ISSN: 0975-766X
CODEN: IJPTFI
Available Online through ***Research Article***
www.ijptonline.com

DEVELOPMENT OF ALTERNATE FAULT TOLERANT MINIMUM WEIGHTED SPANNING TREE USING VERTEX, EDGE CONNECTIVITY

M. Yamuna*[1] and A.Elakkiya[2]
[1,2] School of Advanced Sciences, VIT University, Vellore, India.
Email: myamuna@vit.ac.in

Received on 22-04-2016 *Accepted on 20-05-2016*

Abstract

In networking, broadcasting is also done from a single source to all other sources. The minimum cost of the broadcasting is usually derived using a minimum weighted spanning tree. If a fault occurs at nodes or edges the message will not reach the destination. Fault occurrence cannot be predicted and may occur at any point in near future. In this paper we use vertex connectivity and edge connectivity of a graph for deciding a suitable minimum weighted spanning tree in case of fault occurrence.

Key Words: Cut set, Edge connectivity, Spanning tree, Vertex connectivity.

Introduction

Minimum weighted spanning trees have its widespread applications in networks, the standard applications in telephone networks, railway lines, road lines. In [1] A.Elakkiya has introduced few new techniques which can reduce the execution time in choosing a minimum cost spanning tree when a fault occurs. In [2] Chin Kim Gan**,** performed the PSO algorithm to find the optimal placement of substation and MST algorithm for optimal feeder routing in LV and MV networks. In [3] R. Jayalakshmi et.al have showed that AST is more spatially convergent and it uses shortest path cost for aggregation leads to increase node lifetime and saves energy. In [4] Fakhrosadat Fanian proposed an algorithm in that routes are selected such that, the selected routes length is almost equal and finally results in work load balance among cluster heads and decrease energy consumptions and increase network lifetime. In addition, they also showed that time delay for data transfer to base station becomes leved off in all routes. The main problem in data transmission is to track the fault occurrence and find an alternate path in such cases. In this paper we propose a method of finding an alternate minimum weighted path in case of fault occurrence using standard Prim's algorithm.

Preliminary Note: In this section we provide basic results used in the proposed method.

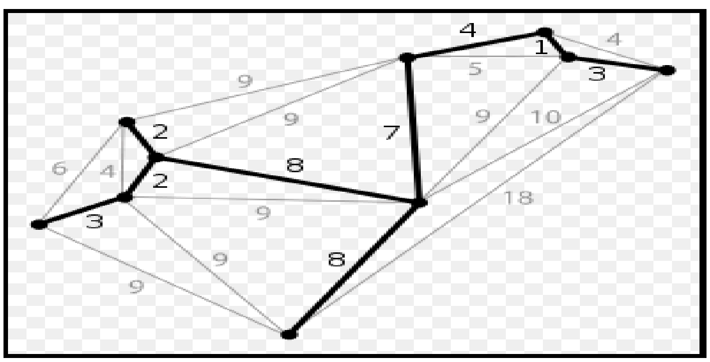
Graph

In the most common sense of the term, a graph is an ordered pair G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge) [5] . For general results in graph theory we refer to [6]

Minimum Weighted Spanning tree

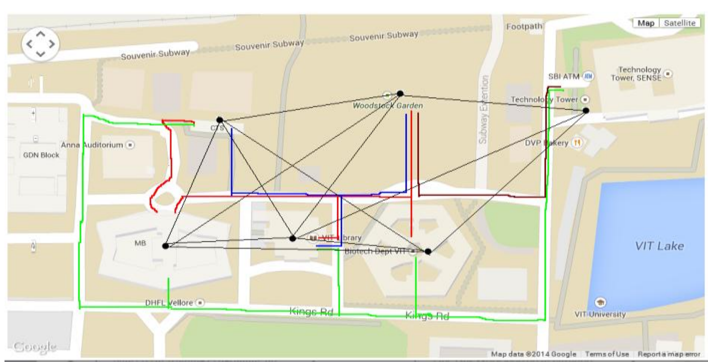
Given a connected, undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. We can also assign a weight to each edge, which is a number representing how unfavorable it is, and use this to assign a weight to a spanning tree by computing the sum of the weights of the edges in that spanning tree. A minimum spanning tree or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree [7].

Snapshot – 1 provides an example of a minimum weighted spanning tree.



Snapshot – 1

The map in snapshot – 2 provides the ariel view of VIT University.



Snapshot – 2

We randomly choose five locations, Main Building (MB), Anna Auditorium, CTS, VIT Library, Biotechnology Department, Technology Tower. The path from different buildings to the other is painted in different colors. For example the red color line presents that there is a path from MB to Anna auditorium, CTS Library, Woodstock and Bio Tech department. This can be converted into a network with vertices representing the locations and edges representing passage between them as seen in the map (black color graph).

If mobility between two buildings is not available (may be due to work done in the roads connecting them), then alternate paths can be picked as seen in the map. This can also be done from the graph, by using graph properties like edge, vertex connectivity. This example is a simple illustration of how a network is made into a graph. In many networks data transfer is done from a single source to all other nodes. A minimum weighted spanning tree is in general used for this purpose. In most of the networks cost is not an issue but fault occurrence that disconnects the network which leads to the message never reaching some part of the network is the main issue. To overcome this when a fault occurs, a new spanning tree is generated and used for broadcasting. In most case the fault occurrence nodes cannot be predicted. Generating a minimum weighted spanning tree after the fault has occurred delays the broadcasted message to certain part of the network atleast.

To overcome this, if a predefined minimum weighted spanning tree is available at the source for all possible fault occurrence then the delay in message broadcasting can be avoided. In this paper, we propose a method to avoid this delay using properties of vertex and edge connectivity.

Proposed Method

Vertex connectivity of any graph is removal of minimum number of vertices that disconnects G. So if in a network we need minimum weighted spanning tree after fault occurrence then the maximum number of nodes at which fault can occur should be less than the vertex connectivity of the graph. If a network has n vertices with vertex connectivity k, then fault can occur at 1, 2, 3,… k – 1 vertices only, because in the remaining cases the network become disconnected and hence data transfer is not possible. We shall denote the data transfer network as a weighted graph G with n vertices with vertex connectivity k.

Proposed Method for Finding Minimum Weighted Spanning Tree

1. Find the vertex connectivity of G (say k).

2. Determine all possible subgraphs $G_1, G_2, \ldots G_h$ where $|h| = 2^h - 1$ that is, we determine all possible subgraphs generated by removing 1 vertex, 2 vertices, … k – 1 vertices (Note that all the subgraphs $G_1, G_2, \ldots G_h$ are connected).
3. Determine minimum weighted spanning tree for $G_1, G_2, \ldots G_h$.

The minimum weighted spanning tree can be calculated using any of the standard algorithms like Prim, Kruskal. We use Prim's algorithm for this. By this method for fault occurrence at any node or set of nodes an alternate spanning tree is available for immediate data transfer which guarantees that there would be less delay in data transfer to the disconnected nodes because of fault occurrence. We explain the method by an example.

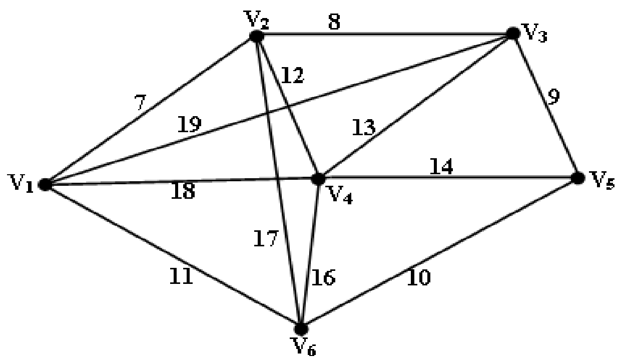
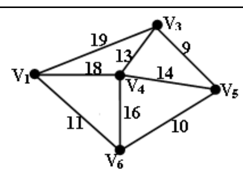
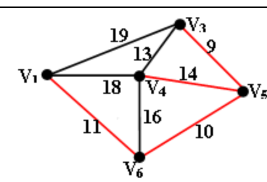
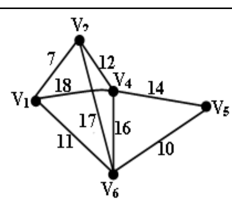
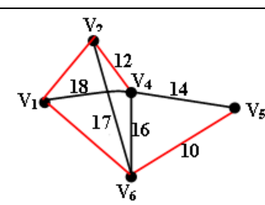
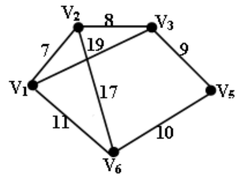
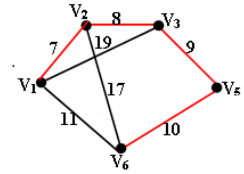
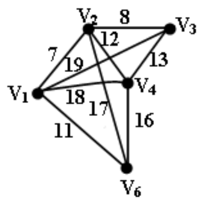
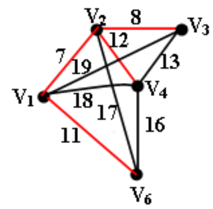
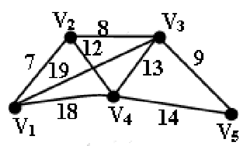
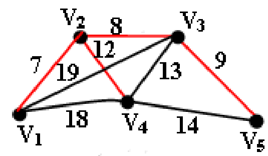
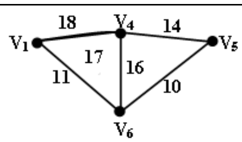
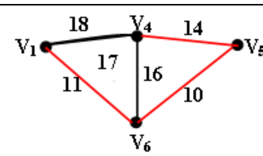
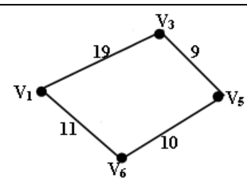
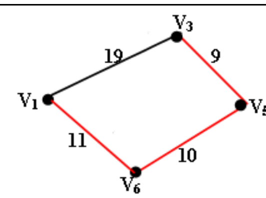
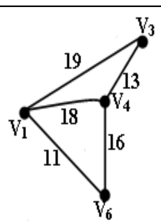
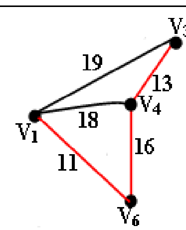
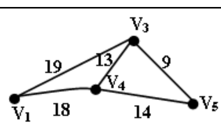
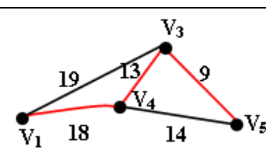


Fig . 1

Let G be a network with 6 vertices and 12 edges has seen in Fig . 1. Vertex connectivity of G is 3 (Removal of 3, 4 ,6 disconnects the graph G). We need to consider all possible subgraphs generated by removal of 1 vertex, 2 vertices. The subgraphs and alternate minimum weighted spanning tree is provided in Table - 1.

S.no	Vertex set removed	Subgraph	Prim's Method	Minimum Weighted Spanning Tree	Cost
1	{ 2 }		$\begin{pmatrix} \_ & 19 & 18 & \infty & 11 \\ 19 & \_ & 13 & 9 & \infty \\ 18 & 13 & \_ & 14 & 16 \\ \infty & 9 & 14 & \_ & 10 \\ 11 & \infty & 16 & 10 & \_ \end{pmatrix}$		43
2	{ 3 }		$\begin{pmatrix} \_ & 7 & 18 & \infty & 11 \\ 7 & \_ & 12 & \infty & 17 \\ 18 & 12 & \_ & 14 & 16 \\ \infty & \infty & 14 & \_ & 10 \\ 11 & 17 & 16 & 10 & \_ \end{pmatrix}$		40

3	{ 4 }		$\begin{pmatrix} \_ & 7 & 19 & \infty & 11 \\ 7 & \_ & 8 & \infty & 17 \\ 19 & 8 & \_ & 9 & \infty \\ \infty & \infty & 9 & \_ & 10 \\ 11 & 17 & \infty & 10 & \_ \end{pmatrix}$		34
4	{ 5 }		$\begin{pmatrix} \_ & 7 & 19 & 18 & 11 \\ 7 & \_ & 8 & 12 & 17 \\ 19 & 8 & \_ & 13 & \infty \\ 18 & 12 & 13 & \_ & 16 \\ 11 & 17 & \infty & 16 & \_ \end{pmatrix}$		38
5	{ 6 }		$\begin{pmatrix} \_ & 7 & 8 & 18 & \infty \\ 7 & \_ & 8 & 12 & \infty \\ 19 & 8 & \_ & 13 & 9 \\ 18 & 12 & 13 & \_ & 14 \\ \infty & \infty & 9 & 14 & \_ \end{pmatrix}$		36
6	{ 2 3 }		$\begin{pmatrix} \_ & 18 & \infty & 11 \\ 18 & \_ & 14 & 16 \\ \infty & 14 & \_ & 10 \\ 11 & 16 & 10 & \_ \end{pmatrix}$		35
7	{ 2 4 }		$\begin{pmatrix} \_ & 19 & \infty & 11 \\ 19 & \_ & 9 & \infty \\ \infty & 9 & \_ & 10 \\ 11 & \infty & 10 & \_ \end{pmatrix}$		30
8	{ 2 5 }		$\begin{pmatrix} \_ & 19 & 18 & 11 \\ 19 & \_ & 13 & \infty \\ 18 & 13 & \_ & 16 \\ 11 & \infty & 16 & \_ \end{pmatrix}$		40
9	{ 2 6 }		$\begin{pmatrix} \_ & 19 & 18 & \infty \\ 19 & \_ & 13 & 9 \\ 18 & 13 & \_ & 14 \\ \infty & 9 & 14 & \_ \end{pmatrix}$		40

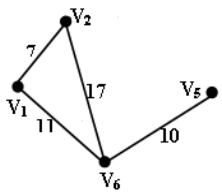
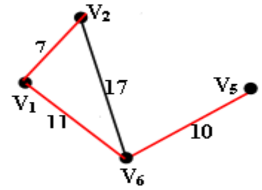
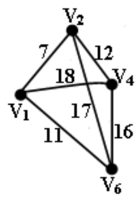
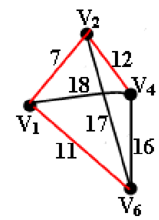
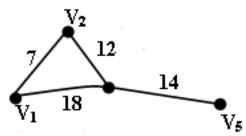
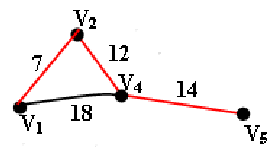
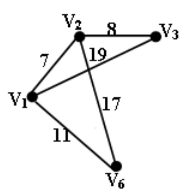
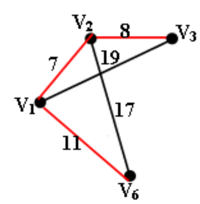
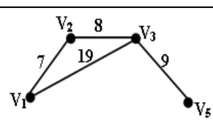
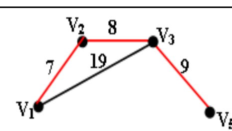
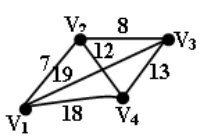
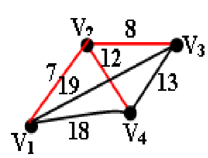
10	{ 3 4 }		$\begin{pmatrix} \_ & 7 & \infty & 11 \\ 7 & \_ & \infty & 17 \\ \infty & \infty & \_ & 10 \\ 11 & 17 & 10 & \_ \end{pmatrix}$		28
11	{ 3 5 }		$\begin{pmatrix} \_ & 7 & 18 & 11 \\ 7 & \_ & 12 & 17 \\ 18 & 12 & \_ & 16 \\ 11 & 17 & 16 & \_ \end{pmatrix}$		30
12	{ 3 6 }		$\begin{pmatrix} \_ & 7 & 18 & \infty \\ 7 & \_ & 12 & \infty \\ 18 & 12 & \_ & 14 \\ \infty & \infty & 14 & \_ \end{pmatrix}$		33
13	{ 4 5 }		$\begin{pmatrix} \_ & 7 & 19 & 11 \\ 7 & \_ & 8 & 17 \\ 19 & 8 & \_ & \infty \\ 11 & 17 & \infty & \_ \end{pmatrix}$		26
14	{ 4 6 }		$\begin{pmatrix} \_ & 7 & 19 & \infty \\ 7 & \_ & 8 & \infty \\ 19 & 8 & \_ & 9 \\ \infty & \infty & 9 & \_ \end{pmatrix}$		24
15	{ 5 6 }		$\begin{pmatrix} \_ & 7 & 19 & 18 \\ 7 & \_ & 8 & 12 \\ 19 & 8 & \_ & 13 \\ 18 & 12 & 13 & \_ \end{pmatrix}$		27

Table - 1

A similar technique can be adopted if fault occurs in the edge of the network, using edge connectivity of the graph. If edge connectivity of G is q then by removing 1 edge, 2 edges, … q - 1 edges we can generate subgraphs. As discussed above we can find minimum weighted spanning tree using Prim's algorithm. Similarly if faults occur in vertices and edges together then we can generate subgraphs and find the minimum weighted spanning tree taking care that the number of vertices removed is less than vertex connectivity and number of edges removed is less than edge connectivity.

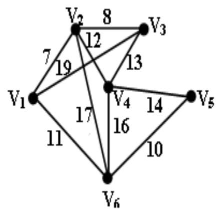
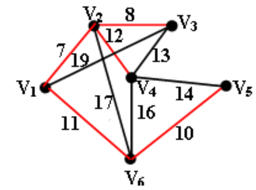
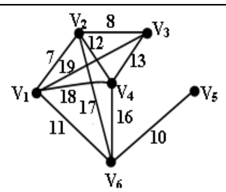
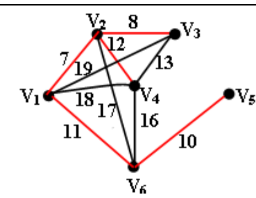
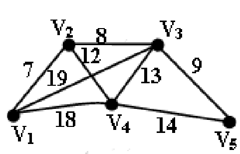
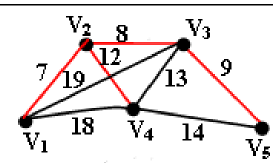
S. No	Edges removed	Subgraph	Prim's Model	Minimum cost spanning tree	Cost
1	{ (3, 5), (1, 4) }		$\begin{pmatrix} \_ & 7 & 19 & \infty & \infty & 11 \\ 7 & \_ & 8 & 12 & \infty & 17 \\ 19 & 8 & \_ & 13 & \infty & \infty \\ \infty & 12 & 13 & \_ & 14 & 16 \\ \infty & \infty & \infty & 14 & \_ & 10 \\ 11 & 17 & \infty & 16 & 10 & \_ \end{pmatrix}$		48
2	{ (3, 5), (4, 5) }		$\begin{pmatrix} \_ & 7 & 19 & 18 & \infty & 11 \\ 7 & \_ & 8 & 12 & \infty & 17 \\ 19 & 8 & \_ & 13 & \infty & \infty \\ 18 & 12 & 13 & \_ & \infty & 16 \\ \infty & \infty & \infty & \infty & \_ & 10 \\ 11 & 17 & \infty & 16 & 10 & \_ \end{pmatrix}$		48
S. No	**Edges and vertices removed**	**Subgraph**	**Prim's Model**	**Minimum cost spanning tree**	**Cost**
1	Vertices { 5, 6 } edges { (v_2, v_6), (v_1, v_6), (v_6, v_4), (v_6, v_5) }		$\begin{pmatrix} \_ & 7 & 8 & 18 & \infty \\ 7 & \_ & 8 & 12 & \infty \\ 19 & 8 & \_ & 13 & 9 \\ 18 & 12 & 13 & \_ & 14 \\ \infty & \infty & 9 & 14 & \_ \end{pmatrix}$		36

Table – 2

For the graph in Fig. 1 edge connectivity equal to 3. Table – 2 provides the shortest path for some random combination for removal of edges, vertices and edges. Now consider the graph in Fig. 2

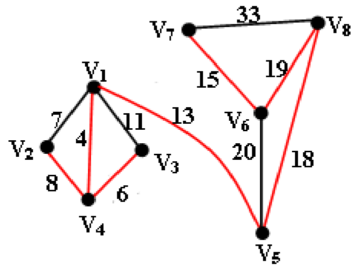


Fig. 2

The minimum weighted spanning tree is given in red color. The weight of the tree is 83. Now we calculate the weight in a modified way. We do it by calculating the weights of subgraphs and then adding them.

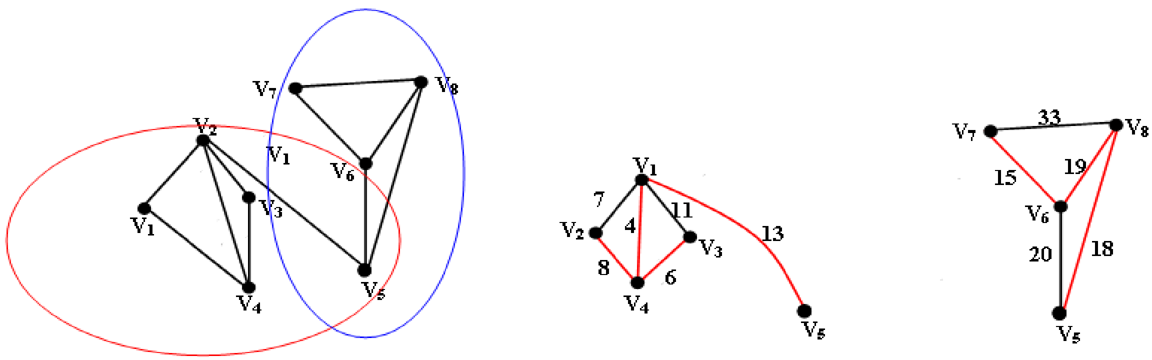
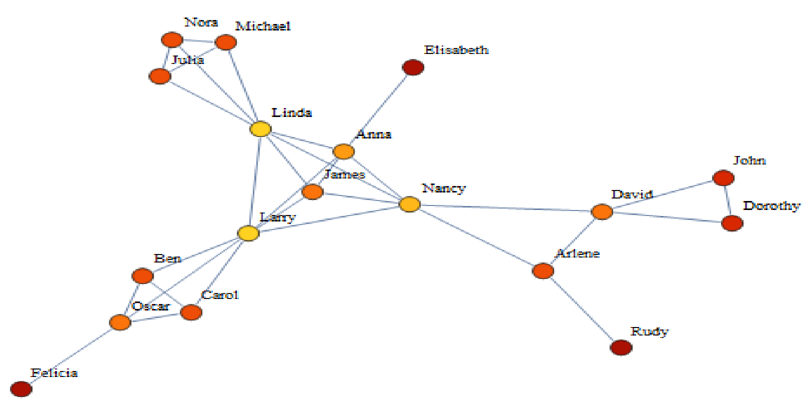


Fig. 3

In Fig.3 the red and blue circles represent the subgraphs chosen. Note that v_5 is included in both the subgraphs. We calculate the minimum tree for the first graph, then for the second graph. The tree can be combined together from the common vertex. There are huge social networks like the one seen in snapshot – 3 [8].



Snapshot – 3

In such cases the proposed calculation becomes lengthy. In such situations we can subdivide the network into different bits as seen in Fig.4.

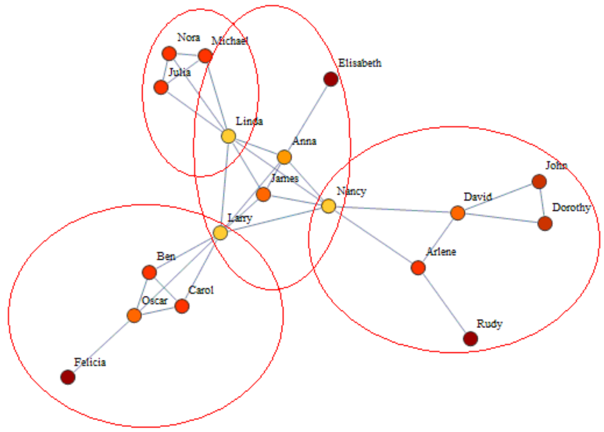


Fig. 4

The network is divided into different bits. For different bits the proposed method can be developed. Once the domain of defect is detected the corresponding path can be picked and linked with the remaining.

Conclusion

The proposed method provides an alternate spanning tree for data transmission. This alternate tree is predefined and immediately available for use which saves the time taken for any transmission. Once created, this alternate way is anytime available which avoids the clumsy situation of deciding alternate tree. This can be implemented for fault occurrence either in nodes or edges or nodes and edges together. This method can be implemented for huge networks also. So, the proposed technique is good in situations which demands time consumption.

Reference

1. A. Elakkiya. Development and Analysis of a Fault Tolerant Algorithm for Spanning Trees of a Computer Network. M. phil thesis, Auxilium College, Vellore, India (2012).
2. Ihsan Jabbar Hasan, Chin Kim Gan, Meysam Shamshiri, Mohd Ruddin Ab Ghani and Rosli Bin Omar, Optimum Feeder Routing and Distribution Substation Placement and Sizing using PSO and MST, 2014, Indian Journal of Science and Technology, Vol 7(10), pp 1682–1689.
3. R. Jayalakshmi, B. Baranidharan and B. Santhi, Attribute based Spanning Tree Construction for Data Aggregation in Heterogeneous Wireless Sensor Networks, 2014, Indian Journal of Science and Technology, Vol 7 (S4), pp 76–79.
4. http://en.wikipedia.org/wiki/Graph_theory.

5. Fakhrosadat Fanian and Marjan Kuchaki Rafsanjani, A Novel Routing Efficient Algorithm Based on Clustering in WSNs, 2013, Indian Journal of Science and Technology, Vol 6 (12), pp 5542–5545.
6. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science. Prentice Hall India (2010).
7. http://en.wikipedia.org/wiki/Minimum_spanning_tree.
8. http://www.wolfram.com/mathematica/new-in-8/graph-and-network-analysis/

Corresponding Author:

M. Yamuna*,

Email: *myamuna@vit.ac.in*