International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

SIMULATION AND IMPLMENTATION OF MULTIAGENT BASED DISTRIBUTION AUTOMATION SOLUTION FOR SELF HEALING GRIDS BY USING PRIM'S ALGORITHM

Author 1
Mohit Haridas Tapase
M.E. Scholar, Department of Electronics & Telecommunication Engineering,
D.Y.Patil College of Engineering & Technology, Shivaji University ,Kolhapur, Maharashtra , India.

Author 2
Dr.Ajitsinh N. Jadhav
Professor, Department of Electronics Engineering,
D.Y.Patil College of Engineering & Technology ,Shivaji University ,Kolhapur, Maharashtra , India.

Author 3
Sanjay B. Patil
Associate Professor, Department of Electronics Engineering,
D.Y.Patil College of Engineering & Technology ,Shivaji University ,Kolhapur, Maharashtra , India.

Abstract
Today's life is totally dependent on electrical supply and in some cases like maintenance at power generation station or failure of node due to natural calamity like flood ,earth quake causes hurdles in maintaining electric supply at each node. The backbone of power supply totally depends upon power grid system. To overcome such drawback self healing techniques used. Self-healing is the important aspect of the nature in which any system recovers itself to the normal level whenever the fault or breakdown occurs system with or even no human intervention . To avoid problems of power failure due to link failures in power grid networks a cost effective greedy prim's algorithm based on minimum spanning tree is introduced .The Fault Location, Isolation and Service Restoration (FLISR) solution is based on Prim's algorithm .A multiagent-based distribution automation solution has been developed to restore the power supply in case of outages in a distribution network. To support prim's algorithm simulation is carried out using MATLAB software and the hardware implementation of above algorithm is backed by using arduino and raspberry pi platform.

The system will enable to find the minimum spanning tree cost of the link to the different nodes in the electric grid network . It will also provide path to connect all the nodes even after failure occurs in between links. The tests have been carried out to show that the proposed algorithm can perform the FLISR process as quickly as within several seconds of computation time and that the system is stable and can handle complex scenarios such as multiple simultaneous faults .

Keywords: Fault Location,Isolation and Service Restoration (FLISR), minimum spanning tree (MST), multiagent systems (MAS), Prim's algorithm.

Introduction
" A self-healing system should recover from the abnormal (or "unhealthy") state and return to the normative ("healthy") state, and function as it was prior to disruption."Self-healing is the important aspect of the nature in which any system recovers itself to the normal level whenever the fault or breakdown occurs. Whenever Electricity is taken into account, then also self-healing of this system should take into consideration. Self-healing is one of the significant features of the smart power system [2]. Figure 1 depicts the formation of the self-healing loop with the data-flow among the three stages and the environmental interfaces.

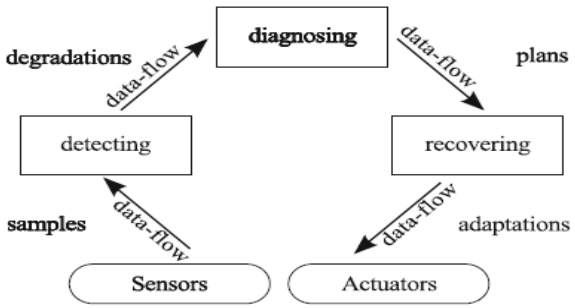

Figure 1: Staged loop of self Healing

Detecting: Filters any suspicious status information received from samples and reports detected degradations to diagnosis.**Diagnosing:**Includes root cause analysis and calculates an appropriate recovery plan with the help of a policy base. **Recovery:** Carefully applies the planned adaptations meeting the constraints of the system capabilities and avoids any unpredictable side effects [3] .

Block Diagram:

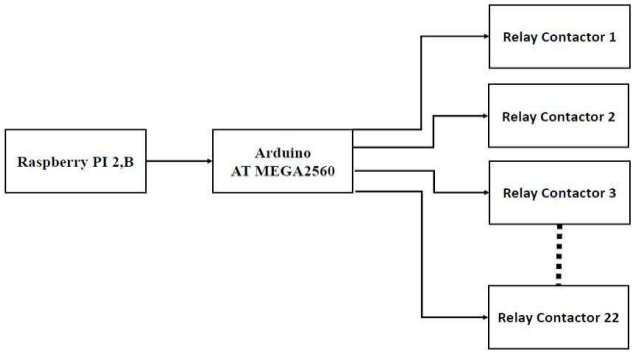

Figure 2: Block diagram of Self Healing System

The design of system is taking into consideration that it has to overcome the problem of power failure due to insufficiency of any system. So we have to design cost effective grid management system in such a manner that it will resolve the problems due to power failure but also find the minimum spanning tree cost of routing the link of the network grid.

Now to implement self healing techniques we used raspberry pi ,arduino as a platform. Raspberry Pi is accessed through the computer through ethernet cable using internet protocol and remote desktop using VNC server.

We have programmed the arduino through arduino IDE for UART reception to relay driver ULN 2003 and arduino AT MEGA 2560 connected to the USB 2.0 port of raspberry pi.The Connection of LED and power supply for the gridding system using relay which is controlled through the arduino AT MEGA 2560.

If any link is failed in this design system then the raspberry pi gets aware of that and it gives the newely calculated prim's algorithm variable and send it to arduino at 9600 baudrate,which arduino receives and turn on specific pin regarding the specific relay to be turn on and pull up that respective pin connection through that relay is connected .

The steps of working hardware connection as follows ;

- Create Initial variable for the algorithm's resources in python language.
- Take input from the user for interaction with code.
- Take the fault condition from the switches using the GPIO package for python language .
- Update variable into python coding .
- Find out minimum spanning tree cost .
- Send code words to Arduino mega2560 using COM port at 9600 baud rate .
- Drive the relay according to the code words received at Arduino serial port .

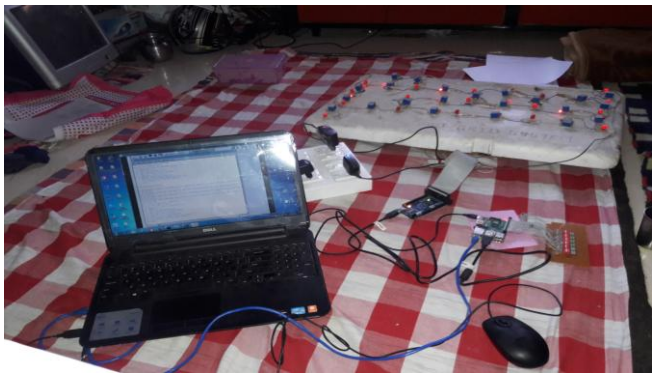

Figure 3: Working Snapshot of Hardware Setup

Prim's Algorithm :

Let us assume that graph G = (V, E), with vertex set V and edge set E is connected and undirected. Without loss of generality, it can be assumed that each weight is distinct, thus G is guaranteed to have only one MST. This assumption simplifies implementation, otherwise a numbering scheme can be applied to edges with same weight, at the cost of additional implementation complexity. Let n be the number of vertices, m the number of edges (|V| = n, |E| = m), and p the number of processes involved in computation of MST. Let w(v, u) denote weight of edge connecting vertices v and u. Input graph G is represented as n X n adjacency matrix A = ($a_{i,j}$) defined as follows ;

$$a_{ij} = \begin{cases} w(V_i, V_j) & if\ (V_i, V_j) \in E \\ 0 & otherwise \end{cases} \quad (1)$$

Prim's algorithm starts from an arbitrary vertex and then grows the minimum spanning tree (MST) by choosing a new vertex and adding it to MST in each iteration. Vertex with an edge with lightest weight incident on the vertices already in MST is added in every iteration. The algorithm continues until all the vertices have been added to the MST. This algorithm requires O (n^2) time. Implementations of Prim's algorithm commonly use auxiliary array d of length n to store distances (weight) from each vertex to MST. In every iteration a lightest weight edge in d is added to MST and d is updated to reflect changes.

Parallelizing the main loop of Prim's algorithm is difficult, since after adding a vertex to MST lightest edges incident on MST change. Only two steps can be parallelized: selection of the minimum-weight edge connecting a vertex not in MST to a vertex in MST, and updating array d after a vertex is added to MST [6] .

Processing Steps for Prim's Algorithm :

1) Partition the input set V into p subsets, such that each subset contains n/p consecutive vertices and their edges, and assign each process a different subset. Each process also contains part of array d for vertices in its partition. Let V_i be the subset assigned to process p_i, and d_i part of array d which p_i maintains. Partitioning of adjacency matrix is illustrated in Figure 4.

Figure 4: Partitioning of adjacency matrix among p processes

2) Every process p_i, finds minimum weight edge e_i (candidate) connecting MST with a vertex in V_i.
3) Every process p_i sends its e_i edge to the root process using all-to-one reduction.
4) From the received edges, the root process selects one with a minimum weight (called global minimum-weight edge e_{min}), adds it to MST and broadcasts it to all other processes.
5) Processes mark vertices connected by e_{min} as belonging to MST and update their part of array d.
6) Repeat steps 2–5 until every vertex is in MST.

Finding a minimum-weight edge and updating of d_i during each iteration costs O(n/p).Each step also adds a communication cost of all to one reduction and all to one broadcast. These operations complete in O (log p). Combined cost of one iteration is O(n/p + log p). Since there are n iterations, total parallel time this algorithm runs in is:

$$T_P = O(n^2/p) + O(n \log p) \quad (2)$$

Minimum Spanning Tree

A minimum spanning tree of an undirected weighted graph G is a spanning tree of which the sum of the edges .To calculate minimum spanning tree for designed system the weights are taken from following table ;

Weight	Shape	Meaning
1		Internally Connected (One nation one grid concept).
2	———	Normal Condition electrical supply is flow through the link the user did not pay extra money.
3	- - - - -	Whenever fault occurs in the link and supply goes off but user still wants electricity then user has to pay extra money in that situation to electricity distribution center .

Table 1: Weights meaning to calculate MST cost

Results

Hardware section result's

For programming Python 2.7.9 shell is used for hardware section.

Show all links of the Grids

It shows the all present link in the grid that can use as the path for electricity distribution. Example shown below is of link in grid system ('A','N01', '2','c') Here 'A 'and 'N01' show end node that connected by above link & 2 denote cost for using this link in grid system. And c is the system denoted name for the respective link.

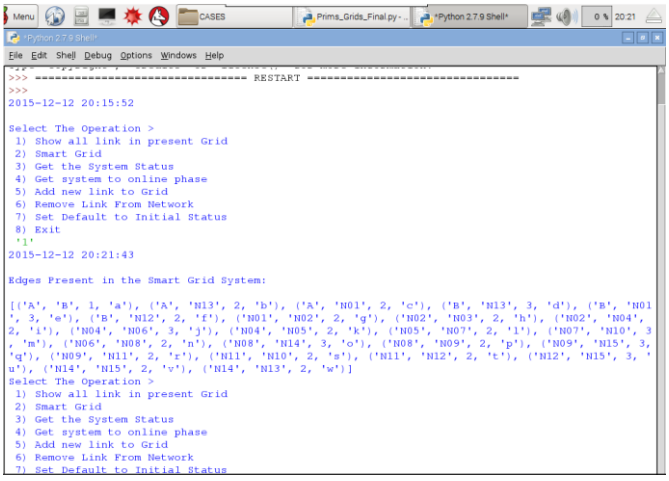

Figure 5: Show all links of the Grids

Smart grid

In this section we formulate the prim's algorithm in the python code using python 2.7.9.Using prim's code we can find out the minimum spanning tree cost for given system having two power supply and 15 user node that can connected through link provided on the basis of the link cost between the system.

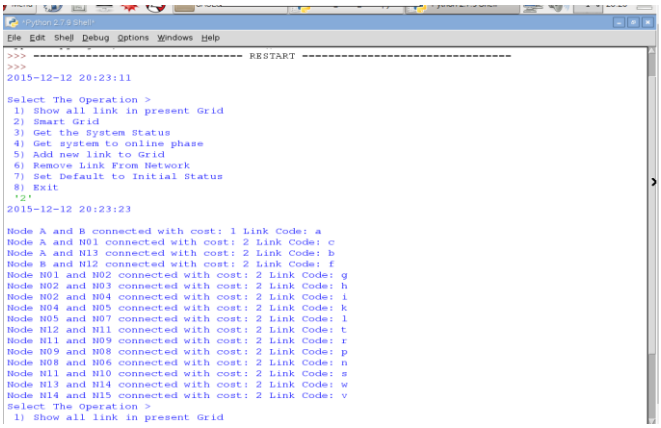

Figure 6: Smart Grid

Get the System Status

In this system case we find out which link is down or say broken and which link are still in work or active. This case used for the finding out the prim's algorithm with latest updated value.

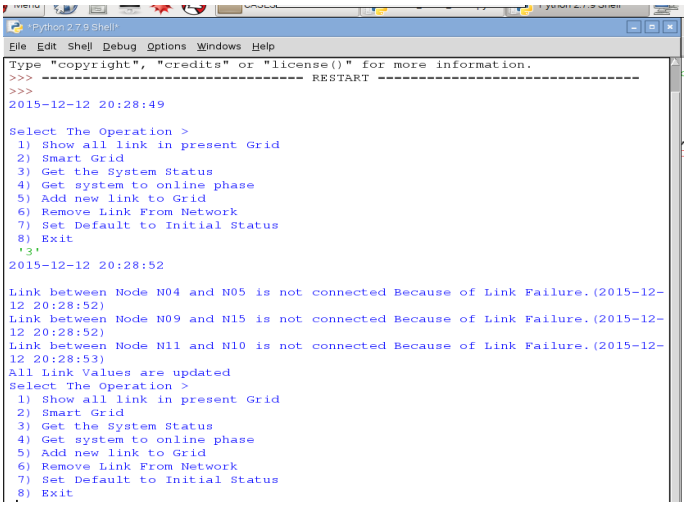

Figure 7: Get the system status

Get the System into online phase

In this case whole code goes into online state in which this code continuously monitor the link status for broken or active and then calculate the prim's algorithm and change the relay status accordingly so that our project output could be real-time.

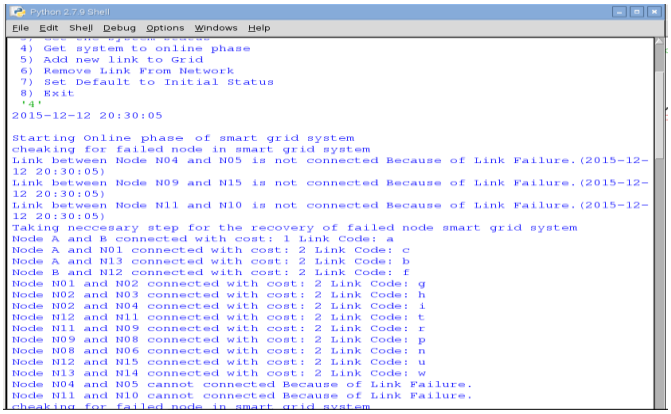

Figure 8: Get the System into online phase

Add new link to the Grid

If the there is chance of adding new link into grid because of the some natural or manmade issues that could result into adding new link into system that can couse the increase the redundancy in the system We can add the link in the system easily and add the parameter into the system to calculate and reformulate the new spanning tree for the system.

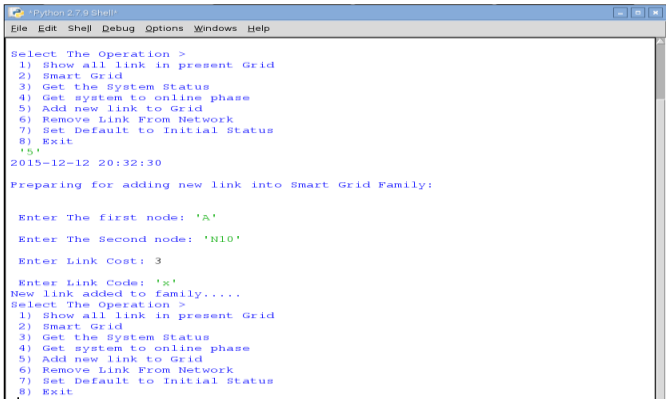

Figure 9: Add new link to the Grid

Remove link from network

By the chance there is also probability of the system to remove the existing link from the grid system because of frequent failure in that link that could result into decrease the efficiency of system. So that we can remove existing system link and recalculate the prim's cost.

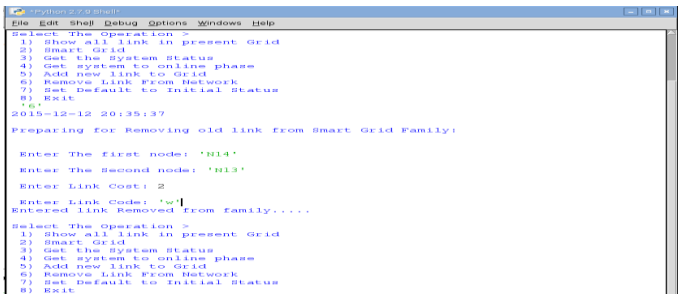

Figure 10: Remove link from network

Calculate minimum spanning tree cost

The calculation of minimum spanning tree cost is done by using prim's algorithm. The steps are as follows;

- Create Initial variable for the algorithm's resources in python language .
- Take input from the user for interaction with code .
- Take the fault condition from the switches using the GPIO package for python language .
- Update variable into python coding .
- Find out minimum spanning tree cost.

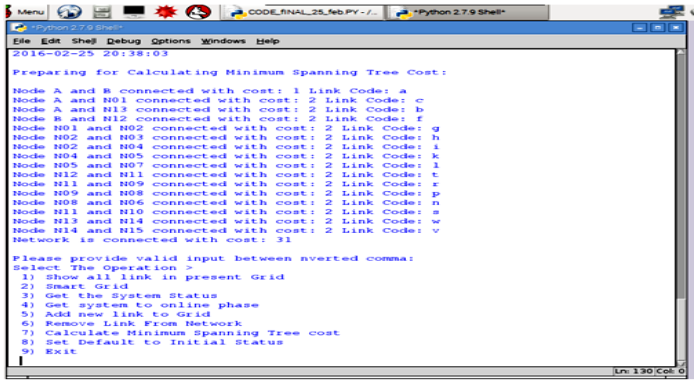

Figure 11: Calculate minimum spanning tree cost

Software section result using MATLAB

To check the minimum spanning tree cost and single link, multiple link faults the graphical user interface is made to check the system reliability.

Calculation of minimum spanning tree cost

For Normal conditions (means without faults in the links) the minimum spanning tree cost is 31 and 16 links has been used to heal all the nodes present in the system.

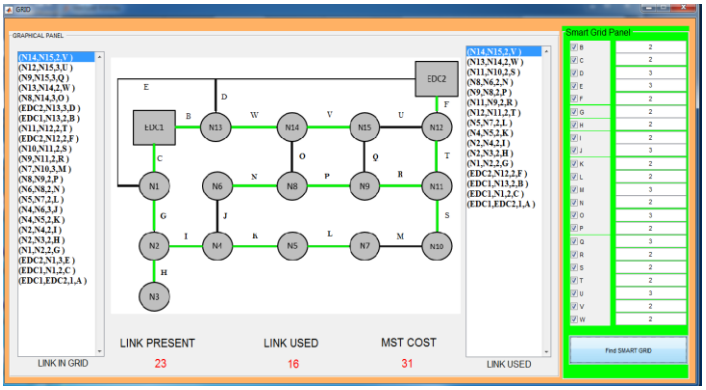

Figure 12: Calculation of minimum spanning tree cost

The steps for calculation MST is as follows for MATLAB section;

- Create Initial variable for the algorithm's resources .
- Select start node for the initial and add it to minimum spanning tree variable.
- Find vertices that are connected to the node added and select minimum weighted link and added to minimum spanning tree.
- Add other links to Dictionary.
- Select the next node by the minimum link cost and see if it is already added to the list and add it to minimum spanning tree .
- Loop reach for all the nodes that are connected in system .

Single link fault

For the design system there are total 23 links present in the system .Out of which "A" link is internally connected (One nation one grid) so we are able to take the 22 single link faults .From 22 faults system are able to heal all the nodes for 21 faults only but for "H" link fault the system wont able to recover the node N3.The below figures shows details for all 22 faults , link present ,link used and MST cost to heal all the nodes.

Single link fault "B" :

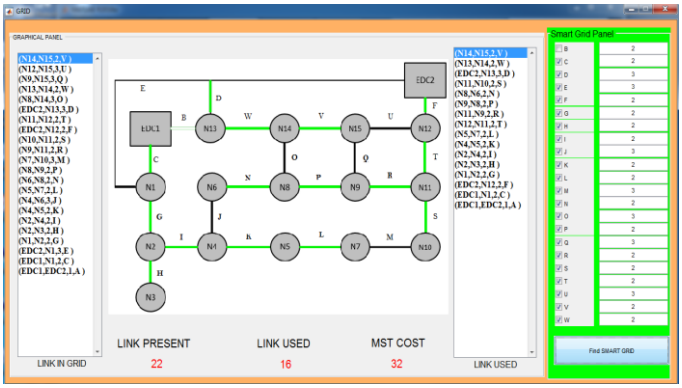

Figure 13: Single link fault "B"

Single link fault "H" :

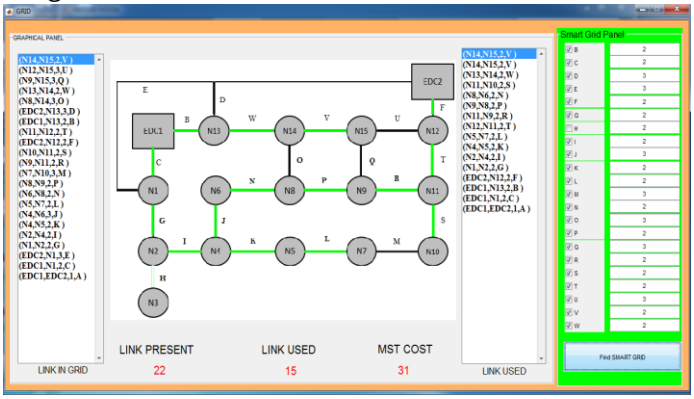

Figure 14: Single link fault "H"

Multiple link fault

Consider "T,F,N,V"link failed then the details are as follows;
T=N11,N12,2,T ;
F=EDC 2,N12,2,F ;
N=N6,N8,2,N ;
V=N14,N15,2,V.

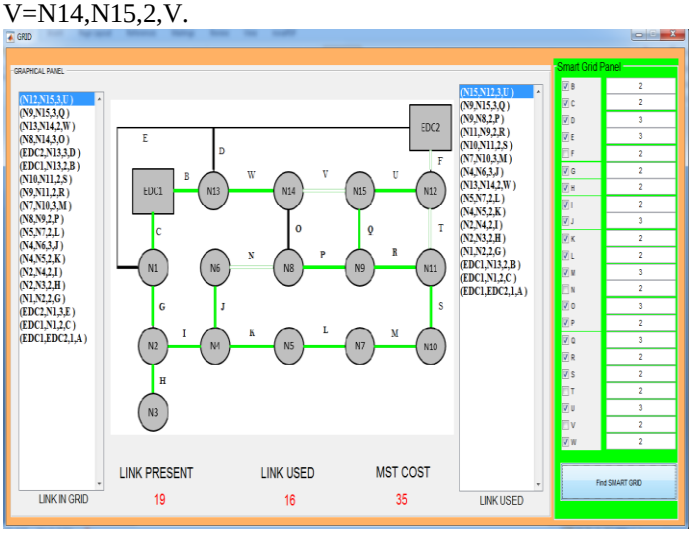

Figure 15: Multiple line fault

Then the system get above data and it will provide the electricity to all nodes by using prim's algorithm and the system does not get affected but the value of minimum spanning tree cost is increased it is 35 for the mentioned fault to heal all the nodes in the system. For multiple link faults the designed system are not guarantee at all time to heal all the nodes in the system .During some multiple link faults sometimes it is not possible to heal all the nodes .

Conclusion

To overcome the scenario of power failure due to insufficiency of any system, a cost effective grid management system is developed. This system will not only help to resolve the problems due to power failure but also find the minimum spanning tree cost of routing the link of the network grid. Smart hybrid electric grid is developed, which is able to heal the fault or failure occurred in electric grid network .

When a fault arises due to failure of single or multiple links, the Smart hybrid grid analyses the problem and find the best possible route to maintain electric supply to each and every node of grid. This is done by Prim's algorithm which calculates minimum spanning tree cost of routing the link over the network grid. The Fault Location, Isolation and Service Restoration (FLISR) solution is based on Prim's algorithm . The prim's algorithm calculates the minimum spanning tree cost which enables to have minimum cost of maintaining the electrical supply under the case of failures of the links. The prim's algorithm for finding the minimum tree cost is implemented in Matlab software and also a graphical user interface is created in it, to demonstrate the grid structure.

In Matlab software a grid of 15 nodes and 2 electricity distribution centers connected to one other in typical manner is considered. It was found that with normal working situation out of 23 links of the grid only 16 links are used. when the link were given a typical weights as per the routing cost the minimum spanning tree cost was found to be 31 which is again the minimum among the total cost.

When the system encounters with a single link fault the smart grid find the new paths to heal all the nodes and keeping the system alive with minimum spanning tree cost. For single link consider example of "H" link if it fails then the system cannot heal all the nodes, because "N3" node having link from only "N2" nodes ,if electricity distribution center want to heal the "N3" node then one possible solution is mesh topology. Any Electricity distribution can't follows the mesh topology because it increases the cost of system .For the condition of multiple link failure, prim's algorithm tries to heal all the nodes of grid with minimum spanning tree, but cannot guarantee that all the nodes will be healed.

A multiagent-based distribution automation solution has been developed to restore the power supply in case of outages in a distribution network. The system is also tested on hardware bases using Raspberry Pi and arduino. The prim's algorithm is embedded in raspberry pi module and gives signal to the arduino board. This controls all the relays and establishes the connection to each and every node. The results obtained using the hardware section are same as obtained by software simulation done by Matlab.

References

[1] Markus Eriksson, Mikel Armendariz, Oleg O.Vasilenko, Arshad Saleem,Lars Nordstrom "Multiagent based distribution automation solution for self-healing grids", IEEE Transactions on Industrial Electronics, Year: 2015, Volume: 62, Issue: 4, Pages: 2620 - 2628, DOI: 10.1109/TIE.2014.2387098.

[2] Gaoxiang, Axin, "The Application of Self-healing Technology in smart grid", Asia-Pacific Power and Energy Engineering Conference (APPEEC), Year: 2011, Pages: 1 - 4, DOI: 10.1109/APPEEC.2011.5748373.

[3] H. Psaier and S. Dustdar, "A survey on self-healing systems: approaches and systems," Computing , vol. 91, Issue: 1, pp.43–73, 2010, DOI 10.1007/s00607-010-0107-y.

[4] Aboelsood Zidan , E.F.EI-Saadany , L. El Chaar , "A Cooperative Agent Based Architecture for Self-Healing Distributed Power Systems", Innovations in Information Technology (IIT), 2011 International Conference on Year:2011,Pages: 100-105, DOI: 10.1109/INNOVATIONS.2011.5893796.

[5] Khosrow Moslehi, Ranjit Kumar, "Vision for Self-Healing Power Grids", ABB Review, Year: 2006, Volume: 4, Pages: 21- 25.

[6] Vladmir Lonˇcar, Srdjan ˇ Skrbi´c and Antun Balaˇz," Distributed Memory Parallel Algorithms for Minimum Spanning Trees", Proceedings of the World Congress on Engineering 2013 Vol II,WCE 2013, July 3 - 5, 2013, London, U.K.