Finding of Shortest Path from Source to Destination by Traversing every Node in wired Network.

Avdhesh kumar sharma[1]
M.TECH IVth Sem (CSE), Swami Vivekanand Subharti University, Meerut, U.P., India

Sourabh kumar[2]
CSE Deptt, Swami Vivekananda Subharti University, Meerut, U.P., India
sharma.avdhesh164@gmail.com

1. Abstract

At present communication and transportation has an important place in every person life in their all business and non-business task. In daily life everybody is facing a problem of choosing a shortest path from one location to another location. Shortest path means the path which has minimum mileage or distance covered. It saves time and money both which are essential parts of our life. In this paper, we want to propose a technique to resolving this problem by traversing the whole network and finding the distance of individual node from the source node. Then we get its spanning tree and also get the shortest path by the spanning tree from source node to destination node.

Keywords-Routing, Traversing, Shortest Path algorithm, Spanning tree.

2. Introduction

A network is a set of devices often referred to as nodes connected by media links. A node can be a computer, printer or any other device capable of sending or receiving data generated by other nodes on the network. Routing is the act of moving information across an inter-network from a source to a destination. It's also referred to as the process of choosing a path over which to send the packets.

In moving this information from source to destination we never checks the length of the route which is very important because if the length is long then it takes time and more affords to reach the destination[1].

To overcome this problem we find the total distance covered by every node from source node. Then we connect these nodes in their increasing distance from source and gets it minimum spanning tree more than 60% cases. And apply algorithm and gets the shortest path from source to destination.

There are two shortest path techniques had been introduced are

1) Dijkstra's Shortest Path First (SPF) Algorithm.
2) Bellman-Ford Shortest Path Algorithm.

We use Dijkstra's shortest path algorithm to find the shortest path for every node from source to it.

3. Related work

Many shortest path techniques are used to find the shortest path from source node to destination node. This short path saves time and affords and also the secure delivery of information from source to destination node.

There are two shortest path techniques had been introduced are

1) Dijkstra's Shortest Path First (SPF) Algorithm.
2) Bellman-Ford Shortest Path Algorithm.

We get our shortest path by applying Dijkstra's shortest path algorithm.

We have different techniques to find the spanning tree of a network or graphs are

1) Kruskal's algorithm.
2) Jarnik prim algorithm.

But we don't use any of these two spanning tree algorithm to find the spanning tree of the network or graph. We use Dijkstra's Shortest Path First (SPF) Algorithm to find the shortest path from source node to every node. We get spanning tree of any network by connecting the vertices and the edge connects all the vertices.

a) Connected: Every node is linked from one node to another node.

b) Undirected: Edges do not have an associated direction.

After connecting all vertices to each other we get our required spanning tree of the network or graph.

4. Proposed work

In finding shortest path from source to destination in wired network we divide our whole work in three phases are:-

1. Finds shortest path for individual node by applying Dijkstra's shortest path algorithm.
2. Connects all the nodes in increasing order of their total distance and get spanning tree.
3. Getting shortest path from source node to destination from spanning tree of network.

We apply Dijkstra's shortest path algorithm firstly and gets distance of every node from source node and assign that distance value to that node. After getting distance to every node we connects all these node to each other in ascending order and this connection makes its minimum spanning tree around 60% times. We have to perform all three steps on this network which we mentioned on top.

Now we have to apply algorithm to find the shortest path from source to destination.

Step1: apply Dijkstra's shortest path algorithm [4] on any given network.

1) Initialize single source (G, S)

For each vertex (user) V€V (G)

Do d[u] ←∞

π[u]←nil

d[s]←0

2) A← φ (A← array of vertex)

Q ≠ φ (Q← queue of edges)

Do u← Extract min (Q)

S←s U (u) (U← union)

For each vertex (user)

V € adjacent (u)

Relax (u, v, w)

If

D[v]>d(u)+w(u,v)

then

D[v]← d(u)+w(u,v)

Set π[v]←u

3) Repeat step 2 till the last node(user)

Now apply Dijkstra's algorithm on given network is

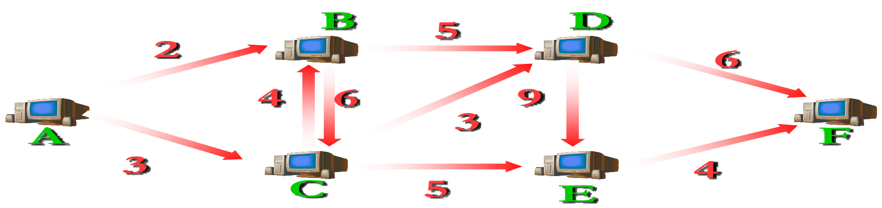

Figure1: network with multiple users.

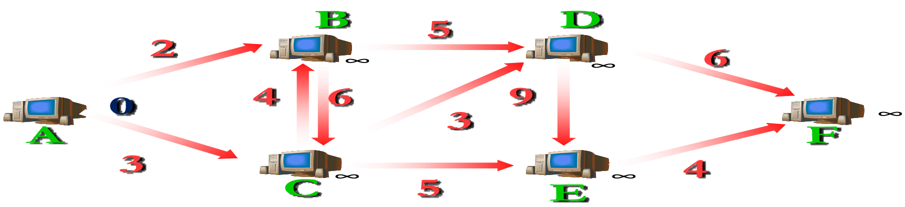

Figure2: network after step1 of Dijkstra's algorithm.

The following table shows the labels of the every vertex.

A=u	B	C	D	E	F
0	∞	∞	∞	∞	∞
0	**2**	3	∞	∞	∞
0	**2**	**3**	7	∞	∞
0	**2**	**3**	**6**	8	∞
0	**2**	**3**	**6**	**8**	12
0	**2**	**3**	**6**	**8**	**12**

After calculation we get the desired weight for each and every node or user.

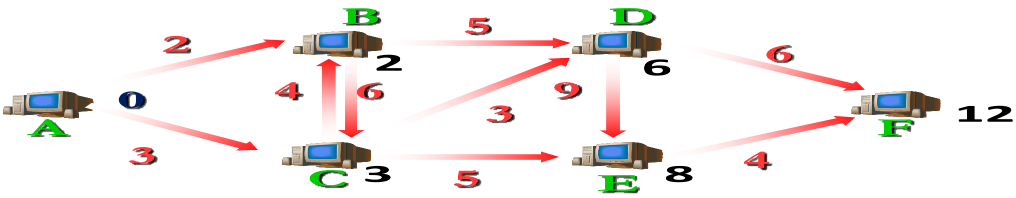

Figure3: network after step2 of Dijkstra's algorithm.

Step 2: steps for joining vertex in increasing order of their total distance from source node.

1) For each vertex V€V(G)
2) Create set (V)
3) Sort the vertex in increasing order of their distance from source node in set (V).
4) Join the vertex in increasing distance.
5) Select the shortest edge/path to join the vertices.
6) No loop/cycle was formed during connection.
7) Repeat step 4and 5 till the last element in set (V).

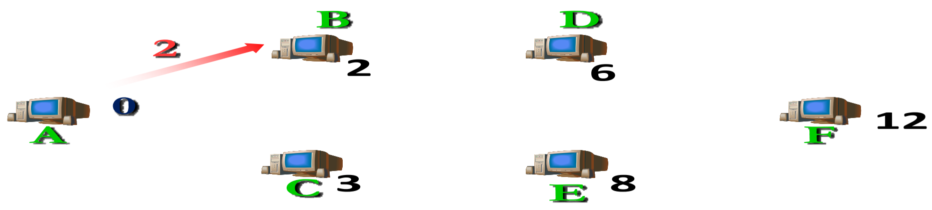

Figure4: shows the path from source to next element of set (V).

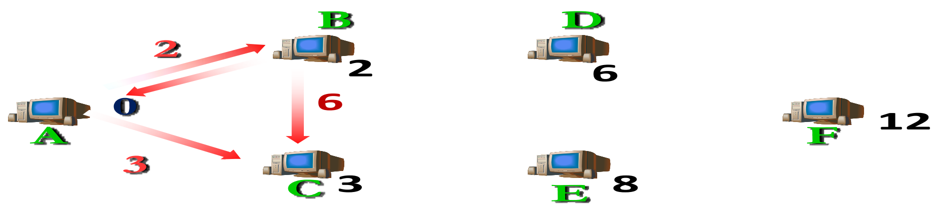

Figure5: shows the path from second element to next element of set (V).

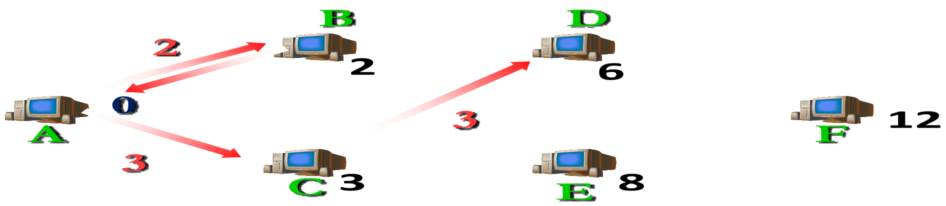

Figure6: shows the path from third element to next element of set (V).

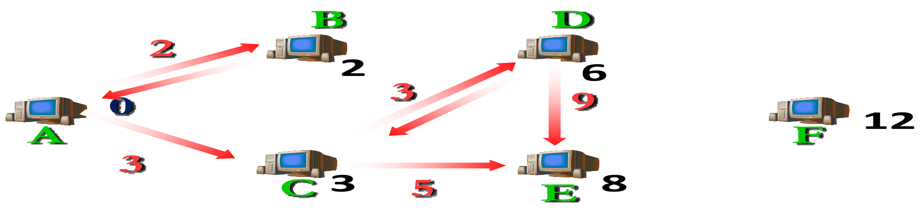

Figure7: shows the path from fourth element to next element of set (V)

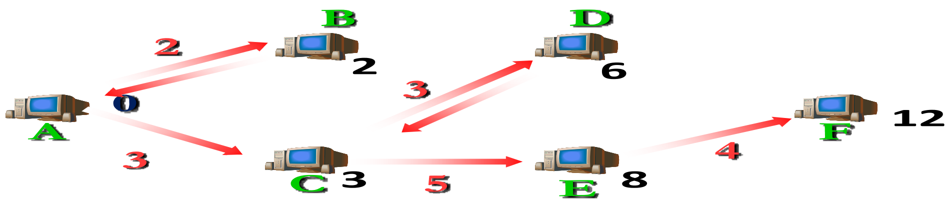

Figure8; shows the path from fifth element to the next and the last element of the set (V) and also its spanning tree.

The image shown in figure8 is our spanning tree and also node connected in increasing order to each other. It also shows the shortest path for each and every node from source.

Now move to third and last step of finding of shortest path from source to destination by traversing each and every node or user in increasing order or weight.

Step3: finding of shortest path from source to destination.

1) Initialize single source (G, S).
2) Move to the next adjacent node or user which is less in weight then all other adjacent nodes.
3) If

   There is short route from current node to next higher node in weight then move to it

   Vi←Vi+1

   Else

   Back to previous node

   Vi← Vi-1
4) Repeat step 2 and 3 till the last node.

Path formed by whole traversing and connecting of nodes from each and every node is the required shortest path of network.

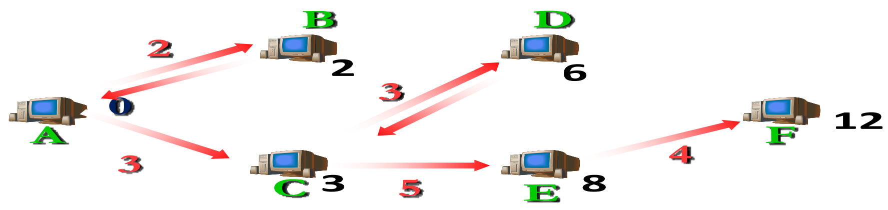

Figure9: this is the shortest path from source to destination after traversing each and every node.

Suppose if a edge 5 is been replaced by 7 then this shortest path changed because we have to select the shortest path from source to destination.

W [e,5]←7

Then we get our new shortest path which is shown in figure10

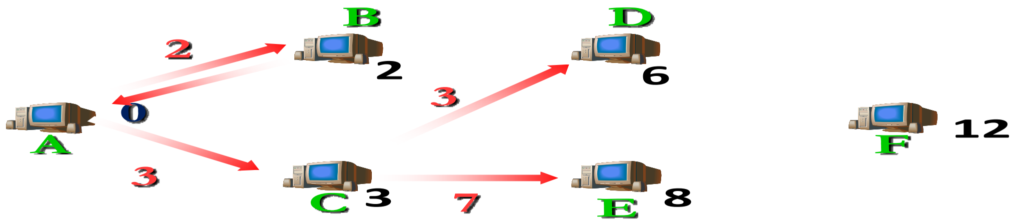

Figure10: weight of an edge 5 is been replaced by 7

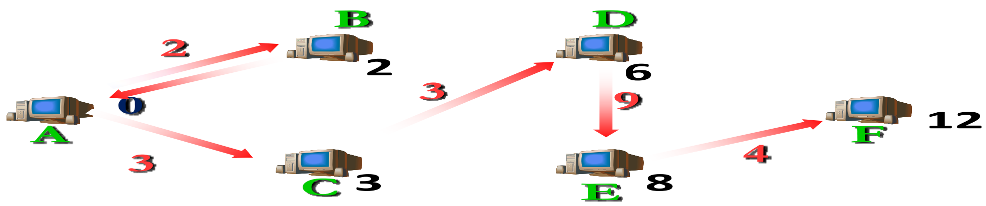

Figure11:shows the shortest path after changing the weight of the edge and gets the shortest path.

5. Conclusion

In this paper we are working to resolve the problem finding of minimum spanning tree without use of any algorithm like kruskal's and the jarnik prim algorithm. We are also tried to find out the shortest path from source to destination by traversing each and every node or user.

6. Future work

In future we try to improve its efficiency of finding minimum spanning tree which is around 60% at present and also try to find some short way of finding the shortest path from source to destination by traversing each and every node .**B**ecause the problem raised at present is to traverse each and every node in minimum time with maximum efficiency.

And also tries it on wireless (Ad Hoc) network to find its shortest path from source to destination without knowing the actual distance between them.

7. References

[1] A. S. Tenenbaum. Computer Networks, 4th edition.PRIENTICE HALL 2002.
[2] http:\\google.co.in
[3] http:\\wikipedia.com
[4] Thomas H. Coreman,Introduction to Algorithm, IT Press 2001