The Pennsylvania State University

The Graduate School

College of Engineering

EFFECT OF WIRELESS MESH NETWORK PARAMETERS ON SMART GRID MONITORING AND CONTROL

A Thesis in

Electrical Engineering

by

Lucas McCoy

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2014

The thesis of Lucas McCoy was reviewed and approved* by the following:

Peter Idowu
Assistant Dean for Graduate Studies
Professor of Electrical Engineering
Thesis Advisor

Seth Wolpert
Associate Professor of Electrical Engineering

Scott Van Tonningen
Associate Professor of Electrical Engineering

Sedig Agili
Professor of Electrical Engineering
Program Coordinator for Master of Science in Electrical Engineering

*Signatures are on file in the Graduate School

ABSTRACT

As the power grid transforms from a unidirectional flow of power to a bidirectional flow of power and data, the need to deploy a fast, reliable, and expandable communications network will be one of the biggest challenges faced by utility companies. The communications network must be capable of supporting the many requirements of a smart grid as set forth by the United States Department of Energy's 2010 Smart Grid System Report (dynamic pricing, real-time system operations data sharing, load participation, distributed generation, grid-responsive demand-side equipment, advanced metering, and renewable resources). This research will focus specifically on evaluating a wireless mesh network in regards to the impact of network characteristics on power control algorithms such as power flow. This research will establish why a wireless mesh network is a reasonable communications network for use in the distribution level of the power grid, define a model for the network based on the characteristics of the radios, and evaluate the impact of the modeled network on power control systems. The development of such a model is crucial in allowing power utility companies to deploy networks that can provide communications capable of maintaining control efficiency and stability while also minimizing deployment cost.

TABLE OF CONTENTS

List of Figures......vi

List of Tables......vii

List of Acronyms......viii

Acknowledgements......ix

Chapter 1 Background......1

Chapter 2 Wireless Mesh Network......8

  Overview......8
  Network Topology......11
  Network Characteristics......18
  Network Model......22
  Smart Grid Communications......24

Chapter 3 Power Flow......26

  Overview......26
  Power Flow Model......27
  Results......28

Chapter 4 Conclusion and Results......35

  Combining Communications and Power Networks......35
  Implications......37

Chapter 5 Future Work......38

Appendix A - WMN_script......45

Appendix B - WMN_readfile......51

Appendix C - WMN_power......53

Appendix D - WMN_routes......55

Appendix E - WMN_paths......59

Appendix F - WMN_pathcharacteristics......60

Appendix G - WMN_utilization......63

Appendix H - WMN_TAPplacement 65

Appendix I - Nodes.xls 66

Appendix J - PF_test 67

Appendix L - PF_Variablesystem 70

LIST OF FIGURES

Figure 1 - IEEE 30 Bus Test Case 10

Figure 2 - Node Location Plot 10

Figure 3 - WMN_power flowchart (Direct communication links and node transmit power) 13

Figure 4 - WMN_routes flowchart (Multi-hop network map)....................16

Figure 5 - WMN_paths flowchart (Check for excluded nodes)....................17

Figure 6 - Node connection plot (Test network with all nodes transmitting at 36 dBm).........18

Figure 7 - WMN_pathcharacteristics flowchart (Communication path characteristics)..........20

Figure 8 - WMN_utilization flowchart (Node utilization)....................21

Figure 9 - WMN_TAPplacement flowchart (TAP placement)....................22

Figure 10 - Node connection plot for network model....................23

Figure 11 - IEEE 30 Bus Test Case with Branch Identifiers 28

LIST OF TABLES

Table 1 - Network Parameters 8

Table 2 - Node Parameters...8

Table 3 - Network Mapping Example 15

Table 4 – Network Model Characteristics 23

Table 5 - Power Flow Results Summary 29

Table 6 - Power Flow Results..31

LIST OF ACRONYMS

AC	Alternating current
AMI	Advanced Metering Infrastructure
DC	Direct current
DER	Distributed Energy Resources
DG	Distributed Generation
DOE	United States Department of Energy
ED	Economic Dispatch
FCC	Federal Communications Commission
KV	Kilovolt
MHz	Megahertz
MW	Megawatt
NOAA	National Oceanic and Atmospheric Administration
OPF	Optimal Power Flow
PF	Power Flow
TAP	Transit Access Point
VC	Venture Capital
WMN	Wireless Mesh Network

ACKNOWLEDGEMENTS

Thank you to my family and friends for supporting through this entire process. Thank you to my professors that have expanded my knowledge in their respective areas especially my advisor for all of the help and guidance offered throughout my graduate studies.

Chapter 1

Background

As the existing power grid in the United States ages, significant updates must be made to ensure continued operation, efficiency, and stability of the grid. These updates will include integrating existing technologies with the power grid as well as creating new technologies designed specifically for improvement of the power grid. The integration of technologies with the power grid will define a new type of power distribution network known as a smart grid. When defining the scope of a smart grid, the United States Department of Energy (DOE) describes several main areas in its 2010 Smart Grid System Report [1] that will shape the direction of this research. These areas include distributed-energy resource (DER) technology; delivery transmission and distribution infrastructure; and information networks and finance. DER consists of distributed generation (DG), storage, and demand-side (load shedding) resources integration in the power grid. Distribution infrastructure includes distribution automation and advanced metering infrastructure (AMI). Information networks include a pervasive communications network as a cornerstone of a smart grid. All of these key areas of a smart grid demonstrate that a critical component to smart grid implementation and operation is the communications network.

The DOE defines six characteristics that will be used to measure the progress of smart grid development: enabling informed participation by customers; accommodating all generation and storage options; enabling new products, services, and markets; providing power quality for the range of needs; optimizing asset utilization and operating efficiency; and operating resiliently: disturbances, attacks, and natural disasters. While all of these characteristics either depend on or are enhanced by communications, two will be addressed in particular. These are enabling informed participation by customers and accommodating all generation and storage options. In a traditional power system, end-users generally do not intelligently schedule power usage, meaning that the decision to use power is based on factors other than the current status of the power grid. In a smart grid, new technologies enable the end-users to make intelligent power usage decisions based on grid conditions and pricing. These technologies are heavily dependent on bidirectional communication and power flow to provide the data to the end-user and allow the end-user to participate in power generation if desired. Traditionally power generation has occurred at large, centralized power plants. In a smart grid, a wide variety of DER will be connected to the grid. These distributed generation and storage resources will require monitoring and control and require reliable, pervasive communications infrastructure. The DOE has identified 21 key metrics to track the progress of smart grid implementation. Some of these metrics that depend on bidirectional communication throughout the grid are dynamic pricing, load participation, grid-connected distributed generation and storage, grid-responsive non-generating demand-side equipment, and advanced metering. Due to the reliance on communications of many key aspects of a smart grid, a “scalable and pervasive communication infrastructure is crucial in both construction and operation of a smart grid.”[2]

The current interest in developing a smart grid is indicated by the amount of government and private funding being directed at smart grid technologies and infrastructure. As of the 2010 Smart Grid System Report, a total of $3.4 billion in government grants have been awarded to encourage development of a smart grid. The total amount reaches $8.2 billion with the addition of private sector investments. From the government grants, at least $812.6 million has gone towards AMI deployments. DOE policies and guidelines have also facilitated the development of DER interconnection policies in 14 states since 2008, further expanding the potential of distributed generation technologies. Private venture capital (VC) funding of startups developing smart grid technologies has increased from $58.38 million in 2002 to $414 million in 2009. From 2007 through 2010, 29% of the VC funding for smart grid startups was for meter communication [1]. This means that meter communication is the single largest category for VC funding in smart grid technologies. This means that VC analysts see a real need for the development of a communications means for AMI and expect a return on that investment. The connection between AMI and DER is strong because AMI is meant to communicate pricing data and grid status between the end-user and power utility company to facilitate energy awareness, demand response, and distributed generation. This information will be crucial to the owners and operators of DER to make decisions regarding the desired energy generation or consumption of the DER. These AMI and DER communication requirements will exist throughout the distribution level of a smart grid.

The advancement of AMI and DER will also have a large impact on power pricing, stability, reliability, and quality. A smart grid relies on AMI and DER to facilitate the use of various power generation and storage methods, including renewable energy resources. The intermittent nature of renewable energy sources throughout the grid provides problems and solutions to help meet peak demand, supply power during disturbances, and reduce overall power costs. These functions require a communications network to enable active monitoring and control of the DER. One of the reasons that communication is essential in a smart grid is the increase in DER and bidirectional power flow. The traditional power grid was designed to be unidirectional and radial [2]. The deployment of DER has already begun and a pervasive communications technology is required to fully utilize the capacity. DG powered by fossil fuel, hydroelectric, and biofuels increased 136 percent from 2004 to 2008 [1]. This expansion of DER is further increased when renewable resources such as wind and solar are factored in. To improve power reliability and quality, a smart grid enables intelligent devices to identify the location of disturbances, isolate faults, restore service, and monitor equipment remotely. These intelligent functions allow a smart grid to more efficiently and reliably operate transmission and distribution networks. These intelligent functions all require a communications network to remotely monitor and control the smart grid devices.

The communication system for the smart grid consists of various layers. One layer is at the generation level. The focus of this research will not be at this level because central generation control systems are considered advanced enough that future changes are less likely to be revolutionary and are not considered a key area for smart grid progress at this time [1]. Another layer is at the transmission level. This level of communication is crucial for planning, maintenance, and fault detection on high capacity lines running throughout the country. This communication layer is accessed by the transmission companies, large scale generation companies, and distribution companies. Due to the great deal of power already flowing through these lines, the technological capabilities of the equipment being installed, and the capital investment involved with these installations, reliable, high speed communications links are often already installed for this equipment. In practice, these networks are typically associated with high voltage lines only and not suitable for wide-spread implementation throughout the grid [2]. Another layer of communications in a smart grid exists at the distribution level. Unlike the transmission level communications network, this network will be accessed directly by end users in a smart grid and does not exist in the current power grid. This network can be used for advanced metering, dynamic pricing, real-time system operations data sharing, grid-responsive demand-side equipment, and grid-connected renewable resources. These uses are all defined as key metrics for smart grid implementation by the DOE in [1]. These functions require the network to be widespread, bidirectional, high speed, and robust.

A key component to a smart grid is the bidirectional transfer of electricity and data enabling an automated and distributed power grid. The integration of communication with the legacy power grid provides the ability to rapidly balance supply and demand. A network at the distribution level is necessary to support AMI and monitor current electrical load on the grid. Another benefit of a pervasive communications network is the possibility for protection coordination and cooperation at the distribution level [2]. Widespread protection coordination can help prevent massive blackouts such as the 2003 blackout that affected a large portion of the power grid in the northeast United States and parts of Canada. As the power grid evolves, a particular emphasis will be placed on distribution networks because they currently have limited active control and the expected increase of DER [3]. This increase in DER will also create additional strain on a system not originally designed for bidirectional power flow. Numerous articles and papers have been published demonstrating the need for communications networks in transforming the existing power grid in to a smart grid. Some of the common themes seen in publications [4] through [14] are the possibilities of using the communications aspect of a smart grid implementation to allow for load shaping instead of load response, allowing for the integration or renewable energy resources, DER, AMI, and overall grid reliability. The needs to start the development of the smart grid in the distribution network as well as make the transition to the smart grid evolutionary starting with remote monitoring and control alongside the existing power grid are discussed in [6].

With the need to establish a pervasive communications network throughout the smart grid (particularly at the distribution level) established, a network technology must be chosen. For this research a wireless mesh network (WMN) will be considered as the technology for the distribution level communication networks. The reasons for choosing a WMN include the robustness to failure, high speed capabilities, expandability, and deployment costs. Some of the key features of a WMN are rapid deployment, minimal configuration, high speed communication, and the ability to be reconfigured. For robustness, WMNs can be designed to operate in disaster situations when other infrastructures (including the power grid) may be nonoperational [15]. In these situations, the data provided by the communications network is crucial to rapid diagnosis and response to reestablish utility service. WMNs are also resilient to individual node failures by the very nature of the network. Mesh nodes are capable of spontaneously creating new multi-hop paths through an unplanned network [16]. For high speed capabilities, the speed of the WMN is determined largely by the size and density of the network and the number of transit access points (TAPs). When properly deployed, a WMN is capable of high speed communications. For expandability, WMNs are scalable in the sense that a local network can be expanded in to a wide area network by placing additional nodes and TAPs. When considering the deployment of a grid-wide network, it is important to consider that wireless infrastructures are becoming less expensive to deploy and maintain than wired network infrastructures [17].

With a network technology chosen, it is important to intelligently design the network layout to optimize performance and cost. Defining the effect of performance characteristics of the smart grid communications network on power control algorithms is the core topic of this research. Various publications exist to examine the impact of communications network in smart grid implementations [18]-[24] as well as the performance characteristics of WMN in general [25]. Due to the fact that a smart grid is defined by distributed monitoring and control, the role of the communications infrastructure becomes increasingly more important. As the reliance on the communications network increases, the performance and stability of the smart grid as a whole will depend on the performance of the communications [26]. The work in [27] shows a simulation of a simple power control system and the effects of the network characteristics on the control system. There are six topological characteristics that impact WMN performance: nodes in each sub network, mean hop count, neighbor node density, number of hidden nodes, number of nodes in the neighborhood of the gateway(s), number of hidden nodes in the neighborhood of the gateway(s) [28]. One of the main complexities in WMN is the difficultly in intelligently designing the network to achieve near-optimal performance and robustness [29]. A key component to WMN performance is the placement of TAPs. The optimal placement of TAPs is also affected by the design requirements of the network. Several TAP placement algorithms exist because the problem of placing TAPs to maximize capacity is different than minimizing latency [16]. This research will explore the TAP placement problem as it relates to modeling a WMN deployment to satisfy the latency and capacity requirements of power control algorithms. A TAP placement algorithm will efficiently utilize wireless capacity, account for wireless interference on performance, and be robust against failures. The general goal of a TAP placement optimization algorithm is to place the minimum number of TAPs required to achieve the necessary network performance [30]. This is because TAPs are more expensive than other wireless nodes because they have multiple network interfaces and also require the additional infrastructure interconnect to the wired network.

The goal of this research is to generate a MATLAB model for a WMN that can be used to aid in the development of both the control algorithms and network layout of a smart grid implementation. For control design, the model can be used to determine the network characteristics between the equipment and controller and design the controller to maintain stability during operation over the network. The model can also be used to evaluate the effect on other power control algorithms such as power flow (PF), which will be the power control algorithm evaluated in this research. For network design it can be used to layout the network and determine the number and location of TAPs for the WMN. This would be accomplished by defining the desired response for the network (based on the requirements of the control algorithms) and minimizing the nodes and TAPs need to fulfill those requirements. The decision to generate the model in MATLAB is in part based on the desire to allow for easily integrating the network model with existing power simulation packages. MATLAB is a widely used software package with a large support network surrounding its user base. This existing user and support base increases the chance of compatibility with other software packages for co-simulation. Co-simulation is the process of combining the power and communications network simulations to determine their performance on each other. Co-simulation is a common practice for smart grid simulations as the communications and power networks are dependent on each other [31]-[33].

Chapter 2

Wireless Mesh Network

Overview

To properly model a wireless mesh network, certain parameters of the network must be specified to determine the performance of the network. For this research the parameter assumptions used can be seen in Table 1. These assumptions were made based on reasonable values for this type of network as well as Federal Communications Commission (FCC) regulations related to wireless transmission power. All of these parameters are configurable in the MATLAB scripts used to generate the network model. This allows a user to easily generate a new model based on the specifications, such as frequency and bandwidth, of various wireless radios. The node parameters are also configurable through the use of an excel file as shown in Appendix I - Nodes.xls. These parameters utilize an input file instead of manual entry because of the possibility of a large number of nodes in the network. The parameters used for this research can be seen in Table 2.

Table 1 - Network Parameters

Parameter	**Value**
Signal Frequency	900 MHz
Transmit Power Maximum	36 dBm
Forwarding Time	0.10 s
Message Length	8000 Kb
Network Overhead	1 Kb/s per node
Duplex or Simplex Radios	Duplex
Direct Link Minimum	2
TAP Percentage Maximum	50%

Table 2 - Node Parameters

Parameter	**Value**
Transmit Power Minimum	6-36 dBm
Receive Sensitivity	-96 dBm
Node Message Rate	1 message every 5 seconds
Node Bandwidth	12000 Kb/s
Antenna Height	15-50 m

In order to generate a network topology that is applicable to an actual utility layout and with information available to perform a power flow analysis, the IEEE 30 bus test case was used as a basis for the network layout. To start, nodes were placed at the seven 132 KV substations identified in the IEEE 30 bus test case (Glen Lyn, Claytor, Hancock, Roanoke, Fieldale, Reusens, and Cloverdale), see Figure 1. Four of the substations are 132/33 KV distribution level substations. The remaining three substations were also considered distribution level for this research because they feed several nearby substations for residential and commercial areas. The test case did not include location information for any of the buses. The substation locations were determined by examining satellite imagery and recording the latitude and longitude of each substation. The remaining buses were placed at locations to allow wireless communications around the local terrain and in residential or industrial areas near or between the 132 KV substations.

With locations specified for 30 wireless nodes, the nodes were plotted on a map of the local terrain using the MATLAB Mapping Toolbox, see Figure 2. The contour information for the plot was obtained from the National Oceanic and Atmospheric Administration (NOAA). The data provided by NOAA is the ETOPO1 model which contains elevation data for the Earth's surface at a one arc-minute resolution [36]. All of the node locations for this project were located in and around Roanoke, Virginia, so the contour data used only includes the area between 36°-38° North and 79°-81° West. The contour line spacing was chosen to be 50 meters for this plot based on the large difference between the minimum and maximum elevations that exist in this area.

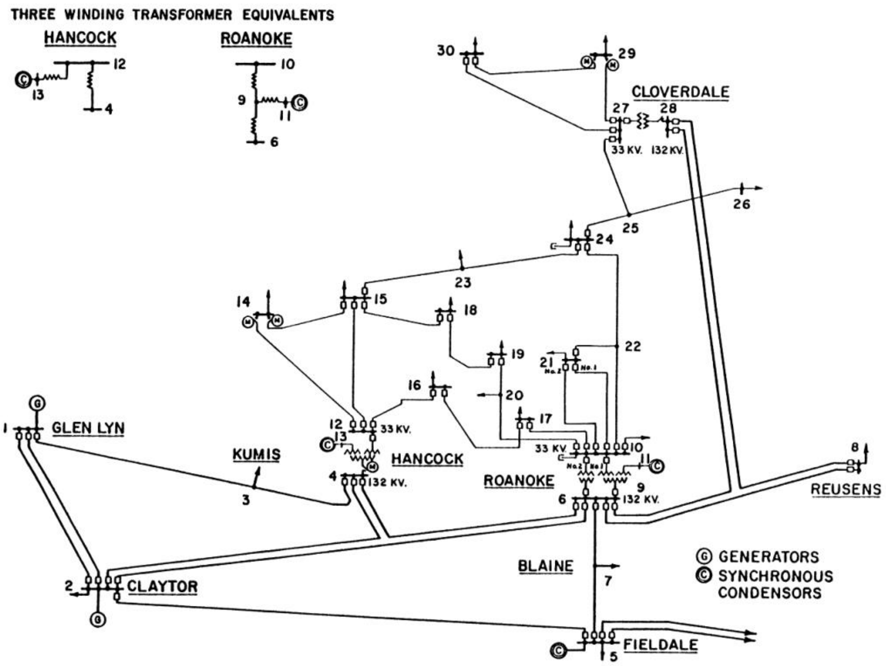

Figure 1 - IEEE 30 Bus Test Case
(Source: http://www.ee.washington.edu/research/pstca/pf30/30bus600.tif)

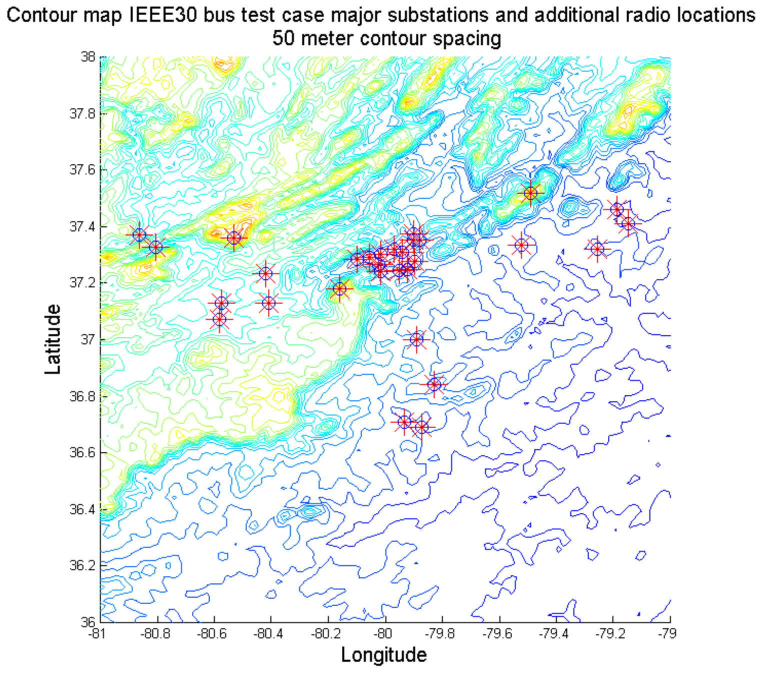

Figure 2 - Node Location Plot

Network Topology

Once the network node locations are specified, the network topology can be generated. The topology includes which nodes can communicate directly with each other as well as the routes that allow nodes to communicate with each other through intermediate nodes.

Before the topology can be generated, the parameters of the network must be initialized. This process is accomplished using the script as shown in Appendix B - WMN_readfile. This script reads an excel file and loads the parameters for the nodes. The parameters loaded are node-specific and include: latitude, longitude, antenna height, bandwidth, message transmission interval, minimum transmit power, receiver sensitivity, and TAP status.

The node latitude and longitude coordinates are used to place each node. The coordinates are used to calculate the distance between the nodes as well as determine the elevation of each node from the topographic data. Each node is assigned an antenna height to help the nodes achieve line of sight over local terrain features. For this research, nodes at the 132 KV substations are given antenna heights of 50 meters and all other nodes are given antenna heights of 15 meters. The substation antenna height was chosen to account for the large metal structures in or around substations that the antennas would need to be mounted above. The antenna height for non-substation nodes was chosen to represent placement on top of a standard utility pole. Each node is also given a bandwidth amount to allow for variations in the types of radios assigned. For this research each node was given a bandwidth of 12,000 Kb/s to match the bandwidth of commercially available 900 MHz radios. A message transmission interval is also assigned for each node. For this research, each node is set to report every 5 seconds to provide a snapshot of the distribution grid. In other scenarios end-user nodes may be assigned a slower reporting rate while substations and grid interconnection points may be assigned a faster reporting rate based on their importance to grid operation. A minimum transmit power for each node was assigned to allow for certain nodes to operate at a higher transmit power and bypass the dynamically assigned transmit power. For example, in this research the minimum transmit power for the substation and repeaters nodes was set to the maximum transmit power. This allows the substation and repeater nodes to establish long-range links within the network. In other cases the nodes would be assigned a low transmit power and simulation would dynamically increase the transmit power required to meet the minimum number of required neighbor nodes. Each node is also assigned a receive sensitivity to allow for different radio types to be installed in the network. For this research, all nodes are assigned a receive sensitivity of -96 dBm, again based on the specifications of commercially available 900 MHz radios. The final node parameter is TAP status. This allows certain nodes to be specified as TAP nodes regardless of the TAP placement algorithms. This allows certain nodes, such as the substation nodes, to be pre-assigned as TAP nodes to minimize the number of TAP nodes assigned algorithmically throughout the network at locations that may not have access to a backhaul network.

With the user defined parameters specified, the scripts designed to model the network can be executed. The model is generated using six steps. The first step is to determine the direct communication links and transmit power for the network nodes. The second step is to generate a multi-hop network map based on the specified node parameters. The third step is to remove communication paths that contain an excluded node. The fourth step is to determine the communication path characteristics for valid paths. The fifth step is to determine the node utilization for the selected communication paths. The sixth step is to determine TAP placement based on node utilization.

Due to the nature of wireless mesh network, the network overhead is a factor of the number of nodes and the number of direct links established in the network. Allowing all nodes to connect to every other node generates excessive network overhead and lowers overall network performance. To increase system performance as described in [35], the transmit power of the nodes is dynamically calculated. The script as shown in Appendix C - WMN_power is designed to incrementally increase the transmit power of each nodes starting from its specified minimum transmit power until it establishes the minimum required number of direct links. If the node does not reach the required number of direct links, the transmission power is set to the maximum allowable transmission power. See Figure 3 for a flowchart description of the WMN_power script.

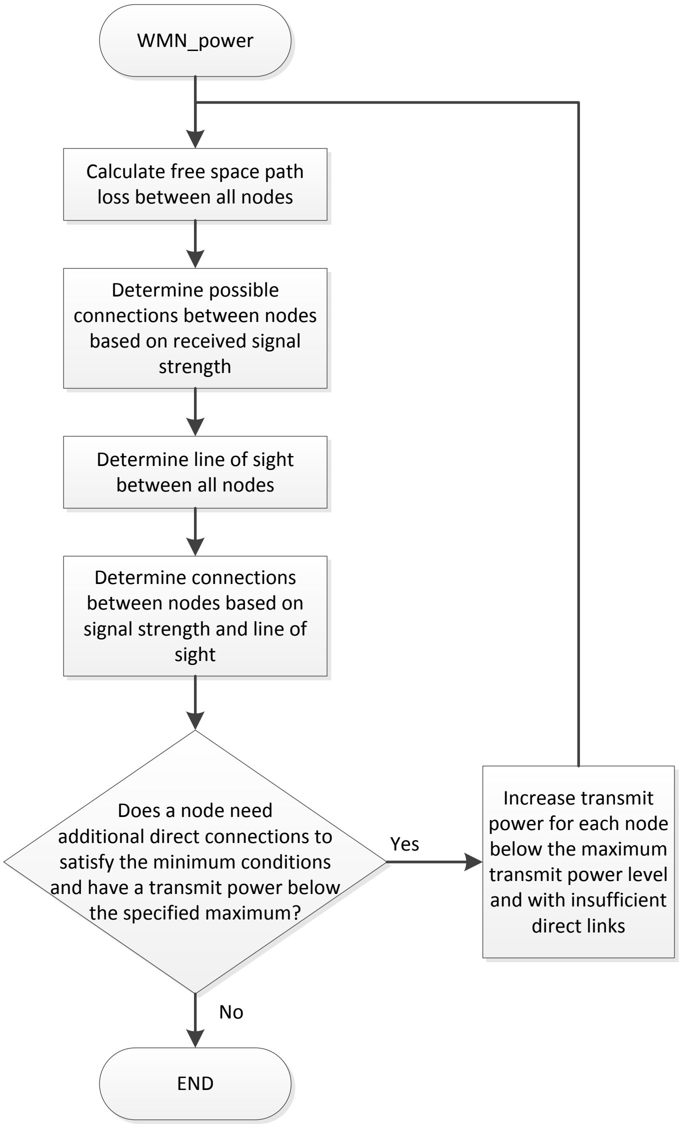

Figure 3 - WMN_power flowchart (Direct communication links and node transmit power)

The first step in generating the network topology is determining the nodes with direct connections. First, the distance between the nodes is calculated and stored. This distance is used to calculate the free space path loss of the signal from signals transmitted for each node to all other nodes in the network. The equation for this calculation is shown in Equation 2-1. The equation assumes that the nodes are using omnidirectional antennas. With the free space path loss calculated, the received signal strength at each node from every other node is calculated as shown in Equation 2-2. Next, the received signal strength is compared to the receive sensitivity to determine if the nodes can communicate directly.

$$\boldsymbol{FSPL = 20\log(4\pi df/c)} \qquad \text{Equation 2-1}$$

Where FSPL is the free space path loss in dB, d is the distance between the nodes in meters, f is the signal frequency in Hertz, and c is the speed of light in meters per second.

$$\boldsymbol{RX = TX - FSPL} \qquad \text{Equation 2-2}$$

Where RX is the received signal strength in dBm, TX is the transmitted signal strength in dBm, and FSPL is the free space path loss in dB.

Comparing the received signal strength and the receive sensitivity establishes a ‘best case’ scenario for the nodes. Nodes satisfying the condition of having a received signal strength greater than the receive sensitivity are within communication range and nodes that do not satisfy the condition are outside of the communication range. However, nodes within range of each other still may not be able to communicate if there are obstacles between them. For this research the only obstacle considered is terrain, because of the data sources used. To account for other obstacles and interference, data for vegetation, buildings, and radio frequency usage would be needed. To be considered as having a direct communications link the nodes must be within range of each other based on received signal strength and must have direct line of sight to each other.

Once the direct communications links are determined and recorded, the multi-hop paths must be mapped. To map the multi-hop paths, the process is as follows. First, a starting node is selected. For each starting node a network map is created. The network map is stored as a matrix. Each column in the matrix represents the nodes in the network that the starting node can communicate with in a given number of hops. Each node appears in the network map only one time, at the minimum number of hops from the starting node, this prevents issues where routes loop back through the same node multiple times. This implementation misses the possibility of routes that take more than the minimum number of hops. However, including all possible routes in the network map creates a significantly larger number of routes for the network. Determining the possible routes in a network is similar to the ‘travelling salesman’ problem meaning that it is computationally intensive. Allowing all possible routes in the network to be considered increases the computation time required considerably. Once the network map is created, the routes for the network are determined based on the node connections between columns of the network map. An example of the paths generated for a simple four node network can be seen in Table 3. For the script described see Appendix D - WMN_routes and Figure 4 for a flowchart description.

Table 3 - Network Mapping Example

Node	**Direct Links**	**Available Paths**
1	2	1 → 2 1 → 2 → 3 1 → 2 → 3 → 4
2	1, 3	2 → 1 2 → 3 2 → 3 → 4
3	2, 4	3 → 2 3 → 2 → 1 3 → 4
4	3	4 → 3 4 → 3 → 2 4 → 3 → 2 → 1

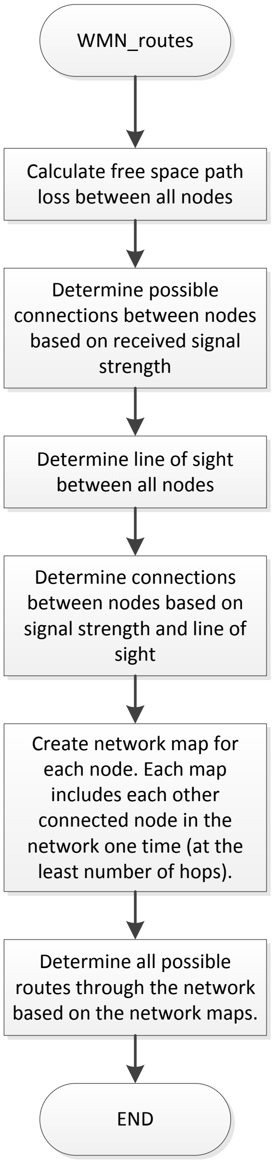

Figure 4 - WMN_routes flowchart (Multi-hop network map)

Once the paths through the network are mapped the next script analyzes the paths and checks for the inclusion of any nodes that are specified to be excluded from the model. Separating this step from determining the available paths is useful to allow for multiple scenarios of a network layout. The most time consuming part of generating the network model is to map the possible paths through the network. Once this part of the process has been done for a given network layout, the network performance can be tested under various scenarios involving node losses. For the script that determines the available paths after removing excluded nodes see Appendix E - WMN_paths and Figure 5 for a flowchart description. For the connections for the test network with no excluded nodes and all nodes transmitting at 36 dBm see Figure 6. As seen in this plot, the network of nodes operating at the maximum transmission power creates many direct node-to-node links. While this is beneficial for ensuring robustness against the loss of nodes, it creates additional wireless interference and congestion due to network overhead required to maintain accurate routing tables which decreases overall network performance.

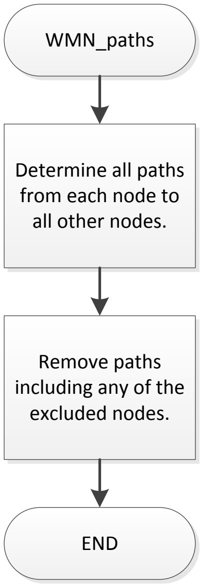

Figure 5 - WMN_paths flowchart (Check for excluded nodes)

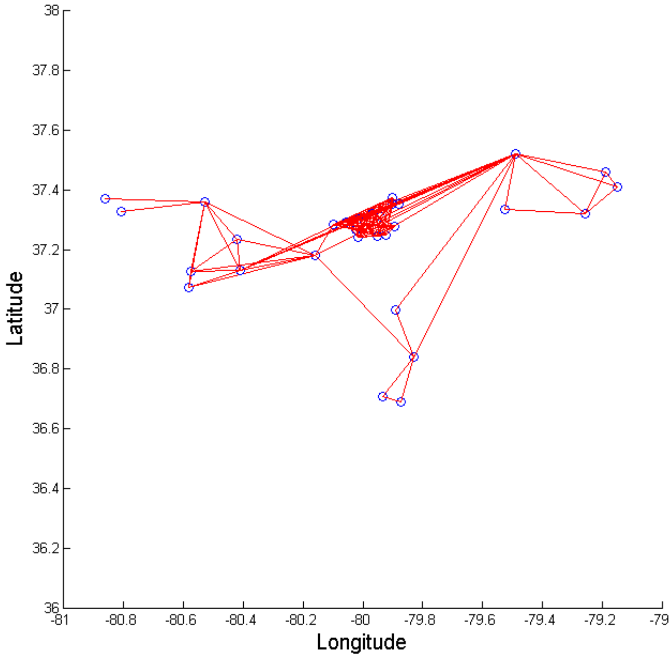

Figure 6 - Node connection plot (Test network with all nodes transmitting at 36 dBm)

Network Characteristics

With the network map for a given layout created, the next step is to evaluate the path characteristics for data travelling through the network for a node to any other node. To determine the network characteristics, each available path from any given non-TAP start node to all TAP nodes is analyzed. This process involves several steps: analyzing network paths, determining utilization, and placing TAP nodes. Due to the automatic assigning of TAP nodes, the process is iterative and consists of three scripts as shown in Appendix F - WMN_pathcharacteristics, Appendix G - WMN_utilization, and Appendix H - WMN_TAPplacement. Once the process completes an iteration, the modified set of TAP nodes is used to begin the next iteration. The process is complete when the maximum node utilization reaches an appropriate level or the percentage of TAP nodes is equal to the predefined maximum allowable percentage of TAP nodes.

The path is recorded with information about each node in the path, the number of hops, path distance, path time, and path bitrate. Only the path to TAP nodes are analyzed because this project assumes the backhaul network connected to the TAP nodes is a high bandwidth, low latency connection back to a control center. For the same reason data originating from a TAP node is not considered because it has a direct connection to the backhaul network. For each non-TAP start node a best case and worst case path are recorded based on the length of time it takes the data to travel from the originating node to any TAP node. To determine the length of time a given path takes Equation 2-3 is used. For this research, the paths were ranked based on time because the path time accounts for the distance traveled, number of hops, and path bitrate. Ranking on other factors, such as bitrate, would tend to focus data on the links with higher bitrate therefore congesting these links. Also, a direct link could be ignored in favor of a multi-hop path with a higher bitrate but lower performance based on the number of hops. For the script that determines the path characteristics see Appendix F - WMN_pathcharacteristics and Figure 7 for a flowchart description.

$$\boldsymbol{PT = \left(\frac{ML}{PB}\right) + (H * FT) + \left(\frac{d}{c}\right)} \qquad \text{Equation 2-3}$$

Where PT is that path time in seconds, ML is the message length in Kb, H is the number of hops in the path, FT is the forwarding time per hop in seconds, d is the length of the current path in meters, and c is the speed of light in meters per second.

$$\boldsymbol{PB = min\left(NB_j\right) - NO} \qquad \text{Equation 2-4}$$

Where PB is the path bitrate in Kb/s, NB is the bandwidth of node j in the path in Kb/s, and NO is the network overhead in Kb/s.

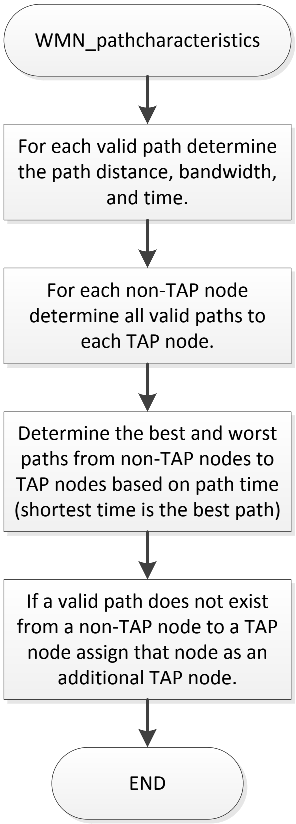

Figure 7 - WMN_pathcharacteristics flowchart (Communication path characteristics)

With the best paths for each starting node identified, the next script as shown in Appendix G - WMN_utilization is used to calculate the utilization percentage of each node in the network. To determine the node utilization percentage, the message length, transmission interval, and node bandwidth are factored in as described in Equation 2-5. For a flowchart description of the WMN_utilization script see Figure 8.

$$\boldsymbol{NU_j = \frac{\sum_{i=1}^{N}\left(\frac{ML}{T}\right)}{NB_j} * 100} \qquad \text{Equation 2-5}$$

Where NU is the node utilization percentage, N is the number of paths that include node j, and T is the message transmission interval.

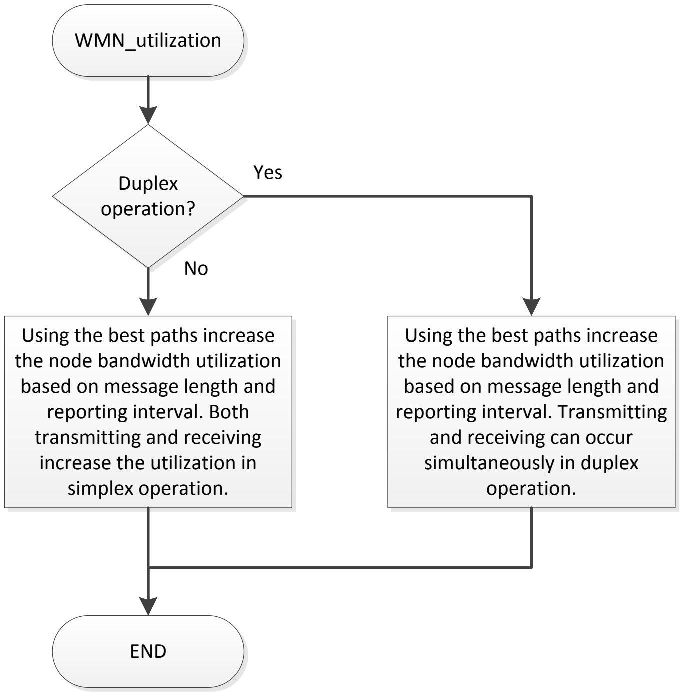

Figure 8 - WMN_utilization flowchart (Node utilization)

Once the node utilization percentages are determined the next script as shown in Appendix H - WMN_TAPplacement is used to determine if more TAP nodes are needed to ensure delivery of all messages. To perform this operation the highest utilization percentage of all nodes is examined. If the maximum utilization is equal to or less than 100%, no additional TAP nodes are required. If the maximum utilization is greater than 100%, the node with the highest utilization is reassigned as a TAP node. A utilization rate of 100% is the maximum allowable rate because over utilizing a node may result in data loss because the node does not have the bandwidth available to send or receive signals. The next iteration of the process then begins with the modified set of TAP nodes. For a flowchart description for the WMN_TAPplacement script see Figure 9.

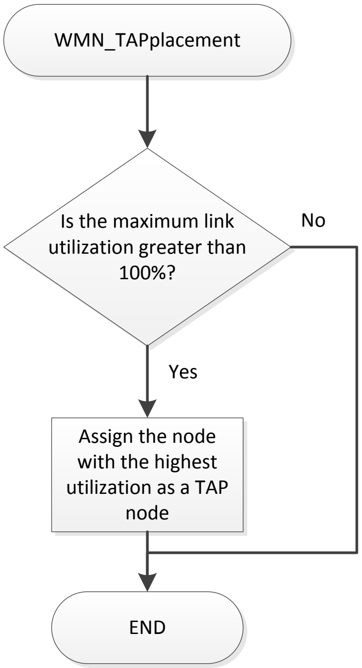

Figure 9 - WMN_TAPplacement flowchart (TAP placement)

Network Model

A network model was generated using the node parameters as shown in Appendix I - Nodes.xls and the MATLAB scripts as shown in Appendix A - WMN_script through Appendix H - WMN_TAPplacement. The resulting node connection plot is shown in Figure 10. The notable network characteristics are shown in Table 4. As seen in the table, dynamically assigning the transmission power to assure nodes had the specified minimum number of direct links resulting in significantly fewer direct links. The same network with all nodes operating at full transmission power had 224 direct links where only 84 direct links are established with the dynamically assigned transmission power. The specified parameters and result of the network model indicate that this network layout based on the desired reporting frequency and message length would be sufficient to generate an updated power grid snapshot in less than two seconds based on the longest communication path time of 1.169 seconds. This updated power grid data can then be used for various power monitoring and control algorithms to ensure efficient operation of the power grid generators, controllable loads, and switching equipment.

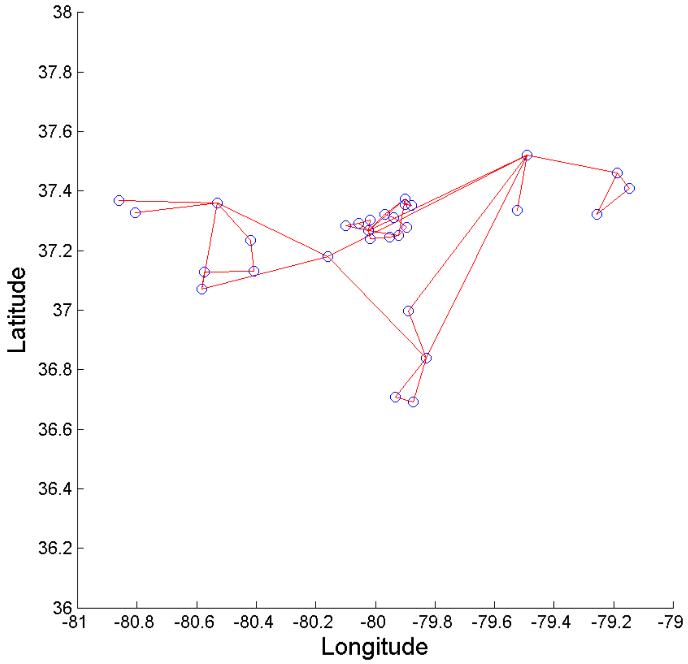

Figure 10 - Node connection plot for network model

Table 4 – Network Model Characteristics

Parameter	**Value**
Number of direct links	84
Maximum node utilization	13.38%
Number of paths	905
Fastest best case path	0.768 seconds
Slowest best case path	0.869 seconds
Fastest worst case path	0.969 seconds
Slowest worst case path	1.169 seconds
TAP nodes	1, 2, 3, 4, 5, 6, 7
Minimum transmit power	6 dBm
Maximum transmit power	36 dBm

In Table 4, the number of direct links are the number of direct node-to-node links in the network. This is a key characteristic for a wireless mesh network because the overall network overhead in the network is affected by the number of direct links. The maximum node utilization is the node or nodes with the highest utilization rate. This is an important factor to consider in the network layout because it provides an indication of the network’s ability to handle an increase in traffic. The number of paths is the number of unique routes from any node to all remaining connected nodes throughout the network. This provides an indication of how many interconnections and variations are possible throughout the network. An increase in the number of paths corresponds to an increase in computation time required for the simulation. The times for the best cases paths are based on the most efficient paths through the network. These paths consider the overall path bandwidth, distance travelled, and number of hops to determine the best paths. The worst case paths use the same parameters to determine the least efficient paths. The TAP nodes gives an indication of how many nodes in the network have been assigned as TAP nodes. In this scenario only the substation nodes that were previously assigned as TAP nodes have been used as TAP nodes. The minimum and maximum transmit powers provide an indication of the dynamically assigned transmit powers. In this scenario, the minimum transmit power is the transmit power for all non-substation nodes that was initially assigned. This means that one or more nodes was able to establish the minimum required number of direct links without increasing the transmit power. The maximum transmit power is the transmit power assigned to the substation nodes.

Smart Grid Communications

The key parameters of the wireless mesh network model simulation provide insight in the feasibility of using such a network for a specific application. For example, if the specified node bandwidth (12,000 Kb/s) and reporting interval (5 seconds) and the resulting worst case network path time (1.169 seconds) fulfill the application requirements, the wireless mesh network as defined and simulated is capable of supporting the monitoring and/or controlling the associated power grid. For other applications, the required network parameters may have higher or lower tolerances. In [37] various smart grid communications networks applications are proposed with different requirements. For a basic current and voltage waveform monitoring device the node bandwidth would only need to be 12 Kb/s to transmit the waveform information. A network as simulated in this research would provide one thousand times the bandwidth required for this type of application. A guideline bandwidth of 2-5 Mb/s is also proposed for smart grid communication networks to support not only waveform information but additional calculated information. The network as simulated in this research also exceeds this proposed guideline, indicating that it would be suitable for a smart grid application.

Another requirement outlined is the communications network latency. Two of the major factors in determining latency requirements are fault detection and DG. Fault detection latency requirements are typically 100 milliseconds or less. The simulated network would not be suitable for fault detection applications. For DG, the latency requirements are typically a factor of the utility governing the interconnection, ranging from two seconds to several minutes. The latency requirement is a function of how quickly the utility requires the DG to be aware of potential islanding and respond accordingly. For California Independent System Operator (CAISO) requirements for a network with DG include performing economic dispatch calculations every five minutes with a communications delay of 2-10 seconds [38]. The simulated network has a latency of 1.169 seconds, making it suitable for a DG communications network. As these few examples have a wide range of network requirements, it can observed that the network requirements will be largely dependent on the application for the network communications.

As discussed in [38], a smart grid communications network also enables customer participation. This customer participation is enabled by the ability to provide customers with real-time pricing data. Customers can use the pricing data to intelligently schedule energy usage. Energy intensive processes and appliances can be scheduled to operate during off-peak times when excess generation capacity is abundant. Customers can also participate with DER which is enabled through the communications network.

Chapter 3

Power Flow

Overview

A common example of a power system monitoring and control algorithm that is performed continuously on an active power gird is power flow. The results of the power flow calculations are used for a wide variety of control decisions within the power grid. These decisions include generator dispatching for capacity and economics, load shedding, and to control switching equipment to maximize efficiency and ensure stability. Because power flow is so crucial to many key elements of maintaining power system efficiency and stability it is examined in this project as an algorithm that would benefit from having a near-real time snapshot of the power grid.

Power flow was also chosen for this research because it is a computationally intensive step in the decision making process for power grid monitoring and control. Two power flow algorithm categories widely used in the power industry are the Newton method and fast decoupled methods. While many techniques can be applied to solving power flow problems each has its own set of tradeoffs. Some algorithms can quickly reach a solution but sacrifice accuracy or the ability to solve specific scenarios that may exist in a power grid. Newton based methods are more computationally intensive but have local quadratic convergence. Fast decoupled methods require less computation due to forward elimination and backward substitution but offers weaker convergence [39]. Both methods use an iterative process that involves matrix inversion for each iteration. As the number of busses in a system grow, the matrix inversion at each iteration becomes more computationally intensive. The fast decoupled methods improves overall computation time by separated the matrix into several small matrices. Another way to reduce computation time for power flow is to reuse the matrix inversion for several iterations before recalculating. Due to the fact that power flow is a crucial part of power gird monitoring and control, methods to reduce the computational requirements and improve accuracy are continually being pursued. Samples of such research can be seen in [39]-[43]. These research projects span from 1967-2013, indicating that power flow calculations are an area of continual improvement.

Power Flow Model

The test case used for the power flow calculations is the same IEEE 30 bus test case used to generate the network model, see Figure 1. As this research is focused on modeling a communications network the power flow solver used was an existing third-party MATLAB package called MATPOWER [39]. The MATPOWER package includes the IEEE 30 bus test case as an example power system. The solution for the IEEE 30 bus test case from MATPOWER was verified against the solution provided with the IEEE 30 bus test case using the information in [45]. All of the information for the power system, including generators, loads, buses, and branches, is loaded via a MATPOWER command. With MATPOWER various power flow algorithms can be performed including AC power flow, AC continuous power flow, AC optimal power flow, DC power flow, DC optimal power flow, and AC optimal power flow with fixed reserve requirements. For this project, the ‘runpf’ command was used to perform a simple power flow calculation. To see variations in the system the IEEE 30 bus test case was run under a variety of scenarios. The scenarios included disabling each branch one at a time, disabling each bus (and any associated generators) one at a time, disabling a cascaded series of branches, and the default test case scenario (the IEEE 30 bus test case). For a single line diagram of the IEEE 30 bus test case with the branch identifiers added see Figure 11.

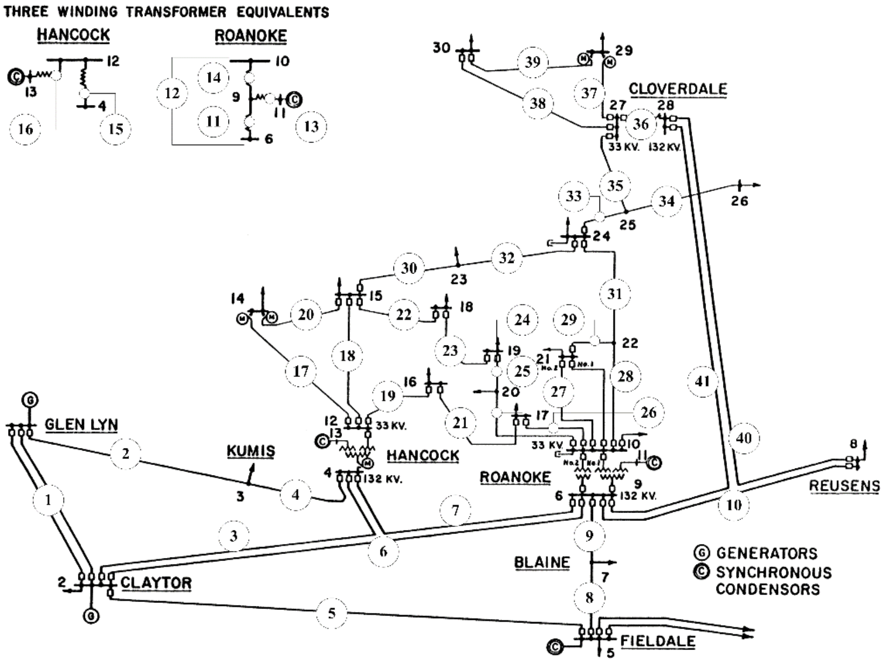

Figure 11 - IEEE 30 Bus Test Case with Branch Identifiers

Results

A summary of the results from the power flow model can be seen in Table 5. The full set of results can be seen in Table 6. The key results include the fact that maximum power loss in the system occurred when the branch utilization was also at its maximum. This result is not unexpected as increased branch utilization indicates increased branch current and the branch power loss is related to the square of the current. Likewise, the test case with the lowest branch utilization also has the lowest branch losses. While this relationship will not always be true, it does provide some insight as to how sudden changes in a power grid can significantly affect the efficiency and stability of the grid.

Table 5 - Power Flow Results Summary

Parameter	Value	Test Case
Branch Losses (Max)	5.47 MW	74
Branch Losses (Min)	1.55 MW	49
Generator Utilization (Max)	75.16%	42
Generator Utilization (Min)	47.99%	49
Maximum Branch Utilization (Max)	114.73%	74
Minimum Branch Utilization (Min)	61.97%	49
Loss Percentage (Max)	2.81%	74
Loss Percentage (Min)	0.96%	49
Computation Times (successful)	0.8 – 2.73 sec	various

In Table 6 the results are sorted in to rows based on the test case number. Each test case number corresponds to a set of circumstances that differentiates one test case from another. The parameters that can vary from case to case are the branches out of service, buses out of service, and generators out of service. For the columns corresponding to these parameters the entry in the table will be “N/A” for no branches, buses, or generators out of service or a number(s) corresponding to the branch and bus numbers as identified in Figure 11. The remaining columns are results from the power flow. Column A is the actual generation of the power system. The actual generation is a combination of the load and losses in the network. Column B is the generation capacity. This is the sum of the generation capacity of the generators in service. Column C is the generator usage percentage, the actual generation as a percentage of the generation capacity. Column D is the total load attached to all in-service buses in the power network. Column E is the total losses in the power network. This accounts for the transmission losses throughout the network. Column F is the total loss percentage, the losses as a percentage of the actual generation. Column G is the maximum line utilization. This value gives the usage percentage of the branch closest to or farthest exceeding its capacity. Column H is the maximum utilization branch. This column identifies the branch number associated with the maximum line utilization percentage. Columns I and J are the minimum line utilization percentage and branch. These columns are the same as the maximum utilization percentage and branch except identifying the branch farthest from its rated capacity. Columns K, L, and M are the varying parameters as described before, branches, busses, and generators out of service. Column N identifies if the power flow completed successfully. This column is important in determining if the completion time can be considered. If a power flow does not complete the computation time will increase as the power flow algorithm iterates until it reaches the iteration limit as the results do not converge. The power flow will fail to complete if certain parameters cause part of the network to be islanded. Column O is the amount of time that the power flow calculation took to complete.

Table 6 - Power Flow Results

Col.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Test Case	Gen Actual (MW)	Gen Cap (MW)	Gen Usage (%)	Total Load (MW)	Total Losses (MW)	Total Loss Percent	Max Line Util. (%)	Max Util. Branch	Min Line Util. (%)	Min Util. Branch	Branches out of service	Buses out of service	Gen out of service	Success	Time (sec)
1	191.73	335	57.23	189.2	2.528	1.32%	77.53	10	0.00	13	1	N/A	N/A	YES	2.73
2	192.03	335	57.32	189.2	2.831	1.47%	77.65	10	1.85	4	2	N/A	N/A	YES	1.514
3	191.88	335	57.28	189.2	2.676	1.39%	77.68	10	0.00	13	3	N/A	N/A	YES	1.233
4	191.93	335	57.29	189.2	2.728	1.42%	77.64	10	1.85	2	4	N/A	N/A	YES	1.747
5	192.04	335	57.33	189.2	2.840	1.48%	77.57	10	0.00	13	5	N/A	N/A	YES	1.31
6	192.06	335	57.33	189.2	2.864	1.49%	77.56	10	1.27	15	6	N/A	N/A	YES	1.342
7	192.18	335	57.37	189.2	2.978	1.55%	77.04	10	1.41	20	7	N/A	N/A	YES	1.201
8	192.06	335	57.33	189.2	2.861	1.49%	77.58	10	0.00	5	8	N/A	N/A	YES	1.201
9	191.84	335	57.27	189.2	2.644	1.38%	77.57	10	2.34	41	9	N/A	N/A	YES	1.17
10	193.14	335	57.65	189.2	3.941	2.04%	98.16	40	0.00	13	10	N/A	N/A	YES	1.092
11	191.78	335	57.25	189.2	2.577	1.34%	78.15	10	0.00	14	11	N/A	N/A	YES	1.357
12	191.70	335	57.22	189.2	2.496	1.30%	77.82	10	0.00	13	12	N/A	N/A	YES	1.201
13	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	13	N/A	N/A	NO	5.133
14	191.78	335	57.25	189.2	2.577	1.34%	78.15	10	0.00	11	14	N/A	N/A	YES	1.264
15	191.68	335	57.22	189.2	2.477	1.29%	77.39	10	3.14	41	15	N/A	N/A	YES	1.373
16	230.14	335	68.70	189.2	3.935	1.71%	83.22	30	3.39	31	16	N/A	N/A	NO	5.429
17	191.80	335	57.25	189.2	2.599	1.36%	77.69	10	1.94	41	17	N/A	N/A	YES	1.107
18	191.82	335	57.26	189.2	2.619	1.37%	78.08	10	0.00	13	18	N/A	N/A	YES	1.123
19	191.86	335	57.27	189.2	2.659	1.39%	77.73	10	0.00	13	19	N/A	N/A	YES	1.108
20	191.64	335	57.21	189.2	2.443	1.27%	77.55	10	2.46	15	20	N/A	N/A	YES	1.092
21	191.70	335	57.22	189.2	2.498	1.30%	77.66	10	2.04	41	21	N/A	N/A	YES	1.107
22	191.96	335	57.30	189.2	2.764	1.44%	77.52	10	0.93	26	22	N/A	N/A	YES	1.108
23	191.75	335	57.24	189.2	2.552	1.33%	77.53	10	2.58	41	23	N/A	N/A	YES	1.076
24	191.74	335	57.24	189.2	2.542	1.33%	81.20	22	0.66	15	24	N/A	N/A	YES	1.045

Col.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Test Case	Gen Actual (MW)	Gen Cap (MW)	Gen Usage (%)	Total Load (MW)	Total Losses (MW)	Total Loss Percent	Max Line Util. (%)	Max Util. Branch	Min Line Util. (%)	Min Util. Branch	Branches out of service	Buses out of service	Gen out of service	Success	Time (sec)
25	191.88	335	57.28	189.2	2.679	1.40%	95.76	22	0.44	20	25	N/A	N/A	YES	1.092
26	191.74	335	57.24	189.2	2.539	1.32%	77.52	10	0.38	15	26	N/A	N/A	YES	1.061
27	191.70	335	57.22	189.2	2.502	1.31%	77.61	10	0.00	13	27	N/A	N/A	YES	1.077
28	191.69	335	57.22	189.2	2.495	1.30%	77.55	10	2.33	15	28	N/A	N/A	YES	1.076
29	192.26	335	57.39	189.2	3.058	1.59%	77.07	10	1.71	15	29	N/A	N/A	YES	0.795
30	192.05	335	57.33	189.2	2.846	1.48%	100.00	32	2.81	15	30	N/A	N/A	YES	1.076
31	191.61	335	57.20	189.2	2.410	1.26%	77.07	10	3.01	27	31	N/A	N/A	YES	1.077
32	191.92	335	57.29	189.2	2.723	1.42%	100.00	30	1.82	41	32	N/A	N/A	YES	1.077
33	191.55	335	57.18	189.2	2.354	1.23%	75.10	10	0.00	13	33	N/A	N/A	YES	1.123
34	188.01	335	56.12	189.2	2.313	1.23%	76.44	10	3.70	15	34	N/A	N/A	NO	5.257
35	191.65	335	57.21	189.2	2.447	1.28%	72.87	10	1.56	15	35	N/A	N/A	YES	1.123
36	192.18	335	57.37	189.2	2.985	1.55%	85.09	35	5.39	11	36	N/A	N/A	YES	1.17
37	192.02	335	57.32	189.2	2.822	1.47%	85.43	38	1.77	41	37	N/A	N/A	YES	1.076
38	192.15	335	57.36	189.2	2.946	1.53%	86.20	37	1.56	41	38	N/A	N/A	YES	1.092
39	191.77	335	57.25	189.2	2.575	1.34%	77.63	10	2.19	41	39	N/A	N/A	YES	1.232
40	191.73	335	57.23	189.2	2.528	1.32%	94.37	10	2.67	15	40	N/A	N/A	YES	1.108
41	191.67	335	57.21	189.2	2.467	1.29%	75.38	10	2.61	15	41	N/A	N/A	YES	1.092
42	191.66	255	75.16	189.2	2.464	1.29%	77.59	10	0.00	13	N/A	1	1	YES	0.873
43	170.38	255	66.82	167.5	2.883	1.69%	77.61	10	2.27	15	N/A	2	2	YES	1.092
44	189.19	335	56.48	186.8	2.393	1.26%	77.56	10	0.00	13	N/A	3	N/A	YES	1.092
45	183.88	335	54.89	181.6	2.281	1.24%	77.54	10	2.37	15	N/A	4	N/A	YES	1.092
46	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	5	N/A	YES	1.108
47	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	6	N/A	YES	1.108
48	168.24	335	50.22	166.4	1.835	1.09%	77.58	10	0.00	13	N/A	7	N/A	YES	1.076
49	160.75	335	47.99	159.2	1.550	0.96%	61.97	29	1.90	4	N/A	8	N/A	YES	1.061
50	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	9	N/A	YES	1.107

Col.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Test Case	Gen Actual (MW)	Gen Cap (MW)	Gen Usage (%)	Total Load (MW)	Total Losses (MW)	Total Loss Percent	Max Line Util. (%)	Max Util. Branch	Min Line Util. (%)	Min Util. Branch	Branches out of service	Buses out of service	Gen out of service	Success	Time (sec)
51	185.61	335	55.40	183.4	2.205	1.19%	77.24	10	3.72	41	N/A	10	N/A	YES	1.108
52	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	11	N/A	YES	1.108
53	180.28	335	53.82	178	2.281	1.27%	77.04	10	0.00	13	N/A	12	N/A	YES	1.092
54	193.14	295	65.47	189.2	3.935	2.04%	83.22	30	0.00	16	N/A	13	13	YES	1.123
55	185.29	335	55.31	183	2.293	1.24%	77.24	10	3.72	41	N/A	14	N/A	YES	1.092
56	183.25	335	54.70	181	2.245	1.23%	77.07	10	4.26	1	N/A	15	N/A	YES	1.092
57	188.02	335	56.12	185.7	2.318	1.23%	77.39	10	3.12	41	N/A	16	N/A	YES	1.092
58	182.25	335	54.40	180.2	2.049	1.12%	77.05	10	3.69	1	N/A	17	N/A	YES	1.248
59	188.28	335	56.20	186	2.285	1.21%	77.37	10	0.00	13	N/A	18	N/A	YES	1.092
60	181.67	335	54.23	179.7	1.971	1.08%	77.00	10	0.65	25	N/A	19	N/A	YES	1.076
61	189.32	335	56.51	187	2.320	1.23%	77.44	10	2.93	41	N/A	20	N/A	YES	1.108
62	173.60	335	51.82	171.7	1.905	1.10%	76.38	10	0.00	13	N/A	21	N/A	YES	1.06
63	193.26	285	67.81	189.2	4.060	2.10%	80.16	10	0.34	20	N/A	22	22	YES	0.983
64	189.03	305	61.98	186	3.026	1.60%	79.10	10	1.15	32	N/A	23	23	YES	1.092
65	182.70	335	54.54	180.5	2.203	1.21%	76.37	10	3.92	1	N/A	24	N/A	YES	1.076
66	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	25	N/A	YES	1.108
67	188.01	335	56.12	185.7	2.313	1.23%	76.44	10	0.00	34	N/A	26	N/A	YES	1.092
68	192.98	280	68.92	189.2	3.784	1.96%	88.05	10	1.11	20	N/A	27	27	YES	1.108
69	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	28	N/A	YES	1.108
70	189.15	335	56.46	186.8	2.351	1.24%	76.51	10	3.10	15	N/A	29	N/A	YES	1.107
71	180.72	335	53.95	178.6	2.118	1.17%	72.88	10	2.86	1	N/A	30	N/A	YES	1.092
72	191.80	335	57.25	189.2	2.599	1.36%	77.69	10	1.94	41	17	N/A	N/A	YES	1.108
73	192.29	335	57.40	189.2	3.093	1.61%	102.70	30	2.88	27	[17;18]	N/A	N/A	YES	1.185
74	194.67	335	58.11	189.2	5.468	2.81%	114.73	23	2.95	41	[17;18;30]	N/A	N/A	YES	1.155
75	191.64	335	57.21	189.2	2.444	1.28%	77.57	10	2.41	41	N/A	N/A	N/A	YES	0

The efficiency changes can be observed by the increase in branch power loss and the increase in loss percentage alongside the increase in branch utilization. The stability effects can be observed in the progression from test case 72 to test case 74. In test case 72 branch 17 is taken out of service without much effect on the system. The total branch losses increase by 0.16 MW and the maximum branch utilization increases by 0.12%. In test case 73 branches 17 and 18 are taken out of service. In this case the effect on the system is much greater. The total branch losses increase by 0.65 MW and the maximum branch utilization increases to 102.70% from 77.57% under normal conditions. In test case 74 branches 17, 18, and 30 are taken out of service. This reflects a possible outcome of a prolonged outage of branches 17 and 18 with branch 30 operating at 102.70% capacity. Again the overall system performance is degraded. The branch losses increase to 5.47 MW and the maximum branch utilization increases to 114.73% for branch 23. If the systems were under these conditions for an extended period of time branch 23 may also be taken out of service by a protection device, leading to a cascade of outages and loss of system stability.

An instance of cascading outages severely impacting power grid stability and reliability is the August 2003 blackout in the Northeast United States and parts of Canada. The causes of the outage are described in [46]. In [47], the author demonstrates why a smart grid implementation could have avoided the cascading outages and rolling blackouts that disrupted the flow of approximately 61,800 MW of power to 50 million people across seven states and one province. The final report on the blackout identified four causes for the outage: failure to understand the weakness in the system with regards to voltage instability, failure to recognize or understand the condition of the system as it progressively worsened, failure to properly manage tree growth around transmission lines, and failure to provide effective diagnostic support [46].

The report also indicates that all of the protective devices acted appropriately based on the information they were receiving from the power grid but the actions negatively impacted the grid. For example, frequency, voltage, and current variations seen by protective devices can indicate fault conditions or severe overloading conditions. Without a knowledge of the status of the remainder of the grid, the protective devices interpreted these conditions are the result of faults and took generators and transmission lines offline. This resulted in increased demands on the remaining lines and generators, continuing the cascade of failure throughout the system.

Having a current status of the major busses in the power grid as provided by a communications network can help in cases this the 2003 blackout in several ways. First, the power flow is based on measurements instead of state estimators. In the 2003 blackout the power flow was not being solved correctly because the state estimator was not aware of the current status of the power grid due to errors. Another way is information sharing for protective devices. With more data about the current grid status, intelligent protective devices can make better decisions to ensure that the actions are in the best interest of maintaining overall grid stability.

Chapter 4

Conclusion and Results

Combining Communications and Power Networks

The results of the network model and power flow model indicate that for this small power system the limitation for how fast an accurate snapshot of the power grid may be generated is based on the speed of the power flow calculations. For this scenario the longest successful power flow calculation was 2.73 seconds whereas the worst case for the communications network was 1.169 seconds. In this scenario the data was being retrieved from the power grid every 5 seconds, this means that every 5 seconds the data can be collected and the power flow calculations performed. For a strictly monitoring application the speed of the network and power flow calculations are sufficient to allow for a 5 second update. For a monitoring and control application that speed of the communications network would factor in twice as the data is polled form the power grid, the power flow is performed, and the commands are sent back out to the network. For this scenario the time for all of that to take place is 5.068 seconds. For this type of application the system should be configured for a slower update, perhaps a 10 second update to allow sufficient time for the worst case networks and worst case power flow calculations to complete. The calculation times for the power flow are also based on the computation time of the equations as performed on a consumer grade laptop. In a utility implementation the power flow computation speeds would be increased based on the availability of high speed servers and optimized software.

The limiting factor most likely to change is the speed of the power flow calculations as the number of buses increases. This trade-off is important to consider in a practical application of these models. For a larger system it would be more appropriate to determine the computation time for the power flow and design the network around that. For example, if the power flow takes one minute, the network can be designed to report data once per minute. With that reporting rate, the appropriate number of TAP nodes can be deployed to minimize costs. If the desired process or calculation can only be performed at a specified rate, collecting data more often will result in greater network costs with little or no benefit.

While using the WMN for power flow is important to consider because of its near real-time performance requirements for use with generation and load management functionality, many other smart grid technologies would also benefit from a pervasive, expandable communications network. One of these technologies is AMI. AMI technology fulfills one of the six smart grid characteristics as defined in the DOE’s 2010 Smart Grid System Report [1], enabling informed participation by customers. AMI allows customers to have access to real-time pricing to make decisions about when to generate or use power based on current costs. WMN are an effective communications technology for AMI because radios can be integrated with meters and retrofitted to existing installations, eliminating the need to establish a new hard-wired communications infrastructure. Another smart grid characteristic that a WMN installation would address is accommodating all generation and storage options. To allow for DER real-time monitoring is required to ensure stability. As generation moves from centralized generation to distributed generation, the predictability of the generation is decreased. Without generators following a pre-determined schedule for power generation and operating intermittently in the case of some renewable energy resources, a way to monitor and control current supply and demand of the grid is essential to maintain stability. As discussed previously, utilities may specify requirements for DG to respond to potential islanding scenarios in a limited time frame achievable only through the use of consistent, high speed communications networks.

Implications

The major implication of this research is allowing a power utility to analyze the costs and benefits of a communications network before installation or costly pilot projects. By knowing how many nodes are required and how many of them are TAP nodes with associated back-haul networks, the utility can estimate an installation and maintenance cost to compare to the cost savings of having an accurate, near-real time snapshot of their power distribution grid. Additionally, utilities can use the network model to determine the effect of utilizing the network for AMI, DER, and real-time monitoring and control simultaneously.

While much research has been done on the need to incorporate a communications network alongside the power grid, the research falls short on providing a method to analyze the interaction between the power grid and the communications network. Attempting to incorporate communications in to the power grid without properly planning the network and determining the characteristics can result in unusable or uneconomical solutions. An unusable solution would be a communications network so slow that determining a near real-time status of the grid would be impossible and state estimators would still need to be relied on to determine the status of sections of the grid. An uneconomical solution would be a network that over performs for the needs of the power monitoring a control. A communications network that can support real-time monitoring of the grid with minimal delay would be excessive if the control algorithms cannot respond to change faster than once a minute.

Chapter 5

Future Work

While this research provides a framework to model a wireless mesh network installation based on the needs of the power monitoring and control algorithms there are additional steps that should be taken before the models are functional for a practical application. One of these improvements would be incorporating the wireless mesh network routing algorithm in to the simulation. This would improve the accuracy of the paths taken for data throughout the network. Another improvement would be to make the network model a real-time model. Because the most computationally intensive aspect of generating the mode is determining the available routes for a physical network layout, that part of the model could be generated off-line and used to model the data flow in a real-time scenario. This issue can also be overcome by incorporating the routing algorithm in to the model to avoid the necessity of pre-calculating possible routes. The model could also be improved by using location and bus data from an operating power grid to better determine the reporting rates needed for the radios based on real scenarios. Future work could also include using the nodes to perform distributed power flow calculations [48] and using market based solutions to solve for optimal power flow [49].

Another interesting test case would be to simulate a neighbor with a node at each home or business all reporting back to a data aggregator at a substation. A similar set up could be used to monitor a micro grid and control generation and distribution within the micro grid for grid connected and islanded operation. For either of these scenarios, more accurate terrain information would be required to determine elevation differences between nearby nodes.

To test the feasibility of using the communications simulation for a larger scale network a test network was created in the Tulsa, Oklahoma area. This area was chosen because it is a population center with very little elevation change which simplifies wireless communication by removing terrain as an obstacle for communication. For this network the nodes were spaced evenly in a grid, 36 nodes by 36 nodes, covering an area 0.2° in both latitude and longitude (36.0-36.2° N and 95.83-96.03° W). The goal of this simulation was to determine if results could be obtained for a larger network in a reasonable amount of time. On the same personal computing hardware and using similar parameters as the 30 bus test case network the 1296 node network simulation completed in approximately 2 hours and 14 minutes, compared to approximately 6 minutes for the 30 bus test case network.

This research provides a foundation to demonstrate the need to generate a network model to determine the response characteristics prior to deployment. Without establishing the network characteristics, remote control and monitoring functionality cannot be designed appropriately to accommodate the network. Using control systems on a network that performs worse than expected may negatively affect stability and reliability. Operating a control system on a network that performs better than expected may cause efficiency losses and unnecessary expense in control design. As the power grid transforms to a smart grid with enhanced communications and control, the design and expansion of the communications networks will pose a challenge to utility companies traditionally focused solely on power distribution networks. Failing to properly integrate the communication networks and power distribution simulations may result in failed or delayed implementations of early smart grid technologies. Based on these factors, the need for tools to combine communications networks and power distribution simulations will take on a larger role in the future of the smart grid and advance in capabilities and technology in step with the advancement of the smart grid.

References

[1] “2010 Smart Grid System Report,” United States Department of Energy, Washington DC, February 2012.

[2] Ye Yan; Yi Qian; Sharif, H.; Tipper, D., "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges," *Communications Surveys & Tutorials, IEEE* , vol.15, no.1, pp.5,20, First Quarter 2013

[3] Monti, A.; Ponci, F., "Power Grids of the Future: Why Smart Means Complex," *Complexity in Engineering, 2010. COMPENG '10.* , vol., no., pp.7,11, 22-24 Feb. 2010

[4] Chun-Hao Lo; Ansari, N., "The Progressive Smart Grid System from Both Power and Communications Aspects," *Communications Surveys & Tutorials, IEEE* , vol.14, no.3, pp.799,821, Third Quarter 2012

[5] Erol-Kantarci, M.; Mouftah, H.T., "Smart grid forensic science: applications, challenges, and open issues," *Communications Magazine, IEEE* , vol.51, no.1, pp.68,74, January 2013

[6] Farhangi, H., "The path of the smart grid," *Power and Energy Magazine, IEEE* , vol.8, no.1, pp.18,28, January-February 2010

[7] Galli, S.; Scaglione, A; Zhifang Wang, "For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid,"*Proceedings of the IEEE* , vol.99, no.6, pp.998,1027, June 2011

[8] Gungor, V.C.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G.P., "Smart Grid Technologies: Communication Technologies and Standards," *Industrial Informatics, IEEE Transactions on* , vol.7, no.4, pp.529,539, Nov. 2011

[9] Ipakchi, A; Albuyeh, F., "Grid of the future," *Power and Energy Magazine, IEEE* , vol.7, no.2, pp.52,62, March-April 2009

[10] Manz, D.; Walling, R.; Miller, N.; LaRose, B.; D'Aquila, R.; Daryanian, B., "The Grid of the Future: Ten Trends That Will Shape the Grid Over the Next Decade," *Power and Energy Magazine, IEEE* , vol.12, no.3, pp.26,36, May-June 2014

[11] Moslehi, K.; Kumar, R., "A Reliability Perspective of the Smart Grid," *Smart Grid, IEEE Transactions on* , vol.1, no.1, pp.57,64, June 2010

[12] Niyato, D.; Qiumin Dong; Ping Wang; Hossain, E., "Optimizations of Power Consumption and Supply in the Smart Grid: Analysis of the Impact of Data Communication Reliability," *Smart Grid, IEEE Transactions on* , vol.4, no.1, pp.21,35, March 2013

[13] Xiang Lu; Wenye Wang; Jianfeng Ma, "An Empirical Study of Communication Infrastructures Towards the Smart Grid: Design, Implementation, and Evaluation," *Smart Grid, IEEE Transactions on* , vol.4, no.1, pp.170,183, March 2013

[14] Yi Xu; Wenye Wang, "Wireless Mesh Network in Smart Grid: Modeling and Analysis for Time Critical Communications," *Wireless Communications, IEEE Transactions on* , vol.12, no.7, pp.3360,3371, July 2013

[15] Braunstein, B.; Trimble, T.; Mishra, R.; Manoj, B. S.; Rao, R., "On the traffic behavior of distributed wireless mesh networks," *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a* , vol., no., pp.6 pp.,586

[16] Lei, W.; Landfeldt, B., "On the problem of placing Mobility Anchor Points in Wireless Mesh Networks," *Operator-Assisted (Wireless Mesh) Community Networks, 2006 1st Workshop on* , vol., no., pp.1,8, Sept. 2006

[17] Redwan, H.; Ki-Hyung Kim, "Survey of Security Requirements, Attacks and Network Integration in Wireless Mesh Networks," *Frontier of Computer Science and Technology, 2008. FCST '08. Japan-China Joint Workshop on* , vol., no., pp.3,9, 27-28 Dec. 2008

[18] Abid, M.R.; Khallaayoun, A; Harroud, H.; Lghoul, R.; Boulmalf, M.; Benhaddou, D., "A Wireless Mesh Architecture for the Advanced Metering Infrastructure in Residential Smart Grids," *Green Technologies Conference, 2013 IEEE* , vol., no., pp.338,344, 4-5 April 2013

[19] Yichi Zhang; Weiqing Sun; Lingfeng Wang; Hong Wang; Green, R.C.; Alam, M., "A multi-level communication architecture of smart grid based on congestion aware wireless mesh network," *North American Power Symposium (NAPS), 2011* , vol., no., pp.1,6, 4-6 Aug. 2011

[20] Peng-Yong Kong, "Effects of Communication Network Performance on Dynamic Pricing in Smart Power Grid," *Systems Journal, IEEE* , vol.8, no.2, pp.533,541, June 2014

[21] Kim, Jaebeom; Kim, Dabin; Lim, Keun-Woo; Ko, Young-Bae; Lee, Sang-Youm, "Improving the reliability of IEEE 802.11s based wireless mesh networks for smart grid systems," *Communications and Networks, Journal of* , vol.14, no.6, pp.629,639, Dec. 2012

[22] Gharavi, H.; Bin Hu, "Multigate mesh routing for smart Grid last mile communications," *Wireless Communications and Networking Conference (WCNC), 2011 IEEE* , vol., no., pp.275,280, 28-31 March 2011

[23] Jingfang Huang; Honggang Wang; Yi Qian, "Smart grid communications in challenging environments," *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on* , vol., no., pp.552,557, 5-8 Nov. 2012

[24] Mihui Kim, "A survey on guaranteeing availability in smart grid communications," *Advanced Communication Technology (ICACT), 2012 14th International Conference on* , vol., no., pp.314,317, 19-22 Feb. 2012

[25] Robinson, J.; Knightly, E.W., "A Performance Study of Deployment Factors in Wireless Mesh Networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* , vol., no., pp.2054,2062, 6-12 May 2007

[26] Ponci, F.; Monti, A.; Benigni, A., "Simulation for the design of Smart Grid controls," *Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on* , vol., no., pp.73,78, 17-17 Oct. 2011

[27] Rong Liu; Monti, A.; Ponci, F.; Smith, A.H.C., "A Design Approach For Digital Controllers Using Reconfigurable Network-Based Measurements," *Instrumentation and Measurement, IEEE Transactions on* , vol.59, no.5, pp.1073,1081, May 2010

[28] Calcada, T.; Cortez, P.; Ricardo, M., "Using data mining to study the impact of topology characteristics on the performance of wireless mesh networks," *Wireless Communications and Networking Conference (WCNC), 2012 IEEE* , vol., no., pp.1725,1730, 1-4 April 2012

[29] M. L. Sichitiu, “Wireless Mesh Networks: Opportunities and Challenges,” in *6th World Wide Congress*, San Francisco, CA, 2005, pp. 318-323

[30] Chandra, R.; Lili Qiu; Jain, K.; Mahdian, M., "Optimizing the placement of Internet TAPs in wireless neighborhood networks," *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on* , vol., no., pp.271,282, 5-8 Oct. 2004

[31] Carullo, S.P.; Nwankpa, C.O., "Experimental validation of a model for an information-embedded power system," *Power Delivery, IEEE Transactions on* , vol.20, no.3, pp.1853,1863, July 2005

[32] Mets, K.; Ojea, J.A; Develder, C., "Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis,"*Communications Surveys & Tutorials, IEEE* , vol.16, no.3, pp.1771,1796, Third Quarter 2014

[33] Stifter, M.; Widl, E.; Andren, F.; Elsheikh, A; Strasser, T.; Palensky, P., "Co-simulation of components, controls and power systems based on open source software," *Power and Energy Society General Meeting (PES), 2013 IEEE* , vol., no., pp.1,5, 21-25 July 2013

[34] Li, W.; Monti, A.; Luo, M.; Dougal, R.A., "VPNET: A co-simulation framework for analyzing communication channel effects on power systems," *Electric Ship Technologies Symposium (ESTS), 2011 IEEE* , vol., no., pp.143,149, 10-13 April 2011

[35] Ramanathan, Ram; Rosales-Hain, R., "Topology control of multihop wireless networks using transmit power adjustment," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.2, no., pp.404,413 vol.2, 2000

[36] Amante, C. and B.W. Eakins, 2009. ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis. NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center, NOAA

[37] Sood, V.K.; Fischer, D.; Eklund, J.M.; Brown, T., "Developing a communication infrastructure for the Smart Grid," Electrical Power & Energy Conference (EPEC), 2009 IEEE , vol., no., pp.1,7, 22-23 Oct. 2009

[38] Hao Liang; Bong Jun Choi; Abdrabou, A.; Weihua Zhuang; Xuemin Shen, "Decentralized Economic Dispatch in Microgrids via Heterogeneous Wireless Networks," Selected Areas in Communications, IEEE Journal on , vol.30, no.6, pp.1061,1074, July 2012

[39] Flueck, A.J.; Hsiao-Dong Chiang, "Solving the nonlinear power flow equations with an inexact Newton method using GMRES," Power Systems, IEEE Transactions on , vol.13, no.2, pp.267,273, May 1998

[40] Tinney, William F.; Hart, C.E., "Power Flow Solution by Newton's Method," Power Apparatus and Systems, IEEE Transactions on , vol.PAS-86, no.11, pp.1449,1460, Nov. 1967

[41] Ahmadi, H.; Marti, J.R., "Power flow formulation based on a mixed-linear and nonlinear system of equations," Environment and Electrical Engineering (EEEIC), 2013 13th International Conference on , vol., no., pp.27,32, 1-3 Nov. 2013

[42] Wu, F.F., "Analysis of Power Flows without Solving Power Flow Equations," American Control Conference, 1983 , vol., no., pp.418,419, 22-24 June 1983

[43] Osano, M.; Capretz, M.A.M., "A distributed method for solving nonlinear equations applying the power load flow calculation," System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on , vol.5, no., pp.676,680 vol.5, 7-10 Jan. 1997

[44] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," Power Systems, IEEE Transactions on, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[45] Group, W., "Common Format For Exchange of Solved Load Flow Data," *Power Apparatus and Systems, IEEE Transactions on* , vol.PAS-92, no.6, pp.1916,1925, Nov. 1973

[46] U.S.-Canada Power System Outage Task Force. (2004, April). Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations [Online]. Available: http://certs.lbl.gov/pdf/blackoutfinal-web.pdf

[47] Chadwick, J.E., "How a smarter grid could have prevented the 2003 U.S. cascading blackout," Power and Energy Conference at Illinois (PECI), 2013 IEEE , vol., no., pp.65,71, 22-23 Feb. 2013

[48] Sun, AX.; Phan, D.T.; Ghosh, S., "Fully decentralized AC optimal power flow algorithms," *Power and Energy Society General Meeting (PES), 2013 IEEE* , vol., no., pp.1,5, 21-25 July 2013

[49] HomChaudhuri, B.; Kumar, M.; Devabhaktuni, V., "Market based approach for solving optimal power flow problem in smart grid," *American Control Conference (ACC), 2012* , vol., no., pp.3095,3100, 27-29 June 2012

Appendix A - WMN_script

```
%WMN_script
%run this script to be prompted to enter the inputs needed to model a
WMN
%this script calls all remaining scripts to generate the WMN model

clear;
clc;
close all;

user.signal_frequency = input('Signal frequency (Hz): ');
user.TX_limit = input('Transmit power limit (dBm): ');
user.time_per_hop = input('Forwarding time per node (seconds): ');
user.message_length_kb = input('Message length (Kb): ');
user.overhead_bitrate_link_kbps = input('Network Overhead per link
(Kb/s): ');
user.overhead_bitrate_node_kbps = input('Network Overhead per node
(Kb/s): ');
user.duplex = input('Duplex (1) or Simplex (0) communications: ');
user.TAP_percentage_max = input('Maximum Percentage of TAP nodes: ');
user.direct_link_min = input('Minimum required direct links per node:
');
latrange_temp = input('Lattitude lower bound in degrees (+ for North, -
for South): ');
latrange_temp2 = input('Lattitude upper bound in degrees (+ for North,
- for South): ');
user.latrange = sort([latrange_temp latrange_temp2]);
longrange_temp = input('Longitude lower bound in degrees (+ for East, -
for West): ');
longrange_temp2 = input('Longitude upper bound in degrees (+ for East,
- for West): ');
user.longrange = sort([longrange_temp longrange_temp2]);
user.excluded_nodes = input('Node(s) to be excluded (enter multiple
nodes as a 1xN array): ');
user.fileoutput = input('Create output files 1 (yes) or 0 (no): ');

%-----------------------------------------------------------------------
----
%store time for start of calculations
format shortg;
time_start = clock;
fprintf('Time for start of script: ');
fprintf('%02i:%02i:%04.2f\n',time_start(4),time_start(5),time_start(6))
;

skip_WMN_readfile = WMN_datacheck(user, 'user');
[status,result] = system(['fc ' 'nodes.xls' ' '
'previous_data/nodes.xls']);

if (skip_WMN_readfile == 1) && (status == 0) &&
(exist('previous_data/readfile_old.mat', 'file') ~= 0)
    load('previous_data/readfile_old.mat');
    readfile = readfile_old;
```

```
else
    fprintf('Reading input file...\n');
    [ readfile.nodes_number, readfile.nodes, readfile.nodes_spacing,
readfile.nodes_bitrate_kbps, readfile.nodes_TX, readfile.nodes_RX,
readfile.nodes_TX_period, readfile.nodes_TAP, readfile.nodes_lattitude,
readfile.nodes_longitude, readfile.nodes_altitude, readfile.Z1,
readfile.refvec1 ] = WMN_readfile( user.latrange, user.longrange );
end

%-----------------------------------------------------------------------
----
%display time between each function
time = clock;
fprintf('WMN_readfile completed: ');
fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

skip_WMN_power = (WMN_datacheck(user, 'user')) &&
(WMN_datacheck(readfile, 'readfile')) &&
(exist('previous_data/power_old.mat', 'file') ~= 0);

if (skip_WMN_power == 1)
    load('previous_data/power_old.mat');
    power = power_old;
else
    fprintf('Calculating node transmit power...\n');
    [ power.nodes_TX, power.nodes_RX, power.direct_links_total ] =
WMN_power( readfile.nodes_number, readfile.nodes,
readfile.nodes_spacing, user.signal_frequency, readfile.nodes_TX,
readfile.nodes_RX, user.TX_limit, user.direct_link_min,
user.fileoutput, readfile.Z1, readfile.refvec1 );
end



%-----------------------------------------------------------------------
----
%display time between each function
time = clock;
fprintf('WMN_power completed: ');
fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

skip_WMN_routes = (WMN_datacheck(user, 'user')) &&
(WMN_datacheck(readfile, 'readfile')) && (WMN_datacheck(power,
'power')) && (exist('previous_data/routes_old.mat', 'file') ~= 0);

if (skip_WMN_routes == 1)
    load('previous_data/routes_old.mat');
    routes = routes_old;
else
    fprintf('Calculating possible routes...\n');
    [ routes.map, routes.route_all, routes.routes,
routes.nodes_receive_boolean, routes.LOS, routes.nodes_connect_boolean,
routes.nodes_signal_power, routes.links_number ] =
```

```
WMN_routes( readfile.nodes_number, readfile.nodes,
readfile.nodes_spacing, user.signal_frequency, power.nodes_TX,
power.nodes_RX, user.fileoutput, readfile.Z1, readfile.refvec1 );
    %[ routes.route_path_valid, routes.route_total_paths,
routes.route_all, routes.nodes_receive_boolean, routes.LOS,
routes.nodes_connect_boolean, routes.nodes_signal_power,
routes.links_number ] = WMN_routes( readfile.nodes_number,
readfile.nodes, readfile.nodes_spacing, user.signal_frequency,
power.nodes_TX, power.nodes_RX, user.fileoutput, readfile.Z1,
readfile.refvec1 );
end

% size(routes.routes,1)

%-----------------------------------------------------------------------
----
%display time between each function
time = clock;
fprintf('WMN_routes completed: ');
fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

%-----------------------------------------------------------------------
----
paths.paths_valid_all = zeros(0,size(routes.routes,2));

skip_WMN_paths = (WMN_datacheck(user, 'user')) &&
(WMN_datacheck(routes, 'routes')) &&
(exist('previous_data/paths_old.mat', 'file') ~= 0);

if (skip_WMN_paths == 1)
    load('previous_data/paths_old.mat');
    paths = paths_old;
else
    fprintf('Calculating path information...\n');
    start_node = 1;
    end_node = 1;
                [ paths.paths_valid ] = WMN_paths( routes.routes,
start_node, end_node, user.excluded_nodes );
end

% size(paths.paths_valid,1)
paths.paths_valid_all = paths.paths_valid;

%-----------------------------------------------------------------------
----
%display time between each function
time = clock;
fprintf('WMN_paths completed: ');
fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

%-----------------------------------------------------------------------
----
placement_count = 1;
```

```
placement.max_utilization = 101;
pathchar.TAP_nodes = find(readfile.nodes_TAP == 1);

overhead_bitrate_kbps =
(user.overhead_bitrate_link_kbps*routes.links_number) +
(user.overhead_bitrate_node_kbps*readfile.nodes_number);

skip_WMN_pathchar = (WMN_datacheck(user, 'user')) &&
(WMN_datacheck(readfile, 'readfile')) && (WMN_datacheck(routes,
'routes')) && (WMN_datacheck(paths, 'paths')) &&
(exist('previous_data/pathchar_old.mat', 'file') ~= 0) &&
(exist('previous_data/util_old.mat', 'file') ~= 0) &&
(exist('previous_data/placement_old.mat', 'file') ~= 0);

if (skip_WMN_pathchar == 1)
    load('previous_data/pathchar_old.mat');
    pathchar = pathchar_old;
    load('previous_data/util_old.mat');
    util = util_old;
    load('previous_data/placement_old.mat');
    placement = placement_old;
else
    fprintf('Calculating characterisitcs and TAP placement...\n');
    while ((placement.max_utilization > 100) &&
(length(pathchar.TAP_nodes) < readfile.nodes_number)) ||
(skip_WMN_pathchar == 1)

    [  pathchar.paths_best, pathchar.paths_worst, pathchar.paths_all,
pathchar.TAP_nodes, pathchar.non_TAP_nodes ] =
WMN_pathcharacteristics( paths.paths_valid, readfile.nodes_number,
readfile.nodes_spacing, user.time_per_hop, user.message_length_kb,
readfile.nodes_bitrate_kbps, pathchar.TAP_nodes,
overhead_bitrate_kbps );
    [  util.link_utilization, util.link_counts, util.nodes_overhead ] =
WMN_utilization( pathchar.paths_best, routes.route_all,
readfile.nodes_number, pathchar.TAP_nodes, readfile.nodes_bitrate_kbps,
readfile.nodes_TX_period, user.overhead_bitrate_node_kbps,
user.message_length_kb, user.duplex );
    [ placement.TAP_nodes, placement.non_TAP_nodes,
placement.max_utilization ] = WMN_TAPplacement( util.link_utilization,
readfile.nodes_number, pathchar.TAP_nodes );

    %display time between each function
    time = clock;
    fprintf('WMN_TAPplacement completed (iteration: %i):',
placement_count);
    fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

    placement_count = placement_count + 1;

    end
```

```
    [  pathchar.paths_best, pathchar.paths_worst, pathchar.paths_all,
pathchar.TAP_nodes, pathchar.non_TAP_nodes ] =
WMN_pathcharacteristics( paths.paths_valid, readfile.nodes_number,
readfile.nodes_spacing, user.time_per_hop, user.message_length_kb,
readfile.nodes_bitrate_kbps, pathchar.TAP_nodes,
overhead_bitrate_kbps );
    [  util.link_utilization, util.link_counts, util.nodes_overhead ] =
WMN_utilization( pathchar.paths_best, routes.route_all,
readfile.nodes_number, pathchar.TAP_nodes, readfile.nodes_bitrate_kbps,
readfile.nodes_TX_period, user.overhead_bitrate_node_kbps,
user.message_length_kb, user.duplex );

    time = clock;
    fprintf('WMN_TAPplacement completed:');
    fprintf('%02i:%02i:%04.2f\n',time(4),time(5),time(6));

end

if (length(pathchar.TAP_nodes) >=
(readfile.nodes_number*(user.TAP_percentage_max/100)))
    fprintf('Insufficient network capacity\n');
end

fprintf('Generating plots...\n');
%----------------------------------------------------------------------
----
WMN_plot( readfile.nodes_number, readfile.nodes,
readfile.nodes_lattitude, readfile.nodes_longitude,
readfile.nodes_altitude, routes.nodes_receive_boolean, routes.LOS,
routes.nodes_connect_boolean, user.latrange, user.longrange,
readfile.Z1, readfile.refvec1, user.fileoutput );

%----------------------------------------------------------------------
----
%store time for end of calculations
format shortg;
time_end = clock;
fprintf('WMN_plot completed: ');
fprintf('%02i:%02i:%04.2f\n',time_end(4),time_end(5),time_end(6));

%----------------------------------------------------------------------
----
%save results for comparison in next iteration
user_old = user;
readfile_old = readfile;
power_old = power;
routes_old = routes;
paths_old = paths;
pathchar_old = pathchar;
util_old = util;
placement_old = placement;
```

```
save previous_data/user_old.mat user_old;
save previous_data/readfile_old.mat readfile_old;
save previous_data/power_old.mat power_old;
save previous_data/routes_old.mat routes_old;
save previous_data/paths_old.mat paths_old;
save previous_data/pathchar_old.mat pathchar_old;
save previous_data/util_old.mat util_old;
save previous_data/placement_old.mat placement_old;

[status, result] = system(['COPY' ' nodes.xls' '
previous_data/nodes.xls']);

%----------------------------------------------------------------------
----
%display total duration of script

duration_hours = time_end(4)-time_start(4);
duration_minutes = time_end(5) - time_start(5);
duration_seconds = time_end(6) - time_start(6);

if (duration_seconds < 0)
    duration_seconds = 60 + duration_seconds;
    duration_minutes = duration_minutes - 1;
end

if (duration_minutes < 0)
    duration_minutes = 60 + duration_minutes;
    duration_hours = duration_hours - 1;
end

if (duration_hours < 0)
    duration_hours = 24 + duration_hours;
end

fprintf('Total script duration: ');
fprintf('%02.0f:%02.0f:%04.2f\n',duration_hours,duration_minutes,durati
on_seconds)
```

Appendix B - WMN_readfile

```
function [ nodes_number, nodes, nodes_spacing, nodes_bitrate_kbps,
nodes_TX, nodes_RX, nodes_TX_period, nodes_TAP, nodes_lattitude,
nodes_longitude, nodes_altitude, Z1, refvec1 ] =
WMN_readfile( latrange, longrange )
%this script reads the information needed for the WMN model based on
the
%nodes.xls file and the elevation data file

Earth_radius = 6371000;

c = 2.99792458e8;                      %speed of light, m/s

%---------------------------------------------------------------------
----
%excel file input for x,y,z coordinates

filename = 'nodes.xls';
    nodes_lattitude = xlsread(filename,'A:A');
    nodes_longitude = xlsread(filename,'B:B');
    nodes_antennaheight = xlsread(filename,'C:C');
    nodes_bitrate_kbps = xlsread(filename, 'D:D');
    nodes_TX_period = xlsread(filename, 'E:E');
    nodes_TX = xlsread(filename,'F:F');
    nodes_RX = xlsread(filename,'G:G');
    nodes_TAP = xlsread(filename, 'H:H');
    nodes_number = min([length(nodes_lattitude),
length(nodes_longitude)]);
    nodes = [nodes_lattitude nodes_longitude];
    nodes(:,4) = nodes_antennaheight;

%---------------------------------------------------------------------
----
%read elevation data

samplefactor = 1;           %arc minutes per sample
scale = 60/samplefactor;
[Z, refvec] = etopo('mapping/etopo1_bed_c_i2.bin', samplefactor);

latminscaled = (min(latrange)+90)*scale;
latmaxscaled = (max(latrange)+90)*scale;
longminscaled = (180+min(longrange))*scale;
longmaxscaled = (180+max(longrange))*scale;

latmax_index = find(abs(latrange) == max(abs(latrange)));
longmax_index = find(abs(longrange) == max(abs(longrange)));
latmax = latrange(latmax_index);
longmax = longrange(longmax_index);

Z1 = Z(latminscaled:latmaxscaled,longminscaled:longmaxscaled);
refvec1 = [scale,latmax,longmax];
clear Z;
```

```
%----------------------------------------------------------------------
----
%calculate node altitude
for m = 1:1:nodes_number
    nodes(m,3)  = Z1(round((abs(nodes(m,2))-
min(abs(longrange)))*scale),round((abs(nodes(m,1))-
min(abs(latrange)))*scale));
end
nodes_altitude = nodes(:,3);

%----------------------------------------------------------------------
----
%calculate node spacing
for m = 1:1:nodes_number;
    for n = 1:1:nodes_number;
    nodes_spacing_xy(m,n) =
distance(nodes(m,1),nodes(m,2),nodes(n,1),nodes(n,2),Earth_radius);
    nodes_spacing_z(m,n)  = abs(nodes(m,3)-nodes(n,3));
    nodes_spacing(m,n) =
sqrt((nodes_spacing_xy(m,n)^2)+(nodes_spacing_z(m,n)^2));
    end
end


end
```

Appendix C - WMN_power

```
function [ nodes_TX, nodes_RX, direct_links_total ] =
WMN_power( nodes_number, nodes, nodes_spacing, signal_frequency,
nodes_TX, nodes_RX, TX_limit, direct_link_min, fileoutput, Z1,
refvec1 )
%this script dynamically assigns a transmit power level to nodes based
on
%the number of direct connections they have to other nodes in the
network.
%The transmit power will be increased up to the maximum transmit level
or
%until the node has the minimum required amount of direct connections

Earth_radius = 6371000;

c = 2.99792458e8;                      %speed of light, m/s

nodes_connections = zeros(nodes_number);
nodes_power_old = zeros(nodes_number);
LOS = zeros(nodes_number, nodes_number);

completed = 0;

while (completed ~= 1)

completed = 1;
    %-------------------------------------------------------------------
--------
    %calculate signal strength
    for m = 1:1:nodes_number;
        for n = 1:1:nodes_number;
        FSPL(m,n) =
20*log10((4*pi*nodes_spacing(m,n)*(signal_frequency/c)));
        nodes_signal_power(m,n) = nodes_TX(m) - FSPL(m,n);
        nodes_receive_boolean(m,n) = ((nodes_signal_power(m,n) >=
nodes_RX(m)) & (nodes_signal_power(m,n) < nodes_TX(m)));
        end
    end

    for m = 1:1:nodes_number
        for n = m:1:nodes_number
            nodes_receive_boolean(m,n) = (nodes_receive_boolean(m,n) &
nodes_receive_boolean(n,m));
            nodes_receive_boolean(n,m) = nodes_receive_boolean(m,n);
        end
    end

    %-------------------------------------------------------------------
--------
    %calculate node line of sight
    for m = 1:1:nodes_number
        for n = 1:1:nodes_number
            if ((nodes_receive_boolean(m,n) == 1) && (LOS(m,n) == 0))
```

```
                LOS(m,n) =
los2(Z1,refvec1,nodes(m,1),nodes(m,2),nodes(n,1),nodes(n,2),nodes(m,4),
nodes(n,4));
            end
        end
    end



    %------------------------------------------------------------------
--------
    %determine the nodes that can theoretically connect based on signal
    %strength and have direct line of sight
    nodes_connect_boolean = nodes_receive_boolean .* LOS;
    nodes_receive_power = nodes_connect_boolean .* nodes_signal_power;

    %------------------------------------------------------------------
--------
    %determine all possible next nodes for every node
    for m = 1:1:nodes_number
        p = 1;
        for n = 1:1:nodes_number
            if (nodes_connect_boolean(m,n) == 1)
                route_all(p,1,m) = n;
                p = p + 1;
                nodes_connections(m) = (p - 1);
            end

        end

            if (p <= direct_link_min) && (nodes_TX(m) < TX_limit);
                completed = 0;
                nodes_TX(m) = nodes_TX(m) + 1;
            end
    end

end

direct_links_total = length(find(route_all ~= 0));


end
```

Appendix D - WMN_routes

```
function [ map, route_all, routes_all, nodes_receive_boolean, LOS, 
nodes_connect_boolean, nodes_signal_power, links_number ] = 
WMN_routes( nodes_number, nodes, nodes_spacing, signal_frequency, 
nodes_TX, nodes_RX, fileoutput, Z1, refvec1 )
%this script determines the network map. The map determines possible 
routes
%for each node to communicate with each other node

Earth_radius = 6371000;

c = 2.99792458e8;                     %speed of light, m/s

%----------------------------------------------------------------------
----
%calculate signal strength
for m = 1:1:nodes_number;
    for n = 1:1:nodes_number;
    FSPL(m,n) = 
20*log10((4*pi*nodes_spacing(m,n)*(signal_frequency/c)));
    nodes_signal_power(m,n) = nodes_TX(m) - FSPL(m,n);
    nodes_receive_boolean(m,n) = ((nodes_signal_power(m,n) >= 
nodes_RX(m)) & (nodes_signal_power(m,n) < nodes_TX(m)));
    end
end

for m = 1:1:nodes_number
    for n = m:1:nodes_number
        nodes_receive_boolean(m,n) = (nodes_receive_boolean(m,n) & 
nodes_receive_boolean(n,m));
        nodes_receive_boolean(n,m) = nodes_receive_boolean(m,n);
    end
end

%----------------------------------------------------------------------
----
%calculate node line of sight

h = waitbar(0,sprintf('LOS Progress'));

for m = 1:1:nodes_number
    for n = m:1:nodes_number
        if (nodes_receive_boolean(m,n) == 1)
            LOS(m,n) = 
los2(Z1,refvec1,nodes(m,1),nodes(m,2),nodes(n,1),nodes(n,2),nodes(m,4),
nodes(n,4));
            LOS(n,m) = LOS(m,n);
        end

        percent = (m/nodes_number);
        h = waitbar(percent,h,sprintf('LOS 
Progress %4.2f%%',percent*100));
```

```
    end
end

close(h);

%-----------------------------------------------------------------------
----
%determine the nodes that can theoretically connect based on signal
%strength and have direct line of sight
nodes_connect_boolean = nodes_receive_boolean .* LOS;
nodes_receive_power = nodes_connect_boolean .* nodes_signal_power;

%-----------------------------------------------------------------------
----
%determine all possible next nodes for every node
for m = 1:1:nodes_number
    p = 1;
    for n = 1:1:nodes_number
        if (nodes_connect_boolean(m,n) == 1)
            route_all(p,1,m) = n;
            p = p + 1;
        end
    end
end

links_number = length(find(route_all ~= 0));

h = waitbar(0,sprintf('Map Progress'));

for m = 1:1:nodes_number
    map(1,1,m) = m;
    previous_neighbors = m;

    complete = 0;
    n = 1;

    while (complete == 0) && (n <= nodes_number)
    complete = 1;

    neighbors = zeros(0,0);

        for nn = 1:1:length(map(:,n,m));
            if (map(nn,n,m) ~= 0)
                temp = route_all(:,1,map(nn,n,m));
                temp = temp(temp~=0);
                neighbors = vertcat(temp,neighbors);
                neighbors = unique(neighbors);
            end
        end

        temp3 = intersect(neighbors,previous_neighbors);
        neighbors = setxor(neighbors,temp3);
        previous_neighbors = vertcat(previous_neighbors,neighbors);
```

```
        previous_neighbors = unique(previous_neighbors);
        if (isempty(neighbors) ~= 1)
            map(1:length(neighbors),n+1,m) = neighbors;
            complete = 0;
        end
        n = n+1;
    end

percent = (m/nodes_number);
h = waitbar(percent,h,sprintf('Map Progress %4.2f%%',percent*100));

end

close(h);

routes = zeros(0,(size(map,2) + 3));
routes_all = zeros(0,(size(map,2) + 3));
route_temp = zeros(1,(size(map,2) + 3));
%route_temp = zeros(1,nodes_number);

h = waitbar(0,sprintf('Routes Progress'));

for m = 1:1:nodes_number
    routes = zeros(0,(size(map,2) + 3));
    route_temp = zeros(1,(size(map,2) + 3));
    route_temp(1) = m;
    route_temp(size(map,2)+1) = m;
    for n = 1:1:length(find(map(:,2,m) ~= 0))
        route_temp(2) = map(n,2,m);
        route_temp(size(map,2)+2) = route_temp(2);
        route_temp(size(map,2)+3) = 1;
        routes = vertcat(routes,route_temp);
        route_temp(2) = 0;
    end

    for p = 2:1:(length(route_temp)-4)
        indices1 = find(routes(:,1) == m);
        indices2 = find(routes(:,p) ~= 0);
        indices = intersect(indices1,indices2);
        for q = 1:1:length(indices)
            route_temp = routes(indices(q),:);
            node1 = route_temp(min(find(route_temp == 0)) - 1);
            for r = 1:1:length(find(route_all(:,1,node1) ~= 0))
                node2 = route_all(r,1,node1);
                if (isempty(find(route_all(:,1,node1) == node2)) == 0)
&& (isempty(find(route_temp == node2)) == 1) &&
(isempty(find(map(:,p+1,m) == node2)) == 0)
                    route_temp(p+1) = node2;
                    route_temp(size(map,2)+2) = node2;
                    route_temp(size(map,2)+3) = p;
                    routes = vertcat(routes,route_temp);
                    route_temp = routes(indices(q),:);
                end
```

```
            end
        end
    end
    routes_all = vertcat(routes_all,routes);

    percent = (m/nodes_number);
    h = waitbar(percent,h,sprintf('Route
Progress %4.2f%%',percent*100));
end

close(h);

end
```

Appendix E - WMN_paths

```
function [ paths_valid ] = WMN_paths( route_path_valid, start_node,
end_node, excluded_nodes )
%this script removes paths with the specified excluded nodes from the
valid
%paths list

%set the default value to exclude a path
include = 0;


%create empty array for the output
matrix_size = size(route_path_valid,2);
paths_valid = zeros(0,matrix_size);
indices = zeros(0,1);
indices_temp = zeros(0,1);

h = waitbar(0,sprintf('Paths Progress'));

for m = 1:1:(size(route_path_valid,2)-3)
    for k = 1:1:length(excluded_nodes)
        if excluded_nodes(k) ~= 0
            indices_temp = find(route_path_valid(:,m) ==
excluded_nodes(k));
        end
    end
    indices = union(indices,indices_temp);

    percent = (m/(size(route_path_valid,2)-3));
    h = waitbar(percent,h,sprintf('Paths
Progress %4.2f%%',percent*100));

end

close(h);

route_path_valid(indices,:) = [];

paths_valid = route_path_valid;

end
```

Appendix F - WMN_pathcharacteristics

```
function [  paths_best, paths_worst, paths_all, TAP_nodes,
non_TAP_nodes ] = WMN_pathcharacteristics( paths_valid, nodes_number,
nodes_spacing, time_per_hop, message_length_kb, nodes_bitrate_kbps,
TAP_nodes, overhead_bitrate_kbps )
%this script calculates the path characteristics (bitrate, time,
distance,
%hops) and determines the best and worst paths available.

%-----------------------------------------------------------------------
c = 2.99792458e8;                          %speed of light, m/s

%-----------------------------------------------------------------------
%find the distance traveled in each path
matrix_size = size(paths_valid,2);
%current_path = zeros(0,matrix_size-3);
paths_valid_distance = zeros(size(paths_valid,1),1);
paths_valid_time = zeros(size(paths_valid,1),1);
paths_valid_hops = zeros(size(paths_valid,1),1);
paths_valid_bitrate = zeros(size(paths_valid,1),1);
paths_all = zeros(0,matrix_size+3);
paths_best = zeros(0,matrix_size+3);
paths_worst = zeros(0,matrix_size+3);
paths_best_temp = zeros(0,matrix_size+3);
paths_worst_temp = zeros(0,matrix_size+3);

paths_valid_hops = paths_valid(:,matrix_size);

paths_nodes = paths_valid(:,(1:(matrix_size-3)));
paths_nodes_bitrate = zeros(size(paths_nodes,1),size(paths_nodes,2));

h = waitbar(0,sprintf('Nodes Bitrate Progress'));

for m = 1:1:nodes_number
    indices = find(paths_nodes == m);
    paths_nodes_bitrate(indices) = nodes_bitrate_kbps(m);

    percent = (m/nodes_number);
    h = waitbar(percent,h,sprintf('Nodes Bitrate
Progress %4.2f%%',percent*100));

end
close(h);

h = waitbar(0,sprintf('Extended Paths Progress'));
for n = 1:1:size(paths_nodes_bitrate,1)
    indices_bitrate = find(paths_nodes_bitrate(n,(1:matrix_size-3)) ~=
0);

    paths_valid_bitrate(n) =
min(paths_nodes_bitrate(n,indices_bitrate)) - overhead_bitrate_kbps;
```

```
    current_path = paths_valid(n,(1:(matrix_size-3)));
    current_path_distance = 0;
    for p = 1:1:paths_valid_hops(n)
        current_path_distance = current_path_distance +
nodes_spacing(current_path(p), current_path(p+1));
    end
    paths_valid_distance(n) = current_path_distance;

    paths_valid_time(n) = (message_length_kb/paths_valid_bitrate(n)) +
(paths_valid_hops(n)*time_per_hop) + (current_path_distance/c);

    if (rem(n,100) == 0)
        percent = (n/size(paths_nodes_bitrate,1));
        h = waitbar(percent,h,sprintf('Extended Paths
Progress %4.2f%%\n %i/%i',percent*100,n,size(paths_nodes_bitrate,1)));
    end
end
close(h)

if (size(paths_nodes_bitrate,1) > 10000)
    for n = 1:1:ceil(size(paths_nodes_bitrate,1)/10000)
        if (n ~= ceil(size(paths_nodes_bitrate,1)/10000))
            paths_extended((((n-1)*10000+1):(n*10000)),:) =
horzcat(paths_valid(((n-
1)*10000+1):(n*10000),:),paths_valid_distance(((n-
1)*10000+1):(n*10000)),paths_valid_time(((n-
1)*10000+1):(n*10000)),paths_valid_bitrate(((n-1)*10000+1):(n*10000)));
        else
            paths_extended(((n*10000):size(paths_nodes_bitrate,1)),:) =
horzcat(paths_valid(((n*10000):size(paths_nodes_bitrate,1)),:),
paths_valid_distance((n*10000):size(paths_nodes_bitrate,1)),
paths_valid_time((n*10000):size(paths_nodes_bitrate,1)),
paths_valid_bitrate((n*10000):size(paths_nodes_bitrate,1)));
        end
    end
else
    paths_extended =
horzcat(paths_valid,paths_valid_distance,paths_valid_time,paths_valid_b
itrate);
end

TAP_nodes_number = length(TAP_nodes);
non_TAP_nodes_indices = 0;
non_TAP_nodes = 1:1:nodes_number;
for m = 1:1:length(TAP_nodes)
    non_TAP_nodes_indices(m) = find(non_TAP_nodes == TAP_nodes(m));
end
non_TAP_nodes(non_TAP_nodes_indices) = [];
non_TAP_nodes_number = length(non_TAP_nodes);

for m = 1:1:non_TAP_nodes_number
    start_node = non_TAP_nodes(m);

    paths_all_temp = zeros(0,matrix_size+3);
```

```
    paths_best_temp = zeros(0,matrix_size+3);
    paths_worst_temp = zeros(0,matrix_size+3);

    for n  = 1:1:TAP_nodes_number
        end_node = TAP_nodes(n);

        indices_start = find(paths_extended(:,(matrix_size-2)) ==
start_node);
        indices_end = find(paths_extended(:,(matrix_size-1)) ==
end_node);
        indices = intersect(indices_start,indices_end);

        for mm = 1:1:length(indices)
            %current_path_extended = paths_extended(mm,:);

            %current_start_node = current_path_extended(matrix_size-2);
            %current_end_node = current_path_extended(matrix_size-1);

            %if ((current_start_node == start_node) &&
(current_end_node == end_node))
                paths_all_temp = vertcat(paths_all_temp,
paths_extended(indices(mm),:));
                paths_worst_temp = paths_all_temp;
                paths_best_temp = paths_all_temp;
            %end
        end

    path_time = paths_best_temp(:,(matrix_size+2));
    max_time = max(path_time);
    min_time = min(path_time);

    paths_best_time_indices = find(path_time == min_time);
    paths_best_temp = paths_best_temp(paths_best_time_indices,:);
    paths_worst_time_indices = find(path_time == max_time);
    paths_worst_temp = paths_worst_temp(paths_worst_time_indices,:);

    end

    if (size(paths_all_temp,1) > 0)
        paths_all = vertcat(paths_all, paths_all_temp);
        paths_best = vertcat(paths_best, paths_best_temp(1,:));
        paths_worst = vertcat(paths_worst, paths_worst_temp(1,:));
    end

    if (size(paths_all_temp,1) == 0)
        TAP_nodes(TAP_nodes_number + 1) = start_node;
        TAP_nodes_number = TAP_nodes_number +1;
    end

end
end
```

Appendix G - WMN_utilization

```
function [  link_utilization, link_counts, nodes_overhead ] =
WMN_utilization( paths_best, route_all, nodes_number, TAP_nodes,
nodes_bitrate_kbps, nodes_TX_period, overhead_bitrate_node_kbps,
message_length_kb, duplex )
%this script determines the node utilization

link_bitrates = zeros(0,3);
matrix_size = size(paths_best,2);
paths_temp = zeros(0,(matrix_size));
path_shortest_only = zeros(0,(matrix_size));
link_counts = zeros(1,nodes_number);
node_usage_kbps = zeros(1,nodes_number);
link_utilization = zeros(1,nodes_number);
nodes_overhead = zeros(1,nodes_number);

non_TAP_nodes_indices = 0;
non_TAP_nodes = 1:1:nodes_number;
for m = 1:1:length(TAP_nodes)
    non_TAP_nodes_indices(m) = find(non_TAP_nodes == TAP_nodes(m));
end
non_TAP_nodes(non_TAP_nodes_indices) = [];

for m = 1:1:size(paths_best,1)
    current_path = paths_best(m,:);

    current_hops = current_path(matrix_size-3);

    if (duplex == 0)
        for n = 1:1:current_hops
            node1 = current_path(n);
            node2 = current_path(n+1);

            link_counts(node1) = link_counts(node1) + 1;
            link_counts(node2) = link_counts(node2) + 1;
            node_usage_kbps(node1) = node_usage_kbps(node1) +
(message_length_kb/nodes_TX_period(node1));
            node_usage_kbps(node2) = node_usage_kbps(node2) +
(message_length_kb/nodes_TX_period(node2));
        end
    end
    if (duplex == 1)
        path = current_path(1:(current_hops+1));
        link_counts(path) = link_counts(path) + 1;
        node_usage_kbps(path) = node_usage_kbps(path) +
(message_length_kb/nodes_TX_period(path));
    end

end
```

```
for m = 1:1:nodes_number
    nodes_overhead(m) = length(find(route_all(:,1,m) ~= 0)) *
overhead_bitrate_node_kbps;
    link_utilization(m) =
((node_usage_kbps(m)+nodes_overhead(m))/nodes_bitrate_kbps(m))*100;
end


end
```

Appendix H - WMN_TAPplacement

```
function [ TAP_nodes, non_TAP_nodes, max_utilization ] =
WMN_TAPplacement( link_utilization, nodes_number, TAP_nodes )
%this script determines if additional TAP nodes need to be assigned
based
%on the utilization of existing nodes in the network

TAP_nodes_number = length(TAP_nodes);
non_TAP_nodes_number = nodes_number - TAP_nodes_number;
non_TAP_nodes_indices = [];
non_TAP_nodes = 1:1:nodes_number;

for m = 1:1:length(TAP_nodes)
    non_TAP_nodes_indices(m) = find(non_TAP_nodes == TAP_nodes(m));
end

non_TAP_nodes(non_TAP_nodes_indices) = [];
max_utilization = max(link_utilization);

if (max_utilization > 100)

    [node_utilization nodes] = sort(link_utilization);

    for m = 1:1:nodes_number;
        if (isempty(find(TAP_nodes == nodes(m))))
            most_congested_node = nodes(m);
        end
    end

    TAP_nodes = vertcat(TAP_nodes, most_congested_node);
    new_TAP_node_index = find(non_TAP_nodes == most_congested_node);
    non_TAP_nodes(new_TAP_node_index) = [];
    link_utilization(most_congested_node) = [];

end

TAP_nodes_number = length(TAP_nodes);
non_TAP_nodes_number = length(non_TAP_nodes);
max_utilization = max(link_utilization);

end
```

Appendix I - Nodes.xls

Latitude	Longitude	Antenna Height	Bandwidth (Kbps)	TX interval	Min TX power	RX	TAP
37.25040556	-79.92116389	50	12000	3	36	-96	1
36.70890278	-79.93444722	50	12000	3	36	-96	1
37.37453056	-79.90068333	50	12000	3	36	-96	1
37.26741944	-80.02298333	50	12000	3	36	-96	1
37.36967778	-80.86140278	50	12000	3	36	-96	1
37.0722	-80.58239167	50	12000	3	36	-96	1
37.46213056	-79.189175	50	12000	3	36	-96	1
37.36	-80.53	15	12000	3	36	-96	0
37.18	-80.16	15	12000	3	36	-96	0
37.52	-79.49	15	12000	3	36	-96	0
36.84	-79.83	15	12000	3	36	-96	0
37.28375	-80.09834	15	12000	3	6	-96	0
37.29299	-80.05672	15	12000	3	6	-96	0
37.30253	-80.01673	15	12000	3	6	-96	0
37.32217	-79.96794	15	12000	3	6	-96	0
37.31043	-79.9393	15	12000	3	6	-96	0
37.3521	-79.87842	15	12000	3	6	-96	0
37.35335	-79.90018	15	12000	3	6	-96	0
37.27866	-79.8955	15	12000	3	6	-96	0
37.24108	-80.0178	15	12000	3	6	-96	0
37.24537	-79.95207	15	12000	3	6	-96	0
37.12995	-80.40898	15	12000	3	6	-96	0
37.2334	-80.41995	15	12000	3	6	-96	0
37.12874	-80.57533	15	12000	3	6	-96	0
36.69135	-79.87286	15	12000	3	6	-96	0
37.33437	-79.52361	15	12000	3	6	-96	0
37.40969	-79.14854	15	12000	3	6	-96	0
37.32171	-79.25631	15	12000	3	6	-96	0
36.9984	-79.88992	15	12000	3	6	-96	0
37.32813	-80.80679	15	12000	3	6	-96	0

Appendix J - PF_test

```
%Script used to generate the power flow results using MATPOWER

clear;
clc;

warning('off','all')
mpopt = mpoption('PF_ALG', 4, 'PF_MAX_IT_GS', 5000, 'PF_TOL', 10^(-
12),'OUT_ALL', 0, 'VERBOSE', 0);

iterations = 75;

[power_system] = PF_system(iterations);

mpc = 0;

for m = 1:1:iterations
    %mpc = 0;
    [mpc] = PF_variablesystem( power_system,mpc,mpopt,m );
    pfresults(m) = runpf(mpc,mpopt);

end

for m = 1:1:iterations

    PFBusNumber(:,m) = pfresults(m).bus(:,1);
    PFBusZone(:,m) = pfresults(m).bus(:,2);
    PFBusLoadP(:,m) = pfresults(m).bus(:,3);
    PFBusLoadQ(:,m) = pfresults(m).bus(:,4);
    PFBusVoltage(:,m) = pfresults(m).bus(:,8);
    PFBusAngle(:,m) = pfresults(m).bus(:,9);
    PFBusStatus(:,m) = pfresults(m).bus(:,11);

    PFGenBus(:,m) = pfresults(m).gen(:,1);
    PFGenActualP(:,m) = pfresults(m).gen(:,2);
    PFGenActualQ(:,m) = pfresults(m).gen(:,3);
    PFGenCapacityQmax(:,m) = pfresults(m).gen(:,4);
    PFGenCapacityQmin(:,m) = pfresults(m).gen(:,5);
    PFGenCapacityP(:,m) = pfresults(m).gen(:,9);
    PFGenStatus(:,m) = pfresults(m).gen(:,8);

    PFBranchFromBus(:,m) = pfresults(m).branch(:,1);
    PFBranchToBus(:,m) = pfresults(m).branch(:,2);
    PFBranchRating(:,m) = pfresults(m).branch(:,6);
    PFBranchR(:,m) = pfresults(m).branch(:,3);
    PFBranchX(:,m) = pfresults(m).branch(:,4);
    PFBranchB(:,m) = pfresults(m).branch(:,5);
    PFBranchFromInjectionP(:,m) = pfresults(m).branch(:,14);
    PFBranchFromInjectionQ(:,m) = pfresults(m).branch(:,15);
    PFBranchToInjectionP(:,m) = pfresults(m).branch(:,16);
    PFBranchToInjectionQ(:,m) = pfresults(m).branch(:,17);
    PFBranchStatus(:,m) = pfresults(m).branch(:,11);
```

```
    PFLoadTotalP(m) = sum(PFBusLoadP(:,m));
    PFLoadTotalQ(m) = sum(PFBusLoadQ(:,m));
    PFBranchLossesP(:,m) = abs(abs(PFBranchFromInjectionP(:,m)) -
abs(PFBranchToInjectionP(:,m)));
    PFBranchLossesTotalP(m) = sum(PFBranchLossesP(:,m));

    PFGenActualTotalP(m) = sum(PFGenActualP(:,m));
    PFGenActualTotalQ(m) = sum(PFGenActualQ(:,m));
    PFGenCapacityTotal(m) = sum(PFGenCapacityP(:,m));
    PFGenUtilization(m) =
(PFGenActualTotalP(m)/PFGenCapacityTotal(m))*100;
    PFLineFlowP(:,m) = max([abs(PFBranchFromInjectionP(:,m))
abs(PFBranchToInjectionP(:,m))]',[],1);
    PFLineUtilization(:,m) =
(PFLineFlowP(:,m)./PFBranchRating(:,m))*100;
    PFLineMaxUtilization(m) = max(PFLineUtilization(:,m));
    PFBranchMax(m) = find(PFLineUtilization(:,m) ==
PFLineMaxUtilization(m));
    LineMinUtilization = PFLineUtilization(:,m);
    LineMinUtilization(LineMinUtilization==0)=100;
    PFLineMinUtilization(m) = min(LineMinUtilization);
    PFBranchMin(m) = find(PFLineUtilization(:,m) ==
PFLineMinUtilization(m));
    PFLineFlowTotalP(m) = sum(PFLineFlowP(:,m));
    PFBranchDisabled{m} = mat2str(find(PFBranchStatus(:,m) == 0));
    if (isempty(find(PFBranchStatus(:,m) == 0) == 1))
        PFBranchDisabled{m} = 'N/A';
    end
    PFBusDisabled{m} =  mat2str(find(PFBusStatus(:,m) == 0));
    if (isempty(find(PFBusStatus(:,m) == 0) == 1))
        PFBusDisabled{m} = 'N/A';
    end
    PFGenDisabled{m} =  mat2str(PFGenBus(find(PFGenStatus(:,m) ==
0),m));
    if (isempty(find(PFGenStatus(:,m) == 0) == 1))
        PFGenDisabled{m} = 'N/A';
    end

    PFSuccess(m) = pfresults(m).success;

    PFTime(m) = pfresults(m).et;

    PFLossPercentage(m) =
(PFBranchLossesTotalP(m)/PFGenActualTotalP(m));

end

warning('on','all')

filename = 'PFscriptResults.xls';

xlswrite(filename, {'Test Case Number'}, 'A1:A1');
```

```
xlswrite(filename, [1:iterations]', sprintf('A2:A%i',iterations+1));
xlswrite(filename, {'PF/OPF'}, 'B1:B1');
xlswrite(filename, {'PF'}, sprintf('B2:B%i',iterations+1));
xlswrite(filename, {'Generation Actual (MW)'}, 'C1:C1');
xlswrite(filename, PFGenActualTotalP', sprintf('C2:C%i',iterations+1));
xlswrite(filename, {'Generation Capacity (MW)'}, 'D1:D1');
xlswrite(filename, PFGenCapacityTotal',
sprintf('D2:D%i',iterations+1));
xlswrite(filename, {'Generation Usage (%)'}, 'E1:E1');
xlswrite(filename, PFGenUtilization', sprintf('E2:E%i',iterations+1));
xlswrite(filename, {'Total Load (MW)'}, 'F1:F1');
xlswrite(filename, PFLoadTotalP', sprintf('F2:F%i',iterations+1));
xlswrite(filename, {'Total Losses (MW)'}, 'G1:G1');
xlswrite(filename, PFBranchLossesTotalP',
sprintf('G2:G%i',iterations+1));
xlswrite(filename, {'Total Loss Percentage'}, 'H1:H1');
xlswrite(filename, PFLossPercentage', sprintf('H2:H%i',iterations+1));
xlswrite(filename, {'Maximum Line Utilization (%)'}, 'I1:I1');
xlswrite(filename, PFLineMaxUtilization',
sprintf('I2:I%i',iterations+1));
xlswrite(filename, {'Maximum Utilization Branch'}, 'J1:J1');
xlswrite(filename, PFBranchMax', sprintf('J2:J%i',iterations+1));
xlswrite(filename, {'Minimum Line Utilization (%)'}, 'K1:K1');
xlswrite(filename, PFLineMinUtilization',
sprintf('K2:K%i',iterations+1));
xlswrite(filename, {'Minimum Utilization Branch'}, 'L1:L1');
xlswrite(filename, PFBranchMin', sprintf('L2:L%i',iterations+1));
xlswrite(filename, {'Branches out of service'}, 'M1:M1');
xlswrite(filename, PFBranchDisabled', sprintf('M2:M%i',iterations+1));
xlswrite(filename, {'Buses out of service'}, 'N1:N1');
xlswrite(filename, PFBusDisabled', sprintf('N2:N%i',iterations+1));
xlswrite(filename, {'Generators out of service'}, 'O1:O1');
xlswrite(filename, PFGenDisabled', sprintf('O2:O%i',iterations+1));
xlswrite(filename, {'Success'}, 'P1:P1');
xlswrite(filename, PFSuccess', sprintf('P2:P%i',iterations+1));
xlswrite(filename, {'Time'}, 'Q1:Q1');
xlswrite(filename, PFTime', sprintf('Q2:Q%i', iterations+1));
```

Appendix L - PF_Variablesystem

```
function [ mpc_new ] = PF_variablesystem( power_system,mpc_old, mpopt,
iteration )
%This function stores information for a certain number of itertions of
a
%power system for use in calculating varying power flows for the
system.
%This function is based on the MATPOWER pf results of the ieee 30 bus
test
%case and formatted to be used with MATPOWER functions
%Iteration count must start at zero to initialize the function


    %mpc_new = runpf('case30',mpopt);
    mpc_new = loadcase('case30');

    %initial branch status, 1 - in service, 0 - out of service
    mpc_new.branch(:,11) = power_system.branchstatus(:,iteration);

    mpc_new.bus(:,11) = power_system.busstatus(:,iteration);
    mpc_new.bus(:,3) =
power_system.busstatus(:,iteration).*mpc_new.bus(:,3);
    mpc_new.bus(:,4) =
power_system.busstatus(:,iteration).*mpc_new.bus(:,4);

    mpc_new.gen(:,8) = power_system.genstatus(:,iteration);
    mpc_new.gen(:,4) =
power_system.genstatus(:,iteration).*mpc_new.gen(:,4);
    mpc_new.gen(:,5) =
power_system.genstatus(:,iteration).*mpc_new.gen(:,5);
    mpc_new.gen(:,9) =
power_system.genstatus(:,iteration).*mpc_new.gen(:,9);
    mpc_new.gen(:,10) =
power_system.genstatus(:,iteration).*mpc_new.gen(:,10);

    if (iteration == 75)
        mpc_new = runpf('case30',mpopt);
    end

end
```