Finding Of Spanning Tree Of A Graph By The Shortest Path Algorithm

Avnish Rana[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

1. Abstract

Now a days communication and transportation has an important place in every person's life in their all business and non-business tasks and routing is the back bone of transportation. In daily life everybody is facing a problem of choosing a shortest path from one location to another location. Shortest path refers to the minimum distance between source and the destination. Choosing a shortest path, one can save both time and money, which are important part of our life. In this paper, we want to propose a technique for resolving this problem by traversing the network and finding the spanning tree. Then we find the spanning tree by using the shortest path algorithm.

2. Introduction

A network is a set of devices often referred to as nodes connected by various medium of 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 over inter-network from a sender node to receiver node. It can also be said to as the process of choosing a path over which the packets of information is to be send. In moving this information from source to destination we always checks the weight of the route which is very important because if the length is long then it takes time and more efforts to reach the destination[1]mainly in the disconnected graphs. To overcome this problem we find the disconnected graphs and then try to connect them with each other so as to form a single spanning tree from the graph There are two shortest path techniques had been introduced are

1) Dijkstra's Shortest Path First (SPF) Algorithm.[1]

2) Bellman-Ford Shortest Path Algorithm.

But we don't use Dijkstra's shortest path algorithm to find the shortest path for every node from source to it.

3. Related work

Many min spanning tree algorithms are used to find the spanning tree from the given directed or undirected graph . Thus found the spanning tree decides the path to access the nodes within the graph. saves time and efforts and also the secure delivery of information from source to destination node as per the requirements.

There are three shortest path techniques had been introduced are

1) Dijkstra's Shortest Path First (SPF) Algorithm.
2) Bellman-Ford 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[2]. We use algorithm to find the spanning tree of the disconnected (as sub-graphs of a graph) graphs , as they are needed to be connected for establishing the communication in between them as a single component. This algorithm is mainly used for finding the spanning forest of the undirected disconnected graphs.

a) Disconnected: Two or more sub graphs disconnected to each other with in a bigger graph.

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:

Our aim in the paper is to find the spanning tree of the weighted undirected graph which has the larger number of nodes. If Kruskal's algorithm or any other algorithm[3] is taken into consideration, It will take more time to find the spanning tree from the given graph .So we have tried to Apply the algorithm [4]which includes the set of steps to reach our goal , Which includes:-

(i) Selecting the minimum weighted edges from each node in the graph.
(ii) Connecting the sub trees formed after step 1.
(iii) choosing the edges of minimum weights. Connecting them will provide the desired spanning tree.

Algorithm:-

:Take a connected Undirected graph G(V,E).

W=weight of an edge ; Wm=edge of minimum weight ; G=Graph ; T=Spanning tree.

Step:1- Initialize the graph G.

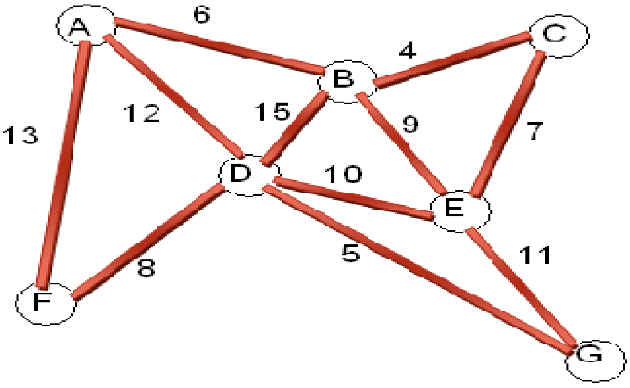

Fig:1 –Shows the initial network of nodes.

Step:2- for first node-

```
        If  W=Wm
            Then "select the edge(Wm)"
        else back to the node.
```

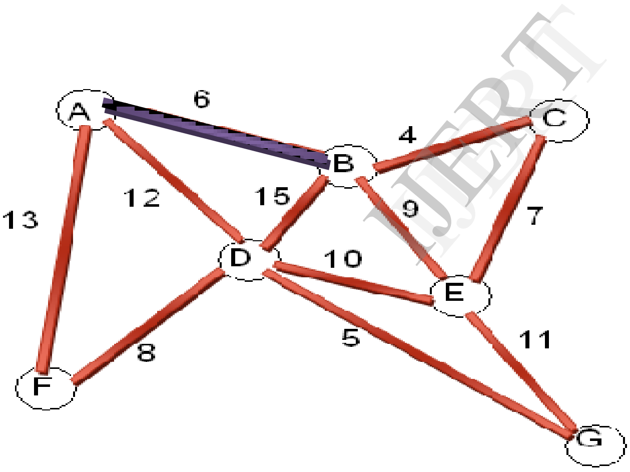

Fig:2-minimum weighted edge is selected for the first node (min {13,12,6}).

Step:3- Repeat step 2 for each node in G.

(i) for node B.

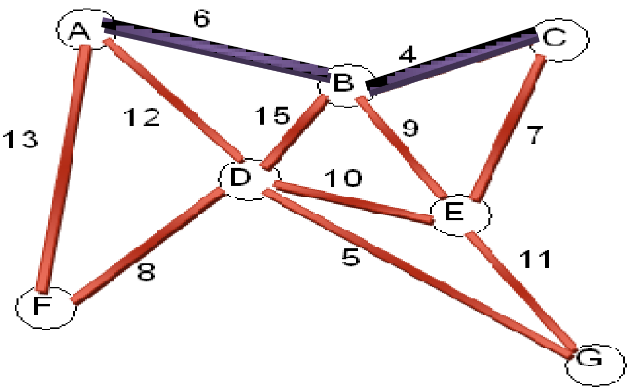

Fig:3-step 2 for nodes B &C.(shows common edge)

(ii) for node E.

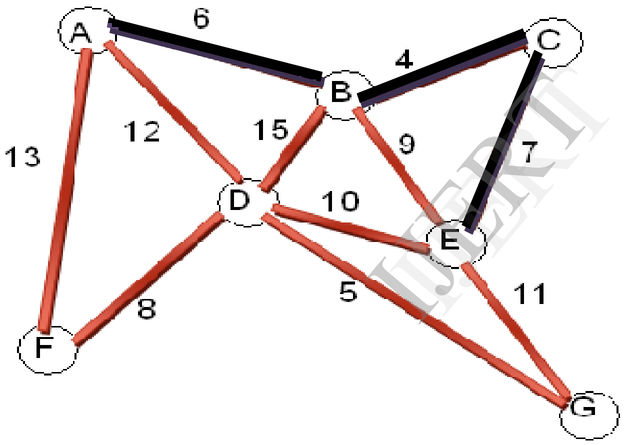

Fig:4-Slection of minimum weighted edge for node E(min{7,9,10,11}).

(here question arises for the node C , the minimum weighted edge from node C is of weight 4, which is already selected for the node B .Hence the edge from B to C is acting commonly for both ie. from both . from Both C as well as from C to B)

[Same case will be followed for node G and D]

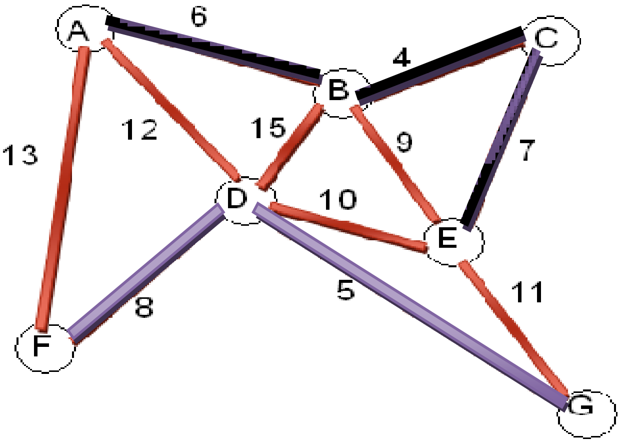

Fig:5-figure showing the completion of step 3.

Here we found two sub graphs ({A,B,C,E} & {F,D,G}) of the graph G.

Now the main aim is to connect these two sub graphs to form the desired spanning tree, with the *shortest path.*

Step:4- connect the sub trees with the minimum weighted edge.

Step:5- No loop/ cycle should be formed.

(find min {13,12,15,10,9,11})

Result is 9.but selection of edge(W=9) makes the loop {BCE}.

Step:6- if loop is founded then take the next higher edge.

So edge(W=10)is to be selected for obtaining the required Spanning tree.

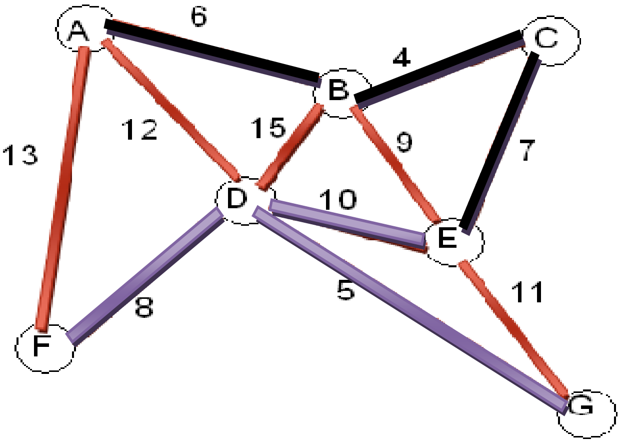

Fig:6-shows the process of connecting two sub trees to form a single one.

Step:7- Repeat step 6 till the spanning tree is formed.

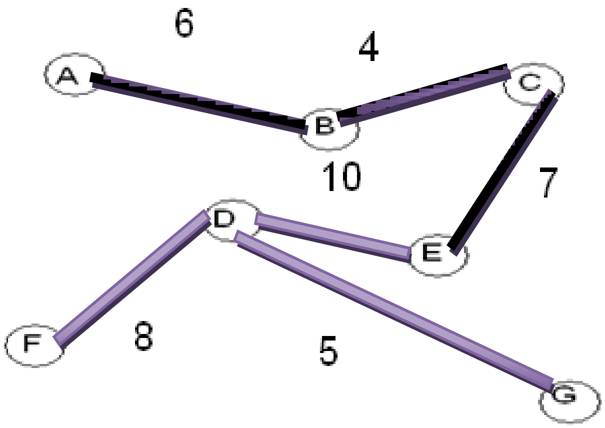

Fig:7- This the required Spanning tree(T).

5. Future Work:

In future we try to improve its efficiency of finding minimum spanning tree and will also try to find some short way of finding the spanning tree .Because the problem raised at present is to traverse each and every node in minimum time with maximum efficiency. And will also try it on wireless (Ad Hoc) network to find its shortest path from source to destination without knowing the actual distance between them.

6. 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.This algorithm is designed to find the spanning tree of the network of larger number of nodes. We are also tried to find out the shortest path from one node to the another by using the minimum weighted edges to make the minimum spanning tree.

7. References:

[1] Thomas H. Coreman,Introduction to Algorithm, IT Press 2001

[2]A. S. Tenenbaum. Computer Networks, The 4 edition .PRIENTICE HALL 2002.

[3]Thomas H.Cormen Charles E.Leiserson,Ronald L.Rivest,And Clifford Stein.Introduction to algorithms.2nd edition,The MIT press2001

[4]David Joyner , Minh Van Nguyen ,And Nathann Cohen, Algorithmic Graph Theory,2010

[5]http:\\google.co.in

[6]http:\\wikipedia.com