been output for different network topologies (namely STAR, RING and TREE). The time complexity of computation is $O(n)$ but the overall time complexity is $O(n^2)$, due

to the formation of the adjacency matrix. The maximum n that can be input without exceeding the memory limit is ~ 10000 .

LAB 2

AIM: Design a hypercube and hybrid topology (Combination of star, ring, tree topology). Determine the adjacency list, adjacency matrix, and the degree of a node.

PROCEDURE(ALGORITHM):

1. Take as input the topologyType (Enum).
2. If the topologyType == HYPERCUBE:
 - a. Take the number of bits(n) to represent the topology(degree).
 - b. The number of nodes in that topology is 2^n
 - c. For i from 0 to 2^n-1 ,
 - i. For j from 0 to n ,
 1. Join the node i with node $i \text{ XOR } (2^j)$.
3. If the topologyType == HYBRID:
 - a. Take the number of nodes(n) and number of edges (e) as input.
 - b. If the $e > n(n-1)/2$, throw error as a simple topology is not possible to be built
 - c. Else
 - i. Take the nodes a and b as input
 - ii. Add an edge between a and b .

4. Output the Adjacency List, Adjacency Matrix and Degree for each node.

5. END

RESULT:

Hybrid Topology

```
C:\Users\KANHU\Downloads\Hybrid.exe
Enter number of nodes in NETWORK :
9
Enter number of edges in NETWORK :
9
0 1
0 2
0 3
1 2
3 5
3 6
3 4
6 7
6 8
The adjacency list for the given input is
0 : 1 2 3
1 : 0 2
2 : 0 1
3 : 0 5 6 4
4 : 3
5 : 3
6 : 3 7 8
7 : 6
8 : 6
The adjacency matrix for the given input is
0 1 1 1 0 0 0 0 0
1 0 1 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0
1 0 0 0 1 1 1 0 0
0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 1 1
0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0
The degrees of the nodes for the given input is
0 : 3
1 : 2
2 : 2
3 : 4
4 : 1
5 : 1
6 : 3
7 : 1
```

Activate Windows
Go to Settings to activate Windows.

Hypercube Topology:

```
C:\Users\KANHU\Downloads\HyperCube.exe
Enter the bits to represent nodes : 4
Enter the topology type: 0(HYPERCUBE),1(HYBRID) : 0
The adjacency list for the given input is
0 : 1 2 4 8
1 : 0 3 5 9
2 : 3 0 6 10
3 : 2 1 7 11
4 : 5 6 0 12
5 : 4 7 1 13
6 : 7 4 2 14
7 : 6 5 3 15
8 : 9 10 12 0
9 : 8 11 13 1
10 : 11 0 14 2
11 : 10 9 15 3
12 : 13 14 8 4
13 : 12 15 9 5
14 : 15 12 10 6
15 : 14 13 11 7

The adjacency matrix for the given input is
0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0
1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0
1 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0
0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0
0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0
0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0
0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0
0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0
0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1
0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1
0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0
0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0

The degrees of the nodes for the given input is
0 : 4
1 : 4
2 : 4
3 : 4
4 : 4
```

CONCLUSION:

The Adjacency Lists, Adjacency Matrices and The Degrees of Nodes have been output for different network topologies (namely HYPERCUBE and HYBRID). The worst-case time complexity is $O(2^n)$. The maximum n that can be input without exceeding the memory limit is ~ 30 .

LAB-3

AIM: Design a weighted hybrid topology and determine the shortest path between random source and destination using Dijkstra's algorithm.

PROCEDURE(ALGORITHM):

1. Take the number of edges (e) and the number of nodes(n) as input
2. For i from 0 to $e-1$:

- a. Take as input two nodes a, b and the weight of the edge between them (d).
 - b. Add an edge between a and b, with weight d.
3. Take as input the source node(src) and the destination node(dest)
4. Create a min priority queue initially empty(pq)
5. Maintain a map of the distance(distanceRecord) of each node. Initially, distanceRecord(src) = 0 and for all other nodes it is +infinity.
6. Push the src node along with distanceRecord(src) to pq.
7. While pq is not empty:
 - a. Pop the top node from pq. Let it be currentNode.
 - b. For all nodes adjacent to currentNode, (adjacentNode):
 - i. If $\text{distanceRecord}(\text{currentNode}) + \text{weight}(\text{edge between currentNode and adjacentNode}) < \text{distanceRecord}(\text{adjacentNode})$
 1. Remove the adjacentNode from pq, if already present
 2. Update the $\text{distanceRecord}(\text{adjacentNode}) = \text{distanceRecord}(\text{currentNode}) + \text{weight}(\text{edge between currentNode and adjacentNode})$.
 3. Push the adjacentNode along with distanceRecord(adjacentNode) to pq.
8. Return the distanceRecord(dest) as the output
9. END

RESULT:

```
C:\Users\KANHU\Downloads\WeightedHybrid.exe
Enter number of nodes in NETWORK :
9
Enter number of edges in NETWORK :
14
0 1 4
1 7 11
0 7 8
1 2 8
7 8 7
2 8 2
7 6 1
0 6 6
2 5 4
6 5 2
2 3 7
3 4 9
3 5 14
5 4 10
The adjacency list for the given input is
0 : 1 7
1 : 0 7 2
2 : 1 8 5 3
3 : 2 4 5
4 : 3 5
5 : 2 6 3 4
6 : 7 8 5
7 : 1 0 8 6
8 : 7 2 6
The adjacency matrix for the given input is
0 1 0 0 0 0 0 1 0
1 0 1 0 0 0 0 1 0
0 1 0 1 0 1 0 0 1
0 0 1 0 1 1 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 1 1 0 1 0 0
0 0 0 0 1 1 0 1 1
1 1 0 0 0 0 1 0 1
0 0 1 0 0 0 1 1 0
The degrees of the nodes for the given input is
0 : 2
1 : 3
2 : 4
3 : 3
4 : 2
5 : 4
6 : 3
7 : 4
8 : 3
Enter SOURCE and DESTINATION
0 7
8
```

CONCLUSION: The shortest path between any source and destination in a network having Hybrid Topology was calculated using the Dijkstra's Shortest Path algorithm, assuming that there is no Negative edges in the graph. The time complexity of the algorithm is $O(V+E\log(V))$.