



Opportunistic Hybrid Communications Systems for Distributed PV Coordination

Jianhua Zhang,¹ S. M. Shafiul Alam,¹ Anthony Florita,¹
Adarsh Hasandka,¹ Jin Wei-Kocsis,² Dexin Wang,³
Liuqing Yang,³ and Bri-Mathias Hodge¹

¹ *National Renewable Energy Laboratory*

² *University of Akron*

³ *Colorado State University*

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5D00-73716
March 2020



Opportunistic Hybrid Communications Systems for Distributed PV Coordination

Jianhua Zhang,¹ S. M. Shafiul Alam,¹ Anthony Florita,¹
Adarsh Hasandka,¹ Jin Wei-Kocsis,² Dexin Wang,³
Liuqing Yang,³ and Bri-Mathias Hodge¹

¹ *National Renewable Energy Laboratory*

² *University of Akron*

³ *Colorado State University*

Suggested Citation

Zhang, Jianhua, S. M. Shafiul Alam, Anthony Florita, Adarsh Hasandka, Jin Wei-Kocsis, Dexin Wang, Liuqing Yang, and Bri-Mathias Hodge. 2020. *Opportunistic Hybrid Communications Systems for Distributed PV Coordination*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5D00-73716.
<https://www.nrel.gov/docs/fy20osti/73716.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report

NREL/TP-5D00-73716
March 2020

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Acknowledgments

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

Executive Summary

A full-scale, operational implementation of the opportunistic hybrid communications systems for distributed photovoltaic (PV) coordination was successfully developed, simulated, and validated in this 3-year project. The system is considered hybrid because it uses different communications pathways from the scale of residential PV inverters to a transmission network; primary technologies harnessed include Low-Power Wireless Personal Area Network (LoWPAN), Power Line Communication (PLC), WiFi mesh, Worldwide Interoperability for Microwave Access (WiMAX), Ethernet cable, and Optical Ethernet systems. It is opportunistic in that it chooses to route messages through each of these systems based on recent data about latency and availability to ensure reliable message passing.

The primary focus of Year 1 was the individual development of three main tasks: the communications system architecture, PV system state computation algorithms, and distributed state estimation algorithms. The fourth task of Year 1, technical review committee (TRC) outreach and feedback, was conducted as expected.

The primary focus of Year 2 was the individual development of two main tasks: robustify decentralized imputation and prediction algorithms and develop the communications system simulation model. The third task of Year 2, the TRC outreach and feedback, was conducted as expected.

The primary focus of Year 3 was the individual development and validation of two main tasks: develop the hardware-in-the-loop (HIL) test bed and validate the impact of hybrid communications design on the distribution system operation and stability; and develop the Hierarchical Engine for Large-Scale Infrastructure Co-Simulation (HELICS) and Network Simulator-3 (NS-3) co-simulation platform and validate the scalability of the hybrid designs on the transmission-distribution-communications co-simulation.

The project is meeting expectations in terms of scope, budget, and timing, as detailed in this report. A number of test cases were developed for architectural and algorithmic (middleware and optimization) development and testing: distribution- and transmission-level data from the Institute of Electrical and Electronics Engineers test cases served as inputs to both the opportunistic hybrid communications system and PV/distributed state estimation tasks, which required substantial testing and modification of the SunShot-sponsored Integrated Grid Modeling System project and; for the PV state estimation task, residential PV panels were driven by simulated, measurement-calibrated spatiotemporal solar data derived from the Solar Integration National Dataset and Hawaiian measured data. A suite of opportunistic hybrid communications system simulation models based on the NS-3 simulator was successfully developed on top of Reference Test Case A (RTC-A) for architectural and algorithmic (middleware and optimization) validation and evaluation: six simulation models of RTC-A that represent the developed communications systems featured with opportunistic and hybrid were solved intelligently. These required substantial development and validation of parameters and functionalities and, for the development task, the challenges of multiple alternative communications technologies, IPv6-to-IPv4 tunneling technology, and scalability issue.

Additionally, the communications system design architectures, parameters, and two intelligent network management mechanisms were intensively tested and validated. The hybrid communications network-based HIL was successfully developed, and both distributed energy resource (DER) monitoring and control applications are conducted. The NS-3 simulator was successfully integrated into the HELICS platform, and both DER monitoring and control in the transmission-distribution-communication co-simulations are conducted. In addition, the distributed ladder-iterative belief propagation-based bad-data-resistant distribution system state estimation algorithm was developed and implemented successfully. A distributed intelligent reinforcement learning-based attack-resilient middleware architecture is developed and validated in the developed Opal-RT and NS-3 co-simulation test bed. From a realistic physical environment perspective, the research shows the current gaps in knowledge on algorithm performance in terms of robustness, which can be filled with the development of robust distributed imputation and prediction algorithms, including PV/distribution system state estimations, to adapt to real-world conditions. The current state of the art is the design of opportunistic hybrid communications systems for a sensor-starved grid that was substantially validated in software simulation, co-simulation, and HIL testing to effectively provide bilateral information sharing to enhance efficiency, reliability, and integrate distributed PV at large scales. Future research will focus on continuing the HIL and co-simulation validation from the perspectives of middleware, algorithms, and system-to-system.

Table of Contents

Background.....	1
Communications Systems Architecture.....	1
Photovoltaic System State Computation.....	3
Distributed State Estimation	4
Project Objective.....	7
Introduction.....	9
Project Results and Discussion.....	10
Task 1.1: Communications System Architecture.....	11
Subtask 1.1.1: Information Mapping	12
Subtask 1.1.2: Opportunistic Hybrid Infrastructure.....	15
Subtask 1.1.3: Middleware Framework	18
Subtask 1.1.4: Cybersecurity and Mitigation.....	26
Task 1.2: Develop Photovoltaic System State Computation Algorithms.....	30
Subtask 1.2.1: Low-dimensional Visualization	30
Subtask 1.2.2: Kriged Kalman Filter	31
Subtask 1.2.3: Decentralized Optimization	35
Task 1.3: Develop Distributed State Estimation Algorithms.....	35
Subtask 1.3.1: Subarea Partitioning.....	35
Subtask 1.3.2: Distribution State Estimation Algorithms.....	38
Subtask 1.3.3: Bad Data Detection and Processing	44
Subtask 1.3.4: Incorporate Statistics of PV States.....	59
Task 1.4: Technical Review Committee.....	59
Subtask 1.4.1: Stakeholder Community	59
Task 2.1: Robustify Decentralized Imputation and Prediction Algorithms.....	59
Subtask 2.1.1: Develop Robust Machine Learning Algorithms for State Estimation	60
Subtask 2.1.2: Create a Data Screening Tool for Distributed State Estimation	64
Subtask 2.1.3: Develop Decentralized Solvers for the Optimization Problems Formulated Under Subtasks 1.2.1 and 1.2.2.....	66
Task 2.2: Develop Communications System Simulation Model.....	67
Subtask 2.2.2: Transmission Distribution Communication Co-Simulation.....	112
Subtask 2.2.3: Alternative Technology Comparison	115
Task 2.3: Technical Review Committee.....	119
Subtask 2.3.1: TRC Meeting.....	119

Subtask 2.3.2: GMLC Projects Meetings	121
Task 3.1: Hardware-in-the-Loop Case Study	123
Subtask 3.1.1: Setup of HIL Test bed	123
Subtask 3.1.2: Results and Analysis of HIL Test	132
Task 3.2: Computational Case Study	136
Subtask 3.2.1: Development of Different High Penetration Distributed PV Scenarios	136
Subtask 3.2.2: Results and Analysis of Co-Simulation	143
Publications	158
References	159

List of Figures

Figure 1. Integrated Grid Modeling System	10
Figure 2. Reference Test Case A topology	11
Figure 1.1. Core-edge structure of the utility communications system. A WAN router (edge router) is considered to be the gateway to the WAN.....	12
Figure 1.2. Data concentrators in the NAN	14
Figure 1.3. Common communications protocols for the power system	14
Figure 1.4. Communications model and information mapping of the HAN and NAN	15
Figure 1.5. Digraph representation	16
Figure 1.6. Digraph network representation	17
Figure 1.7. Structure of the proposed middleware architecture	21
Figure 1.8. Interaction between the proposed middleware instances, power application layers, and network infrastructure layers.....	21
Figure 1.9. Mechanism of the proposed universal middleware	22
Figure 1.10 Network topology used in the simulation.....	27
Figure 1.11. Throughput of Data Flow 2 without/with using our proposed middleware architecture.....	28
Figure 1.12. End-to-end latency of Data Flow 2 without/with using our proposed middleware architecture	29
Figure 1.13. End-to-end latency of Data Flow 2 is recovered by avoidance strategy in Case Study 2.	29
Figure 1.14. (a) GHI sensor locations at Kalaeloa Airport, Oahu, Hawaii, United States; (b) raw GHI at 1-second resolution.....	31
Figure 1.15. (a) Log of clear-sky index, (b) partial auto correlation coefficient	31
Figure 1.16. Dynamic estimation performance with 5-minute interval observation update	33
Figure 1.17. Dynamic estimation performance with 15-minute interval observation update	34
Figure 1.18. TBS-AR.....	36
Figure 1.19. MBS-AR.....	36
Figure 1.20. WMBS-AR.....	36
Figure 1.21. Manual regionalization Case 1	37
Figure 1.22. Manual regionalization Case 2	37
Figure 1.23. MSE comparisons Figure 1.24. Number of iterations comparison.....	38
Figure 1.25. Accuracy comparison in transmission system state estimation.....	39
Figure 1.26. Number of Gauss iterations vs. number of regions Figure 1.27. Execution time vs. number of regions	39
Figure 1.28. The Gauss-Newton algorithm diverges in RTC-A.	40
Figure 1.29. The high-level flowchart of our proposed DiLISE algorithm	41
Figure 1.30. AR with eight regions for RTC-A (nodes with the same marker and color belong to the same region).....	42
Figure 1.31. RMSE achieved by DiLISE with various number of regions compared to that achieved by the centralized solution	43
Figure 1.32. Execution time of DiLISE with various numbers of regions compared to that of the centralized solution	44
Figure 1.33. Scalability of DiLISE algorithm. The numbers of buses are positioned in log scale for better presentation.	44
Figure 1.34. Impact of bad data contamination level.....	45
Figure 1.35. Impact of bad data magnitude	46
Figure 1.36. Overall flowchart of LiGAN algorithm.....	47
Figure 1.37. Flowchart of forward and backward sweep.....	48
Figure 1.38. Algorithm comparison based on mean squared residual	48
Figure 1.39. An example of a distribution network for notation clarification	49
Figure 1.40. Original message-passing schedule for LIBP.....	49

Figure 1.41. Improved message-passing schedule for LIBP.....	50
Figure 1.42. IEEE 13-bus feeder topology	51
Figure 1.43. Comparisons of the RMSE at each bus using LIBP algorithm with the original and the improved message-passing schedule 1.....	51
Figure 1.44. Forney-style factor graph of the Bayesian network with load power measurements after local linearization.....	52
Figure 1.45. Message-passing schedule for LIBP with load power measurements incorporated.....	53
Figure 1.46. Comparison between the conventional LI algorithm and the LIBP algorithm.....	54
Figure 1.47. Performance of LIBP under various measurement configurations.....	54
Figure 1.48. RMSE comparisons of conventional LI and LIBP algorithm. $\sigma_S = 5\%$, $\sigma_V = 0.005\%$	56
Figure 1.49. RMSE comparisons of conventional LI and LIBP algorithm. $\sigma_S = 30\%$, $\sigma_V = 0.03\%$	57
Figure 1.50. Accuracy of centralized and distributed versions of conventional LI and LIBP algorithms. The results of the centralized algorithms are plotted at number of regions = 1.....	58
Figure 1.51. Execution time comparisons of centralized and distributed versions of conventional LI and LIBP algorithms. The results of the centralized algorithms are plotted at number of regions = 1.....	58
Figure 2.1. RTC-A collocated with 17 DeSoto, Florida, irradiance sensors. Stars indicate the NSRDB data points.....	60
Figure 2.2. Kriging-based wide-area PV footprint formation.....	61
Figure 2.3. (a) Example AC power output of PV inverter, (b) calculated solar power index.....	62
Figure 2.4. Fitting of exponential semivariogram model.....	62
Figure 2.5. RTC-A clustering into five regions: (a) PV footprint, (b) network admittance.....	63
Figure 2.6. Synchronous integration of static and dynamic state estimation.....	64
Figure 2.7. Basic flow of the distributed LIBP algorithm.....	65
Figure 2.8. Design framework of opportunistic hybrid communications systems.....	68
Figure 2.9. Alternative topologies for opportunistic hybrid communications systems	71
Figure 2.10. NS-3 simulation models of hybrid communications architectures.....	72
Figure 2.11. Screenshot of LoWPAN-WiFi network animation on top of RTC-A	75
Figure 2.12. RTC-A regional communications performance by assuming that the PV has a generating interval of 30 ms: (a) average latency, (b) maximum latency.....	78
Figure 2.13. RTC-A regional communications performance by assuming that the PV has a generating interval of 30 ms: (a) average packet availability, (b) minimum packet availability.....	79
Figure 2.14. RTC-A regional communications performance by considering different generating intervals for PV: (a) average latency, (b) maximum latency	80
Figure 2.15. RTC-A regional communications performance by considering different generating intervals for PV: (a) average packet availability, (b) minimum packet availability	81
Figure 2.16. Middleware-based proactive networking management mechanism.....	82
Figure 2.17. Vertical structure of proposed opportunistic communications systems	83
Figure 2.18. NS-3 simulation models of opportunistic hybrid communications architectures.....	84
Figure 2.19. Implementation structure of master middleware in NS-3.....	85
Figure 2.20. Implementation structure of slave middleware in NS-3.....	86
Figure 2.21. RTC-A regional hybrid communications performance by considering: (a) PLC-CSMA hybrid link, (b) PLC-WiFi hybrid link, (c) PLC-WiMAX hybrid link	88
Figure 2.22. RTC-A regional hybrid communications performance of end-to-end by considering (a) PLC-WiFi hybrid link and (b) LoWPAN-WiFi hybrid link; and of throughput by considering (c) PLC-WiFi hybrid link and (d) LoWPAN-WiFi hybrid link	89
Figure 2.23. Proposed deep CNN-DCNN-based cyberattack detection scheme	90
Figure 2.24. End-to-end latency: (a) without middleware, (b) with middleware; throughput: (c) without middleware, (d) with middleware	92
Figure 2.25. End-to-end latency: (a) without middleware, (b) with middleware; throughput: (c) without middleware, (d) with middleware	92

Figure 2.26	Illustration of improved middleware functionality along with the hybrid model.....	93
Figure 2.27.	Illustration of the improvement in middleware mechanism	94
Figure 2.28	(a) End-to-end latency versus packet sizes with and without failure recovery and (b) throughput versus packet sizes with and without failure recovery	95
Figure 2.29.	End-to-end latency (a) when there is no control mechanism on network layer and (b) when there is control mechanism on network layer.....	95
Figure 2.30.	Structure for improved middleware architecture with RL-based real-time opportunistic control scheme	96
Figure 2.31.	High-level illustration of alternative multi-controller SDN-based communications system that was developed to realize RL-based real-time decision-making mechanism.....	97
Figure 2.32.	Illustration of deep actor-critic RL-based real-time opportunistic control scheme for middleware mechanism.....	98
Figure 2.33.	Network topology for experimental evaluation.....	98
Figure 2.34.	Integrated QoS information presented via reward value achieved: (a) without congestion attacks and (b) with congestion attacks.....	99
Figure 2.35.	Illustration of A3C RL-based hierarchical real-time opportunistic control mechanism.....	99
Figure 2.36.	Integrated QoS information presented via reward value achieved: (a) with congestion-attack Scenario 1 and (b) with congestion-attack Scenario 2	100
Figure 2.37.	Integrated QoS information presented via reward value	101
Figure 2.38.	High-Level framework block diagram	103
Figure 2.39.	Comparison of hybrid algorithm in terms of test functions.....	110
Figure 2.40.	Design diagram of HELICS-NS-3 integration	113
Figure 2.41.	Prototype diagram of HELICS-NS-3 integration on top of RTC-A.....	114
Figure 2.42.	Latency performance of PV nodes of RTC-A in NS-3-HELICS co-simulation in case of LoWPAN-Ethernet hybrid network	115
Figure 2.43.	Performance characterization of six hybrid communications designs in terms of UDP packet size: (a) average latency, (b) average throughput, (c) average packet loss rate	118
Figure 2.44.	Performance characterization of six hybrid communications designs in terms of UDP transmission rate: (a) average latency, (b) average throughput, (c) average packet loss rate	119
Figure 3.1.	HIL test objectives.....	123
Figure 3.2.	Schematic of HIL test system.....	125
Figure 3.3.	Physical diagram of HIL test system.....	126
Figure 3.4.	Layout of HIL test system at NREL's ESIF	127
Figure 3.5.	Smart meter with Zigbee/PLC/WiFi/Ethernet.....	127
Figure 3.6.	PV Inverter with Zigbee/PLC.....	128
Figure 3.7.	Communications hardware connection diagram	128
Figure 3.8.	Communications hardware network diagram.....	129
Figure 3.9.	Topologies of each phase of RTC-A according to the ePHASORSIM model.....	132
Figure 3.10.	Volt-VAR control algorithm	133
Figure 3.11.	Flowchart of PV inverter/smart meter monitoring and Volt-VAR control at data concentrator	133
Figure 3.12.	Flowchart of smart meter monitoring and PV inverter data transfer function at smart meter	134
Figure 3.13.	Flowchart of monitoring and control function at PV inverter	134
Figure 3.14.	Locations PV inverters (yellow, orange, and sienna circles) for 10%, 40%, and 100% PV penetrations. The edge router, data concentrators, and smart meters are denoted as red, green, and yellow circles, respectively. The polygons show the coverage of 10 individual data concentrators.	138
Figure 3.15.	Impact of PV penetration rate for LoWPAN-WiFi and LoWPAN-Ethernet.....	141
Figure 3.16.	Performance at 100% PV penetration: a) latency, b) packet loss rate.....	142

Figure 3.17. House model simulation results.....	144
Figure 3.18. Office model simulation results.....	144
Figure 3.19. Illustration of MATLAB-NS-3 co-simulation test bed	145
Figure 3.20. Impact of co-simulation on end-to-end latency	146
Figure 3.21. Illustration of application simulation mechanism.....	146
Figure 3.22. Illustration of setting up in co-simulation.....	147
Figure 3.23. Co-simulation evaluation: (a) average throughput and (b) average latency	147
Figure 3.24. Diagram of GridLAB-D and NS-3 co-simulation through HELICS	148
Figure 3.25. Installation configuration commands for HELICS, GridLAB-D, and NS-3	149
Figure 3.26. Configuration files for the GridLAB-D simulator.....	150
Figure 3.27. Screenshot of running HELICS broker.....	151
Figure 3.28. Screenshot of running the GridLAB-D federate.....	151
Figure 3.29. Screenshot of running the NS-3 federate.....	152
Figure 3.30. Average latency performance of each NAN at the HELICS co-simulation	153
Figure 3.31. Packet loss rate performance of each NAN at the HELICS co-simulation	153
Figure 3.32. Two single solar power systems in GridLAB-D	154
Figure 3.33 Implementation architecture of GridLAB-D + Python-based controller.....	155
Figure 3.34. Controller results without considering communications issues.....	155
Figure 3.35. Implementation architecture of GridLAB-D+NS-3+Python-based-controller.....	156

List of Tables

Table 1.1. Alternative Communications Technologies for Distributed PV Coordination Communications System.....	15
Table 1.2. Information Provided by QoS Database	23
Table 1.3. Parameters of Observed QoS Information.....	24
Table 1.4. Control Algorithm with Local QoS Alert.....	26
Table 1.5. QoS Requirements of Simulated Data Flow Service	28
Table 1.6. Measurement Configurations for IEEE 13-Bus Feeder	51
Table 2.1. Implementation Environment	64
Table 2.2. Hybrid Communications Architectures	69
Table 2.3. Basic Configuration and Parameter Verification of PV-SM Links	77
Table 2.4. Basic Configuration and Parameter Verification of SM-DC Links.....	77
Table 2.5. Modeling of the QoE Evaluation Process.....	86
Table 2.6. Comparison of Different Optimization Methods	108
Table 2.7. Configurable Parameters of the BPLC-WiMAX Hybrid Design	111
Table 2.8. Optimal Parameters of the BPLC-WiMAX Hybrid Design	111
Table 2.9. Optimal Results for Simulated Configuration	112
Table 2.10. Results of Sampling Simulated Configuration.....	112
Table 3.1. Mapping of Scenarios of Metrical and Functional Requirements	125
Table 3.2. Results of Monitoring Function at HIL Testing.....	135
Table 3.3. Type and Number of Communications Nodes in RTC-A.....	138
Table 3.4. Hybrid Communications Architectures	139
Table 3.5. Simulation Configurations	139
Table 3.6. Results of Impact of PV Penetration Rate	140
Table 3.7. Capacity Limitation Results of Hybrid Designs	142

Background

The primary focus of this project was the development of three main tasks: (1) communications systems architecture, (2) photovoltaic (PV) system state computation, and (3) distributed state estimation. A discussion on up-to-date references is provided as follows. Each reference is tied to the project on the technical development of the opportunistic hybrid communications systems for distributed PV coordination.

Communications Systems Architecture

***Context:** Developing the opportunistic hybrid communications system architecture requires the relay of information developed in subsequent tasks to and from distributed PV generators to distribution and transmission system operators. Because adding dedicated communications hardware to each PV generator would be extremely capital cost-intensive, we use existing communications hardware as much as possible with intelligent routing algorithms to provide the information needed at the appropriate latency, bandwidth, and cybersecurity protection level. This enables the use of existing communications infrastructure in a reliable and efficient way to provide the necessary information from millions of distributed PV generators to the areas of the electric power grid where that information is required in a scalable, reliable, cost-effective, and secure manner. The developed architecture and algorithms provide utilities with visibility into and control over distributed PV penetration. The developed hybrid architecture is designed to minimize the amount of new communications infrastructure needed to make distributed PV monitoring and control possible. The opportunistic algorithmic framework also uses existing communications hardware that is currently underused to further reduce capital costs.*

Both wired and wireless communications play an important role in the blend of communications technologies considered for enabling the vision of a smart grid communications network. Different from mono-technology simulation models, however, for which only a onefold communications model is applied in the simulation networks, almost no attention is paid to the development of hybrid communications models to simulate the envisioned blended smart grid communications networks. In particular, hybrid simulation models, which emulate IPv6-to-IPv4 tunneling technology, are needed to capture the mixed Internet Protocol (IP) address mechanisms in a communications network simulation. To this end, we aim to close this gap and present a suite of hybrid communications system simulation models using the Network Simulator-3 (NS-3) to verify the critical system design criteria of smart grid communications networks. Further, extensive hardware-in-the-loop (HIL) and computational validation of the proposed communications system was conducted to detail the strengths and limitations of the proposed communications system framework.

The literature and our approach: Distributed energy resources (DERs) are being increasingly accepted in smart grids as an excellent complement to traditional energy sources. Because most of these generators are geographically dispersed, dedicated communications investments for every generator are capital cost-prohibitive. Combining real-time distributed communications middleware with Internet of Things (IoT) technologies allows for the use of existing communications infrastructure. The IoTs provide emerging real-time data acquisition solutions via the use of various networked measurement devices, and the communications middleware supervises, organizes, and schedules tremendous amounts of data traffic in smart grids with high penetrations of DERs.

Traditionally, power generation comes from centralized power stations, and power consumers cannot produce electricity by themselves. As DER technologies become increasingly popular, power resources also tend to be scattered [1]. DERs, such as solar PV panels and wind turbines, are accepted by an increasing number of power consumers because these resources are able to generate electricity by themselves and even sell surplus electricity back to the grid. It is necessary to design efficient data acquisition systems and communications infrastructures to maintain power system economic efficiency and reliability. IoT technologies provide effective solutions to address these challenges. Instead of investing in new communications infrastructures, IoT uses existing networks such as the Internet, wireless sensor network, and mobile communications network to transfer sensing data from DERs to locations in the electricity system where this information can be effectively used [2].

Current IoT networks, however, such as the Internet, cannot fully meet the quality-of-service (QoS) requirements of critical smart grid application services [3]. Further, the use of IoT techniques significantly increases the complexity of Internet connections and thus reduces the effectiveness of the conventional Internet management in detecting unusual network events. Therefore, communications networks are more vulnerable to potential cyberattacks, such as denial-of-service (DoS) attacks, which could potentially impede the progress of DER expansion [4]. This essential challenge inspires the development of autonomous network systems with adaptive control solutions. Considering the fact that not all power operators are familiar with network programming and configuration [5], middleware has emerged to abstract network topology and infrastructure, which allows end users and network programmers to configure the whole network by calling only local middleware objects attributes and methods [6]. GridStat, an objected-oriented-broker middleware providing QoS information, was designed for smart grid applications [7]. A middleware structure was proposed in [8] to optimize a Transmission Control Protocol (TCP) congestion window of gateway in NS-3. Zaballos et al. studied intelligently managing the interaction and adaptation between the ubiquitous sensor network and next-generation network at the link and the physical layers [9]. For middleware implementation, some work has been done by using the co-simulation of VTB or network controller and OPNET [10], [11]. Most middleware architectures do not consider security issues. From the security perspective, Mostafa et al. presented and verified a scheme using key cryptography to enhance the security of peer-to-peer communications. Kim et al. introduced a middleware architecture using encryption methods to enhance smart grid security [12]. These cryptographic approaches focus on preventing unauthorized users from accessing the network infrastructure, but they are not effective against inside attacks. To address this problem, we propose a real-time attack-resilient distributed middleware architecture. In this paper, we focus on DoS attacks, which are one of the most common network attacks [13]. Further, we implement our proposed middleware infrastructure in a co-simulation environment of NS-3 [14] and a commercial software package (MATLAB 9.0.0.341360, The MathWorks Inc., Natick, MA, 2016).

During the last few years, many studies have been dedicated to communications network architectures for coordinating renewables in the smart grid, and they fall into the following three major categories. First, some investigated generic communications network requirements and design principles, such as communications network requirements for smart grid applications [15], communications network architecture and design principles for the smart grid [16], and wireless communications architecture for smart grid distribution networks [17]. Second, some focused on the communications network for large-scale renewable energy resources.

Specifically, the communications systems used in real renewable energy resources, research challenges, and possible solutions of the communications systems for grid integration renewable energy resources are presented in [18]. Third, regarding communications networks for DERs, hybrid network architectures using both wireless and dedicated wired mediums for monitoring and controlling the distributed small-scale energy systems were proposed in [19]. Although useful insights have been provided in these studies, the existing results cannot be directly extended and applied to the practical communications system design and deployment for coordination of high penetrations of distributed PV resources. A hybrid network architecture for the integration of renewables was proposed in some of these papers; however, almost no attention has been paid to the development of hybrid communications architecture simulation models to verify the critical system design criteria.

To this end, we developed multiple hybrid communications system simulation models using the discrete-event NS-3 for distributed PV coordination. Our software models use the NS-3 library because of its popularity and already available models for numerous networking functionalities instead of other simulation tools, such as the Optimized Network Engineering Tool (OPNET) modeler and LabVIEW. The envisioned communications network comprises home area networks (HANs) and neighborhood area networks (NANs). In a HAN, the PV panel connects to a smart meter through two alternative communications technologies: low-power wireless personal area networks (LoWPANs) and Power Line Communication (PLC). Within a NAN, the data are transmitted from the smart meter to the data concentrator, and they eventually arrive at the wide-area network (WAN) through the WAN edge router along with three alternative technologies: WiFi, Ethernet cable, and WiMAX. With full combination of these five optional technologies, we considered six possible hybrid architectures and developed six corresponding hybrid prototype simulation models. The main challenge of developing NS-3-based hybrid communications simulation models is to integrate different communications technologies and IP address mechanisms into one simulation network. To address this challenge, we specifically designed a NetRouter forwarding function in the application level.

Photovoltaic System State Computation

***Context:** Currently, system operators have very little information about how much generation they are receiving from distributed PV generators. Having each generator send this information to the system operator would lead to a deluge of information and extreme communications bandwidth requirements. Machine learning algorithms can be applied to effectively estimate the state of large numbers of distributed PV systems from a much smaller number of sample measurements. Because these algorithms need to function in the real world, with lost and corrupted data, the developed algorithms are robust to missing or misleading data. We provide a suite of intelligent algorithms that fully exploit pervasive sensing and control capabilities to enable not only dimensionality reduction but also reconstruction, inference, prediction, imputation, and clustering of depleted and heterogeneous power data. The development of these algorithms will provide system operators at multiple levels with increased visibility into the current state of distributed PV generation. Because there will soon be millions of individual distributed PV systems, it would be cost-prohibitive to capture measurement data from each location. The development of these reduced-order state estimation techniques will allow for the accurate estimation of the aggregated distributed PV power output at any time using measurements from only a small subset of the total number of generators.*

The literature and our approach: The ever-increasing penetration of cost-effective PV panels within the distribution grid requires a robust and efficient method for PV system monitoring. In particular, the geographic proximity of PV panels can play an important role in reducing the dimension of measurements required for full observability of the system. Further, the direct impact of intermittent clouds necessitates the development and validation of a spatiotemporal dynamical model that captures the relation between PV-related spatial measurements and time-varying states. In addition to this spatial variability, the historical data-driven approach led to modeling the temporal evolution of the PV states. The spatiotemporal model thus obtained can be applied to a Multi-Rate and Event DRIVEN Kalman Kriging (MREDRIKK) filter to dynamically estimate PV system states. The Kriging step exploits the spatial correlation to estimate PV states at locations from where measurements are not being obtained. The multi-rate and event-driven feature of the MREDRIKK filter represents the sampling of measurements at a rate much lower than the temporal dynamics of the PV states. Accordingly, a recursive and scalable dynamic estimation of the PV footprint can be obtained for a spatially correlated neighborhood of PV mounted end users.

Distributed State Estimation

Context: The information about distributed PV system states gained in the previous task to update the power system state estimation methods for large-scale distribution and transmission systems must be extended to the power system. Because these are enormously complex problems, we divide the problem into regions of interest and then stitch the individual solutions back together to provide a picture of the overall system state. The proposed distributed state estimation (DSE) algorithm ensures the scalability, responsiveness, and robustness of the state estimation by leveraging the novel publish-subscribe pattern-based communications network architecture; henceforth, this will enable effective control of the power system as distributed PV generation rates increase. This allows the DSE to be applied to the larger scale needed to consider potentially millions of distributed PV generators. The disaggregation of the state estimation problem into many subproblems is a critical step that allows for the PV system state estimation to provide distributed PV information at the level of independent system operators (ISOs). Making the developed algorithms robust against missing or intentionally corrupted data is critical for its real-world application to ensure accurate operation and resilience to attacks.

The literature and our approach: State estimation is an essential tool for power system monitoring and analysis. Its results, the bus voltage magnitudes and phase angles are critical inputs for other operational tools, such as contingency analysis, optimal power flow, and economic dispatch [20], [21]. Traditionally, the state estimation algorithm takes the supervisory control and data acquisition (SCADA) measurements from substations and power plants as inputs and solves a weighted least-squares (WLS) problem at a central processor. With the emergence and rapid development of power generation from renewable energy sources, the faster dynamics and the increased uncertainty of the availability of these energy sources will require that state estimation is performed more frequently. The time interval between consecutive state estimation updates needs to be dramatically reduced from 2–4 minutes to about 1 second. A single central processor is not capable of handling the tremendous computational load required by such state estimation tasks. In addition to the aspect of processing power shortage, the communications load increases because of the increased frequency of the state estimation.

With the installation of phasor measurement units (PMUs) in the smart grid, measurement data are becoming available at a much higher rate. A centralized structure would impose enormous load on the communications network, and delays are inevitable. DSE has attracted significant attention as a promising solution to this challenge (see, e.g., [22]–[25]). In DSE, the entire system is first partitioned into smaller regions. This process is called regionalization. It is a necessary step before DSE can take place. With DSE, the computational load is distributed to multiple regional processors. The regional processors cooperate to estimate the state of the system. Each has access to local measurements within the region. In the literature, there are two types of DSE algorithms: in one, each regional processor maintains a copy of the state vector of the entire system; in the other, each regional processor maintains only a copy of the state vector of local buses. Both types require some information exchange between neighboring regional processors to achieve consensus among all processors. From our investigation, we observe that regionalization plays a crucial role in the performance of the DSE algorithms. In all the existing work on DSE, however, regionalization is performed manually, and the DSE performance is typically evaluated with only a single arbitrary regionalization instance. Further, manual regionalization is inefficient and tedious for large systems.

The main motivation for DSE is to improve the scalability of the state estimation algorithms to cope with large systems with hundreds or even thousands of buses. In such systems, the number of regions (regional processors) needs to be large to exploit more computational power so the processing time remains acceptable. Hence, manual regionalization becomes difficult and insufficient, if not impossible. Therefore, in this report, we investigate automatic regionalization for DSE. Inspired by [26], in which spectral clustering is employed for geographic regionalization, we propose applying spectral clustering to power system regionalization. Our contributions include the following: (1) To the best of the authors' knowledge, this is the first investigation on automatic regionalization algorithms for power system DSE. (2) We propose three regionalization algorithms based on spectral clustering with different similarity measures. (3) We investigate the impact of the number of regions on the DSE accuracy and convergence speed through simulations.

The objective of the state estimation is to get the best estimate of the current system states given a set of measurements and network parameters. Thus, the performance of an estimator is highly dependent on the accuracy of the measurements and the assumed estimation model. All measurements are subject to random noise and the discrepancy of modeling parameters, however, and the true parameters of the power system always exists. Besides these sources of relatively small errors, bad data because of instrument failures, impulsive communication noise, measurement time skewness, and cyberattacks exist in today's power system and might pose an increased threat to the security and reliability of the power system [27].

Bad data are a problem in power system state estimation, and it has been extensively studied since the 1970s. State estimation algorithms with bad data suppression capabilities were first proposed in [28] by replacing the quadratic cost function in the WLS estimator by a cost function that weighs the large errors less severely. This type of method to combat the influence of bad data was later extended to many different cost functions based on similar ideas [29]. Another approach for bad data processing is based on residual analysis, which is also discussed in [29]. This type of method attempts to identify and fix/remove the bad data from the measurements by observing the residuals generated by the state estimates. Many of these methods, however, suffer

with the high complexity caused by the iterative search for bad data-contaminated measurements; hence, they are not suitable for real-time online state estimation. More recent studies in this area focused on the integration of PMUs [30], distributed processing [31], and exploiting the temporal correlation of the measurements [32]. In [31], 1-1 norm-regulated least-squares optimization is used to enhance the robustness of WLS-based state estimation, and the alternating direction method of multipliers (ADMM) is employed to transform the state estimation problem so that it can be solved in a distributed manner by multiple processors. This is quite similar to the problem we face in this project; however, the methods based on cost functions with bad data suppression capability and the one proposed in [31] exhibit a poor convergence rate when implemented in a distributed way and applied to the distribution system in our Reference Test Case A (RTC-A). Therefore, in this project, we focus on developing a working distributed algorithm with bad data suppression capability for RTC-A.

Project Objective

Objective 1: Develop a novel publish-subscribe pattern-based communications network to better suit the needs of the monitoring and control of distributed PV generators.

Publish-subscribe networks offer a number of benefits for distributed PV systems, including better scalability, than traditional client-server operation and a loose coupling that allows for flexibility with changing system topology. The content of the events published by distributed generators should follow a standardized format to guarantee interoperability. The Common Information Model (CIM) is a standard developed by the electric power industry and adopted by the International Electrotechnical Commission (IEC) that facilitates information exchange in electrical networks. CIM defines an object-oriented model that contains a complex class structure and allows for information exchange among different sectors of the electricity system, including energy management systems, distribution management systems, and energy markets. CIM's object-oriented approach accommodates the addition of new components, which are necessary in the ever-evolving electricity landscape. Fortunately, the hierarchical structure of CIM can also facilitate the design of content filter aggregation algorithms in the publish-subscribe network.

Objective 2: Develop transformative, novel, decentralized state estimation and prediction schemes with dimensionality reduction algorithms that are resilient to measurement outliers and missing measurements.

State estimation and prediction schemes are a critical component of the proposed communications framework because they can greatly reduce the number of measurements (and hence communications) necessary as well as provide estimates where no data are currently available. Big data analytics become necessary only if the prediction algorithm demands do not scale well with the ability to efficiently store, transfer, and access large amounts of PV and network data. The use of robust machine learning algorithms will enable large-scale application of the state estimation techniques with computational bottlenecks minimized. It is not only the scale of the data that can give rise to formidable computational complexity but also faster-than-real-time computation, which is critically important for enhancing the value of prediction. Hence, we also pursue a sparse system representation as well as distributed algorithms to facilitate real-time execution of the information exchange.

Objective 3: Rigorously validate the communications systems developed through HIL testing and communications systems coupled with integrated distribution-transmission grid joint simulation.

A two-pronged validation approach was planned for the proposed communications system to ensure that it is ready for pilot-scale demonstration and deployment. First, HIL testing ensures that the system architecture, middleware, and algorithmic layers can operate with the necessary speed to fulfill system response time goals for both normal and contingency event operational states. This testing also ensures that the interoperability design goals are met through the physical connection with real PV inverters. The second approach of system testing is through a Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) platform developed at NREL for combined distribution-communication-controller system simulation. This testing ensures that the scalability goals are met and provides a platform for certifying not only that the

communications systems functions as designed but also tests implications of communications system behavior on the physical power system.

Introduction

Increased distributed PV penetration is causing a shift in the electricity system operations paradigm, with this new generation currently having no communications links to distribution- and transmission-level system operators, and thus it not visible or controllable. If this generation is to make up larger amounts of the total generation fleet, communications systems must be designed that allow for two-way communications with distributed generation to maintain system economic efficiency and reliability. Because most of these generators are relatively small, dedicated communications investments for every generator are capital cost-prohibitive. By combining intelligent communications middleware with distributed state estimation techniques and a unique communications system architecture that allows for the use of existing infrastructure, actionable information about distributed PV generation can be used in multiple levels of the power system. The next-generation smart grid will function by combining a range of real-time data sources to make intelligent decisions in the face of uncertainty. The design of an opportunistic hybrid communications system begins with existing electric grid technologies and harness them to their fullest to provide enhanced, bilateral information sharing—including investigations on how and what additional equipment could be installed to bring more value to the smart grid and the development of a suite of machine learning-based algorithms for PV system state computation and distributed state estimation in Phase 1. Further, in Phase 2, the research focuses on two investigations to approach such an expected opportunistic hybrid communications system to distributed PV coordination: (1) the development of communications system simulation models along with opportunistic and hybrid features for the validation and evaluation of the developed communication systems of Phase 1, and (2) the robustness of decentralized imputation and prediction algorithms for both PV and power system state estimation to adapt to real-world applications by incorporating methods for handling missing and corrupted data of a real system. Finally, in Phase 3, validation on both the HIL test bed and HELICS co-simulation platform are conducted. The following sections provide a complete overview of the Opportunistic Hybrid Communications Systems for Distributed PV Coordination Project.

Project Results and Discussion

The following task breakdown provides insight into the results and discussion of the Opportunistic Hybrid Communications Systems for Distributed PV Coordination Project within the 3 years. Because we are on track in terms of scope, budget, and time, approximately 100% of the work is complete at the time of this report writing.

Reference Test Case A (RTC-A)

Reference Test Case A was designed for model development and validation. This test case was designed by taking the IEEE 118-bus test system to model a large transmission system and attaching multiple feeders based on typical network taxonomies to the buses of the transmission grid. The 24 prototypical taxonomy feeders used were constructed under funding from the U.S. Department of Energy’s Modern Grid Initiative. The full system consists of 384 test feeders attached to the 118 buses of the transmission system. The Integrated Grid Modeling System (IGMS) developed at NREL couples the transmission system and distribution systems using MATPOWER, FESTIV to simulate the transmission system, and GridLAB-D to simulate the distribution system [33]. FESTIV simulates a security-constrained unit commitment day-ahead market and real-time market for the transmission system as input for an AC optimal power flow model to determine the power dispatched voltages at each transmission node. These are provided as set points for the feeder heads, which are connected to the transmission buses. Each feeder runs a separate instance of GridLAB-D to simulate power usage on the distribution network and feeds load data up to the transmission system to simulate the next real-time market dispatch. This process is outlined in Figure 1.

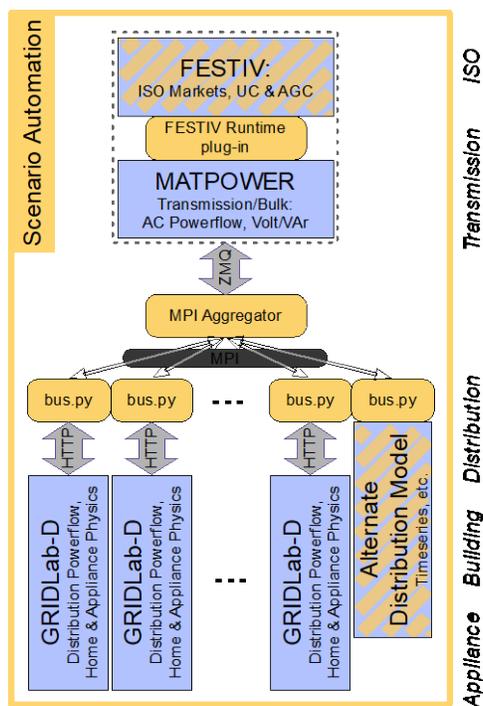


Figure 1. Integrated Grid Modeling System

For Reference Test Case A, the taxonomy feeder titled R2-25.00-1 containing 1,080 nodes was selected from the full system for model testing [34]. This feeder is shown in Figure 2. Penetrations of 5%, 10%, and 15% solar uptake are modeled on this feeder. The location and availability of existing communications infrastructure is modeled using data from SMUD, which has rolled out smart metering across their utility network and uses the Silver Spring Networks for data communications infrastructure. Typical smart meter installation rates and placement of data concentrators, which were built for current smart meter communication requirements, are scaled to the R2-25.00-1 feeder. Link parameters are taken from IEEE standards for each communications channel. The performance of the nodes and links along the network are achieved from network simulations performed using NS-3 to provide a distribution of latency, packet loss, jitter, bandwidth, and failure rate of the communications equipment, as described in the following sections.

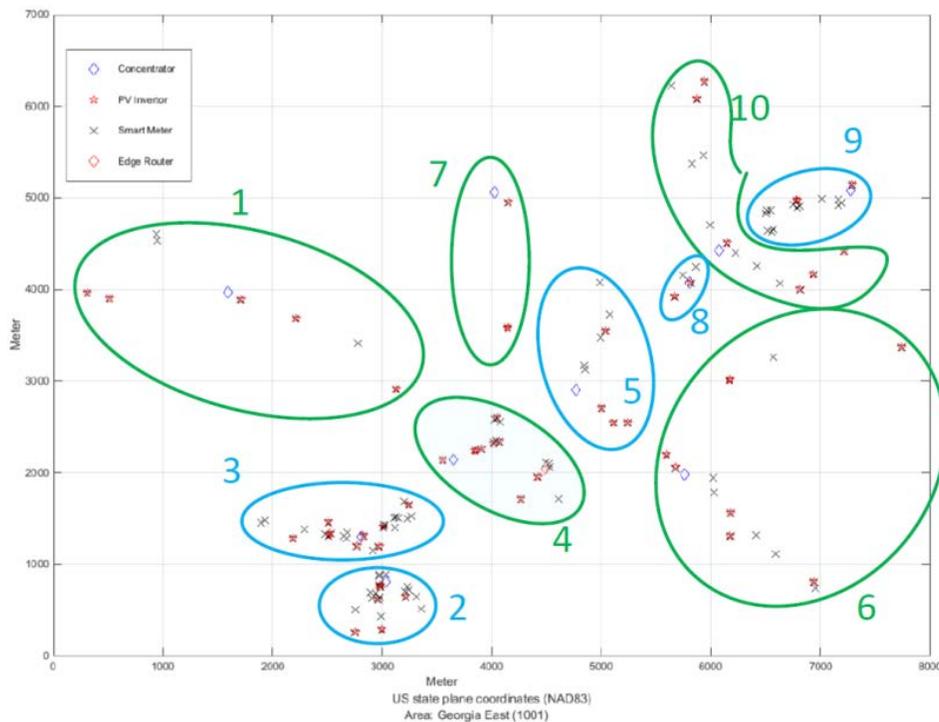


Figure 2. Reference Test Case A topology

Task 1.1: Communications System Architecture

This task develops a novel communications system architecture that enables the reliable and efficient real-time monitoring, control, and optimization of large-scale transmission-distribution power systems with millions of distributed PV generators. It leverages the existing communication infrastructure to achieve a novel opportunistic hybrid communications architecture with a real-time publish-subscribe messaging pattern. The architecture uses intelligent message scheduling and prioritization algorithms and robust channel allocation schemes to address scalability concerns without sacrificing the response time and ensure the in-time delivery of high-fidelity information, even when experiencing localized availability issues. The architecture of the existing communications systems is “designed around,” with a cost-minimal optimization of additional infrastructure, so that the information produced by the state

estimation algorithms can be directed through the middleware framework to provide intelligence on distributed PV to different portions of the grid as needed.

Subtask 1.1.1: Information Mapping

Objective: Create information mapping of timescale, priority, latency, and bandwidth requirements for each message needed from measurements to usage in the combined distribution-transmission system. Specific use cases include transmission system operator visibility and control during both steady-state and contingency operations and distribution system operator system state detection. Identify those critical messages from/to distributed PV systems to ensure the reliable operation of the transmission and distribution power systems. Accomplish this work by examining the topology and taxonomy of existing distribution-transmission systems. Examine their coupling to determine what, where, and when information from distributed PV would provide benefits. Place particular emphasis on harnessing existing systems to provide enhanced communications with the minimal addition of equipment.

The objective of this task is to develop an appropriate network architecture model for the proposed communications system with emphasis on using open-source and/or standardized protocols and existing network infrastructure as much as possible.

In modern power systems, utility communications systems are designed and deployed in the form of a core-edge network (Figure 1.1) [35]. This way, a wide-area network (WAN), usually based on fiber optics, forms the backbone of the system, whereas the connections between end devices and the WAN are established through neighborhood area networks (NAN) and field area networks (FAN). Hence, it is not common for an individual end device to be directly connected to the control center local area network (LAN), i.e., all connections must eventually go through the WAN. As such, the focus in this task was narrowed to designing the communications network that enables data transmission between the individual PV panels and the first WAN router (edge router).

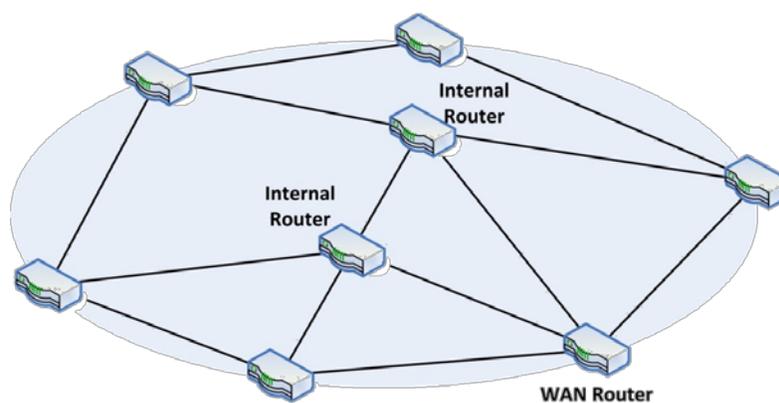


Figure 1.1. Core-edge structure of the utility communications system. A WAN router (edge router) is considered to be the gateway to the WAN.

Typically, this communications network would be hierarchical and would consist of:

- A home area network (HAN) that connects the PV panel to the smart meter located at the home's gateway to the utility's network. The geographic size of a HAN would be up to tens of meters.
- A NAN that collects the data from multiple smart meters and transmits it to the WAN through a WAN router. The geographic size of this network depends on the topology of the distribution system and the distances between the houses, and it could range from hundreds of meters to several kilometers.

Note that, similar to the case with smart meters, the data collected from individual PV panels is usually not transmitted to the control center as individual data points. Instead, the common practice is to aggregate individual measurements into a combined measurement using data concentrators (Figure 1.2). The data concentrators would then transmit this aggregated data to the WAN router. The number of concentrators in a region would depend on the number of houses, the rate at which data are being transmitted, the geographic size of the area, and the availability of physical locations for locating the concentrators, among other things. The latter, in particular, can be a limiting factor because the location must be physically secure, owned by the utility, and preferably equipped with remote monitoring capabilities. Some potential locations are the distribution substations, SCADA control rooms, utility-owned buildings, or buildings owned by private or public companies that are leased by the utility. If more than one concentrator covers the PV panels in an area, their topology could be either hierarchical (where multiple concentrators would communicate to another concentrator at a higher level) or horizontal (where each concentrator is directly connected to the WAN through a WAN router). In either case, without loss of generality, it is assumed that communications links exist that allow the concentrators to transmit the data to the WAN.

To decide the appropriate communications technology, the communications requirements of the data flows are assessed in terms of reliability and QoS. It is expected that the data are generated and transmitted at a second level from individual PV panels. These data could consist of solar irradiance data, voltage phasor at the panel, and the current phasor injected by the panel. Depending on the PV panel control scheme (i.e., a single DC/AC conversion stage versus a combined DC/DC regulation stage followed by a DC/AC conversion stage), it is possible to add more information to the payload, e.g., DC link voltage at the output of the DC/DC converter. With a payload limited to a few bytes in size, most major protocols used today can manage the bandwidth requirements for a one second data flow.

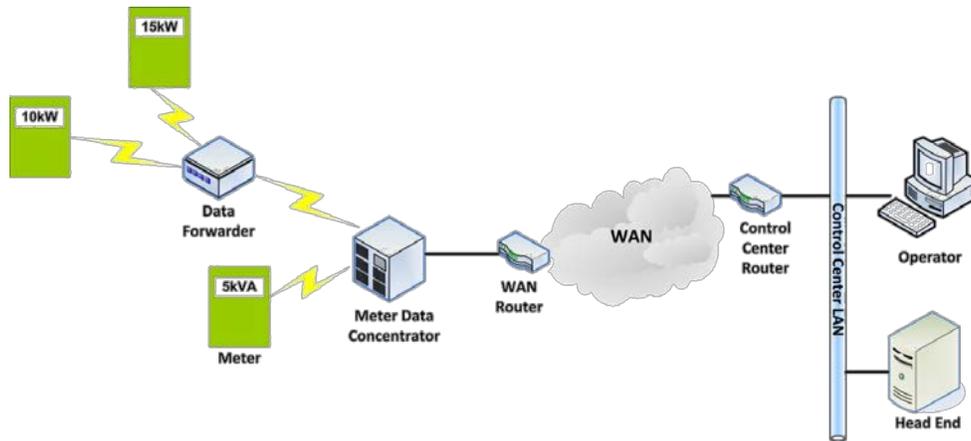


Figure 1.2. Data concentrators in the NAN

Various communications protocols and technologies exist that can be used for data communications in power systems (see Figure 1.3) [36], [37]. The ideal technologies for the current problem were chosen based on the level of maturity of the technology, being open source and nonproprietary, and offering sufficient data rate. Different scenarios can be envisioned as tabulated in Table 1. The envisioned smart grid communications system is shown in Figure 1.4.

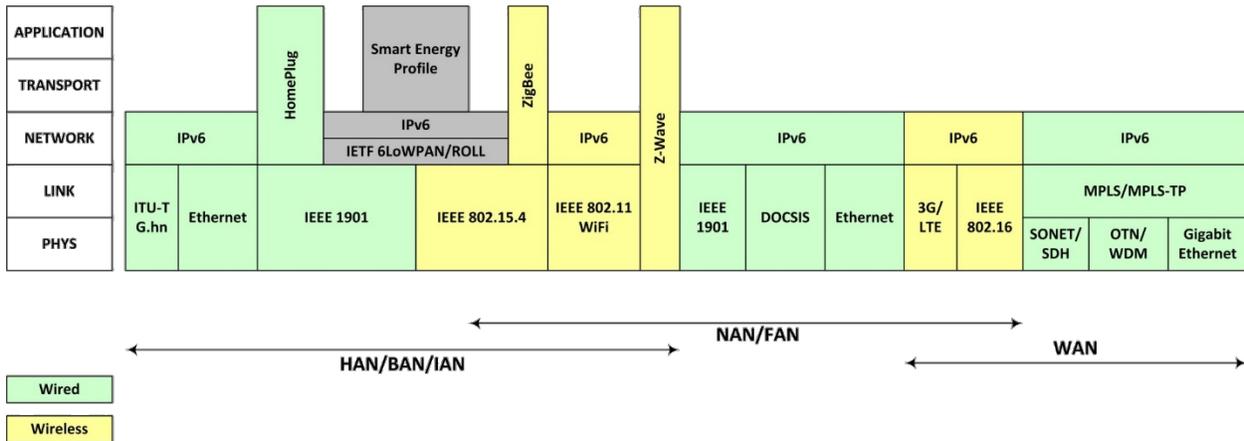


Figure 1.3. Common communications protocols for the power system

Table 1.1. Alternative Communications Technologies for Distributed PV Coordination Communications System

Data Flow	Category	Technology	Coverage	Bandwidth	Benefits	Disadvantages
PV inverter – smart meter	Wireless	LoWPAN (IEEE802.15.4)	10–30 m	250 kbps	Low power consumption, lost cost	Low data rates, limited range
	Wired	PLC (HomePlug)	30 m	80–200 Mbps	Compatible with SEP 2.0 of PV inverters	Network impedance changes with loads, noise during transients in the network
Smart meter – data concentrator	Wireless	WiFi (IEEE 802.11s)	30–1000 m	54 Mbps	Robust against power line outages, allows for multihop communications in case of a node failure	If using unlicensed spectrum, it could be subject to intrusion
	Wireless	WiMAX (IEEE802.16)	50 km	70 Mbps	High bandwidth, licensed spectrum, lower cost	No direct control over the data
	Wired	Ethernet cable (IEEE802.3)	Unlimited	10–100 Mbps	Faster speed, lower latency, no interference	High cost
Data concentrator – edge router	Wired	Optical Ethernet (IEEE802.3ba)	Unlimited	10 Gbps	High bandwidth, secure. Infrastructure may already be in place	No direct control over the data

Critical messages are required to guarantee performance of the publish-subscribe middleware described in Section 1.1.3. The coordination messages outlined in that section define the critical messages to and from nodes throughout the network—primarily concentrator nodes with middleware framework installed.

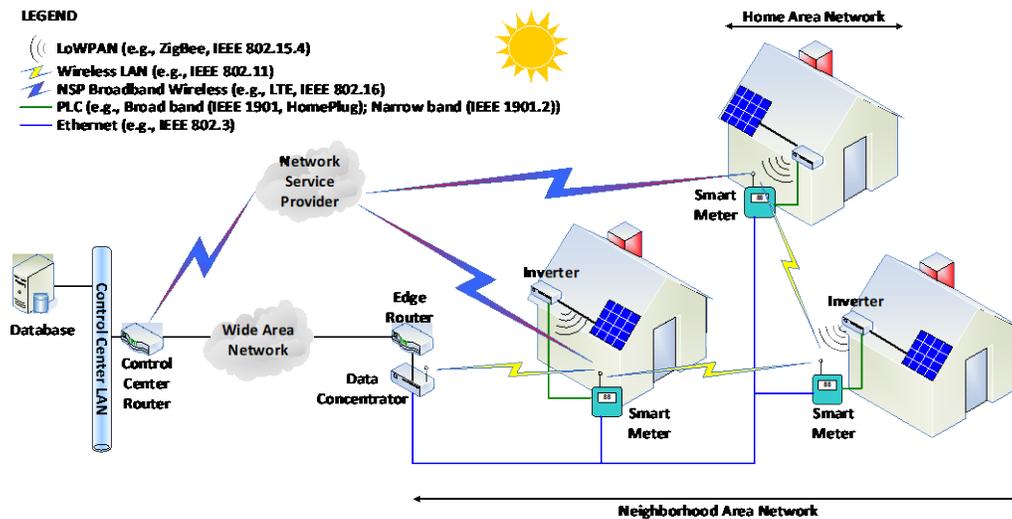


Figure 1.4. Communications model and information mapping of the HAN and NAN

Subtask 1.1.2: Opportunistic Hybrid Infrastructure

Objective: Develop the opportunistic hybrid communications infrastructure with robust channel allocation schemes. Investigate a prototype communications system, modeled on top of the Reference Test Case A power system, to quantify the bandwidth and communications infrastructure required to implement the opportunistic hybrid communications system.

An opportunistic hybrid communications system is designed by augmenting existing communications infrastructure with upgrades. Potential locations for additional communications infrastructure are identified in the network, and an optimization problem is used to determine where upgrades should be made to guarantee that the QoS requirements are met. This formulation used existing hardware to the fullest extent, and necessary upgrades are made at minimum cost. To guarantee problem tractability, a simplification of the full network model presented in Section 1.1.3 is outlined in this section and is used to approximate the performance of network with full middleware capability.

The hardware on the communications network can be divided into two broad categories of equipment: nodes and links. As outlined by the information mapping in Section 1.1.1, nodes are classified from the following: inverters, smart meters, data concentrators, edge routers, or control center routers. Links are classified as: ZigBee, in-home PLC, neighborhood PLC, RF connection, fiber cable, or NSP (network service provider) communication. Data links have properties of bandwidth, time delay, packet loss, and failure rate. Nodes are represented as having the properties of time delay, capacity, packet loss, and failure rate. These properties are represented as random variables, which could depend on the state of the network.

When designing the hybrid communications infrastructure, it is important to guarantee channel allocation schemes for all realizations of the network properties. Optimizing only for expected network performance will not capture the QoS in extreme events. The optimization problem needed to take the variability of the equipment parameters into consideration. The following section outlines a Monte Carlo facility location problem, which models the movement of data through the communications network as a minimum-cost network flow problem.

The full input network is represented as a digraph $\bar{G}(\bar{N}, \bar{L})$ with nodes \bar{N} and links \bar{L} , shown in Figure 1.5. To simplify our model representation, each network node n with incoming links L_n^{in} and outgoing links L_n^{out} is represented by two nodes, n_1 and n_2 , in $G(N, L)$, which are connected by a single link from n_1 to n_2 . The incoming links L_n^{in} are connected to n_1 , and n_2 is connected to the outgoing links L_n^{out} . This allows links to capture all the all network properties and nodes in $G(N, L)$ to facilitate their connection.

$p(n)$

Figure 1.5. Digraph representation

The existing network is defined as directed graph $\hat{G}(\hat{N}, \hat{L})$, which is a subset of the full network $G(N, L)$ and contains the existing nodes $\hat{N} \subseteq N$ and links $\hat{L} \subseteq L$. $l \in \{L \setminus \hat{L}\}$ and $n \in \{N \setminus \hat{N}\}$, representing links and nodes that do not exist in the current communications network but have the potential to be constructed. Each link $l \in L$ has properties $p(l) = \{D(l), B(l), L(l), F(l)\}$. $D(\cdot)$, $B(\cdot)$, $L(\cdot)$, and $F(\cdot)$ refer to delay, bandwidth, loss, and failure rate, respectively. These parameters are random variables that are drawn from a distribution defined for each node and link type. The distributions for these random variables are determined by the simulation of the middleware framework defined in Section 1.1.3 using the parameters obtained in Section 1.1.1. These random variables have realizations of $d(\cdot)$, $b(\cdot)$, $l(\cdot)$, and $f(\cdot)$, respectively.

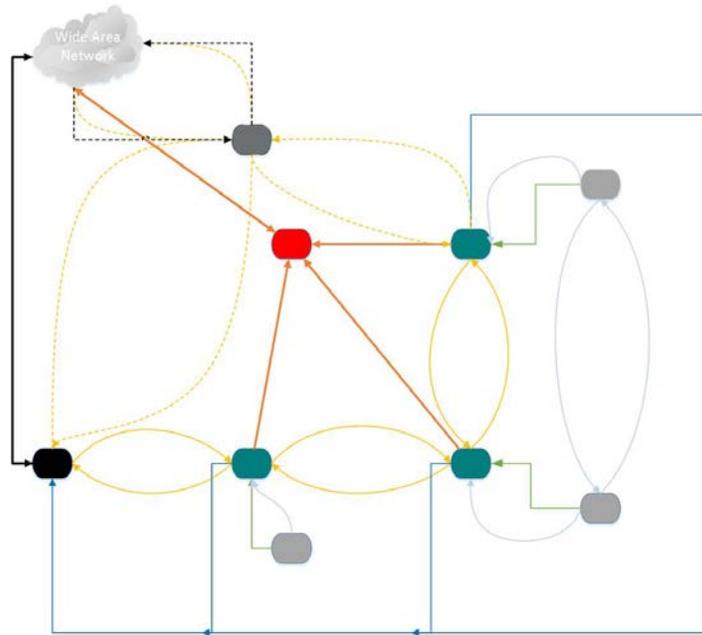


Figure 1.6. Digraph network representation

The network depicted in Figure 1.4 can be represented as the digraph $\bar{G}(\bar{N}, \bar{L})$ shown in Figure 1.6, which can be used to construct the digraph $G(N, L)$ for the optimization formulation. Nodes and links that are already built are solid, whereas nodes and links that could potentially be constructed are dashed. This digraph represents having one additional concentrator node in the neighborhood network that could potentially be constructed. Each different link type has different properties, which are defined in Table 1.

The movement of packets through the network could be represented as flow, with the number of packets entering a node equaling the number of packets leaving the node plus any packets injected at the node less any packets removed at the node. The number of packets routed along link l is determined by the variable x_l . Packets are injected at inverter nodes and could be removed at nodes that perform state estimation. The net number of packets added at node n is given by the parameter a_n . It is possible to model this as a function of the number of packets that enter the node, but for simplicity in this formulation we treated this as a parameter. Costs for packet losses and message delays are defined as c_L and c_D , respectively. Intelligent message scheduling and prioritization is performed across the network to optimize the performance network. The minimum-cost network flow is used to determine the optimal network utilization required to send packets from the inverters to the control center with minimal delay and packet losses. These costs provide soft constraints for QoS requirements. To guarantee the QoS requirements, the total packet loss and latency across the entire network are also bounded by hard constraints. Packet loss on a link $L(\cdot)$ determines the percentage of packets that are lost along the link. Time delay $D(\cdot)$ is the time that a packet takes to travel along a link. The bandwidth $B(\cdot)$ defines the maximum number of packets that can travel along a link. The failure of a line $F(\cdot)$ is a binary value that determines whether the capacity of the link is nonzero. To model this network, we draw the values $d(\cdot)$, $b(\cdot)$, $l(\cdot)$, and $f(\cdot)$ from their respective distributions $D(\cdot)$, $B(\cdot)$, $L(\cdot)$, and $F(\cdot)$ by Monte Carlo sampling. Further insight is provided by

instantiating $d(\cdot)$, $b(\cdot)$, $l(\cdot)$, and $f(\cdot)$ conditioned on various events. Intelligent message scheduling for the constructed network is represented in Formulation (1.1):

$$\begin{aligned}
\text{Minimize} \quad & \mathbf{c}_L \sum_{l \in \hat{L}} l(l) \mathbf{x}_l + \mathbf{c}_D \sum_{l \in \hat{L}} d(l) \mathbf{x}_l & (1.1) \\
\text{Subject to:} \quad & \sum_{k \text{ s.t. } l=(k,i) \in \hat{L}} (1 - p(l)) \mathbf{x}_l - \sum_{k \text{ s.t. } l=(i,k) \in \hat{L}} \mathbf{x}_l = a_i \quad \forall i \in \{\hat{N} \setminus s\} \\
& 0 \leq x_l \leq b(l) f(l) \quad \forall l \in \hat{L} \\
& \sum_{l \in \hat{L}} l(l) \mathbf{x}_l \leq L^{max} \\
& \sum_{l \in \hat{L}} d(l) \mathbf{x}_l \leq D^{max}
\end{aligned}$$

To design the opportunistic hybrid communications network, facility location constraints are added to formulation 1.1. For simplicity, in the modified graph $G(N, L)$, the construction decisions are all made for links, and the N is assumed to be the same as \hat{N} . The Boolean parameter \hat{y}_l is 1 if link l is in the existing network $\hat{G}(\hat{N}, \hat{L})$, and 0 otherwise. The binary decision variable y_l is 1 if link l exists in the augmented network and 0 otherwise. All links in the existing network are assumed to also exist in the augmented network as described by the constraint providing a lower bound on y_l . Construction costs for each link $l \in \{L \setminus \hat{L}\}$ are added to the objective function with cost c_y^l . The optimization problem of selecting infrastructure upgrades to the existing communications network to meet QoS and reliability requirements is described in the Formulation (1.2):

$$\begin{aligned}
\text{Minimize} \quad & \sum_{l \in L} c_y^l (y_l - \hat{y}_l) + \mathbf{c}_L \sum_{l \in L} l(l) \mathbf{x}_l + \mathbf{c}_D \sum_{l \in L} d(l) \mathbf{x}_l & (1.2) \\
\text{Subject to:} \quad & \sum_{k \text{ s.t. } l=(k,i) \in L} (1 - p(l)) \mathbf{x}_l - \sum_{k \text{ s.t. } l=(i,k) \in L} \mathbf{x}_l = a_i \quad \forall i \in \{N \setminus s\} \\
& 0 \leq x_l \leq b(l) f(l) y_l \quad \forall l \in L \\
& \sum_{l \in \hat{L}} l(l) \mathbf{x}_l \leq L^{max} \\
& \sum_{l \in \hat{L}} d(l) \mathbf{x}_l \leq D^{max} \\
& \hat{y}_l \leq y_l \quad \forall l \in L \\
& y \in \{0,1\} \quad \forall l \in L
\end{aligned}$$

To guarantee performance in worst-case scenarios, a Monte Carlo simulation is run over multiple instantiations of the distribution parameters $D(\cdot)$, $B(\cdot)$, $L(\cdot)$, and $F(\cdot)$. Packet loss and latency requirements need to be met for each instance. The additional network infrastructure required to meet all scenarios with 99.9% availability and latency of messages < 1 second for all instances is then constructed. To account for message priorities, the variables x_l and the parameters a_i could be decomposed into separate variables for each message priority. Messages with higher priority would be sent first along the communication channels to reflect the middleware protocol described in Section 1.1.3. Different QoS requirements could then be placed on each message category.

This model is applied to Reference Test Case A to determine the optimal placement of additional hardware required to meet the QoS and reliability requirements. The system requirements are determined by the constraints in the optimization formulation.

Subtask 1.1.3: Middleware Framework

Objective: *Develop the real-time publish-subscribe middleware framework with decentralized management and intelligent scheduling and prioritization algorithms for the transmission of*

multicast messages. This work designs a novel real-time publish-subscribe messaging pattern with decentralized management to provide high flexibility, which appropriately meets the need of the monitoring and control of distributed PV generators. Also, develop intelligent message scheduling and prioritization algorithms to enable communication-environment-awareness and address the challenge in promoting resilience of time-critical data delivery in the presence of congestion and denial-of-service attacks on communications infrastructure.

This section outlines a real-time distributed middleware architecture that is appropriate for IoT protocols and enables leveraging the existing communications infrastructure. The middleware provides a reliable and efficient architecture for communicating necessary information from distributed DERs to a remote command center in a scalable, reliable, cost-effective, and secure manner. There are three main components related to the proposed middleware architecture: the power application layer, the control layer, and the network infrastructure layer.

Power Application Layer

The use of DERs is increasingly being pursued as a supplement and an alternative to large conventional central power stations. Community-scale distributed energy systems are often interconnected with the conventional power distribution system. IoT technologies can be employed to provide visibility into DER device usage. Currently, sensing devices such as smart meters and solar panel sensors send measurement data to the destination through the Internet; however, the power grid at the distribution level has a very large and complex network, with a large quantity of data streams. A dramatic increase in smart-grid communications load significantly challenges the existing network capacity of IoT. To reduce the communications burden, intermediate processing units such as data concentrators are used to compress and extract data to reduce the communication traffic.

In smart grids various applications are implemented simultaneously, each of which has its own QoS standard. This is categorized by the quality of experience (QoE). To optimally use the limited communications bandwidth, it is necessary to analyze the QoS standards of the individual applications. For example, contingency application services have very critical QoS requirements, whereas some monitoring services have relatively low QoS requirements.

Control Layer

As current distributed network topologies are becoming more complex and heterogeneous due to the larger numbers of devices, the control layer becomes increasingly essential to allocate network resources, protect network infrastructure, and simplify network configurations. As one type of control layer, a communications middleware is designed to handle data transmission. In our work, real-time distributed middleware is developed for intelligent data flow management and security control.

Network Infrastructure Layer

The communications infrastructure layer consists of the first four network layers in the common OSI model [38], including the physical layer, data link layer, network layer, and transport layer [3]. Protocols and devices in these layers might vary widely based on the power system environment, location, and end user requirements. Our proposed middleware efficiently addresses this diversity by abstracting the infrastructure layer as an object that contains functions

of these various protocols and network devices. By doing so, the middleware is able to efficiently configure the infrastructure layer to achieve data flow management via intelligent routing control for the network infrastructure.

Proposed Distributed QoS-Resilient Middleware Architecture

We proposed a real-time distributed middleware to manage data flows of different smart grid application services by exploiting the collaboration of different OSI layers. To achieve the balance among computational power consumption, operating efficiency and accuracy, and congested channel bandwidth, middleware instances are installed only in application-service hosts and cross-domain gateways. In our work, middleware instances are installed in both concentrators and the gateway nodes of the mesh network between the aggregators and the remote servers, as illustrated in Figure 1.7. Note that in Figure 1.7 the link between each pair of gateway nodes represents a abstraction of a small-scale subnet rather than a direct connection. Further, the middleware instances belonging to different network devices are able to communicate with neighboring nodes. In our architecture, all the installed middleware instances have the same structure and functions. The middleware instance installed at the control center node has an Application Program Interface (API) that can send control commands to the middleware instances installed in the individual gateway nodes in real time. For simplicity, for the remainder of the section, we consider only gateway nodes in the network. Figure 1.8 illustrates the interactions between the middleware instances, the power application layer, and the network infrastructure. The middleware instances are allocated between the power application layer and the network infrastructure layer. Only the end hosts in the network have power application layers have middleware with control capabilities.

Our proposed mechanism of the middleware instance for one network device is illustrated in Figure 1.9. As shown in Figure 1.9, the QoS monitoring system probes the real-time data-link-layer QoS information of each data flow, evaluates the observed QoS information with our proposed QoS criteria, and stores the QoS information in a buffer for potential usage in the control algorithm module. Our proposed QoS criteria is implemented by using the information provided by the QoS standard database that could be updated based on the QoE specification reported by either local users or remote users via the power application layer. If the QoE for some data flow does not pass the QoS criteria, an alert is sent out to trigger the attack-resilient control algorithm for further data flow management.

Our proposed control algorithm achieves intelligent data flow management by using information from different OSI layers. This includes QoS information from the data link layer, routing information from the network layer, and QoE specification from the application layer. Additionally, as shown in Figure 1.9, in addition to being triggered by the local QoS alert, the control algorithm in our proposed middleware instance could also be activated by the request sent from the middleware instances in the neighboring nodes.

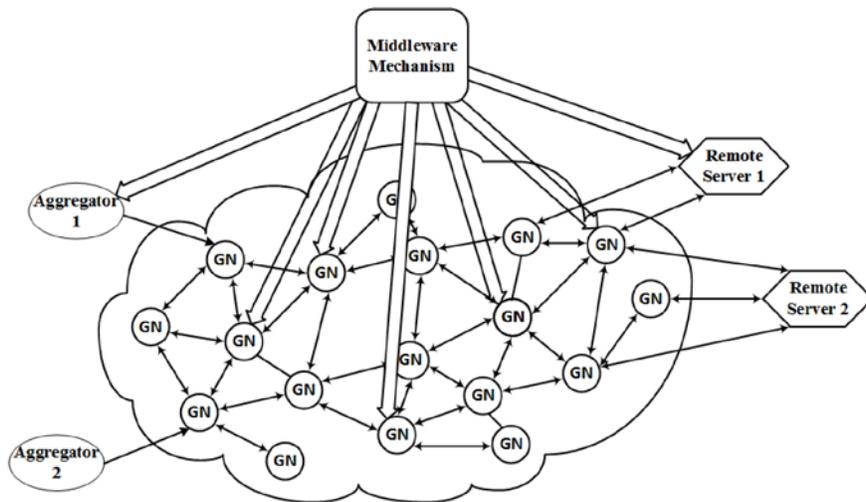


Figure 1.7. Structure of the proposed middleware architecture

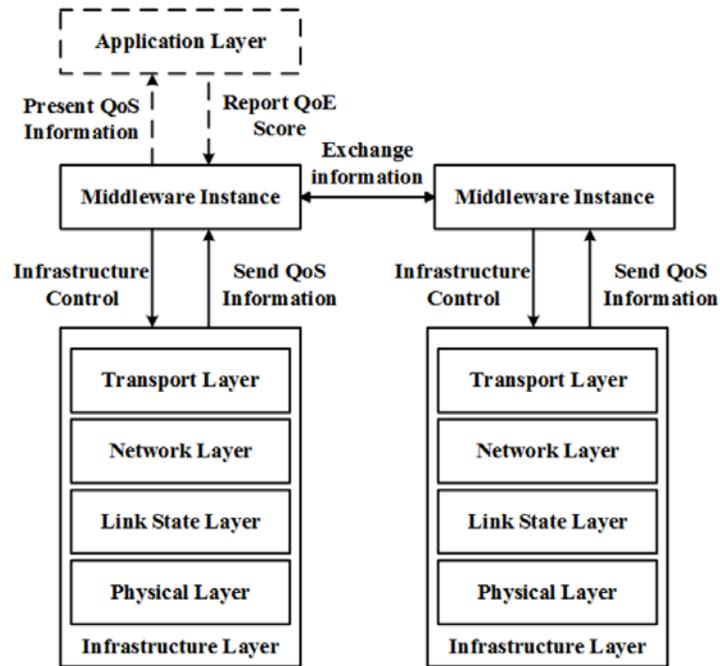


Figure 1.8. Interaction between the proposed middleware instances, power application layers, and network infrastructure layers

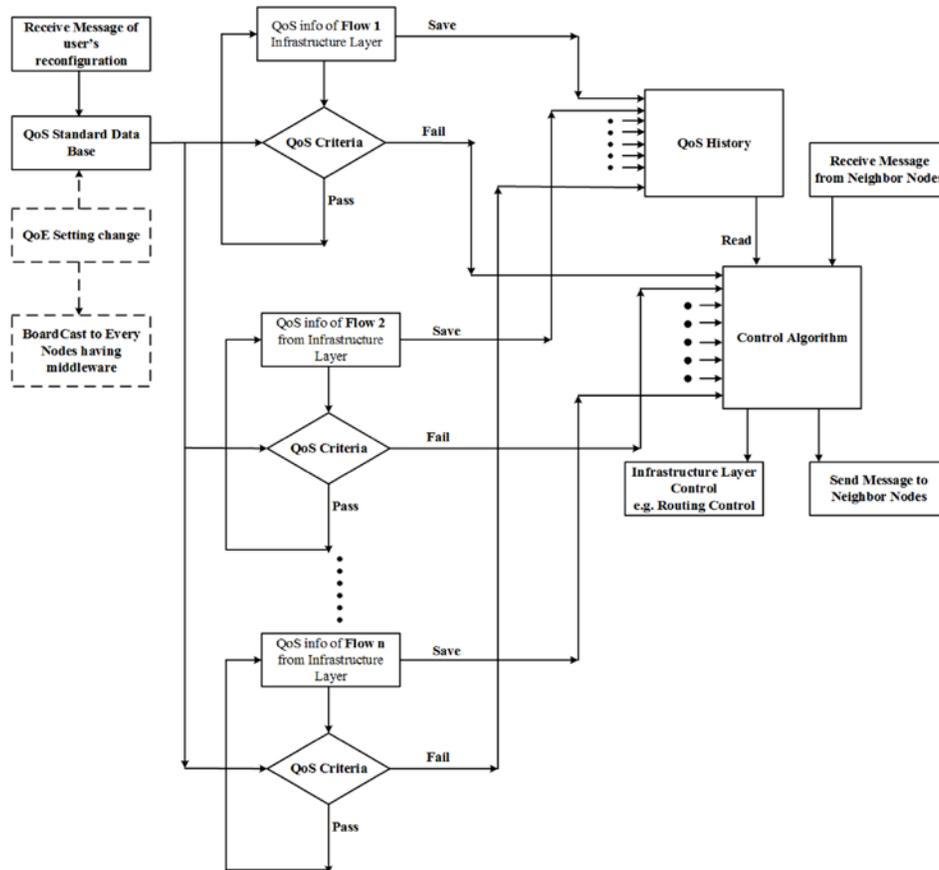


Figure 1.9. Mechanism of the proposed universal middleware

QoS-Based Attack Detection

The network system is assumed to be designed to fulfill the QoS criteria if there were no attack. Moreover, in real-world applications, the lowest two layers—the physical layer and link state layer—are integrated in one network device, and one network node can own more than one network device. For one node, only one middleware instance can be installed, which means that the single middleware instance should be designed to monitor all network devices on that host node. As shown in Figure 11, each middleware instance has a QoS standard database that specifies the priorities and requirements of different application services that are consistent with those in other middleware instances. The information provided by QoS database is detailed in Table 1.2.

Table 1.2. Information Provided by QoS Database

Attribute	Description
i	Interface ID of network devices
n	ID of data flow
\mathcal{L}_n	Priority level of data flow n
\mathcal{S}_n	Source IP address of data flow n
\mathcal{D}_n	Destination IP address of data flow n
p_n	Maximum packet loss rate of data flow n
r_n	Minimum data rate of data flow n
d_n	Maximum delay of data flow n
j_n	Maximum jitter of data flow n
w_n^p	Weight of maximum packet loss rate of data flow n
w_n^r	Weight of data rate of data flow n
w_n^d	Weight of maximum delay of data flow n
w_n^j	Weight of maximum jitter of data flow n

The middleware instance probes the QoS information of different data flows from the data link layer and network layer—such as throughput, latency, jitter, source IP address, and destination IP address in real time—then converts the QoS information to a standard format consistent with QoS criteria stored in its database. The observed QoS information is listed in Table 1.3. By using the observed QoS information in Table 1.3, the quality of the delivery of data flow n in interface i is evaluated as follows [39]:

$$Q_{n,i} = \begin{cases} \mathbf{1}, & \text{if } \Gamma_{n,i} = \mathbf{1}; \\ \mathbf{0}, & \text{Otherwise.} \end{cases} \quad (1.3)$$

where

$$\Gamma_{n,i} = (Q_{n,i}^p \geq 0) \wedge (Q_{n,i}^r \geq 0) \wedge (Q_{n,i}^d \geq 0) \wedge (Q_{n,i}^j \geq 0),$$

$$\begin{cases} Q_{n,i}^p = \frac{p_n - p_{n,i}^o}{p_n}, \\ Q_{n,i}^r = \frac{r_n - r_{n,i}^o}{r_n}, \\ Q_{n,i}^d = \frac{d_n - d_{n,i}^o}{d_n}, \\ Q_{n,i}^j = \frac{j_n - j_{n,i}^o}{j_n}, \end{cases} \quad (1.4)$$

and $p_{n,i}^o, r_{n,i}^o, d_{n,i}^o, j_{n,i}^o$ are observed QoS parameters of data flow n in interface i stated in Table 1.3. If the quality of the delivery $Q_{n,i} = 0$, an alert that indicates the failure of the corresponding data flow is sent to the module of control algorithm for further data flow management. Further,

the evaluation procedures of different flows in different interfaces are executed in parallel at a rate ν .

Table 1.3. Parameters of Observed QoS Information

Parameters	Description
$p_{n,i}^o$	Observed packet loss rate of data flow n transmitted through interface i
$r_{n,i}^o$	Observed throughput of data flow n transmitted through interface i
$d_{n,i}^o$	Observed maximum delay of data flow n transmitted through interface i
$j_{n,i}^o$	Observed maximum jitter of data flow n transmitted through interface i

QoE Modification

With our proposed middleware instances, an authorized user is able to report the QoE specification of certain application data flow at the power application layer that results in the modification of the corresponding settings in the QoS standard database of the local middleware instance. This modification is further synchronized among the middleware instances of the neighboring nodes via network layer broadcast. Once the middleware instances of neighbouring nodes receive the modification message, they continue to broadcast the message to their neighbouring nodes. If the middleware instance of one node has received the same message twice, it updates the QoS standard database according to the message.

DoS-Resilient Control Algorithm

As stated previously, the QoS monitoring system in the middleware instance uses the proposed QoS criteria to evaluate the quality of delivery $Q_{n,i}$ for each data flow n at each interface i in the associated nodes. If $Q_{n,i} = 0$, a local QoS alert is sent out to activate the control algorithm for further data flow management and network infrastructure control. Because the default routing table is calculated by applying link-state routing protocol, there might exist multiple shortest paths between the current host node and the destination. Each path refers to one network device interface on the host node, and each interface is assigned with an distinct index. In our work, we predetermine the default routing path such that the data flow is forwarded via the interface with the smallest index value. Section 1.1.2 describes the abstract modeling of this communications network performed using a minimum-cost network flow algorithm.

The details of the control algorithm are illustrated in Table 4. After receiving a local QoS alert, the control mechanism in node p checks the priority of each data flow that causes the alert and then recovers the QoS of the data flows according to the descending order of their priorities. The control mechanism first verifies whether there are available alternative routing paths based on the local routing data. If the result is positive, the control mechanism evaluates the performance of each local interface i associated with an alternative route. Let ℓ_i be the link associated with interface i , \mathcal{D}_i denote the set of the indices of data flows transmitted through interface i , and \mathcal{F}_i denote the set of the data flows transmitted through interface i . We evaluate the performance of interface i by calculating the performance score as follows:

$$S_i = \begin{cases} 3, & \text{if } \phi_i = 0 \text{ and } \ell_i \text{ is idle;} \\ 2, & \text{if } \phi_i = 0, \Phi_i = 1, \text{ and } \forall f \in \mathcal{F}_i \text{ has lower priority} \\ & \text{compared with the target data flow;} \\ 1, & \text{if } \phi_i = 0, \Phi_i = 1 \text{ and } \exists f \in \mathcal{F}_i \text{ has higher or equal} \\ & \text{priority compared with the target data flow;} \\ 0, & \text{if } \{\phi_i = 0\} \vee \{\Phi_i = 0\}. \end{cases} \quad (1.5)$$

where τ is a constant parameter and:

$$\phi_i = \begin{cases} 0, & \text{if there is no QoS alert sent from Interface } i; \\ 1, & \text{otherwise,} \end{cases} \quad (1.6)$$

$$\Phi_i = \bigwedge_{n \in \mathcal{D}_i} \{S_{n,i} \geq \tau\}, \quad (1.7)$$

$$S_{n,i} = w_n^p \times Q_{n,i}^p + w_n^d \times Q_{n,i}^d + w_n^j \times Q_{n,i}^j, \quad (1.8)$$

where w_n^p, w_n^d, w_n^j are constant parameters. Let \mathcal{K} be the set of interfaces that have positive performance scores. If $|\mathcal{K}| = 1$, there is only one interface with a positive performance score, and the control mechanism changes the route to the only interface $k \in \mathcal{K}$. Otherwise, the control mechanism multicasts the request message to the middleware instance on the nodes $m \in \mathcal{M}$, where \mathcal{M} is the set of nodes that have interfaces connecting with interface $k \in \mathcal{K}$. Each middleware instance that receives the request messages returns a feedback message containing the performance score of the associated node m for being a hop node to transmit the target data flow. To achieve the performance score, the middleware instance on node m first identifies which interfaces are included in alternative routes and then calculates the performance score for each identified interface based on Eqs. (1.3) to (1.6). Letting \mathcal{U}_m be the set of the indices of the identified interfaces, the performance score \tilde{S}_m for GN m can be obtained as follows:

$$\tilde{S}_m = \max_{u \in \mathcal{U}_m} \{S_u\}, \quad (1.9)$$

where S_u is achieved by using Eqs. (1.3) to (1.6). After collecting the feedback messages, the control mechanism updates the performance score of each interface i in its associated node p using Eq. (8) and routes the target data flow to the interface with the highest performance score. If more than one interface has a joint highest performance score, the target data flow is transmitted through the interface with the smallest index value.

$$S_i^* = \min\{S_i, \tilde{S}_m\}, \quad (1.10)$$

We assume that interface i connects with an interface node m . If there is no alternative route for the target data flow with a positive performance score, the associated middleware instance hands over the routing task to the previous node in the original routing path by sending a handover message, as stated in Table 1.4.

Control Algorithm with Neighboring Request

The control algorithm can also be activated by the request sent from the mechanism instances of the neighboring nodes that asks for help in managing certain data flow. Therefore, in addition to the mechanism stated in the previous section, there is another parallel mechanism whose function is to process the requests from neighboring nodes. After receiving a request message from the

middleware instance of the neighboring node, the control algorithm first identifies the type of the request message. If the message is a handover message that includes the target data flow ID and destination IP, the control mechanism begins to identify a proper route for the target data flow with the mechanism that executes the algorithm in Table 1.4 beginning from Step 3. Otherwise, if the message is a request for checking the quality of the interface of the potential alternative route, the control algorithm checks the relative interface used for forwarding the target data flow and sends the feedback message to the middleware instance on the hop node from which the request is sent.

Table 1.4. Control Algorithm with Local QoS Alert

Start:
Step 1: Check the priorities of data flows that fail QoS criteria test.
Step 2: Recover QoS of the data flows beginning from the one with the highest priority.
Step 3: Check locally whether there are alternative routes
If there exists alternative routing paths,
jump to Step 4
Else jump to Step 7
Step 4: Check the quality of each alternative Interface i associated with an alternative route by calculating performance score S_i by using Eqs. (3) to (6).
If $\exists i$ such that $S_i > 0$, jump to Step 5
Else jump to Step 7
Step 5: Multicast to the middleware instances on the potential alternative hop nodes and collect the feedback signals stating performance score of these nodes for transmitting the target data flow, that is defined in Eq. (7)
Step 6: Update the performance score of each alternative Interface i as S_i^* by using Eq. (8)
If $\exists i$ such that $S_i^* > 0$, change route to the interface with the highest performance score
Else jump to Step 7
Step 7: Hand over the routing task to the previous hop node in the original routing path and then jump to the End
End

Subtask 1.1.4: Cybersecurity and Mitigation

Objective: Design cybersecurity measures and mitigation strategies. Focus initially on identifying availability, integrity, and confidentiality threats at each level of communications used, from the individual PV unit to the ISO. Develop mitigation strategies against data integrity attacks that aim to harm efficient middleware and state estimation algorithms. Design strategies that ensure the confidentiality of consumer data are employed at all levels. Take an all-hazards approach within the know, protect, monitor, respond, and recover the framework to ensure security against intrusion detection and prevention.

In this section, we demonstrate and evaluate the performance of our proposed distributed middleware architecture in two cases using a real-time co-simulation test system of NS-3 and MATLAB/Simulink. Our test system consists of the sensing devices that monitor the states of the renewable energy resources, as shown in Figure 1.10, smart meter aggregator, mesh network, and the remote server. The mesh network is simulated with NS-3, and our proposed middleware

architecture is developed as a NS-3-based module for simulation. The power system is simulated with MATLAB/Simulink. Data generated from the power system is fed into the simulated network via the tap bridge module. Denial-of-service (DoS) attack flows are generated based on the protocol stack of Linux Ubuntu 14.04 LTS.

Simulated Network Topology

The topology of the mesh network used in our simulation is shown in Figure 1.10. Elliptical nodes are aggregators that send data to remote servers. Circle nodes in Figure 1.10 denote gateway nodes that are used for routing and forwarding data flow between two different subnet domains. All links between these nodes are CSMA links, which represent a complex subnet. To simplify the explanation, we use $Link_{i,j}$ to denote the CSMA link between GNs i and j . In our test system, the links between $Link_{5,8}$, $Link_{6,8}$, $Link_{7,8}$, and $Link_{8,11}$ are set to be bottleneck links with bandwidth set to 10 Mbps and propagation delay set to 10 milliseconds. Further, the bandwidth and propagation delay of the other links in our system are 10-Gbps channel bandwidth and 0 millisecond, respectively.

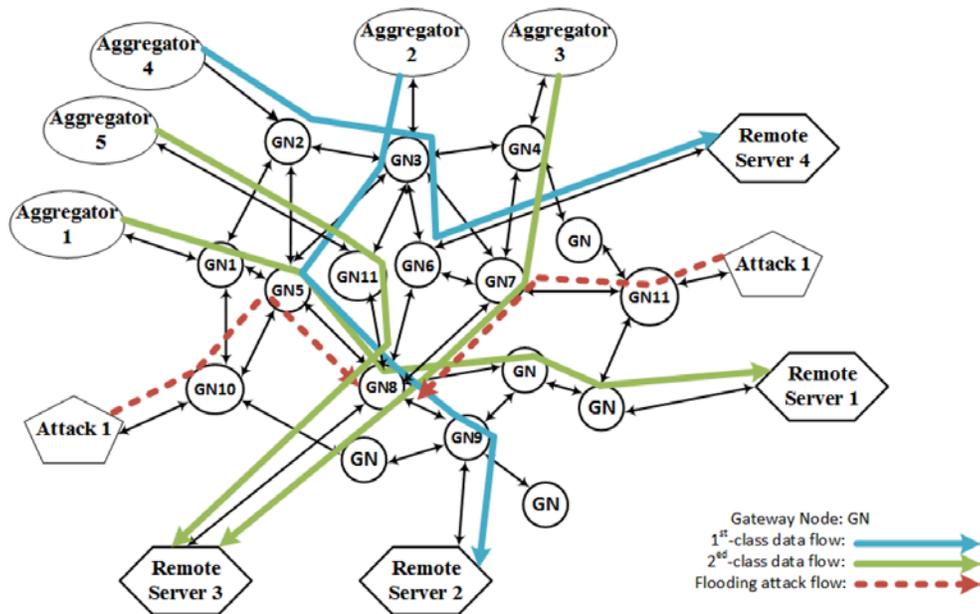
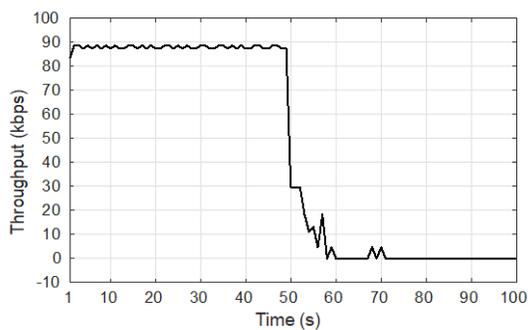


Figure 1.10 Network topology used in the simulation

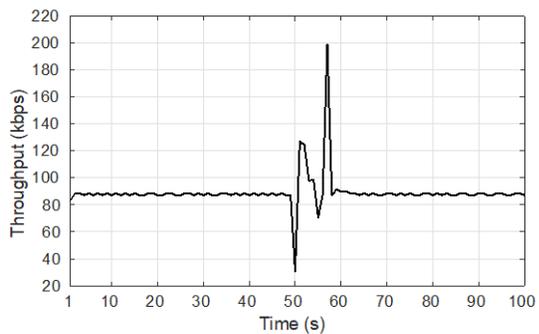
In our simulation, we consider that there are simultaneously five power grid data flows, some of which share the same network link. The detailed QoS requirements of each data flow is specified in Table 1.5. As shown in Table 1.5, data flows 2 and 4, which are used for critical contingency analysis, have the highest priority; and flows 1, 3, and 5, which come from advanced metering infrastructure, have the second priority. In following case studies, we focus on demonstrating the effectiveness of our proposed distributed middleware architecture in protecting the critical Data Flow 2 against potential DoS attacks in the system.

Table 1.5. QoS Requirements of Simulated Data Flow Service

Flow ID	NO.1	NO.2	NO.3	NO.4	NO.5
Service priority	2nd	1st	2nd	1st	2nd
Constant data rate (kbps)	0.128	0.128	0.512	1280	6.4
One-way delay(ms)	<200	<50	<50	N/A	<50
Round-trip delay(ms)	<500	<150	<200	<10000	<250
Jitter(ms)	<50	<50	<50	N/A	<50
Packet loss rate	<5%	<1%	<3%	<10%	<3%
Packet size (Byte)	128	128	512	65536	3072
w^p	0.40	0.45	0.40	0.20	0.40
w^r	0.01	0.01	0.06	0.80	0.06
w^d	0.40	0.45	0.45	0.00	0.45
w^j	0.19	0.09	0.09	0.00	0.09



(a) Without middleware



(b) With middleware

Figure 1.11. Throughput of Data Flow 2 without/with using our proposed middleware architecture

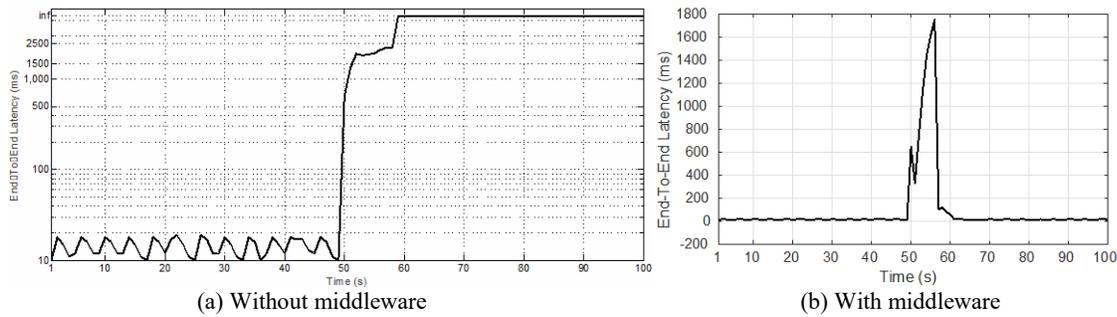


Figure 1.12. End-to-end latency of Data Flow 2 without/with using our proposed middleware architecture

In this case study, we consider one additional UDP flooding attack, called Attack 2, which compromises *Link*_{7,8} and which is identified as the best alternative route in Case I. We assume that the two UDP flooding attacks begin simultaneously at $t = 49$ s. The cooperation procedure of distributed middleware instances is similar to that in Case I. In this case, the middleware instance on GN 7 detects Attack 2 by monitoring the QoS information of the data flows transmitted through the interfaces in GN 7 and implementing the QoS criteria test defined in Eqs. (1.5) and (1.4). Therefore, there is a QoS alert for Data Flow 3 on the interface of GN 7 that is associated with an alternative route. Because of the QoS alert, the feedback signals received by the middleware instance on GN 5 indicate that the performance score of GN 7 is 0, and thus Data Flow 2 is routed to the interface on GN 11. The throughput performance by using our middleware architecture is shown in Figure 1.13(b), from which we can gather that our proposed middleware architecture is able to recover the throughput performance of Data Flow 2 within 15 s after the two attacks occur. Figure 1.13(a) shows the end-to-end latency of Data Flow 2 by using our middleware architecture. From Figure 1.13(a), we can observe that the end-to-end latency is reduced to the specified QoS requirement within 10 s after the attacks occur. Comparing the performances between Figures 1.11 and 1.12(b) and the performances between Figures 12(a) and 1.13(a), we can observe that throughput and end-to-end latency of Data Flow 2 have considerable oscillations after being recovered to meet the QoS requirement in Case II. This metastable behavior of QoS performance implies the sensitivity of the route of Data Flow 2 to potential cyber or physical disturbances in the future.

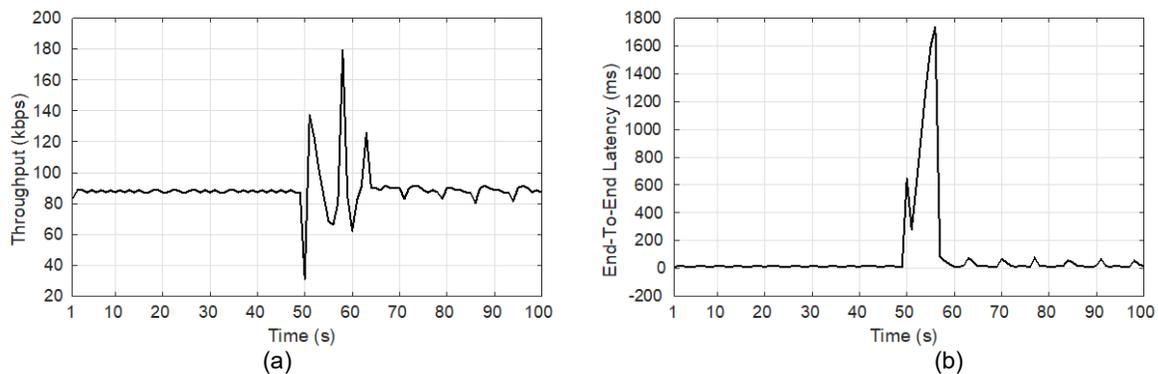


Figure 1.13. End-to-end latency of Data Flow 2 is recovered by avoidance strategy in Case Study 2.

We proposed a distributed attack-resilient middleware architecture to effectively manage the application services based on their priority. Prioritizing the critical data flows with high priority can reduce the loss of the entire power grid in the face of severe physical failures and cyberattacks. We verify the mechanism by considering two attack scenarios. The simulation results illustrate that our middleware architecture is effective at protecting critical data flows against DoS attack. The simulations also show that our middleware architecture provides convenient access for the end user to modify the service priority by reporting the updated QoE specification. We assert that our work will aid in promoting the larger-scale DER integration by providing a resilient communications environment. Future work will examine a generalized class of cyberattacks that our proposed middleware architecture is able to address.

Task 1.2: Develop Photovoltaic System State Computation Algorithms

This task develops a comprehensive imputation and prediction algorithm for electrical (e.g., voltages) and ambient (e.g., solar irradiation) parameters that are missing due to communications outages and measurement failures. By re-examining traditional signal processing and statistical learning tools under today's high-dimensional data regimes, it adapts these tools to power systems and PV-related data. The collection of valid network and PV state measurements is critical for supporting decision-making in a smart grid system; however, in the process of acquiring and transmitting such massive volumes of information, the data are oftentimes corrupted or lost due to meter failures and communications errors. In our smart monitoring context, incomplete state profiles emerge due to three reasons: (1) PMU-instrumented buses are few; (2) SCADA data become available at a considerably slower timescale than PMU data; and (3) residential customers might not be willing to share their PV readings. Accurate prediction and imputation of missing load data, in addition to cleansing those corrupted profiles, is thus of paramount importance for accurately performing the grid monitoring and management tasks.

Subtask 1.2.1: Low-dimensional Visualization

Objective: *Develop low-dimensional visualization techniques that enable systems operators to quickly gauge the state of large-scale systems. To this end, leverage multidimensional scaling and local linear approximations approaches for lower dimensional representations of data on high-dimensional manifolds.*

The geographic proximity of distributed PV generators is exploited in reducing the dimension of measurements required for full observability of the system. Because solar generation is directly affected by the cloud coverage, special consideration is taken in modeling the detrended (normalized) global horizontal irradiance (GHI) during intermittent clouds and overcast conditions. It is understood that a higher order autoregressive model can capture the rapid variation in the normalized GHI caused by cloudy weather. As a test case, the GHI recorded at the Kalaeloa Airport, Oahu, Hawaii, was used to obtain a suitable time-series model [40]. Figure 1.14(a) shows the geographic distribution of 18 GHI sensors at the premise of Kalaeloa Airport. As an example, the raw GHI obtained on June 4, 2011, is shown in Figure 1.14(b). The GHI recorded every second is normalized with respect to the clear-sky GHI obtained from the NSRDB database at a nearby location [41]. The maximum spatial separation among sensors is 0.7 mile, resulting in the minimum spatial correlation coefficient of 0.5. The normalized GHI (i.e., the clear-sky index (CSI)) of 1-second resolution is then smoothed to obtain 1-minute

resolution time series, which still can capture the temporal variation with significant accuracy, as shown in Figure 1.15(a). The corresponding partial autocorrelation coefficient (Figure 1.15(b)) suggests autoregressive (AR) behavior. As a consequence, we consider AR(1) as well as higher orders to model the time series of the logarithmic 1-minute resolution CSI.

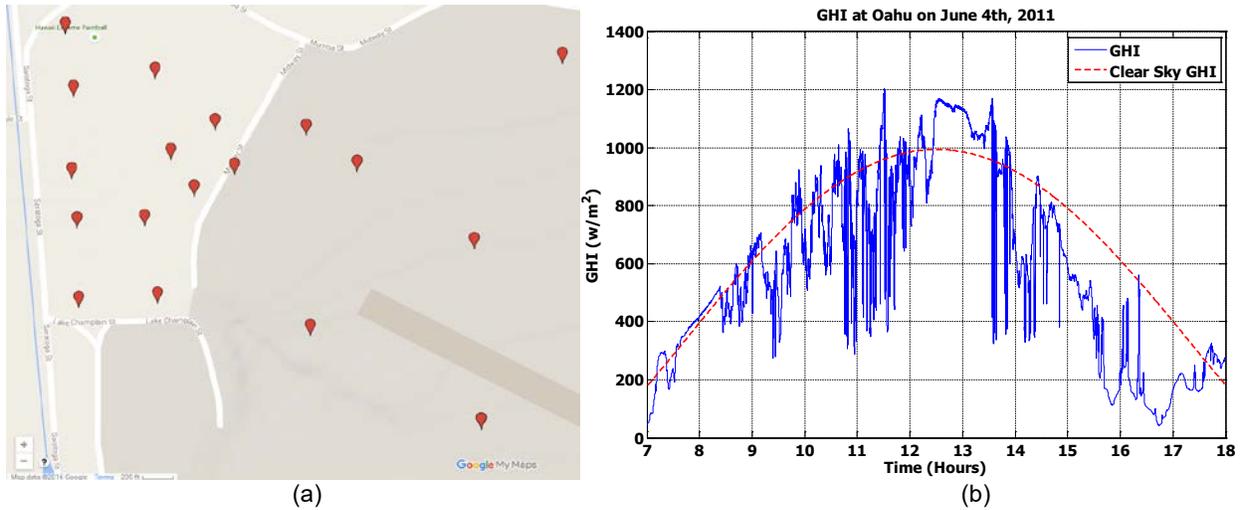


Figure 1.14. (a) GHI sensor locations at Kalaeloa Airport, Oahu, Hawaii, United States; (b) raw GHI at 1-second resolution

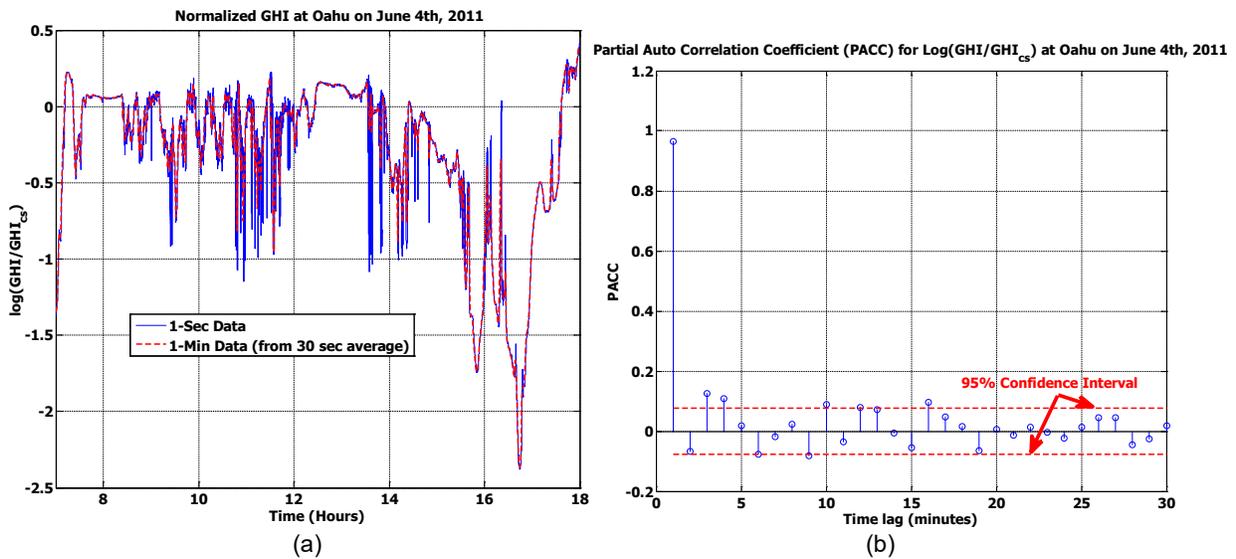


Figure 1.15. (a) Log of clear-sky index, (b) partial auto correlation coefficient

Subtask 1.2.2: Kriged Kalman Filter

Objective: Develop data-driven imputation and prediction techniques to acquire a comprehensive view of the state of distribution networks and PV systems. To this end, leverage Kriged Kalman filtering approaches, low-rank models, as well as dictionary learning techniques.

The Kalman filter operates recursively to obtain a minimum mean square error estimate following a state-space model of system evolution and observation. As a consequence, we present a generic state-space form capturing an autoregressive time-series model of order p ,

$$x_t + [a_1 \quad a_2 \quad \dots \quad a_{p-1} \quad a_p] \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-p} \end{bmatrix} = w_t; \quad w_t \sim \mathcal{N}(0, \sigma_w^2) \quad (1.11)$$

Defining the following vectors,

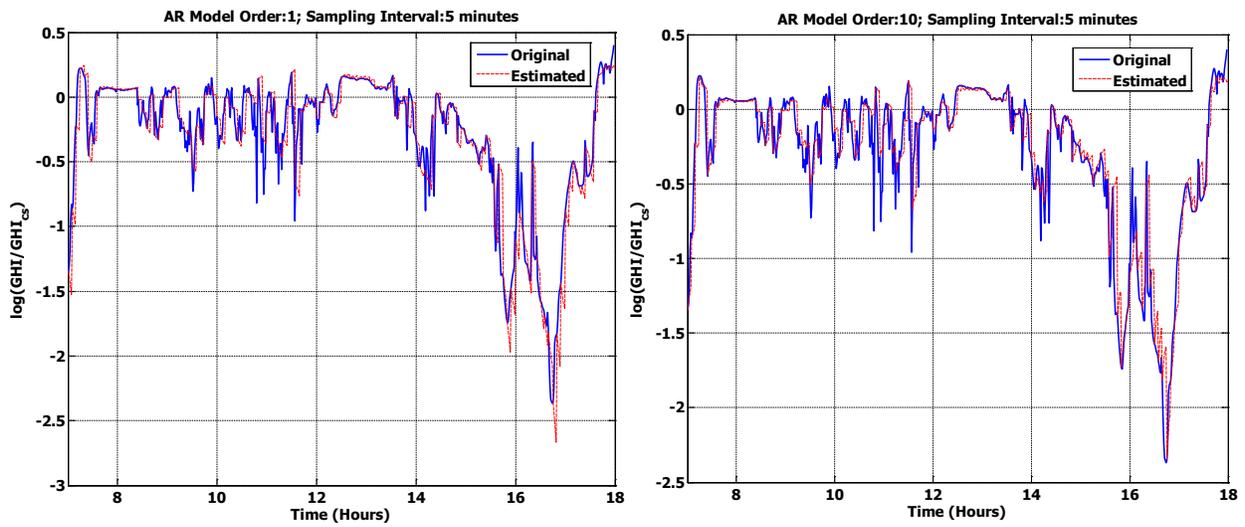
$$w_t = \begin{bmatrix} w_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad x_t = \begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-p+1} \end{bmatrix}$$

We have the following state-space model,

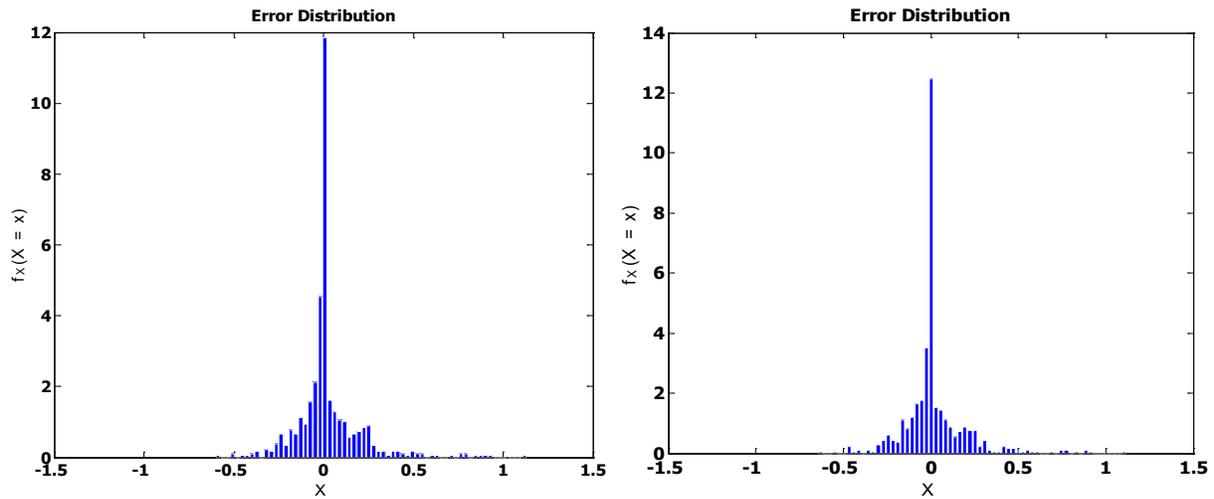
$$x_t = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_p \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} x_{t-1} + w_t = Fx_{t-1} + w_t \quad (1.12)$$

$$y_t = [1 \quad 0 \quad \dots \quad 0] x_t = h^T x_t + v_t; \quad v_t \sim \mathcal{N}(0, \sigma_v^2) \quad (1.13)$$

With a higher order autoregressive model, a Kalman filter can give a satisfactory estimate of PV states with measurements taken at a temporal resolution much lower than the system dynamics. The measurement updates can also be “event driven,” i.e., the Kalman filter will have the measurement only when it deviates from previous measurements beyond a prespecified threshold [42]. Consequently, PV measurements can be recorded at a low sampling rate and thus the temporal dimension of the data is reduced. Note that in the multi-rate and event-driven Kalman filter, the estimation error covariance matrix and Kalman-gain-based estimation correction take place only when a measurement is taken. Thus, it contributes to the reduction of computation and processing time as well. We investigate the effect of down-sampled observation in Kalman filtering for different AR model orders. Specifically, we choose to model the 1-minute resolution clear-sky index of a particular day using 1st- and 10th-order autoregressive models. In both cases, we form the state-space model according to Eqs. (1.12) and (1.13). Consequently, the system dynamics evolve every minute. To implement the multi-rate characteristic, we update the measurement in Eq. (1.13) every 5 and 15 minutes. The corresponding estimation performance for the 1st- and 10th-order AR model is shown in Figure 1.16 and Figure 1.17. We observed that the estimation performance is affected more by the measurement update rate than the autoregressive model order. This is due to the high level of irradiance variability caused by the intermittent clouds. Therefore, the measurement update rate needs to be adaptive according to the level of irradiance variability. One way to achieve this is to keep record of the trace of estimation error covariance matrix. The sampling rate is raised if the trace exceeds a certain threshold and vice versa.

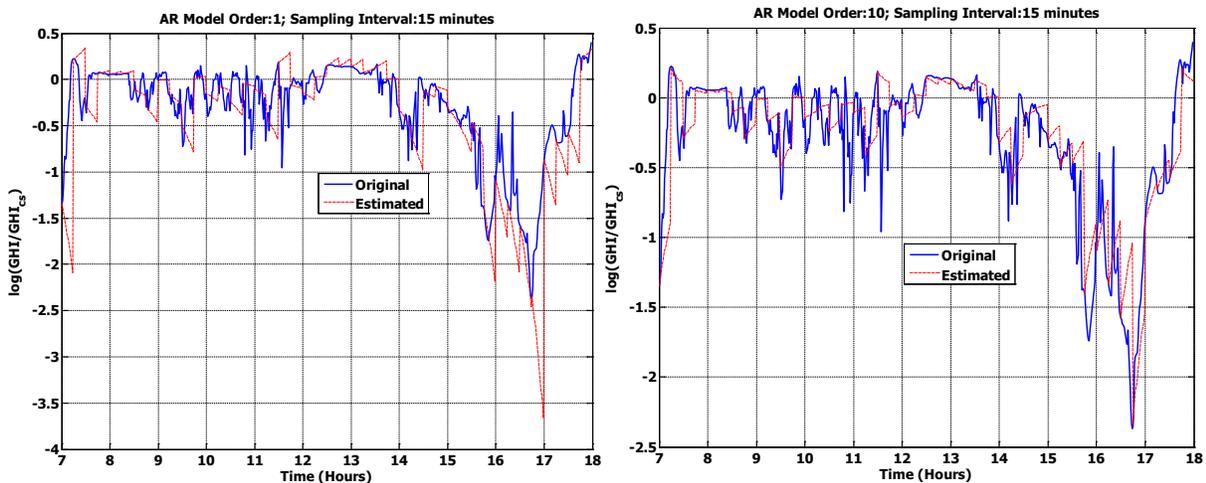


(a)



(b)

Figure 3.16. Dynamic estimation performance with 5-minute interval observation update



(a)

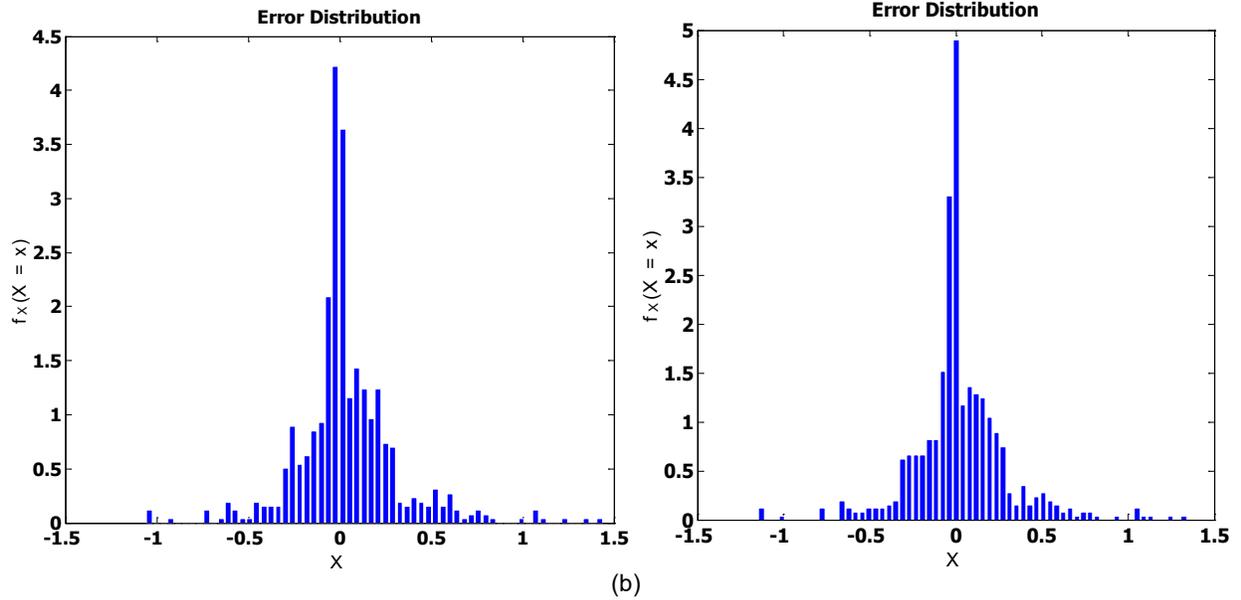


Figure 1.17. Dynamic estimation performance with 15-minute interval observation update

The additional Kriging step within the filtering procedure exploits the spatial correlation to make linear minimum mean square error (LMMSE) estimates at locations from where no measurements are recorded [43]. For n GHI sensors located at $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$, the spatial observation at time t ,

$$\mathbf{y}_t = \begin{bmatrix} y(\mathbf{s}_1, t) \\ \vdots \\ y(\mathbf{s}_n, t) \end{bmatrix} = \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t; \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_v) \quad (1.14)$$

The spatial covariance matrix is denoted by $\mathbf{C}_\mathcal{S}$. Let \mathbf{O} and \mathbf{U} denote two binary matrices mapping the observed and unobserved sensors, respectively. Therefore,

$\mathbf{y}_t^o = \mathbf{O} \mathbf{y}_t$; $\mathbf{y}_t^u = \mathbf{U} \mathbf{y}_t$. Let's denote $\mathbf{H}_t^o = \mathbf{O} \mathbf{H}_t$ and $\mathbf{H}_t^u = \mathbf{U} \mathbf{H}_t$. In the original Kalman filtering steps, we use \mathbf{y}_t^o and \mathbf{H}_t^o to estimate the state as $\hat{\mathbf{x}}_{t|t}$, which is being used in the Kriging step to estimate the unobserved measurements,

$$\hat{\mathbf{y}}_t^u = \mathbf{H}_t^u \hat{\mathbf{x}}_{t|t} + \mathbf{H}_t^u \mathbf{C}_\mathcal{S} (\mathbf{H}_t^o)^T \boldsymbol{\Sigma}^{-1} (\mathbf{y}_t^o - \mathbf{H}_t^o \hat{\mathbf{x}}_{t|t}) \quad (1.15)$$

Here, $\boldsymbol{\Sigma} = \mathbf{H}_t^o \mathbf{C}_\mathcal{S} (\mathbf{H}_t^o)^T + \mathbf{C}_v$. Incorporating the multi-rate and event-driven feature along with the Kriging step a Multi-Rate and Event DRiven Kalman Kriging (MREDRIKK) filter is formed that can make satisfactory state estimates with sufficiently low spatiotemporal measurements. Note that the estimation of unobserved quantities (either solar irradiance or individual PV panel output at electricity customer premises) through Kriging requires an estimation of ‘‘PV states’’ based on a time-varying spatial observation model as in Eq. (1.14). As a consequence, a historical data-driven approach is being investigated to model the observation matrix \mathbf{H}_t and state vector \mathbf{x}_t . In this regard, we are currently investigating the viability of using Gaussian weight kernels [44], spline back-fitted kernels [45], and the probabilistic approach [46]. The

time-dependent state vectors thus obtained will then be used to train the system evolution model through vector autoregressive analysis [47].

Subtask 1.2.3: Decentralized Optimization

***Objective:** Develop decentralized solvers for the optimization problems formulated under Subtask 1.2.1 and Subtask 1.2.2. To this end, leverage primal-dual algorithms and alternating direction method of multipliers to identify saddle-flow points of augmented Lagrangian functions of pertinent optimization problems associated with the imputation and prediction tasks.*

Task 1.3: Develop Distributed State Estimation Algorithms

This task develops distributed state estimation (DSE) algorithms for both the transmission and distribution networks, accounting for the complexity and heterogeneity of the distribution system and the intermittent nature of PV; and provides real-time monitoring of generation variations, load dynamics, and power flows among different elements across the entire system. In particular, scalable algorithms that provide consistent computational performance as the system size grows are developed, enabling the state estimation to be performed at larger scales. Bad data processing is incorporated into the algorithms for resilience against measurement outliers and missing measurements.

Subtask 1.3.1: Subarea Partitioning

***Objective:** Develop an efficient subarea partitioning algorithm that works with the complex and heterogeneous network topologies in both the transmission and distribution systems. The complexity of the topologies of the distribution networks is one of the motivations and challenges to developing DSE algorithms for distribution systems. Therefore, it is important to develop an efficient algorithm that partitions the system into subareas to facilitate the execution and convergence of the DSE algorithms. The topology of the distribution network is radial in general. In some cases, however, loops and meshed topology might also exist. The network topology is also time-variant due to the existence of temporary network elements, such as cuts, jumpers, and grounds. Therefore, the algorithm should consider these characteristics and ensure consistent performance as the system scales.*

An automatic regionalization algorithm is developed for distributed state estimation (DSE). From the investigation, it is observed that regionalization plays a crucial role in the performance of the DSE algorithms; however, in all the existing work on DSE, regionalization is performed manually, and the DSE performance is typically evaluated only with a single arbitrary regionalization instance. Further, manual regionalization is inefficient and tedious for large systems. The main motivation for DSE is to improve the scalability of the state estimation algorithms to cope with large systems with hundreds or even thousands of buses. In such systems, the number of regions (regional processors) also needs to be large to exploit more computational power so that the processing time remains acceptable. Hence, manual regionalization becomes difficult and insufficient, if not impossible. The problem of automatic regionalization (AuReg) for DSE was investigated and the use spectral clustering [48] is proposed for AuReg. The similarity graph of the buses is the basis on which spectral clustering is performed. In the context of AuReg for DSE, the desirable similarity should describe how tightly the two buses are coupled in the state estimation process. We proposed three different similarity measures, i.e., topology-based similarity (TBS), measurement-based similarity (MBS), and

weighted measurement based similarity (WMBS). Some regionalization results on the IEEE 14-bus system are presented in Figure 1.18, Figure 1.19, and Figure 1.20.

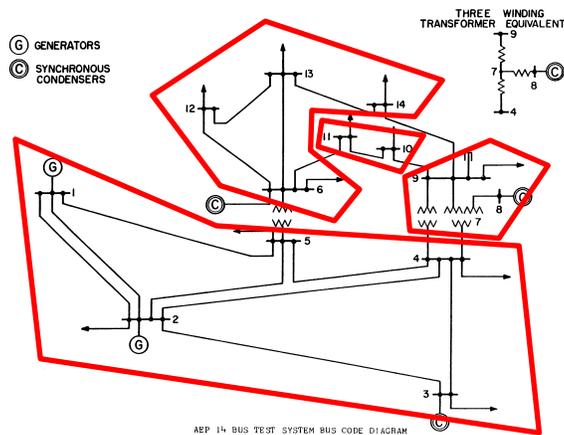


Figure 4.18. TBS-AR

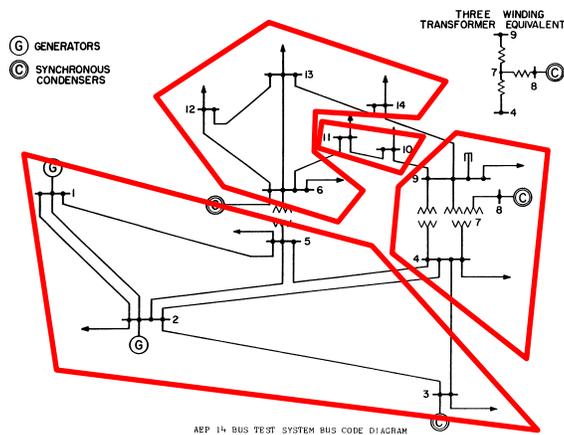


Figure 1.19. MBS-AR

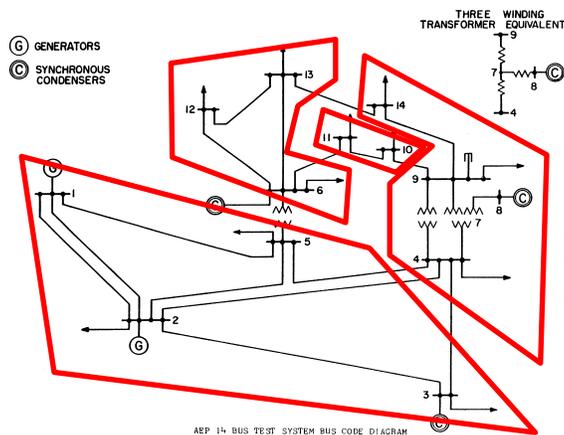


Figure 1.20. WMBS-AR

Note that the regionalization results are different when different similarity measures are employed. All are tested on the IEEE 14-bus system and the 30-bus system. In addition to the three AR methods, two manual regionalization cases, as presented in Figure 1.21 and Figure 1.22, are included in the comparisons.

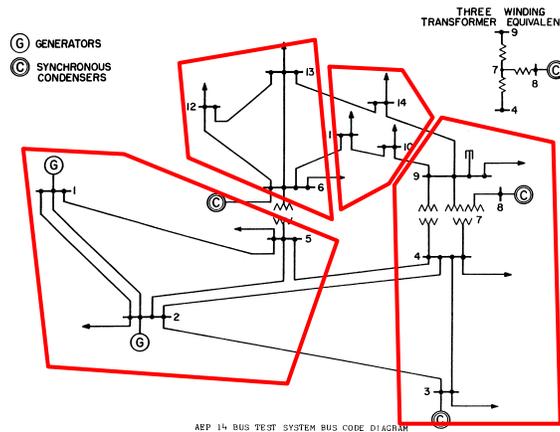


Figure 1.21. Manual regionalization Case 1

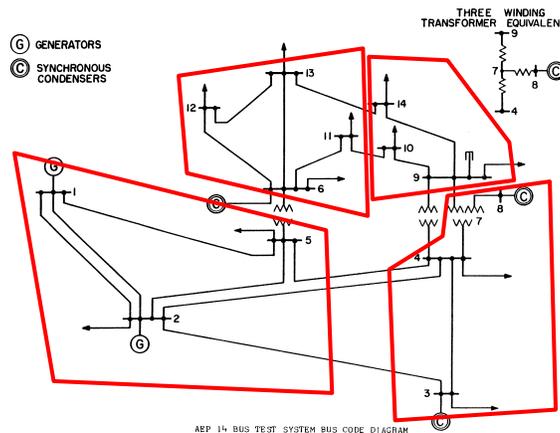


Figure 1.22. Manual regionalization Case 2

Their performance is evaluated in terms of the accuracy and the number of iterations needed to converge within a specified tolerance threshold when applying a DSE algorithm with the regionalization results, as shown in Figure 1.23 and Figure 1.24. In these comparisons, we focus on the DC state estimation problem, and the modified coordinated state estimation (M-CSE) is employed [23]. The figures show that more regions generally lead to more iterations necessary to converge, and in most cases the WMBS-AR yields better regionalization that needs much fewer iterations, especially for the three-region and four-region cases.

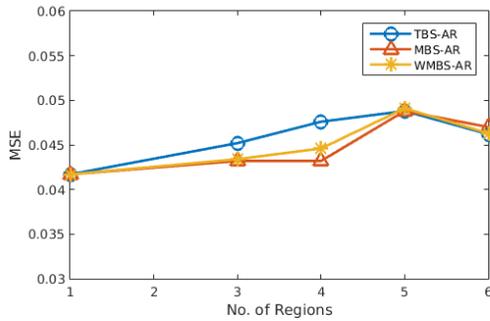


Figure 1.23. MSE comparisons

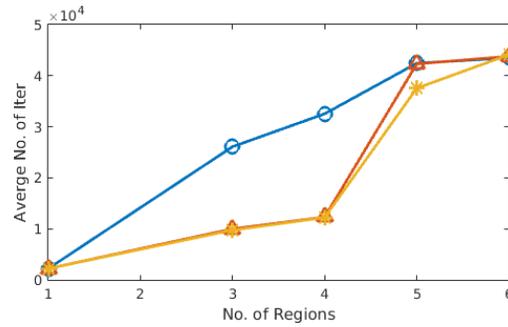


Figure 1.24. Number of iterations comparison

Subtask 1.3.2: Distribution State Estimation Algorithms

Objective: Develop DSE algorithms that ensure scalability, accounting for the proposed communications network architecture, the intermittent nature of PV generation, and the limited measurement availability in distribution systems. The distributed algorithms that solve the weighted least-squares (WLS) problem of state estimation must yield accurate estimates of the system state variables to be effective for stable operation of the power system. The decomposition of the state estimation problem to enable distributed processing is crucial to the performance of the entire system. By applying the auxiliary problems principle (APP), the original optimization problem is decomposed into smaller auxiliary problems, which can be solved in parallel at distributed processors. The communications strategy that provides necessary data exchanges among the processors plays a vital role in the performance of the distributed algorithms as well. Therefore, an efficient communications topology as well as sparse system representations are designed in the context of the proposed communications network architecture. The developed algorithms are tested in transmission and distribution systems of various sizes, and their performance is compared with existing algorithms, including both distributed and centralized algorithms.

DSE in Transmission Systems

We implemented the modified cooperative state estimation (M-CSE) algorithm [23] and the matrix-splitting-based distributed Gauss-Newton (MS-DGN) algorithm [25]. Tests are conducted for both DC and AC state estimation on the IEEE 14-bus system. During the test, we found that the performance, even convergence, of the algorithm is highly sensitive to the weight parameters in each iteration.

The M-CSE algorithm exploits only the gradient of the objective function and therefore requires a large number of iterations to converge. The MS-DGN algorithm approximates the second-order derivative, i.e., the Hessian matrix, with the Jacobian matrix and achieves a much steeper convergence rate. Hence, we employ the MS-DGN algorithm for DSE in transmission systems. Simulations show that, compared with the centralized WLS algorithm, the MS-DGN-based SE achieves similar accuracy with dramatically reduced computational complexity for each processor. Figure 1.25 shows the mean square error (MSE) of the state estimates with respect to the ground truth. We can see that the MS-DGN achieves similar MSE as the centralized state estimation (within 15% increase), which is plotted in the figures where the number of regions is one, except the eight-region case for the 30-bus system. The speed of the algorithm is

investigated in terms of the number of Gauss iterations and the execution time normalized by the number of regional processors. Figure 1.26 and Figure 1.27 shows that the MS-DGN converges with similar numbers of iterations with the centralized WLS and yields more than 70% reduction in the normalized execution time.

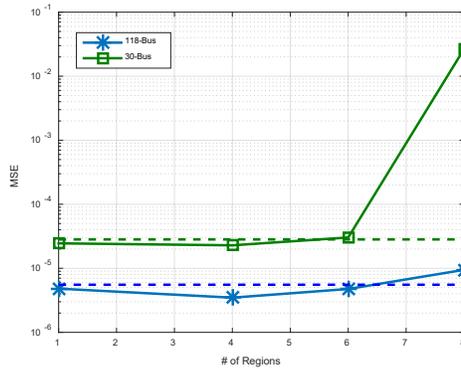


Figure 1.25. Accuracy comparison in transmission system state estimation

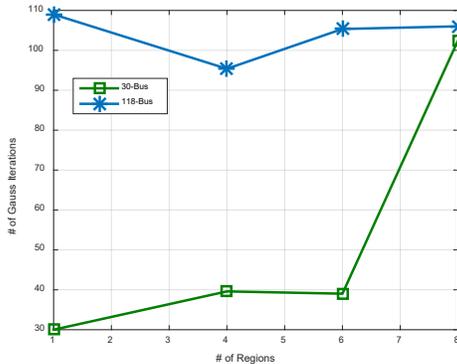


Figure 1.26. Number of Gauss iterations vs. number of regions

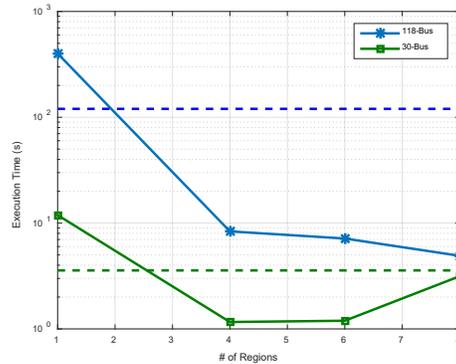


Figure 1.27. Execution time vs. number of regions

DSE in Distribution Systems

The state estimation problem in distribution systems is different than that in transmission systems mainly in the following aspects [76]–[79]:

1. Compared with the heavily meshed topology of transmission systems, the topology of distribution networks in the United States is mostly radial.
2. Due to the imbalanced nature of distribution systems, the state variables in distribution state estimation need to be modeled in three-phase explicitly to accurately describe the state of the system.
3. Available real-time measurements in distribution systems are usually much fewer than those in transmission systems.
4. High R/X ratio.

Consequently, the DSE algorithms for transmission networks discussed in the previous subsection usually do not converge in distribution networks, at least in our test cases.

Figure 1.28 shows that the mean square residual of the WLS formulation oscillates and grows as the Gauss-Newton algorithm proceeds. Therefore, the Gauss-Newton algorithm cannot be applied to distribution systems without significant adjustment [80], [81].

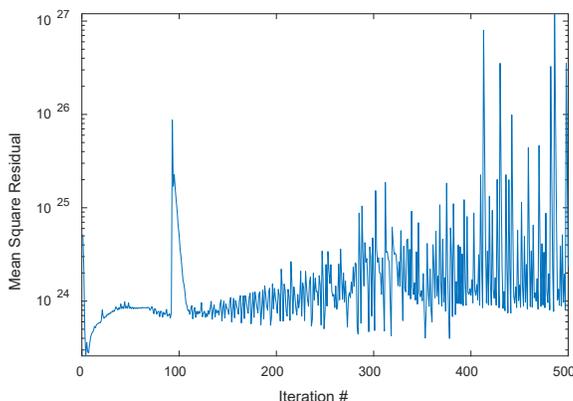


Figure 1.28. The Gauss-Newton algorithm diverges in RTC-A.

On the other hand, the ladder-iterative technique [82], [83] is designed for radial networks. It involves forward and backward sweeps to calculate the voltages. During the forward sweep, the voltage at each node is updated according to the current in branches that connect it with neighboring nodes in the order from the root to the leaves of the tree, according to:

$$V_m = A_{n \rightarrow m} \cdot V_n - B_{n \rightarrow m} \cdot I_{\rightarrow m} \quad (1.16)$$

where the three-by-three matrices A and B can be obtained from the parameters regarding the distribution line (or regulators/transformers) connecting nodes n and m. The vector $I_{\rightarrow m}$ denotes the three-phase current that flows into node m from node n. Initially, all currents are assumed to be zero. After the forward sweep is completed, the load current at each node and the current in branches connecting its parent are updated according to the measurements and the voltages updated in the previous forward sweep, as presented in (1.17), where c and d are also three-by-three matrices determined by distribution line (or regulators/transformers) parameters. The vector $I_{n \rightarrow}$ denotes the three-phase current that flows out of node n toward node m.

$$I_{n \rightarrow} = c_{n \rightarrow m} \cdot V_m + d_{n \rightarrow m} \cdot I_{\rightarrow m} \quad (1.17)$$

The forward and backward sweeps are conducted in turn until the voltages converge to within a given threshold. Nonetheless, the traditional ladder-iterative algorithm cannot exploit multiple processors due to its centralized design, and henceforth, it suffers poor scalability. We propose a distributed state estimation algorithm, named DiLISE, for radial distribution networks based on similar ladder-iterative forward and backward sweeps. The high-level flowchart of the algorithm is presented in Figure 1.29.

Note that the communications in DiLISE is asynchronous, meaning that the regional processors do not need to wait for updates from other regions before they proceed to the next iteration. This helps reduce the communications overhead and avoid typical asynchronous issues, both of which are often problematic for the distributed algorithms.

As for any DSE algorithm, the network must be partitioned into multiple regions [56] before DiLISE could work, and the subnetworks after regionalization should retain the radial topology because of the working mechanism of the ladder-iterative technique. An example of the AR results is presented in Figure 1.30, in which the distribution feeder is partitioned into eight radial subnetworks.

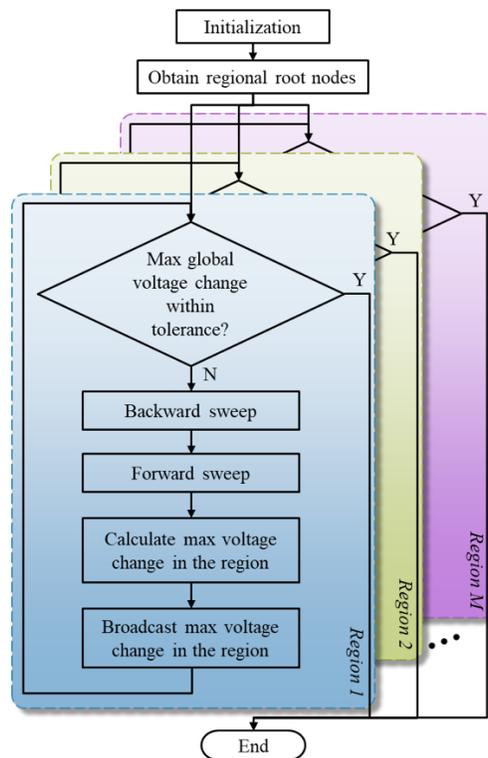


Figure 1.29. The high-level flowchart of our proposed DiLISE algorithm

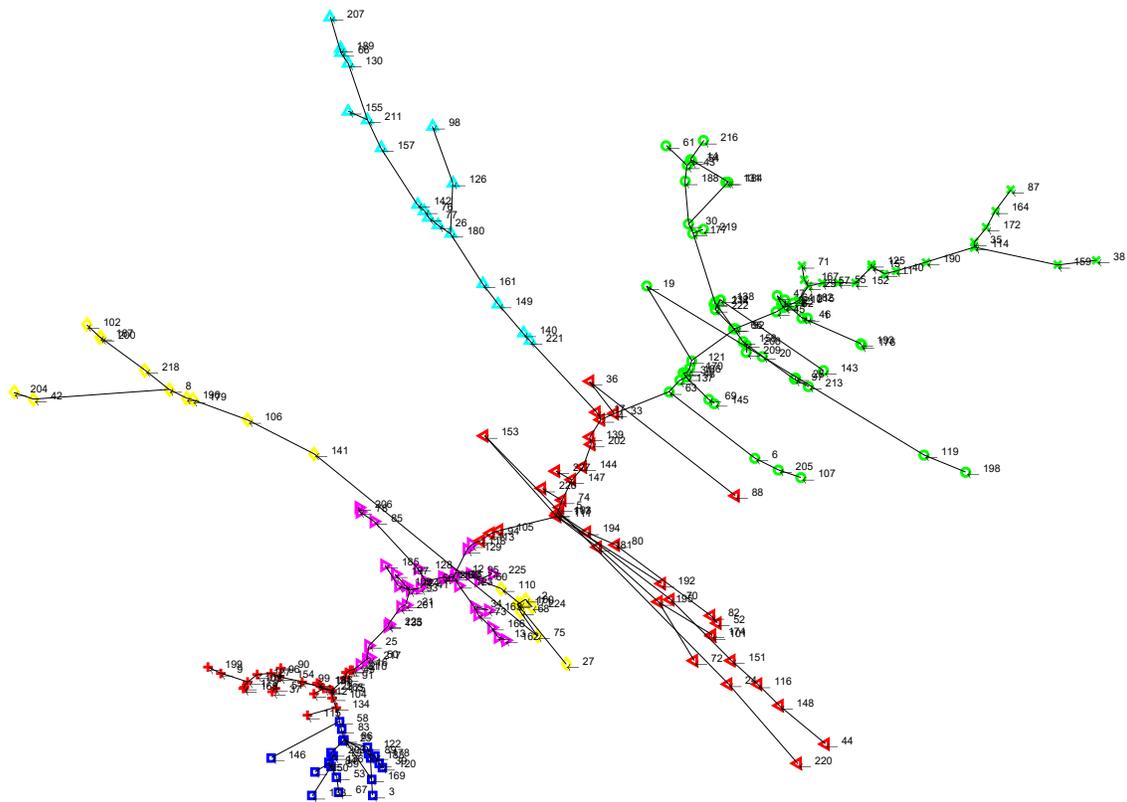


Figure 50. AR with eight regions for RTC-A (nodes with the same marker and color belong to the same region)

We tested our proposed DiLISE algorithm on IEEE test feeders and the RTC-A. It is important to verify that the state estimates obtained by the DiLISE algorithm are as accurate as their centralized counterparts. The comparisons between the centralized and the distributed ladder-iterative state estimation in terms of root mean square error (RMSE) of the state estimates are presented in Figure 1.31. The vertical axis represents the percentile increase in RMSE of the DiLISE results with respect to that of the centralized LISE algorithm. We are confident to conclude that the accuracy deterioration of the DiLISE from the centralized LISE is minimal and negligible for all the test cases visited in our simulations. Additionally, the accuracy degradation of DiLISE remains negligible as the number of regions increases.

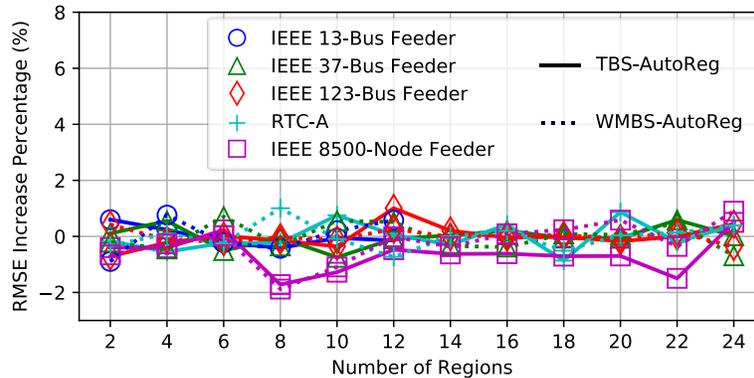


Figure 1.31. RMSE achieved by DiLISE with various number of regions compared to that achieved by the centralized solution

The main goal of distributed processing is to reduce the time that the algorithms take to return a solution. The execution time comparisons between the centralized and the distributed LISE algorithms are shown in Figure 1.32 for each test feeder. The execution time of the distributed algorithm is calculated by dividing the total execution time by the number of regions (the number of processors) and the communications delay between the regional processors is ignored. Nonetheless, the impact of communications delay is expected to be smaller than that in transmission systems with weighted-least-squares-based state estimation algorithms because the communications mechanism in DiLISE is asynchronous and henceforth is innately more resistant to communications delay. As shown in the figure, the execution time of DiLISE, although partitioning the feeders into increasing number of regions, is reduced dramatically with respect to the centralized algorithm in all test cases we investigated. Compared with the transmission-level test cases, the distribution level test cases involve more nodes and a much larger number of state variables. Further, the asynchronous communications in our proposed DiLISE algorithm enables us to exploit the processing power of all regional processors more thoroughly with less waiting and smaller communications overhead. Thus, the benefit we enjoy from employing more regions/processors is considerably more significant.

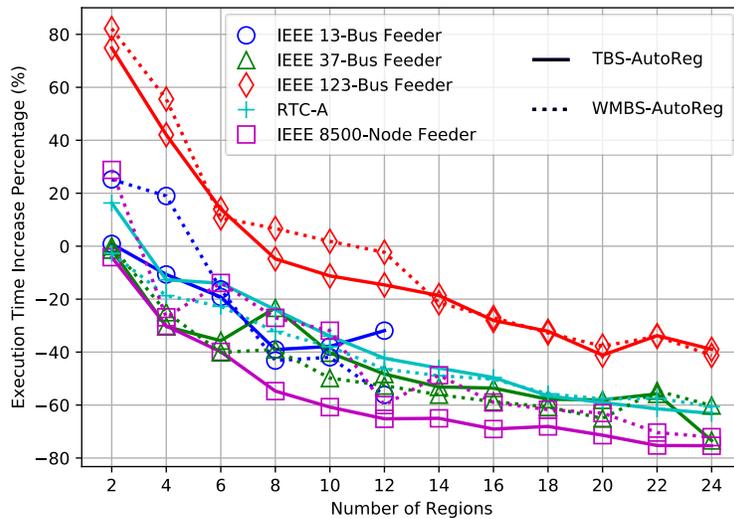


Figure 1.32. Execution time of DiLISE with various numbers of regions compared to that of the centralized solution

A major motivation for DSE is resolving the SE problem for large systems. In this section, we evaluate the scalability of our proposed DiLISE algorithm. In Figure 1.32, the execution time reduction is quite similar for test feeders at various scales. To have a clearer evaluation of the scalability of our proposed DiLISE algorithm, we select the best number of regions for each test case empirically based on the results shown in Figure 14 and plot the execution time reduction versus the number of buses in Figure 1.33. We can see that DiLISE achieves similar percentage of execution time reduction in all test cases except the IEEE 123-bus test feeder.

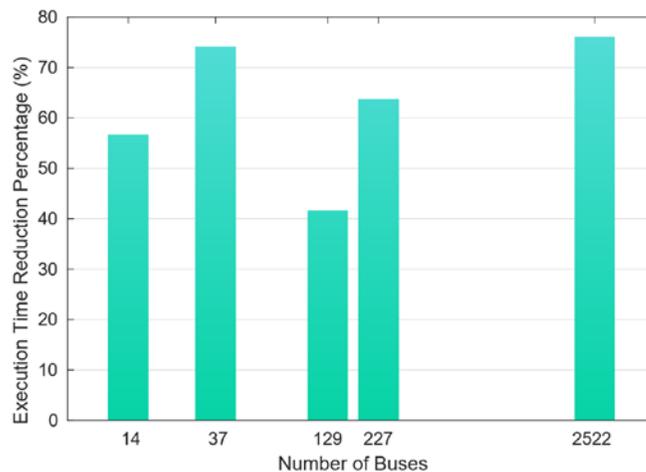


Figure 1.33. Scalability of DiLISE algorithm. The numbers of buses are positioned in log scale for better presentation.

Subtask 1.3.3: Bad Data Detection and Processing

Objective: Incorporate bad data detection and processing into the state estimation algorithms. Missing or low-quality measurements could deteriorate the performance of the state estimation algorithms, and they do occur, especially in distribution systems. Therefore, effective processing

of the bad data among all the measurements could help improve the robustness of the proposed DSE algorithms.

Impact of Bad Data Processing in Distribution System State Estimation

Bad data are present in power system measurements because of the instrument failures, impulsive communications noise, measurement time skewness, or even cyberattacks [56]. A small subset of the measurements will exhibit extremely large errors and should be treated as outliers. Without proper bad data detection and removal, the performance of our proposed distributed ladder-iterative state estimation (DiLISE) algorithm is severely impacted by bad data. As shown in Figure 1.34 and Figure 1.35, both the percentage of measurements that are contaminated with bad data and the magnitude of bad data cause significant degradation in the accuracy of both the centralized and the distributed algorithms.

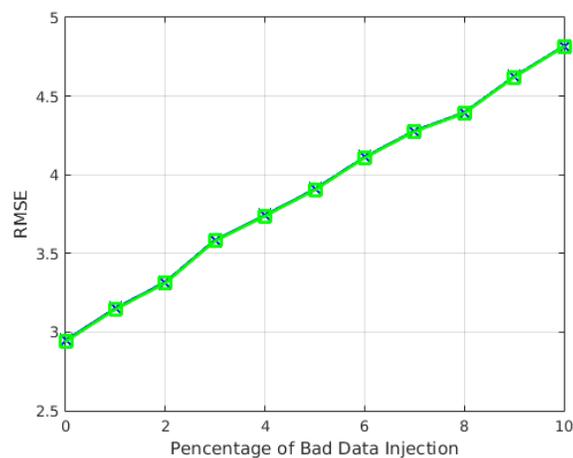


Figure 1.34. Impact of bad data contamination level

In Figure 1.34, as the percentage of bad data injection increases, the root mean square error (RMSE) also increases steadily. Therefore, we can see that the ladder-iterative state estimation algorithms have very limited resistance to bad data. Similar observations can be made from Figure 1.35. Most of the measurements have magnitudes in the order of 10^4 to 10^5 . When the magnitudes of the bad data are comparable with the actual measurements, the estimation accuracy degrades severely.

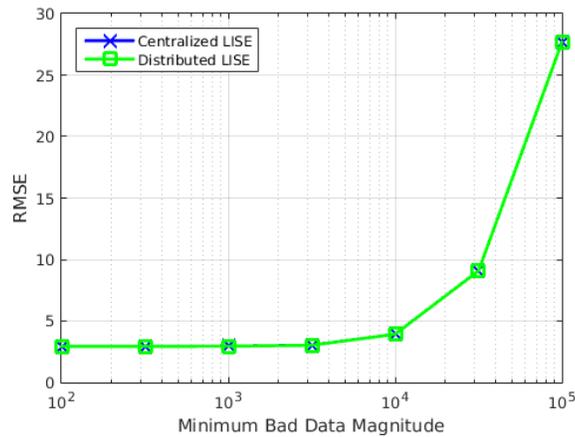


Figure 1.35. Impact of bad data magnitude

Bad Data Processing in Distribution System State Estimation

Traditionally, in transmission systems, bad data in measurements are detected by residual analysis before they are removed or corrected. In the DiLISE algorithm we proposed earlier in this project, however, such processing is infeasible because the measurements employed in the DiLISE algorithm are not redundant. Therefore, we incorporated the weighted least squares (WLS) formulation into the DiLISE algorithm so that redundant measurements can be employed, and bad data processing becomes possible. More specifically, we propose solving the WLS problem in distribution systems by updating the states in forward and backward sweeps, with the aim toward better convergence performance. The overall flowchart of the algorithm is shown in Figure 1.36.

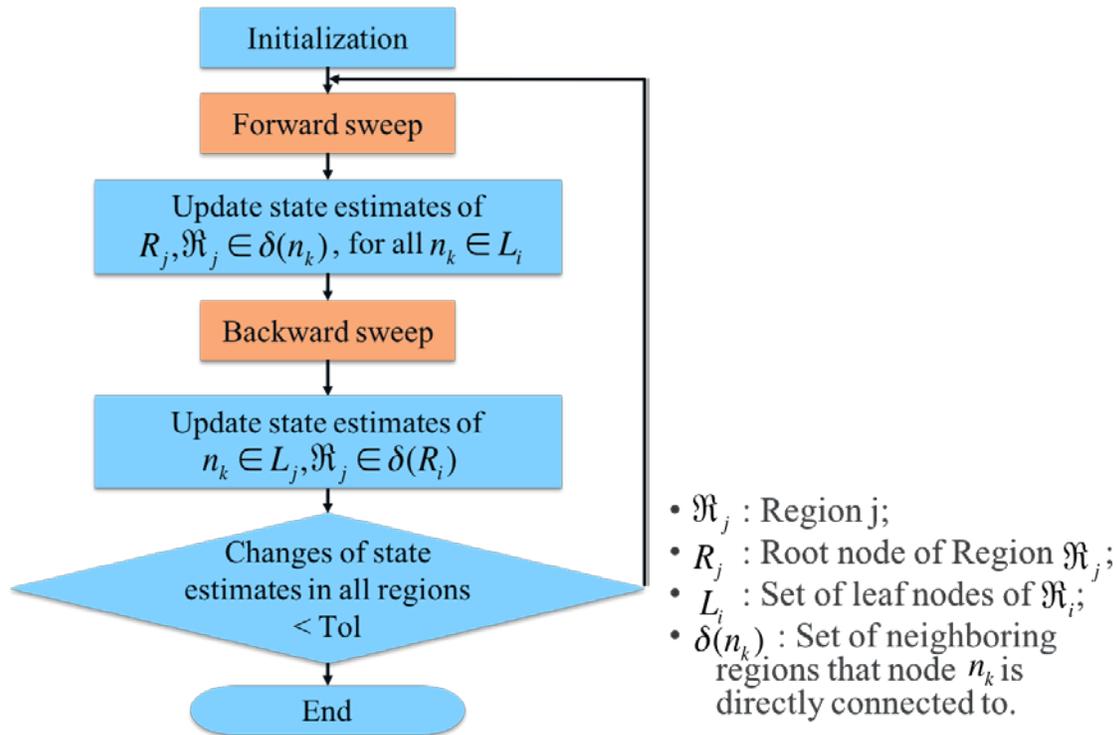


Figure 1.36. Overall flowchart of LIGaN algorithm.

The detailed flowchart of the “Forward sweep” and “Backward sweep” steps, both colored in blue, are presented in Figure 1.37. The basic idea of the forward and backward sweeps in LIGaN is similar to DiLISE, which is to update the states of the nodes along the distribution lines in the network. The difference is at the actual updates that take place at each node. In DiLISE, the currents are calculated based on the load power and voltages at the node of interest, and the voltages are calculated based on the current flow into the node of interest from its parent node. In LIGaN, the updating process is slightly more complicated because it solves a small-scale optimization problem treating only the voltages at the current node as optimization variables.

In our test of the LIGaN algorithm on RTC-A, however, it does not converge as quickly as expected. The variations of the mean squared residual with respect to the iteration number for centralized and distributed versions of the traditional Gauss-Newton [58] and LIGaN are shown in Figure 1.38. Therefore, we need to speed up the distributed algorithms for them to be suitable for real-time operation.

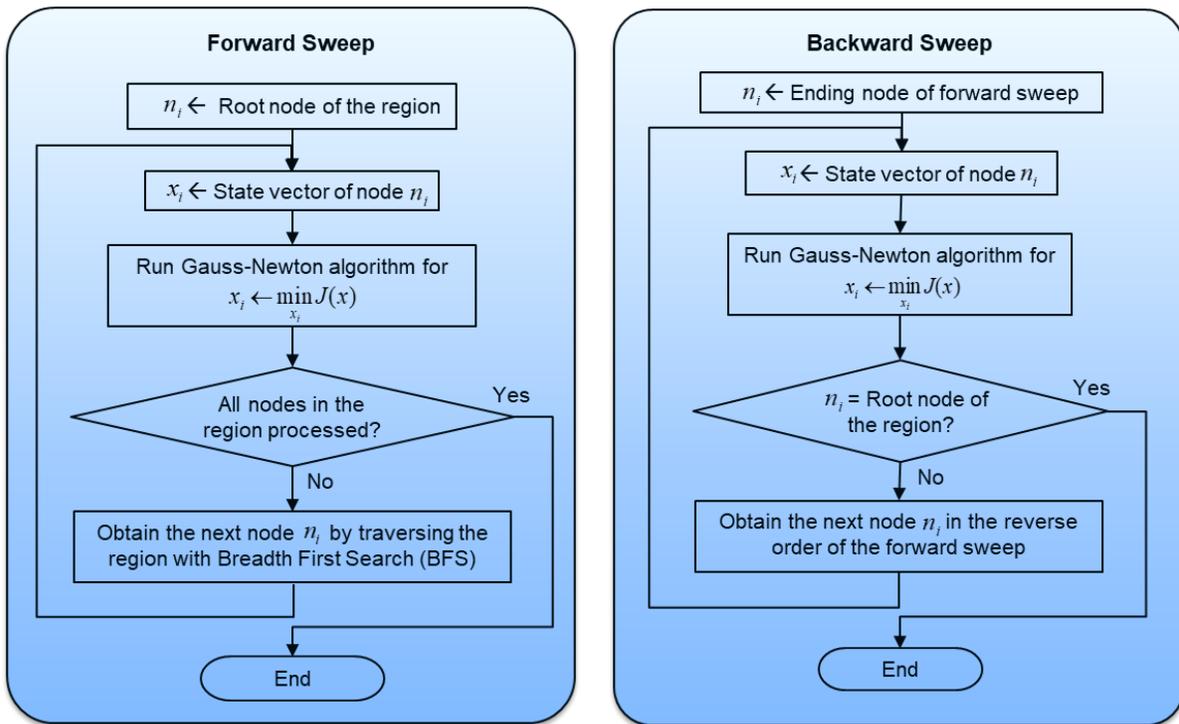


Figure 1.37. Flowchart of forward and backward sweep

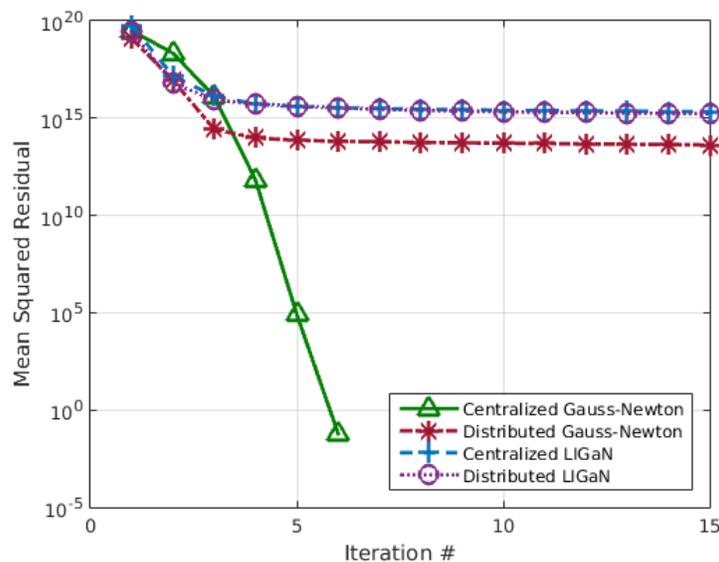


Figure 1.38. Algorithm comparison based on mean squared residual

Further, the bad data suppression capability of WLS-based algorithms is limited because they simply rely on the uncontaminated measurements to outweigh the ones contaminated by bad data. Therefore, considering the distribution of measurement errors with bad data, we incorporate

information theoretic learning techniques [59], such as error entropy minimization, to address this issue.

Bad-Data-Resistant Distribution System State Estimation Algorithm

All measurements are subject to various levels of noise. In the ladder-iterative algorithm, redundant measurements cannot be exploited to improve accuracy and resist impact from bad data, i.e., measurements with gross errors. To resolve these issues, we propose a ladder-iterative belief propagation-based (LIBP-based) state estimation (SE) algorithm.

Simplified Case with Only Linear Measurements

We first investigate the simplified case with only linear measurements, i.e., voltage and current measurements. To clarify the notation, we consider a simple distribution network with four buses, as shown in Figure 1.39. Node j is associated with four variables, namely, V_j , I_j , J_j , and S_j , which represent node voltage, in-flow current from upstream buses, load current drawn at the node, and load power, respectively.

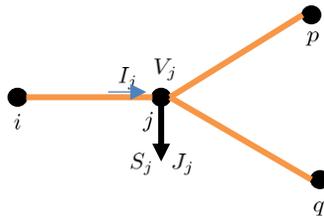


Figure 1.39. An example of a distribution network for notation clarification

The original message-passing schedule we devised is presented in Figure 1.40 for Bus j . The blue and orange arrows denote messages in the forward and backward sweeps, respectively.

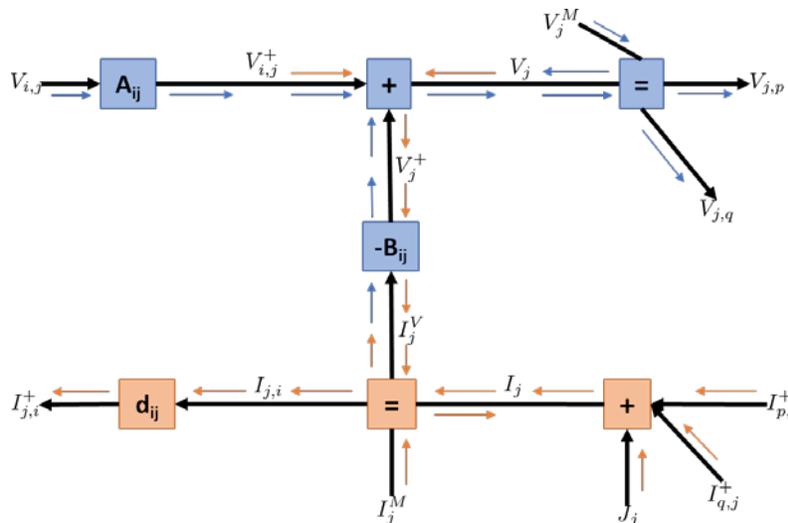


Figure 6 Original message-passing schedule for LIBP

Nevertheless, this message-passing schedule can employ only the voltage measurement at Bus j to improve the voltage inference at Bus j and its descendants in the distribution system. When the load current measurements at some nodes are missing, the SE accuracy at many nodes deteriorates severely, as shown in Figure 1.43.

Therefore, we develop an improved message-passing schedule, which is illustrated in Figure 1.41, to address this problem. The numbers in circles denote the step order in the message-passing schedule, where the blue and orange numbers indicate steps in forward and backward sweeps, respectively. The main differences are:

1. In the forward sweep, Step 4 and Step 5 infer I_j based on V_j and the voltage at the parent bus.
2. In the backward sweep, Step 5 infers V_j by integrating the voltage measurements at Bus j and the messages from its children buses. Steps 6 and 7 infer the voltage of the parent bus based on V_j and I_j .

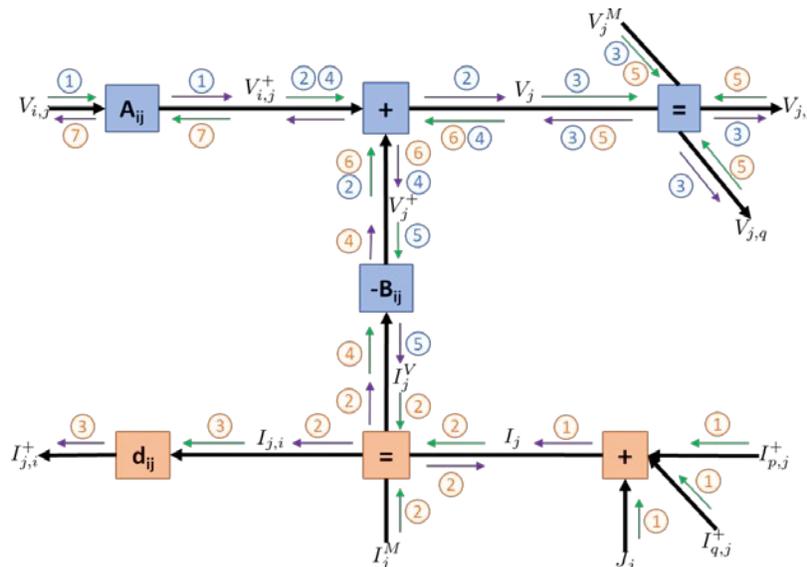


Figure 1.41. Improved message-passing schedule for LIBP

With these improvements, the belief originated from voltage measurements can propagate up and down the network so that the accuracy across the entire network can be improved.

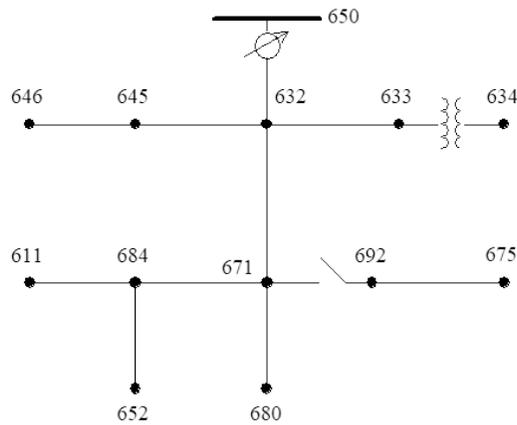


Figure 1.42. IEEE 13-bus feeder topology

The performance comparison between the original and the improved message-passing schedules is conducted in the IEEE 13-bus test feeder, whose topology is presented in Figure 1.42. We consider two cases with different measurement configurations, as shown in Table 1.6. The results are presented in Figure 1.43.

Table 6.6. Measurement Configurations for IEEE 13-Bus Feeder

	Case 1	Case 2
Load current measurements	All buses	All buses except 671
Voltage measurements	None	632, 652, 675, 680

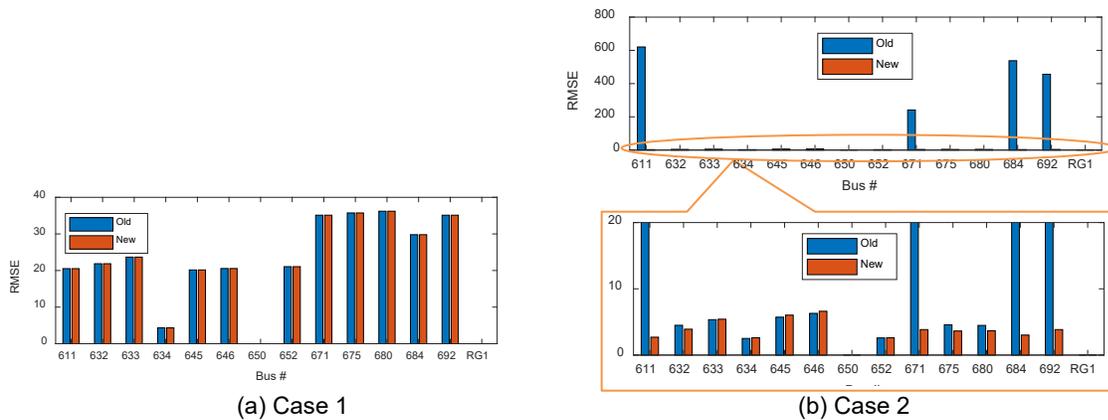


Figure 1.43. Comparisons of the RMSE at each bus using LIBP algorithm with the original and the improved message-passing schedule 1

Figure 1.43(a) shows that when all load current measurements are available, both message-passing schedules yield identical accuracy. Figure 1.43(b) shows that when the load current measurement at Bus 671 is missing, the improved message-passing schedule is able to exploit the voltage measurements at buses 632, 652, 675, and 680 to achieve high accuracy across the feeder. The original message-passing schedule, on the other hand, yields much worse accuracy at

many nodes, even with voltage measurements at four buses. In fact, the root mean square error (RMSE) at every descendant of Bus 671 that does not have a voltage measurement is extremely large.

These results demonstrate that the LIBP algorithm with the improved message-passing schedule can achieve high SE accuracy even with some load measurements missing or discarded because of bad data contamination as long as a small number of redundant voltage measurements are available.

Including Nonlinear Measurements

The Gaussian distribution assumption in the Bayesian network requires linear relationships between random variables. To incorporate load power measurements, we apply local linearization to the relationship between the load power S_j and the load current J_j

$$S_j = \text{diag}(V_j)\bar{J}_j$$

where \bar{x} denotes the complex conjugate of x and V_j, J_j , and S_j are all three-by-one complex vectors. The Forney-style factor graph after the linearization is shown in Figure 1.44. \tilde{X} denotes the measurement of X . Correspondingly, the message-passing schedule is updated as shown in Figure 1.45. The main difference is the first step in the backward sweep, which updates the values in $\text{diag}(\bar{V}_j)^{-1}$ based on the result from previous forward sweep, the voltage measurement, and the messages from children buses. Other steps are similar to the those presented in Figure 1.41.

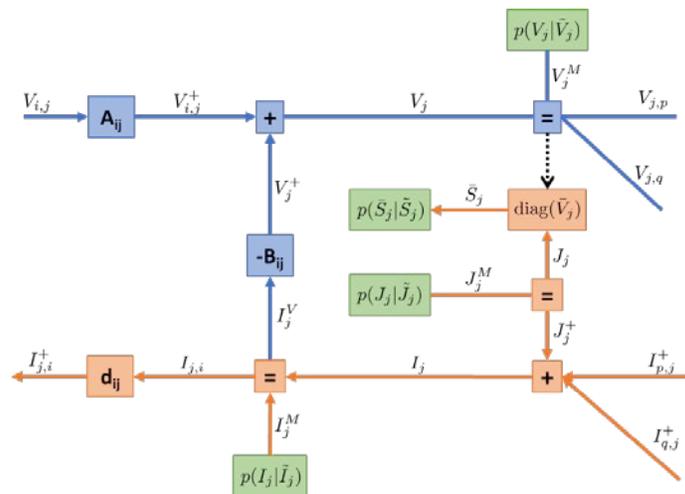


Figure 1.44. Forney-style factor graph of the Bayesian network with load power measurements after local linearization

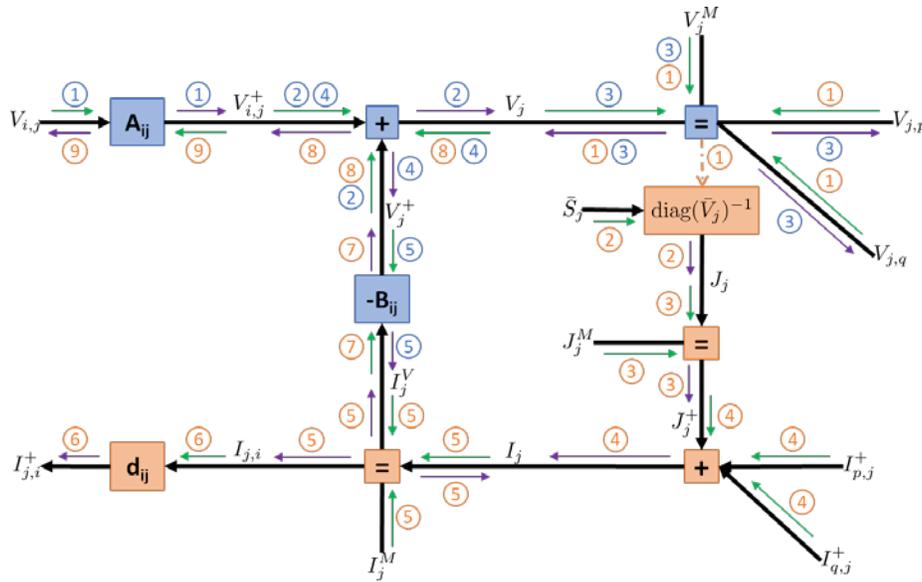


Figure 1.45. Message-passing schedule for LIBP with load power measurements incorporated

Performance Evaluation of the Proposed LIBP Algorithm

We evaluate the performance of our proposed LIBP algorithm in RTC-A with various combinations of parameter settings.

Accuracy improvement by exploiting V measurements

Figure 1.46 shows the RMSE of the LIBP with load power measurements at all nodes and voltage measurements at randomly selected nodes. Compared with the conventional LI algorithm, which cannot take voltage measurements, the LIBP algorithm achieves similar accuracy with no voltage measurements. When increasing amounts of voltage measurements become available, the margin by which the LIBP algorithm outperforms the conventional LI algorithm also increases, especially when the noise is low. The voltage measurements are assumed to be provided by micro PMUs and henceforth have much higher accuracy than load power measurements, which are usually obtained by load forecasting-based pseudo-measurements. The noise standard deviation of voltage measurement σ_V is assumed to be 1% of that of load power measurements σ_S . This is a reasonable assumption because the commercially available micro PMU can achieve 0.01% measurement error [75], whereas errors of pseudo-measurements range from 5% to 30% [23], [25].

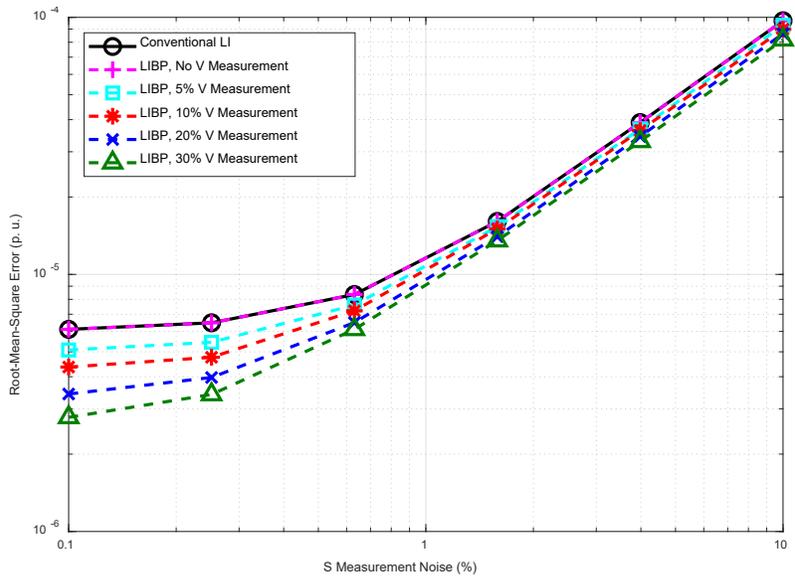


Figure 1.46. Comparison between the conventional LI algorithm and the LIBP algorithm

Performance with Insufficient S Measurements

The performance of LIBP under various measurement configurations is presented in Figure 1.47. In general, the RMSE is lower when more voltage (V) measurements are available; more accurate V measurements yield lower RMSE. We can also observe that when some load power (S) measurements are missing and the SE accuracy is significantly improved with V measurements, even at only 5% of the nodes. This means that a small amount of micro PMUs in the distribution network can significantly improve the system awareness, especially when some load measurements are missing.

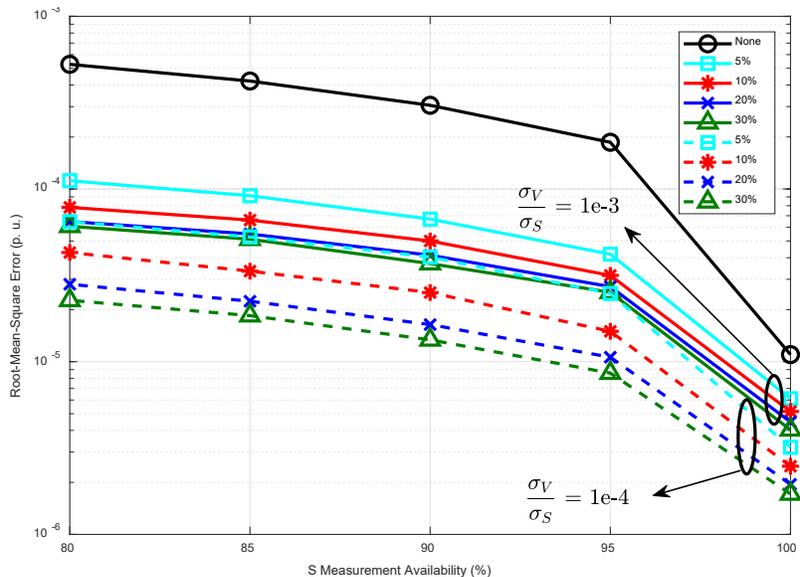


Figure 1.47. Performance of LIBP under various measurement configurations

Performance in the Presence of Bad Data

We further investigate the performance of LIBP in the presence of bad data and compare the performances with that of the conventional LI algorithm. The simulations are grouped into two sets according to different measurement noise levels. The results are presented in figures 1.48 and 1.49, respectively.

In both sets of simulations, we investigate the accuracy of the conventional LI algorithm (which can accept only load power measurements) and the LIBP algorithm (which can exploit load power and bus voltage measurements), with 0%–20% of load power measurements contaminated with bad data. We assume that load power measurements are available at all buses and voltage measurements, though more accurate, are available only at a fraction of buses. For the LIBP algorithm, we investigated its performance with voltage measurements at (1) no bus, (2) 10% of buses, and (3) 20% of buses, respectively. We also consider two levels of bad data magnitude: 100% and 200%.

In figures 1.48 and 1.49, we observe that the LIBP algorithm yields significantly better accuracy when voltage measurements are available. More specifically:

- Even with voltage measurements available at only 10% of the buses, the state estimation RMSE and its slope against bad data percentage are both significantly improved in most cases.
- When voltage measurements are used, the accuracy of the LIBP algorithm only slightly deteriorates when the bad data magnitude increases from 100% to 200%. The RMSE is mainly affected by the noise level of voltage measurements.

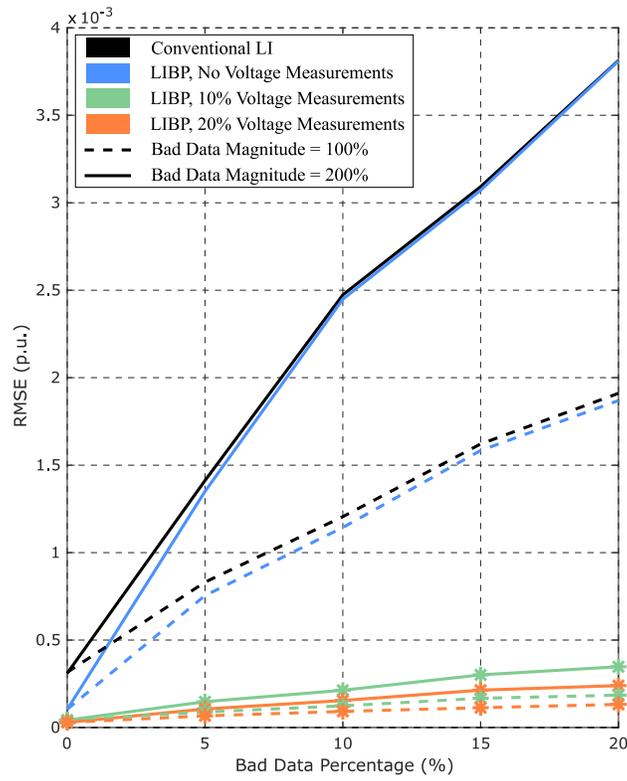


Figure 1.48. RMSE comparisons of conventional LI and LIBP algorithm. $\sigma_S = 5\%$, $\sigma_V = 0.005\%$

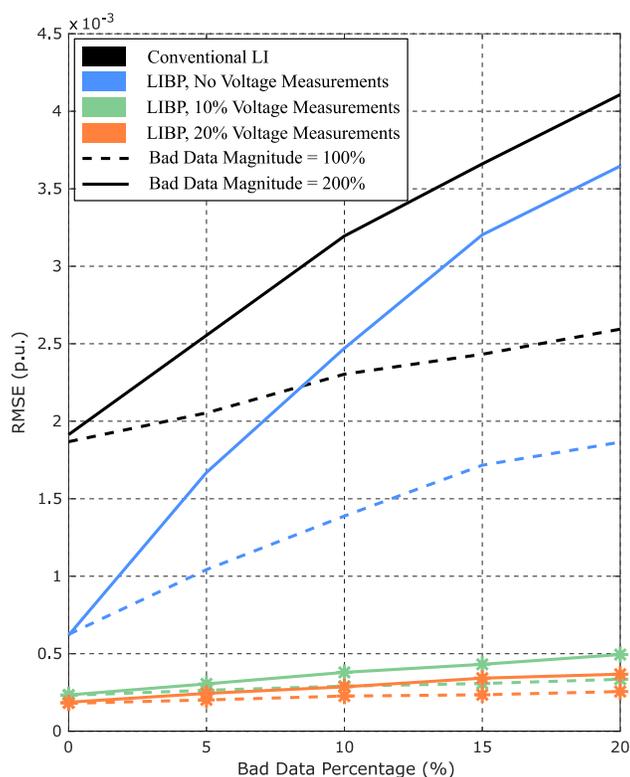


Figure 1.49. RMSE comparisons of conventional LI and LIBP algorithm. $\sigma_S = 30\%$, $\sigma_V = 0.03\%$

Therefore, our proposed LIBP algorithm exhibits strong robustness in the presence of bad data if relatively more accurate voltage measurements are available at a small portion of buses. The LIBP algorithm can exploit redundant measurements of various types, and we have shown that it can still achieve good accuracy even with some missing S measurements. The accuracy of the LIBP algorithms can be further improved if we can identify and remove some of the bad data in the S measurements. In this research, we attempt to remove the BD based on an ARIMA prediction model. This is a mature prediction method, and we do not cover the details of the prediction itself.

The performance of the distributed LIBP algorithm with and without data removal is evaluated with simulations and compared with the centralized algorithm (number of regions = 1). The accuracy comparisons are presented in Figure 28 with different amounts of bad data. In general, LIBP algorithms achieve better accuracy than conventional LI. From the results shown in Figure 1.50(a), LIBP achieves accurate results with or without BD removal in cases where the influence of bad data is relatively small. When the amount of bad data increases, as shown in Figure 1.50(b), conventional LI accuracy deteriorates severely, whereas LIBP errors do not increase as much. BD removal helps further improve accuracy. Figure 1.51 shows that distributed algorithms execute much faster than centralized ones. BD removal sometimes helps the algorithm converge faster. Note that LIBP algorithms deal with more measurements and involve more complicated message-passing operations. This why the LIBP algorithms run much slower than convention LI. Nonetheless, this expansion of the conventional LI algorithm, at the cost of higher complexity, is worthwhile because of the significant improvement of the robustness and accuracy of the state estimates in the presence of bad data and noise.

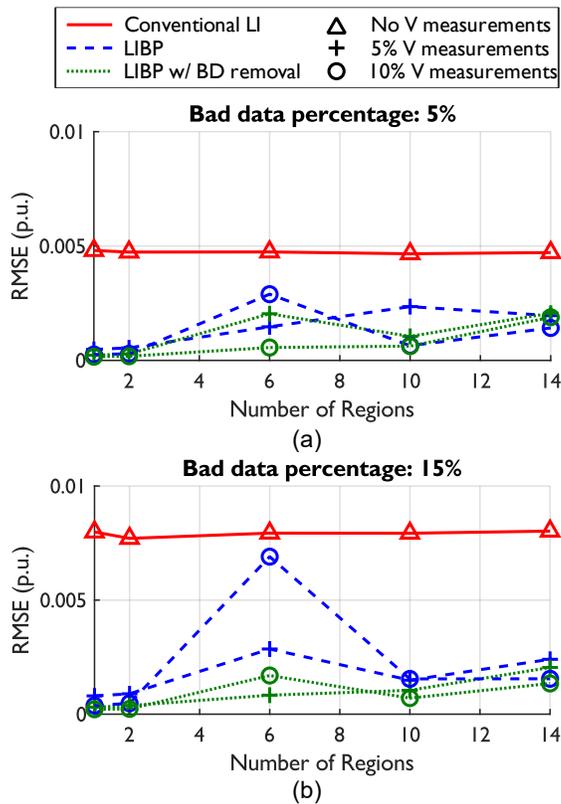


Figure 1.50. Accuracy of centralized and distributed versions of conventional LI and LIBP algorithms. The results of the centralized algorithms are plotted at number of regions = 1.

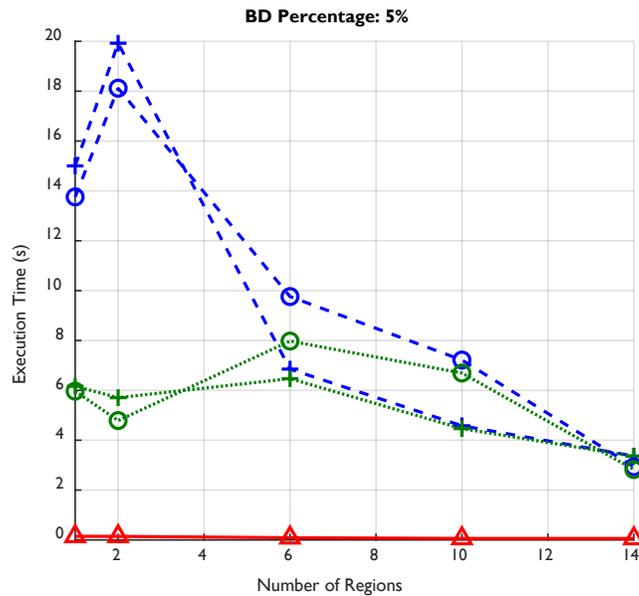


Figure 1.51. Execution time comparisons of centralized and distributed versions of conventional LI and LIBP algorithms. The results of the centralized algorithms are plotted at number of regions = 1.

Subtask 1.3.4: Incorporate Statistics of PV States

Objective: *Incorporate statistical data of PV states to help improve the accuracy of the DSE algorithms. In addition to traditional power system measurements, other types of measurement can assist better monitoring of the activities of PV generation. For example, solar irradiance monitoring and forecasting data from weather stations and spatial-temporal PV state statistics could be used in the PV generation state prediction to successfully capture the variability of PV. One major problem with this type of estimation/prediction is that it is performed in a coarser timescale and captures only slow variations in PV generation. Hence, in the proposed DSE algorithm, we leverage the results of PV state computation from Task 1.2 and develop an asynchronous data fusion mechanism to incorporate data/measurements that become available at various timescales to improve accuracy while maintaining the dynamic capability.*

Our proposed LIBP-based SE algorithm has the capability of incorporating data or measurements that become available at various timescales. Further, PV state computation results from Task 1.2 can be used as input to the LIBP-based SE algorithm with corresponding metadata describing the statistical distribution of the measurements.

Task 1.4: Technical Review Committee

In this task, the team gathers a technical review committee (TRC) populated with members from a broad stakeholder group. The targeted participants are employees of transmission- and distribution-level utilities, ISOs, industry vendors, other national labs, and academia.

Subtask 1.4.1: Stakeholder Community

Objective: *The team invited selected members of the stakeholder community to participate in the TRC. Biannual meetings were envisioned to ensure that the project met the needs of the various stakeholders and conformed to practical utility constraints. The first two meetings occurred during Budget Period 1.*

A conference call was set up with a group of stakeholders involved in similar research and practical applications of the Opportunistic Hybrid Communications Systems for Distributed PV Coordination Project. The conference call established the objectives of the research and preliminary results. At the conclusion of Year 1, another conference call, or in-person meeting, commenced to provide feedback on Year 1 progress; the second meeting happened in September 2016.

Task 2.1: Robustify Decentralized Imputation and Prediction Algorithms

The state estimation algorithms developed in Phase 1 focus on algorithmic performance measures such as accuracy and computation time. This task prepared the algorithms developed for real-world application by incorporating methods for handling missing and corrupted data, such as would be seen in a real system. The decentralized state estimation algorithms were further developed to perform, even when faced with bad or missing data, as well as outliers. This was an important development for application to real systems where perfect conditions and information are the exception.

Subtask 2.1.1: Develop Robust Machine Learning Algorithms for State Estimation

Objective: Improve the state estimation algorithm performance in the presence of real-world data by applying advanced statistical techniques such as Kriged Kalman Filtering and dictionary learning.

PV Power Output Profile for the Reference Test Case A

The distributed estimation algorithm for power distribution system (DiLISE) requires the knowledge of PV states, which are being dynamically estimated using the MREDRIKK filter; however, the “Kriging” part of a single MREDRIKK filter can take care of a small geographic area having very high level of spatial correlation. This necessitates a strategy to define the local spatial PV footprint for a typical distribution network that possibly experiences a different irradiance pattern over the larger geographic area. Therefore, a detailed profile of geographically distributed PV power is built for the Reference Test Case A (RTC-A). At present, only 17 locations from DeSoto, Florida, [50] can be used for sample measurements over the RTC-A (Figure 2.1).

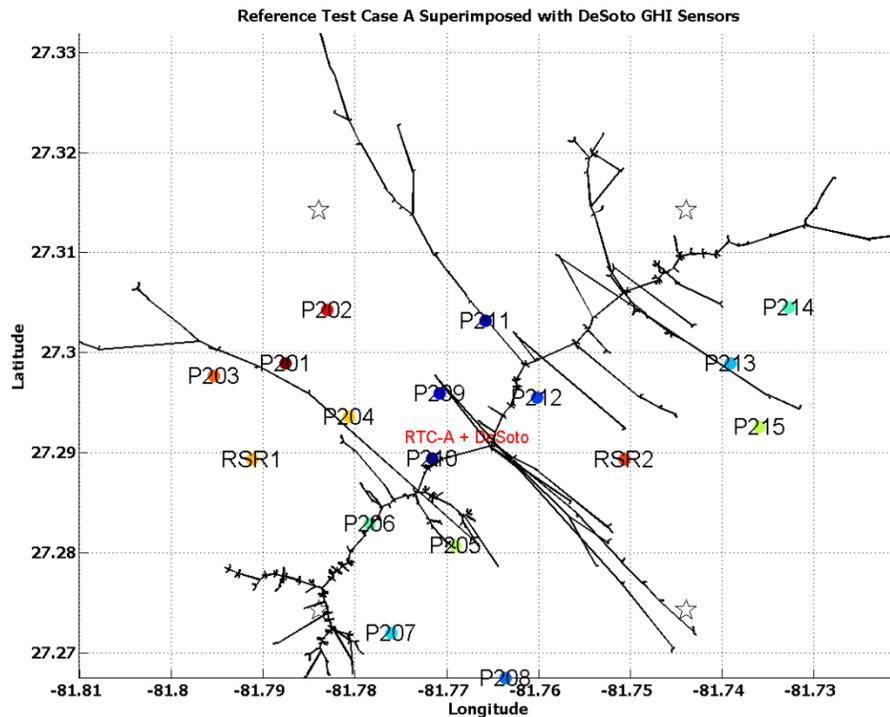


Figure 2.1. RTC-A collocated with 17 DeSoto, Florida, irradiance sensors. Stars indicate the NSRDB data points.

The overall process of creating the PV power footprint is summarized in Figure 2.2.

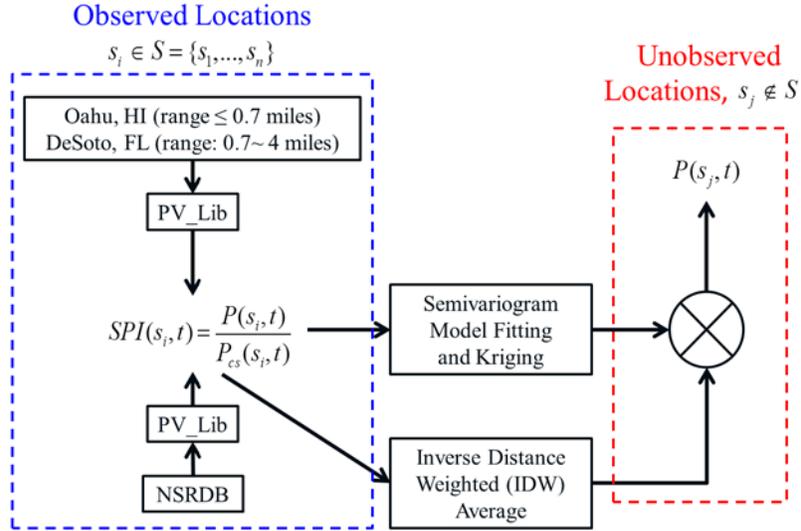


Figure 2.2. Kriging-based wide-area PV footprint formation

In the first step, we start with the PV_LIB toolbox from Sandia National Laboratories [51] to calculate the AC power output with the assumptions that (1) the installed PV system at each home is 2.5 kW, and (2) the inverter loading ratio is 1.48. The toolbox requires irradiance and weather (air pressure, temperature, and wind speed) information, which are collected from Oahu, Hawaii, [52] and DeSoto, Florida [50], representing small (≤ 0.7 miles) and large (0.7 ~ 4 miles) spatial dispersion, respectively. The AC powers thus obtained are normalized by the clear-sky AC powers at respective locations. We name this normalized value the solar power index (SPI). Figure 2.3 illustrates the inverter outputs and SPI obtained on a typical day. Note that the NSRDB database [53] and the clear-sky AC power at each DeSoto sensor and RTC-A location is obtained by using the inverse distance weighting (IDW) [54]. The IDW clear-sky AC power at any location \mathbf{s}_0 over RTC-A is given by:

$$P_{CS}(\mathbf{s}_0, t) = \frac{\sum_{i=1}^4 P_{CS}(s_i, t) / |\mathbf{s}_0 - \mathbf{s}_i|^2}{\sum_{i=1}^4 1 / |\mathbf{s}_0 - \mathbf{s}_i|^2} \quad (2.16)$$

Here, the set of four locations $\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4\}$ are the NSRDB grid locations denoted as a “star” on Figure 2.1. The AC power ratios (SPI) from Oahu and DeSoto are then used to fit the following exponential model of semivariogram [55]:

$$\gamma(\mathbf{d}) = \begin{cases} \mathbf{0}, & \mathbf{d} = \mathbf{0} \\ \alpha + \beta\{1 - \exp(-\theta \mathbf{d})\}, & \mathbf{d} \neq \mathbf{0} \end{cases} \quad (17.2)$$

Here, the semivariogram, $\gamma(\mathbf{s}_i, \mathbf{s}_j) = 0.5 \times \text{Var}\{y(\mathbf{s}_i) - y(\mathbf{s}_j)\}$. The fitted model is shown in Figure 2.4.

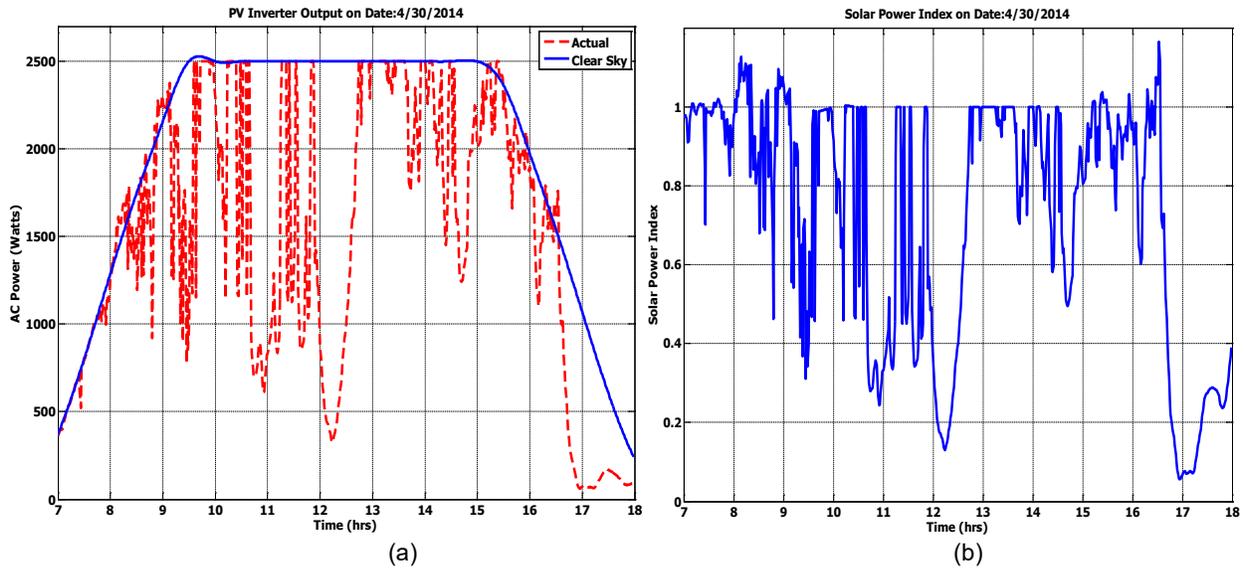


Figure 2.3. (a) Example AC power output of PV inverter, (b) calculated solar power index

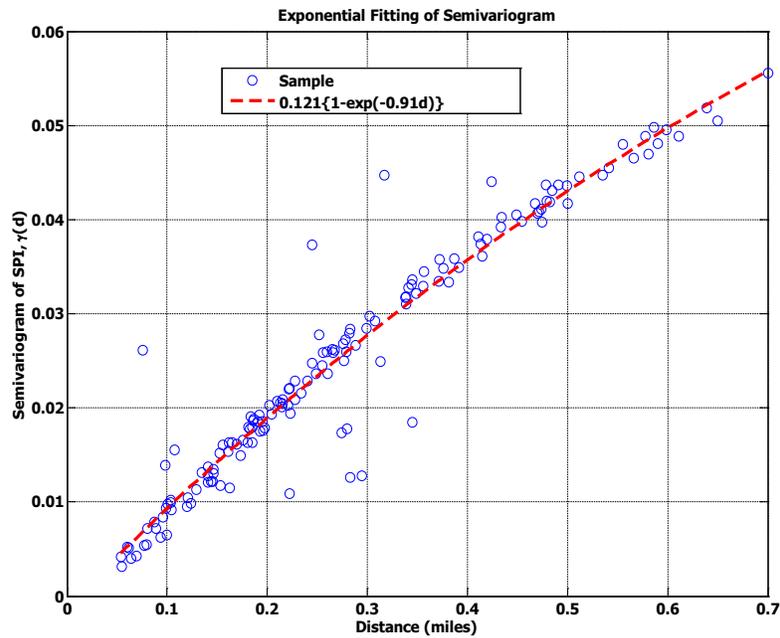


Figure 2.4. Fitting of exponential semivariogram model

The spatial covariance can then be obtained as:

$$C_{ij} = Covar(s_i, s_j) = \alpha + \beta - \gamma(s_i, s_j); 1 \leq i, j \leq 17 \quad (2.18)$$

We use this spatial covariance model to estimate PV power at a given location on RTC-A using the concept of “Kriging” [56] . Given any unobserved location s_0 over RTC-A, the Kriging weights $\{\lambda_k\}$ are obtained by solving the following equation for $n = 17$:

$$\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{bmatrix} \quad (2.19)$$

Therefore, the Kriged SPI at location \mathbf{s}_0 , $\widehat{SPI}(\mathbf{s}_0, t) = \sum_{k=1}^{17} \lambda_k SPI(\mathbf{s}_k)$. At the final step, the Kriged SPIs are multiplied by the clear-sky AC power to obtain the estimate of the PV inverter AC power output at unobserved locations. Assuming 100% solar penetration, we can thus create a 1-minute resolution PV power footprint of 11-hour duration.

In the next step, we apply k-means clustering to observe the PV power-based regionalization of RTC-A. The synthesis and regionalization of such a wide-area PV footprint helps us validate the integrated approach of the MREDRIKK filter and ADMM for distributed and dynamic estimation of PV system states. As a comparison, we also perform the same level of clustering based on the network information of RTC-A [57]. The findings are shown in Figure 2.5. It can be observed that the same power distribution network is regionalized differently based on the geographic diversity of the PV systems and the electric network itself.

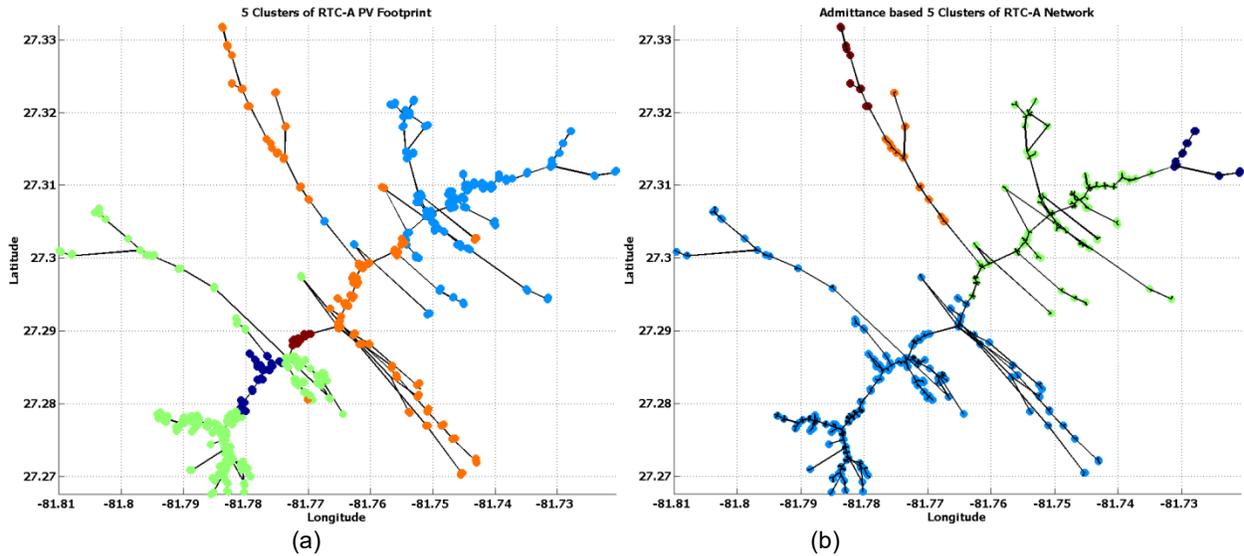


Figure 2.5. RTC-A clustering into five regions: (a) PV footprint, (b) network admittance

We consider this aspect in synchronously integrating the dynamic PV state estimation and static power system state estimation. Mathematically, the power system state at discrete time instance t , \mathbf{x}_t^{SYS} follows the measurement model:

$$\begin{bmatrix} \mathbf{y}_t^{SYS} \\ \hat{\mathbf{x}}_t^{PV} \end{bmatrix} = \mathbf{h}(\mathbf{x}_t^{SYS}) + \mathbf{v}_t^{SYS} \quad (2.20)$$

Here, $\hat{\mathbf{x}}_t^{PV}$ represents the PV system states, which are dynamically estimated using MREDRIKK filter. \mathbf{v}_t^{SYS} is the aggregation of (1) power system measurement error and (2) PV system estimation error as reflected by the estimation error covariance.

This integration is conceptualized in Figure 2.6. As evident, the distributed static estimation algorithm needs to be fast enough to support the real-time operation.

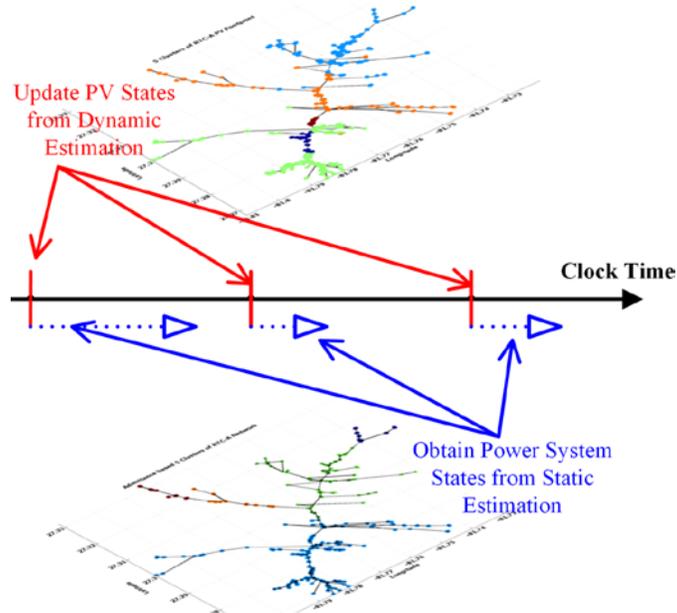


Figure 2.6. Synchronous integration of static and dynamic state estimation

Subtask 2.1.2: Create a Data Screening Tool for Distributed State Estimation

Objective: Improve system performance through enhancing the accuracy of the state estimation algorithm by filtering out missing or outlier data without sacrificing computational time.

Distributed LIBP Algorithm Implementation in C++

We implemented the distributed LIBP algorithm in C++. The matrix calculation/manipulation is handled by the Armadillo C++ library. The network communications are implemented using the Boost Asio library. A more specific description of the environment of the implementation is presented in Table 2.1.

Table 2.7. Implementation Environment

Platform	Ubuntu 18.04 LTS
Compiler	GNU C Compiler 6.4
Libraries	Armadillo 8.400.0
	Boost 1.67

Two programs are developed in the process: ddse_manager and ddse_worker. For a distribution system partitioned into N regions, one manager and N workers are required. The manager is responsible for initialization, receiving and monitoring the status (the maximum change of

voltages in each iteration) of workers during the execution of the algorithm, and sending termination signals to workers. The workers are responsible for the actual computation, the message exchange with peer workers, and sending status updates to the manager. The basic flow of the algorithm is illustrated by Figure 2.7. Workers i and j are shown, and others are omitted for clarity of presentation.

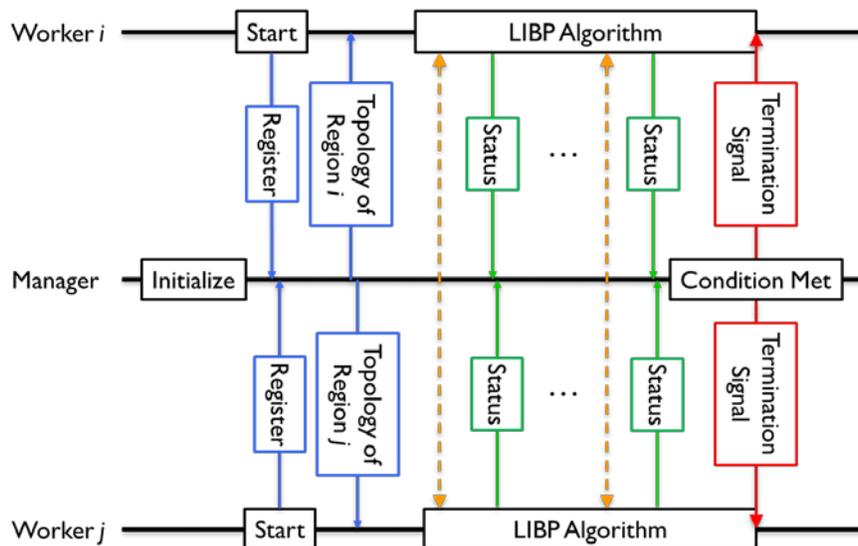


Figure 2.7. Basic flow of the distributed LIBP algorithm

To run a distributed state estimation procedure:

1. The manager starts first for initialization.
2. When the manager is ready to accept registrations, the workers can be started at regional processors.
3. Once the manager receives the registration from a worker, it sends the topology (and measurements) of the corresponding region back to the worker.
4. Once the worker receives the topology, it starts the actual computation.
5. During the computation, the workers communicate with peer workers and send the maximum voltage change to the manager once in each iteration.
6. The manager receives and monitors the update messages from the workers.
7. When the maximum voltage changes of all regions fall below the tolerance threshold, the manager sends termination signals to all workers.
8. When a worker receives the termination signal, it sends the results back to the manager and terminates the local program.
9. The manager terminates itself when all results are received.

The performance of the programs in C++ is consistent with that in MATLAB in our initial testing.

Subtask 2.1.3: Develop Decentralized Solvers for the Optimization Problems Formulated Under Subtasks 1.2.1 and 1.2.2.

Objective: To this end, leverage primal-dual algorithms and alternating direction method of multipliers (ADMM) to identify saddle-flow points of augmented Lagrangian functions of pertinent optimization problems associated with the imputation and prediction tasks. The dynamic characteristics of solar irradiance lead to the implementation of Kalman filtering. The particle filter can be a solution. The cloud movement over time gives rise to different zones, and thereby distributed Kalman filtering needs to be implemented; however, the states might be overlapped among the various spatial correlation-based clusters. The ADMM will take care of the agreement of inter-region shared PV states. One of the challenges is the temporal evolution of the regions due to the weather condition. Consequently, the spatial footprint as well as the set of inter-region shared state variables will have some level of temporal behavior.

The distributed and dynamic estimation of PV system states can be formulated as a decentralized optimization problem as follows. Let the wide-area PV footprint be clustered into K regions. For each region k , the set of sensors collecting measurements is denoted by \mathcal{S}_k . Thus, the K -region PV system has the following state-space model:

$$\mathbf{x}_{t,k} = \mathbf{F}_k \mathbf{x}_{t-1,k} + \mathbf{w}_{t,k}; \mathbf{w}_{t,k} \sim N(\mathbf{0}, \mathbf{Q}_k) \quad (2.6)$$

$$\mathbf{y}_t(\mathcal{S}_k) = \mathbf{H}_k \mathbf{x}_{t,k} + \mathbf{v}_{t,k}; \mathbf{v}_{t,k} \sim N(\mathbf{0}, \mathbf{R}_k) \quad (2.7)$$

Therefore, the optimization problem to get the minimum mean square error (MMSE) estimate at each region:

$$\begin{aligned} \min_{\{\mathbf{z}_k \in \mathcal{Z}_k\}} \sum_{k=1}^K \mathbb{E}[(\mathbf{x}_{t,k} - \mathbf{z}_k)(\mathbf{x}_{t,k} - \mathbf{z}_k)^T] \\ \text{Subject to, } \mathbf{z}_k[l] = \mathbf{z}_l[k]; \forall l \in \mathcal{N}_k, \forall k. \end{aligned} \quad (2.8)$$

Here, \mathcal{N}_k denotes the set of regions sharing states with the region- k . The equality constraint represents the overlapping among local state vectors, i.e., the l^{th} entry of region- k state vector equals the k^{th} entry of region- l state vector. The decentralized optimization problem with such equality constraint can be solved using the ADMM [60]. Specifically, with the help of augmented Lagrange-based Proposition 1 of [60], the optimization problem can be represented as a nested loop algorithm consisting of the local MREDRIKK filter in the inner loop and a weighted averaging of exchanged information in the outer loop. We define the following quantities for better representation of the algorithm:

- $\mathbf{x}_k[i]$: The i^{th} entry of \mathbf{x}_k
- \mathcal{N}_k^i : Set of regions sharing $\mathbf{x}_k[i]$. This is an empty set with $|\mathcal{N}_k^i| = 0$ for strictly local state elements.
- $\mathbf{x}_l[i]$: The entry of \mathbf{x}_l corresponding to $\mathbf{x}_k[i]$, while $l \in \mathcal{N}_k^i$
- $\mathbf{p}_k[i]$: Zero entry for $\mathcal{N}_k^i = \emptyset$ (will be forced in (6)) and nonzero otherwise.
- $\mathbf{D}_k = \text{diag}(|\mathcal{N}_k^i|)$.

Therefore, in each region k , the strictly local MREDRIKK filter estimates:

$$\hat{\mathbf{x}}_{t,k|t} = \arg \min_{\mathbf{z}_k} \left(\mathbb{E} \left[(\mathbf{x}_{t,k} - \mathbf{z}_k)(\mathbf{x}_{t,k} - \mathbf{z}_k)^T \right] + \frac{c}{2} \mathbf{D}_k (\mathbf{p}_{t-1,k} - \mathbf{z}_k)(\mathbf{p}_{t-1,k} - \mathbf{z}_k)^T \right) \quad (2.9)$$

The outer loop steps are:

- Information exchange and averaging: $\mathbf{s}_{t,k}[i] | (\mathcal{N}_k^i = \emptyset) = 0$. Otherwise:

$$\mathbf{s}_{t,k}[i] | (\mathcal{N}_k^i \neq \emptyset) = \frac{1}{D_k[i,i]} \sum_{l \in \mathcal{N}_k^i} \hat{\mathbf{x}}_{t,l|t}[i] \quad (2.10)$$

$$\mathbf{p}_{t,k} = \mathbf{p}_{t-1,k} + \mathbf{s}_{t,k} - \mathbf{0.5}(\hat{\mathbf{x}}_{t-1,k|t-1} + \mathbf{s}_{t-1,k}) \quad (2.11)$$

Task 2.2: Develop Communications System Simulation Model

This task develops both the software simulation and real-time HIL simulation models to validate and test the reliability, efficiency, and scalability of the proposed communications system architecture. To test interactions with a simulated power grid, such as assessing the impacts of the developed communications architecture on the physical power system, the communications system simulator, NS-3, is integrated with the Hierarchical Engine for Large-scale Infrastructure of Co-Simulation (HELICS) platform.

This project aims to develop a novel opportunistic hybrid communications system architecture that enables the reliable and efficient real-time monitoring, control, and optimization of large-scale transmission-distribution power systems with millions of distributed PV generators. It leverages the existing communications infrastructure to achieve such hybrid communications architecture. To this end, we proposed a design framework of opportunistic hybrid communications systems, shown in Figure 2.8. The proposed framework provides a systematic view of the overall design procedure of the future communications systems for distributed PV coordination along with three stages: (1) initial optimal data concentrator placement design, (2) main opportunistic and hybrid feature design, and (3) final network parameter tuning stage. For the data-concentrator node placement optimization algorithm at the first stage, the known input topology consists of three main components: (1) the physical power system topology information including feeder lines and distributed PV inverters; (2) existing communications infrastructure containing smart meters, existing data concentrators, and edge router; (3) candidate data concentrators. And the output of the algorithm gives the optimal data concentrator placement and their corresponding regionalization of PV inverters and smart meters. Then, the NS-3-based hybrid communications models and middleware-based network manage schemes were developed to implement two unique features of the proposed communications systems in the subsequent stage. Further, the optimal parameter tuning box is developed to allow for exhaustive testing of predefined scenarios for a given network topology. The first stage design and part of the second stage were completed and reported in Phase 1. We introduce the left two stages about the development of NS-3 based communications system simulation model as follows.

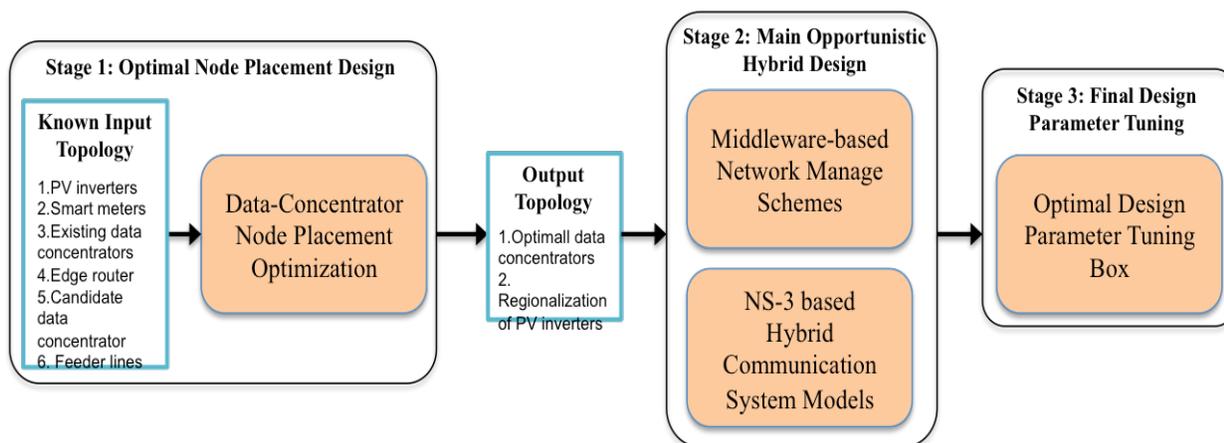


Figure 2.8. Design framework of opportunistic hybrid communications systems

Subtask 2.2.1: Opportunistic and Hybrid Communications Software Simulation Models

Objective: *Develop the communications system software simulation models. These allow for testing critical system criteria, such as scalability, availability, and response time.*

The objective of this subtask was to develop the software simulation models for the proposed communications systems in Year 1, with emphasis on two distinguished features: opportunistic and hybrid. In modern power systems, utility communications system are designed and deployed in the form of a core-edge network [25]. In this way, a wide-area network (WAN), usually based on fiber optics, forms the backbone of the system, whereas the connections between end devices and the WAN are established through neighborhood area networks (NAN), and at the end, all end devices—including home appliances, storage batteries, renewable generators, and smart meters—form multiple home area networks (HAN) as the envisioned communications systems of a smart grid, described in Phase 1. Hence, it is not common for an individual end device in a HAN to be directly connected to the control center local area network (LAN), and all connections from end devices to control centers at different portions of the power grid must eventually go through the existing well-developed WAN, which is either dedicated or public. As such, our focus is narrowed to designing the communications network that enables data transmission between the individual PV panels and the first WAN router named the edge router. Typically, this communications network would be hierarchical and would consist of:

- *Home area network:* A HAN connects the PV panel to the smart meter located at the customer house serving as the gateway to the utility’s network. The geographic size of a HAN would be up to tens of meters.
- *Neighborhood area network:* A NAN collects the data from multiple smart meters and transmits it to the WAN through a WAN edge router. The geographic size of this network depends on the topology of the distribution system and the distances among the houses. It could range from hundreds of meters to several kilometers.

Various communications protocols and technologies exist that can be used for data communications in power systems [1], [26], [27]. To enable data transmission between the individual PV panels and the WAN edge router, the envisioned communications network comprises three different data flows: (1) PV inverter - smart meter, (2) smart meter - data concentrator, (3) data concentrator - edge router. The proper technologies for each data link have been investigated and chosen based on the level of maturity of the technology, being open source and nonproprietary, and offering a sufficient data rate. Different scenarios have been shown and are briefly discussed in Table 1.1. There are two alternative communication technologies—Low-power Wireless Personal Area Networks (LoWPAN) and Power Line Communication (PLC)—between PV inverters and smart meters in a home area network (HAN). The communications between smart meters and data concentrators in a neighborhood area network (NAN) also have three alternatives: WiFi, Ethernet cable, and WiMAX. Note that we choose LoWPAN instead of the well-known ZigBee because ZigBee cannot easily interoperate with other protocols and it is not suitable to design the hybrid communications systems. Considering these alternative communications technologies, we are intuitively interested in not only the network performance comparison of hybrid communications architecture designs but also how the intelligent middleware-based network management schemes effectively improve the overall performance of such hybrid architectures. To this purpose, we next develop opportunistic and hybrid communications system software simulation models to verify the critical system design criteria.

Hybrid Communications Simulation Models

In this section, we focus on the development of the software simulation models that represent the envisioned opportunistic hybrid communications systems to test and verify the critical system design criteria, such as *feasibility*, *scalability*, and *reliability*. To this purpose, the simulation tool NS-3 is chosen for its popularity, open source, and already available modules for numerous networking functionalities [28], [29], [30]. With alternative communications technologies for both home area networks and neighbor area networks described in the preceding section, we consider the cases when these five candidate technologies are designed in combination. Six possible such hybrid communications architectures, shown in Table 2.2, are considered. We next describe the topology design and development of prototypical hybrid simulation models in NS-3 for each hybrid architecture.

Table 8.2. Hybrid Communications Architectures

Hybrid Type	Home Area Network	Neighborhood Area Network
Hybrid-1	LoWPAN	Ethernet cable
Hybrid-2	LoWPAN	WiFi
Hybrid-3	LoWPAN	WiMax
Hybrid-4	PLC	Ethernet cable
Hybrid-5	PLC	WiFi
Hybrid-6	PLC	WiMax

Topology Design

From the earlier discussions on existing utility communications infrastructures and distributed PV coordination application, the expected choices for the network topology would be predominantly the tree and mesh structures. For a HAN typically consisting of one or two PV inverters and a smart meter, we have the obvious choice of tree topology without regarding which communications technology is selected.

The topology design of a NAN, however, usually containing up to hundreds of smart meters and tens of data concentrators, is mainly affected by the fact that the choice of network topology is driven more by the convergence of communications technologies than by the traffic volume. Although a NAN consists of two types of data flows—smart meter - data concentrator and data concentrator - edge router—we consider only the technology selection for links between smart meters and data concentrators because there is one option of optical Ethernet for the link of data concentrator - edge router, shown in Table 1.1.

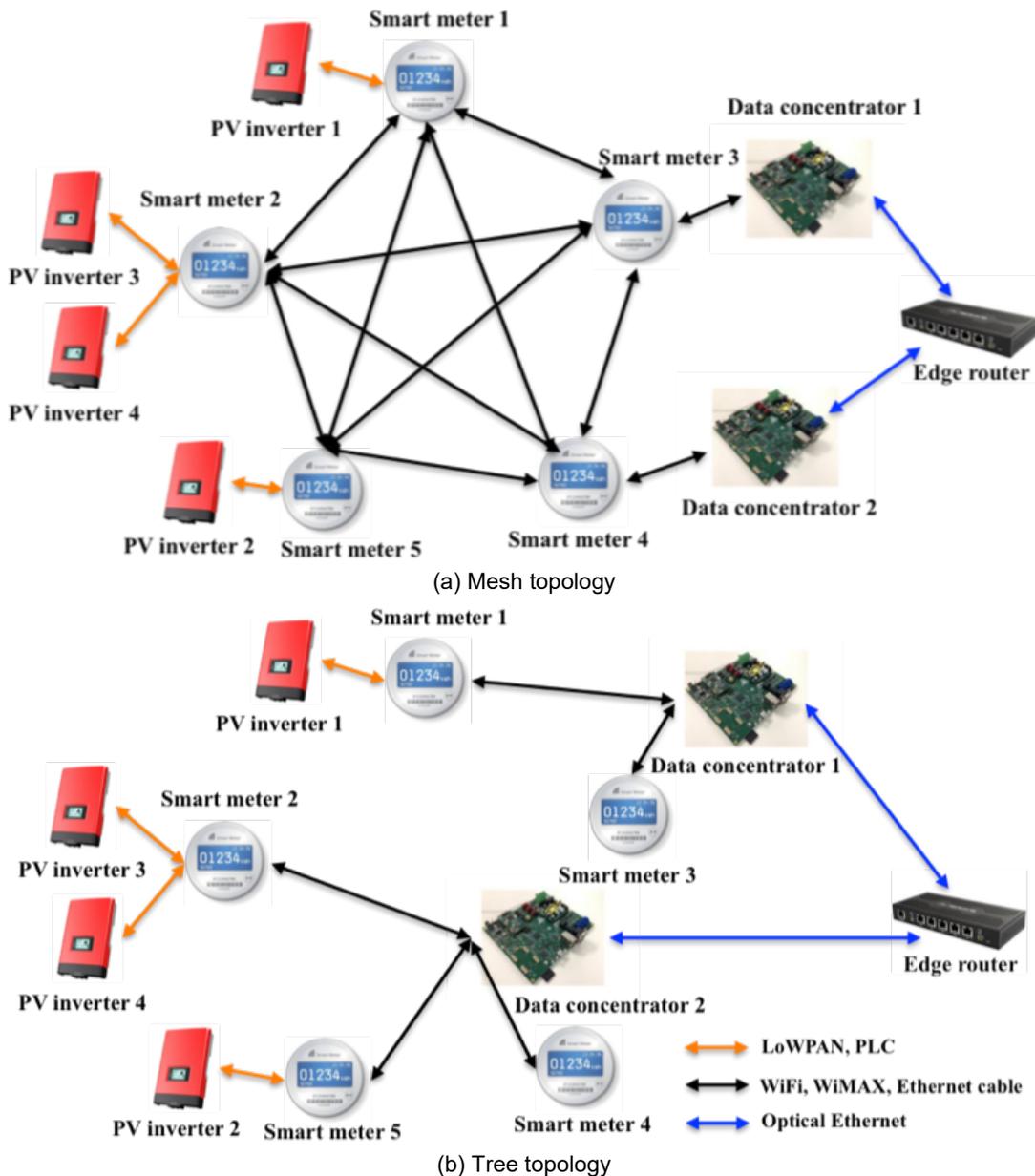


Figure 2.9. Alternative topologies for opportunistic hybrid communications systems

From the column of *Coverage* in Table 1.1, the wireless technology of WiFi using the IEEE 802.11s has the shortest transmitting range of 30 m– 1 km, compared to the WiMAX and Ethernet cable. To cover the geographic range of up to 10 km of a NAN, we choose the mesh structure for WiFi to provide cost-effective and dynamic high-bandwidth networks over a specific large coverage area. Therefore, the expected mesh topology for both Hybrid-2 and Hybrid-5 is shown in Figure 2.9(a), where all smart meters and data concentrators are connected in the mesh topology. Regarding Ethernet technology, we intuitively select the tree topology for Hybrid-1 and Hybrid-4 because of its unlimited coverage with wired cables, shown in Figure 2.9(b), where the whole hybrid communications system is demonstrated as a typical tree topology.

Further, considering WiMAX, the IEEE 802.16 standard defines two possible network topologies: (1) PMP (point-to-multiple-point) topology, in which traffic may take place between a Base Station (BS) and its Subscriber Stations (SSs); and (2) mesh topology. With this mode, the traffic can be routed through other SSs until the BS and even take place only between SSs. Although the mesh mode offers a major advantage in that the reach of a BS can be much greater, we use the PMP mode because (1) WiMAX's coverage of 50 km, shown in Table 1, enables the decent reachability of the envisioned NAN with several kilometers; and (2) the official latest release of NS-3 does not support the mesh topology function yet in the WiMAX module. As such, the WiMAX-based Hybrid-3 and Hybrid-5 topologies are also designed as the tree type of Figure 2.9(b).

Prototypical Hybrid Simulation Models

The main challenge of developing NS-3-based hybrid communications simulation models is to integrate different communications technologies and IP address mechanisms into one simulation network. To address this challenge, we start with the development of prototypical hybrid simulation models for each proposed hybrid architecture to validate the primary hybrid system design criteria of *feasibility*.

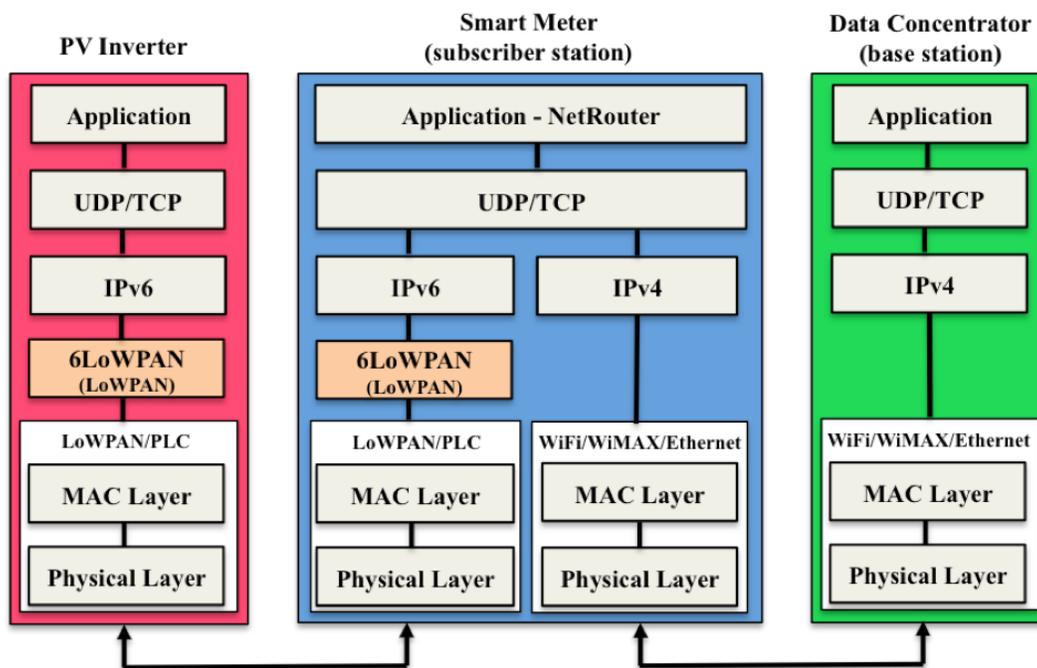


Figure 2.10. NS-3 simulation models of hybrid communications architectures

For the simplification of such a prototypical model, we consider that it consists of 1 PV inverter, 1 smart meter, and 1 data concentrator along with the P2P (point-to-point) topology. The PV data generated at the PV inverter is sent to the smart meter through LoWPAN/PLC, and the smart meter relays the packet to the data concentrator through WiFi/WiMAX/Ethernet, shown as two black double-headed lines on the bottom of Figure 2.10. Also, the schematic in Figure 16 presents the components of the network Open Systems Interconnection (OSI) model in each communications node: physical layer, MAC layer of data link layer, network layer, transport

layer, and application layer. Different communications technology are mainly characterized by (1) physical layer, which defines the electrical and physical specifications of the data connection and the relationship between a device and a physical transmission medium (e.g., a copper or fiber-optic cables, radio frequency); (2) MAC layer, which is responsible for controlling how devices in a network gain access to a physical medium and permission to transmit data. We next describe these two lowest layers, grouped into one white box shown in Figure 16, in terms of protocols and communications nodes, followed by network layer, transport layer, and application layer, respectively.

Physical- and MAC-Layer Attributes

In the PV inverter node of Hybrid-1 to Hybrid-3, the NS-3 model of *LR-WPAN* (Low-Rate Wireless Personal Area Network) is employed to implement the LoWPAN specified by IEEE standard 802.15.4. Its physical layer consists of a Phy model, an error rate model, and a loss model. And its MAC layer implements the unslotted CSMA/CA variant without beaconing. We configure them as default values. Note that IEEE 802.15.4 specifies only the physical layer and media access control, thus it cannot make the IP packet run over the LoWPAN network. Therefore, the *6LoWPAN* model with function of transmission of IPv6 packets over IEEE 802.15.4 networks needs to be employed as an agent between the lowest two layers and the network layer, shown in the orange block of Figure 2.10. It plays a critical role to enable IEEE 802.15.4 to cooperate with other protocols. We use the *PLC* model developed by researchers of the University of British Columbia in the PV inverter node of Hybrid-4 to Hybrid-6 [29]. It is an NS-3-based implementation to simulate the signal propagation in Power Line Communication (PLC). Compared to the *LR-WPAN* model, we need to specify the following physical- and MAC-layer attributes: spectrum model, transmit power spectral density, cable types, background noise, channel, and outlets. These configurable attributes enable us to improve the performance of the PLC networks by tuning them.

As a relay, the smart meter plays an essential role of enabling the integration of two different communications technologies through equipping with two separate sets of network devices and physical transmission mediums in practical networks. Correspondingly, the smart meter node is required to configure two sets of software net devices for both the MAC layer and physical layer. As shown in the Smart Meter block, the left white box containing LoWPAN/PLC protocols is for communications with the PV inverter node, and the right one containing WiFi/WiMAX/Ethernet protocols is used to connect with the data concentrator node. To enable data transferring between these two software net devices, the *Forwarding* function of *IPv6* model needs to be set “on” in all six hybrid models.

In the data concentrator node, we implement three alternative protocols. For WiFi with the proposed mesh topology, we use the *WiFi* and *Mesh* models. The *Mesh* model extends the *WiFi* model to provide mesh networking capabilities according to the IEEE 802.11s standard. These two models allow us to configure the parameters including the physical WiFi protocol, channel propagation model, mesh protocol, remote station manager, spread interface channels, and Mac layer type. Next, through the *WiMAX* model, we configure the smart meter as a Subscriber Station and the data concentrator as a Base Station in the cases of Hybrid-3 and Hybrid-6. Among four scheduling services defined by the IEEE 802.16 standard, Unsolicited Grant Service (UGS) is chosen because it means low latency and low jitter in the proposed hybrid

communications network [30]. At last, the basic *CSMA* model is applied to model a simple bus network in the spirit of the Ethernet along with setting these channel attributes of data rate and delay.

Network Layer: IPv6 to IPv4

As mentioned, the *LR-WPAN* model along with the *6LoWPAN* model supports only IPv6 in the network layer. Both the *Mesh* and *WiMAX* models support IPv4 in the current version of the NS-3 simulator. To address this limitation, we designed a *NetRouter* forwarding function in the application layer, as illustrated in the smart meter node of Figure 16. This model is designed to decapsulate and encapsulate both IPv6 and IPv4 packets of the application layer and to realize the successful integration of IPv6 and IPv4 networks.

Transport and Customized Application Layer

We choose both TCP and UDP protocols at the transport layer for all three nodes. To accommodate the specific distributed PV coordination application, we developed a customized *Client* model where we can set the packet size and sending rate to mimic the real PV data packet. Further, the scalability and modularity of hybrid communications system simulation models is further improved by designing a *Server* model responsible for autonomous online tracing and data post-processing—namely, statistically collecting the network performance metrics, such as latency, max latency, and throughput. To achieve this objective, we added a QoS (quality of service) header at the application layer. This header is capable of carrying the information of client ID and starting the time stamp. The client ID is used to identify the data flow, and the time stamp is used to track the QoS information such as latency and throughput in the server side.

Hybrid Communications System Simulation of RTC-A

Based on six preceding developed prototypical hybrid simulation models, we further developed a full NS-3-based communications simulation test bed on top of RTC-A and validated the second primary hybrid system design criteria of *scalability*. The communications infrastructure of RTC-A consists of 57 PV inverters as yellow dots, 275 smart meters as both yellow and green dots, 10 data concentrators of red dots, and 1 edge router as a black dot, shown in Figure 2.11. The RTC-A is divided into 10 subareas in terms of the location of 10 data concentrators, shown in Figure 2. In the implemented simulation test bed, the PV inverters communicate with smart meters via LoWPAN or PLC, and the smart meters communicate with the data concentrators via WiFi mesh, WiMAX, or Ethernet cable.

Figure 2 of the topology of RTC-A shows that each home area network (HAN) usually consists of only 1 or 2 PV inverters and 1 smart meter, whereas each neighborhood area network (NAN) always consists of around 26 smart meters and 1 data concentrator. It indicates that each HAN has 1 or 2 PV-SM links without the scalability issue. As such, our focus is narrowed to the scalability problem of SM-DC links in the 10 NANs. For the case of Ethernet cable, the solution of the scalability problem is straightforward by installing the declared *CSMA* model instance into a large number of nodes, which does not work for both WiFi mesh and WiMAX cases. To address this challenge, we find that we need to declare new WiFi/Mesh and WiMAX model instance for each subarea. Figure 2.11 shows the packet flow animation for the Hybrid-2 case of LoWPAN-WiFi.

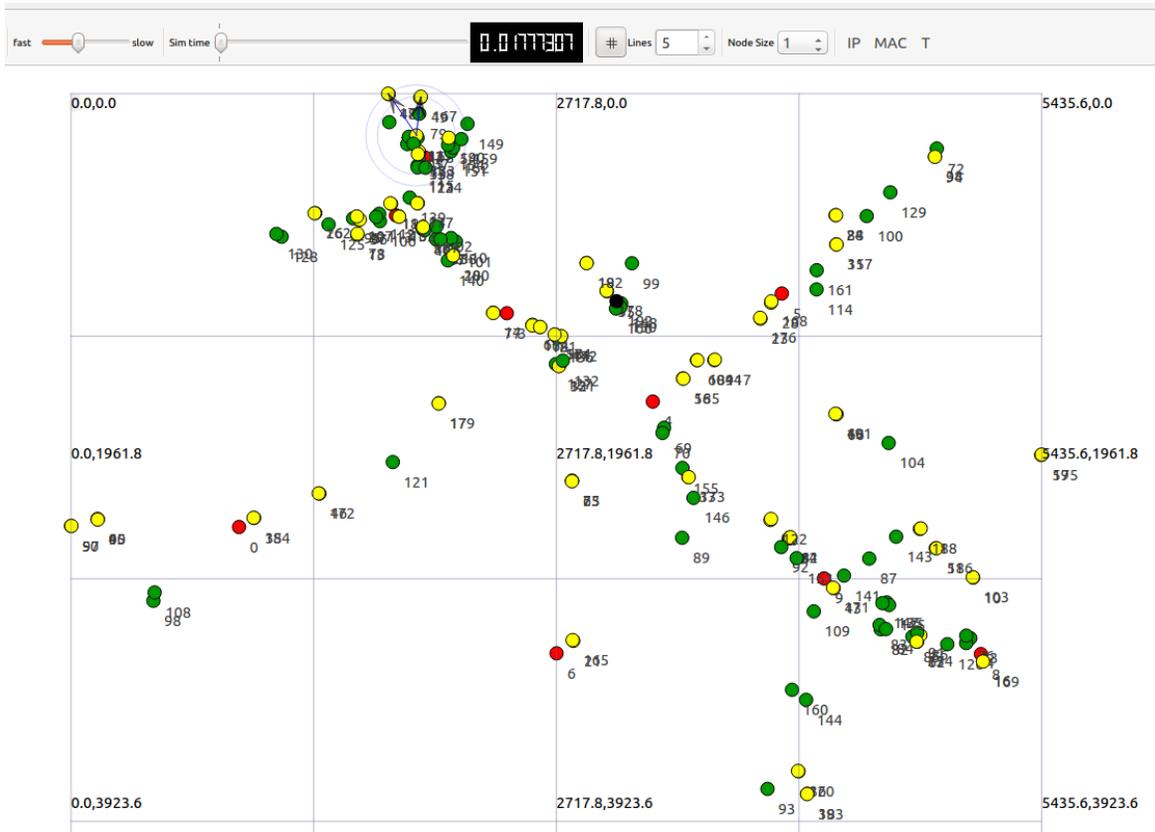


Figure 2.11. Screenshot of LoWPAN-WiFi network animation on top of RTC-A

Verification Results

In the simulations, we assumed that all the PV sent data simultaneously, smart meters acted as relay nodes, and concentrators received the data from the smart meters and sent the processed data to the edge router. Further, because we used the TCP/IP connection solution for the test bed initial verification stage, there was no packet loss at the application layer theoretically. Therefore, it was difficult to trace the packet loss by using our current customized application module. Because the throughput degradation is a consequence of the packet loss in the link layer, however, we used throughput as the first-priority QoS evaluation for the packet availability by using our modules. We next verified the developed communications simulation test bed via (1) parameter testing of each communications technologies and (2) analyzing the performance of the proposed communications architecture.

Basic Configuration and Parameter Verification of Communications Models

In this subsection, we aim to identify the configurable parameters of the NS-3 models of five alternative communications technologies and evaluate the impact of these parameters on the performance of the proposed hybrid communications systems. We first consider two alternative communications technologies of LoWPAN, implemented by the *LR-WPAN* and *6LoWPAN* models, and PLC implemented by PLC models for the PV-SM link. The parameter verification results are demonstrated in Table 3, which lists all adjustable parameters for each technology in the column of ‘**Parameter.**’ Note that alternative configurations for each adjustable parameter are listed in the column named as ‘**Values,**’ and the first alternative setting is the optimal value

based on our testing. We identified three adjustable parameters for the LoWPAN link: (1) physical model with optimal values of single or multiple model spectrum channel, (2) propagation loss model with several available models, and (3) propagation delay model with several alternative models shown in the WiFi mesh case of Table 4. We further validate their impact on the system performance.

For PLC links, we have three interesting observations from Table 2.3. (1) Both the spectrum model and the payload modulation coding scheme play a critical role in the hybrid system performance. There are two spectrum models available for the PLC communication. Compared to the *G3 spectrum model*, the time-invariant spectrum model can give better system performance. The configurable parameter setting for the spectrum model consists of a low bound frequency, high bound frequency, and a number of channels in which the channel number can impact the system performance—as it increases to 300, the lowest message latency and highest throughput can be achieved. Also, different payload modulation coding schemes show quite different system performance in terms of latency, throughput, and packet loss rate. (2) The modulation coding scheme for the header has relative medium impact because the header message has less data length, compared to the big-size payload message. (3) The system performance shows a very low sensitivity to the settings of transmit power spectral density, background noise, and cable type.

Next, we are interested in the impact of parameters of three alternative communications technologies in the SM-DC links on system performance, respectively. Table 2.4 shows the verification results, the same as with Table 2.3. In the Ethernet cable case, among four configurable parameters, only the delay has a slight impact on the system performance. It implies that the Ethernet cable link always demonstrates stable performance regardless of its parameter setting. Although there are seven adjustable parameters for both the *WiFi* and *Mesh* models in NS-3, only the propagation loss model and mesh protocol stack show visible importance. For the topology of RTC-A, the log distance and random propagation models are not suitable because the data concentrator cannot receive the PV message from the PV inverters with high possibility over these two propagation models. The testing result shows that the Dot11-s mesh protocol outperforms the flame stack in the random topology of RTC-A, which is consistent with the findings of [31]. Compared with the WiFi technology, the WiMAX model is subject to four adaptable parameters, among which the physical layer modulation is more important than the other three. This result is comparable with PLC technology. Due to the larger transmission range, the propagation mode in WiMAX has less impact on system performance than it does in WiFi.

In summary, the basic configuration and parameter verification results enable the initial optimal parameter setting of each hybrid simulation model ready for the subsequent alternative technology comparison. It also sets the stage for the development of the optimal design parameter tuning toolbox in the third stage of our proposed design framework.

Table 2.9. Basic Configuration and Parameter Verification of PV-SM Links

Technology	Parameter	Values	Impact
LoWPAN	Phy Model	'SingleModelSpectrumChannel', 'MultiModelSpectrumChannel'	To test
	Loss Model	'LogDistancePropagationLossModel'	To test
	Delay Model	'ConstantSpeedPropagationDelayModel'	To test
PLC	Spectrum Model	'TimeInvariantSpectrumModel' - (0, 10e6, 300), (0, 10e6, 100), (0, 10e7, 100), (0,10e6, 200) 'G3SpectrumModel' - (0, 1e5, 300), (60Hz, 2240us), (0, 5e4, 5)	High
	Payload Modulation Coding Scheme	'QAM64_RATELESS', 'QAM4_RATELESS', 'QAM32_RATELESS', 'QAM64_12_21', 'BPSK_1_2', 'BPSK_RATELESS'	High
	Header Modulation Coding Scheme	'BPSK_1_2', 'BPSK_1_4'	Medium
	Transmit Power Spectral Density	'1e-8', '1e-6'	No
	Background Noise	'1e-9', '1e-10'	No
	Cable Type	'NAYY50SE', 'NAYY150SE', 'AL3x95XLPE', 'MV_Overhead', 'NYCY70SM35'	No

Table 2.10. Basic Configuration and Parameter Verification of SM-DC Links

Technology	Parameter	Values	Impact
Ethernet cable	Data Rate	'100 Mbps', '30 Mbps'	No
	Delay	'3.33us', '6560 nanosec'	Low
	Encapsulation Mode	'Dix', 'Llc', 'IpArp', 'EthernetV1'	No
	Maximum Transmit Unit	'1500 bytes', '1492 bytes'	No
WiFi mesh	Mesh Protocol Stack	'Dot11sStack', 'FlameStack'	Medium
	Mac Type	'RandomStart' - '0.1 s', '0.5 s'	Low
	Propagation Delay	'RandomPropagationDelayModel', 'ConstantSpeedPropagationDelayModel'	Low
	Propagation Loss	'FixedRssPropogationLossModel', 'FrissPropagationLossModel', 'LogDistancePropogationLossModel', 'RandomPropogationLossModel'	High
	WiFi Standard	'80211a', '80211b', '80211g'	Low
	Spread Interface Channel	'SPREAD_CHANNEL', 'NumberOfInterface' - '3', '2', '1'	Low
	Remote Station Manager	'AarfWifiManager', 'ArfWifiManager', 'AparfWifiManager', 'AarfedWifiManager', 'AmrrWifiManager', 'IdealWifiManager', 'CaraWifiManager', 'MinstreWifiManager', 'ConstantRateWifiManager', 'RraaWifiManager'	Low
WiMAX	Phy Layer Modulation Type	'QAM16-12', 'QAM16-34', 'QAM64-32', 'QAM64-34', 'BPSK-12', 'QPSK-12', 'BPSK-34'	High
	Service Flow Type	'UGS', 'RTPS', 'NRTPS', 'BE'	Medium
	Propagation Mode	'Friis_Propagation', 'Cost231_Propagation', 'Random_Propagation', 'Log_Propagation'	Medium
	Scheduler	'SIMPLE', 'MBQOS', 'RTPS'	Low

Impact of Diverse Traffic Loads

We tested the performance of our architecture in addressing different traffic loads by considering that the PV generates data with different rates. In Case I, we assume that all the PV generates the data with the generating interval of 30 ms. The average latency and the maximum latency of the five areas are shown in Figure 2.12. Additionally, the average packet availability and the minimum packet availability of the five areas are shown in Figure 2.13.

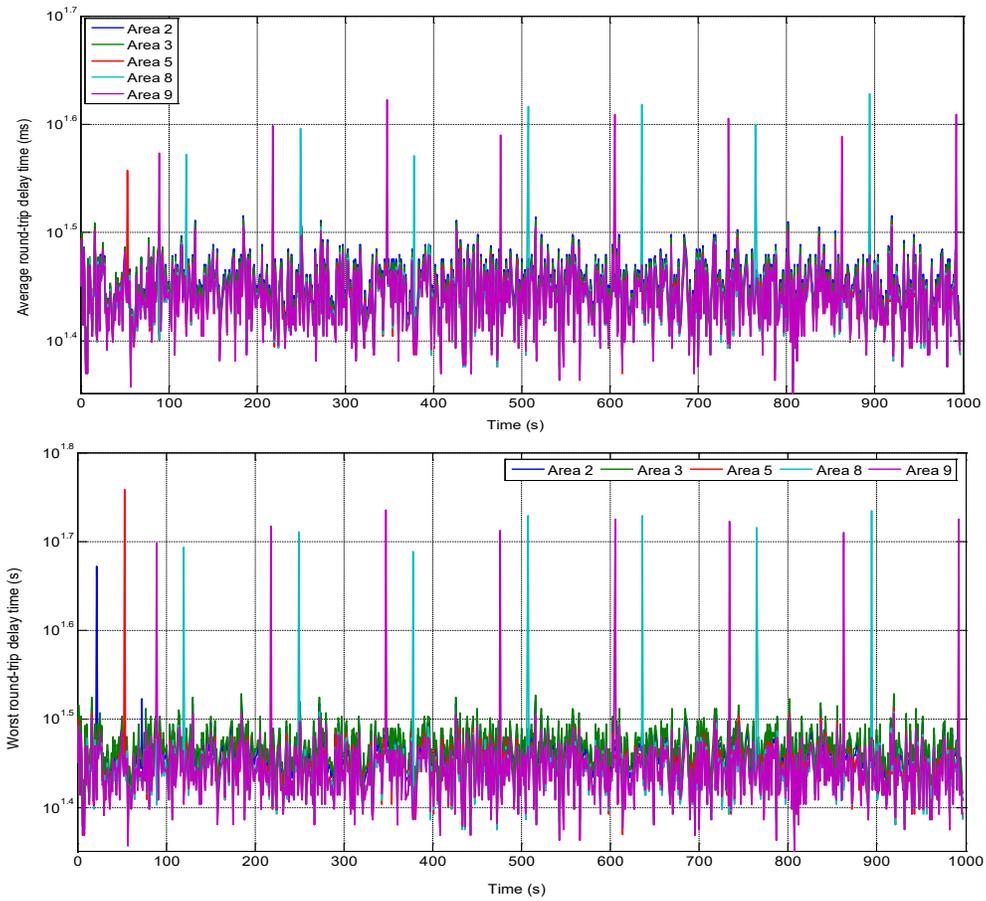


Figure 2.12. RTC-A regional communications performance by assuming that the PV has a generating interval of 30 ms: (a) average latency, (b) maximum latency

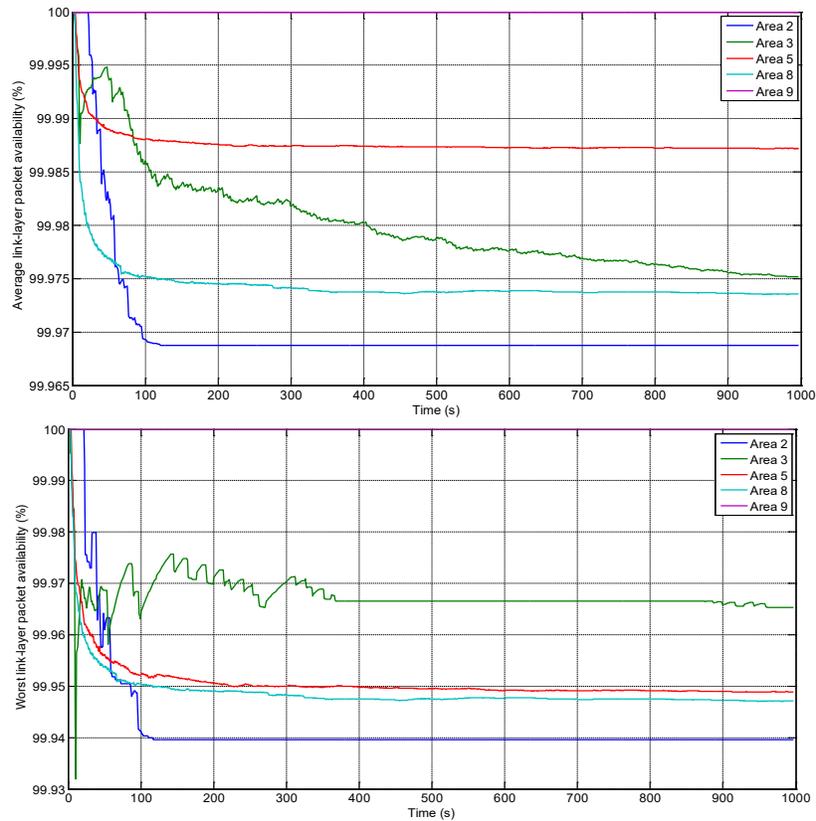


Figure 2.13. RTC-A regional communications performance by assuming that the PV has a generating interval of 30 ms: (a) average packet availability, (b) minimum packet availability

In Case II, we assume that all the PV generates data with 10 different generating intervals. The performance evaluation results of the average latency and the maximum latency are shown in Figure 2.14. The performance evaluation results of the average packet availability and the minimum packet availability are shown in Figure 2.15.

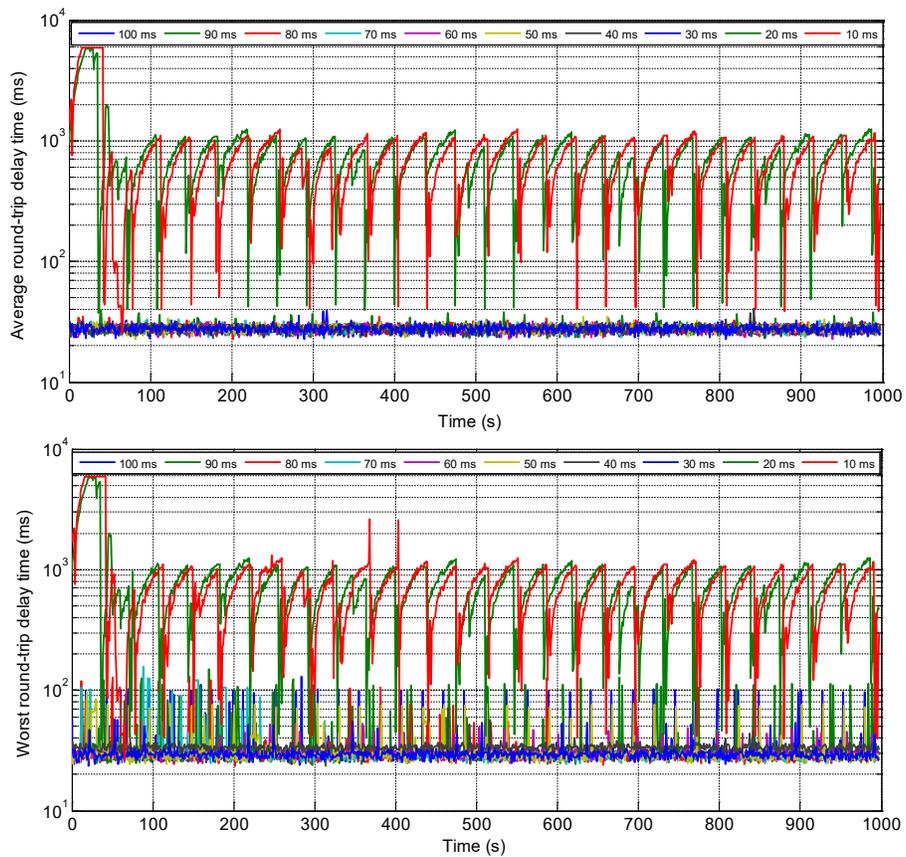


Figure 2.14. RTC-A regional communications performance by considering different generating intervals for PV: (a) average latency, (b) maximum latency

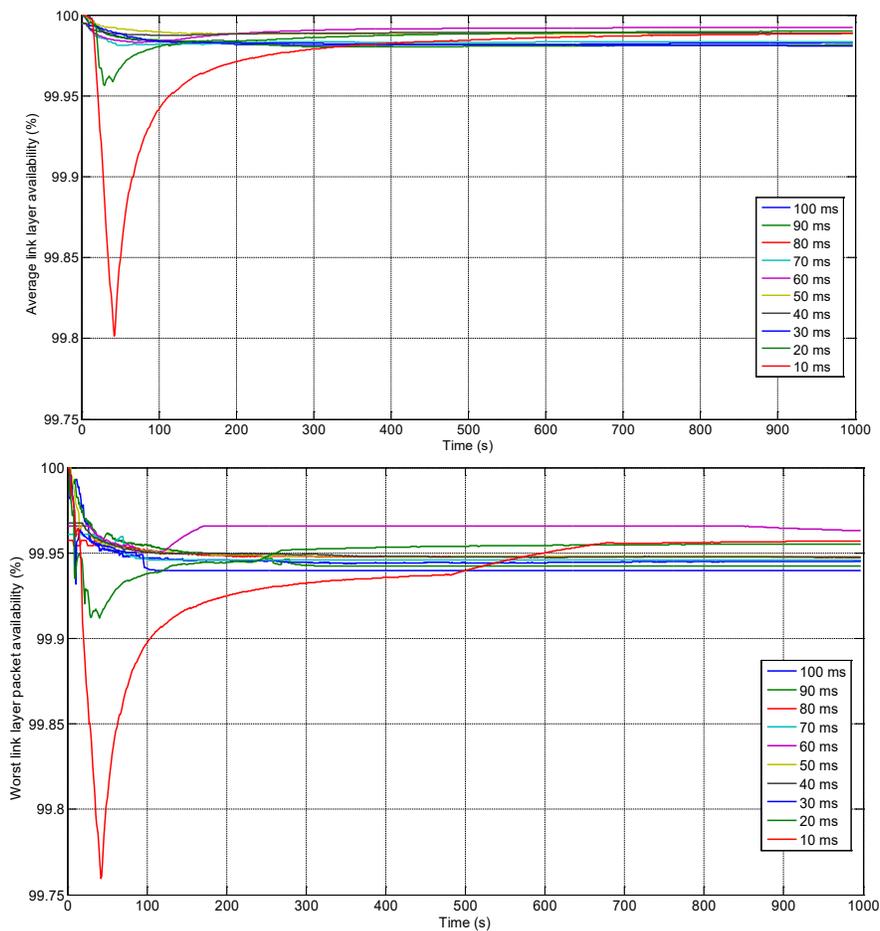


Figure 2.15. RTC-A regional communications performance by considering different generating intervals for PV: (a) average packet availability, (b) minimum packet availability

II. Opportunistic Hybrid Communications Simulation Models

Real-Time Resilient Middleware Framework

The objective of these middleware-based network management schemes is to implement the opportunistic functionality of the proposed communications systems. Such an opportunistic function implemented at the control layer enables the message to be routed through each system based on recent data about latency and availability of multiple communications technologies at the infrastructure layer and the QoE information from the power system applications to ensure reliable message passing. It currently consists of a (1) proactive networking management mechanism and (2) intelligent networking-resource allocation scheme, described as follows.

Proactive Networking Management Mechanism

We developed a middleware-based proactive networking management mechanism in which the middleware infrastructure can work together with the optimization control algorithm in the application layer to realize the proactive networking management scheme. As illustrated in Figure 2.16, our middleware infrastructure consists of the networked master middleware instances and slave middleware instances. Each master middleware instance receives the

commands from the optimization control algorithm and carries on the corresponding networking reconfiguration by interacting with the slave middleware instances. Further, our middleware mechanism can provide the optimization control algorithm with the feedback on current QoS information.

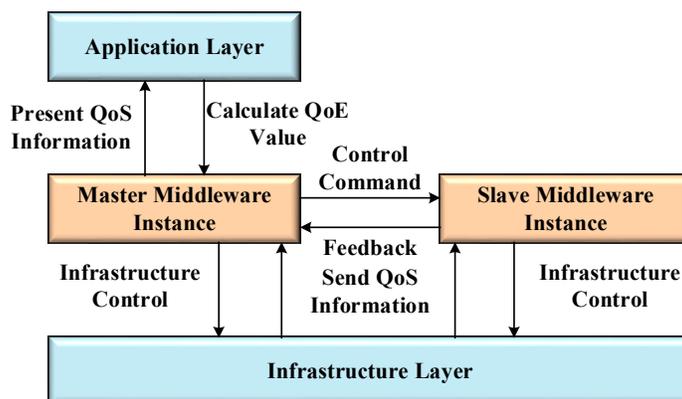


Figure 2.16. Middleware-based proactive networking management mechanism

The main difference between the master middleware instances and the slave middleware instances is that the master ones installed in the data concentrator and servers have Application Program Interfaces (APIs). Further, the master middleware instances can send the control commands to other middleware instances installed in the individual gateway nodes via their local middleware instances in real time. Meanwhile, power operators can directly operate on the middleware instances via these APIs.

The vertical structure of our proposed communications network is illustrated in Figure 2.17. Middleware acts as a network controller, which organizes, coordinates, and supervises diverse application services in power systems. As shown in Figure 2.17, master middleware instances on data concentrators have four service modules: QoS monitor, QoE evaluation, false data detector, and dispatcher. And the slave middleware instances on relay nodes consist of two modules: QoS monitor and dispatcher. In our middleware architecture, all the messages need to go through the dispatcher module for message classification. The QoE evaluation module is responsible for transforming power operators' qualitative experiences to quantitative QoE-based QoS criteria, which can be used by the QoS monitor module. Moreover, the QoS monitor module observes the QoS performance of each data flow at the link layer, refers the QoE feedback from the QoE evaluation module, and determines the routing strategy in the network layer. The false data detection module conducts the function of verifying the realness of the data content. Our proposed middleware also analyzes the various objective QoS criteria defined by IEC 61850 for individual applications and integrates the information with the subjective QoE criteria provided by the operators.

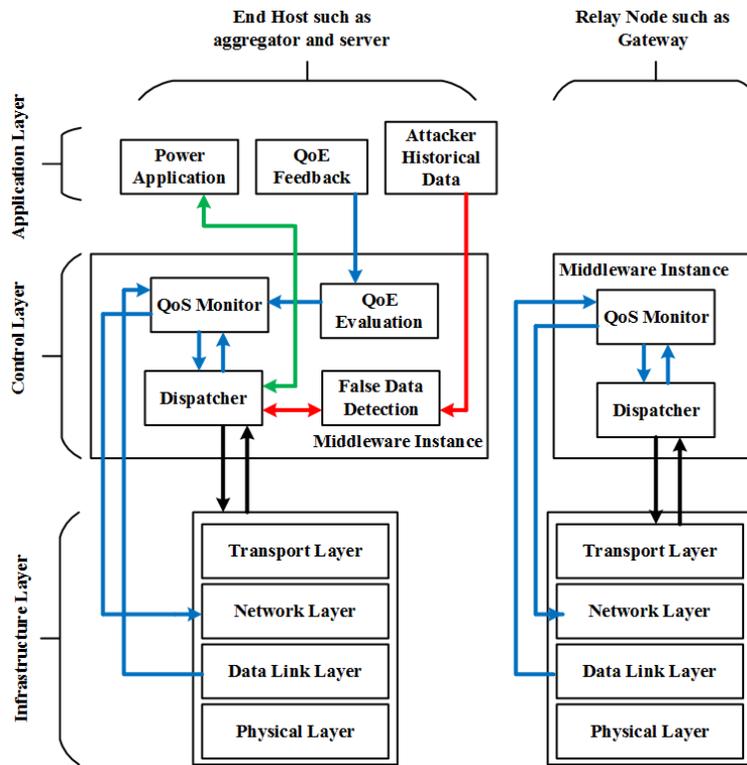


Figure 2.17. Vertical structure of proposed opportunistic communications systems

Intelligent Networking-Resource Allocation Scheme

In this work, we also designed an intelligent networking-resource allocation scheme for the middleware mechanism that adaptively and automatically distributes the data flow to the available links based on their real-time status. By doing so, we not only can enhance the QoS of each traffic but also efficiently use the network resource. As the first step, we developed an equal-cost multipath routing (ECMP) protocol to realize the link aggregation. In the ECMP protocol, the cost of selecting any available path is equal and the networking resource allocation is random.

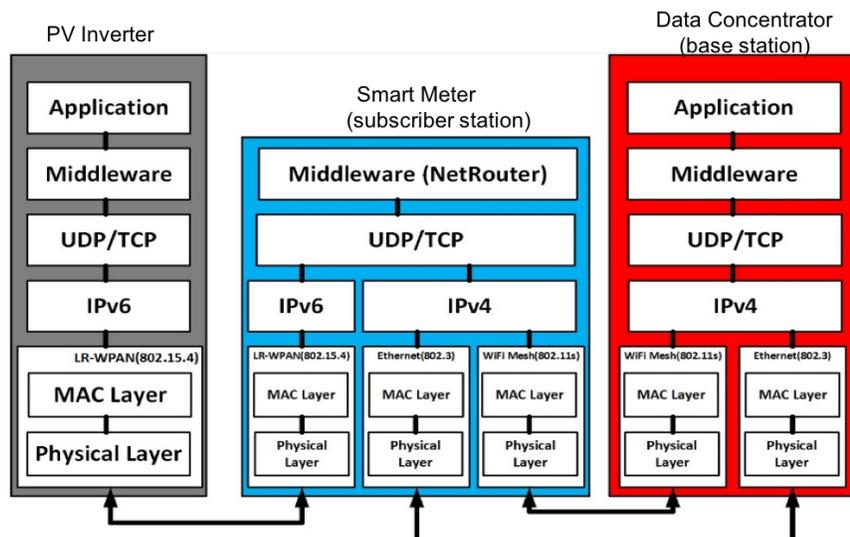


Figure 2.18. NS-3 simulation models of opportunistic hybrid communications architectures

Based on the developed prototypical hybrid communication system simulation models of Figure 2.10, we further integrated these proposed opportunistic middleware mechanisms into NS-3 simulation models via (1) implementation of the middleware layer, also known as the control layer, between the transport layer and application layer; (2) configuration of multiple communications technologies in both the physical and MAC layers at each communications node. Figure 2.18 demonstrates the complete NS-3 simulation models for the opportunistic hybrid communications systems. From Figure 2.18, both the proactive network management mechanism and intelligent networking resource allocation scheme are conceptually implemented at the middleware layer. Note that the middleware source code is actually running at the application layer. To further mimic the multipath situation in the envisioned smart grid communications systems, we implemented (1) both PLC and LoWPAN links between PV inverters and smart meters, (2) all three WiFi Mesh, WiMAX, and Ethernet cable links between smart meters and data concentrators, respectively. The lower two layers of both the smart meter and data concentrator nodes in Figure 2.18 show the example of implementing both WiFi mesh and Ethernet cable links. Based on this implementation, we verify the functionalities of two proceeding proposed multipath networking management schemes in the next step.

Implementation of Middleware Architecture at NS-3 Hybrid Simulation Models

The proposed middleware architecture instances are implemented in NS-3 by customizing certain application models and calling relative objects to configure and modify the network settings. As shown in Figure 2.19, the master middleware instances—that is, application driven—are built in NS-3 to achieve three main functions: (1) implementing security control on the infrastructure layer, (2) providing a platform to enable the application layer to call the objects of the infrastructure layer to directly configure the network either locally or remotely, and (3) converting the format of the application data to a general format used for the IoT cloud. In Figure 2.19, the blue arrows refer to the data flow, including sensing data and QoE evaluation. The red arrows refer to the security control information including the

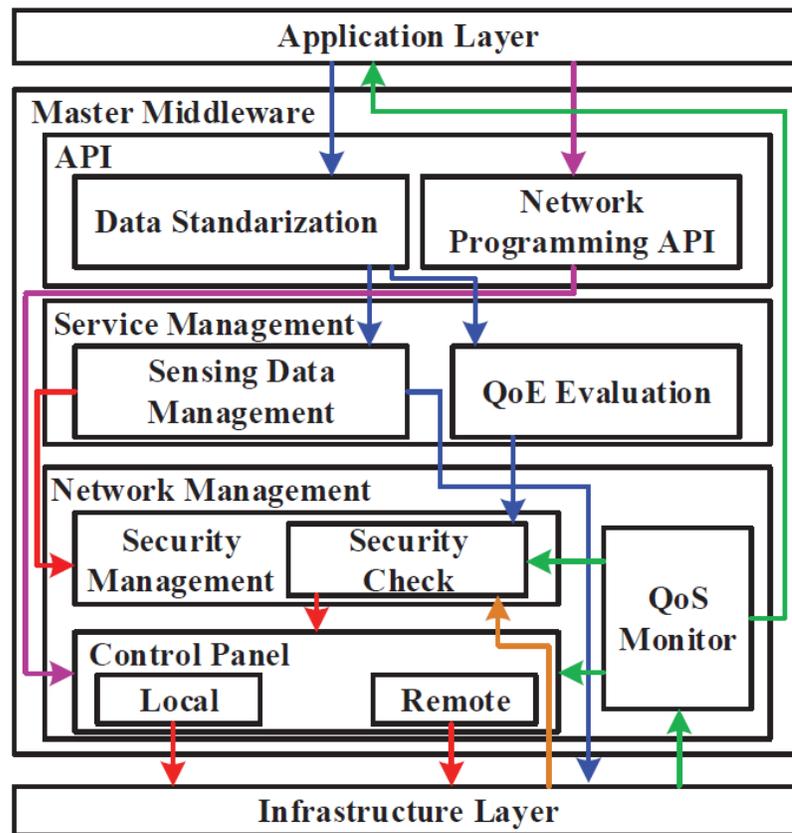


Figure 2.19. Implementation structure of master middleware in NS-3

local control signal for the infrastructure layer and the remote-control command sent to the related slave middleware instances. The green arrows represent the QoS information obtained by monitoring the infrastructure layer. The pink arrows represent the control command for the infrastructure layer that is sent directly from the application layer. And the brown arrow is the feedback signal sent from the slave middleware instances through the infrastructure layer. As illustrated in Figure 2.19, the master middleware instance consists of three main modules: API, service management, and network management. In the API, there are two submodules: data standardization and network programming API, respectively. The first submodule is designed to change the application’s specific data format to a standardized format and vice versa. The second one is designed to allow the end users in the application layer to configure the infrastructure layer by providing a device-specific library function no matter what kind of network devices and protocols they use. Service management consists of two submodules: sending data management and QoE evaluation. The QoE evaluation submodule transmits the QoE information to the network management module to improve the network condition. In the network management module, the security management module is implemented to identify the potential attacks and the control panel is designed to configure the local network and send the control command to the slave middleware instances to control the network remotely. The NS-3-realization of slave middleware is shown in Figure 2.20. The blue arrow refers to the control command sent from the master middleware instance through the infrastructure layer. The red arrows represent the control

signal. The green arrows are the QoS information obtained from infrastructure layer. And the brown arrow is the feedback information describing the updated QoS information after executing the infrastructure layer control.

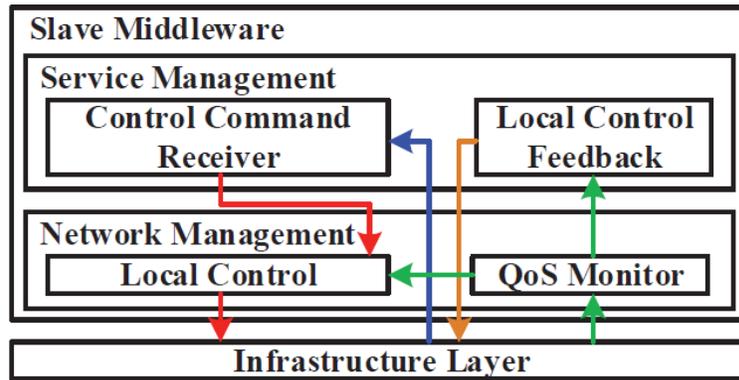


Figure 2.20. Implementation structure of slave middleware in NS-3

Further, our middleware treats QoE as an objective measure of the network condition and quantifies QoE by using quantitative scales from 0 to 2 representing poor, fair, and good, respectively. In our work, we consider that the operators have two types of personality: critical and tolerant. We model the QoE evaluation process based on the operator’s experience after the data transfer between the aggregator and the remote server starts. For example, letting ρ be the throughput and B denote the assigned bandwidth, we illustrate the QoE evaluation process modeling in Table 2.5.

Table 2.5. Modeling of the QoE Evaluation Process

	0	1	2
Critical	$\rho < 0.6B$	$0.6B \leq \rho < 0.9B$	$\rho \geq 0.9B$
Tolerant	$\rho < 0.15B$	$0.15B \leq \rho < 0.7B$	$\rho \geq 0.7B$

In addition to the above method, we proposed another QoE-based evaluation process modeling method, in which QoE is defined as the weighted sum of the QoS information. Based on the QoS information, such as the average throughput T_i and end-to-end latency D_i , QoE metrics can be defined as follows:

$$QoE = w_1 \sum_{i=1}^n T_i + w_2 \sum_{i=1}^n D_i \quad (2.12)$$

where w_1 and w_2 are the normalized parameters selected carefully based on the preference of the network specification. In other words, the critical weights in the QoE metrics are determined according to the predefined preference among the QoS information. For example, if we consider the availability of the data as our primary concern, the value of weight w_2 associated with the end-to-end delay D_i is higher than that of the weight w_1 associated with the throughput T_i .

Experimental Verification Results

In this subsection, we aim to test the proposed multipath networking management schemes for the congestion attacks. The middleware instance is designed to leverage the QoS information of latency to detect the congestion attacks. We evaluated the performance of our middleware mechanism by using Test Case A simulated via a NS-3-based communications test bed. In the test bed, the PV inverters communicate with smart meters via ZigBee, the smart meters communicate with the concentrators via RF Mesh or Ethernet, and the concentrators communicate with the edge router via optical fiber. We analyzed the performance of our proposed communications architecture by using this communications test bed. In the simulations, we assumed that all the PV sent data simultaneously, smart meters acted as relay nodes, and the concentrators received the data from the smart meters and sent the processed data to the edge router. We considered two scenarios in which the middleware instance is designed to leverage the QoS information of latency to detect the congestion attacks. The threshold for the attack detection was set as $\tau=100ms$. Considering the environment uncertainty, we assumed that the alarm was triggered if the situation that the latency $T \geq \tau$ lasted for more than 2 s. In the simulations, as soon as receiving the alarm, the middleware instance autonomously switched to another available link for data delivery by adjusting the routing table. First, we considered that the congestion attack occurred on the PLC link between the PV inverters and the smart meters at time $t=50$ s. The simulation results are presented in Figure 2.21. Second, we considered that the congestion attack occurred on the WiFi link between the smart meters and the data concentrators at time $t=50$ s. The simulation results are shown in Figure 2.22.

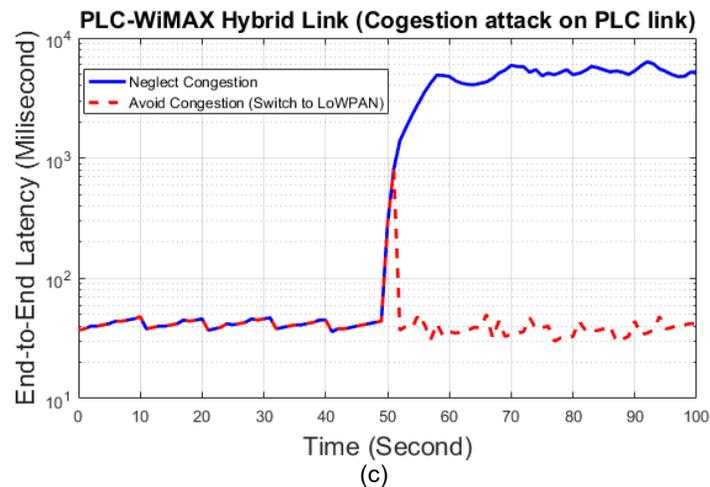
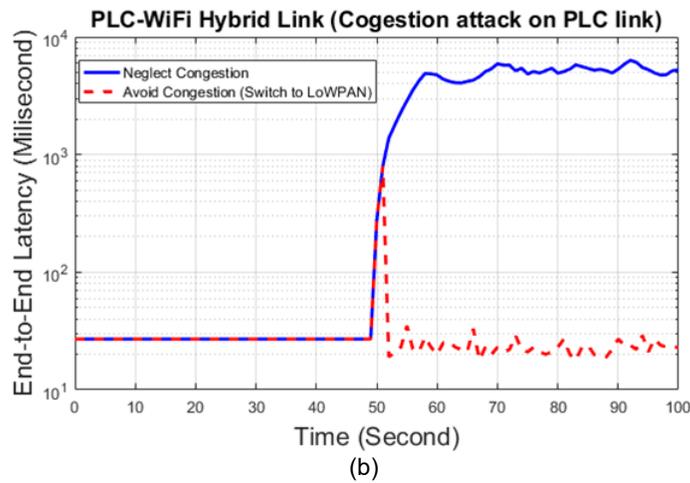
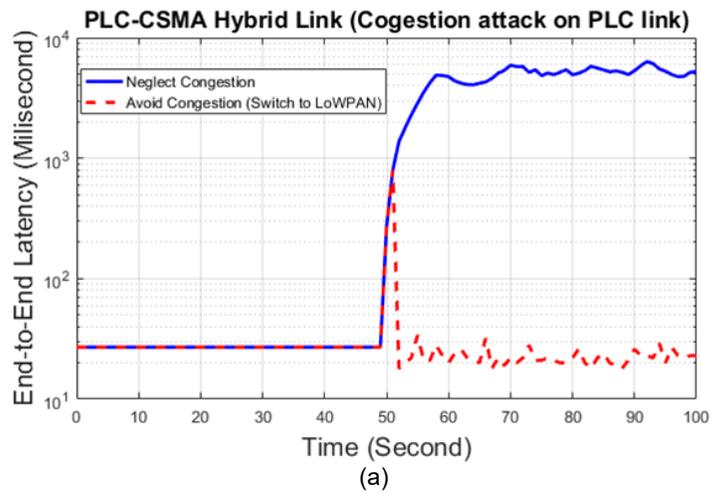


Figure 2.21. RTC-A regional hybrid communications performance by considering: (a) PLC-CSMA hybrid link, (b) PLC-WiFi hybrid link, (c) PLC-WiMAX hybrid link

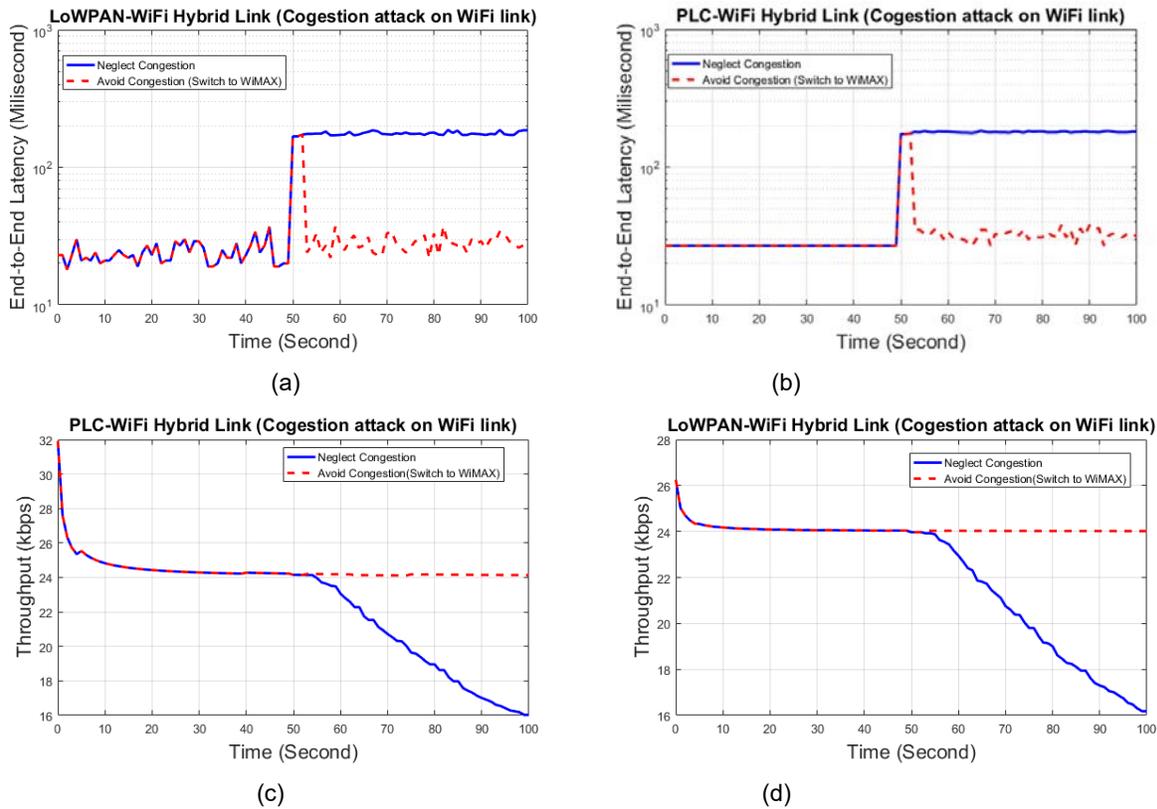


Figure 2.22. RTC-A regional hybrid communications performance of end-to-end by considering (a) PLC-WiFi hybrid link and (b) LoWPAN-WiFi hybrid link; and of throughput by considering (c) PLC-WiFi hybrid link and (d) LoWPAN-WiFi hybrid link

Deep Convolutional Neural Networks (CNN)-Deconvolutional Neural Networks (DCNN) Situational Awareness Scheme

We exploited the deep learning technology to develop a deep convolutional neural networks (CNN)-deconvolutional neural networks (DCNN) scheme for situational awareness, as shown in Figure 2.23, which illustrates the structure of our proposed deep learning-based cyberattack detection scheme. As shown in Figure 2.23, our scheme mainly consists of deep CNNs and deep DCNNs. Additionally, a long short-term memory (LSTM) based recurrent neural network (RNN) network is designed to process the abstract features achieved by deep CNN and to extract temporal features from the abstract CNN features. The main building block of deep CNN is the convolution layer. The convolution layer performs convolution on the inputs with a set of learnable kernels. The obtained values are further evaluated by using an activation function that produces a set of output feature maps, as shown in Figure 2.21. Convolution output \mathbf{y}_j^l corresponding to the j th kernel in the l th layer is calculated as follows:

$$\mathbf{y}_j^l = f(\sum_i \mathbf{x}_i^{l-1} * \mathbf{k}_{ij}^l + \mathbf{b}_j^l) \quad (2.13)$$

where \mathbf{x}_i^{l-1} is the i th input vector, \mathbf{k}_{ij}^l is the corresponding learnable kernel mapping the i th input to the j th output for the l th layer, \mathbf{b}_j^l is a bias vector corresponding to the j th output in the l th layer, and $f(\cdot)$ is an activation function.

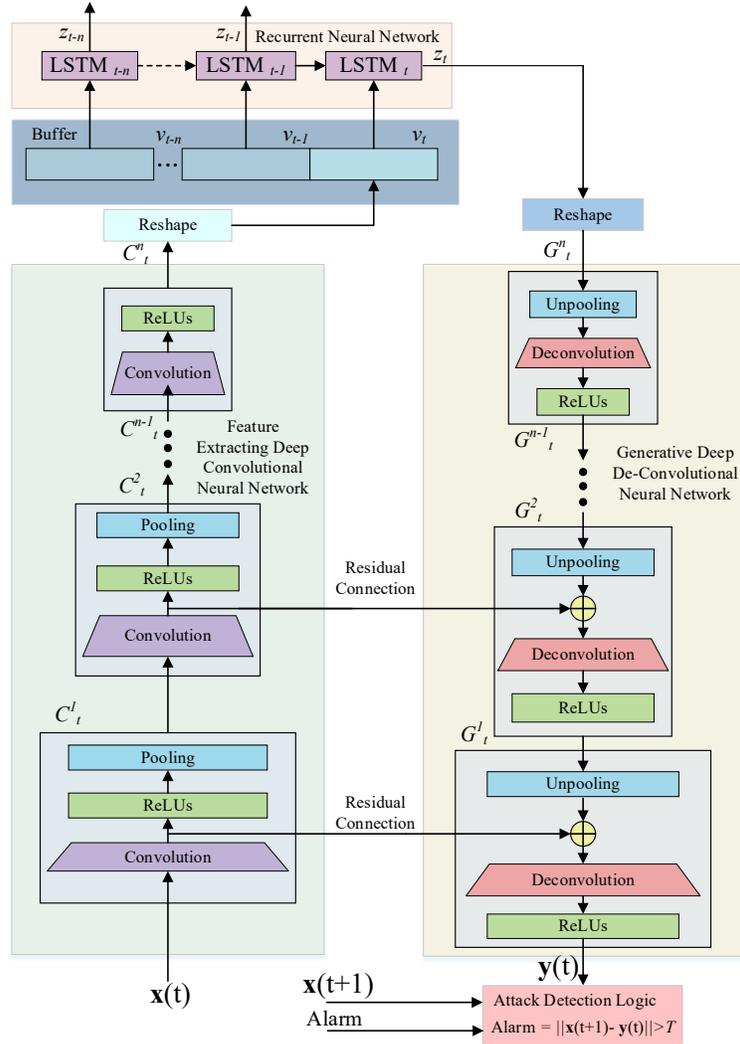


Figure 2.23. Proposed deep CNN-DCNN-based cyberattack detection scheme

During the training process, each learnable value of the kernel's weight and bias are adjusted to extract some specific features available in the training set. In the proposed scheme we adopt rectifier activation defined as $f(\cdot) = \max(0, \cdot)$. Convolution layers are followed by a sub-sampling layer to reduce computations and to gradually build up further spatial invariance. Our implementation uses 2×2 maximum pooling as sub-sampling. The second main component of the machine learning method is the deep DCNN whose essential components are deconvolutional layers. Similar to the convolution operation, the deconvolution layer outputs are calculated as follows:

$$\mathbf{x}_j^l = f(\sum_j \mathbf{y}_j^{l+1} * \mathbf{k}_{ji}^l + c_i^l) \quad (2.14)$$

where \mathbf{y}_j^{l+1} is the j th unpooled deconvolution kernel output from the above layer, \mathbf{k}_{ji}^l is the corresponding learnable deconvolution kernel mapping the j th input to the i th output for the l th

layer, \mathbf{c}_i^l is a bias vector corresponding to the i th output in the l th layer, and $f(\cdot)$ is the rectifier activation function.

Additionally, a 2×2 unpooling operation is implemented on the input of the deconvolution layer before performing the deconvolution. The number of kernels decreases for lower deconvolution layers, and thus the lowest deconvolution layer produces a single output vector with the same dimension as the original data vector. As shown in Figure 2.23, the output of the top CNN is reshaped and buffered to set as the inputs to an LSTM-based RNN network. The operation of the LSTM block is governed by the following equations:

$$\left\{ \begin{array}{l} \mathbf{i}_t = \sigma(W_{ix}\mathbf{x}_t + W_{im}\mathbf{m}_{t-1} + W_{ic}\mathbf{c}_{t-1} + \mathbf{b}_i), \\ \mathbf{f}_t = \sigma(W_{fx}\mathbf{x}_t + W_{fm}\mathbf{m}_{t-1} + W_{fc}\mathbf{c}_{t-1} + \mathbf{b}_f), \\ \mathbf{c}_t = f_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot g(W_{cx}\mathbf{x}_t + W_{cm}\mathbf{m}_{t-1} + \mathbf{b}_v), \\ \mathbf{o}_t = \sigma(W_{ox}\mathbf{x}_t + W_{om}\mathbf{m}_{t-1} + W_{oc}\mathbf{c}_t + \mathbf{b}_o), \\ \mathbf{m}_t = \mathbf{o}_t \odot h(\mathbf{c}_t), \\ \mathbf{y}_t = \varphi(W_{ym}\mathbf{m}_t + \mathbf{b}_y). \end{array} \right. \quad (2.15)$$

where the W terms denote weight matrices (e.g., W_{ix} is the matrix of weights from the input gate to the input); the \mathbf{b} terms denote bias vectors; $\sigma(\cdot)$ is the logistic sigmoid function; \mathbf{m} is the cell output activation vector; \mathbf{i} , \mathbf{f} , \mathbf{o} , and \mathbf{c} are the input gate, forget gate, output gate and cell activation vectors, respectively, all of which have the same size as \mathbf{m} ; \odot is the element-wise product of the vectors; $g(\cdot)$ and $h(\cdot)$ are the cell output and input activations, respectively, which use hyperbolic tangent activation functions; and $\varphi(\cdot)$ is the softmax output activation function.

Further, the output of the final LSTM block is reshaped and provided as input for the deep DCNN, which characterizes the temporal features learned by the CNN and provides the DCNN with additional temporal features. Moreover, the residual connections in Figure 2.23 are used to improve the convergence of the overall architecture. Our deep CNN-DCNN architecture is trained with gradient descent with the loss function defined in the following to minimize the mean square error (MSE) between \mathbf{x}_t and \mathbf{y}_t .

$$L(\mathbf{x}_t, \mathbf{y}_t) = \frac{1}{M} \sum_{\alpha=1}^M (\mathbf{x}_t - \mathbf{y}_t)^2 \quad (2.16)$$

where M is the length of \mathbf{x}_t and \mathbf{y}_t . Additionally, by setting $\mathbf{y}_t = \mathbf{x}_{t+1}$, we are able to predict the value of the streaming measurement \mathbf{x} at time $t + 1$. In our work, \mathbf{x}_t is a time window of the packet delay values measured in real time and thus \mathbf{x}_{t+1} is the next time window of them. Therefore, once our proposed network can infer the next time of the packet delay values, we compare the predicted next window with the exact next window. Figure 2.23 shows that an alarm is triggered if the error of the predicted window is greater than a predefined threshold T .

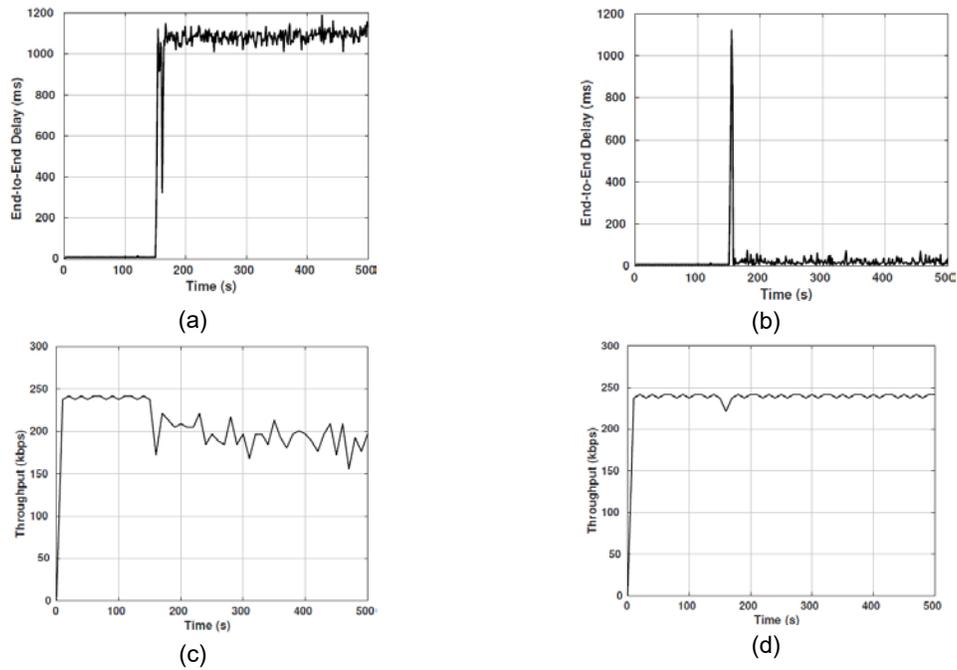


Figure 2.24. End-to-end latency: (a) without middleware, (b) with middleware; throughput: (c) without middleware, (d) with middleware

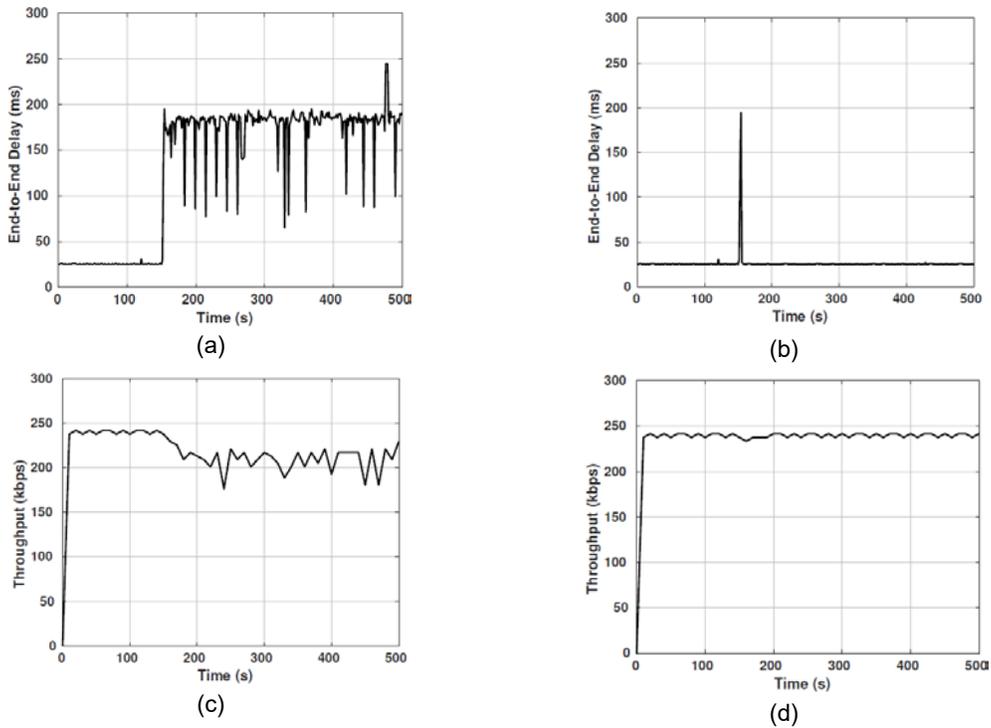


Figure 2.25. End-to-end latency: (a) without middleware, (b) with middleware; throughput: (c) without middleware, (d) with middleware

Experimental evaluations: We evaluated the performance of our middleware with our deep CNN-DCNN situational awareness scheme and the second QoE-based evaluation process modeling method by using the Case-A hybrid communications test bed in NS-3. We consider different communications media in two different scenarios. In the first scenario, the communications medium between the PV inverters and smart meters is LoWPAN and the communications medium between the smart meters and the data concentrators is Ethernet. One denial-of-service (DoS) attack occurs when a smart meter is attacked at time $t = 150$ s. The simulation results are shown in Figure 2.24. The simulation results shown in Figure 2.25 demonstrate the performance of our middleware for the test system with different communications media of LoWPAN and WLAN.

Enhancement of Middleware Mechanism

The integration of middleware instance to hybrid communications models is completed. the monitoring and control functionality of the middleware instance is enhanced by incorporating a hierarchical scheme, as shown in Figure 2.26.

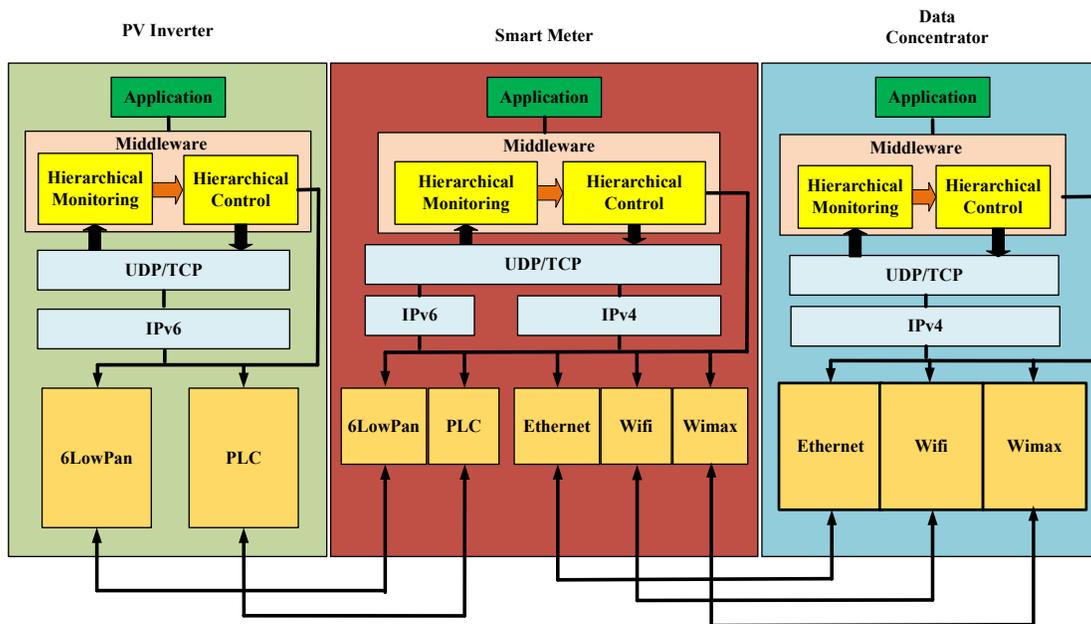


Figure 2.26 Illustration of improved middleware functionality along with the hybrid model

The improvement of the middleware instance is detailed in Figure 2.27, which is achieved via NS-3 network simulator. In the hierarchical monitoring mechanism, we focus on the online acquisition of the Quality of Service (QoS) information—including throughput, jitter, end-to-end latency, and packet loss—from two hierarchical levels. The system-level QoS information is probed by using the *FlowMonitor* module in the NS-3, and the lower-level QoS information with adjustable granularity is obtained by leveraging the *FlowProbe* class in the NS-3. Further, as shown in Figure 2.27, based on the online multi-granularity QoS information achieved by using the hierarchical monitoring mechanism, the hierarchical control mechanism is designed to realize two functionalities: (a) the communications media selection among 6LoWPAN-CSMA, 6LoWPAN-WiMAX, 6LoWPAN-RF Mesh, PLC-CSMA, PLC-WiMAX, and PLC-RF Mesh on

the physical and data-link layers; and (b) the online decision-making on routing paths for critical data delivery on the network layer.

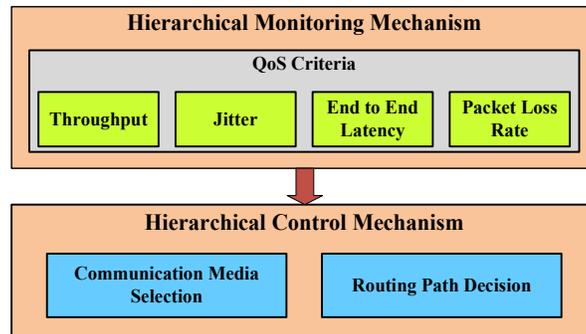
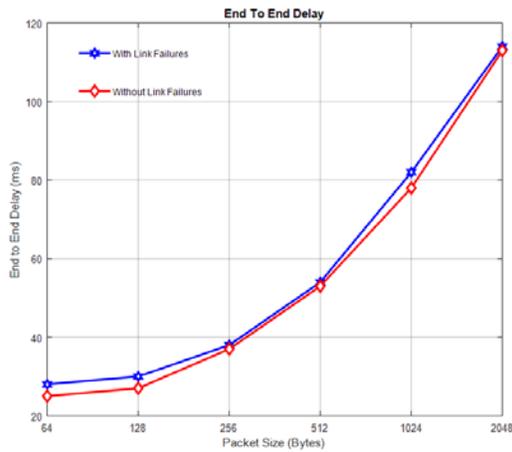
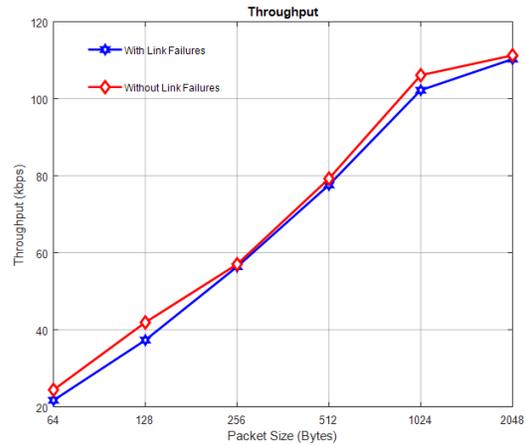


Figure 2.27. Illustration of the improvement in middleware mechanism

Experiment evaluations: Some simulation results are presented in figures 2.28 and 2.29. In the first scenario, we evaluated the effectiveness of the redundancy in communications media in our hybrid networking infrastructure in mitigating the impact of link failure. The priorities of the communications media between smart meters and data center from the highest to the lowest are: CSMA, WiFi, and WiMAX. In our mechanism, when the link with the communications medium having higher priority fails, the link with lower priority is assigned. In the simulation, at $t = 0$ s, the data packets are transferred through CSMA links. At $t = 20$ s, one CSMA link on the routing path fails. After the failure is detected, the failed CSMA link is replaced by the corresponding WiFi link. At $t = 40$ s, this WiFi link fails. After the failure is detected, this failed link is replaced by the corresponding WiMAX link. The simulation result on end-to-end latency versus different packet sizes is shown in Figure 2.28(a). In this figure, the red curve represents the networking performance when there are no link failures, and the blue curve shows the networking performance when there are link failures in our hybrid communications infrastructure. As shown in Figure 2.28(a), it is clear that our hybrid communications infrastructure achieves a good trade-off between the media redundancy and communications resiliency. The performance of our networking infrastructure based on throughput is shown in Figure 2.28(b).

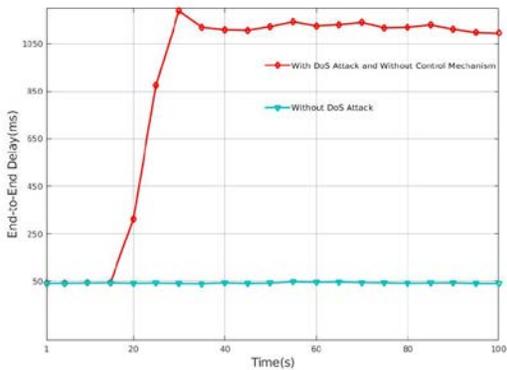


(a)

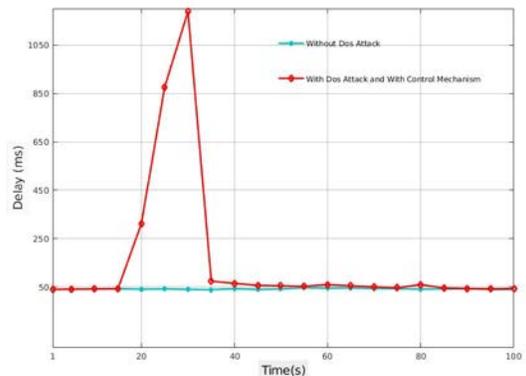


(b)

Figure 2.28 (a) End-to-end latency versus packet sizes with and without failure recovery and (b) throughput versus packet sizes with and without failure recovery



(a)



(b)

Figure 2.29. End-to-end latency (a) when there is no control mechanism on network layer and (b) when there is control mechanism on network layer

In the second scenario, we evaluated the effectiveness of the control mechanism in our middleware architecture in mitigating the Denial of Service (DoS) attacks. In this scenario, a DoS attack occurs at $t = 20$ s. The performance of our test bed without using the control mechanism is shown in Figure 2.29(a), from which it can be seen that the DoS attack causes the end-to-end latency increases from 40 ms to 1200 ms. The performance of our test bed with our control mechanism is shown in Figure 2.29(b). From Figure 2.29(b), we can get that the end-to-end latency reduces to around 42 ms by using our control mechanism.

Reinforcement Learning (RL)-Based Opportunistic Control Scheme

In addition to extending the functionalities of our middleware, we worked on enhancing its intelligence by exploiting reinforcement learning (RL) techniques. Our improved middleware mechanism with the RL-based real-time opportunistic control scheme is illustrated in Figure

2.30. As shown in Figure 2.30, our RL-based control mechanism is designed to achieve two main functionalities: (1) optimal and adaptive routing in network layer and (2) cross-layer optimization between network and physical layers. We focus on completing the first functionality of the control mechanism in our middleware architecture.

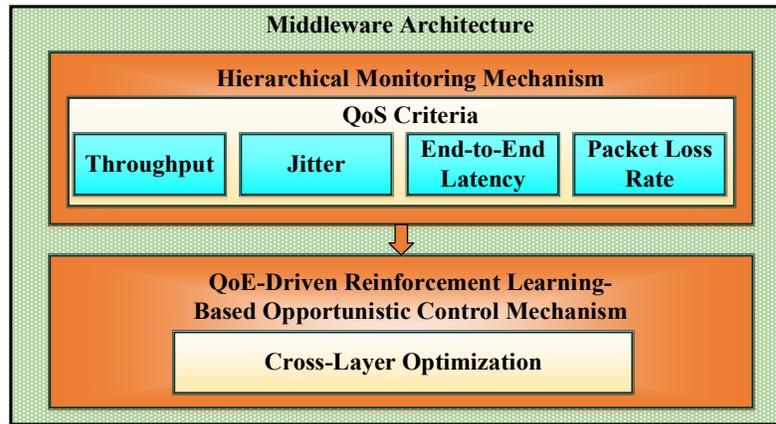


Figure 2.30. Structure for improved middleware architecture with RL-based real-time opportunistic control scheme

We have been trying to deploy the RL-based real-time opportunistic control scheme in our NS-3-based test system; however, there still remains challenges because of the limitation of NS-3. Therefore, instead of applying the scheme in our established communications system that is demonstrated via the Case-A test system, we realize our objective by developing a multi-controller software-defined networking (SDN) infrastructure that is demonstrated in Figure 2.31. As shown in Figure 2.31, in our infrastructure (1) the PV constitutes an end-device layer, (2) the smart meters have the functionality of being virtual switches (vSwitches) and constitute the switching layer, and (3) data concentrators constitute the controller-plane layer. Our RL-based real-time control scheme is equipped in the middleware mechanism in the data concentrators. Because of the high controllability and interoperability, our SDN-based infrastructure can be considered an alternative solution for our established system to realize opportunistic communications. Additionally, the scalability concerns of the conventional SDN infrastructure are addressed by our multi-controller mechanism.

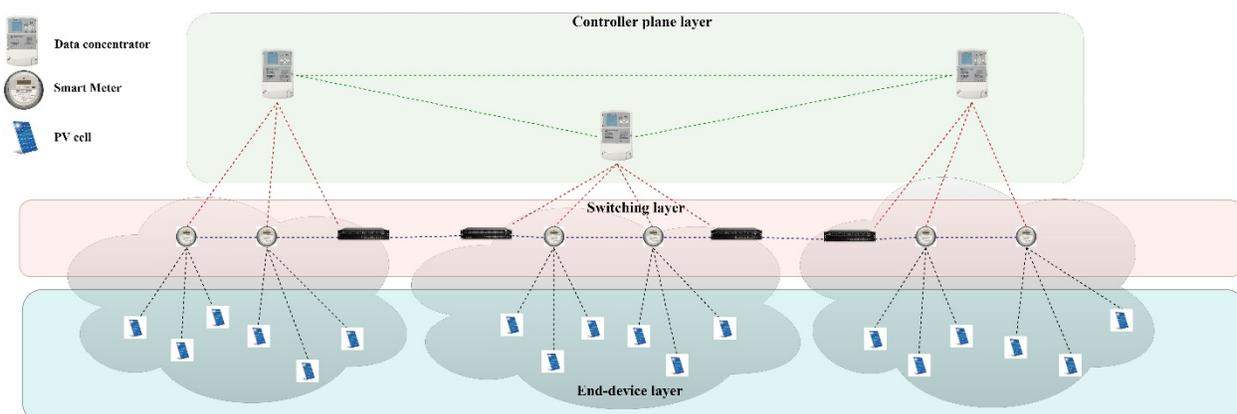


Figure 2.31. High-level illustration of alternative multi-controller SDN-based communications system that was developed to realize RL-based real-time decision-making mechanism

To develop a data-driven opportunistic control mechanism for our middleware architecture, we first exploited actor-critic RL, which is illustrated in Figure 2.32. As shown in Figure 2.33, our control mechanism takes the state s_t of the current routing path as the input. In our work, s_t depicts the QoS criteria of the network, such as the expected end-to-end latency and the average packet loss rate. Upon receiving the state s_t , the control mechanism makes decision, a_t , to select the following smart meter, to which the packets are routed, based on a policy $\pi_\theta(s_t, a_t) \in [0,1]$ that demonstrates the probability that the decision/action a_t is taken in state s_t . The policy is presented by a neural network called the actor network, which is trained via a policy gradient method that is formulated as follows:

$$\theta \leftarrow \theta + \alpha \sum_t \nabla_\theta \log \pi_\theta(s_t, a_t) A(s_t, a_t) + \beta \nabla_\theta H(\pi_\theta(\cdot | s_t)) \quad (2.17)$$

where $A(s_t, a_t)$ is an advantage function representing how much better a specific decision a_t is compared to the average decision taken according to the policy π_θ when the state is s_t , and $H(\cdot)$ is the entropy of the policy. For a given experience, the advantage $A(s_t, a_t)$ is estimated as $r_t + \gamma V^{\pi_\theta}(s_{t+1}; \theta_v) - V^{\pi_\theta}(s_t; \theta_v)$, where $V^{\pi_\theta}(\cdot; \theta_v)$ is the estimate of the value function $v^{\pi_\theta}(s)$ via a neural network, called the critic network, and θ_v is the parameters of the critic network. The critic network is trained according to the standard temporal difference method that is formulated as follows: $\theta_v \leftarrow \theta_v - \alpha' \sum_t \nabla_{\theta_v} (r_t + \gamma V^{\pi_\theta}(s_{t+1}; \theta_v) - V^{\pi_\theta}(s_t; \theta_v))^2$, where r_t denotes the reward reflecting the performance evaluated by our QoE metric. At the current stage, we model r_t the reward as follows: $r_t = \alpha \mathbb{E}(w_2 \tilde{D} + w_3 \tilde{R})$, where $\mathbb{E}(\cdot)$ is the expectation function; α is negative when the packet does not reach the destination node and positive when the packet reaches the destination; \tilde{D} and \tilde{R} are the remaining time to reach the destination and the packet loss, respectively; and w_2 and w_3 are the importance weights. More QoS factors will be included in future modeling.

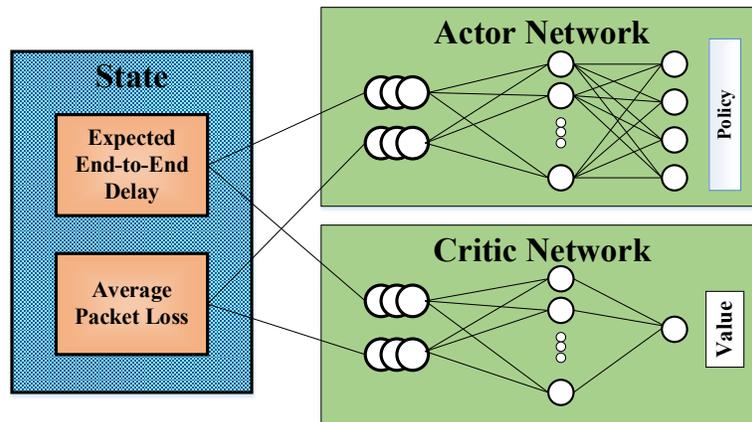


Figure 2.32. Illustration of deep actor-critic RL-based real-time opportunistic control scheme for middleware mechanism

Experimental evaluation: In this early stage, we evaluate the performance of our middleware mechanism with the RL-based real-time control scheme by using the networking topology (i.e., topology of smart meters) shown in Figure 2.33.

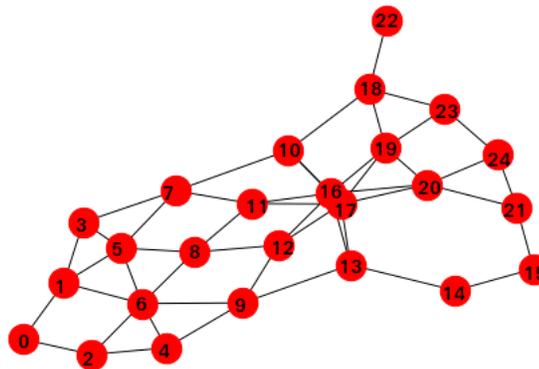


Figure 2.33. Network topology for experimental evaluation

We consider two scenarios for the performance evaluation. In both scenarios, the data of PV connected with Smart Meter 0 are sent to the data concentrator connected with Smart Meter 24. In the first scenario, the routing selection converges to [0, 1, 6, 8, 12, 16, 20, 24]. The reward value, which characterizes the integrated QoS information, is shown in Figure 2.34(a). In the second scenario, we consider that there are congestion attacks launched on the Links 16–19, 16–20, and 17–19 beginning from $t = 7000$. After the attacks occur, the routing path is adapted from [0, 1, 6, 8, 12, 16, 20, 24] to [0, 1, 3, 7, 10, 18, 23, 24]. The reward value is presented in Figure 2.34(b). From Figure 2.34(a), we can observe that the final selection decision was achieved within the time [0, 3000]. From Figure 2.34(b), it is clear that our actor-critic method is resilient to the congestion attacks.

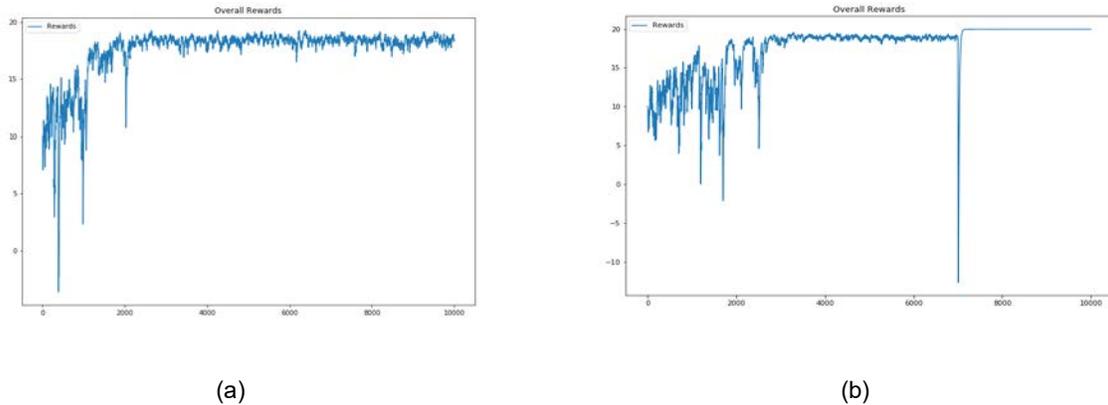


Figure 2.34. Integrated QoS information presented via reward value achieved: (a) without congestion attacks and (b) with congestion attacks

We further enhance the resilience and the scalability of our RL-based control mechanism by the exploiting Asynchronous Actor Critic (A3C) RL technique, whose mechanism is illustrated in Figure 2.35.

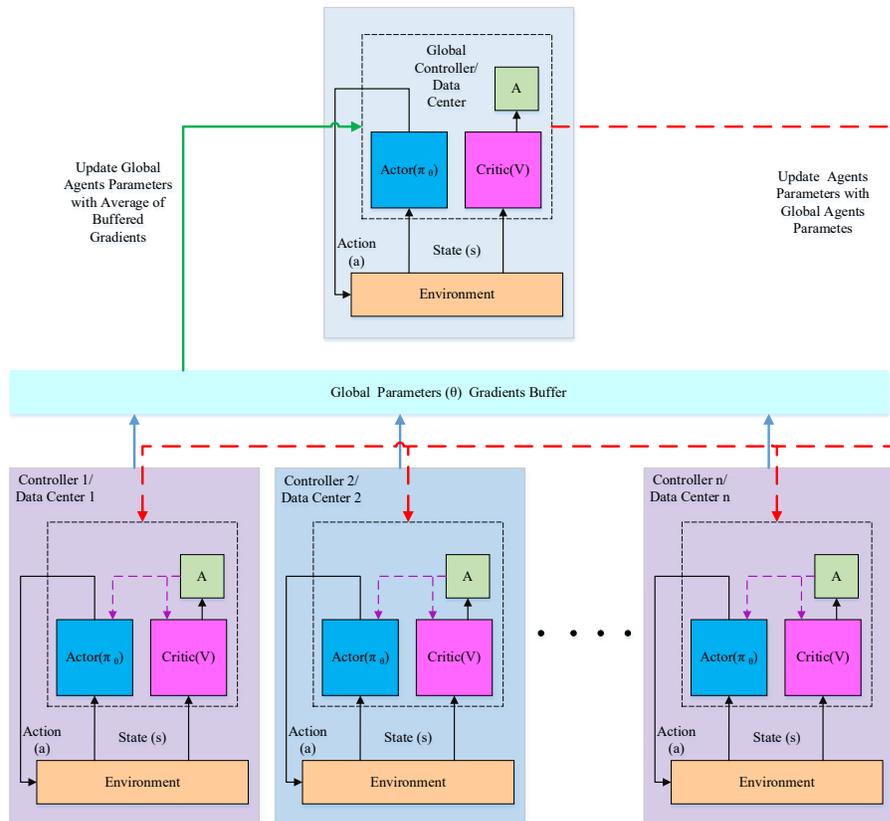


Figure 2.35. Illustration of A3C RL-based hierarchical real-time opportunistic control mechanism

As illustrated in Figure 2.35, our A3C RL-based real-time opportunistic control mechanism has a two-tier hierarchical structure. In the lower tier, the individual data centers in our SDN-based networking infrastructure, as illustrated in Figure 2.31, interact with the environment independently and conduct the actor-critic RL scheme locally in their own controlling domain. One data center, which has more computing power, acts as a global controller in the upper tier and cooperate with the local data centers to achieve the global optimal solution for the overall network. We evaluate our A3C RL-based real-time opportunistic control mechanism via two stages. In the first stage, we assume that the individual controllers have the view of the whole networking environment via the communications between the controllers. In other words, in the first stage, we assume that the communications delays between the data centers, acting as controllers in our proposed networking infrastructure, are comparable with the communications delays between the data centers and their associated smart meters acting as vSwitches in our networking infrastructure. Under this assumption, the performance is evaluated using the topology presented in Figure 2.33. We consider two scenarios for the performance evaluation. In both scenarios, the data of PV connected with Smart Meter 0 are sent to the data concentrator connected with Smart Meter 24. In the first scenario, we consider that there are congestion attacks launched on Links 16–20 and 17–19 beginning from $t = 8000$. In the second scenario, we consider that there are congestion attacks launched on the Links 16–19, 16–20, and 17–19 beginning from $t = 8000$. The reward values, which characterize the integrated QoS information, in these two scenarios are shown in Figure 2.36(a) and (b), respectively. From Figure 2.36, it is clear that in our A3C RL-based real-time opportunistic control mechanism the data centers cooperate with each other to obtain the attack resilience.

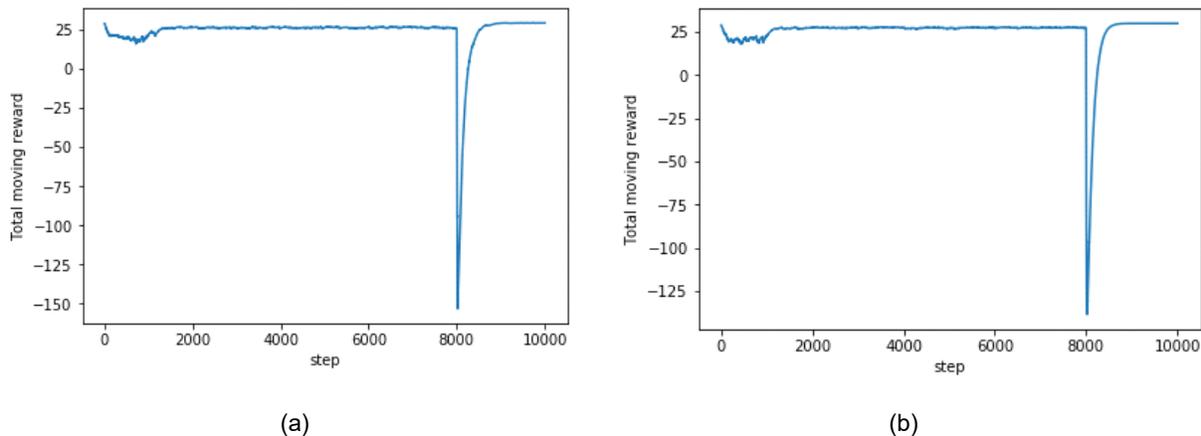


Figure 2.36. Integrated QoS information presented via reward value achieved: (a) with congestion-attack Scenario 1 and (b) with congestion-attack Scenario 2

In the second stage, we consider a more practical situation in which the communications delays between the data centers are longer compared with the communications delay between the data centers and their associated smart meters. In other words, in this situation, the data centers are more sensitive to the networking environment in their own controlling domain. Under this assumption, we assume the 25 smart meters in the network in Figure 2.33 belong to four controlling domains of four associated data centers: $c_0 = [0,1,2,3,4,5,6,7,8,9]$, $c_1 =$

[10,11,12,13,14,15], $c2 = [16,17,18,19,20,21]$, and $c3 = [22,23,24]$. In the simulation, we consider that the congestion attacks are launched on the Links 16–19, 16–20, and 17–19 beginning from $t = 5000$ and the reward value achieved for 100000 time steps in the simulation are presented in Figure 2.37, which validates the attack-resilience of our control scheme.

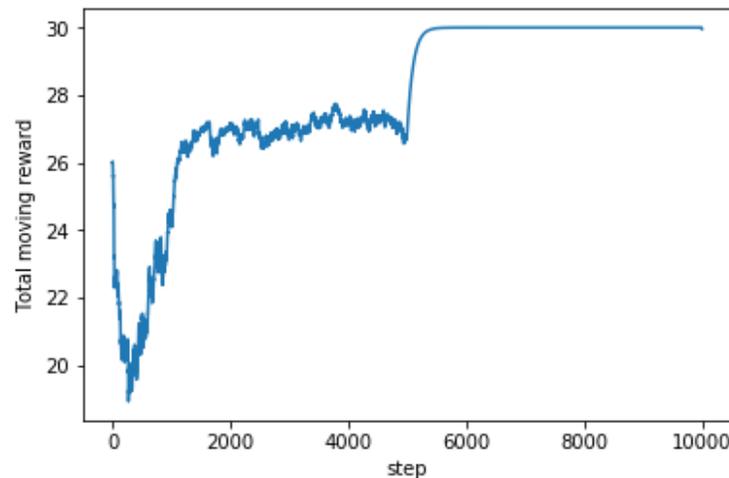


Figure 2.37. Integrated QoS information presented via reward value

III. Simulation-Based Parameter Optimization Framework

The design of reliable, dynamic, fault-tolerant hybrid smart grid communications networks is a challenge to achieve for autonomous power grids. Hybrid networks use different communications technologies for different area networks. A simulation-based parameter optimization framework is proposed to tune parameters of hybrid communications technologies to achieve the optimal network performance. It consists of three main components: a parallel executor used to speed up a list of simulations; a sampler running simulations using the parallel executor at each generation; and a hybrid stochastic optimization algorithm for tuning configurable parameters of hybrid designs and applications. The proposed hybrid metaheuristic optimization algorithm combines an evolutionary algorithm with a gradient method to quickly achieve an approximately global optimum solution. Three optimization test functions are employed to train the adjustable parameters of the hybrid algorithm. Results show the proposed parameter optimization framework can help the designer choose the right hybrid architecture with an optimal parameter set for a large-scale broadband PLC-WiMAX hybrid smart grid communications network.

Motivation

The design of hybrid communications networks is not straightforward because the different technologies used in sub-networks have a large number of configurable parameters, which increases the amount of experimental (or simulation) tests necessary for their evaluation. The nondeterministic nature of the environment is another factor that makes network design difficult. The hybrid smart grid communications network must be fault tolerant and adaptive because of

the dynamic network topology caused by dynamic power grid topology and the changing objectives of smart grid applications. The design of a large-scale hybrid smart grid communications network requires a simulation-based optimization method to tune the configuration parameters of communications technologies and parameters of smart grid applications. A simulation-based parameter optimization framework is proposed to help the designer choose the right hybrid architecture with an optimal parameter set. This scalable and extendable framework may accept different communications technologies with different line facilities—i.e., trunks, loops, or links—on top of variable topologies and identify the optimal configurable parameters for each related communications model and application parameters for that hybrid design.

The novel contribution of this work is a simulation-based parameter optimization framework with features of parallel computing and using a hybrid evolutionary search algorithm. The proposed design provides a simulation-based optimization tool then can help designers identify the optimal parameter set for a selected hybrid communications configuration. The algorithm is used to develop a new tool that performs network parameter optimization and was designed to be used with NS-3, but it could be extended to work with other network simulators. Using the simulator, the optimization algorithm tunes all the input parameters, at both the application and architectural level, to provide an optimum set within the required QoS metrics. The large parameter space and the simulation-based genetic algorithm impose a heavy computational load. It is beneficial to parallelize execution of these computationally intensive simulations and thus speed up the performance of the simulation-based optimization algorithm. Through combining a gradient-based algorithm and a genetic algorithm, the hybrid evolutionary gradient algorithm is proposed as a new parameter identification algorithm. The primary application of this framework is thus the optimization of network configuration parameters and application parameters through extensive hybrid communications system simulations. The proposed solution provides a way to design and optimize hybrid smart grid communications systems in a highly nondeterministic environment for a large number of cooperating intelligent power grid devices.

Simulation-Based Parameter Optimization Framework

The framework for this tool was designed to use three main components. It consists of a parallel executor wrapped inside a sampler, which itself is wrapped inside the hybrid optimization algorithm. The core module of the design is the parallel executor, which also processes the raw results. By separating the three components, the framework provides the user with the option to run the sampler alone to help inform the input parameter ranges for the optimizer and hence slightly reduce the computational and time requirements. The list of simulations to run is provided by the sampler, which simulates all scenarios required by the optimizer algorithm for a single generation. It also handles data storage, additional post-processing, and initial comparison of the results. The optimizer algorithm determines which simulation scenarios are required to be tested and provides them as a list of parameters to the sampler. The high-level block diagram of this framework is provided in Figure 2.38. The parameters used in this high-level diagram are detailed in the following subsection.

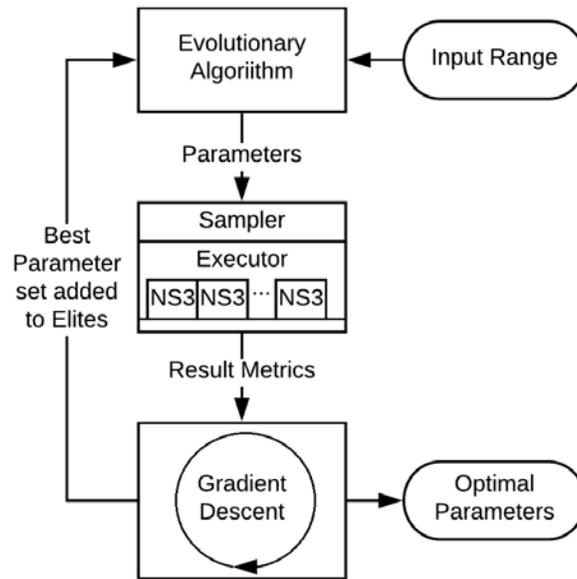


Figure 2.38. High-Level framework block diagram

Parallel Executor

The purpose of the parallel executor is to run a large number of simulations simultaneously by using the ubiquitous multicore physical or virtual processors available to most systems, and it was implemented using available Python modules. It may also be extended using a message-passing interface module to take advantage of highly scalable cluster computing resources. The parallelism and scalability of this parallel executor allows for greatly improved computational time by efficiently using all available computational resources.

Sampler

Three critical system design requirements are considered to validate the performance of each hybrid communications design, and they are a single trip latency of 300 ms or less, throughput of 9.6 kbps or more, and packet loss rate of 1% or less. To further quantify the performance value of different parameter settings and to better compare similar cases, a weighted cost function is proposed as follows:

$$\begin{aligned}
 cost &= \sum_{i=1}^3 w_i * x_i + CC_i \\
 \text{where } CC_i &= \begin{cases} 10000 & x_i \geq lim_{i+} \\ 0 & lim_{i-} < x_i < lim_{i+} \\ 10000 & x_i \leq lim_{i-} \end{cases}
 \end{aligned} \tag{2.18}$$

where x_1 , x_2 , and x_3 stand for latency (ms), throughput (bps), and packet loss rate (%), respectively. w_i , $i = 1,2,3$ is the weight factor of the i^{th} metric. This weight factor is the product of the corresponding metric's importance factor and unit normalization factor. Hence, w_i , $i = 1,2,3$ is the corresponding weight normalizing the incremental cost. In this paper, w_1 is set as 1000/ms and w_2 is set as 0.001/bps. w_3 is set as 20 to normalize the cost increment for every

0.05% change in packet loss rate. CC_i , $i = 1,2,3$ refers to the conditional costs of the i^{th} metric, which allows setting predefined conditional limits, such as boundary conditions, of this metric $[\text{lim}_i^-, \text{lim}_i^+]$. Thus, using these conditional costs, the out-of-bounds regions of these metrics result in a very high cost, which effectively restricts the optimization algorithm to search for results within the specified boundary of these metrics.

Hybrid Evolutionary Gradient Algorithm

To identify the optimal parameter set for the specific hybrid communications system design, there are two commonly used algorithms: namely, gradient descent and evolutionary algorithms. Similar to hill climb algorithms, the gradient-based algorithms perturb an initial guess along all available degrees of freedom to improve the objective function value, and the best perturbed position becomes the new position at each iteration, until no perturbation can improve the objective. Meanwhile, the evolutionary algorithm is a selective random search algorithm designed to achieve a global optimum within a large parameter space. The general idea of many variants of these algorithms is to identify dominant solutions and to breed these solutions until the global optimum is found. There is a finite chance for mutation of each parameter every time a new solution is bred. As shown in Algorithm 2.1, only the best solutions are retained in the breeding population as the elite population and passed along between generations. Both populations are limited in size so to reduce the overall computational requirements.

Algorithm 1 Hybrid Evolutionary Gradient Algorithm

```
1:  $(Parameters, Values) \leftarrow GetParamVals(script)$ 
2: Evolutionary algorithm starts
3: for  $gen = 0$  to  $maxGen$  do
4:   Run Sampler with ExecutorPool
5:   for  $Result$  from  $ExecutorPool$  do
6:     parse  $Result$  into  $ParsedResult$ 
7:     obtain  $Hash$  of  $ParsedResult$ 
8:     store  $ParsedResult$  with  $Hash$  as the key
9:   end for
10:   $BestHash \leftarrow hash(\min(cost(Elites)))$ ,
11:   $Position$  is obtained from values of  $BestHash$ 
12:  Gradient algorithm starts
13:  while  $Position$  has changed do
14:    Perturb  $Position$  to  $(Parameters, Values)$ 
15:    Run Sampler with ExecutorPool
16:    for  $Result$  from  $ExecutorPool$  do
17:      parse  $Result$  into  $ParsedResult$ 
18:      obtain  $Hash$  of  $ParsedResult$ 
19:      store  $ParsedResult$  with  $Hash$  as the key
20:    end for
21:    if  $\min(cost(Elites)) < cost(BestHash)$  then
22:       $BestHash \leftarrow hash(\min(cost(Elites)))$ 
23:      obtain new  $Position$  from new  $BestHash$ 
24:    end if
25:  end while
26:  Gradient algorithm ends
27:  Preparation of the next generation starts
28:  Limit population of  $Elites$  and  $Results$ 
29:   $PopList$  selected as  $Elites$  or  $Results$ 
30:  clear  $Values$ 
31:  for  $ParentA$  in  $PopList$  do
32:    select different  $ParentB$  from  $PopList$ 
33:    for  $j = 0$  to  $Length(Position)$  do
34:      breed traits of  $ParentA$  and  $ParentB$ 
35:      add child traits to  $Values$ 
36:    end for
37:  end for
38:  Preparation of the next generation ends
39: end for
40: Evolutionary algorithm ends
41: return  $Results(hash(\min(cost(Elites)))$ 
```

Algorithm 2.1. Hybrid Evolutionary Gradient Algorithm

To take advantage of both the quick optima identification ability of the gradient-based algorithm and the broad optima search ability of the evolutionary algorithm, the hybrid algorithm is proposed in this paper. This hybrid approach allows the algorithm to initially perform a broad search along the parameter space using a fixed population size with random mutations and elites and then quickly narrow down on the optimum by performing a gradient descent. In this way, the hybrid algorithm consists of the following three steps: Step 1) the evolutionary algorithm is executed first for each generation; Step 2) if a new optimum is found, the gradient algorithm is

executed with this solution as the initial guess; Step 3) if the gradient descent algorithm identifies a more optimal solution, the new solution is added to the breeding population for the next generation. This leads to the algorithm quickly finding local minima and breaking out of them over multiple generations. The simulations are run using the sampler. The sampler first builds a list of simulation commands using a set of adjustable parameters and their available values in the master processor. Then, the sampler runs the parallel executor to execute these simulations in all available slave processors in parallel. Finally, the master processor continues to post-process the simulation results including evaluating their performance values through the above proposed weighted cost function and sorting them into different categories. The detailed hybrid optimization algorithm is listed in Algorithm 1. ExecutorPool refers to the pool of workers that is maintained by the parallel executor. BestHash holds the hash of the simulation result with the lowest cost so far. Position refers to the initial parameter set the gradient algorithm perturbs. MaxPop is the general population size. PopList refers to the population used to breed Values for the next generation. Simulated annealing is implemented using a random chance to use the general population in Results to breed the next generation instead of Elites. Due to the structure of the algorithm involving an outer loop running for the maximum number of generations and three disjoint inner loops, the total computational complexity of the algorithm is determined to be $O(n^2)$.

Optimization Test Functions

To validate the performance of the above proposed parameter optimization framework, many different types of test functions were used to benchmark the optimization. For comparison, the Rastrigin, Eggholder, and Rosenbrock functions were selected specifically due to their different natures, which pose different challenges to the optimization algorithm design.

Rastrigin function: it features a periodic nature and a distinct global optimum, and it has a global minima at (0,0) with many evenly spaced local minima surrounding it, and it is defined in Eq. (2.19):

$$f(x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i)) \quad (2.19)$$

Eggholder function: it has a distinct global optima with a more pseudo-random arrangement of the local minima surrounding it. It is given by:

$$f(x) = \sum_{i=1}^{n-1} [-x_i \sin(\sqrt{|x_i - x_{i+1} - 47|}) - (x_{i+1} + 47) \sin(\sqrt{|0.5x_i + x_{i+1} + 47|})] \quad (2.20)$$

Rosenbrock function: it features a distinct global optimum within a long valley and one optimal solution at (1,1). It shows very little variation across a wide range of the input parameters, and it is defined as:

$$f(x_n) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2) \quad (2.21)$$

Three test functions were designed to accept two numerical and two nonnumerical input parameters to more accurately model the NS-3 simulation environment. The nonnumerical values were designed to emulate NS-3 protocol configuration parameters and as such were implemented using a different constant offset based on the configuration. To ensure that the global optimum is within the range of the inputs, the functions are scaled or offset as needed.

Experimental Simulation and Result

The subsequent communications infrastructure of RTC-A consists of 57 PV inverters, 275 smart meters, 10 data concentrators, and one edge router. It is divided into 10 subareas based on the location of 10 data concentrators. The BPLC-WiMAX hybrid communications design is simulated with a large set of configurable parameters of two communication models in the NS-3 simulator on top of the RTC-A.

Verification of metaheuristics optimization algorithms: Using the three above described optimization test functions, the purpose is to verify the training parameters of the proposed hybrid optimization algorithm. The training parameters considered in this paper are mutation rate (MR), mutation chance (MC), and maximum elites (ME) of the evolutionary algorithm, and the step size (SS) of the gradient descent algorithm. The mutation rate defines the maximum extent of a single mutation as a percentage of the parent trait, mutation chance is the percentage probability of a mutation occurring, and maximum elites is the maximum size of the elites as a percentage of the total population. Increasing step size decreases the number of gradient descent steps to be performed and hence impacts only computation speed. The resulting performance comparison using the two metrics of average failure cost and failure rate is conducted as below.

1. Performance comparison of three algorithms: The performance of three optimization algorithms is compared in Table 2.6: gradient, evolutionary, and hybrid. The failure rate is defined as the percentage of cases where the algorithm fails to identify the global optimum. The average failure cost is calculated as the average distance of the found solution to the global optimum upon failure. The genetic algorithm performance, shown in the first column of Table 6, indicates that the evolutionary algorithm is capable of determining the optimal solution on its own. There is certainly room for improvement, however. The failure rates and costs of the gradient descent algorithm are very high, as shown in Table 6. This indicates that gradient descent alone is not a good method for identifying the optimum solution in such cases. As expected, the verification results show that the hybrid algorithm has improved performance compared with either the evolutionary or gradient descent algorithm individually.

Table 2.6. Comparison of Different Optimization Methods

Optimization Method	Failure Rate (%) (Average Failure Cost)		
	Rosenbrock	Rastrigin	Eggholder
Evolutionary	8.44 (8.493)	11.09 (8.3148)	10.16 (7.8154)
Gradient	95 (113930)	93.33 (141430)	95 (131800)
Hybrid	4.69 (6.05)	7.03 (6.0889)	6.02 (6.1299)

- Performance characterization of the hybrid algorithm: The hybrid algorithm in terms of failure rate and average failure cost was identified as the best option in the above subsection. The effects of the parameters of this algorithm are further explored using the three different optimization test functions. The metric used to compare the performance is the solution generation. This refers to the number of generations required to obtain the optimal result. To perform this analysis, the MC, MR, and ME parameters are varied between 10%–50% in intervals of 10%. The results are shown in Figure 2.39. For each blue box plot, the central mark indicates the median of the solution generation, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. Also, the red cross indicates the lowest and highest outliers.

For the Rastrigin function, the result shown as the left red highlights in Figure 28 indicate that the best results can be achieved when the mutation chance is as low as 10%, the mutation rate is 40%, and the maximum elites is 50%. The effect of these parameters is not as large as the effect of the size of the elites list. This conclusion is derived from the fact that with higher mutation chances, the performance is still acceptable as long as the number of elites is 50%. This indicates that the optimizer is able to search the breadth of the function across the input variable range; however, it has difficulty narrowing down to the optimal solution unless supported by a bigger elite list and lower mutation chance. The average solution generation is the same or lower for all cases in this hybrid scenario compared to the genetic algorithm. This definitively indicates the benefit of implementing this hybrid algorithm for these types of functions.

For the Rosenbrock function, the blue box plot indicates that the best results are achieved when mutation chance=10%, mutation rate =40%, and maximum elites=50%, as shown in the left red highlights in Figure 2.39. The results do not indicate as much of an impact of these parameters on the solution generation, compared to the other two test functions. This indicates that although the parameters have some effect, the overall effect of these parameters does not have as large an impact on the solution generation. This may be attributed to the fact that the function has a large valley handled by the gradient descent portion of the algorithm.

For the Eggholder function, the box plot indicates that the best results are located when the mutation chance is 20% , the mutation rate is 30%, and the maximum elites is 50%, as shown in the right red highlights of Figure 2.39 Compared to the Rastrigin function, the best performance is achieved with mutation chance=20%, not 10%, and mutation rate=30% instead of 40%. This indicates that the optimizer is able to search the full width of the input variable range; however, it has difficulty narrowing down to the optimum. Thus, this function requires more aggressive mutation to avoid getting trapped in a local minimum.

From these results, it may be concluded that a setting of mutation chance=10%, mutation rate=40%, and maximum elites=50% will result in good performance across a wide variety of test functions or applications. The size of the elites poses an inherent trade-off, however. A larger number of elites will reduce the number of generations required to achieve the optimal result while also increasing the computation required per generation. A value of 50% was found to adequately balance this trade-off. These settings of the proposed hybrid optimization algorithm are applied to tune the parameters of the NS-3-based hybrid smart grid communications system design.

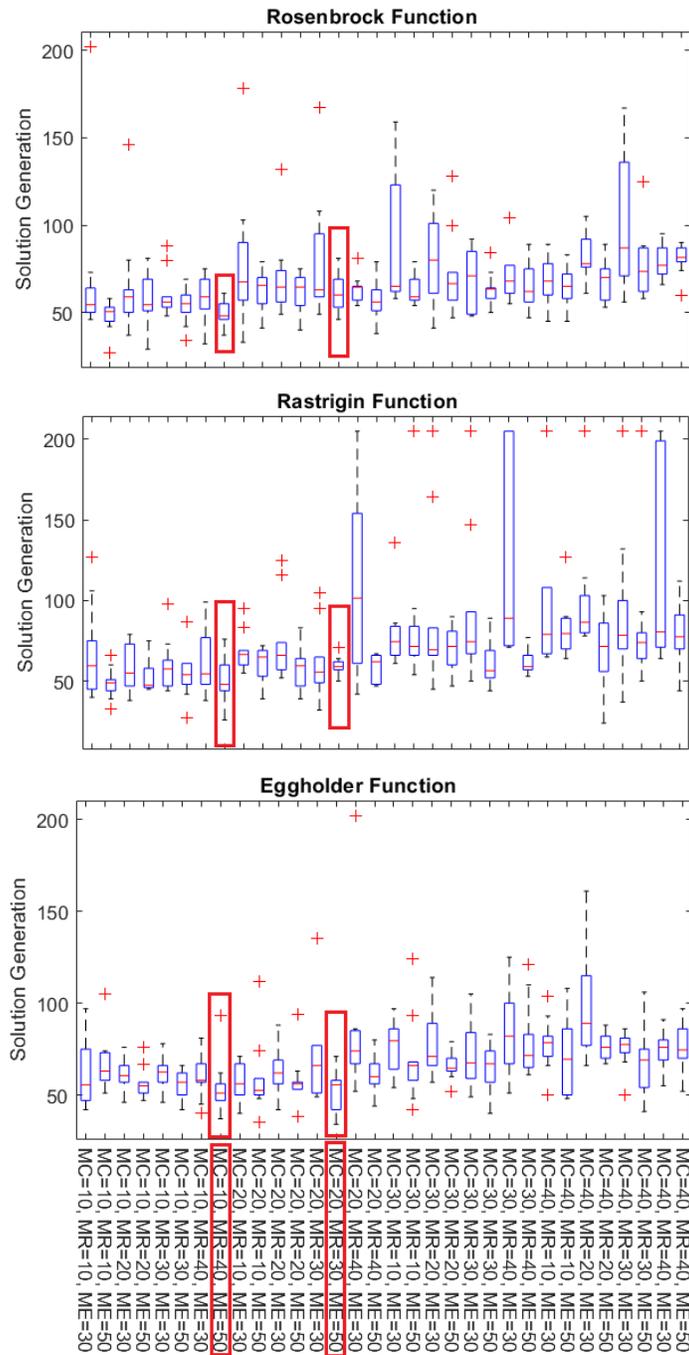


Figure 2.39. Comparison of hybrid algorithm in terms of test functions

3. Performance comparison of three algorithms: The proposed hybrid optimization algorithm takes both numeric and nonnumeric configurable parameters. The BPLC-WiMAX hybrid design to be tested has the configurable parameters shown in Table 2.7. The optimal parameters as identified by the proposed parameter optimization framework are summarized in Table 2.8. Note that (1) the optimal parameters of communications technology modes—namely, BPLC and WiMAX models—are almost coincident with the initial manual results in Table 3 and

Table 4; (2) the Best Effort service flow of the WiMAX model is also same with the optimal profile configuration of the latest developed WiGrid NS-3 module in [74]; (3) this WiGrid NS-3 module along with new configurable parameters—such as frame duration, priority-based scheduler, and unsolicited grant allocation scheme—are implemented in the future to further improve the accuracy of the WiMAX-based hybrid simulations.

Table 2.7. Configurable Parameters of the BPLC-WiMAX Hybrid Design

Model	Parameter	Values
BPLC	Low frequency	2 – 3 (MHz)
	High frequency	3 – 100 (MHz)
	Channel number	100 – 1200
	Payload modulation coding scheme	QAM64_rateless, QAM32_rateless, QAM4_rateless, QAM64_12_21, BPSK_1_2, BPSK_rateless
	Header modulation coding	BPSK_1_2, BPSK_1_4
WiMAX	Phy layer modulation	QAM16-12, QAM16-34, QAM64-32, QAM64-34, BPSK-12, QPSK-12, BPSK-34
	Service flow	UGS, RTPS, NRTPS, BE
	Propagation model	Friis, Cost231, Random, Log
	Scheduler	SIMPLE, MBQOS, RTPS
Application	Data rate	16 – 56 (kbps)
	Packet size	64 – 2048 (Bytes)

Table 2.8. Optimal Parameters of the BPLC-WiMAX Hybrid Design

Model	Parameter	Values
BPLC	Low frequency	2 (MHz)
	High frequency	3 (MHz)
	Channel number	1146
	Payload modulation coding scheme	QAM64_rateless
	Header modulation coding	BPSK_1_2
WiMAX	Phy layer modulation	QAM16-12
	Service flow	Best Effort
	Propagation model	Friis
	Scheduler	SIMPLE
Application	Data rate	55.98 (kbps)
	Packet size	66 (Bytes)

A small portion of optimal cases as determined by the optimizer for the BPLC-WiMAX hybrid design are shown in Table 2.9. To observe the effect of the parameters on the metrics, the sampler was used to determine the performance of the configuration with specific, different inputs, as shown in Table 2.10. The optimizer results shown in Table 2.9 clearly have higher granularity and identify more optimal cases compared to the sampler results. To achieve this set of optimal cases, the trade-offs made by the optimizer are apparent in observing the parameter values used. Depending upon which trade-offs made by the optimizer are acceptable, the user has the option of selecting from a large set of options, the size of which is determined by the maximum elites parameter. It is noticeable that the optimizer pushes the data rate as high as it

can while simultaneously optimizing the packet size for low latency and packet loss rates. This is exactly the behavior needed from the optimization framework.

Table 2.9. Optimal Results for Simulated Configuration

Throughput (kbps)	Latency (ms)	PLR (%)	DR (bps)	Size (B)
55.937	5.512	0.074	55978	66
55.959	5.649	0.034	55980	66
55.977	5.661	0.035	55997	66
55.979	5.562	0.035	55999	67
55.979	5.487	0.057	56000	66
55.859	5.564	0.034	55880	68
55.971	5.612	0.055	55998	68
55.948	5.571	0.056	55973	66
55.964	5.472	0.038	55985	64

Table 2.10. Results of Sampling Simulated Configuration

Throughput (kbps)	Latency (ms)	PLR (%)	DR (bps)	Size (B)
55.979	6.832	0.057	56000	64
55.884	10.466	0.213	56000	128
55.993	17.923	0.027	56000	256
55.962	34.183	0.098	56000	512
56.016	64.539	0.031	56000	1024
55.91	129.253	0.296	56000	2048

From these results it is clear that the optimizer works well in achieving the best possible network configuration within the given range of parameters.

Subtask 2.2.2: Transmission Distribution Communication Co-Simulation

***Objective:** Connect the developed communications system simulation model to the existing HELICS co-simulation platform. This work will allow for the development of the case study in the third phase, which tests the impact of the communications system architecture on the performance of the underlying physical power system.*

Integration Framework

We successfully completed the integration of the HELICS platform and NS-3 simulator. Figure 2.40 provides a schematic of the interface of the HELICS co-simulation platform and NS-3 simulator. Figure 2.40 shows that there are three components: (1) HELICS platform with three main co-simulation-related functions; (2) three interface modules: (a) Helics-simulator-implementation module; (b) NS-3-HELICS module; (c) NS-3-HELICS application module; (3) and an NS-3 script, which consists of the simulated topology and the installation of the specific HELICS client/server applications on the corresponding client/server nodes as well as invoking the HELICS-helper module to create the NS-3 federate and connect to the HELICS broker.

We next introduce three interface modules. To accommodate three main functions of the HELICS platform, we implemented the HELICS-simulator-implementation module, which mainly has three critical functions: (1) synchronizing the NS-3 simulator to the HELICS platform by requesting an external-event time and receiving the granted time; the external-event

specifically refers the receiving HELICS message from the HELICS platform, and the internal event refers to the networking event that happened only in the NS-3 simulator. (2) Scheduling the external events and internal events properly in this way, the NS-3 simulator executes the internal events through the *ProcessEvent()* built-in function and executes the external-events through the HELICS endpoint functionality. (3) Last but not least, the HELICS built-in *requestTime()* function was used to invoke the callback function of

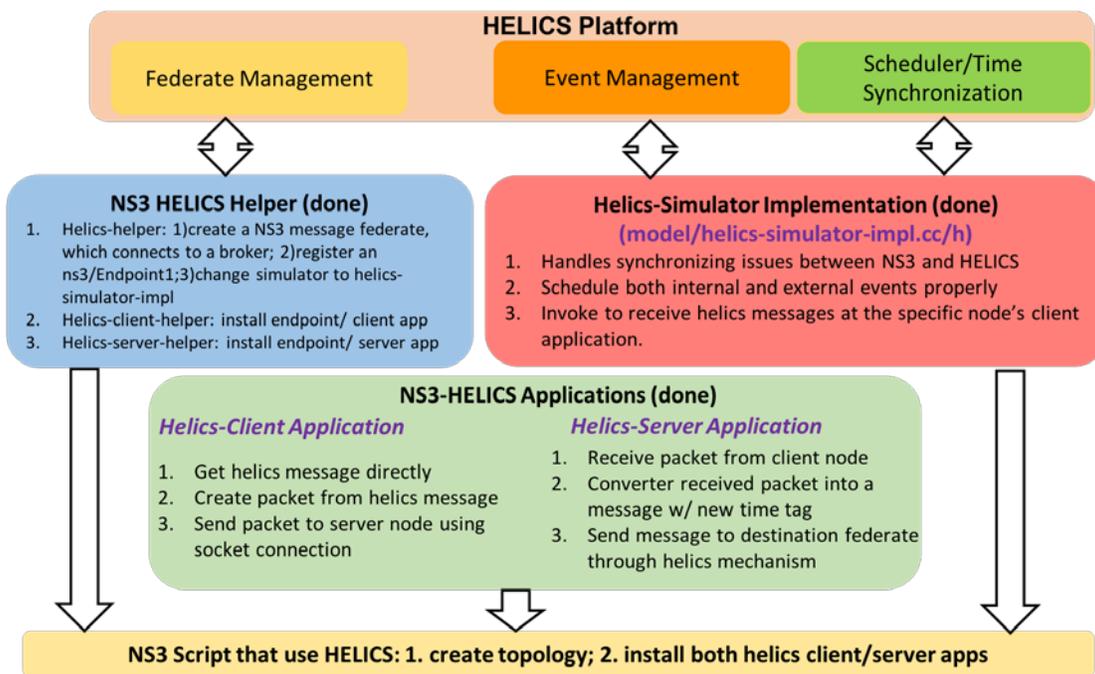


Figure 2.40. Design diagram of HELICS-NS-3 integration

end point to receive the HELICS message at all client node applications. It is critically important because the HELICS-simulator-implementation module is the open access to the NS-3 simulator, which means it is the only place to exchange message with the outside. It is not reasonable to exchange all messages through this module and then distribute them to the client nodes, especially for the large-scale NS-3 simulation. It is wise to use the *requestTime()* function executed in this module to invoke the callback function of the end point attached in all client applications of the client nodes. This embodies the scalability of integration scheme.

The second interface module of the NS-3-HELICS applications consists of two specific HELICS applications. The *Helics-client()* application has three functions: (1) receive HELICS messages directly from the HELICS platform; (2) parse HELICS messages and format the packet with the sending time stamp; (3) and send packet to the corresponding server node using socket connection. Specifically, this client application set up the end point and its callback function, which is used to receive HELICS messages, as well as set up the client socket and connects to the server socket. The *Helics-server()* application also has three functions: (1) receive the packet from the client node; (2) parse the packet and create a new HELICS message with a new time stamp; 3) send the HELICS message back to another federate through the HELICS mechanism.

The server application also set up an end point without a callback function, and it is used to send the HELICS message back to the HELICS platform as well as set up the server socket.

The third interface module of the NS-3-HELICS helper focus on two functions: (1) NS-3 federate management, *Helics-helper()* function is to create a NS-3 federate, connect to the broker, and change the default simulator to the specific HELICS simulator. (2) Installation of Helicsendpoint to all nodes. The *Helics-client-helper()* function helps install the HELICS-client-application and its end point into a group of client nodes as well as set up the protocol and remote address for the client socket. Similarly, the *Helics-server-helper()* function helps install the Helics-server-application and its end point into a group of server nodes as well as set up the protocol and port number for the server socket.

Prototype of HELICS-NS-3 Integration on Top of RTC-A

Based on the initial integration, we further developed the prototypical HELICS-NS-3 co-simulation test bed on top of RTC-A, as shown in Figure 2.40. In this prototypical test bed, we first implement a simple federate, which could be any type of power system simulator. The corresponding functions are listed in the left blue box of Figure 2.41, and one main function is to send the specific dummy HELICS message to the specific PV nodes and receive these dummy HELICS messages with a delay time from the 10 DC nodes. Then the prototypical NS-3 federate in the NS-3 simulator is created to implement the time synchronization with HELICS and exchange messages between the NS-3 federate and simple federate through the HELICS platform. In the NS-3 script, the detailed RTC-A topology includes 52 PV nodes connecting to the corresponding smart meters through LoWPAN, 275 smart meters divided into 10 groups and forwarding data to the corresponding data concentrators through Ethernet cables, and 10 data concentrators that receive the PV messages from the smart meters. At each PV node, the HELICS-client applications and NS-3 local end points are installed. At each data concentrator node, the HELICS-server applications and the corresponding end points are also installed.

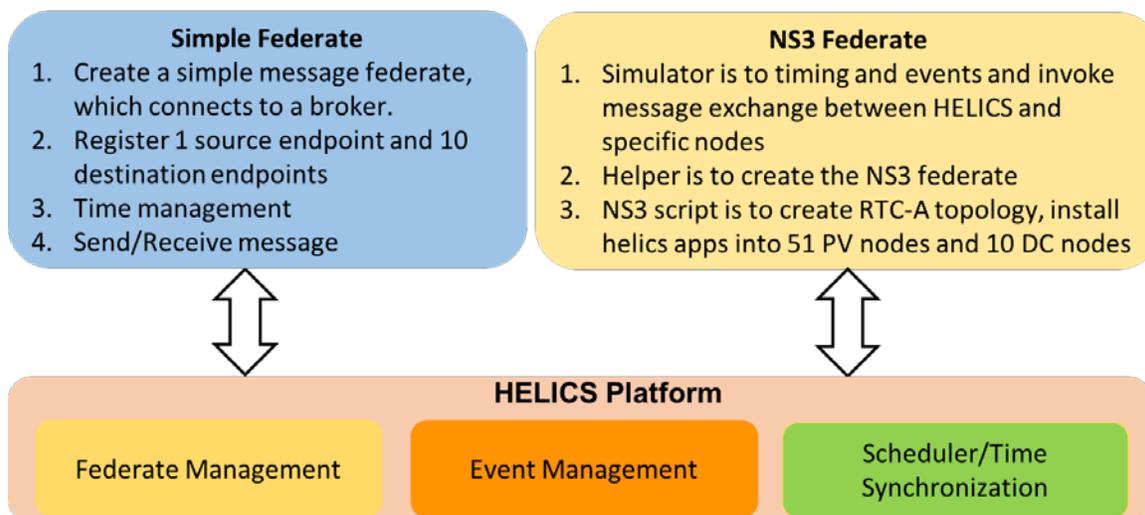


Figure 2.41. Prototype diagram of HELICS-NS-3 integration on top of RTC-A

The experimental results shows that at each iteration the simple federate can sequentially send 52 specific HELCIS messages to the corresponding PV node, each PV node receives the HELCIS message, and then sends them with sending time stamp through the simulated LoWPAN-Ethernet hybrid network to the predefined data concentrator node with receiving stamp. Then each data concentrator node receives several packets from its owned smart meters and PV nodes and calculates the delay time for each new HELCIS message, then sends them back to the specific simple federate end point. Finally, each federate end point receives the delayed messages. Figure 2.42 shows the average latency of each PV message in the LoWPAN-Ethernet hybrid network, which is around 10 ms.

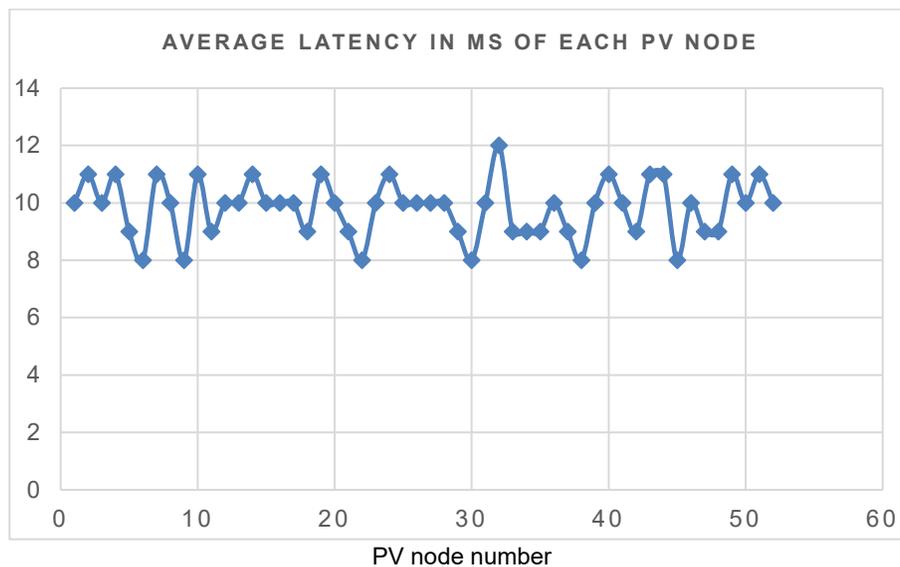


Figure 2.42. Latency performance of PV nodes of RTC-A in NS-3-HELICS co-simulation in case of LoWPAN-Ethernet hybrid network

Subtask 2.2.3: Alternative Technology Comparison

Objective: Through the communications system simulation developed for Test Case A, demonstrate the differences in communications system design if other technologies were to be deployed, i.e., PLC instead of LoWPAN or WiMAX, WiFi mesh instead of Ethernet cable.

There are two objectives in this subtask. The first objective is to validate the third hybrid system design criteria of *reliability*, which refers to three performance metrics of (1) *Latency*, the expected one-way latency for distributed energy resource (DER) control and monitoring at the distribution grid is in the range of 300 ms – 2 s; (2) *Throughput*, the requirement for DER and distributed storage is between 9.6 kbps to 56 kbps; (3) *Packet Loss Rate*, its benchmark value for DER applications is set to 0.01% – 1% [32], [33]. These criteria of packet loss rate are specific for the TCP traffic; however, we consider only the UDP traffic at this initial comparison validation stage because this benchmark setting is not suitable for our current analysis and is referred to in a later stage when we also implement the TCP/IP solution for all six alternative hybrid communications system simulations.

The second aim is to demonstrate and evaluate the comparison of the performance of six proposed hybrid communications system simulations developed for Reference Test Case A

(RTC-A) in terms of the characterization of three performance metrics of the first goal. The description of RTC-A refers to the previous Subtask 2.2.1, where RTC-A represents the modern urban power system distribution grid. Note that the alternative technology comparison does not consider the effect of opportunistic functionality at this moment.

In this section, the performance characteristics of the hybrid designs are investigated through simulations of hybrid communications networks conveying UDP traffic. In particular, the effects of alternative communications technologies on the network performance are examined to discover how a distributed PV coordination communications network should be designed with best performance metrics while remaining compatible with the existing communications infrastructure as much as possible.

To achieve these goals, we first set the optimal parameter configuration for each alternative communications model in the NS-3 simulations according to the testing results of the previous Subtask 2.2.1. Second, the developed optimal design parameter toolbox was employed to run different communications system designs and to post-process the data. The PV data traffic is parameterized according to the distributed PV coordination applications. PV packet size is thus set at 32, 64, 128, 256, 512, 1024, and 2048 bytes with the addition of UDP and IP headers. The transmission rate is set at 12, 24, 36, 48 Kbps. Several combinations of packet size and transmission rate are examined. All simulations are run at NREL's High Performance Computer (HPC) named as *aflorita*. The following results are based on up to 100 simulation runs carried out for 100 simulation seconds for each system design. This duration and times of the simulation runs can be considered long enough and many enough to estimate the performance characteristics of each hybrid communications system design because we did not obtain significantly different characteristics from results even for the duration of 1000 simulation seconds and up to 200 simulation runs on preliminary experiments. The characteristics of hybrid communications system designs are investigated in terms of the following two items of UDP packet size and UDP transmission rate.

Performance Comparison in Terms of UDP Packet Size

In this subsection, we examine comparable performance characteristics of hybrid communications system designs in terms of variable UDP packet size setting. In these simulations, all 57 PV inverters are sending the PV data packet at the rate of 24 Kbps simultaneously to the corresponding data concentrators via one or more smart meter relays. Figure 2.43 shows the performance characterization of six hybrid communications system designs in the general case—in terms of latency, throughput, and packet loss rate, respectively—and Figure 30 shows the one in the worse cases.

As shown in Figure 2.43, there is an obvious and important observation in three subfigures: the network performance, regardless of which metric are considered, is clearly grouped in terms of two alternative communications technologies applied in the PV-SM link. This phenomenon implies that the communications technology choice of PV-SM links plays a more critical role than it does on SM-DC links. That is to say, the PV-SM link design will dominate the overall performance of the designed hybrid networks. This observation also indicates that the performance difference between LoWPAN and PLC is much bigger than the difference among Ethernet cable, WiFi mesh, and WiMAX. Another outstanding observation is that the UDP packet size does have the interesting effect on the hybrid communications network design. In the

next step, we focus on how the packet size affects the network performance of each design in detail.

Regarding average latency, we have the following findings from Figure 2.43(a). (1) When increasing packet size, the performance metric of average latency of three LoWPAN-based hybrid cases is generally improved, whereas the opposite results happen to three PLC-based hybrid cases. (2) In the case of PLC, the best performance is achieved when the PV data packet size of 256 bytes is used, which is consistent with the result of initial field trials [34]; however, the case of LoWPAN, the optimal packet size is 64 bytes. (3) The most important observation is that when the packet size is set to 32–1024 bytes, three LoWPAN-based hybrid designs with optimal parameters satisfy the critical latency requirement of 300 ms. And whatever the packet size is set, the lowest bound of latency criteria are also satisfied by the LoWPAN-based designs. For the case of PLC, however, only when packet size is ≥ 256 bytes, the latency performance is accepted even according to the highest bound requirement. Otherwise, the latency is too big for DER coordination applications. (4) Regardless of the settings of the PV-SM link's technology and packet size, the Ethernet cable case always has the best performance in the SM-DC links. Note that the LoWPAN-WiFi design outperforms the LoWPAN-WiMAX, whereas the result is opposite in the PLC case. The observation indicates that there exists wireless interference when both PV-SM and SM-DC links are using the wireless communication. (5) Among six optional designs, the LoWPAN-CSMA case has the best latency performance of 8 ms along with the packet size of 64 bytes.

From Figure 2.43(b), we can observe that the throughput requirement of 9.6–56 Kbps are always satisfied in six hybrid designs regardless of the packet size setting. Compared to the PLC case, the LoWPAN case has much less effect of the packet size on the throughput performance. The packet loss rate in the case of UDP is further analyzed when compared with the TCP case.

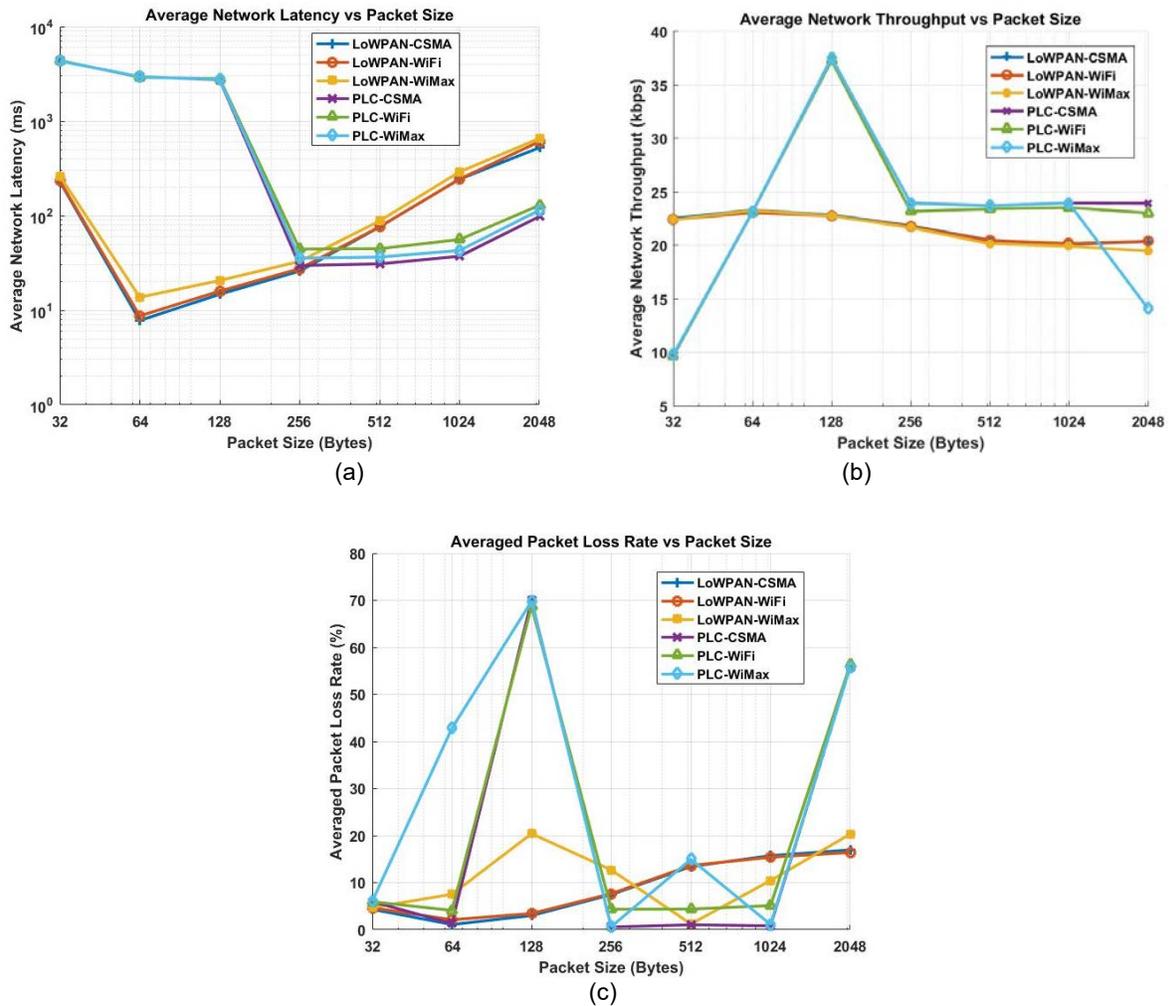


Figure 2.43. Performance characterization of six hybrid communications designs in terms of UDP packet size: (a) average latency, (b) average throughput, (c) average packet loss rate

Performance Comparison in Terms of UDP Transmission Rate

In this subsection, we further investigate comparable performance characteristics of hybrid communications system designs in terms of UDP transmission rate. As shown in Figure 2.44, both average latency and packet loss rate are degraded for six hybrid designs when the transmission rate increases, whereas the average throughput is improved dramatically. As a result, in determining an appropriate transmission rate for the UDP packet, there can be a trade-off between throughput performance improvement and the performance degradation of latency and packet loss rate.

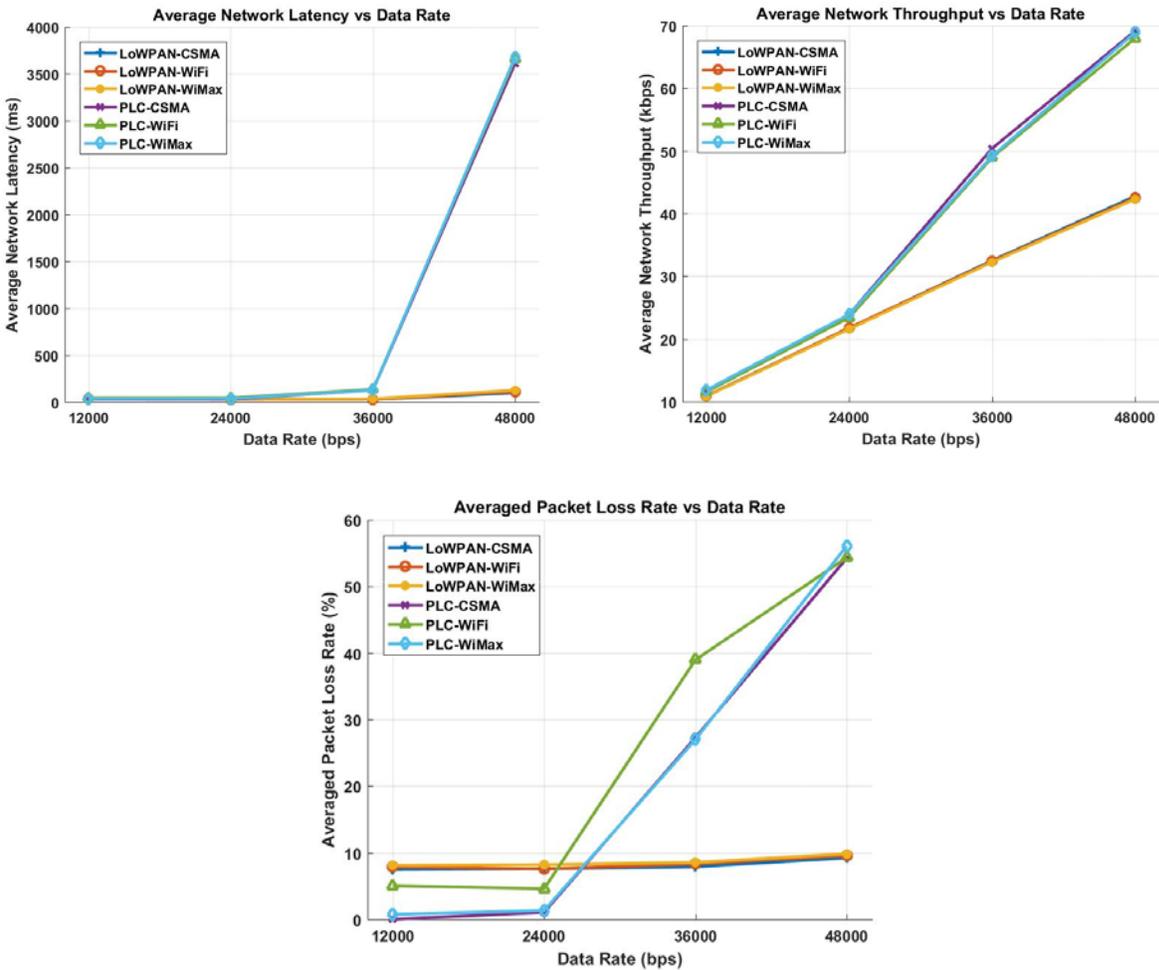


Figure 2.44. Performance characterization of six hybrid communications designs in terms of UDP transmission rate: (a) average latency, (b) average throughput, (c) average packet loss rate

Task 2.3: Technical Review Committee

The second year of the TRC involved an additional peer review of results as well as the initial steps toward developing closer relationships with some of the TRC member organizations who might be interested in a pilot program of the communications system developed after the validation phase of Year 3 is completed. In addition, conference calls with other SuNLAMP teams with similar projects, i.e., SLAC and LBNL, were held to discuss how the projects could coordinate.

Subtask 2.3.1: TRC Meeting

Objective: With the TRC participants already defined, the team reconvened the TRC to receive feedback on the results obtained during the first two budget periods. The third and fourth TRC meetings occurred during Budget Period 2.

First TRC Meeting (09/30/2016)

The NREL team met with three industry personnel:

Elaine Sison-Lebrilla: She is a senior project manager at the Sacramento Municipal Utility District (SMUD) with nearly 25 years of experience in energy. She manages the Renewable Energy Program in the Energy Research and Development department. Previous to SMUD, Sison-Lebrilla worked for the California Energy Commission as manager of the Geothermal Program and the Energy Generation Research Office under the Public Interest Energy Research Program. Sison-Lebrilla has a Bachelor's of Science degree in electrical engineering from the University of California at Berkeley and is a registered professional engineer in California.

Sison-Lebrilla was involved in a project validating the solar forecast over the 2330 km² of SMUD service territory, a partnership with NEO Virtus Engineering for installation, maintenance, and data collection from 71 irradiance sensors deployed over the region. The technical details are published in [35] and [36].

She showed interest in being involved in the TRC. In particular, she mentioned SMUD's third-party dependence on communications system and is interested in the outcome of this project to see how such dependency can be overcome.

Frances Cleveland: She is president of Xanthus Consulting International [37] and has consulted on smart grid information and control system projects in the electric power industry for more than 36 years. Cleveland's expertise has focused primarily on information interoperability standards, cybersecurity issues, and integration of systems, including distribution automation (DA), distributed energy resources (DER), substation automation, SCADA and EMS systems, advanced metering infrastructures (AMI), electric vehicles (EV), and energy market operations. She is currently consulting to NIST as a Technical Champion for the Smart Grid Interoperability Panel (SGIP) on DER and cybersecurity and to EPRI on the National Electric Sector Cyber Security Organization Research (NESCOR).

In the International Electro-technical Commission (IEC), Cleveland is convener of IEC TC57 WG15 for IEC 62351 cybersecurity standards for power system operations and is the editor for IEC TC57 WG17 for IEC 61850-7-420 information standards for DER, EV, and DA. In the IEEE, she is past chair of the IEEE Power and Energy Society's (PES) Power System Communications Committee (PSCC).

Cleveland has a bachelor's degree in applied physics and electrical engineering, Harvard University; a master's degree in electrical engineering and computer science, University of California at Berkeley; and a master of business administration from San Jose State University.

She also showed interest in being involved in the TRC. Note that Cleveland is also involved in the Smart Inverter Working Group (SIWG) for the State of California [38] to develop the IEC 61850 information model-based IEEE 2030.5/SEP 2.0 protocols to enable advanced communications and other functionalities for inverter-based distributed energy resources (I-DER). Consequently, her experience and advice is a valuable component for the successful completion of this project.

Greg Smith: As a technology strategy architect at San Diego Gas and Electric Company (SDG&E), Smith is responsible for the company's efforts in the California Smart Inverter Working Group (SIWG). He also led development of the Common Smart Inverter Profile (CSIP)

and proposed default means of communications with smart inverters. Currently, he is supporting the acquisition and integration of battery resources into the California Independent System Operator (CAISO) marketplace. He encourages statistical sampling because the smart meters may not be able to provide sufficient information as required by the DSO. The distribution management system at SDG&E is centralized, and hence they are interested to see the performance of the decentralized approach being developed in NREL's project. He is a graduate of University of California at San Diego.

Second TRC Meeting (05/04/2017)

The NREL team met with Elaine Sison-Lebrilla and Frances Cleveland on May 4 (Thursday) at 3:00 p.m. MDT. We received the following valuable feedback:

- Capability of current vs. future of smart meters
- Communications capability of PV inverters
- Current and future of communications architectures of distributed PV inverters
- The way to achieve local voltage profile monitoring in the automated fashion
- Identify the use cases of the developed communications systems for motivation.

Subtask 2.3.2: GMLC Projects Meetings

***Objective:** Coordination meetings were held with other GMLC projects in similar areas. This allowed for coordination between the projects, as well as learning from the experience of the other teams, i.e., SLAC and LBNL.*

NREL's SuNLaMP Project

NREL's SuNLaMP project, titled "Opportunistic Hybrid Communications Systems for Distributed PV Coordination (HybridComm)," aims to implement a full-scale opportunistic hybrid communications system to provide bilateral information sharing for enhancing efficiency and reliability of the power grid with high penetrations of distributed PV resources. NREL's main objective is to build an opportunistic hybrid communications system, including (1) attack-resilient middleware architecture, (2) communications architecture design, and (3) NS-3-based communications system simulation. HybridComm also involves the development of automatic partitioning of the power system network as well as distributed algorithms for state estimation. In addition, the optimal resource placement in communications networks of distribution grids is an essential part of this project for gaining the best possible knowledge of behind-the-meter PV contribution to the grid. HybridComm specifically studies how additional distributed PV information impacts T+D+C system operations.

Coordination Meeting with SLAC (Date: 12/06/2016)

Researchers from the Stanford Linear Accelerator Laboratory (SLAC) are working on a SuNLaMP project entitled "Visualization and Analytics of Distribution systems with deep penetration of Distributed Energy Resources (VADER)." VADER is a unified *data analytics* platform that enables the integration of massive and heterogeneous data streams for granular real-time monitoring and control of distributed energy resources (DER) in distribution grids. This Web-based open-access visualization and analysis toolbox is developed on a machine learning platform by applying *big data analytics* and *statistical inference over the large volume*

of real time data streaming from distribution substations. It is focusing on the big data analytics and machine learning application for DER integration in distribution system. The prime objective is to build the VADER system, including (1) system architecture design, (2) distributed database, (3) data plug and validation module, (4) dashboard and dynamic visualization capability; and to develop data-driven machine learning algorithms, including (1) advanced topology identification, (2) distribution grid detection and state estimation, and (3) sensor placement algorithm.

There are some similarities in the general research questions, methodology, and evaluation metrics for the VADER and HybridComm projects. The SLAC research group studies the same basic question of how to improve operations and control of the distribution system with deep penetrations of DERs using a data-driven approach. Both projects employ such research methodologies, which consist of (1) developing data communications/analytics systems and data-driven machine learning algorithms for planning and operation of distribution networks; and (2) validating them using utility data and hardware-in-the-loop (HIL) simulation. From a system perspective, they share some evaluation metrics, including scalability, reliability and real-time, robustness, small predictive error, as well as interoperability.

Unlike NREL, SLAC's research does not involve the communications network, the T+D+C co-simulation framework, or distributed PV state estimation algorithms. Rather, the main focus of VADER is to investigate how full topology and measurement information impact operations of distribution grids.

It was concluded that there is a potential to integrate the developed communications systems (under HybridComm project) with the VADER platform to verify distributed machine learning algorithms.

Coordination Meeting with LBNL (Date: 12/21/2016)

The CyDER (Cyber Physical Co-Simulation platform for **D**istributed **E**nergy **R**esources in Smart Grids) project led by Lawrence-Berkeley National Laboratory (LBNL) is developing a scalable (up to 10,000 nodes), modular, and interoperable T+D+C co-simulation platform. In particular, simulation tools for transmission (GridDyn developed by Lawrence Livermore National Laboratory), distribution (utility software CYMDIST), building models (Modelica), and communications networks (OMNeT++) are designated as Functional Mockup Units (FMU) to integrate over the Virtual Grid Integration Laboratory (VirGIL), a co-simulation platform developed by LBNL. The high level of PV penetration is also accommodated in the CyDER platform, offering quasi-static time-series co-simulation and optimization, real-time data acquisition capability, as well as HIL applications. To validate the simulation outcome, CyDER pulls in distribution-level micro-synchrophasor data from the Berkeley Tree Database (BTrDB). A Web interface CyDER is also being developed to ease the planning, study, and analysis in the presence of high PV penetration. This will enable a “plug/click/approve and play” type of interface as opposed to the conventional bureaucratic approach in deciding potential PV customers.

It was concluded that there is potential to integrate the developed algorithms for distributed state estimation, optimal placement of communications, and measurement sensors (under HybridComm project) with the CyDER platform.

Task 3.1: Hardware-in-the-Loop Case Study

With the opportunistic hybrid communications system fully developed, the focus shifts toward validation of the system on realistic test cases. For small-scale systems, we test the capabilities of the system with a HIL case study. HIL simulation can provide a more complete validation of the fully developed opportunistic hybrid communications system by integrating physical small-scale communications system and distributed PV inverters into the simulated RTC-A power system within the testing procedure. This also allows for a more accurate representation of the actual inverter capabilities as well as a verification process for the coordination algorithms of PV inverters.

Subtask 3.1.1: Setup of HIL Test bed

Objective: *Develop distribution system scenarios with greater than 100% local PV penetration. This step is the design of the experiment that ensures that the communications system meets the response time, availability, and scalability goals in an ultrahigh distributed PV scenario. The size of the developed system is anticipated to be on the order of 10,000 nodes, with multiple physical inverters included.*

Test Objectives

Four fundamental HIL test objectives have been identified, as shown in Figure 3.1.

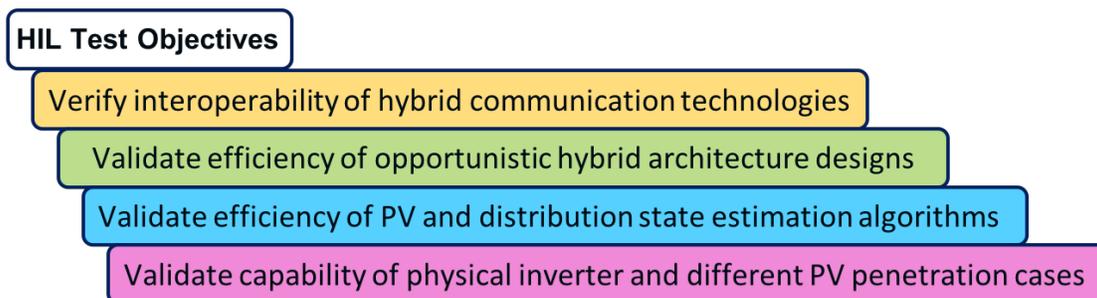


Figure 3.1. HIL test objectives

- *Verify interoperability of hybrid communications technologies.* Because we plan to set up the PLC link and Zigbee link between the PV inverters and the corresponding smart meters, and set up Ethernet cable and WiFi mesh from the smart meters to the predefined data concentrator, we have four hybrid designs to verify in the NREL Power System Integration Laboratory (PSIL). The dummy data set is sent out from each to-be-tested PV and smart meter communications node to the specific data concentrator nodes through four alternative hybrid architectures to validate three system design criteria—namely, a single trip latency of 300 ms, packet loss rate of 1%, and throughput of 9.6 Kbps.
- *Validate efficiency of opportunistic hybrid architecture design.* The opportunistic feature of the hybrid architecture design is embodied through this proposed data-driven middleware. Two main functions of the data-driven middleware are evaluated in this HIL test bed: (1) redundancy in communications media of our hybrid networking infrastructure to mitigate the impact of link failure; (2) control mechanism in our middleware architecture to mitigate the denial-of-service (DoS) attacks. Additionally,

three design criteria are evaluated if they are satisfied under these two contingency scenarios.

- *Validate efficiency of PV and distribution state estimation algorithms.* Four proposed hybrid communications systems are validated if they can support the efficient operation of the proposed distributed ladder-iterative belief propagation (LIBP) distribution system state estimation and distributed multi-rate and event-driven Kalman Kriging (MREDRIKK) PV system state estimation algorithms. In other words, for the PV state estimation application, we validate whether all the PV data from physical PV inverters and simulated PV inverter models in Opal-RT can be transmitted to the corresponding data concentrator nodes within the required delay and availability metrics as well as the throughput metrics. For the distribution system state estimation, we validate whether all the practical smart meter measurements and simulated measurements from Opal-RT can be delivered to the specific data concentrator nodes within the required single-trip delay and availability metrics as well as the throughput metrics.
- *Validate control capability of PV inverters and different PV penetration cases.* The PV power state estimation-based active power curtailment algorithm is evaluated under different PV penetration scenarios [1]. The main steps of this algorithm implementation include: (1) each data concentrator streams the on-site PV power measurements from either physical PV nodes or simulated PV modules in Opal-RT with high-resolution requirement of 15 s; (2) the MREDRIKK PV power forecast algorithm is executed to achieve the estimated active PV power generation in each data concentrator; (3) the active power curtailment algorithm is executed to calculate the curtailment threshold value in each data concentrator; (4) each data concentrator dispatches these curtailment threshold control signal to each PV inverter within its own subarea. We mainly evaluate the end-to-end delay and packet loss rate of active curtailment threshold control signal.

Based on these four objectives, we have the performance metrical and functional requirements and the corresponding test scenarios, as follows:

The purpose of the HIL testing is the validation of the proposed opportunistic hybrid communications systems to meet the following four system design metrical requirements and two functional requirements: C1 (interoperability with mixed communication technologies), C2 (round trip response time/latency of less than 1 second), C3 (availability of greater than 99%, or packet loss rate of less than 1%), C4 (throughput of greater than 9.6 Kbps), C5 (redundancy in communication media against link failure), C6 (control mechanism against denial-of-service (DoS) attacks). To validate metrical and functional requirements C1-C6, the proposed communications architecture is tested for RTC-A under three major scenarios:

- Dummy open-loop application without contingency
- Dummy open-loop application with link failure
- Dummy open-loop application with DoS attacks
- Distributed ladder-iterative belief propagation (LIBP) distribution system state estimation open-loop application without contingency

- Distributed multi-rate and event-driven Kalman Kriging (MREDRIKK) PV system state estimation open-loop application without contingency
- PV power state estimation based active power curtailment close-loop application without contingency under different PV penetration cases

Table 3.1 shows the mapping of the test scenarios to metrical and functional requirements.

Table 3.1. Mapping of Scenarios of Metrical and Functional Requirements

Scenario Description	C1 Interoperability	C2) Response time < 1 s	C3 Availability > 99%	C4 Throughput > 9.6 Kbps	C5 Redundancy against link failure	C6 Control against DoS attack
Dummy open-loop application without contingency	X	X	X	X		
Dummy open-loop application with link failure		X	X	X	X	
Dummy open-loop application with DoS attacks		X	X	X		X
Distributed LIBP algorithm without contingency		X	X	X		
Distributed MREDRIKK algorithm without contingency		X	X	X		
Active power curtailment control without contingency		X				

Architecture Design of HIL Test System

The design schematic diagram of this HIL test system is shown in Figure 3.2. The system consists of the physical subsystem and the cyber subsystem. In the physical subsystem, the Opal-RT is used to simulate the RTC-A power distribution grid, shown in the left side of Figure 3.2. The Chroma grid simulator is used to connect the physical PV inverter's AC supply to the Opal-RT. Each PV inverter is supplied by the PV simulator with DC. Each smart meter is connected to the Opal-RT through the analog Input/Outputs, which are mapped with the corresponding bus information in RTC-A. The home area network consists of the physical PV inverter and the attached smart meter through PLC and Zigbee. The neighborhood area network consists of smart meters and data concentrators through WiFi mesh and Ethernet cable. Each PV inverter or smart meter is equipped with Intel board to enable advance computation and communication functions.

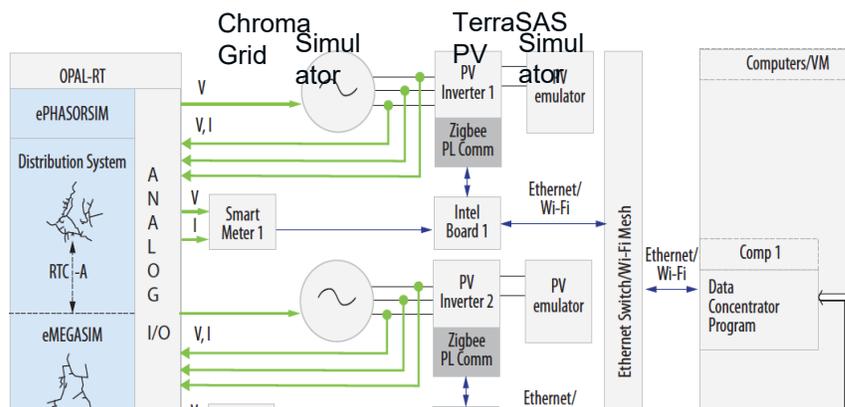


Figure 3.2. Schematic of HIL test system

The physical diagram of the HIL test system is shown in Figure 3.3. In this figure, we can see the physical Opal-RT power grid simulator, Chroma grid simulator, TerraSAS PV simulator, Solar Edge single-phase inverters, and Shark 200 smart meters. The TerraSAS PV simulator supplies the direct current to the solar energy single-phase PV inverter and the Chroma grid simulator regulates the voltage of the PV inverter, where the power signal is shown as the red line in Figure 3.3. The PV inverter pushes the current analog signal into the Opal-RT as the power input to the simulated grid, shown by the orange dashed line. Each smart meter directly is connected to the Opal-RT through its analog output signal, also shown by the orange dashed line. The PLC/Zigbee communications link between the PV inverter and the attached smart meter is shown by the green arrow line. The WiFi mesh/Ethernet communications link among smart meters and data concentrators is shown by the light blue arrow line. The control command for the PV inverter can be delivered through the hybrid communications network, and the control command for other power devices can be transferred from the data concentrator to the Opal-RT through the Ethernet cable, shown by the bright blue line.

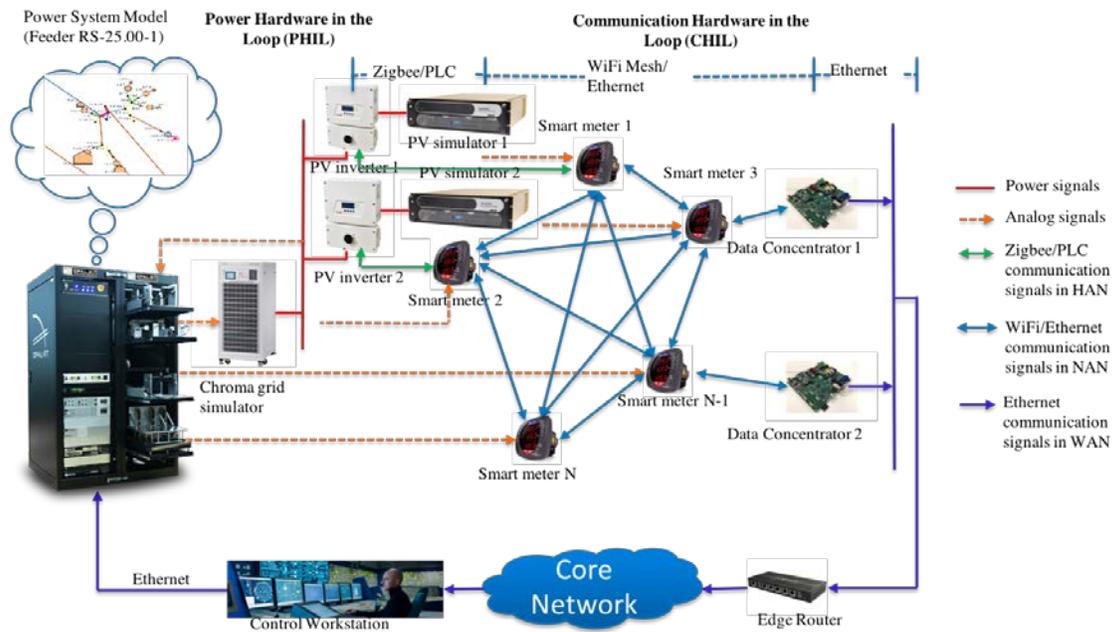


Figure 3.3. Physical diagram of HIL test system

Layout of HIL Test System in ESIF and Hardware Devices

The layout of HIL test system at ESIF is shown in Figure 3.4. The personal Mac laptop works as the data concentrator and the attached USB adaptor enables both Ethernet cable and WiFi connections.

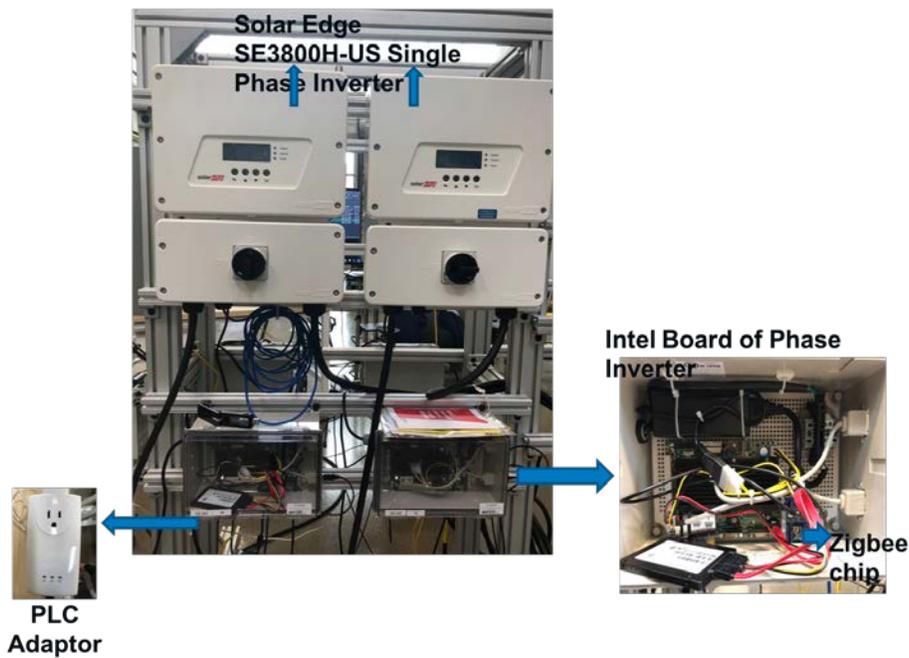


Figure 3.6. PV Inverter with Zigbee/PLC

Figure 3.6 shows the physical Solar Edge SE3800H-US single-phase inverter and its two communications devices of PLC adaptor and Zigbee chip located in the Intel board of the PV inverter.

Communications Hardware Configuration for HIL Testing

The devices to test have been individually verified to be fully functional and have been connected according to Figure 3.7. All links in the diagram have been individually verified.

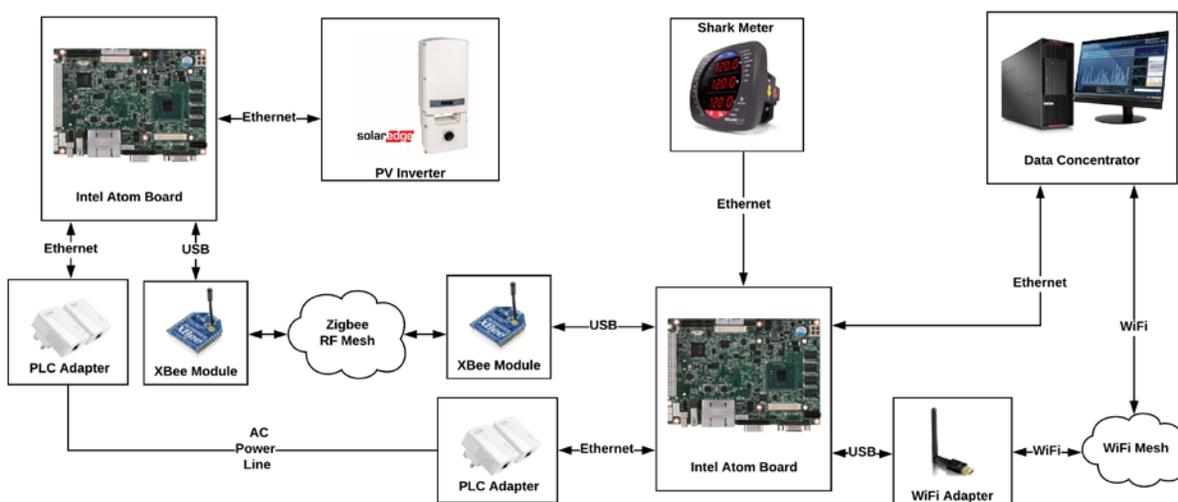


Figure 3.7. Communications hardware connection diagram

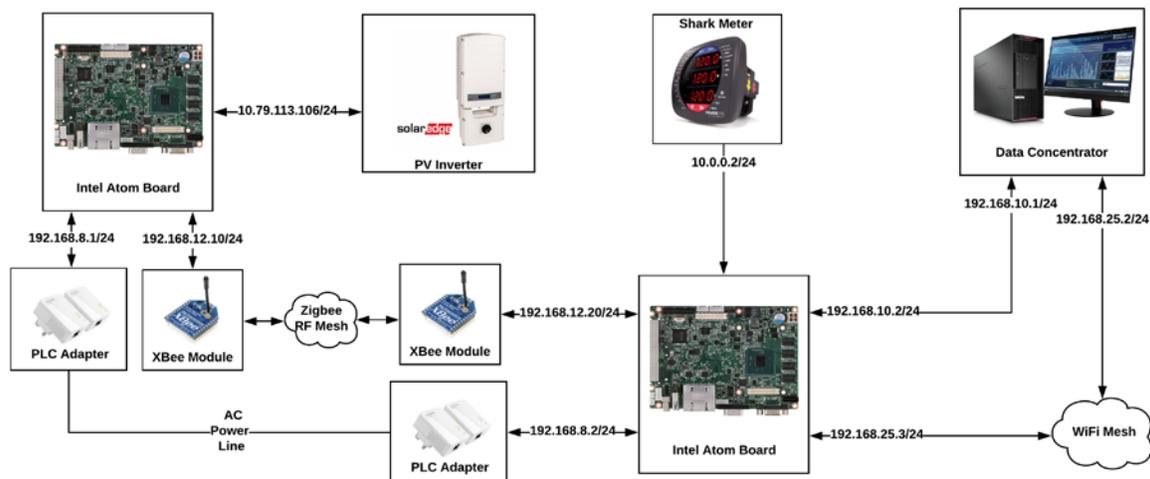


Figure 3.8. Communications hardware network diagram

The links have been set up according to the addressing scheme shown in Figure 3.8. The communications pathways have been tested by sending sample packets obtained from a basic configuration for both the inverter and smart meter. The example logs show the data available at the current stage. The accuracy of the time stamp reporting and the simultaneous use of the multiple communications links is yet to be verified. Note that each communications node is synchronized to the apple time server to enable the accuracy of the time stamp. Once the veracity of the data is confirmed, additional data points may be added to the packet to generate more detailed logs of system performance. Additional avenues that may improve performance are to tune system kernel parameters or switch to a low latency kernel to try and improve network latency. Also, techniques to reduce system overhead and latency were used after the initial improvements have been made to the script.

Development of RTC-A's ePHASORSIM model

We have the GridLAB-D model of the RTC-A, which is a quasi-static time-series tool and needs to be converted into the real-time dynamic phasor simulator (ePHASORSIM). GridLAB-D is an open-source agent-based simulation tool. It means that (1) changing states of each device is modeled independently; (2) iterations between individuals are captured; and (3) the environment in which agents evolve emerges from interactions. GLM files describe the simulation. In the GLM file, (1) each physical power device is described into an object, which belongs to different modules such as buildings, electrical network, and generators. (2) Inputs and output objects are needed to be constructed as players, recorders, and collectors.

Although the ePHASORSIM's phasor domain solver typically performs at a time step of a few milliseconds, providing voltage and current information as phasor magnitudes and angles, the ePHASORSIM solver block is typically integrated into a larger Simulink/RT-LAB model. Any status change to the feeder components are fed externally in the Simulink environment to the ePHASORSIM solver block. The feeder topology is interfaced with ePHASORSIM in an Excel template saved as either ".xls" or ".xlsx". For the RTC-A distribution feeder, the system is based on unbalanced three-phase models. In the Excel workbook, all the components, their required

parameters, and initial values are defined. The resulted ePHASORSIM model of RTC-A converting from its GridLAB-D model is described in the following Excel pages:

1. General

It contains the basic information of the power system. In this page, the base frequency of RTC-A is 60 Hz and its MVA is 100.

2. Pins

Outgoing information from the Simulink/RT-Lab blocks and incoming information to the Simulink/RT-Lab blocks are represented in this sheet. There are two types of outgoing and incoming pins.

- A. Outgoing: define measurement probes or status monitoring in the power system. In the RTC-A, we define five outgoing items, and they are the voltage magnitudes and angles of node 217, current magnitudes and angles of node 13, and real and reactive power of load 65.
- B. Incoming: send operational commands to the simulator (e.g., applying faults) or fed in voltage, current, active, and reactive power signals from Simulink models or HIL. In the RTC-A, we set two incoming signals from the Simulink models, and they are the tap positions of transformer 118 and the setting values of real and reactive power of load 65.

3. Bus

This sheet contains all the bus ID information along with voltages and angle information for the initialization of the simulation run. The RTC-A feeder consists of a total of three slack single-phase buses with base voltage of 46 kV, 1000 primary-level PQ single-phase buses with base voltage of 83 kV, and 22 secondary-level PQ single-phase buses.

4. Vsource-three-phase

The information about the substation bus and voltage with its series impedance is provided in this sheet.

5. Multiphase transformer

Transformer information in the RTC-A feeder is provided in this sheet. Two voltage regulators are implemented at the transformer by changing the tap position with the range from -16 to 16 via an external signal of the Pins sheet. This sheet contains one regulator, one substation transformer, 19 3-phase transformers, and 27 single-phase transformers.

6. Multiphase line

Both three-phase pi-section model and single-phase line information are provided in this sheet. Overhead, underground, and service lines are described in this sheet. In the RTC-A feeder, there are a total of 229 three-phase lines and 131 single-phase lines.

7. Multiphase load

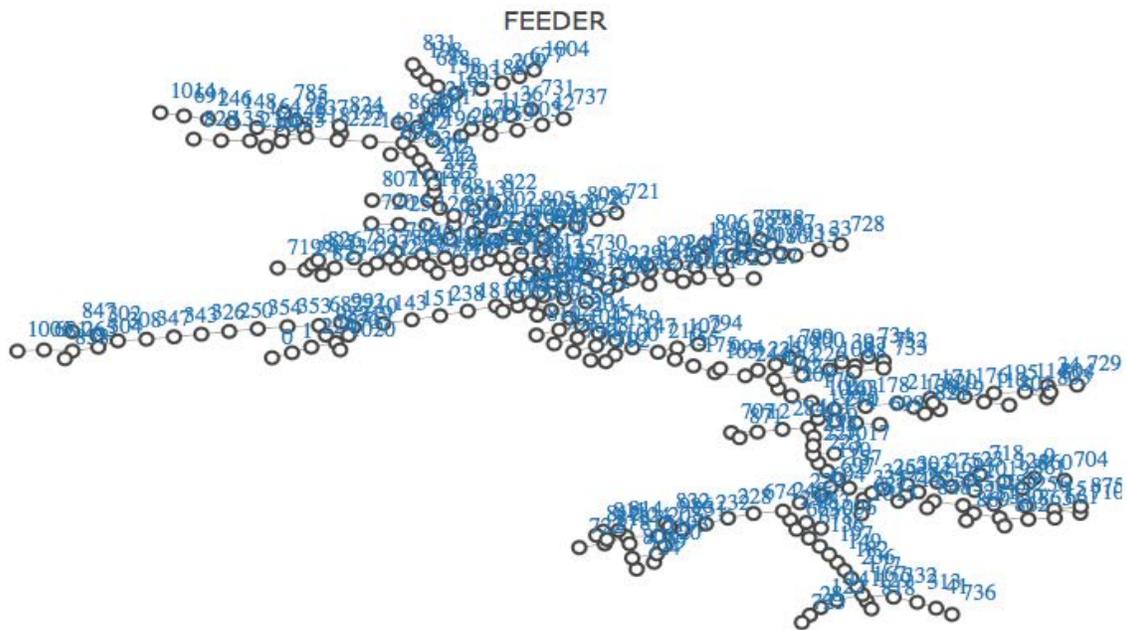
Both three-phase and single-phase ZIP loads are a combination of constant impedance, constant current, and constant power loads. In the RTC-A feeder, the portion of each load

of K_z , K_i , and K_p are set to 0.2, 0.4, and 0.4, individually. There are 10 three-phase loads and one single-phase load.

8. Switch

Switches such as breakers and disconnects are described in the “Switch” sheet. A switch is open if the status is one and closed if the status is one. Similar to the transformer tap control command, the on/off commands are given externally. In RTC-A feeder, there are 83 single-phase switches with closed status.

The converted ePHASORSIM model is shown in Figure 3.9.



(a) Topology of Phase A

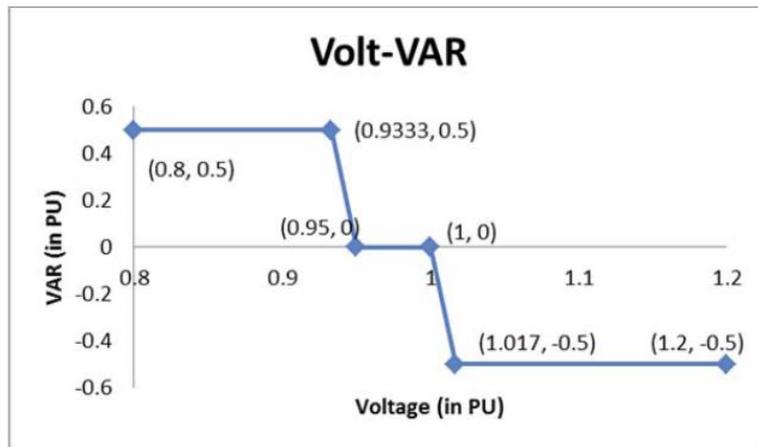


Figure 3.10. Volt-VAR control algorithm

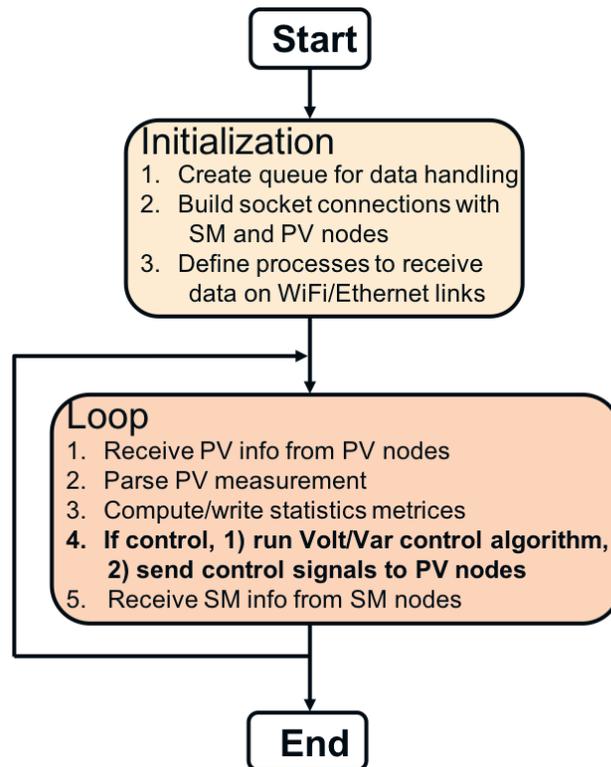


Figure 3.11. Flowchart of PV inverter/smart meter monitoring and Volt-VAR control at data concentrator

In the smart meter, the meter data monitoring and inverter data transfer function are implemented, shown in Figure 3.12.

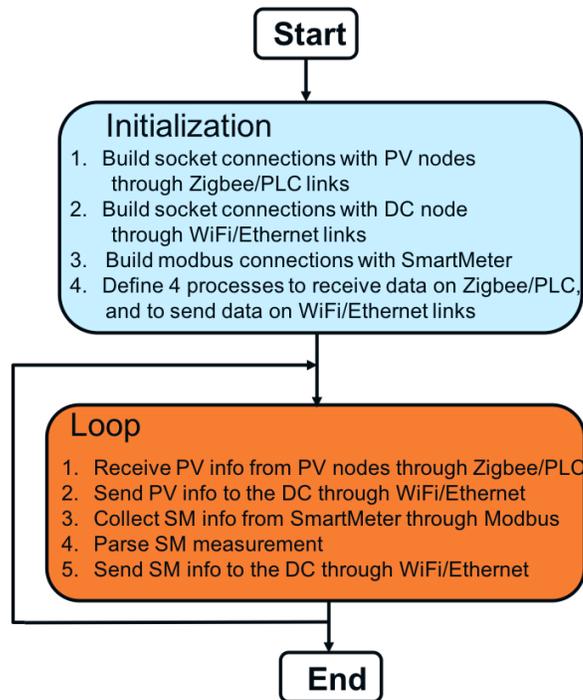


Figure 3.12. Flowchart of smart meter monitoring and PV inverter data transfer function at smart meter

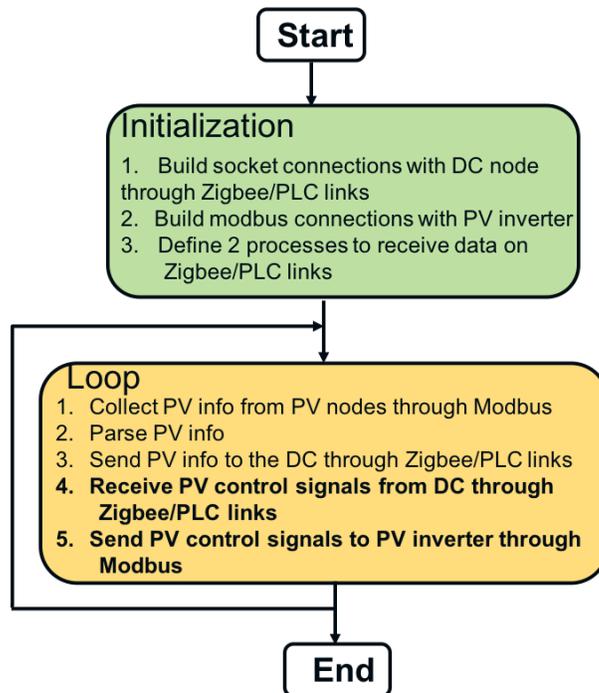


Figure 3.13. Flowchart of monitoring and control function at PV inverter

In the PV inverter, the corresponding monitoring and control are implemented and shown in Figure 3.13.

Results and Analysis of HIL Test

We initially conducted the monitoring function test and aimed to answer three questions: (1) whether four hybrid designs can satisfy the latency and packet loss rate requirements; (2) which subnetwork design dominates the whole hybrid network performance; and (3) how different traffic impacts the performance. We consider three scenarios in terms of traffic types: (1) PV data traffic on four hybrid designs, (2) smart meter data traffic on two types of NAN, and (3) both PV and smart meter data traffic on four hybrid designs. The PV data sample rate is 0.5 sample per second, and each sample with 14 bytes length includes 4 bytes grid voltage, 3 bytes PV inverter name, 3 bytes link name, and 4 bytes time stamp. Thus, the sending rate of the PV data traffic is 7 bytes/s and more in practice. The smart meter sample rate is the same with the PV data, about 0.5 sample per second, and each sample has 72 bytes measurement data, 3 bytes PV inverter name, 3 bytes link name, and 4 bytes time stamp, and total has 82 bytes length. The measurement data from the smart meter include three-phase line-to-line voltage, current, real power, reactive power, apparent power, and power factor. Thus, the sending rate of the smart meter data traffic is 41 bytes/s.

Table 3.2. Results of Monitoring Function at HIL Testing

Scenario	Latency (ms)	Packet Loss Rate (%)
SM_WiFi	6.48	0
SM_Ethernet	1.95	0
PV_Zigbee_WiFi	1311	0
PV_Zigbee_Ethernet	839	0
PV_PLC_WiFi	1302	0
PV_PLC_Ethernet	837	0
PVSM_Zigbee_WiFi	1210	0
PVSM_Zigbee_Ethernet	804	0
PVSM_PLC_WiFi	1237	0
PVSM_PLC_Ethernet	786	0

Table 3.2 shows the results of the monitoring function at three scenarios at the HIL testing. Note that the packet loss rate for all scenarios are zero, which is satisfied with less than 1% packet loss rate. Regarding the latency performance, we have the following observations: (1) Due to the high bandwidth of WiFi and Ethernet, in case of Scenario 2 with smart meter data traffic, the latency

of the WiFi and Ethernet-based NANs are 6.48 ms and 1.95 ms, respectively, although the smart meter data rate of 41 bytes/s is much higher than the PV data rate of 7 bytes/s, and they are much less than the requirement of 300 ms. (2) From the results of scenarios 1 and 3 involving PV data traffic, the latency performance for all four hybrid designs are pretty high, around 1 second, which is not satisfied with the requirement of 300 ms. These results indicate that the link bandwidth of HAN denominates the latency performance of the hybrid networks due to the lower capacity, which is consistent with the simulation results. (3) Compared to the latency performance of Scenario 1, with only PV data traffic, and Scenario 3, with both PV and smart meter traffic, we can realize that the traffic types do not impact the latency performance because of the limited bandwidth in HAN and sufficient bandwidth in NAN.

Task 3.2: Computational Case Study

Because the scalability of the system cannot be well validated in a real system, or through HIL testing, we integrate the previously developed simulation models with the HELICS platform to validate the power system effects of the communications at large scales. The objective of this task is to conduct a computational case study that tests the ability of the designed communications system architecture and algorithms to meet the SunShot goals. Of chief concern is the issue of scalability, and the case study includes more than 1 million PV generators and represents both the transmission and distribution systems. In addition, it provides a coupled simulation that overlays the communications layer with a representation of the physical power layer through an integration of the communications system simulation with HELICS. This ensures that the communications system developed is able to meet the performance standards required in a high distributed PV penetration scenario where both the communications and power layers are explicitly considered.

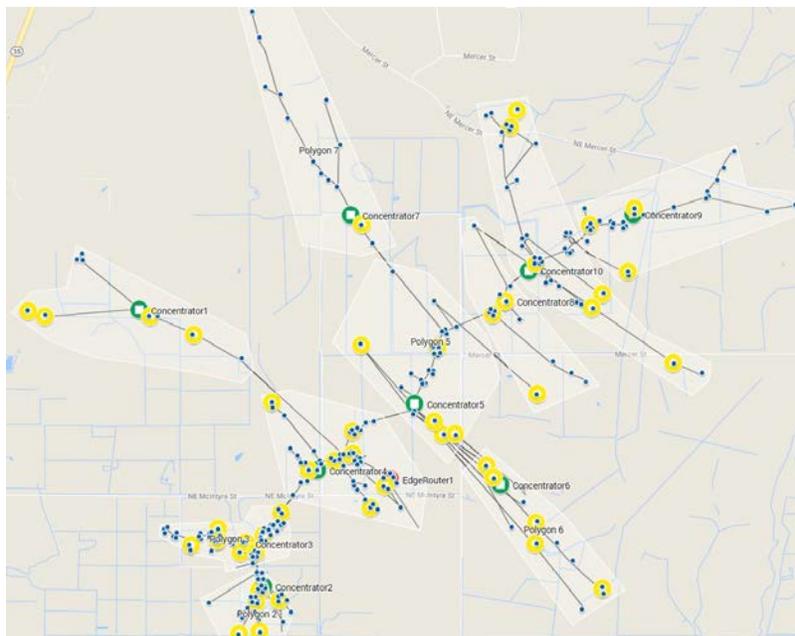
Subtask 3.2.1: Development of Different High Penetration Distributed PV Scenarios

***Objective:** Develop high-penetration distributed PV penetration scenarios. This subtask builds off Subtask 3.1.1 but significantly expands the number of PV generators to include more than 1 million total PV generators to ensure the scalability of the architecture to realistic utility-sized systems.*

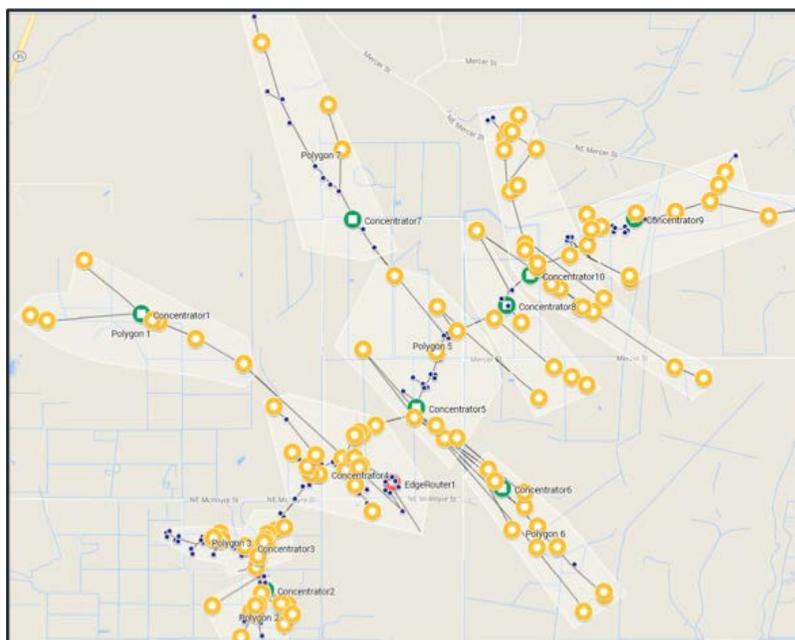
Generation of Five PV Penetration Cases for RTC-A

To evaluate the impact of different PV penetration on hybrid communications system design, we further generated five PV penetration sub-RTC-As with 10%, 20%, 40%, 60%, 80% and 100% penetration with respect to annual load. To generate these five sub-RTC-As, two steps are carried out: 1) the simulation models of Integrated Grid Modeling System (IGMS) was used to generate each PV penetration scenario, and the corresponding GridLAB-D glm files of RTC-A can be obtained. 2) In RTC-A, there are 73 commercial smart meters attached to the corresponding large-size commercial building loads, each of which could contain several office loads, and 202 residential triplex smart meters attached to the corresponding triplex nodes. Following the deployment of inverters in the GridLAB-D file, we manually constructed PV nodes for the selected commercial smart meters and chose the location of triplex node as the PV node location for the selected triplex meters on RTC-A. Thus, the communication node topologies of RTC-A with different PV penetrations were generated for the hybrid communications system design.

The communication node type and their quantity in RTC-A with different PV penetration scenarios are summarized in Table 3.3. The communication node topologies with such locations of solar PV inverters for three representative penetration scenarios of 10% (low), 40% (medium), and 100% (high) are shown in Figure 3.14. The locations of PV inverters are denoted as yellow (10%), orange (40%), and sienna (100%) circles, respectively. The edge router and data concentrators are marked as red and green circles, respectively, and the smart meters are shown as solid green dots. The 10-data concentrators indicate there are ten NANs, and the transparent white polygons show the coverage of ten individual NANs.



(10%)



(40%)

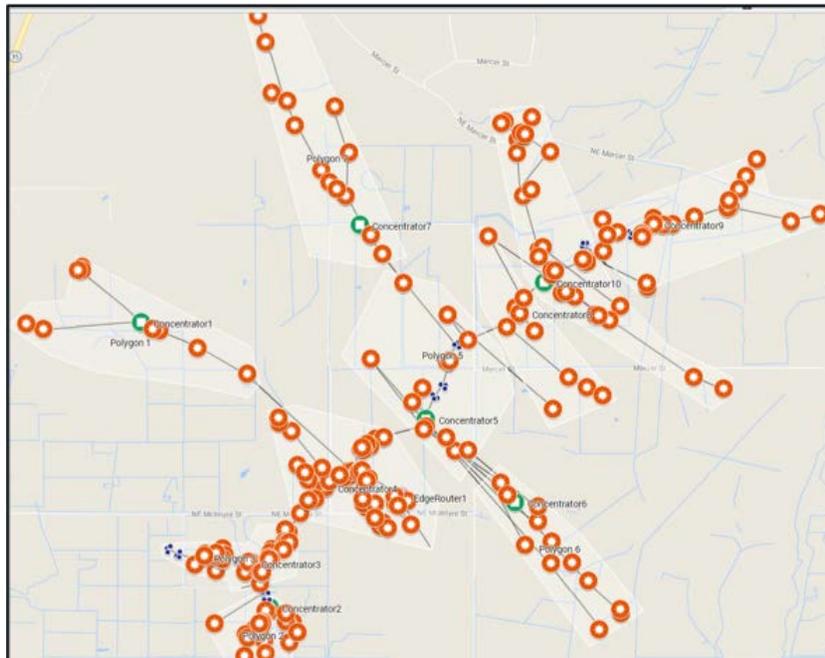


Figure 3.14. Locations PV inverters (yellow, orange, and sienna circles) for 10%, 40%, and 100% PV penetrations. The edge router, data concentrators, and smart meters are denoted as red, green, and yellow circles, respectively. The polygons show the coverage of 10 individual data concentrators.

Table 3.3. Type and Number of Communications Nodes in RTC-A

Node Type	Quantity
Edge Router	1
Data Concentrators	10
Smart Meters	275
10% of PV Inverters	51
20% of PV Inverters	72
40% of PV Inverters	117
60% of PV Inverters	160
80% of PV Inverters	179
100% of PV Inverters	259

The nine hybrid communication system designs in Table 3.4 will next be imposed on these PV penetration scenarios to study the communication design performance.

Table 3.4. Hybrid Communications Architectures

Hybrid Type	Home Area Network	Neighborhood Area Network
Hybrid 1	LoWPAN	Ethernet cable
Hybrid 2	LoWPAN	WiFi
Hybrid 3	LoWPAN	WiMAX
Hybrid 4	BPLC	Ethernet cable
Hybrid 5	BPLC	WiFi
Hybrid 6	BPLC	WiMAX
Hybrid 7	NPLC	Ethernet cable
Hybrid 8	NPLC	WiFi
Hybrid 9	NPLC	WiMAX

Results and Discussion

In this section, the simulation results of nine hybrid networks integrated with six PV penetration scenarios of RTC-A are presented and discussed. There are two main focuses: (1) validation of the hybrid network performance in terms of latency and packet loss rate with increasing PV penetration; (2) study of the capacity limitations of the hybrid designs. The performance characteristics of the hybrid designs in terms of different PV penetration are investigated through the NS-3 simulation developed above for the hybrid communication networks conveying UDP traffic. For the above first focus, all nine NS-3 hybrid network simulations on top of six PV penetration scenarios ranging from 10% to 100% are conducted with the same packet size of 2048 bytes and the same data rate of 56 Kbps. It is reasonable to consider a maximum packet size of 2048 bytes to investigate the worst-case performance limits of the network design. Regarding the second focus, the 100% penetration rate is only considered to achieve the capacity limitation with the variable data rate from 10 Kbps to 1 Gbps and the constant packet size of 2048 Bytes. The configurations of these two types of simulations are summarized in Table 3.5. In every simulation event, 10,000 packets were sent at each PV node with different data rates. The results were averaged among all paths and 100 runs.

Table 11.5. Simulation Configurations

Simulation Type	Packet Size	Data Rate	Penetration Rate
1	2048 Bytes	56 Kbps	10% - 100%
2	2048 Bytes	10 Kbps-1 Gbps	100%

Impact of PV Penetration Rates:

In this subsection, the impact of PV penetration rates on the hybrid network performance metrics of latency, packet loss rate (PLR), and throughput are examined in detail. Although the PV data packets are transmitted through a HAN and a NAN subnetwork, the performance of the HAN is unchanged and only the PV data traffic increased significantly in the NAN with the increasing PV penetration. Therefore, the performance characteristic of hybrid designs with different penetration rates critically depends on the communication technology of the NAN. Note that throughout all nine hybrid designs, the throughput of each design is always around the data rate of 56 Kbps, which satisfies the critical throughput requirement of 9.6 Kbps, because the application data rate is the benchmark value of throughput for the proposed hybrid network. The subsequent network performance analysis focuses on the metrics of latency and packet loss rate.

For the hybrid designs of LoWPAN-WiFi and LoWPAN-Ethernet, the latency and PLR results of this hybrid design are shown in Figure 3.15. For each blue box plot, the central mark indicates

the median of the latency/PLR, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The red cross indicates the lowest and highest outliers. It is observed that both latency and PLR increase in a linear fashion with the increased PV penetration. This is mainly because the ad hoc WiFi only supports 11 Mbps connection speed, less than the 100 Mbps bandwidth of Ethernet cable and 70 Mbps bandwidth of WiMAX. At 56 Kbps data rate and a packet size of 2048 bytes, the resulting PV traffic corresponding to 2.8, 3.94, 6.4, 8.8, 9.8, 14 Mbps for each of six PV penetration rates. Even the heaviest traffic of 14 Mbps is still much less than the bandwidth of Ethernet and WiMAX, thus both latency and PLR performance are constant with the increasing penetration rate, as shown in Figure 3.15(c) and (d). While the traffic increasing linearly to close the ad hoc WiFi bandwidth and even exceed it, the latency and PLR degrade correspondingly, shown in Figure 3.15(a) and (b). Additionally, the rest of the seven hybrid designs have the similar performance characteristics when compared to the LoWPAN-Ethernet design.

Table 3.6. Results of Impact of PV Penetration Rate

HAN	NAN	Max Penetration Based on Latency	Max Penetration Based on PLR
LoWPAN	Ethernet	100%	100%
	WiFi	100%	40%
	WiMAX	100%	100%
BPLC	Ethernet	100%	60%
	WiFi	100%	10%
	WiMAX	40%	20%
NPLC	Ethernet	100%	10%
	WiFi	100%	10%
	WiMAX	100%	20%

Table 3.6 shows the max allowable penetration rates of hybrid designs in terms of latency and packet loss rate, otherwise the corresponding metric can't satisfy the requirement. From Table IV, there are several interesting observations as below: 1) In terms of latency, the BPLC-WiMAX design can satisfy the latency requirement of 300 ms only when the penetration rate is less than 40%, and all of the other designs can achieve the required latency performance even the penetration rate is up to 100%. 2) Regarding packet loss rate, the NPLC-based designs have the worst performance.

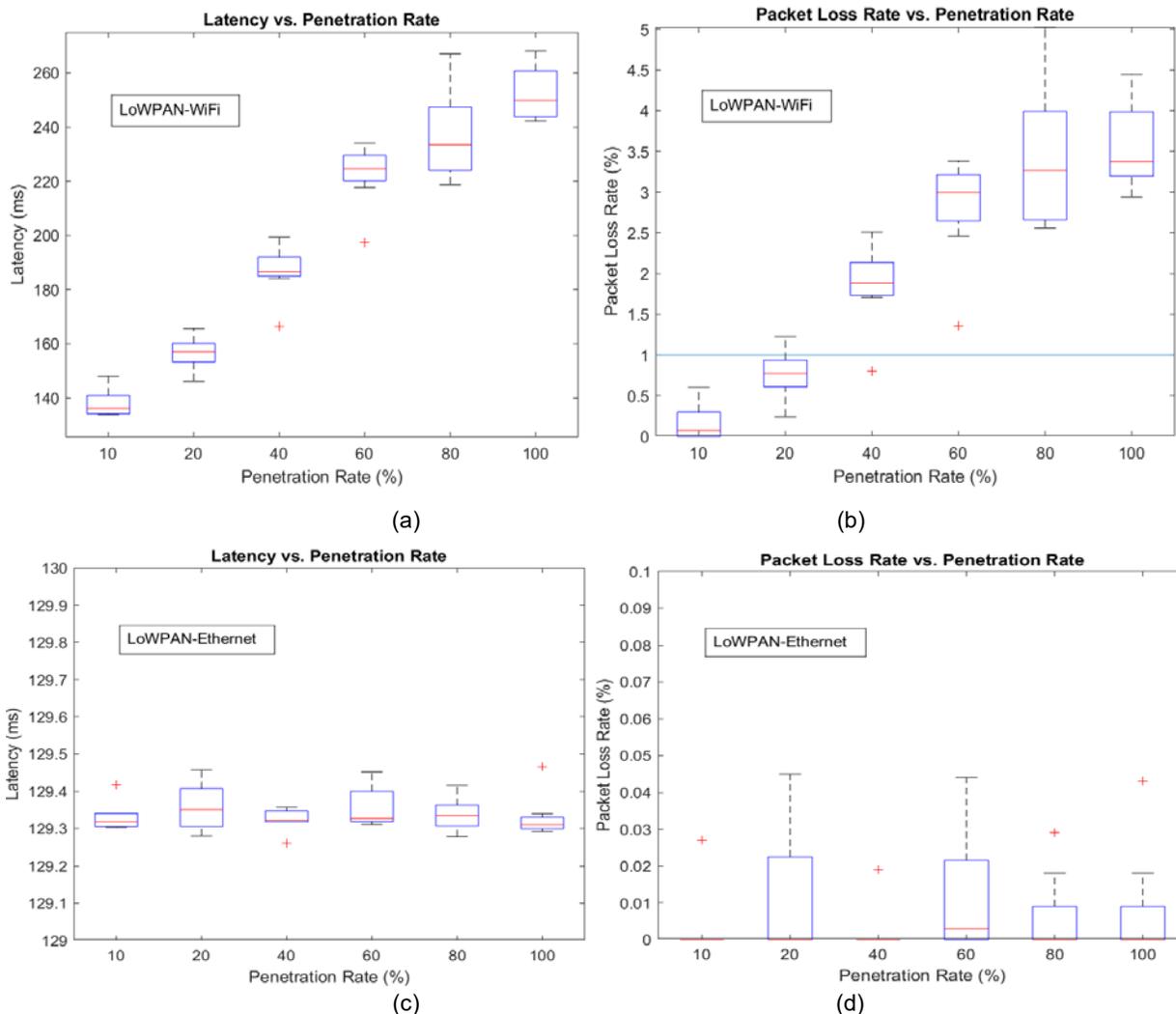


Figure 3.15. Impact of PV penetration rate for LoWPAN-WiFi and LoWPAN-Ethernet.

These results indicate whether the network design will support current smart grid communication requirements for higher levels of PV penetration soon. In addition, finding the penetration at which the QoS starts degrading will serve as a guideline of how many physical PV systems are capable of being monitored. This information dictates the limits of observability of distributed PV due to the communications system, though this can be overcome using state estimation techniques that estimate unmeasured values based on historical data and spatial correlation characteristics.

Capacity Limitation of Hybrid Designs:

Another interesting investigation is to identify the capacity limitation of each hybrid design according to the data rate. Assuming all packets are 2048 bytes in size with a 3 packet per second generation rate will result in a network data rate of 48 Kbps. Because it is envisioned that the packet generation and response rate of the smart grid communication network may increase in the future, data rates of 10 Kbps to 1 Gbps have been considered and simulated.

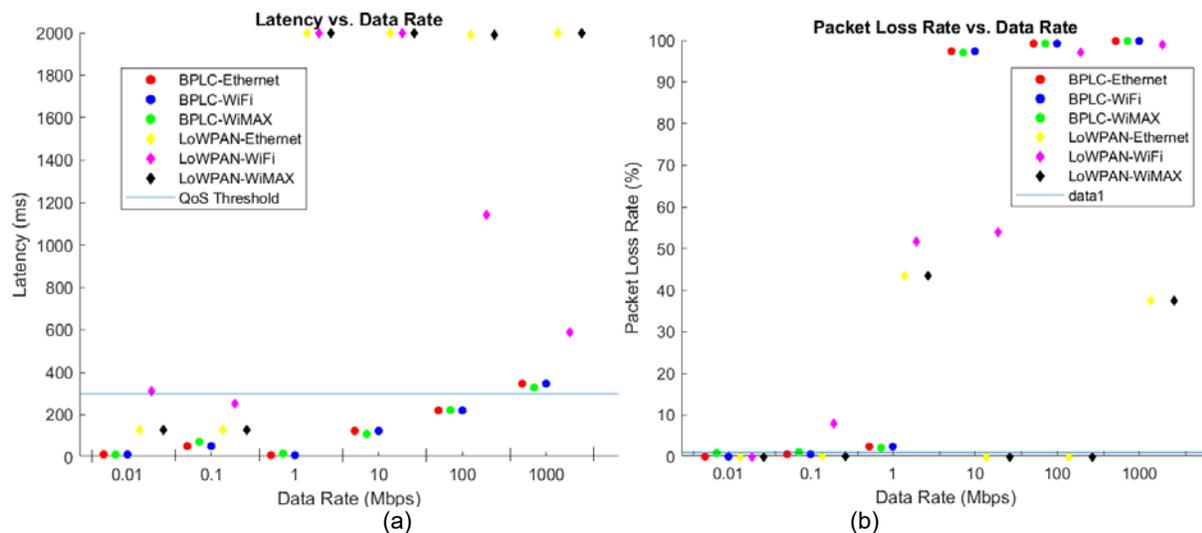


Figure 3.16. Performance at 100% PV penetration: a) latency, b) packet loss rate

The simulation results are shown in Figure 3.16. There are several interesting observations as below: (1) It is noticeable that the performance limitation of each hybrid network at higher data rates is mainly determined by the HAN link, because the HAN link has less bandwidth and the results are grouped by the HAN link type. (2) The NPLC-based designs show consistently poor performance in all penetration scenarios and hence its usage in smart grid communication networks does not appear to be promising. (3) The LoWPAN-WiFi configuration shows very poor performance with both higher penetration and data rates. This implies that even though this hybrid design would be the most economic to implement, its performance is the least scalable among all candidate hybrid designs. (4) The performance metrics are achieved for the widest range of PV penetration using Hybrids 1, 3 and Hybrid 4, except the 60% penetration scenario. All designs except Hybrid 2 can maintain the satisfied performance up to 1 Mbps. (5) The LoWPAN-based designs outperform the BPLC-based designs in the practice. However, the BPLC-based designs are preferred for the scenarios with much higher data rates.

Table 3.7. Capacity Limitation Results of Hybrid Designs

HAN	NAN	Max Tested Datarate Based on Latency	Max Tested Datarate Based on PLR
LoWPAN	Ethernet	100 Kbps	100 Kbps
	WiFi	100 Kbps	10 Kbps
	WiMAX	100 Kbps	100 Kbps
BPLC	Ethernet	100 Mbps	100 Mbps
	WiFi	100 Mbps	10 Mbps
	WiMAX	100 Mbps	100 Mbps

The simulation results in Table 3.7 indicate whether the network design can support potential future smart grid communication requirements at the 100% penetration level from the following points. The low latency for BPLC based designs at 1 Mbps shows that the design is feasible for the required data rate provided the tolerance on Packet Loss Rate may be relaxed. As an example, this may be accomplished by improving state estimation methods or algorithms which allow for improved error detection and correction. It may be seen that none of the LoWPAN-

based designs support data rates higher than 100 Kbps, if such levels of performance are expected to become necessary, the technology used for the HAN link must be of a higher bandwidth or it may use Broadband PLC. As would be expected, the throughput is limited by the bandwidth of the used technology. The theoretical limit is 250 Kbps for LoWPAN and 500 Mbps for IEEE 1901 BPLC Standard. These limits are reflected in the results. Whereas WiFi may be sufficient to meet the network requirements, the results indicate that it is the least scalable NAN link solution. Amongst Wireless solutions, WiMAX is the more scalable option. Similarly, while the LoWPAN-based designs may be sufficient to meet the network requirements, the results indicate that it is the least scalable HAN link solution. The BPLC-based designs show the greatest sensitivity to the higher packet size but is still the better option in terms of scalability. It may be concluded that the fully wireless solution, while the cheapest to implement, will only support up to 40% penetration at 56 Kbps data rate or 10 Kbps data rate with 100% penetration.

Conclusion

This study presents a new approach for studying the impact of the distributed PV penetration on the hybrid communication network design. Key findings from the simulations include: (1) due to the higher bandwidth of the NAN, the performance of Ethernet and WiMAX based designs are constant with increasing PV penetration. While the LoWPAN-WiFi design shows a linear increase of latency and packet loss rate because the PV traffic linearly scales in usage of the ad-hoc WiFi bandwidth. (2) With high data rates, the HAN link technologies determine the network design performance due to their limited bandwidth. The LoWPAN-Ethernet, LoWPAN-WiMAX, and BPLC-Ethernet almost always can satisfy the three performance requirements and accommodate a wide-range of smart grid applications and PV penetrations. Future research includes the implementation of distributed state estimation algorithms that reduce the communication requirements in these hybrid designs and their impact on overall system performance.

Subtask 3.2.2: Results and Analysis of Co-Simulation

Objective: Complete simulations and analyze results. The results of the HIL testing are compared against communications system goals and help identify areas where the architecture and algorithmic layer can be improved.

I.OPAL-RT and NS-3 Co-Simulation

(1) House and Office Models in OPAL-RT

To prepare for the co-simulation test system, we also restructured the “House” and “Office” models provided by NREL and simulate them in real time successfully in the OPAL-RT in UA research lab. The simulation runs on the OP5600 real-time simulator. We used RT-LAB software and MATLAB/Simulink R2017a to simulate the models. After restructuring the models, we import the models into separate projects in the RT-LAB software and compile the using “build” option for the model. The software generates a C-code for the simulator to use and builds a system that can run on the simulator. Afterwards, we “load” the built model that opens the Simulink model and then, via the RT-LAB, we “execute” the built model. The model starts the simulation in real time and run the models. The restructured models alongside the built projects are provided as supplementary material.

To evaluate the capabilities of the simulator, we ran both models in extended periods of time. For both the house and office models, the simulator managed to run the models in real time whereas it took a standard PC with 8GB of RAM and an Intel™ core i-5 CPU, respectively 64 seconds for the house and 14 minutes and 11 seconds for the office model. The simulation of the responsive and unresponsive load is shown in Figure 3.17. The simulation results for the responsive and unresponsive load is provided in Figure 3.18.

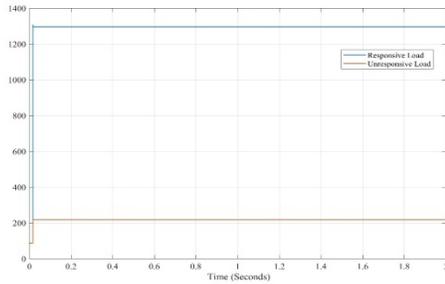


Figure 3.17. House model simulation results

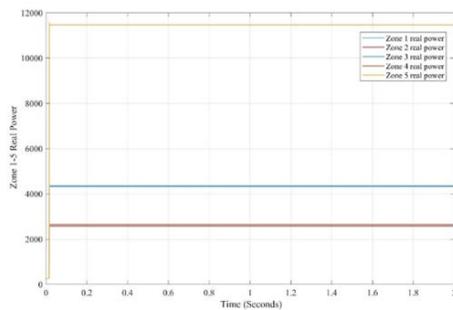


Figure 3.18. Office model simulation results

(2) OPAL-RT and NS-3 Co-Simulation

To evaluate the performance of the proposed hybrid communication infrastructure, the OPAL-RT and NS-3 co-simulation test bed is developed. Considering that MATLAB/Simulink is the essential part of the real-time simulator OPAL-RT, it is critical to successfully realize the co-simulation between the MATLAB/Simulink and NS-3 Co-simulation that is illustrated in Figure 3.19. From the MATLAB/Simulink and NS-3 co-simulation perspective, there are four fundamental steps: (1) generating and sending data packet from MATLAB/Simulink to the corresponding NS-3 nodes: the packets are generated in MATLAB/Simulink module which is connected to the corresponding nodes of the NS-3-based communication model through the *Tap Bridge Module* using *Configure Local mode*. Before sending the sequence of packets, it is necessary to format the data in UDP Datagram packet unit. Using the Port 80 which is the sending port of MATLAB/Simulink and receiving port of NS-3, the packets are then sent to the NS-3 node. (2) Receiving data packets in NS-3: using the same IP address and Port 80, the data packets sent from MATLAB/Simulink are received in the tap nodes in NS-3 and are confirmed using WireShark that the same sequence of data is received. (3) Sending data packets from NS-3 to

MATLAB/Simulink: after the packets are received, they are routed through the NS-3 model and received by the corresponding nodes which are specified in the application layer. We have used UDP socket to transmit the data in this case. From these nodes the packets are sent to MATLAB/Simulink through the sending port, Port 9090, which is the receiving port in MATLAB/Simulink. To monitor the data stream, we have exported the data using different tools such as GnuPlot and FlowMonitor. We have developed a function to extract the data packets in a string format to make the data packet more meaningful. (4) Receiving and extracting data packets in MATLAB/Simulink: after the data are sent to MATLAB/Simulink, it is confirmed that the exact data are received in MATLAB/Simulink.

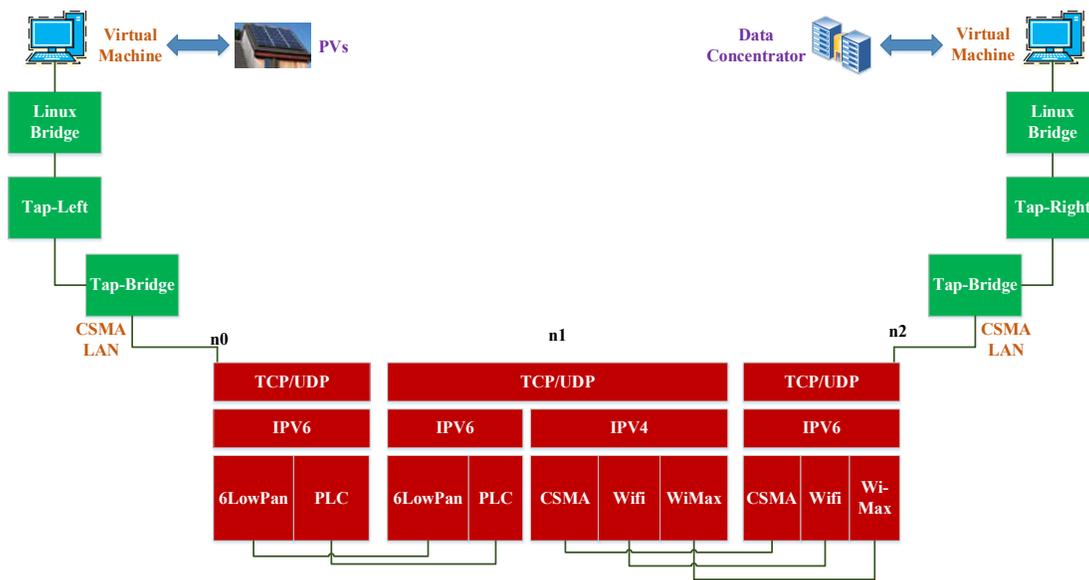


Figure 7.19. Illustration of MATLAB-NS-3 co-simulation test bed

In Figure 3.20, the end-to-end latency achieved within the NS-3 simulation is compared with that achieved by implementing NS-3-MATLAB co-simulation. In this simulation, the communication media is selected as 6LowPAN-CSMA and the data rate is 24Kbps. We would like to mention that this co-simulation set-up provides the potential of extending to the HIL test bed by using the hardware devices in place of the virtual machines.

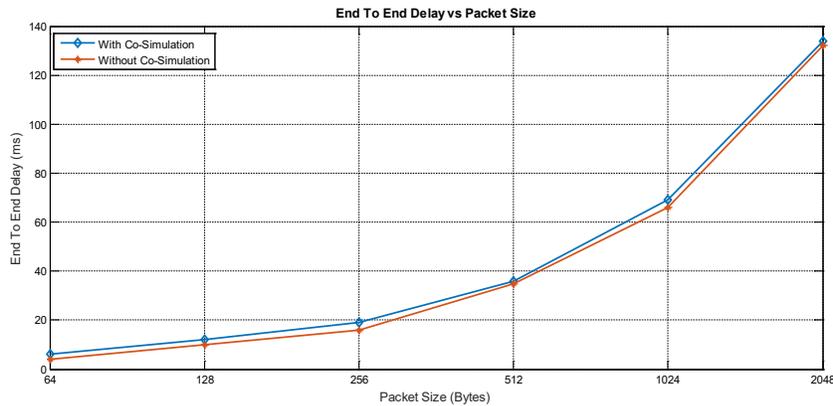


Figure 3.20. Impact of co-simulation on end-to-end latency

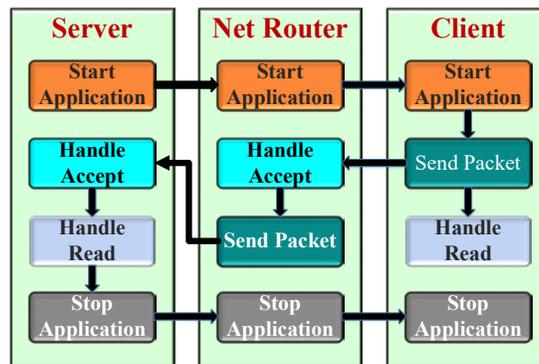


Figure 3.21. Illustration of application simulation mechanism

The detailed structure of the application layer is shown in Figure 3.21. Because we have three types of components, we have used three application containers for the communications between PV inverters and data concentrator via the smart meters. The PV inverters send the packets after the client application containers are started. Upon receiving the packets, the net router application containers, which are installed in the smart meters, are informed via the HandleAccept method. The net routers then send the packets to the corresponding server application containers that receive the packets, confirm the receptions via HandleAccept method, and trigger the HandleRead function to calculate the Quality-of-Service (QoS) information, such as end-to-end delay and throughput. This whole process is completed within the predetermined simulation time and after the specified time all the applications are stopped.

Further, considering the hardware requirement of OPAL-RT and hardware limitation of the PCs in our lab, we finalized the co-simulation experiment setup using two PCs as illustrated in Figure 3.22. This setup consists of eight main steps: (1) connecting the two computers with an Ethernet cable, (2) setting up IP address for both computers and ensuring the IP addresses belong to the same network, (3) in PC1, adding a preprocessing unit with the data generation block in the Simulink model to process the data. (For example, if the data generated is complex then we need to use the *Complex to Real-Imag* block to separate the real part from the complex data), (4) using a sinusoid block as a control input in a switch to regulate the packet generation rate, (5) using a

UDP Send block and assign an IP Address and port number, (6) in PC2, defining a Tap Node from the PV inverter nodes and using the *UseBridge mode* as the *tapbridge mode*, (7) configuring the bridge with PC2 and tap interface by following the below steps:

- `sudo brctl addbr "Bridge Name"`
- `sudo tunctl -t "TapName"`
- `sudo ifconfig mytap hw ether 00:00:00:00:00:01`
- `sudo ifconfig mytap 0.0.0.0 up`
- `sudo brctl addif "Bridge Name" "TapName"`
- `sudo ifconfig "Bridge Name" 10.1.1.1 netmask 255.255.255.0 up`

and (8) assigning an IP address to the tap interface. The averages of the throughput and latency are evaluated for the co-simulation as illustrated in Figure 2.33(a) and (b), respectively.

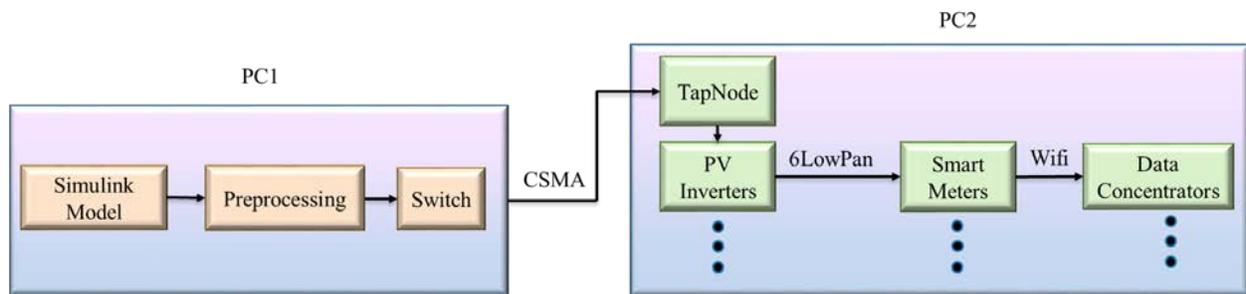


Figure 3.22. Illustration of setting up in co-simulation

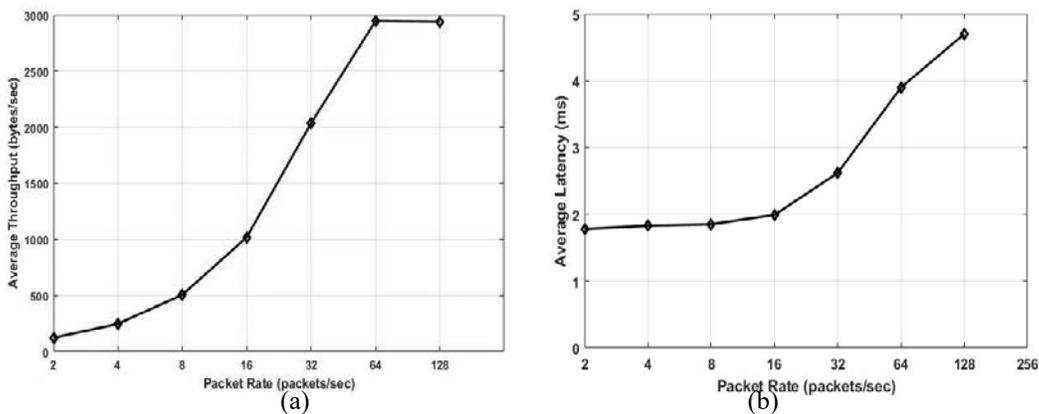


Figure 3.23. Co-simulation evaluation: (a) average throughput and (b) average latency

II. Co-Simulation Results at HELICS Platform

After the integration of NS-3 to the HELICS platform, we further conducted two use cases: 1) PV inverter monitoring through LoWPAN-WiFi and LoWPAN-Ethernet hybrid communications architecture designs; 2) Optimal demand response algorithm for the PV inverter control. These two use cases are discussed in detail as below.

Open-loop Use-case:

To investigate how hybrid communications systems design impact the DER monitoring performance, we conducted the distribution grid and communication system co-simulation on top of the RTC-A. The RTC-A with 10% PV penetration is composed of the traditional physical power system and 51 sets of solar panels and PV inverters. It is simulated in the GridLAB-D simulator, where the available active power from each solar panel and actual outputs of active power from each PV inverter are recorded and sent the HELICS platform, shown in Figure 3.24. The RTC-A’s communication network consists of 51 PV nodes, 275 smart meters and 10 data concentrators, within 51 HANs and 10 NANs, and are simulated in the NS-3. Each PV node in the NS-3 simulation receives the PV related messages from the corresponding physical PV device, sends the message to the redesigned data concentrator through its attached smart meter. In the HAN, we consider the LoWPAN technology and both WiFi mesh and Ethernet cable are implemented in the NAN, thus there are two hybrid designs, LoWPAN-WiFi and LoWPAN-Ethernet.

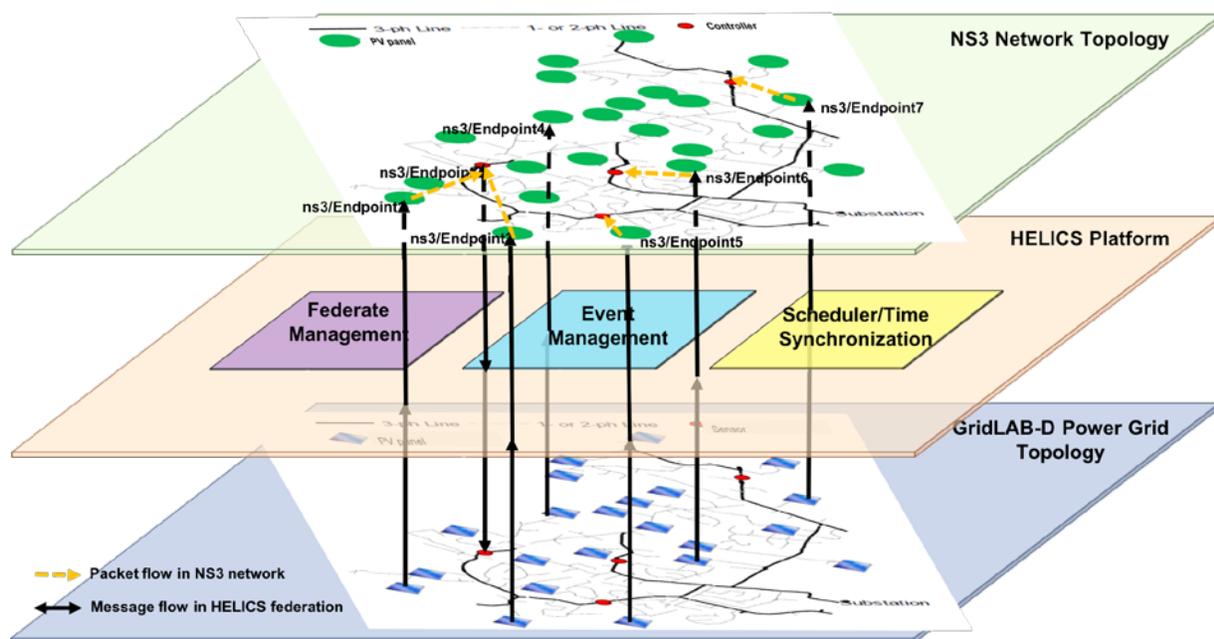


Figure 3.24. Diagram of GridLAB-D and NS-3 co-simulation through HELICS

1. Framework of GridLAB-D and NS-3 Co-simulation at HELICS

Figure 3.24 shows the framework of GridLAB-D and NS-3 co-simulation at the HELICS platform which consists of three components:

1. GridLAB-D model, it models the RTC-A feeder with 51 distributed PV panels and creates a GridLAB-D message federate. The federate topology and detailed locations of PV

panels are shown in the bottom layer of GridLAB-D topology. The physical PV meter data are formatted into the HELICS message and sent to the HELICS platform, shown as the solid black arrow between GridLAB-D topology layer and HELICS platform layer.

2. NS-3 model, it models the hybrid communication network on top of RTC-A and makes sure the communication network topology exactly maps to the physical feeder topology shown on the GridLAB-D topology layer. Similarly, it creates the NS-3 message federate through which NS-3 simulator can associate with/plug into the HELICS platform. Each PV communication nodes can fetch the corresponding PV HELICS message from the HELICS platform by using the identified end point name, shown as the solid black arrow between HELICS platform layer and NS-3 topology layer. Additionally, the HELICS message received by the PV node is converted into the normal Internet message, and then is transmitted in the simulated hybrid communication network, shown as the dash yellow arrow within the NS-3 network topology layer.
3. HELICS platform, it manages both GridLAB-D and NS-3 message federates through the federate management function and the scheduler and time synchronization functions are used to schedule the message process and exchange events and to synchronize the different simulation timing of different simulator, respectively.

2. Challenges in Software Installation

In this co-simulation implementation, three software packages of HELICS, GridLAB-D, and NS-3 are required to install by compiling with gcc, which increases the complexity of software installation. Also, both HELICS and NS-3 software packages are required to compile with enabling Python due to the Python interface of GridLAB-D software. Each installation configuration command for each software is shown in Figure 3.25.

```

➤ HELICS
  ➤ Installation configuration command
cmake -DCMAKE_INSTALL_PREFIX="/Users/jzhang1/local/helics-gcc-1.0.0" -
DBOOST_ROOT="/usr/local/Cellar/boost/1.67.0_1" -DBUILD_PYTHON_INTERFACE=ON -DPYTHON_LIBRARY=$(python3-
config --prefix)/lib/libpython3.6m.dylib -DPYTHON_INCLUDE_DIR=$(python3-config --prefix)/include/python3.6m -
DCMAKE_C_COMPILER=/usr/local/Cellar/gcc/7.3.0_1/bin/gcc-7 -DCMAKE_CXX_COMPILER=/usr/local/Cellar/gcc/
➤ NS3
  ➤ Software download: git clone -b modular-helics-application https://github.com/GMLC-TDC/ns-3-dev-git.git
  ➤ Installation configuration command
CC="gcc-7" CXX="g++-7" CFLAGS="-g -O0 -w" CXXFLAGS="-g -O0 -w -std=c++14" LDFLAGS="-g -O0 -w" .waf --
python=/Users/jzhang1/miniconda3 configure --enable-examples --enable-mpi --boost-
includes=/usr/local/Cellar/boost/1.67.0_1/include --boost-libs=/usr/local/Cellar/boost/1.67.0_1/lib --with-
helics=/Users/jzhang1/local/helics-gcc-1.0.0
➤ Gridlab-d
  ➤ Installation configuration command
./configure --prefix=/Users/jzhang1/local/gridlabd-gcc --with-helics=/Users/jzhang1/local/helics-gcc-1.0.0 --
enable-silent-rules CC="gcc-7" CXX="g++-7" CFLAGS="-g -O0 -w" CXXFLAGS="-g -O0 -w -std=c++14"
LDFLAGS="-g -O0 -w"

```

Figure 3.25. Installation configuration commands for HELICS, GridLAB-D, and NS-3

3. Configurations of GridLAB-D and NS-3 models

In the GridLAB-D simulator, there are two files we need to configure for the co-simulation. The GridLAB-D general linear model (glm) file of RTC-A should be configured as the GridLAB-D

message federate to the HELICS platform by adding the *connection* module which is used to connect the GridLAB-D simulator to the HELICS platform, and the *helics_msg* object which is used to create the GridLAB-D message federate. In the *helics_msg* object, we need to specify (1) the unique name of the GridLAB-D message federate, no default value; (2) name configuration file. Another file is the json configuration file, which is a JSON structured text file and to specify subscriptions and publications for the GridLAB-D federate. Figure 3.26 shows the detailed configuration commands in the glm file and the schema of json configuration file. Note that (1) the simulation parameter *Minimum_timestep* of both GridLAB-D and NS-3 simulators are set to 1.0 sec; (2) the specific PV meter must map correctly to the end point name of the corresponding PV nodes of NS-3 in the JSON configuration file, which further makes sure the topology of communication network completely overlaps the topology of physical distribution feeder of RTC-A.

<p>I) RTC_A_model.glm</p> <ol style="list-style-type: none"> 1. Minimum_timestep = 1.0 2. Add module connection 3. Add object helics_msg <pre> module connection; object helics_msg { name RTC_A_model; parent substation_transformer; configure RTC_A_model.json; }; </pre>	<p>II) RTC_A_model.json</p> <pre> "endpoint_publications" : { "pv_tm_R2-25-00-1_meter_70_R2-25-00-1_load_70_tm_B_1" : { "measured_real_power" : "ns3/endpoint1" }, "pv_m_R2-25-00-1_meter_59_R2-25-00-1_load_59_office_m4" : { "measured_real_power" : "ns3/endpoint2" }, "pv_tm_tpm1_R2-25-00-1_tm_1_R2-25-00-1_tn_203" : { "measured_real_power" : "ns3/endpoint51" }, "pv_tm_tpm4_R2-25-00-1_tm_64_R2-25-00-1_tn_266" : { "measured_real_power" : "ns3/endpoint52" } } </pre>
--	---

Figure 3.26. Configuration files for the GridLAB-D simulator

In the NS-3 simulator, there are also two files we need to configure for the co-simulation. In the Helics-helper source file, the NS-3 message federate is created along with setting the simulation timing step. In another import NS-3 script, we conduct the following five steps: (1) read in the topology map of communication network on top of RTC-A; (2) configure the communication technologies between nodes of the network topology; (3) configure HELICS server applications on all data concentrators; (4) configure HELICS client application and the corresponding end point object on all PV inverter nodes; (5) configure HELICS Netrouter applications on all smart meter nodes.

4. Run the GridLAB-D+NS-3 Demo

The demo is setting to run 20 seconds. Figure 3.27 shows the screenshot of executing HELICS broker, from which we can two time_request and time_grant from both GridLAB-D and NS-3 federates.

```

bin — helics_broker 2 --loglevel=3 --name=mainbroker — 110x29
..._RResults/output/CSV_data — Python  ..._RResults/output/CSV_data — -bash ...  helics_broker 2 --l...-name=mainbroker +
mainbroker(1)::time request update time_request:From (1) Time(18.000000, 18.000000, 18.000000)
mainbroker(1)::time request update time_request:From (1) Time(18.000000, 18.000000, 18.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(18.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(18.000000)
mainbroker(1)::time request update time_request:From (1) Time(18.000000, 19.000000, 18.000000)
mainbroker(1)::time request update time_request:From (1) Time(18.000000, 19.000000, 18.000000)
mainbroker(1)::time request update time_request:From (1) Time(19.000000, 19.000000, 19.000000)
mainbroker(1)::time request update time_request:From (1) Time(19.000000, 19.000000, 19.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(19.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(19.000000)
mainbroker(1)::time request update time_request:From (1) Time(19.000000, 20.000000, 19.000000)
mainbroker(1)::time request update time_request:From (1) Time(19.000000, 20.000000, 19.000000)
mainbroker(1)::time request update time_request:From (1) Time(20.000000, 20.000000, 20.000000)
mainbroker(1)::time request update time_request:From (1) Time(20.000000, 20.000000, 20.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(20.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(20.000000)
mainbroker(1)::time request update time_request:From (1) Time(20.000000, 21.000000, 20.000000)
mainbroker(1)::time request update time_request:From (1) Time(20.000000, 21.000000, 20.000000)
mainbroker(1)::time request update time_request:From (1) Time(21.000000, 21.000000, 21.000000)
mainbroker(1)::time request update time_request:From (1) Time(21.000000, 21.000000, 21.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(21.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(21.000000)
mainbroker(1)::time request update time_request:From (1) Time(21.000000, 22.000000, 21.000000)
mainbroker(1)::time request update time_request:From (1) Time(21.000000, 22.000000, 21.000000)
mainbroker(1)::time request update time_request:From (1) Time(22.000000, 22.000000, 22.000000)
mainbroker(1)::time request update time_request:From (1) Time(22.000000, 22.000000, 22.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(22.000000)
mainbroker(1)::time request update time_grant:From (1) Granted Time(22.000000)

```

Figure 3.27. Screenshot of running HELICS broker

Figure 3.28 shows the screenshot of running GridLAB-D simulator with the time is 20 seconds.

```

RTC_A — gridlabd.bin • gridlabd RTC_A_model.glm — 108x36
~/HELICS-Tutorial/test_system_data/gldFeeders/B2/RTC_A — gridlabd.bin • gridlabd RTC_A_model.glm +
jzhang1-26059s:RTC_A jzhang1$ gridlabd RTC_A_model.glm
WARNING [INIT] : Fuse:R2-25-00-1_fuse_1 has a negative or 0 mean replacement time - defaulting to 1 hour
WARNING [INIT] : last warning message was repeated 56 times
WARNING [INIT] : waterheater::init() : height and diameter were not specified, defaulting to 3.78 ft
WARNING [INIT] : last warning message was repeated 328 times
WARNING [INIT] : Node:pv_m_R2-25-00-1_meter_59_R2-25-00-1_load_59_office_m4 does not have the same nominal
voltage as its parent - copying voltage from parent.
WARNING [INIT] : Solar object:pv_R2-25-00-1_meter_59_R2-25-00-1_load_59_office_m4 is in player mode - be su
re to specify relevant values
WARNING [INIT] : Node:pv_m_R2-25-00-1_meter_68_R2-25-00-1_load_68_office_m3 does not have the same nominal
voltage as its parent - copying voltage from parent.
WARNING [INIT] : Solar object:pv_R2-25-00-1_meter_68_R2-25-00-1_load_68_office_m3 is in player mode - be su
re to specify relevant values
WARNING [INIT] : last warning message was repeated 3 times
WARNING [INIT] : Node:pv_m_R2-25-00-1_meter_51_R2-25-00-1_load_51_office_m1 does not have the same nominal
voltage as its parent - copying voltage from parent.
WARNING [INIT] : Solar object:pv_R2-25-00-1_meter_51_R2-25-00-1_load_51_office_m1 is in player mode - be su
re to specify relevant values
WARNING [INIT] : last warning message was repeated 1 times
WARNING [INIT] : Node:pv_m_R2-25-00-1_meter_56_R2-25-00-1_load_56_office_m1 does not have the same nominal
voltage as its parent - copying voltage from parent.
WARNING [INIT] : Solar object:pv_R2-25-00-1_meter_56_R2-25-00-1_load_56_office_m1 is in player mode - be su
re to specify relevant values
WARNING [INIT] : last warning message was repeated 46 times
WARNING [INIT] : group_recorder::init(): unable to open file 'csv_output/pv_meter_real_power.csv' for writi
ng
WARNING [2014-04-30 11:00:00 PDT] : last warning message was repeated 49 times
WARNING [2014-04-30 11:00:00 PDT] : transformer:R2-25-00-1_meter_36_R2-25-00-1_load_36_CTF_C_2 is at 103.8
7% of its rated power value
recorder file csv_output/substation_transformer_power.csv: No such file or directorycollector file csv_outpu
t/pv_meter_summed_real_power.csv: No such file or directorycollector file csv_output/pv_triplex_meter_summed
_reactive_power.csv: No such file or directoryrecorder file csv_output/network_node.csv: No such file or dir
ectoryrecorder file csv_output/swing_bus.csv: No such file or directorycollector file csv_output/pv_triplex_
meter_summed_real_power.csv: No such file or directorycollector file csv_output/occupancy_sum.csv: No such f
ile or directorycollector file csv_output/pv_meter_summed_reactive_power.csv: No such file or directoryProce
ssing Processing 2014-04-30 11:00:20 PDT...

```

Figure 3.28. Screenshot of running the GridLAB-D federate

Figure 3.29 shows the screenshot of running NS-3 simulator, from which we can see a lot of logs about HelicsServer received the message from the client ID something and the message context is about the real power generated by the PV panel and the communication delay.

Figure 3.29. Screenshot of running the NS-3 federate

5. Results and Analysis

We consider two hybrid designs, namely LoWPAN-WiFi and LoWPAN-Ethernet. Figure 3.30 shows that the average latency performance for both hybrid designs at each NAN satisfies the requirement of 300 ms. While Figure 3.31 shows that the packet loss rate of LoWPAN-WiFi has the worse performance in most NANs.

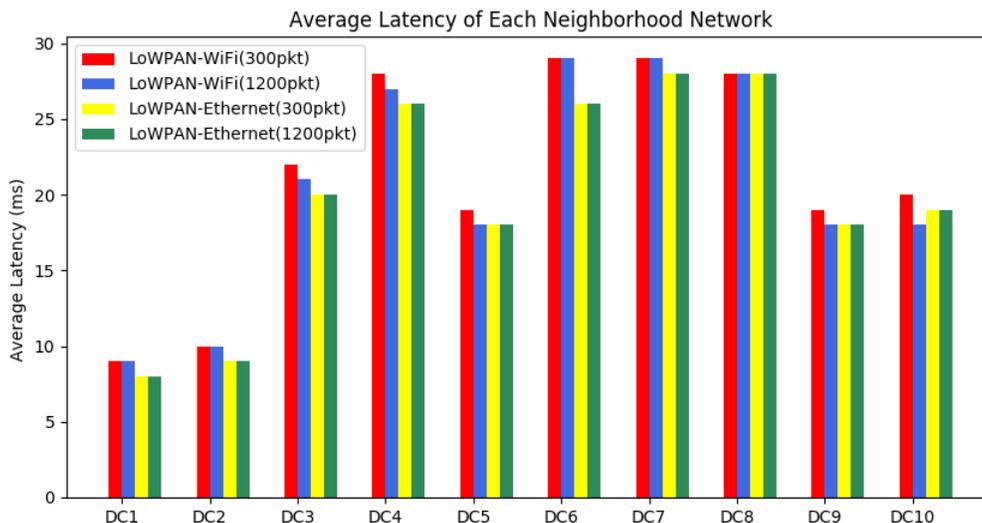


Figure 3.30. Average latency performance of each NAN at the HELICS co-simulation

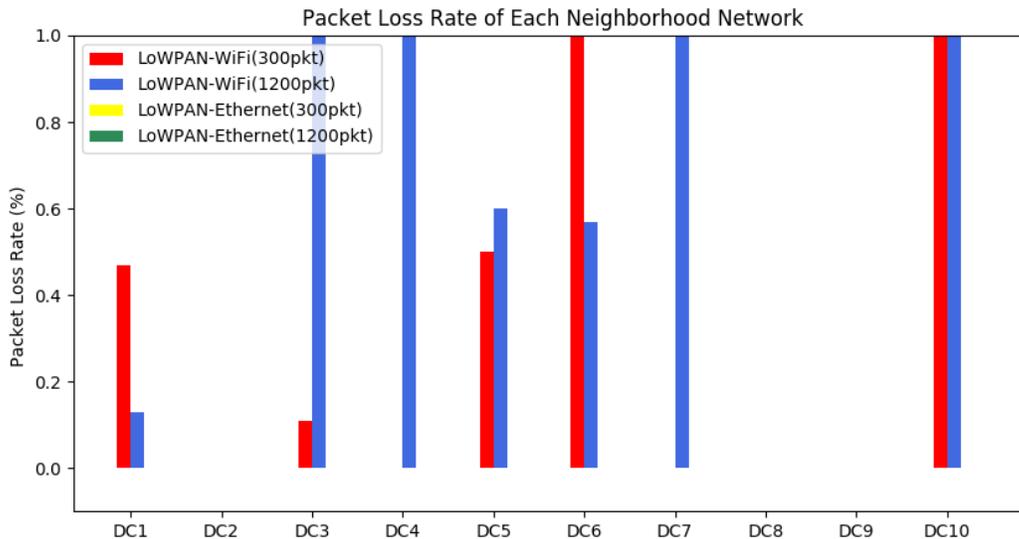


Figure 3.31. Packet loss rate performance of each NAN at the HELICS co-simulation

Closed-loop Use-case:

The closed-loop use case is to study the impact of DER control on the distribution system stability. We conducted the GridLAB-D and python-based controller co-simulation and the controller is located at the Transmission grid. The research motivation is that the stability of distribution grid with high DER penetration critically depends on the distributed DERs control including optimal demand response and voltage regulation. In this case, we conducted the optimal DER control algorithm. The control objective is to minimize the total cost and total power output which follows to the setting values. The control variables are controllable PV nodes output. The original algorithm is described as below,

$$\begin{aligned} \min_{P_i} \sum_{i=0}^N C_i(P_i) \quad & C_i(P_i) = a_i(P_i^{av} - P_i)^2 \\ \text{s.t. } 0.95 P_0 \leq P_I + \sum_{i \in N} P_i \leq 1.05 P_0 \\ & P_i \in [0, P_i^{av}] \end{aligned}$$

Then, this constraint optimization problem is reformulated into the Lagrange equation.

$$L(P_i, \lambda, \xi) = \sum_{i=0}^N C_i(P_i) + \lambda(P_I + \sum_{i \in N} P_i - 1.05 P_0) + \xi(0.95 P_0 - P_I - \sum_{i \in N} P_i)$$

After solving, we have three updates at the central controller,

$$\begin{aligned} P_i^+ &= P_i - \varepsilon_1 \frac{\partial L}{\partial P_i} = P_i - \varepsilon_1 (C_i'(P_i) + \lambda P_i - \xi P_i) \quad P_i^+ \in [0, P_i^{av}] \\ \xi^+ &= \xi + \varepsilon_2 \frac{\partial L}{\partial \xi} = \xi + \varepsilon_2 (0.95 P_0 - P_I - \sum_{i \in N} P_i) \\ \lambda^+ &= \lambda + \varepsilon_2 \frac{\partial L}{\partial \lambda} = \lambda + \varepsilon_2 \left(P_I + \sum_{i \in N} P_i - 1.05 P_0 \right) \end{aligned}$$

We use the simple two branches solar feeder, shown in Figure 3.32.

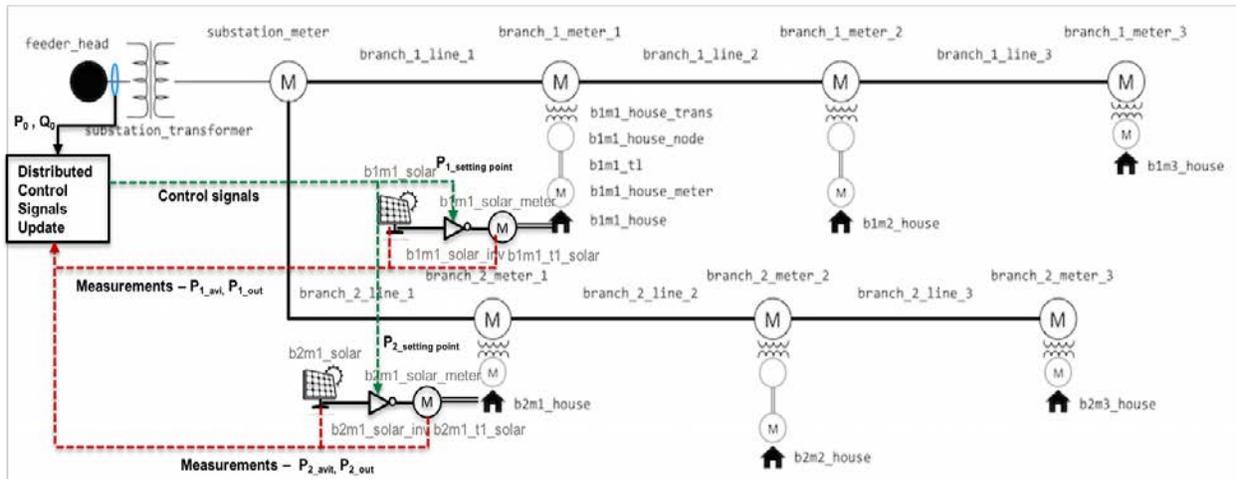


Figure 3.32. Two single solar power systems in GridLAB-D

The implementation Figure 3.33 shows that the GridLAB-D federate publishes PV meter and PV solar status messages to the HELICS. The controller federate subscribes these two messages, and run the optimal algorithm then publish the control signals to HELICS. Finally, GridLAB-D collects these control signals.

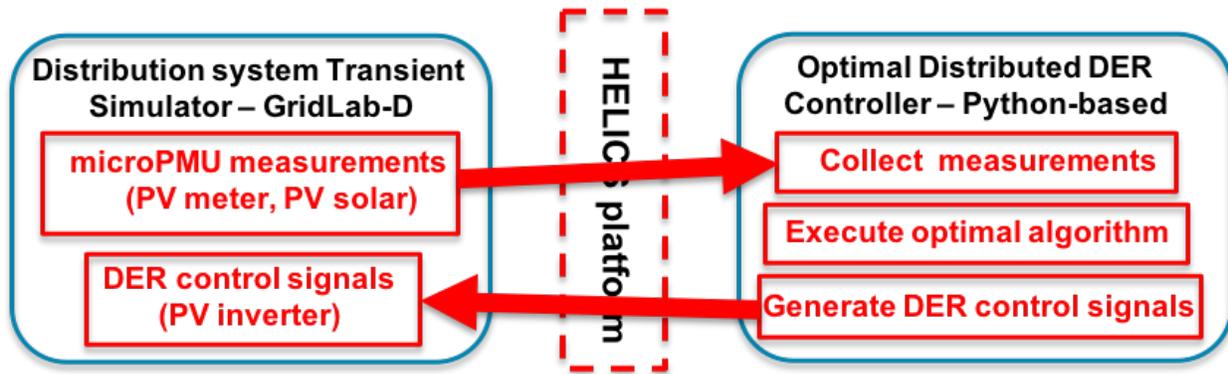


Figure 3.33 Implementation architecture of GridLAB-D + Python-based controller

The result in Figure 3.34 shows that at the beginning the algorithm oscillates. After around 25 iterations, it converges and follows the reference P_0

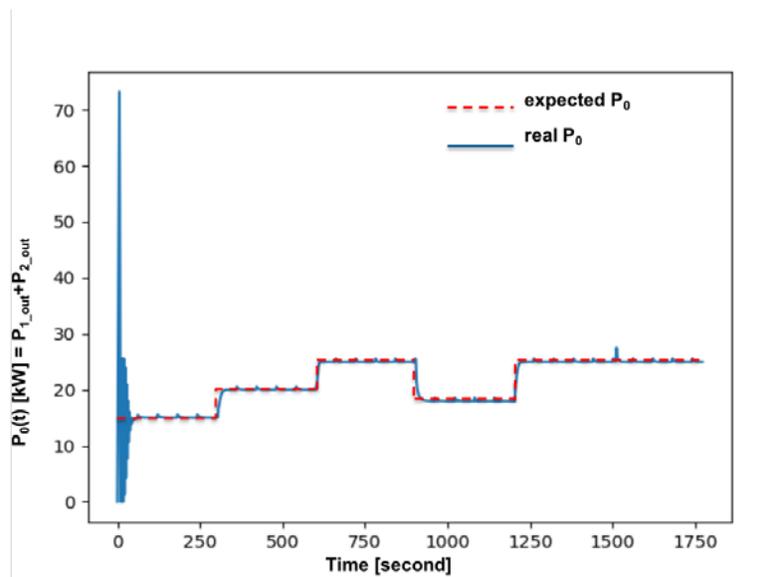


Figure 3.34. Controller results without considering communications issues

Figure 3.35 shows the implementation architecture of GridLAB-D+NS-3+Python-based-Controller. At step 1: the GridLAB-D federate publishes PV meter and PV solar status messages to the HELICS. And the NS-3 federate subscribes these two messages and they are transferred from the PV nodes to the central controller node with delay information. At step 2: the NS-3 federate published these two messages with the delay information to the HELICS, and the controller federate collects these messages and runs the control algorithm. At step 3, the controller federate finally releases the control signals to the HELICS, and the NS-3 federate also

collects these control signals at the data concentrator and then distributed to each PV node through the communication network. Each PV node publishes the control signal to the HELICS. 4) Finally, the GridLAB-D federate receives these control signals back to the corresponding PV inverter.

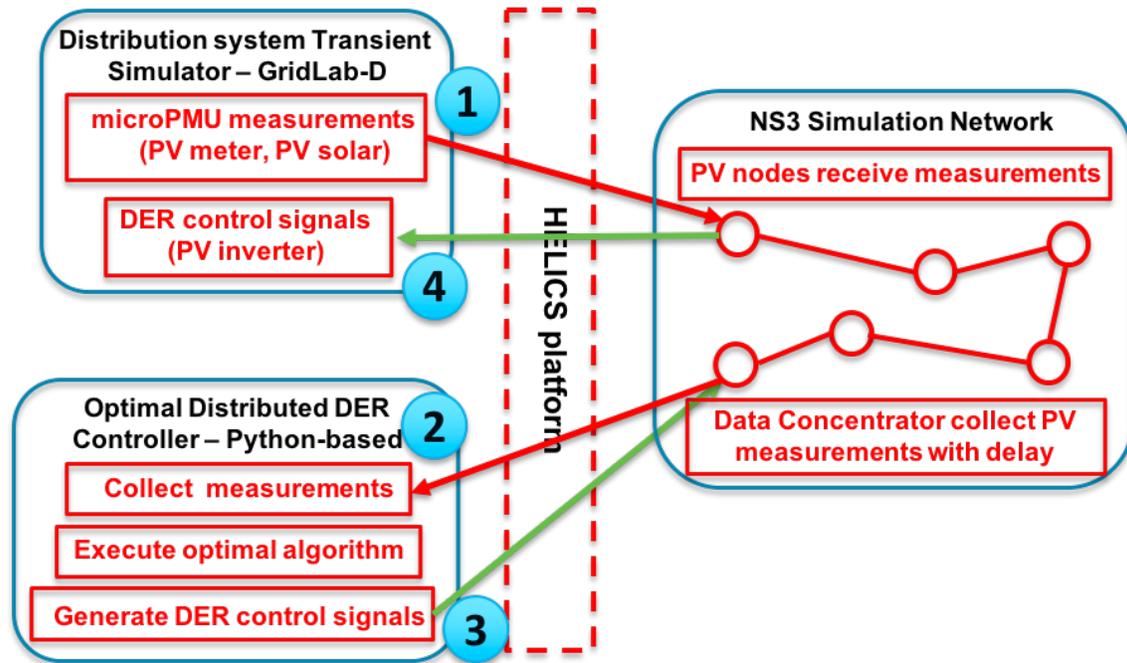


Figure 3.35. Implementation architecture of GridLAB-D+NS-3+Python-based-controller

Significant Accomplishments and Conclusions

In summary, the main project conclusions were as follows in terms of development and validation of three main tasks: (1) communications systems architecture, (2) PV system state computation, and (3) distributed state estimation. First, the innovative opportunistic and hybrid communications system architecture design framework was developed with three components. At step 1, the automatic and optimal data concentrators placement algorithm was developed to optimize the residual buffer capacity of the network while ensuring communication between smart meters and utility control centers in a multi-tier network. At step 2, a suite of hybrid communication models was developed to validate the performance of the optimal network designs from the step 1. At the same time, the distributed intelligent reinforcement learning-based attack-resilient middleware architecture is developed and validated that it can improve the performance of all hybrid designs in terms of three performance metrics. At step 3, to further identify the optimal hybrid communications network, the optimal simulation-based parameter identification framework was developed, and results show that it can identify the optimal parameter configuration ensuring all hybrid designs satisfy three performance requirements. The developed opportunistic and hybrid communications system architecture design framework can guide the utility planner to design the optimal communications infrastructure to coordinate the ever growing DERs. Second, the multi-rate and event-driven Kalman Kriging (MREDRIKK) filter was developed for distributed PV system state estimation. The proposed method will

improve estimates of solar power generated using very little data, require a few PV pilot systems to be observed, and allow assessing the expansion of residential PV systems.

Finally, based on the effective automated regionalization, the DiLISE distributed distribution system state estimation algorithm was developed and can achieve the similar estimation accuracy with the centralized algorithm with considering the number of regions. Additionally, to countermeasure the bad data, the distributed ladder-iterative belief propagation (LIBP)-based bad-data-resistant distribution system state estimation algorithm was developed and validated that it exhibits strong robustness in the presence of bad data if relatively more accurate voltage measurements are available at a small portion of buses, as low as 10%.

Inventions, Patents, Publications, and Other Results

In this project, we achieved the following inventions:

1. Optimal placement of data concentrators for expansion of distribution system communication infrastructure
2. A suite of hybrid communications network simulation models
3. A distributed intelligent middleware architecture for defending against cyberattacks
4. Optimal simulation-based parameter identification algorithm
5. HELICS and NS-3 co-simulation platform
6. Hybrid-communications-network-based Hardware-In-the-Loop test bed
7. A suite of automated regionalization algorithms for distribution systems
8. DiLISE distributed state estimation algorithm for distribution systems
9. Ladder-Iterative Belief Propagation (LIBP)-based state estimation algorithm for distribution systems, which is a bad-data-resistant distribution system state estimation algorithm.

Publications

- [1] J. Zhang, A. Hasandka, J. Wei, S M Alam, T. Elgindy, A. R. Florita, and B.-M. Hodge, "Hybrid Communication Architectures for Distributed Smart Grid Applications", *MDPI Energies*, vol. 11, no. 4, 2018.
- [2] Y. Wu, G.J. Mendis, Y. He, J. Wei, B.-M. Hodge, "An Attack-Resilient Middleware Architecture for Grid Integration of Distributed Energy Resources," Proc. IEEE International Conference on Cyber, Physical, and Social Computing, Chengdu, China, pp. 485-591, December 2016.
- [3] D. Wang, L. Yang, A. Florita, S. M. S. Alam, T. Elgindy, and B.-M. Hodge, "Automatic regionalization algorithm for distributed state estimation in power systems," in IEEE Global Conference on Signal and Information Processing (GlobalSIP), December 2016.
- [4] Y. Wu, J. Wei, B.-M. Hodge, "A Distributed Middleware Architecture for Attack-Resilient Communications in Smart Grids," Proc. IEEE International Conference on Communications, Paris, France, pp. 1-7, May 2017.
- [5] A. Hasandka et al., "Simulation-based Parameter Optimization Framework for Large-Scale Hybrid Smart Grid Communications Systems", *IEEE SmartGridComm, Aalborg, Demark, Oct. 2018*.
- [6] J. Zhang, A. Hasandka, S. Alam, A. Florita, T. Elgindy, and B. Hodge. Analysis of Hybrid Smart Grid Communication Network Designs for Distributed Energy Resources Coordination. ISGT, Washington DC, Feb. 2019.
- [7] D. Wang, L. Yang, S. M. Alam, J. Zhang, A. Florita, T. Elgindy, and B. Hodge. Distributed Ladder Iterative State Estimation and Automatic Regionalization for Distribution Systems. *submitted to IEEE Power and Energy Technology Systems Journal*, under review.
- [8] S M Alam, A. Florita, and B.-M. Hodge, "Multi-rate and Event-driven Kalman Kriging (MREDRIKK) Filter for Distributed PV System State Estimation", *IEEE Transactions on Sustainable Energy (Under 2nd Review)*.
- [9] T. Zhen, T. Elgindy, S M Alam, B.-M. Hodge, C. D. Laird, "Optimal Placement of Data Concentrators for Expansion of the Smart Grid Communications Network", under review.
- [10] I. Zaman, J. Wei, J. Zhang, A. Hasandka, S.M.S. Alam, T. Elgindy, A. Florita, B.M. Hodge, and Y. Wu, "A Distributed Intelligent Middleware Architecture for Defending against Cyber Attacks in Grid Integration of Distributed Energy Resources," to be submitted to IEEE Transactions on Industrial Informatics.
- [11] P. Fernando, B. Hossain, J. Wei, J. Zhang, A. Hasandka, S.M.S. Alam, T. Elgindy, A. Florita, B.M. Hodge, and Y. Wu, "A Reinforcement Learning-Based Attack-Resilient Middleware Architecture for Communications in Grid Integration of Distributed Energy Resources," to be submitted to IEEE Transactions on Smart Grid.
- [12] J. Zhang, A. Hasandka, J. Wei, S. Alam, A. Florita, T. Elgindy, and B. Hodge. A Hybrid Communications System Architecture Design Framework for Distributed Energy Resource Coordination. *Ready to submit to IEEE Transactions on Smart Grid*.

- [13] J. Zhang, A. Hasandka, K. Prabakar, S. Alam, A. Florita, T. Elgindy, and B. Hodge. Hardware-In-the-Loop Test of Opportunistic Hybrid Communications Systems Design and Distributed PV and Distribution System State Estimation. *in preparation*.
- [14] J. Zhang, J. Daily, R. Mast, B. Palmintier, D. Krishnamurthy, A. Florita, T. Elgindy, and B. Hodge. Development and Case Study of NS-3-HELICS High-Performance Distribution-Communication Framework. *in preparation*.

References

- [1] S. Karnouskos, "The Cooperative Internet of Things enabled Smart Grid," in *IEEE International Symposium on Consumer Electronics, ISCE*, Braunschweig, Germany, 2010.
- [2] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, *et al.*, "Internet of things strategic research roadmap," O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, *et al.*, *Internet of Things: Global Technological and Societal Trends*, vol. 1, pp. 9-52, 2011.
- [3] H. Zimmermann, "OSI reference model—The ISO model of architecture for open systems interconnection," in *Innovations in Internetworking*, C. Partridge, Ed., ed: Artech House, Inc., pp. 2-9, 1988.
- [4] J. A. P. Lopes, N. Hatziargyriou, J. Mutale, P. Djapic, and N. Jenkins, "Integrating distributed generation into electric power systems: A review of drivers, challenges and opportunities," *Electric Power Systems Research*, vol. 77, pp. 1189-1203, 7// 2007.
- [5] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "A Survey of Middleware for Internet of Things," in *Recent Trends in Wireless and Mobile Networks: Third International Conferences, WiMo 2011 and CoNeCo 2011, Ankara, Turkey, June 26-28, 2011. Proceedings*, A. Özcan, J. Zizka, and D. Nagamalai, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 288-296, 2011.
- [6] A. T. Campbell, G. Coulson, and M. E. Kounavis, "Managing complexity: Middleware explained," *IT professional*, vol. 1, pp. 22-28, 1999.
- [7] H. Gjermundrod, D. E. Bakken, C. H. Hauser, and A. Bose, "GridStat: A flexible QoS-managed data dissemination framework for the power grid," *IEEE Transactions on Power Delivery*, vol. 24, pp. 136-143, 2009.
- [8] M. Casoni, C. A. Grazia, M. Klapez, and N. Patriciello, "A congestion control middleware layer with dynamic bandwidth management for satellite communications," *International Journal of Satellite Communications and Networking*, 2015.
- [9] A. Zaballos, A. Vallejo, and J. M. Selga, "Heterogeneous communication architecture for the smart grid," *IEEE Network*, vol. 25, pp. 30-37, 2011.
- [10] W. Li and A. Monti, "Integrated simulation with VTB and OPNET for networked control and protection in power systems," in *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation*, 2010, pp. 386-391.
- [11] M. Bartl, K. Molnar, and J. Hosek, "Control of Network Operation Generator from OPNET Modeler Environment," *IJCSNS*, vol. 11, p. 17, 2011.

- [12] Y.-J. Kim, M. Thottan, V. Kolesnikov, and W. Lee, "A secure decentralized data-centric information infrastructure for smart grid," *IEEE Communications Magazine*, vol. 48, pp. 58-65, 2010.
- [13] M. G. a. V. Palanisamy, "Detection and prevention of congestion attacks and packet loss using piggyback methods in wireless network," *International Journal of Engineering Trends and Technology*, vol. 3, pp. 301-303, 2012.
- [14] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 15, p. 17, 2008.
- [15] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," *Computer Networks*, vol. 67, pp. 74-88, 2014/07/04/ 2014.
- [16] K. C. Budka, J. G. Deshpande, T. L. Doumi, M. Madden, and T. Mew, "Communication network architecture and design principles for smart grids," *Bell Lab. Tech. J.*, vol. 15, pp. 205-227, 2010.
- [17] A. Abdrabou, "A Wireless Communication Architecture for Smart Grid Distribution Networks," *IEEE Systems Journal*, vol. 10, pp. 251-261, 2016.
- [18] F. R. Yu, P. Zhang, W. Xiao, and P. Choudhury, "Communication systems for grid integration of renewable energy resources," *IEEE Network*, vol. 25, pp. 22-29, 2011.
- [19] M. A. Ahmed and Y. C. Kim, "Communication Networks of Domestic Small-Scale Renewable Energy Systems," in *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, 2013, pp. 513-518.
- [20] D. M. Falcao, F. F. Wu, and L. Murphy, "Parallel and distributed state estimation," *IEEE Transactions on Power Systems*, vol. 10, pp. 724-730, 1995.
- [21] Y. Liu, W. Jiang, S. Jin, M. Rice, and Y. Chen, "Distributing power grid state estimation on hpc clusters-a system architecture prototype," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, 2012, pp. 1467-1476.
- [22] R. Ebrahimian and R. Baldick, "State estimation distributed processing," *IEEE Trans. Power Syst*, vol. 15, pp. 1240-1246, 2000.
- [23] L. Xie, D. H. Choi, S. Kar, and H. V. Poor, "Fully Distributed State Estimation for Wide-Area Monitoring Systems," *IEEE Transactions on Smart Grid*, vol. 3, pp. 1154-1169, 2012.
- [24] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, pp. 1617-1626, 2013.
- [25] A. Minot, Y. M. Lu, and N. Li, "A distributed Gauss-Newton method for power system state estimation," 2015.
- [26] S. Yuan, P.-N. Tan, K. S. Cheruvellil, S. M. Collins, and P. A. Soranno, "Constrained spectral clustering for regionalization: Exploring the trade-off between spatial contiguity and landscape homogeneity," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1-10, 2015.
- [27] J. Zhao, G. Zhang, and M. L. Scala, "A two-stage robust power system state estimation method with unknown measurement noise," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1-5.

- [28] H. M. Merrill and F. C. Schweppe, "Bad Data Suppression in Power System Static State Estimation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, pp. 2718-2725, 1971.
- [29] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Transactions on Power Apparatus and Systems*, vol. 94, pp. 329-337, 1975.
- [30] J. Chen and A. Abur, "Placement of PMUs to Enable Bad Data Detection in State Estimation," *IEEE Transactions on Power Systems*, vol. 21, pp. 1608-1615, 2006.
- [31] L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih, and Z. Han, "Detecting False Data Injection Attacks on Power Grid by Sparse Optimization," *IEEE Transactions on Smart Grid*, vol. 5, pp. 612-621, 2014.
- [32] B. Palmintier, E. Hale, T. M. Hansen, W. Jones, D. Biagioni, H. Sorensen, *et al.*, "IGMS: An Integrated ISO-to-Appliance Scale Grid Modeling System," *IEEE Transactions on Smart Grid*, vol. 8, pp. 1525-1534, 2017.
- [33] Y. C. Kevin P. Schneider, David P. Chassin, Robert Pratt, Dave Engel, Sandra Thompson, "Modern Grid Initiative: Distribution Taxonomy Final Report " Pacific Northwest National Laboratory November 15, 2008.
- [34] J. G. D. Kenneth C. Budka, Marina Thottan, *Communication Networks for Smart Grids: Making Smart Grid Real*, 1 ed. London: Springer-Verlag London, 2014.
- [35] K. I. Lars T. Berger, *Smart Grid Applications, Communications, and Security*: Wiley, 2012.
- [36] D. M. K. Takuro Sato, Bin Duan, Martin Macuha, Zhenyu Zhou, Jun Wu, Muhammad Tariq, Solomon Abebe Asfaw, *Smart Grid Standards: Specifications, Requirements, and Technologies*: Wiley Online Library, 2015.
- [37] U. Black, *OSI: a model for computer communications standards*: Prentice-Hall, Inc., 1991.
- [38] J. Mirkovic, A. Hussain, S. Fahmy, P. Reiher, and R. K. Thomas, "Accurately Measuring Denial of Service in Simulation and Testbed Experiments," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, pp. 81-95, 2009.
- [39] M. a. A. Sengupta, A., "Oahu Solar Measurement Grid (1-Year Archive): 1-Second Solar Irradiance; Oahu, Hawaii (Data)," National Renewable Energy Laboratory DA-5500-56506, 2010.
- [40] NREL, "National Solar Radiation Database (NSRDB)," N. R. E. Laboratory, Ed., ed: National Renewable Energy Laboratory.
- [41] B. Sridhar, P. Smith, R. E. Suorsa, and B. Hussien, "Multirate and event-driven Kalman filters for helicopter flight," *IEEE Control Systems*, vol. 13, pp. 26-33, 1993.
- [42] B. Baingana, E. Dall'Anese, G. Mateos, and G. B. Giannakis, "Robust kriged Kalman filtering," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 1525-1529, 2015.
- [43] J. R. Stroud, P. Muller, and B. Sanso, "Dynamic Models for Spatiotemporal Data," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 63, pp. 673-689, 2001.
- [44] J. D. Patrick, J. L. Harvill, and C. W. Hansen, "A semiparametric spatio-temporal model for solar irradiance data," *Renewable Energy*, vol. 87, Part 1, pp. 15-30, 2016.

- [45] W. Wu, K. Wang, B. Han, G. Li, X. Jiang, and M. L. Crow, "A Versatile Probability Model of Photovoltaic Generation Using Pair Copula Construction," *IEEE Transactions on Sustainable Energy*, vol. 6, pp. 1337-1345, 2015.
- [46] C. A. Glasbey and D. J. Allcroft, "A spatiotemporal auto-regressive moving average model for solar radiation," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 57, pp. 343-355, 2008.
- [47] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395-416, 2007.
- [48] Power Systems Test Cases Archive [Online]. Available: <https://www.ee.washington.edu/research/pstca/>
- [49] M. S. Michael Dooraghi, Barry Mather, Pete Gotseff, Afshin Andreas, Chris Wright, Ryan McMorro, Jimmy Paulino, Adam Kankiewicz, "High resolution solar measurements at the DeSoto next generation solar energy center: The deployment experience," in *41st American Solar Energy Society Annual Conference (ASES 2012)*, Denver, CO, 2012, pp. 2299-2306.
- [50] (2017). *PV_LIB Toolbox*. Available: https://pvpmc.sandia.gov/applications/pv_lib-toolbox/
- [51] M. a. A. Sengupta, A., "Oahu Solar Measurement Grid (1-Year Archive): 1-Second Solar Irradiance; Oahu, Hawaii (Data)," National Renewable Energy Laboratory DA-5500-56506, 2010.
- [52] "National Solar Radiation Database (NSRDB)," N. R. E. Laboratory, Ed., ed: National Renewable Energy Laboratory.
- [53] L. Mitás and H. Mitásova, "Spatial interpolation," *Geographical information systems: principles, techniques, management and applications*, vol. 1, pp. 481-492, 1999.
- [54] A. B. McBratney and R. Webster, "Choosing functions for semi-variograms of soil properties and fitting them to sampling estimates," *Journal of Soil Science*, vol. 37, pp. 617-639, 1986.
- [55] D. G. Krige, "A Statistical Approach to some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, pp. 201-203, Mar, 1952 1952.
- [56] L. Y. Dexin Wang, Anthony Florita, S M Shafiul Alam, Tarek Elgindy, and Bri-Mathias Hodge, "Automatic Regionalization Algorithm for Distributed State Estimation in Power Systems," presented at the 4th IEEE Global Conference on Signal and Information Processing (GlobalSIP 2016), Washington D.C., USA, 2016.
- [57] A. Minot, Y. M. Lu, and N. Li, "A distributed Gauss-Newton method for power system state estimation," 2015.
- [58] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. New York, USA, 2010.
- [59] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1-122, 2011.
- [60] J. G. D. Kenneth C. Budka, Marina Thottan, *Communication Networks for Smart Grids: Making Smart Grid Real*, 1 ed. London: Springer-Verlag London, 2014.

- [61] K. I. Lars T. Berger, *Smart Grid Applications, Communications, and Security*: Wiley, 2012.
- [62] D. M. K. Takuro Sato, Bin Duan, Martin Macuha, Zhenyu Zhou, Jun Wu, Muhammad Tariq, Solomon Abebe Asfaw, *Smart Grid Standards: Specifications, Requirements, and Technologies*: Wiley Online Library, 2015.
- [63] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 15, p. 17, 2008.
- [64] F. Aalamifar, A. Schlögl, D. Harris, and L. Lampe, "Modelling power line communication using network simulator-3," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 2969-2974.
- [65] J. Farooq and T. Turletti, "An IEEE 802.16 WiMAX module for the NS-3 simulator," presented at the Proceedings of the 2nd International Conference on Simulation Tools and Techniques, Rome, Italy, 2009.
- [66] K. Andreev and P. Boyko, "Simulation Study of VoIP Performance in IEEE 802.11 Wireless Mesh Networks," in *Multiple Access Communications: Third International Workshop, MACOM 2010, Barcelona, Spain, September 13-14, 2010. Proceedings*, A. Vinel, B. Bellalta, C. Sacchi, A. Lyakhov, M. Telek, and M. Oliver, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 139-150.
- [67] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, *et al.*, "A Survey on Smart Grid Potential Applications and Communication Requirements," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 28-42, 2013.
- [68] "Communications Requirements of Smart Grid Technologies," ed: Department of Energy, 2010, pp. 1-69.
- [69] K. Razazian and J. Yazdani, "CENELEC and powerline communication specification in realization of smart grid technology," in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, 2011, pp. 1-7.
- [70] W. T. J. Bing, R. Williams, and, E. S.-Lebrilla, "SMUD-NEO Irradiance (SNI) Network Open Access Database," in *42nd ASES National Solar Conference*, Baltimore, Maryland, USA, 2013, pp. 1-5.
- [71] W. T. J. Bing, R. Williams, and, E. S.-Lebrilla, "High Penetration PV Initiative: PV Monitoring and Production Forecasting," NEO Virtus Engineering, Inc., Consultant Report2013 2013.
- [72] (2017). *Xanthus Consulting International*. Available: <http://xanthus-consulting.com/index.html>
- [73] (2017). *Smart Inverter Working Group - California Public Utilities Commission*. Available: <http://www.cpuc.ca.gov/General.aspx?id=4154>
- [74] F. Aalamifar, and L. Lampe, "Optimized WiMAX Profile Configuration for Smart Grid Communications". *IEEE Transactions on Smart Grid*, vo. 8, no. 6. pp.2723-2732, Nov., 2017.
- [75] Power Systems Test Case Archive, <https://www.ee.washington.edu/research/pstca/>
- [76] B. Hayes and M. Prodanovic, "State estimation techniques for electric power distribution systems," in European Modelling Symposium, October 2014.
- [77] Z. Zhang, M. Liu, and S. Xiao, "A survey on state estimation algorithm of distribution grid," in Spring Congress on Engineering and Technology, May 2012.

- [78] S. Elsaiah, M. Ben-Idris, and J. Mitra, "Power flow analysis of radial and weakly meshed distribution networks," in IEEE Power and Energy Society General Meeting, July 2012.
- [79] M. Karimi, H. Mokhlis, A. H. A. Bakar, and A. Shahriari, "Distribution state estimation for smart distribution system," in IEEE Grenoble Conference, June 2013.
- [80] S. M. S. Alam, B. Natarajan, and A. Pahwa, "Agent based state estimation in smart distribution grid," in IEEE Latin-America Conference on Communications, pp. 1–7, November 2013.
- [81] S. Pajic and K. Clements, "Power system state estimation via globally convergent methods," IEEE Transactions on Power Systems, vol. 20, no. 4, pp. 1683–1689, November 2005.
- [82] W. H. Kersting, Distribution System Modeling and Analysis, Second Edition (Electric Power Engineering). CRC Press, 2006.
- [83] U. Eminoglu and M. H. Hocaoglu, "Distribution systems forward/backward sweep-based power flow algorithms: a review and comparison study," Electric Power Components and Systems, vol. 37, no. 1, pp. 91–110, 2008.