Decentralized Network Reconfiguration and Service Restoration using Priority based Load Management

Samikshya Mishra, *Odisha University of Technology and Research, Odisha*,
Rajan Kumar Mishra, *Silicon University, Odisha*,
Jasmine Hansda, *Odisha University of Technology and Research, Odisha*,,

***Abstract*—A new decentralized multi agent algorithm is proposed for service restoration in smart distribution grid. Each node identifies their inability to serve the load while finding alternate path for bringing back the power as fast as possible. Load priority also taken into account when alternate path for service is bottle-necked by any other constraint. This new algorithm is realized by open source software (python) and low cost computing infrastructure. In the present work, priority based algorithm is developed for resource scarcity based scenarios, increasing reach to number of connected customer. The algorithm negotiates with its environment depending upon the requirement of the node using popular and well established TCP/IP protocol. Multiple cases are studied for a small test system, along with modified RBTS test system.**

***Index Terms*—Demand side management, Multi agent system, Open source implementation, Multi-threaded operation, Network reconfiguration, Service Restoration**

I. Introduction

Demand for reliable, more efficient, and environment friendly energy is on the rise. This in turn gave rise to new trend in power network called smart grid. Smart grid enabled user to get a clear picture of the network in real time participation with the help of advance metering infrastructure[1], making it more transparent for improved control and efficient operation while maintaining sufficient privacy. It also provides us tools to estimate the failures. However it is not possible to estimate all possible contingencies in such a huge sophisticated network. Outages are inevitable in any network. Further addition of renewable sources and distributed generation to the customer end, increases the complexity to estimate all possible scenario. Centralized algorithm are becoming more complex as the network evolves which in turn taking a lot of time and computational resources to get a solution.

Smart grid also gave us new opportunities with improved computational infrastructures, active control, reliable communication at the back-end of power system. This yields new opportunity for a well developed autonomous, intelligent and decentralized computational multi agent systems(MAS)[2]. There have been many implementation of autonomous agent can be found in the literature. MAS has already been implemented for various sub function of power network for monitoring, protection, control, state estimation and so on[3], [4], [5]. In this work, MAS is used for improved distribution restoration with multi priority load for fast improved priority based self healing of network.

Multi agent system is an interactive platform where several agent communicate between themselves to reach their individual goals, which in turn progresses into achieving a global goal. Individual agent are autonomous with intelligent goal which may or may not be same for other agents in the system. There are many implementation of MAS for restoration in power distribution network. Most of the work improves the amount of load served as fast as possible to most number of feeders[5], [6], [7], [8], [9]. This limits the number of customer, or load points, getting connected after a contingency. As the full load is supplied to all customer in one node, which limits the number of node which can be restored in low resource condition.

Inclusion of load priority based management schemes, which can be implemented to improve number of load point connected to the grid, has never been implemented before. Most of the present day grid, has smart metering infrastructure which has the capability to store and transmit high priority and low priority load connected to the meter. This data can be used to restore major part of the distribution system rather than restoring a small part with full load restoration. In most of the emergency situation we have a resource scarcity condition due to available transmission capacity limit. Modern distribution systems are highly meshed for alternative path for maintenance or unavoidable circumstances. These meshed network, along with load priority, provides a new opportunity for restoring increased number of connected load where it is of high priority.

This work is based on power restoration in an agent based environment. All nodes in the system are based on multiple agent, depending on the interaction, agent can be of different type viz. generator agent, bus agent or load agent. Each of them are having an objective and limited constrained to obtain best possible solution for that particular node. Each agent has their priority involved to produce a brown out situation rather than complete blackout to the node. A new method to restore the load with priority is demonstrated in present paper. Here a parallel computing approach with multi-threaded operation is shown where more than one load is connected to the same bus via different feeders. A completely new framework is designed to minimize the data transmission and increase the efficiency of restoration process. Restoration time is also minimized so as not to affect any reliability indices of the system due to interruption.

II. Distribution Network Load Management

Advance Metering Infrastructures(AMI) are on the rise, which can give us precise data about each node with preferential load connected to it. This helps to segregate load into group of high priority and low priority loads. In this paper two different priority load are considered in each node, which will give rise to a large restored area with brownout zones. Last recorded metering data can directly fed to the agents for performing real time and accurate restoration as and when required. There are circumstances when the network is bottlenecked by transformer at some node can not provide power to the whole blackout region. This algorithm will help in that circumstances with emergency power restoration to high priority load with real time history analysis. Priority is based on individual customer priority instead of priority of nodes. So maximum number of customer will be restored with only high priority device getting power from the utility with the help of AMI.

Total load on any node is,

$$L_i = \sum p_i * L_{(i_j)} \tag{1}$$

where L_i is total load on i^th node, $L_{(i_j)}$ is one individual load in i^th, and p_j is the priority of a particular load. p_j is the percentage of load decided by the utility and the customer as high priority load. In normal system operation p_j is 1. In resource scarcity condition p_j will give rise to a value less than 1 which has to be restored with high priority. All node will have a predefined set which stores the value of p_j for every load connected to that node. This node will behave as one entity and take part in negotiation for restoration.

Most of the algorithm in the past has to shed the entire load for specific area for limited resource condition as centralized system has to solve for multiple load scenario to restore the system with different priority of load. Load management will help in restoring larger area with limited amount of resources.Load management takes AMI data before the outage and calculate the load requirement for a particular node. Smart grid's AMI and data accumulation in real time will help avoid shedding on particular load points by providing priority based loading at the time of restoration. As only node connected to the present load has to keep the data and make the calculation, it doesn't affect the algorithm's performance. In the present work, based on RBTS system, high priority is given to the government building and commercial building followed by only 30% of total load for residential and small industry. As residential and small industry load can be shifted. Only lighting and minimal loading in residential building is given high priority. Washing machine, refrigeration, air conditioning, swimming pool pumps are all low priority load only taken into consideration in residential area.

III. Decentralized Intelligent System

Centralize algorithm has its own advantages, as the result is mostly optimal as the centralized system has all the network data to reach its objective. But it falls short in terms of time taken for the algorithm to converge. The need for computational power increases, as the network complexity changes, as result of increased size of matrix manipulation.

In recent years we have seen dramatic improvement in the price and efficiency of computational and communication field. This made the implementation of decentralized computing more common with lesser computational burden to individual node an increasingly feasible solution. The only issue with distributed approach is the final system may not be fully optimal because optimal path might be blocked by less optimal ones depending on proximity of energized agents. Sub-optimal solution is the price what we pay for the fast, reliable, resilient restoration system. During the process of restoration the first objective is to minimize the load not served rather than optimality of the solution, this lead us to selecting agent based restoration algorithm.

The smallest computational node in a multi agent system in known as an agent. Each individual agent has their own objective and set of constraints. Individual agent has their intelligence to be opportunistic[10] for optimizing their local objective and constraints. The constraints depends on agent's environment, which is the data sent by its own ieds or from agent in its vicinity.

Each node must have some level of self-governing intelligence to reach its objective. It should be capable of sensing and improving the state of the node as fast as possible. Agent should have belief, desire and interaction(BDI) to reach its goal[11]. Belief are the local data, facts it has in the node itself, restoration case should have the details of load not supplied, extra power needed, addresses of nearest agent can be taken as the belief of individual node. Desire are the objective and constraints of the node. Interaction provides the negotiation capability.

IV. Restoration System Modelling

A. Agent architecture

In this work three different agent are considered, which can be used interchangeably depending on the environment and requirement of the node. Each node has all the agent active at all time in multiple thread. They can be active or inactive depending on the requirement of the node. Depending on the stage they are in, i.e. requesting for power, providing power, or the bridging the gap between both, they can be defined as load agent, generator agent or bus agent.

- Load agent(L_A): Every consumer, group of consumers, feeder or any clustered combination of any of the above, can be modeled as load agent. This is the unit which initiate the request or communication in the network. Depending on the last recorded data available to the agent, it will request for a connection to all possible agent in its vicinity. Responses from all possible agent, if there are multiple energized agent present in the vicinity and have sufficient capacity, who responded to the request is evaluated depending upon the voltage profile . Voltage profile can be taken as the measure of loading of the path, best to consider the path with lowest possible loading, to obtain an optimal solution. Acceptance is sent to the node with $max\{V_{i1}, V_{i2}....V_{in}\}$ where, $V_{i1}, V_{i2}...V_{in}$ are the precalculated value of voltage from parth $1, 2...n$, n is the number of connected node and $\{V_{i1}, V_{i2}....V_{in}\} \geq V_{min}$, V_{min} is the minimum allowable voltage at that node. Once it is energized, the search for connection ends and the request are no longer sent to other nodes. This result in stable system, which is radial at all time, without reconfiguring for better solution at all time reducing disturbance. So reconfiguration can only be introduced manually or by a centralized system, to get back to predefined state.

If all possible agent in the vicinity denies the initial power request, or voltage constraints can not be achieved, by present load condition then load agent starts exploring other possibility using load management. It will create a set of load with higher priority index. It will take a set and start communicating to the smart meters at customer end to power only priority loads. Meters are not programmed in the present work. This is just a tabulated information. In the 8 bus network it is considered that only one third of load on each node is of high priority for demonstration purpose.

- Generator agent(G_A): All the power producer or a substation comes under this category. Generator agent always listen to the requests from down stream L_As, but never initiates a request. G_A will wait for the request, once request is received, acceptance or denial flag is sent to L_As, depending on the available capacity. G_A will wait for the down stream agent for confirmation.If no confirmation comes within specific time frame, G_A deallocates the allocated power. G_A does not choose which one to accept or reject. Allocation is based on simply first come first serve basis. The selection of path is in the hand of loads only.
  Same L_A can request to one G_A via multiple path which is treated separately by the generator agent. This provides the L_A an opportunity to choose the better energized path. Algorithm also takes care of the meshed network where there are multiple paths possible from one load to the substation but select only one path to keep the network radial. Once the L_A is energized all requests are scratched by the G_A without any acknowledgement status from the L_A. This ensures the radial nature of the distribution network.
- Bus agent(B_A): Most sophisticated agent in the entire architecture is the bus agent. It has to listen to the request of loads down stream and initiate another for the generator upstream. Basically it serves both as generator and load agent, depending on the communicating part is upstream or downstream. It also relays the acceptance of load and acknowledgement of generator to one another. Once any load agent in energized it automatically convert it to bus agent with opening ports to listen for requests downstream. Once there is a power failure to this node, it changes its status again to load agent, by starting the request initiation process to all its neighboring agent. It also provides emergency load with high priority to its downstream by curtailing its own load. These curtailed load is again filled from the downstream if the downstream is having any less priority load to spare. This is the opportunistic model for the bus agent. Condition for providing low priority load to downstream, with higher priority than self, can be given by the following equation.

$$P_{li} < \sum u_j * P_{dlj}, \; j = 1...nd \tag{2}$$

Where P_{li} is the amount of low priority load on particular node, P_{dlj} is the low priority load required by any downstream node getting supplied from the particular agent, nd is the number of downstream node, u_j is boolean variable defining downstream agent is supplied with low priority load or not.i.e. if low priority load is supplied u_j is 1 , 0 otherwise.

From this introduction we can see a node can be G_A when it is constrained by limits. Same node can be a L_A when requesting for initial condition. All intermediate nodes are B_As. Every B_A node has one L_A and multiple G_A running in multiple threads. As B_A can receive request from multiple node, for which there is a connection possibility through tie switch, But only one upstream node which can approve the request. This is to ensure the radial characteristics of the distribution system.

B. *Objective formulation*

Power system restoration process can be described in three simple steps turn the light on, keep the light on and turn it back on when they go out completely or partially.

The objective taken for this architecture is to minimize the load not served or maximize the amount of load served. Mathematically,

$$max \sum_{i \epsilon N} x_i L_i \tag{3}$$

where x_i is the boolean variable which gives status of bus i i.e. bus i is connected or not, L_i is the load connected to the respective bus and N is number of total buses.

All load agent are responsible for maximizing the objective, L_A will keep on requesting for power, until they are energized, in multiple path parallely. If any one path is available, it will be connected to the main grid, followed by a negotiation.

Operational constraints should not be violated on the process of restoration. In this work, constraints under consideration are listed below, where equation 4 provides constraints for load generation balance, each G_A ensures this constraints, and equation 5 and 6 provides nodal constraints to each node. Voltage constraint is taken care of by the L_A at the time of negotiation to ensure voltage should be within limit. Each nodal power balance is kept in check with B_As.

$$\sum S_{Generation_i} \geq \sum_{i \epsilon N} S_{Load_i} \tag{4}$$

Here the S represent respective apparent power for load and generation in respective buses. G_As are accountable for maintaining these constraints. As G_As are the bottleneck of the system, they will only allow a load to be fed through them if and only if they have the reserve to serve.

$$V_i \geq V_{min} \tag{5}$$

$$P_i - L_i - \sum_{j\epsilon O_i} (P_j + P_{L_{ij}}) = 0 \quad (6)$$

Where, V_i is the nodal voltage in i^{th} node, V_{min} is the minimum allowable voltage, P_i and P_j is incoming power to bus i,L_i in load connected to bus i, O_i is the set of number of buses fed from bus i and $P_{L_{ij}}$ is the loss in the line carrying power from bus i to bus j. Line losses are simply calculated from the voltage drop between two nodes.

As most of the loop network and meshed distribution network is designed to transfer load from one path to another, there is no violation in the voltage profile in the demonstrated cases. These constraints are taken only as a check of system operating limits.

There might be some case few node might not get energized after the whole process, that would mean one of two things, there is no node adjacent to present node which is energized, or none of the energized node can provide power while satisfying the voltage constraints. If in some case voltage violation comes from all possible path attached to a node, that would mean there is no optimal path available to that particular node. If none of the adjacent node is connected in that case there is faulted line in between and no alternate path. These are few contingencies even exist in centralized algorithm.

C. Selection of communication system

In this work, LAN via ethernet/wi-fi is considered as the communication backbone, however any communication medium can be taken into consideration. As TCP/IP is standardized and independent of communication medium will not affect the simulation result. The delay may vary with channel under consideration without affecting the outcome. It is tested under lab environment with ipv4 for its popularity.

TCP is used over UDP for reliability of packet delivery with packet sequence to be intact in communication. As these protocol are independent of medium, and wide availability, interoperability with modern day wireless communication channel like mobile network and WAN, can easily be used in remote locationwith no additional installation cost.

Simulation is done on existing communication network, without any modification or special changes to the network. This procedure ensures the network condition and congestion of real world scenario. As internet works on the same protocol we can use any internet backbone network irrespective type of medium, $viz.$ optical, cellular, dedicated copper line.

D. Open source implementation

Each node is implemented in raspberry pi using python as the programming language.All individual node are programmed to behave as L_A/G_A/B_A, depending on their status and constraints. As raspberry pi is a low cost computing unit along with many advantageous GPIO integrated in it, this is more suitable for laboratory based testing and real world implementation.

Python is taken as the programming language of choice for its many advantages over other language viz. ease of use, open source royalty free program, low hardware requirement, availability of libraries for implementation of required protocols and so on. Multi threaded operation is utilized to open multiple thread to communicate with multiple node at any given time. Number of listening port initialization is limited to 5 considering no more that 5 branches are connected to any bus node.

Each node is connected with the help of an ethernet switch without DHCP. IP allocation is done by central router to get realistic network delays. Arduino UNO is used to collect data for switching on sequence every 10ms through serial connection at a baud rate of 9600. A simplistic model of the arrangement can be seen in figure-1a.

Only nodal interaction is done in this work, there is no physical distribution system. Computation in the agent will calculate the voltage drop before hand during negotiation process, there is no sensor to calculate physical drop in the network. GPIO pin from raspberry pi are used as physical interaction or switching and sensors for the nodal connectivity. There is no adc to detect physical drop only connectivity status flag will come to the computation framework.

Each node is provided with two data set. The first one store the required amount of load and its priority at each node which is gathered from automatic meter reading. The second dataset provides the ip addresses and port number of the neighboring node for communication.The whole system model can be easily changed by adding or removing one or more node at required place and providing ip address to its data set for the connecting node.

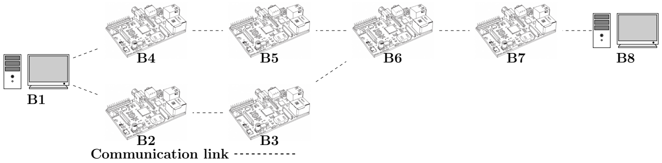
V. Test system

A. Test system-I: 8-Bus system

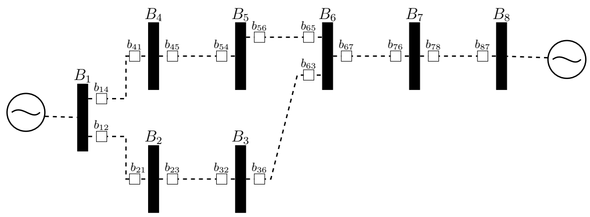
A small test system is considered for all possible scenario to be demonstrated properly. System under consideration is a 8 bus system, where the bus 1and 8 are taken as generator bus of the system. It can be taken as a point of common coupling for the small system. Both bus 1 and 8 are allocated with 7.5MW and 4.5MW. This is taken as the rating of the transformer at the point of common coupling. Power factor is considered to be unity. Lines are considered to be purely resistive, as the result shows flow of real power only as a proof of concept.

From the figure 1b we can see arrangement of the network. Here each "$\mathbf{B_i}$" represents i^th bus and each breaker is denoted by "b_{ij}"where i represents the bus number and the j signifies preceding bus. Each node is having a load of 1.5MVA, out of which 0.5MVA is emergency load. Breaker of individual node is named by the convention that first letter is the connected bus and second signifies the bus number the branch is leading to. In practical distribution system switching is done by tie switches and sectionalizing switches. each line in the 8 bus system is having uniform impedance of $(0.366 + j0.358)\Omega$. Voltage level is considered to be 11kV at substation level. Minimum allowable voltage to connect a load is 0.9pu. Impedance is choose such that under full load when supplied from one source there will be an intentional voltage limit violation.

Fault in the system is fed manually by switching off one of the generator node, forcing the bus node to get a feeder off-



(a) Test setup for agent based restoration



(b) Distribution System Representation

Fig. 1: 8-Bus System Schematic and Network Model

line flag. It is considered that protection system is working properly and can isolate the fault in an effective manner.

B. Test system-II: RBTS system

A part of Roy Billinton Test System (RBTS)[12], BUS2, is taken as proof of concept that it will work in large network without much network delays. System under consideration consists of 14 sectionlizing switches, 2 tie switches, 22 load points(LP) distributed over 4 laterals. RBTS system is modified and all the load points are clustered depending on the sectionlizing switch and tie switch location.Modified system diagram is given in figure-2 This also provides a combination of different types of load viz. commercial, residential, small industrial and government buildings. Load data is stored in tabular fashion to feed into the agent model as requested by the nodal agent. Typical aluminum conductor steel reinforced(ACSR) conductor data is taken as per 50Hz distribution system with impedance of $(0.39 + j0.27)\Omega/km$

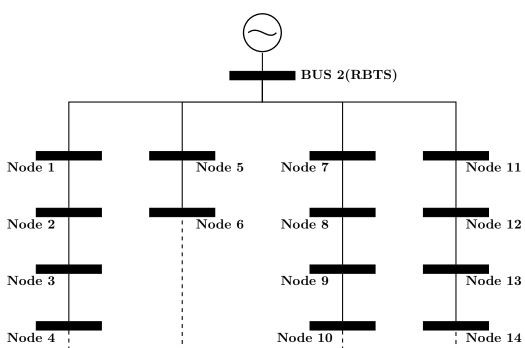


Fig. 2: Modified RBTS system diagram

Total 15 cluster nodes are considered. Detail description of all load connected to each node along with average loading is given in the table-I below.

VI. Agent interaction for system restoration

Restoration can be initiated from complete blackout situation or from failure of any bus in the system. For the test purposes two scenario are chosen one is restoring after complete blackout, another is the failure of one of the generator agent i.e. one of the feeding substation. Both the cases are considered to be starting after fault is identified and isolated.

TABLE I: Agent description for RBTS system

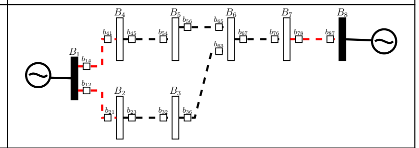
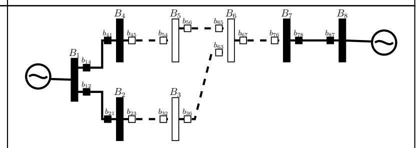
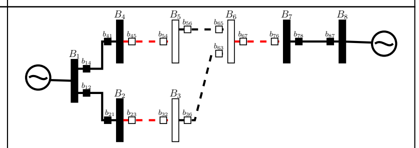
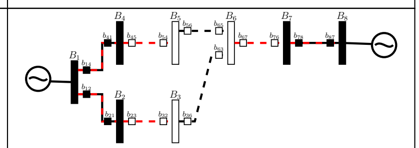
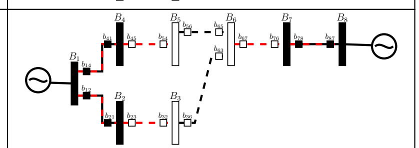
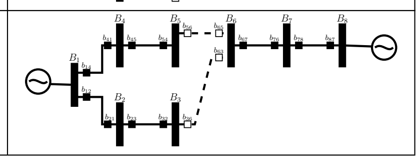
Node	Connected LP	Total Load(MW)	Node	Connected LP	Total Load(MW)
0	11kV Bus				
1	1,2	1.07	8	11,12	0.985
2	3,4	0.985	9	13,14	1.132
3	5,6	1.02	10	15	0.454
4	7	0.454	11	16,17	0.904
5	8	1	12	18,19	0.9
6	9	1.15	13	20	0.566
7	10	0.535	14	21,22	1.02

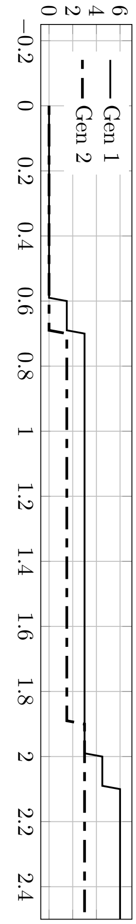
A. Blackout restoration

Present scenario provides insight into black start of the agents when there is no power at all. All the node agents are actually running but are getting no response from the neighboring agents as listening ports are not activated, till the node is energized. The first bus will be energized from external connection. This will start the following chain of event. Communication between different agent is given pictorially in table II. Dotted lines are representation of open lines and solid line shows power flow. Red dashed line show communication between neighboring agent. Failed communication may be because of no active listening port or no initiation of any request. Till the listening port of the neighboring agent is open communication will not start, so there is no red line representation between two de-energized nodes.

- Energizing bus1 agent will activate listening port for G_A of bus1 which in turn receives the request from L_A of bus 2 and 4 as E_1 in table-II. It will be stored in s queue and processed one by one. As selection is not in the hand of the generator it has to be handled one by one. The delay for every transaction is set to 500ms. So if there is no response from the load it will be unallocated automatically processing the next request. If generator has enough reserve it can accept both request at the same time from the queue.
- L_A in bus 1 will compare the allocated power with the request and send an approval to bus 2. G_A of bus 2 upon getting the approval request, wait for 250ms before responding for better alternative, sends the acknowledgement signal to bus 1. This negotiation will be parallel in different threads.
- G_A of bus1 will close the breaker b12 with breaker close signal, which in turn asks L_A in bus 2 to close b21. At the same time of closing of breaker b12 bus1 G_A continues processing request from bus4. In practical situation there might not be breakers next to all nodes, and remote tie switches will replace the breakers.
- The moment bus 2 get energized it will open its listening port by changing its mode from L_A to B_A. This will give an opportunity to bus 3 to connect and submit the request(E_3). This request is forwarded to bus 1 shown in event E_4.

TABLE II: Agent negotiation during restoration in 8-bus test system

Event	Node status	From	To	Message	Time (sec)	Network
$E_{\_1}$	B_1:G_A B_5:L_A B_2:L_A B_6:L_A B_3:L_A B_7:L_A B_4:L_A B_8:G_A	B_2 B_7 B_1 B_8 B_4 B_1	B_1 B_8 B_2 B_7 B_1 B_4	req req acp acp req acp	0	
$E_{\_2}$	B_1:G_A B_5:L_A B_2:B_A B_6:L_A B_3:L_A B_7:B_A B_4:L_A B_8:G_A	B_2 B_7 B_1 B_8 B_4 B_1	B_1 B_8 B_2 B_7 B_1 B_4	ack ack brc brc ack brc	0.6	
$E_{\_3}$	B_1:G_A B_5:L_A B_2:B_A B_6:L_A B_3:L_A B_7:B_A B_4:L_A B_8:G_A	B_3 B_5 B_6 B_2 B_4 B_7	B_2 B_4 B_7 B_1 B_1 B_8	req req req req req req	0.6	
$E_{\_4}$	B_1:G_A B_5:L_A B_2:B_A B_6:L_A B_3:L_A B_7:B_A B_4:L_A B_8:G_A	B_1 B_1 B_8 B_2 B_4 B_7	B_2 B_4 B_7 B_3 B_5 B_6	acp acp acp acp acp acp	0.7	
$E_{\_5}$	B_1:G_A B_5:L_A B_2:B_A B_6:L_A B_3:L_A B_7:B_A B_4:L_A B_8:G_A	B_3 B_5 B_6 B_2 B_4 B_7	B_2 B_4 B_7 B_1 B_1 B_8	ack ack ack ack ack ack	1.8	
$E_{\_6}$	B_1:G_A B_5:B_A B_2:B_A B_6:B_A B_3:B_A B_7:B_A B_4:B_A B_8:G_A	B_1 B_1 B_8 B_2 B_4 B_7	B_2 B_4 B_7 B_3 B_5 B_6	brc brc brc brc brc brc	2	



$V_{min} = 0.966pu$ after reconfiguration
Flags: req=Request, acp=accept, ack=acknowledge, brc=close path

- Bus 1 will have the same interaction steps with bus 3 but with an intermediary bus 2 B_A as shown in $E_{\_5}$. Same sequence is followed with bus 4 and 5.
- Bus8 G_A will follow similar steps for bus 6 and 7. Each of the generator agent will have a copy of power allocation to every bus they are suppling. But the IP addresses and port number is only with the neighboring nodes. End result of restored system given in last column of table-II in $E_{\_6}$ row.
- Minimum voltage through the network is seen to be,0.97pu, which is well above voltage limit.

B. Partial failure restoration

Partial failure restoration is more complex than complete restoration because in full restoration all of the device are considered to be working. Once the outage is cleared system will come back to normal without any problem. Where as for partial failure chances of local failure might be an issue. So we cannot provide full load to all nodes. This is also true for partial allocation of requested power. For this case we have considered bus 1 failure as this is handling maximum power and it will push the network to its knees without having sufficient dispatch-able power. As the other part of the system is healthy bus 6, 7, 8 are already energized. So bus 6 will receive the first request, the sequence of event are listed below. Dashed red line in table-III represents communication initialization but not restored where as dark black line represents restored path. Gray buses represents partially restored load in the present node.

- Bus 6, B_A will receive the request from bus 3 and 5 as shown in $E_{\_1}$ in table-III as soon as there is a power failure without any fault flag. As bus 3's L_A request is received first it is processed as mentioned before.
- The moment bus 6 processes bus5's request it get denial flag from source due to unavailability of sufficient reserve at bus 8($E_{\_2}$) . But the first negotiation $E_{\_2}$ is approved and bus3 is energized.
- On getting denial from bus 6, node 5 will send request again with higher priority and reduced load demand($E_{\_3}$). This forces bus 6 to shed its own low priority load and provide the same to bus 5 for higher priority load. This condition only works if the amount low priority load connected to bus 6 is higher than the amount of high priority load of bus5. At the same time bus 6 will send a signal to deallocate the extra consumption at its node. This will help another emergency restoration possible or can provide the same for another node whose load requirement is less.

TABLE III: Agent negotiation during resource scarcity condition in 8-bus test system

Event	Node status	From	To	Message	Time (sec)
$E_{\_1}$	B_1:NA B_5:L_A	B_3	B_6	req	6.0
	B_2:L_A B_6:B_A	B_5	B_6	req	
	B_3:L_A B_7:B_A	B_6	B_7	req	
	B_4:L_A B_8:G_A	B_7	B_8	req	
$E_{\_2}$		B_8	B_6	acp/dec	6.7
	B_1:NA B_5:L_A	B_6	B_3	acp	
	B_2:L_A B_6:B_A	B_6	B_5	dec	
	B_3:B_A B_7:B_A	B_3	B_6	ack	
	B_4:L_A B_8:G_A	B_6	B_8	ack	
		B_8	B_3	brc	
$E_{\_3}$		B_5	B_6	emreq	6.6
	B_1:NA B_5:L_A	B_2	B_3	req	
	B_2:L_A B_6:B_A	B_3	B_6	req	
	B_3:B_A B_7:B_A	B_6	B_8	req	
	B_4:L_A B_8:G_A	B_8	B_2	dec	
		B_2	B_3	emreq	
$E_{\_4}$		B_6	B_5	acp	7.4
	B_1:NA B_5:B_A	B_3	B_2	acp	
	B_2:B_A B_6:B_A	B_5	B_6	ack	
	B_3:B_A B_7:B_A	B_2	B_3	ack	
	B_4:L_A B_8:G_A	B_6	B_5	brc	
		B_3	B_2	brc	
$E_{\_5}$	B_1:NA B_5:B_A	B_4	B_8	req	8.2
	B_2:B_A B_6:B_A	B_8	B_4	acp	
	B_3:B_A B_7:B_A	B_4	B_8	ack	
	B_4:B_A B_8:G_A	B_8	B_4	brc	

$V_{min} = 0.926pu$ after reconfiguration
Flags: dec=decline, ack=acknowledge, emreq=emergency request, brc=close path

- L_A of Bus 4 will then initiate its request through bus 5 and 6 B_A to bus 8 G_A, which will again end in denial as shown in $E_{\_5}$. The high priority request on the other hand will be accepted by Bus 8's G_A which was deallocated from bus 6 will be assigned to bus 4 emergency load.
- Same will happen to the request of bus 2 to bus 3.
- As bus 6 is the opportunist node. It will request back the unallocated power to G_A whether it meets its present demand. This process will go on in every few minutes to check for availability and improve it's status to full load restoration. Status after restoration is shown in last figure of table-III. At the end of restoring and reconfiguring the network minimum voltage seen to be 0.926pu.
- Even if the G_A at bus 8 is capable of supplying sufficient load to the end of the node after connecting every load in it's full capacity, voltage at bus 2 and bus 4 is at 0.895pu, so both cannot accept the path. So in that case only one ,either bus 2 or bus 4, has to serve it's emergency load which brings the voltage to acceptable limit.

VII. Results and discussion

A. Test system-I: 8-Bus system

Agent interaction mentioned above is realized in real time using the given hardware. The blackout restoration works flawlessly in scenario when all the equipments are fine and load allocated to the generator agent is higher than the load required by all the load agent combined, as seen in case I. After the system is restored from blackout another contingency is realized at 6th second to see the system response by creating a fault in bus 1. All results are collected from polling the files, responsible for storing all the connected lines and power flow through it, every 100milisecond to get real time results.

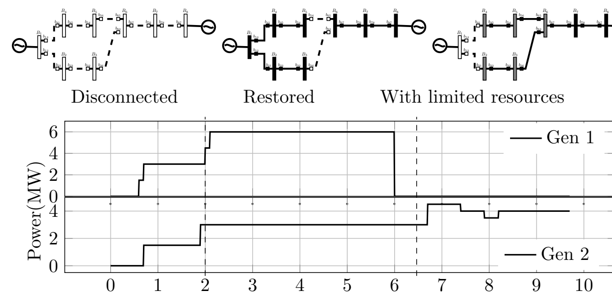


Fig. 3: Power flow through bus during restoration

Detail of power flow through the bus is given in figure-3. Here we can see only changes in bus 2 to bus 6. Because there is no change in bus 7 load as it is not having any role in partial load restoration. So total load consumed by all load is 4MVA i.e.Bus 7's load consumes 1.5MVA along with 2.5MVA through bus 6. This method may be inefficient in condition where the requirement of load is higher than allocated power to generator node, because 0.5MVA power is unallocated to any of the agent in case II as seen by simulation. The node agent has only two switch for only two different priority load and none of the node is having 0.5MVA requirement in either high or low priority load. But this is the fastest possible way to restore the power distribution system.

B. Test system II: RBTS system

Test system II or modified RBTS system restoration duration with respect to each feeder is evaluated. Here we can see there is a time delay between each load pick up. This is mainly due to establishing the connection, hand shaking, allowing and acknowledging the path for power flow. But once it is active the path is closed.

The moment power is acknowledged broadcasting for request is no longer required. As the system has to be radial in all time, upstream path for power flow remains the same. So all downstream request go only to one specified address. Energized load will act as a bus agent which has multiple port receiving for request to only one upstream. Multi threading operation of distributed computing shows a vast improvement at this time as shown in figure-4. Because of multiple parallel path system restoration of all four feeder is equivalent to the time required to restore one feeder. Total restoration time for these system is well within 5 seconds which is a vast improvement over traditional system.

There is failure of line,from bus 2 to node 13, is also simulated at 6 second after load is restored to the system and reconfiguration is show. Ring mode of the network reconfigures the system after the failure. Only in the last node, i.e. node 13, power is not restored even if bus 2 in the system is capable of the extra load because of the voltage violation. If full load is connected to the system from node 13 minimum voltage seen by the load is 0.897 pu, which is below specified limit of 0.9pu. So, only emergency load which it taken as 30%

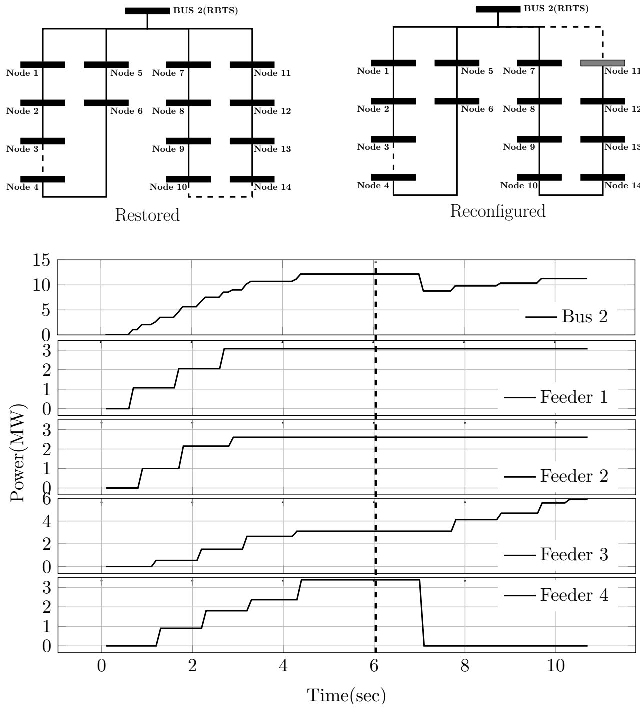


Fig. 4: Restoration and Reconfiguration of RBTS system

is fed to keep the voltage profile of the system under limit. After complete restoration V_{min} is 0.916 pu. Detail V_min of all the feeder after each operation is given in table-IV

TABLE IV: Voltage profile of different feeder of RBTS system

	Fdr 1	Fdr 2	Fdr 3	Fdr 4
V_{min} after Restoration	0.979	0.986	0.975	0.972
V_{min} after Reconfiguration	0.979	0.986	0.916	no load

VIII. Conclusion

This work presents a decentralized algorithm using MAS for automatic power distribution system restoration using minimal hardware with existing system infrastructure employing local area network and TCP/IP protocol. Different operating scenarios are simulated for two case studies of 8 bus system and a clustered RBTS system. The software requirement was shown to be of open source with low cost in terms of speed and efficiency. Potential benefits of distributed computing with multi-threading were utilized to improve the system restoration time for laboratory scale model, which is easily scalable. It has been shown that the proposed restoration model provides a robust architecture since the nodes are independent of each other, which ensures that failure of any node will not hamper the performance of the system.

References

[1] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473–484, 2014.

[2] A. Saleem and M. Lind, "Requirement analysis for autonomous systems and intelligent agents in future danish electric power systems," *International Journal of Engineering, Science and Technology*, vol. 2, no. 3, pp. 60–68, 2010.

[3] T. Logenthiran, D. Srinivasan, and A. M. Khambadkone, "Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system," *Electric Power Systems Research*, vol. 81, no. 1, pp. 138–148, 2011.

[4] Z. Wang and J. Wang, "Service restoration based on ami and networked mgs under extreme weather events," *IET Generation, Transmission & Distribution*, 2016.

[5] J. M. Solanki, S. Khushalani, and N. N. Schulz, "A multi-agent solution to distribution systems restoration," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1026–1034, 2007.

[6] C. P. Nguyen and A. J. Flueck, "Agent based restoration with distributed energy storage support in smart grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 1029–1038, 2012.

[7] M. Eriksson, M. Armendariz, O. O. Vasilenko, A. Saleem, and L. Nordström, "Multiagent-based distribution automation solution for self-healing grids," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2620–2628, April 2015.

[8] J. Lagorse, D. Paire, and A. Miraoui, "A multi-agent system for energy management of distributed power sources," *Renewable Energy*, vol. 35, no. 1, pp. 174–182, 2010.

[9] V. J. Garcia and P. M. França, "Multiobjective service restoration in electric distribution networks using a local search based heuristic," *European Journal of Operational Research*, vol. 189, no. 3, pp. 694–705, 2008.

[10] P. Li, B. Song, W. Wang, and T. Wang, "Multi-agent approach for service restoration of microgrid," in *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*. IEEE, 2010, pp. 962–966.

[11] A. S. Rao, M. P. Georgeff *et al.*, "Bdi agents: From theory to practice." in *ICMAS*, vol. 95, 1995, pp. 312–319.

[12] R. N. Allan, R. Billinton, I. Sjarief, L. Goel, and K. So, "A reliability test system for educational purposes-basic distribution system data and results," *IEEE Transactions on Power systems*, vol. 6, no. 2, pp. 813–820, 1991.