UNIVERSITY OF OTTAWA

Device to Device Communications for Smart Grid

by

Kevin Shimotakahara

A thesis submitted in partial fulfillment for the
degree of Master of Applied Science - Electrical and Computer Engineering

in the

Faculty of Engineering
Department of Electrical and Computer Engineering

"I am not young enough to know everything."

Oscar Wilde

UNIVERSITY OF OTTAWA

Abstract

Faculty of Engineering
Department of Electrical and Computer Engineering

Master of Applied Science

by Kevin Shimotakahara

This thesis identifies and addresses two barriers to the adoption of Long Term Evolution (LTE) Device-to-Device (D2D) communication enabled smart grid applications in out of core network coverage regions. The first barrier is the lack of accessible simulation software for engineers to develop and test the feasibility of their D2D LTE enabled smart grid application designs. The second barrier is the lack of a distributed resource allocation algorithm for LTE D2D communications that has been tailored to the needs of smart grid applications.

A solution was proposed to the first barrier in the form of a simulator constructed in Matlab/Simulink used to simulate power systems and the underlying communication system, i.e., D2D communication protocol stack of Long Term Evolution (LTE). The simulator is built using Matlab's LTE System Toolbox, SimEvents, and Simscape Power Systems in addition to an in-house developed interface software to facilitate D2D communications in smart grid applications. To test the simulator, a simple fault location, isolation, and restoration (FLISR) application was implemented using the simulator to show that the LTE message timing is consistent with the relay signaling in the power system.

A solution was proposed to the second barrier in the form of a multi-agent Q-learning-based resource allocation algorithm that allows Long Term Evolution (LTE) enabled device-to-device (D2D) communication agents to generate orthogonal transmission schedules outside of network coverage. This algorithm reduces packet drop rates (PDR) in distributed D2D communication networks to meet the quality of service requirements of microgrid communications. The PDR and latency performance of the proposed algorithm was compared to the existing random self-allocation mechanism introduced under the Third Generation Partnership Project's LTE Release 12. The proposed algorithm outperformed the LTE algorithm for all tested scenarios, demonstrating 20-40% absolute reductions in PDR and 10-20 ms reductions in latency for all microgrid applications.

Acknowledgements

Above all I would like to thank Medhat Elsayed for the indispensable help he provided me throughout my graduate studies, including allowing me to build on the existing code he developed for MAC/Physical layer level LTE simulations. He graciously helped to integrate the code and Q-learning tuple I developed into his existing software architecture so I did not have to start from scratch, saving me tremendous amounts of time. This thesis would not have been possible without him.

Also, I express deep gratitude to my supervisors Dr. Erol-Kantarci and Dr. Hinzer for their regular support both in the direction of my research and their valuable feedback in my written works, including this thesis and my publications in IEEE Access and the GlobalSIP 2019 conference.

Finally, I thank NSERC for all their financial aid that was given to me in the form of their NSERC CGS-M scholarship in addition to the stipends I received from Dr. Hinzer's NSERC CREATE-TOPSET program.

List of Publications

K. Shimotakahara, M. Elsayed, K. Hinzer, and M. Erol-Kantarci. High-Reliability Multi-Agent Q-Learning-Based Scheduling for D2D Microgrid Communications. *IEEE Access,* 7:74412–74421, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2920662.

K. Shimotakahara, M. Elsayed, K. Hinzer, and M. Erol-Kantarci. Integrated Power and Device-to-Device (D2D) Communications Simulator for Future Power Systems. 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 2019, pp. 1-5

M. Elsayed, K. Shimotakahara, M. Erol-Kantarci, "Machine Learning-based Inter-Beam Inter-Cell Interference Mitigation in mmWave", Accepted by IEEE ICC 2020

K. Shimotakahara, M. Elsayed, K. Hinzer and M. Erol-Kantarci, "Mobile Communications-Enabled Smart Grid Co-Simulator System Design", Accepted with Revisions by IEEE Systems Journal

K. Shimotakahara, A. Surmann, R. Kohrs, K. Hinzer and M. Erol-Kantarci, "Improving Microgrid Autonomy with Reinforcement Learning Electric Vehicle (Dis)Charging Algorithms", Under review by DACH+ Energy Informatics 2020

Contents

List of Figures

List of Tables

Abbreviations

3GPP	Third Generation Partnership Project
4G LTE	Fourth Generation Long Term Evolution
5G NR	Fifth Generation New Radio
AC	Alternating Current
ACK	Acknowledgement
AMI	Advanced Metering Infrastructure
API	Application Programming Interface
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
CB	Circuit Breaker
CE	Control Element
CIM	Common Information Model
Co-SU	Co-Simulation Unit
CQI	Channel Quality Information
CT	Continuous Time
CUE	Cellular User Equipment
D2D	Device to Device
DC	Direct Current
DE	Discrete Event
DER	Distributed Energy Resource
DEVS	Discrete Event Simulation
DNN	Deep Neural Network
DR	Demand Response
DUE	D2D User Equipment
eNB	evolved NodeB

EPC	Evolved Packet Core
FLISR	Fault Location Isolation Restoration
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
FPGA	Field Programmable Gate Array
HLA	High Level Architecture
HRQ	High Reliability Q-Learning
HSS	Home Subscriber Service
ICT	Information and Communications Technology
IEC	International Electrotechnical Commission
IED	Intelligent Energy Device
IEEE	Institute of Electrical and Electronics Engineers
IEM	Intelligent Energy Management
IFM	Intelligent Fault Management
IoT	Internet of Things
ISM	Industrial Scientific and Medical
KKT	Karush Kuhn Tucker
LAN	Local Area Network
LCID	Logical Channel ID
LDC	Local Distribution Company
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
MILP	Mixed Integer Linear Programming
NAN	Neighbourhood Area Network
NASPSI	North American SynchroPhasor Initiative
NETCORE	Networked Systems and Communications Research
NOMA	Non Orthogonal Multiple Access
OFDM	Orthogonal Frequency Division Multiplexing
PDC	Phasor Data Concentrator
PDCP	Packet Data Convergence Protocol
PDR	Packet Drop Rate
PDU	Protocol Data Unit

PLC	Power Line Communication
PMU	Phasor Management Unit
PRB	Physical Resource Block
ProSe	Proximity-based Service
PSCCH	Physical Sidelink Control Channel
PSLF	Positive Sequence Load Flow
PSSCH	Physical Sidelink Shared Channel
PUSCH	Physical Uplink Shared Channel
PV	Photovoltaic
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RBG	Resource Block Group
RIV	Resource Indication Value
RLC	Radio Link Control
RTI	Run Time Interface
SA	Scheduling Access
SCI	Sidelink Control Information
SDU	Service Data Unit
SINR	Signal to Interference plus Noise Ratio
SL-SCH	Sidelink Shared Channel
SST	Solid State Transformer
SU	Simulation Unit
TBS	Transport Block Size
TCP	Transmission Control Protocol
TM	Transmission Mode
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
V2I	Vehicle to Infrasructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VoIP	Voice over IP
VTB	Virtual Test Bed

WAMPAC Wide Area Monitoring Protection And Control

WAN Wide Area Network

Dedicated to Richard Parker

Chapter 1

Introduction

1.1 Background and Motivation

Smart grid infrastructure has the potential to upgrade a fundamental pillar of human civilization by integrating Information and Communication Technology (ICT) into our power grid. ICT enables a suite of unprecedented grid services which are capable of reducing peak power consumption, optimizing power system performance, and reducing carbon emissions by facilitating higher penetration of renewable energy sources [1, 2].

To make smart grid solutions economically feasible, there is interest in implementing mobile wireless communication technology, i.e. Fourth Generation Long Term Evolution (4G LTE) and in the coming years Fifth Generation New Radio (5G NR) to establish a network of communication links over a distribution system with minimal investment in physical infrastructure [1–3]. Nevertheless, the data transfer delay (latency) in existing mobile communication technology is not guaranteed to be sufficient for latency-critical smart grid services (e.g. synchrophasor applications), and addressing this problem is an ongoing area of research [3–5].

Device-to-Device (D2D) communication is a promising means to improve communication performance at a neighbourhood area network scale by allowing direct data exchange between users, which in certain transmission modes can be controlled directly by such users [6, 7]. Moreover, such transmission modes also have the benefit of being usable even when devices are not within the coverage of an evolved NodeB cell tower, making D2D smart grid solutions a possibility in even the most remote communities.

To investigate the design performance of D2D-enabled smart grid architectures, a simulation tool capable of accurately representing the combined dynamics of power and information and communications technology (ICT) systems is highly needed. However,

as a consequence of power systems dynamics being governed by continuous time differential equations while the states of ICT systems evolve through the occurrence of discrete events, characterizing their combined dynamics in a simulation engine is a non-trivial task. There are several combined power and communication simulators in the literature [8–10], that usually take a co-simulation approach that aims to combine two mature ICT and power simulation tools with a novel software interface. Moreover, these co-simulators commonly use network simulators that have existing D2D communication solutions [11–13]. There also exists an active area of study that attempt novel ways to combine power and communication simulation engines, or create their own smart grid simulation tool from the ground up [14–23].

This thesis aims to jointly explore the research opportunities of LTE D2D-enabled smart grid communications and smart grid simulation design by creating a comprehensive simulation tool for Device-to-Device (D2D) communication-enabled power systems in Matlab/Simulink, then using it to test a novel cellular resource allocation algorithm that leverages machine learning to improve the QoS of D2D communications in out of network coverage scenarios.

1.2 Problem Statements and Proposed Solutions

This thesis is driven to solve two fundamental problems in tandem, which are presented in the following subsections. The first problem involves improving the reliability of communication for LTE compliant D2D communications so that they are suitable for smart grid communications. The second problem involves the task of developing a co-simulation tool in Matlab/Simulink in order to test the feasibility of the solution to the first problem.

1.2.1 D2D Distributed Resource Allocation Problem

Among the transmission modes of LTE D2D communication that allow users to self-allocate resources are Transmission Mode 2 (TM-2) introduced in LTE Release 12, and TM-4 introduced in LTE Release 14 [24]. TM-2 was originally designed for public safety applications and prioritized prolonging battery life [24], but requires that D2D agents self-allocate cellular resources in a random manner and thus is not currently suitable for high reliability and low latency communications. On the other hand, TM-4 was designed for high reliability and low latency communication in addition to being able to manage fast-moving devices to facilitate vehicle-to-vehicle (V2V) communications, but disregards battery life considerations [24]. As such, neither transmission mode has been designed

as an optimal smart grid communications solution. It can be argued that based on the communication requirements of smart grid applications, some combination of elements from both transmission modes would be best. It is proposed that the resource allocation mechanism of TM-2 be upgraded to meet the quality of service requirements of smart grid applications in order to have a less complex and more power-efficient solution to D2D-enabled smart grid communications than TM-4.

A critical issue with TM-2 that impedes its communication performance is how it randomly self-allocates cellular resources, which leads to high packet drop rates (PDR), making it inadequate for certain smart grid applications sensitive to information losses. For example, a typical frequency stability synchrophasor application is sensitive to message failure rates exceeding 0.33% based on data loss sensitivity metrics published by the North American Synchrophasor Institute [25].

To overcome this problem, this thesis presents my multi-agent high-reliability Q-learning (HRQ) resource allocation scheme proposed to replace the random allocation mechanism. Naturally, HRQ is powered by Q-learning, which is a decision-making algorithm that considers the situation it is in, and chooses the best action to take based on past experience. Formally, that which takes the action is referred to as the "agent", and this agent interacts with its "environment" by taking "actions". The agent's situation is considered the "state" of the environment, and there are a finite set of actions that can be taken in any given state. The best action for a given state is the one with the largest expected "reward" associated with it, which is based on an action's influence on the environment [26]. HRQ divides the LTE resource grid into orthogonal partitions of resources from which a scheduling agent can select, and the agent is rewarded or penalized depending on whether or not it takes the same scheduling action as one or more other agent(s).

1.2.2 Lack of Self-Contained Smart Grid Simulation Tool

There is little work done in developing a smart grid simulation tool solely in the Matlab/Simulink environment despite its powerful software packages for simulating the LTE physical layer, discrete event dynamics, and conventional power systems. Due to the widespread use of Simulink's Simscape Power Systems to facilitate power engineering research [27–33], it is believed that extending its functionality into the realm of LTE communications without needing software outside of the Matlab/Simulink environment would be of value to the academic community.

As such, a portion of this thesis is dedicated to formulating a solution to a comprehensive smart grid simulation tool that leverages Matlab/Simulink's LTE System Toolbox, Simscape Power Systems, and SimEvents software packages, while supplementing the

missing upper protocol layers not present in the LTE System Toolbox with our custom code. This tool was used to test the performance of the resource allocation scheme described in the previous subsection.

1.3 Contributions

The work conducted over the course of this thesis was successful in developing a functional D2D-enabled smart grid simulation tool in the Matlab/Simulink environment, complete with demonstrative demand response and fault location, isolation, and restoration smart grid applications developed to validate it. Also, the proposed HRQ algorithm proved successful in outperforming the existing LTE D2D TM-2 resource allocation algorithm for all tested network scenarios, demonstrating 20-40% absolute reductions in PDR and 10-20 ms reductions in latency for all microgrid application traffic modeled in simulation.

1.4 Organization

The remaining chapters of this thesis are organized as follows:

- **Chapter 2** provides some background information and a literature review of the work related to this thesis, and has been divided into three topics, each containing an overview and literature survey. The topics are mobile resource allocation, co-simulation design, and the general study of how D2D communications are being considered in various smart grid contexts.
- **Chapter 3** presents the smart grid simulation engine constructed, and provides a formal characterization of the ICT-power system interface that synchronizes their dynamics. A simple fault location, isolation, and restoration application is described and implemented for simulator validation purposes.
- **Chapter 4** presents the novel HRQ resource allocation algorithm. The LTE TM-2 control and data channel architecture is described to the extent necessary for understanding the HRQ strategy in addition to the existing random self-allocation algorithm of the LTE standard. Then, the HRQ and random self-allocation algorithms are presented, the network model under consideration is characterized, and the performance of the two algorithms are compared.
- **Chapter 5** concludes the thesis, and briefly describes current research being performed and its desired future directions.

Chapter 2

Literature Review and Background Information

2.1 Introduction

This chapter provides background information and a literature review for the key theory and papers that are relevant to the topic of this thesis. As this thesis aims to design a D2D-LTE enabled smart grid co-simulator and develop a new D2D cellular resource allocation algorithm, the relevant theory and research is somewhat interdisciplinary in nature, including works in simulation design, mobile D2D communications resource allocation, and the viable application space of neighbourhood area network scale smart grid applications. As such, this chapter is organized into three different topics, each with their own section. The three topics cover device to device (D2D) mobile communication resource allocation, smart grid simulation design, and leveraging D2D communications in future power systems.

2.2 Device to Device Communications and Cellular Resource Allocation

2.2.1 On Device to Device Communication

This section provides an overview of the background knowledge required to design LTE-compliant D2D resource allocation algorithms while providing some context to D2D communications. First, a summary of D2D communication is presented, which provides an overview of the potential uses for D2D communication, the various types of D2D

communication, and the actively researched topics on D2D communication. Then, a summary of how D2D data and control channels have been designed in the LTE standards is provided for both Release 12 and Release 14 D2D specifications, covering all transmission modes of D2D created by 3GPP in existence today. Finally, as one of the main research topics of this thesis is on D2D resource allocation, a small literature review on existing D2D resource allocation strategies is provided.

2.2.1.1 Background and Motivation

The fundamental idea of D2D communication is to enable direct information exchange between two devices that are within close proximity to one another without the need for any intermediary nodes. Upon writing of this thesis, Jameel *et al.* [34] (2018) is the most up to date survey on D2D communication, and provides background information as well as insight into the various research topics, open issues, and research directions of D2D communication. In addition, Asadi *et al.* [35] (2014) provides a comprehensive introduction to D2D communications.

The Types of D2D Communication: In the context of mobile communications, Asadi *et al.* identifies three different types D2D communication, namely overlay inband, underlay inband, or outband [35]. Inband D2D communication means that the D2D nodes communicate within the same frequency bands as the "regular" cellular users communicating with a base station. Outband refers to D2D communication occurring outside of the cellular network's allocated spectrum, i.e. using the unlicensed ISM spectrum. The distinction between overlay and underlay communication is that overlay D2D communication is such that the D2D nodes have dedicated licensed spectrum to operate within that regular cellular users do not use, whereas in underlay communication, both D2D and cellular users access the same licensed spectrum. Overlay D2D communication mitigates co-channel interference caused by D2D nodes, while underlay D2D communication aims to increase throughput on the cellular network via frequency reuse. Jameel *et al.* introduces two sub types to outband D2D communications, namely controlled vs. autonomous outband communication. In controlled outband communication, the base station still allocates ISM resources to the D2D nodes despite their operation in unlicensed spectrum. In autonomous outband communication, the D2D nodes self allocate the unlicensed frequency resources that they use.

Common Proposed D2D Use Cases: Jameel *et al.* provides a concise summary of the different use cases for D2D communication, namely traffic offloading, emergency services/public safety, reliable health monitoring, mobile tracking and positioning, and data dissemination [34]. Traffic offloading refers to the application of D2D communication to

reduce the data traffic that a base station must manage by "offloading" some data plane communications to D2D links where possible, i.e. when two users are exchanging information in close proximity. Emergency services refer to using D2D communications to maintain continuity of service for public safety applications under contingency scenarios where communication infrastructure has been destroyed. Contingencies aside, extension of cellular coverage can be achieved by D2D communication by enabling proximity based communications between two devices outside the coverage of a base station, or by allowing an out of coverage user send its information to a nearby in-coverage user that can then forward the information to the base station. Reliable health monitoring was proposed as another use case where devices monitoring the vitals of a patient can send their information to a central terminal accessible to medical staff, but this use case could be generalized to any IoT information system developed at the neighbourhood area network scale or lower. Mobile tracking and positioning refers to the practice of using D2D with a network of fixed position nodes that can collectively execute triangulation positioning methods to track and localize mobile users in a cellular network. Finally, data dissemination is the practice of broadcasting information to mobile users passing by the information source, which is analogous to an early twentieth century child newspaper peddler crying out the headlines on a street corner to passersby. Such a mechanism is useful for storefronts that want to inject ads into the foot traffic in front of their store, or for proximity based applications.

Since the advent of LTE Release 14, D2D communication for vehicle to vehicle communication (V2V) has become yet another commonly considered use case for D2D communication; V2V was mentioned by Jameel *et al.* as a potential 5G D2D use case, but the LTE standard introduced V2V standards before Release 15, which marks the advent of the 5G standards.

Primary D2D Research Topics: Jameel *et al.* divided the areas of D2D related research into 6 different subtopics, namely device discovery, mode selection, resource management, mobility management, security and privacy, and economic aspects [34].

Device discovery refers to the control plane signaling used to discover potential user pairs for D2D communication, and the information exchange required to determine if it is worthwhile to establish a D2D connection. The two types of device discovery mechanisms discussed in the literature are distributed and centralized device discovery. As the name implies, centralized device discovery strategies require a central node, e.g. a cellular base station, in order to function. This central node can either be partially or completely involved in the discovery process. In the complete involvement scenario, all control plane messages exchanged between D2D UEs (DUEs) pass through the central entity. In the partial involvement process, DUEs discover each other with

direct information exchange, but ultimately forward channel and SINR information to a central entity, which will respond with a final verdict as to whether or not the two DUEs can commence communication. Distributed discovery requires no centralized entity to function.

In the context of "mode selection" as defined by Jameel *et al.*, a "mode" refers to how resources are shared between DUEs and cellular UEs (CUEs). The authors define four different modes:

- **Pure Cellular Mode:** In pure cellular mode, all D2D communication is disabled.
- **Partial Cellular Mode:** In partial cellular mode, all UEs must communicate through the base station, which is the same as pure cellular mode.
- **Underlay Mode:** Underlay mode refers to allowing DUEs to reuse the frequency spectrum allocated to the CUE users.
- **Overlay Mode:** Overlay mode refers to having dedicated (separate) resources for DUEs and CUEs to eliminate co-channel interference between the two communication types.

Thus, the research activity of "mode selection" pertains to strategies that decided which mode to use for differing network scenarios.

Resource management is the D2D research topic that applies to this thesis, and is the popular topic of determining the best way to allocate cellular resources to various DUEs and CUEs. There are three types of resources that are allocated to a cellular user that are required for the transmission of data across the air interface, namely time, frequency, and transmission power. Mode selection is a key aspect of resource management, many resource allocation algorithms are only applicable to certain communication modes, e.g. underlay or overlay communications.

Network security is another research topic for D2D communications; conventional cellular users have security support from the evolved packet core (EPC), in the form of the home subscriber server (HSS) component that provides user authentication services. However, with D2D communication, it is possible for DUEs to communicate with each other without interacting with any core network infrastructure, making network security solutions formerly used not applicable in such scenarios.

D2D economic analysis is a research topic that delves into the details of information markets, and the influence that D2D communication has on them.

2.2.1.2 LTE Release 12 D2D

The Third Generation Partnership Project (3GPP) is the regulatory body that publishes the industry leading standards on mobile communication, including 4G LTE and 5G NR. Each 3GPP release adds new features to the mobile communication standards, and in Release 12, the first standards on LTE D2D communication were introduced, allowing users to communicate directly via "sidelink" channels.

Sidelink channels refer to the D2D communication channels that were specified to provide 3GPP's vision of "proximity-based services", or ProSe [36]. The service of ProSe Direct Communication was intended to allow for the support of public safety communication systems [36], allowing for a UE to send data directly to a group of other UEs in close proximity, even outside of the coverage of a base station. The physical D2D channels are embedded within the uplink physical channel when needed [36]. Unlike uplink and downlink physical channels, sidelink channels require two steps in order to prepare resource block allocations to a user. The first step is to configure "resource pools", or reserved physical resources on the uplink channel within which D2D links can be allocated resources for their data transmission, followed by the D2D link resource allocation itself. Fig. 2.1 taken from a Matlab application note [37] visualizes the concept of resource pools, and the allocation of resources within these pools. Referring to fig. 2.1, resources are allocated (light green) within Physical Sidelink Shared Channel resource pools (PSSCH, yellow) that in turn are embedded within the Physical Uplink Shared Channel (PUSCH, purple). The light and dark blue pools are sidelink synchronization signals and the Physical Sidelink Control Channel (PSCCH) respectively. The light green allocations within the PSCCH are reserved for sidelink control messages that specify which PSSCH resources the D2D transmitter will be using to send its information. For example, a device can decode a sidelink control message on the PSCCH, and then know which resource blocks and subframes to decode within the upcoming PSSCH.

These resource pools are repeated in time every "PSCCH period" (aka Scheduling Access period, or SA period) defined as a series of frames within which sidelink resource pools can be defined. Thus, every SA Period, a new resource allocation decision can be made for the entirety of the next PSSCH resource pool. In the case of this study, the dimensions of the resource pools are fixed, repeating every 40 subframes, i.e. the SA period duration. D2D links can be allocated resources within resource pools one of two ways, depending on the Transmission Mode. If the D2D link is operating in Transmission Mode 1 (TM-1), the eNB performs the resource allocation; if the D2D link is operating in Transmission Mode 2 (TM-2), the UE sending the data randomly selects resources within the resource pool to send its data with. The network model used in this thesis

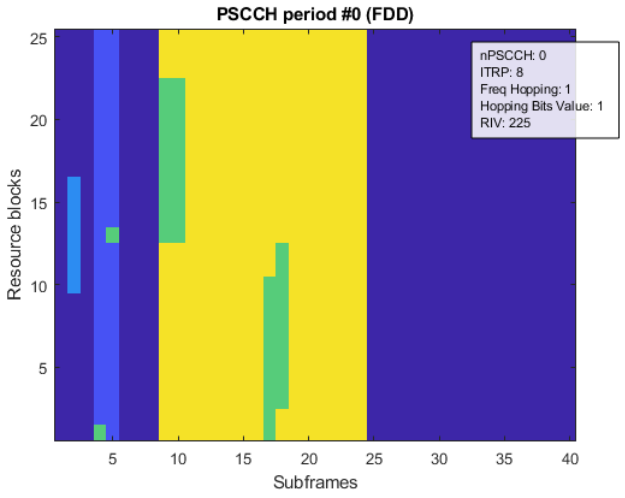

FIGURE 2.1: Visualization of resources being allocated (light green) within Physical Sidelink Shared Channel resource pools (PSSCH, yellow) that in turn are embedded within the Physical Uplink Shared Channel (PUSCH, purple). The light and dark blue pools are sidelink synchronization signals and the Physical Sidelink Control Channel (PSCCH) respectively. The light green allocations within the SA are reserved for sidelink control messages.

assumed smart grid devices are communicating to one another via D2D links operating in Transmission Mode 2, meaning that they will self-select resources.

2.2.1.3 LTE Release 14 D2D

3GPP's Release 14 of the LTE standard introduced two new transmission modes for sidelink communication, namely transmission modes 3 and 4 (TM-3 and TM-4). The work of Masegosa and Gozalvez [24] provides an excellent overview of these transmission modes. These transmission modes were designed for ultra reliable and low latency vehicle to vehicle (V2V) communication. The difference between the transmission modes lie within the resource allocation scheme as did the difference between TM-1 and TM-2 of Release 12. In TM-3, the base station controls the allocation of cellular resources for V2V links, where in TM-4, the vehicles autonomously select their own resources.

What enables the low latency communication needs for V2V communication is the restructuring of the data and control channel resource pool architecture at the physical layer. With the transmission modes introduced in Release 14, the Physical Sidelink Control and Data Channels (PSCCH and PSSCH) now coexist on the same subframes, allowing for the simultaneous transmission of a transport block and the sidelink control information (SCI) that goes with it. There are two architectures defined by the Release 14 standard to choose from, namely adjacent and non-adjacent PSSCH/PSCCH schemes. In the adjacent scheme (fig. 2.2), the PSSCH and PSCCH are interleaved so that the resources allocated for the transport block and its SCI message are always side

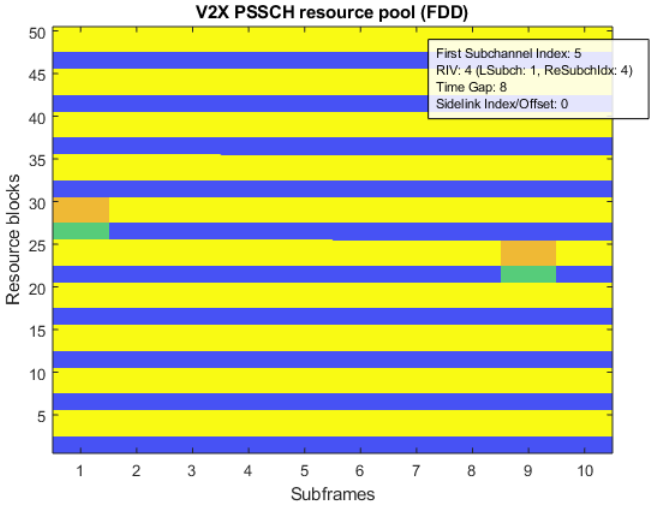

FIGURE 2.2: V2V adjacent mode visualization of resources being allocated for sidelink control information (green) and transport blocks (orange) within the PSCCH (blue) and PSSCH (yellow) respectively.

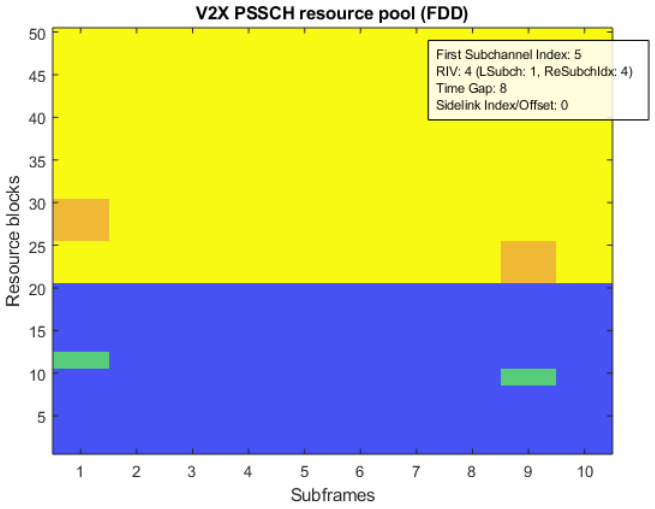

FIGURE 2.3: V2V non-adjacent mode visualization of resources being allocated for sidelink control information (green) and transport blocks (orange) within the PSCCH (blue) and PSSCH (yellow) respectively.

by side (adjacent) in the frequency domain. In the non-adjacent scheme (fig. 2.3), there are two contiguous pools of resources reserved for the PSSCH and PSCCH as in Release 12, except now the pools are separated in frequency, not time.

The reliability of V2V communication is achieved with the resource allocation scheme prescribed by the Release 14 standard. TM-3 can easily achieve high reliability due to its resource allocation being managed by an eNB. However, TM-4 required a new distributed resource allocation mechanism to foster reliable data communication when vehicles are outside of network coverage. This allocation mechanism allowed for semi-persistent scheduling of cellular resources, and involves the vehicle transmitting having

to sense the past 1000 subframes of the sidelink channel, and read SCI information from other vehicles that may have resources already reserved among the prospective resources under consideration. Additionally, a retransmission feature is still available for TM-4, but instead of being able to transmit the same transport block 4 times as in Release 12, a user can transmit the same transport block 2 times. The retransmission is scheduled within 15 subframes of the first transmission, and the resources it is transmitted on are chosen at random among the feasible allocation sets available.

2.2.1.4 NR Release 15 and 16 D2D

As of the writing of this thesis, few details are known regarding how upcoming 5G NR standards will add to existing D2D/V2V communication standards. It should be noted that V2V communications still use the same PC5 (sidelink) communication interface of the 3GPP standards as original Release 12 D2D communications. Thus, V2V is still a D2D protocol, but it is named after vehicular communications due to its intended use case. Since Release 14, 3GPP has seemed to be pushing D2D standards to make them usable in vehicular communications due to it being a highly attractive market, especially with the current development of autonomous driving. Thus, all new content on D2D is to be found in V2V, or V2X (vehicle to everything) standards. Release 15 made no fundamental additions to the V2X standard introduced in Release 14; instead, it was focused on generating the standards for the first iteration of non-stand alone 5G communications, and detailing the pivotal physical and MAC layer alterations from LTE to NR. These alterations include but are not limited to the use of mmWave spectrum, polar coding, beam forming, scalable OFDM, mini-slots, self-contained subframes, and massive MIMO [38, 39].

There is intent to add more features to V2X communication in Release 16 in mid-2020, which will allegedly facilitate low latency use cases [40]. It appears that autonomous driving is the primary use case that these features will be targeting [41]. Three 5G technologies that stand out as possible means to generating the new low-latency-enabling V2X features are scalable OFDM, mini-slots, and self-contained subframes.

Scalable OFDM allows for shorter OFDM symbols in the time domain, allowing for more symbols to be sent in the same amount of time. The trade off is that subcarrier spacing must be increased to retain their orthogonality. If Release 16 allows for variable length subframes, then this in conjunction with scalable OFDM could result in lower latency communications. Otherwise, it can be used with self-contained subframes to reduce communication latency. Self-contained subframes allow the transmitter to allocate later slots within a subframe for the receiver to respond with their own message, e.g. an

acknowledgement (ACK) message [38, 41]. Thus, two-way communication can be done in a single subframe, when before any response would have needed to wait until the next subframe at the very earliest.

To avoid having to wait for the next subframe to send or receive a transport block, Release 15 standards have introduced the idea of "mini-slots", which can send 7, 4, or 2 OFDM symbols without having to adhere to the frame structure used in existing resource allocation practices [38]. Thus, it is possible that Release 16 V2X novelties may use mini-frames to facilitate low latency communication. For example, if an urgent alert needs to be sent from one vehicle to another, a mini-slot can be used so that this information can be sent instantaneously.

To summarize, despite not knowing the specifics of what Release 16 will introduce to V2X communications, it is likely that it will make autonomous driving more feasible through enhanced V2X latency performance. Also, one can speculate that some combination of the latency-reducing techniques introduced with the advent of 5G will be leveraged in these V2X standards. The three most promising 5G technologies for this have been identified to be scalable OFDM, self-contained subframes, and mini-frames.

2.2.2 Literature Review - D2D Resource Allocation

In general, resource allocation in cellular networks is the practice of determining the spectrum, time frame, and transmission power that each user device in a cellular network may use for their communication needs. As such, frequency, time and power can be referred to as the three different resource types that must be shared among users. The goal of resource allocation depends on what the designer of the resource allocation strategy is trying to achieve, but in general it is to improve some QoS metric and/or the energy efficiency of the network. A resource allocation algorithm can be concerned with any combination of reducing interference, communication latency, power consumption, or increasing throughput. Moreover, these performance metrics are subject to a higher order tier of resource allocation strategy, namely determining how performance metrics are distributed among the users within the network. For example, if one is concerned with throughput maximization, they must also consider whether it is the overall aggregate throughput they want to increase, or would they rather increase the minimum level of performance for all users within a network. Usually both cannot be done because it is profitable from the maximum aggregate throughput perspective to neglect edge users in favour of those with higher channel capacities due to closer proximity to the base station.

Instead of sorting works by the aforementioned goals and resource types that their proposed resource allocation strategy manages (e.g. some strategies may only focus on power allocation for throughput maximization, and others joint power and frequency allocation for latency reduction, etc.), this literature review sorts the works by algorithm type, namely distributed, centralized, or hybrid in order to limit the taxonomy to three groups.

2.2.3 Distributed Resource Allocation Algorithms

Li et al. [42]: Vehicle to vehicle (V2V) communication enables two vehicles to communicate directly via mobile communication protocols without the need to relay messages through a base station. In order to take advantage of information and communication technology and develop vehicular communication services that can empower the driver with automated services, stringent QoS metrics must be realized. Such metrics, as the authors claim, are millisecond-scale latency and virtually perfect reliability [42] in order for vehicles to be able to make use of the information they receive in such a fast paced environment. This high mobility nature of vehicular communication devices makes achieving the required QoS metrics difficult, as the condition of the communication channel is changing at times faster than the mobile network can provide channel state estimations [42]. As such, there has been research in new ways to overcome the challenges in V2V communication, namely various strategies to optimally allocate cellular resources and transmission power levels to V2V links in a manner that both provides sufficient bandwidth and SINR to achieve the required QoS. Works preceding this paper either allocated resources to meet strict latency constraints using a centralized control scheme burdened with large communication overhead that only work for in-coverage scenarios, or used a distributed approach, but neglected explicit latency constraints in their optimization, only focusing on regulating SINR of V2V links [42].

The authors of this paper proposed a multi-agent, deep reinforcement learning resource and power allocation scheme to solve the V2V communication problem [42]. The V2V communication network was modeled as a set of cellular users needing vehicle to infrastructure (V2I) channels to communicate to a base station, and another set of V2V users. The V2I users were denoted as CUEs, and the V2V users were denoted DUEs [42]. The V2V communications were implemented as an underlay to the macrocell's communication channel, allowing for the chance that DUEs interfere with CUEs. For the Q-learning portion of their deep reinforcement learning scheme, the authors defined the algorithm's state, action, and reward function parameters. The state of a V2V link at a point in time t was defined as a vector of 6 different parameters, namely the link's

channel information at time t, the previous time iteration (*t-1*) interference levels observed, the channel information for the hypothetical communication link from a V2V device to the base station, the neighbouring V2V channels selected at *t-1*, the queued data, and remaining time for the link to send the queued data before latency performance is compromised [42]. The action that a V2V link takes every time iteration is simpler to describe, as it is merely the selection of cellular resources and transmitter power settings. [42]. Finally, the reward function output is proportional to the weighted sum of aggregate DUE and CUE device channel capacities, and inversely proportional to how near the latency deadline is in time [42].

Due to the large state-action space of the formulated Q-Learning model, a deep neural network (DNN) was trained to output all Q values for each state-action pair for a given state that is input into the neural network [42]. However, in order to set the weights of the DNN so they result in the output of reasonably accurate Q value estimations, the DNN first had to be trained. This training was performed by placing the "agents", i.e. DUEs and CUEs in a simulated environment that they must interact with. This simulated environment generates the next state of the agents in addition to the reward obtained for having taken each action at a given state in the simulation [42]. Deep Q learning with experience replay [42] was used while training the DNN. This experience replay functionality saved the results of the experience (state, action, resultant reward and next state) incurred from taking actions in the simulated environment, then in each iteration of the training algorithm, a "mini-batch" of experiences is sampled from memory and used to train the DNN [42].

Once the DNN was trained, the resource allocation methodology was tested in a network simulation, and its performance in terms of spectral bandwidth and probability of achieving the latency requirements as a function of vehicles in the network was compared to a pre existing method in addition to the performance of simply performing random resource allocations for the V2V links [42]. The multi-agent deep reinforcement learning method achieved data sum-rates that were consistently 15% higher both alternative approaches presented tested for all traffic congestion scenarios. Nonetheless, it should be noted that the latency performance of all methods tested rapidly deteriorated as more vehicles were added to the network.

Shih et al. [43]: The authors aim to solve the reliability issue for VoIP traffic using two different solutions, both of which involving D2D transmitters intermittently sensing for the control channel messages of other transmitting entities. The first approach, named the "explicit" approach, increases the probability of an agent being able to successfully schedule a control channel message without a conflict from a scheduling decision of another agent. The second approach, namely the "implicit" approach, proposes that

the frequency channel in which the control channel message is scheduled in be mapped one-to-one to the data channel scheduling decision; in other words, the frequency channel that a control channel message lies within "implies" the frequency channel within which its corresponding data channel transmission lies. To figure out which frequency channel to use, the implicit approach has each agent randomly decide to sense each subframe in the control channel with a probability p. If the agent decides to sense a subframe, it would take note of any frequency channels in which it detects the energy of other agents using that channel to transmit their control message, and update a "channel list" that contains whether or not a frequency channel is vacant. Otherwise, it would try to transmit a control message by consulting its channel list, choosing the same channel it did previously if it is in the list, and a random channel if not, and if no channels are vacant according to the list, the agent does not transmit, and instead senses the control channel for its entire remaining duration, keeping its list up to date. In the end, the implicit approach results in reducing the collision probability of the data messages in comparison to the scheduling mechanism prescribed by the LTE standard. In high levels of congestion, the LTE standard demonstrated near 100% collision rates, their approach kept collisions at roughly 15%. However, their algorithm achieves this performance gain in collision rate at the cost of decreased latency performance in high congestion scenarios. Both the latency and collision rate performance is still better than the LTE standard for low to moderate levels of traffic, with 5-15% reductions in collision rate, and 40-60 ms reductions in latency.

Asheralieva and Miyanaga [44]: This paper presents a means for optimizing D2D communication performance via a multi-agent Q learning approach for autonomous D2D scheduling within a multi-cell heterogeneous network that aims to achieve an optimal data throughput constrained by minimum Signal to Interference plus Noise Ratio (SINR) requirements. The authors also model this multi-agent resource allocation scheme as a stochastic non-cooperative game in order for the D2D pairs to learn the strategies of other pairs while formulating their own and achieve a Nash Equilibrium.

2.2.4 Centralized Resource Allocation Algorithms

Wen et al. [45]: This paper uses a centralized resource allocation scheme constrained by user-specific QoS requirements to maximize the data throughput of a cellular network containing D2D users and regular users. The strategy makes the decision of whether a D2D link will underlay conventional cellular channels, or use their own dedicated resources, thus incorporating mode selection into their research. Once the mode selection process is complete, the resource allocation problem is solved by two different resource

allocation algorithms, one for the allocation of resources for users using dedicated resources, and the other for the users communicating with underlay resources.

Xiao et al. [46]: The authors present a heuristic algorithm for the joint problem of allocating frequency and power resources in addition to mode selection and modulation scheme (referred to the authors as "bit allocation") selection problems of DUEs and CUEs that operate within a single cell OFDM-based mobile network. The modes are defined by the authors to either communicate via direct D2D links, or via conventional cellular communication. The DUEs are assumed to be capable of simultaneously transmitting in both modes, provided disjoint subcarriers are used for each mode. The objective function of the author's formulated resource and mode allocation problem targets the minimization of power consumption required for downlink communications. An overlay D2D communication network model is assumed, thus no frequency reuse is possible with their resource allocation strategy.

The first stage of their algorithm uses some pre-existing methodology found in the literature to solve the joint frequency, power, and bit allocation sub-problem (the authors explored the use of two different pre-existing methodologies for their algorithm, namely [47] and [48]), while keeping the transmission mode for all users fixed in conventional cellular mode. Afterwards, working within the frequency allocations provided, the second stage of the algorithm determines if power reductions can be made by allowing some (perhaps all) of the assigned subcarriers of the DUEs to operate in D2D mode.

Aside: At this point, it is essential to understand that bit allocation and power allocation are intimately related; in order to "allocate a bit" to a user, one must allocate enough power to the user so that this bit may be transmitted and received with a high enough SINR to be decoded successfully. In this work, a single bit can be added to some subcarrier by incrementing the modulation scheme, e.g. from no modulation scheme to BPSK, to move from 0 bits to 1 bit allocation, BPSK to QPSK to move from 1 bit to 2 bit allocation, and so on. Thus, depending on the subcarrier and its current bit allocation (i.e. modulation scheme), the marginal power increase that corresponds to adding 1 bit of information more to that subcarrier varies. Moreover, the distance between nodes that are communicating over the air interface influences the power required to transmit a certain number of bits as well, making mode selection influence the power consumption to bit allocation mapping as well, since a user's prospective D2D link is likely to be a different length than its conventional cellular mode link.

The second stage starts off by re-setting the power allocation and bit allocation for all DUEs to zero. Then, for each DUE, the smallest marginal increase in power consumption required to increase the bit allocation of a DUE by one bit is determined based on the available transmission modes and subcarriers. This process is repeated until the data

requirements of the DUEs are met, having bits allocated to each DUE one at a time in a manner that is always resulting in the lowest possible marginal power increase for every bit allocated. This algorithm managed to reduce power consumption by the network by 34% on average under the tested scenarios presented by the authors.

H. Wang et al. [49]: This paper aims to reduce the computational complexity of power allocation algorithms for underlay D2D networks that find allocation solutions using nonlinear convex optimization techniques that are solved via interior point method. The authors exploit the observation of the guaranteed existence of a "strong" D2D link transmission power being allocated to a D2D pair while conducting the interior point method optimization, which allows the Hessian matrix involved in the conventional process to be simplified to a diagonal matrix whose inverse is easier to compute. This "approximated" interior point method reduces the computational burden of executing the Newton-Raphson iterations used to solve the Karush-Kuhn-Tucker (KKT) conditions of the barrier problem. The new approach resulted in significant reductions in computation time while still achieving near-optimal solutions to the power allocation problem.

Mei et al. [50]: In this paper, the authors present a resource allocation algorithm that maximizes the lowest SINR of CUEs in a cellular network underlain with V2V communication links, with the latency and reliability QoS constraints of the vehicular communications taken into account. This algorithm handles the joint allocation of cellular resource blocks (i.e. combined time and frequency resource units), transmission power, and modulation and encoding scheme of the V2V links while assuming CUEs transmit at a fixed power and are provided with fixed cellular resource allocations. Moreover, the data traffic model of the vehicular communication links characterizes the data packet generation behaviour and payloads with Poisson and exponential random variables respectively.

For ease of implementing their optimization algorithm, the latency constraints of the V2V links are converted into data rate requirements, and the conversion process is mathematically justified in the paper. The authors present an intuitive but hard to solve joint optimization problem that they recast as an optimization of the Lagrange multiplier vectors of its Lagrange function, which assumes a fixed minimum CUE SINR. This problem is then broken up into smaller, more easily solvable sub-problems via Lagrange dual decomposition method. At this point, the solution to their formulated problem yields the various resource allocation decisions that result in achieving the constant minimum CUE SINR while meeting the QoS constrains of the V2V links. However, one must recall that the minimum CUE SINR was what was initially desired to be maximized, so an iterative algorithm is implemented that uses a binary search

(aka bisection search) method that closes in on the largest minimum CUE SINR that can be input into their resource allocation optimization while still resulting in a feasible solution.

The performance of their algorithm was compared to that of the SOLEN algorithm presented in [51], which is a heavily cited V2V resource allocation algorithm published in 2016. Despite yielding 4% lower average CUE SINR levels, their algorithm managed to outperform SOLEN in fairness by boosting the Jain's index score by 12% while improving the QoS performance of the V2V links, observing 67% and 7% reductions in latency and block error rate respectively.

Pan et al. [52]: This letter presents a power control and frequency allocation scheme to optimize the sum rate of underlay D2D users in a NOMA-based cellular network. The authors identify the additional complexities of D2D-enabled frequency reuse schemes when NOMA is being used by the CUEs in the network, due to the potential of the interference caused by the D2D links to alter the successive interference cancellation order of the multiplexed NOMA signals. To avoid this from happening, the authors devise additional constraints on the D2D power settings in their optimization problem formulation. The authors ultimately work to convert the optimization problem into a convex form, allowing them to use the standard dual method to find the optimal solution. This letter is an example of the active research being performed for the integration of D2D communications in mobile networks for standards even beyond 5G.

2.2.5 Hybrid Resource Allocation Algorithms

Lee et al. [53]: In another study, the end use of D2D communication was not investigated, but instead the question of how to best share resources between conventional mobile communication links (eNodeB to UE) and D2D communication links was investigated [53]. It was claimed that higher network performance can be attained through the spatial reuse of resources when implementing D2D communication alongside conventional communication [53]. However, to take advantage of such performance gains, the mechanism by which resources are allocated to various communication links must be cognizant of the intra cell interference that is generated as a consequence of spatial reuse of resources [53]. Various centralized and distributed resource allocation strategies were cited by the authors, and the trade-off between centralized and distributed strategies were explained. Centralized allocation mechanisms were good at allocating resources optimally, but introduced lots of computational overhead to the system as it requires lots of channel quality information from all UEs in the cell [53]. Distributed allocation mechanisms had lower overhead, but struggled to allocate resources effectively

[53]. The authors of this paper proposed combining such strategies by breaking down the optimization process into two stages, one centralized stage for a longer time scale allocation of resources conducted by the eNodeB, followed by shorter time scale link adaption techniques executed by the UEs [53]. In the end, the author's simulations suggested their approach to resource allocation allowed for near optimal usage of resources while reducing communication overhead and computation.

Maghsudi and Staczak [54]: The authors propose a joint power and frequency channel allocation scheme involving to maximize the aggregate throughput of cellular users while minimizing the interference impact of an arbitrary number of D2D links in a cellular network. The frequency and power allocation problem is solved in two stages, where first the frequency channels are allocated to the DUEs and UEs of the network using a centralized graph theory based approach. After the users have been "clustered" together into certain frequency channels, a distributed power allocation algorithm problem is modeled as a game. The utility functions of the players in the author's game model are unknown, but it is estimated by each player via reinforcement learning techniques that allow for the game to reach a Nash Equilibrium. This approach is not concerned with any specification of QoS constraints for the D2D communication links, as the scheduling approach proposed bases the D2D channel allocations on what benefits the data rates of cellular users the most.

Nie et al. [55]: This paper applies Q-learning in both distributed and centralized fashions in order to control underlay D2D communications in a manner that optimizes the communication system throughput while meeting the QoS constraints of regular cellular users. The Q-learning algorithms proposed aim to optimize the total sum rate of both cellular and D2D nodes by selecting appropriate power settings for devices in a single cell network. The first Q-learning scheme presented was a centralized "Team-Q learning" approach. This approach is a cooperative multi agent Q-learning scheme, where agents collectively learn a singular Q function that gets characterized over time by a common reward function. The reward function of the Team-Q learning algorithm is designed to optimize the total sum rate of all users operating on a single resource block. Thus, this algorithm is performed in parallel for each resource block in the cellular network. Alternatively, a distributed multi agent Q-learning scheme is also presented, which aims to address the scalability issues of the Team-Q learning approach. The distributed Q-learning algorithm instead has each agent rewarded based solely on its own throughput instead of the collective sum rate observed for all agents on the same resource block to allow for a smaller state-action space while eliminating the need for a centralized entity to implement a common reward function and Q-table for all agents. Results showed an improved convergence rate for the distributed Q-learning approach over the Team-Q learning approach.

As a concluding remark for this section, it is acknowledged that background information on reinforcement learning, especially Q-Learning would be desirable to include at some point in the literature review on cellular resource allocation algorithms. However, in Chapter 4, the section describing the design process for the reinforcement learning algorithm developed in this thesis naturally outlines all relevant concepts of reinforcement learning, so for the sake of efficiency, it is not repeated here.

2.3 Simulating Smart Grids

While designing the D2D-LTE enabled smart grid co-simulator presented in this thesis, the fundamentals of discrete event simulation, continuous time simulation, and co-simulation were studied. Moreover, a literature review on the already existing combined power and information and communications technology (ICT) simulation tools was conducted to identify a vacant niche in the smart grid simulator world in which the proposed simulator can fill.

Traditional power system dynamical models are generally continuous time based, whereas ICT dynamical models are discrete event based. Thus, to design and validate a simulator for a smart grid application, one must characterize and couple continuous time (power) and discrete event (ICT) dynamical systems.

This section provides the fundamental background knowledge required to formally engage in smart grid simulation design, providing an overview of basic simulation design terminology, how to characterize discrete event and continuous time dynamical models, and various strategies for coupling discrete event and continuous time simulation units. Then, a review of various smart grid simulation tools proposed in the literature is provided.

2.3.1 On Continuous Time and Discrete Event Models

The work of Gomes *et al.* [56] provides a detailed description of formal definitions used for simulation design in addition to discrete event and continuous time simulation units, and is summarized in this subsection to provide fundamental background knowledge required to appreciate, analyze, and devise smart grid simulation design strategies.

2.3.1.1 Terminology

- **Dynamical System:** A model that characterizes a real system by formulating state parameters and rules for how they change.

- **Behaviour Trace:** A behaviour trace was defined by Gomes *et al.* [56] as "the set of trajectories followed by the state (and outputs) of a dynamical system". However, in the context of this thesis, the aforementioned definition can be re-stated less generally as the collection of state variables and outputs each plotted as a function of time belonging to a dynamical system.
- **Simulated Time:** The hypothetical interval of time over which a behaviour trace is realized.
- **Wall Clock Time:** Time that passes in the real world.
- **Experimental Frame:** Sets the scope of scenarios where the dynamical system intends to be representative of reality.
- **Dynamical System Validity:** The difference between the behaviour traces produced by the dynamical system and the real system it is modeling under scenarios specified by the experimental frame.
- **Simulator:** An algorithm that computes the behaviour trace of a dynamical system.
- **Simulator Accuracy:** The difference between a real system's analytical behaviour trace and that produced by the simulator. A simulator can be deemed "accurate" should this difference be below some specified threshold value.
- **Simulation Unit (SU):** An entity that generates behaviour traces. A simulation unit can be either a simulator paired with a dynamical system, or some real thing (e.g. hardware) that is fed inputs.
- **Simulation:** A behaviour trace obtained by a simulation unit.
- **Orchestrator:** That which mediates the advancement of simulation time within and coupling between the simulation units involved in a co-simulation scenario.
- **Co-Simulation Scenario:** A rather vague construct containing that which is needed for the co-simulation to be accurate.
- **Co-Simulation Unit (Co-SU):** A simulation unit that is composed of an orchestrator and co-simulation scenario. The purpose of a Co-SU is to abstract the meshing of SUs via orchestrators and co-simulation scenarios, hiding such details in a black box model that can be treated like any other SU.
- **Co-Simulation:** A behaviour trace created by a co-simulation unit.

2.3.1.2 Discrete Event Simulation and Co-Simulation

In general, a dynamical system can be classified based on the nature of its state and its relationship with time. Specifically, a dynamical system can either have discrete or continuous states that either evolve over time continuously, or in discrete increments [57]. A discrete event dynamical system has discrete states that evolve over a continuous timescale. The dynamics of discrete event systems can be modeled with the Discrete EVent System (DEVS) specification [57]. The DEVS atomic model is the fundamental mathematical construct of the DEVS specification, which characterizes how some discrete event dynamical system changes state and interacts with its environment in terms of input, output, state, and various transition function parameters [57]:

$$DEVStuple \equiv \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle, \tag{2.1}$$

where

- X is the set of all possible inputs into the system
- Y is the set of all possible outputs out of the system
- S is the set of all possible states of the system
- δ_{int} is the internal state transition function, which updates the state of the system after the current state expires
- δ_{ext} is the external state transition function, which updates the state of the system after an input is received
- ta is the time advance function, which maps each state to how long it persists in time before it expires (at which point δ_{int} is called to update the state)

This construct was the basis for the definition of a discrete event SU presented by Gomes *et al.* [56]. Moreover, a DEVS coupled model was the basis for their definition a Co-SU required for the co-simulation of two discrete event simulations [57]:

$$coupledDEVS \equiv \langle X, Y, D, \{M_i\}, IC, EIC, EOC, select \rangle, \tag{2.2}$$

where

- X is the set of all possible inputs into the coupled system

- Y is the set of all possible outputs out of the coupled system
- D is the set of indices used to refer to each component of the coupled model
- $\{M_i\}$ is the i^{th} component of the coupled model, where a component is either another coupled model or an atomic model; note that $i \in D$
- IC is the set of input couplings, i.e. a mapping of component outputs to component inputs; perhaps a more intuitive name for IC would be "internal couplings"
- EIC is set of external input couplings, i.e. the mapping of inputs into the coupled system to inputs of components within the coupled system
- EOC is the set of external output couplings, i.e. the mapping of outputs of various components within the coupled system to the overall outputs of the coupled system
- *select* is a function used in the scenario where there are output events generated by multiple components at the same instant in time to decide the order in which the output events are processed by the system

With this mathematical machinery, combined with an orchestrator like the one presented in Algorithm 2 of [56], the co-simulation of two discrete event systems can be achieved.

2.3.1.3 Continuous Time Simulation and Co-Simulation

A continuous time dynamical system has a continuous state space, and its state evolves continuously over time. This class of dynamical systems can be characterized with a state space representation, where a set of dynamical system variables and inputs are used to formulate a system of coupled first order differential equations whose solutions can predict all salient characteristics of the dynamical system at any time following a given initial state [58]. However, a continuous time dynamical system must be a state-determined system in order to be fully characterized by a state-space model [58].

Such a system of differential equations can be expressed using vector notation:

$$\dot{\vec{x}} = f(\vec{x}, \vec{y}, t), \tag{2.3}$$

where each element x_i of $\vec{x}$ is a state variable, and each element y_k of $\vec{y}$ is an input, and t is time. $f(\bullet)$ Is a mapping of the state vector, input vector, and time to the state vector's derivative. For example, for linear time invariant system state space models, $f(\bullet) = A\vec{x} + B\vec{y}$, where A and B are constant coefficient matrices. $f(\bullet)$ can also be

broken down into a collection of functions $f_i(\bullet)$ each responsible for computing a different element $\dot{x_i}$ of $\dot{\vec{x}}$.

Ultimately, a continuous time simulator is responsible for computing the behaviour trace of the continuous time dynamical system by using its state space representation. The trajectories of each state variable x_i that constitute this behaviour trace can be approximated numerically with a n^{th} order Taylor polynomial [59]:

$$x_i(t^* + h) \doteq x_i(t^*) + f_i(t^*) \cdot h + \frac{df_i(t^*)}{dt} \cdot \frac{h^2}{2!} + ... + \frac{df_i^{(n-1)}(t^*)}{dt^{(n-1)}} \cdot \frac{h^n}{n!} \tag{2.4}$$

Where t^* is the point in time that the Taylor polynomial is centred around, and h is the time elapsed relative to t^*. The value of n depends on the desired level of precision in the estimation, and various differential equation solvers select different values for n based on how they are designed [59].

Gomes *et al.* referred to the increment h as a "micro-step" in time that a CT-SU implementing equation 2.4 would use to incrementally produce the behaviour trace of a continuous time dynamical system. However, in order to facilitate the co-simulation of CT-SUs, it was also identified that CT-SUs can only exchange information during orchestrator-enforced "macro-steps" in time due to the asynchronous micro-stepping of each CT-SU. As such, CT-SUs need a way to estimate the inputs coming from other CT-SUs for each micro-step in between macro-steps. To do this, the notion of an extrapolation function was introduced, which is used to provide such estimates at each micro-step given the last known exchange of information from the external CT-SUs. Characterization of the extrapolation function is an area of study in its own right. After this, the authors established a formal definition of a CT-SU:

$$S_i \equiv \langle X_i, U_i, Y_i, \delta_i, \lambda_i, x_i(0), \phi_{U_i} \rangle, \tag{2.5}$$

where

- X_i is the state vector space, where $\vec{x_i}(t) \in X_i$
- U_i is the input vector space, where $\vec{u_i}(t) \in U_i$
- Y_i is the output vector space
- δ_i advances the behaviour trace of the SU by a macro-step
- λ_i is the output function, generating some $\vec{y_i}(t) \in Y_i$ for every micro-step

- $x_i(0)$ is the initial state
- ϕ_{U_i} is the extrapolation function, estimating $\vec{u_i}(t)$ for every micro-step in between macro-steps

Finally, Gomes *et al.* defines a CT co-simulation scenario:

$$CTcosimScenario \equiv \langle U_{cs}, Y_{cs}, D, \{S_i : i \in D\}, L, \phi_{U_{cs}} \rangle, \tag{2.6}$$

where

- U_{cs} is the input vector space of the coupled system
- Y_{cs} is the output vector space of the coupled system
- D is the set of indices that reference the SUs that constitute the coupled system
- $\{S_i : i \in D\}$ are the SUs of the coupled system
- L is used to map output-input couplings among $\{S_i : i \in D\}$
- $\phi_{U_{cs}}$ is the set of all extrapolation functions for each SU in the coupled system

With this mathematical machinery, combined with an orchestrator like the one presented in Algorithm 3 of [56], the co-simulation of two continuous time systems can be achieved.

2.3.1.4 Hybrid Co-Simulation

Due to the fundamental differences in the characterization of DE and CT simulations, it was identified by Gomes *et al.* that the hybrid co-simulation of CT and DE SUs is difficult to characterize, and as such the prevailing co-simulation standards currently available, namely the Functional Mockup Interface (FMI) and the High Level Architecture (HLA), have limited capacity for hybrid co-simulation [56]. As such, Gomes *et al.* did not attempt to define a hybrid co-simulation scenario, as there currently is no universally "correct" way to couple DE and CT SUs. Nonetheless, the two main approaches to developing a hybrid co-simulation scenario tailored to a specific pair of CT and DE dynamical systems were identified, namely Hybrid DE and Hybrid CT approaches. These approaches are named after the type of orchestrator (CT or DE) they use in order to manage the time advances of the coupled CT and DE SUs. The orchestrator can only manage SUs of the same type, so the SU that is not the same type as the orchestrator must convert its outputs and inputs to/from CT signals and event entities

as appropriate. As such, to study co-simulation in the context of smart grid simulators, we must now depart from the generalities described in this subsection, and explore the work done in developing hybrid co-simulators that can meet the specific needs of various smart grid applications.

2.3.2 Literature Review - Smart Grid Simulators

This subsection aims to summarize the study of smart grid co-simulation design, providing an overview of co-simulation design strategies and characteristics in addition to an overview of the most prevalent smart grid co-simulators. First, key articles are introduced that collectively provide a detailed introduction to the fundamentals of smart grid co-simulation, and a snapshot of state of the art simulators and their characteristics. Then, by referring to these surveys, a concise overview of smart grid co-simulation design strategies is provided; such strategies include both time synchronization strategies as well as methods used to instantiate the orchestration mechanism of a co-simulator. Finally, the most popular smart grid co-simulations designed to date are presented, and their key characteristics are identified.

2.3.2.1 Key Co-Simulation Articles

The works of Palenski, *et al.* [60], Mets *et al.* [9], and Müller *et al.* [61] provide excellent foundational knowledge of the field of smart grid simulation, and its current state of the art.

Palenski, *et al.* recasts the core fundamentals of co-simulation design summarized mathematically by Gomes *et al.* into plain English explanations tailored to the smart grid context. The topics of co-simulation synchronization strategies, communication sequences, model/state variable coupling, error estimation and time step size control, orchestration, and simulation interfacing (referred to as simulation framework by Müller *et al.*) are covered at a high level. Moreover, this work also offers insight into the practical considerations that should be taken into account when designing a smart grid simulator.

Mets *et al.* presents a 2014 survey on smart grid simulation that aimed to inform smart grid researchers on the available simulation tools in existence while also providing smart grid simulation developers an overview of the popular co-simulation design strategies and technical standards. This survey is perhaps the most comprehensive of its kind. First, an overview of the fundamentals of smart grid (simulators aside) is provided, explaining the role of information and communications technology in power systems,

outlining the various use cases of smart grid infrastructure applicable to wide/neighbourhood/home area networks. Then, a summary of the available power and communications simulators is provided, identifying the different characteristics and use cases of each. Afterwards, the state of the art in co-simulators that combine selected power and communication simulators is presented, and excellent diagrams that depict the orchestration strategies implemented are provided for each co-simulator. Moreover, the fidelity of the co-simulators discussed is broken down into communication system fidelity and power system fidelity dimensions, and qualitatively rated and compared among themselves. Finally, the authors discuss the common simulation design "architectures" used by the simulators surveyed, in addition to the various technical standards (including communication protocols) that were used. In this paper, the notion of "architecture" is interpreted as the topological design of a co-simulator, whose components are the power system simulator, communications simulator, the power/communications simulator synchronization mechanism, and the smart grid application that operates on top of the joint power/communications system. In the same section, a brief discussion on multi-agent systems and their role in smart grid co-simulation was provided as well.

Müller *et al.* provides a more recent (2018) and concise survey of the most popular smart grid so-simulators, and includes a section on the state of the art simulation "frameworks" used by the co-simulators identified in the survey. In this paper, the notion of "framework" is defined as that which "...define(s) standardized APIs to combine different simulators or to improve their efficiency" [61], i.e. standards that facilitate interfacing simulation units. Moreover, this work has a case study section that showcases the performance of two leading smart grid co-simulators, namely GECO and INSPIRE, using them to simulate distribution system scale wide area monitoring, protection, and control applications. Such simulated applications were exposed to various communication network vexations caused by link failures and cyber attacks. With the exception of two new co-simulators, all other co-simulators identified in this survey are covered by Mets *et al.* in greater detail.

2.3.2.2 Smart Grid Co-Simulation Design Strategies and Characteristics

The various design choices that influence the overall characteristics of a smart grid co-simulator can be grouped into three categories, namely orchestration strategies, co-simulation architectures, and simulation frameworks.

Orchestration strategies specify the communication sequences and synchronization methodologies used in the simulator. Communication sequences manage the order that coupled simulation units feed each other information in order to overcome the algebraic

loop paradox. "Algebraic loops" occur when two coupled simulators both need each other's outputs from the next time step as inputs in order to generate outputs for the next time step; neither can advance without the other advancing first. There are three main communication sequencing strategies to handle this, namely parallel (aka Jacobi), serial (aka Gauss-Siedel), and iterative sequencing [60]. The notions of macro and micro time steps introduced in Section 2.3.1.3 are useful when describing these sequencing strategies. The parallel approach has both simulators advance simultaneously, exchanging their outputs that were computed by estimated inputs (based on the information exchanged previously) at every macro step. In other words, the inputs received at each macro step by a simulation unit are what should have been used to compute the output it just generated, but instead is used by an extrapolation function to estimate the inputs needed to advance through the following micro steps up to the next macro step. The serial approach limits the need for input extrapolation, having only one of the coupled simulation units extrapolate inputs to the next macro step. This is done by having one simulation unit advance to the next macro step via extrapolated inputs first in order to compute the outputs required as inputs for the other simulation unit to advance. Finally, the iterative approach is essentially a combination of serial and parallel sequencing approaches, where the two simulation units make a first pass at advancing to the next macro step via parallel sequencing, then make a second pass via serial sequencing, exchanging the first pass outputs of the next macro step.

Synchronization methodologies dictate the timing of information exchange among coupled simulation units, and three kinds of such methodologies were introduced in [60], namely point based, event driven, and master-slave. Point based synchronization sets predetermined points in time of the co-simulation at which data is exchanged, which is a simple approach, but is not accommodating to the instantaneous exchange of information generated by intermittent events occurring among simulation units, which can introduce errors in the behaviour trace. Event driven synchronization is essentially a hybrid DE co-simulation approach where every time advance of the CT simulation unit is treated as an event, at which point the hybrid co-simulation can be treated as a discrete event co-simulation, making the formulation of the co-simulation scenario easily expressible with the DEVS formalism. Such a synchronization approach allows for tight coupling of the simulation units, but can experience co-simulation scalability issues should the time step of the CT simulation unit solver be sufficiently small. Finally, master-slave synchronization involves having a dominant simulation unit in charge of dispatching the temporal advances and data exchanges of subordinate simulation units. The main drawback to this approach is that the slave simulation units cannot initiate any data synchronization events, and must wait to be queried by the master to share new information. In addition to these synchronization methodologies, there is the alternative of performing

a real time co-simulation, where each simulation unit is "synchronized" to wall clock time, making the coordination of simulation time advancement among simulation units for proper information exchange a non-issue. The main challenge with implementing real time co-simulation is that the computational speed of the simulators must be able to be fast enough to produce their simulation traces in real time. However, if this can be achieved, a bonus to this synchronization strategy is that it makes hardware-in-the-loop simulations feasible, where various components of the simulation can be swapped out with real hardware and/or hardware emulators for even more realistic testing.

Co-simulation architecture is introduced by Mets *et al.* as the organizational layout of the four main "high level functional components" of a smart grid co-simulator, namely the power system, network, control, and "synch" components [9]. Naturally, the power system and network components are the power and communications simulation units respectively. The control component is what simulates the smart grid application manipulating the smart grid dynamics. The synch component is essentially the orchestrator of a co-simulation unit if thought of in terms of Gomes *et al.*. Mets *et al.* identifies three main smart grid co-simulation architectures upon review of the simulations they surveyed, namely master-slave, dedicated synchronization and control, and layered architectures. In a master-slave architecture, the synch and control components are implemented within either the power system or network components, allowing for the application of a master-slave synchronization methodology. The dedicated synchronization and control architecture keeps all components separate, with the synch component being central to the other three in a star topology. This way, the synch component is centralized and independent, and coordinating the interactions between the control, network, and power system components. Such an architecture could support any synchronization methodology previously mentioned. Finally, the layered simulation architecture is similar to the dedicated synchronization and control architecture, except that there is a dedicated synch component between every simulation component (be it control, power system, or network) being coupled. It should be noted that the ideas of co-simulation architectures and synchronization methodologies overlap; however, co-simulation architectures are aid in co-simulation design at a more macroscopic level, i.e. the organization of the aforementioned high level components. The reason for the overlap is that the architecture can be influenced by the desired synchronization methodology implemented.

Simulation frameworks were introduced by Müller *et al.* as application programming interfaces (APIs) used to streamline simulation coupling in co-simulation development. Three standardized simulation frameworks were identified, namely High Level Architecture (HLA), Functional Mockup Interface (FMI), and Mosaik, in addition to a class of

non-standard "ad-hoc" simulation frameworks that refer to those custom-built by smart grid co-simulation developers to meet their specific needs.

HLA is specified by the IEEE 1516 standard, and has been traditionally used to facilitate DE co-simulation, and details interfacing protocols required to implement a dedicated synchronization architecture, which is referred to as a "federation". Each simulation unit is considered a "federate" of the federation, and the synch component (i.e. orchestrator) is referred to as a Runtime Infrastructure (RTI), which handles information mapping, federate synchronization, and the orderly execution of federates (i.e. federate coordination). HLA-based simulation frameworks has been successfully applied to some of the leading smart grid co-simulators.

FMI has been traditionally used for CT co-simulation, and specifies a standard interface for exchanging "functional mockup units" (FMUs) that contain either dynamical systems, or entire simulation units (dynamical system and solver) among different systems. However, none of the leading smart grid co-simulations identified in Müller *et al.* make use of this standard.

Mosaik is an interesting simulation framework, as unlike HLA and FMI, was intentionally designed for smart grid co-simulations running distributed energy resource and load control applications. Mosaik uses a master control program for the construction and orchestration of co-simulation scenarios expressed in its own Mosaik Specification Language (MoSL) to combine simulation units. The simulation units provide information to the master control program using a simulation interface (SimAPI).

Ad-hoc simulation frameworks are implemented within the simulation units, which can result in less organizational overhead in the framework design in comparison to advanced (standardized) simulation frameworks, but require the developer to deal with the development details of the co-simulation unit they are making. This approach is sometimes the only option available for co-simulation developers when they are trying to combine simulation tools that are proprietary and do not have support for the APIs that advanced simulation frameworks rely on.

2.3.2.3 Leading Smart Grid Co-Simulators

This section describes the most prominent smart grid co-simulators in existence that implement detailed power system dynamics (i.e. use non-steady-state power simulators) in no particular order.

EPOCHS [62]: The Electric Power and Communications Synchronization Simulator (EPOCHS) was the first smart grid co-simulator designed, combining PSCAD and PSLF

power systems simulators with the ns-2 network simulator. This co-simulator aimed to simulate protection and control systems that are implemented by a distributed system of "agents"; an agent is defined by the authors as autonomous smart grid applications that are programmed into the IEDs of the FREEDM smart grid architecture. EPOCHS uses an HLA-inspired ad-hoc simulation framework to build a dedicated synchronization and control co-simulation architecture, referring to its orchestration mechanism as a RTI, and each simulation and control entity as a federate, and the overall system as the federation. Despite their interest in HLA paradigms, custom interfacing techniques were implemented to avoid the complications of retrofitting the software of each simulator used to comply with HLA interfacing specifications. The RTI uses a simple fixed point synchronization methodology. The control component of the architecture was labeled the "AgentHQ", which executes the event-driven dynamics of all smart grid agents in an object-oriented manner, making the actions of smart grid applications controllable through methods at every macro step of the simulation.

GECO [63]: The Global Event-Driven Co-Simulation (GECO) framework is an ad-hoc framework used to merge General Electric's Positive Sequence Load Flow (PSLF) power simulator and ns-2 communication network simulator to simulate wide area protection and control (WAMPAC) smart grid applications. As the name implies, their orchestration strategy uses an event-driven synchronization methodology, treating every micro step of the PSLF simulator as events, which are managed globally alongside the events generated by the communication network model execution. A master-slave co-simulation architecture is used, where the orchestration mechanism is programmed in the ns-2 environment, making the communication simulator the "master". The authors developed an "epcmod" API in the PSLF environment that takes orders from the master; such orders include the pausing and starting of the PSLF simulator, and the querying and editing of the PSLF state variables [63]. A C++ class named tcl_PSLF in ns-2 was created that acts as the API on the communication simulator side, and are used by ns-2 application classes programmed to emulate the smart grid applications. The performance of GECO was compared to an EPOCHS-based co-simulator, by comparing the behaviour traces generated by both co-simulators for a WAMPAC application implemented on an IEEE 39-bus distribution system model. The main difference between EPOCHS and GECO is that EPOCHS uses a fixed time-stepped (point-based) synchronization methodology. It was shown that the fidelity of EPOCHS was only able to match that of GECO when the time step chosen was the same as the micro step of the power simulator. Finally, the authors studied the scalability of their co-simulation framework by seeing how much longer the simulation run time increased when scaling up the WAMPAC application to cover a 127 bus distribution system model. The results showed that the simulation time nearly doubled; the actual run time of the simulation depends on the time step size of

the PSLF solver. In the case where the PSLF solver was set to have a step size of a millisecond, the simulation run time increased from roughly 20 minutes to 40 minutes moving from the 39 bus model to the 127 bus model.

INSPIRE [64] is an HLA-based smart grid co-simulation scheme implemented by combining DIgSILENT PowerFactory and OPNET simulators for the simulation of WAMPAC smart grid applications. Despite using the aforementioned simulation units, INSPIRE co-simulation design in principle can be used to combine any pair of power and communication network simulators. The co-simulation architecture follows a dedicated synch and control design, which is needed for the application of HLA interfacing specifications. The authors describe their implementation of their co-simulation architecture in terms of "Hybrid Simulation Architecture" terminology [65], where the synch component defined in this thesis is analogous to their definition of a "simulation core", while our notion of a "simulation framework" is referred to as the "networking layer".

The simulation core performs the tasks of orchestration and co-simulation scenario characterization, with the former task being implemented by a "request-grant" dynamic between federates and the RTI, and the latter task being lumped into their notion of a "generic network description"; both tasks are implemented in part through the use of their so-called "substation data processing unit" or SSDPU. The SSDPU achieves this by accepting the IEC 61970 common information model (CIM) compliant description of the power system topology generated by the power simulator component, and identifying the IEDs of each substation bay whose communication system components require instantiation in the ICT simulation domain. At this point, the SSDPU can also link the substation nodes (in the power simulator) as appropriate to the ICT nodes (in the ICT simulator) derived from the substation information just processed.

The network layer, not to be confused with the communication network simulated by the ICT simulator component, is the communication network used by the co-simulation framework itself to connect the logically and/or physically distributed federates of the co-simulation so they can exchange information. The authors draw upon HLA-based communication protocols that allow the network layer to support the needs of the orchestrator. The SSDPU also helps with the network layer interfacing, handling the coupling between power system, ICT, and control components (aka federates) on a substation. In INSPIRE, as interpreted by fig. 2 of [64], has a SSDPU for every substation in the power system topology, which is responsible for populating an IEC 61850 substation model with state information that needs to be shared with the ICT simulator and control components via CIM-compliant data packages. Fig 2.4 is a version of fig. 2 of [64] that has omitted some finer details to place emphasis on the overall co-simulation architecture.

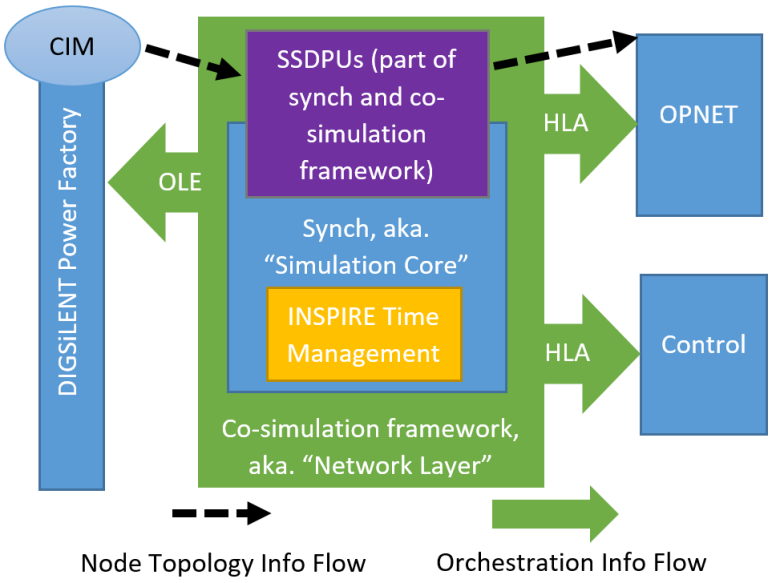

FIGURE 2.4: Simulation architecture of INSPIRE smart grid simulator.

The authors developed "INSPIRE time management", a dynamic point-based synchronization methodology, where the synchronization points occurred at every time step of the differential algebraic equation solver of the power system simulator. However, this means the event dynamics generated by the ICT simulator may be delayed in their influence on the power systems side, as they cannot be shared the instant they evolve, and must wait until the next synchronization point. This results in a loss in precision for the point in time that a discrete event can actually occur on, which is described by the authors as "temporal fuzziness". The impact of temporal fuzziness was investigated by running the co-simulation multiple times, each time changing the maximum allowable time step of the power system solver, thus elongating the time between synchronization points, increasing the presence of temporal fuzziness. As temporal fuzziness increased, the performance of the co-simulator deteriorated in that there were artificial communication delays that were an artifact of the fuzziness, and not the communication latency desired to be produced by the ICT simulator.

ORNL Hybrid DE Approach [66]: The Oak Ridge National Laboratory (ORNL) provide a Hybrid DE approach to coupling the CT and DE dynamics of power and communication system simulators. The authors mathematically characterize a DEVS tuple that internalizes the CT evolution of the power system, allowing the state of the dynamical system to advance through the integration steps of a differential equation (DE) solver in between discrete event triggered state transitions. As such, the input and output dynamics of the power system SU become purely DE based from the perspective of other DE SUs (i.e. communication and control SUs), making the coupling of their dynamics easier. The authors implemented this CT internalization feature with the their

novel "evolution" and "integration step" functions. The evolution function can return the new state of a CT system based on an input state and the margin of time the current state should advance by. The integration step function maps the state of the system to the time duration of the next DE solver integration step. The conventional DEVS functions were then expressed in terms of these functions (among others) to accommodate the CT dynamics that cause the state of the DEVS tuple to change continuously, and not only due to the occurrence of discrete events. The authors implement this hybrid coupling strategy through the use of ADEVS, a hybrid dynamic systems simulator, for the power system model, and ns-2 for the communication system model. The power system dynamics modeled in ADEVS is boiled down into a single discrete event process, and wrapped in an ns-2 TelObject, making it controllable by ns-2 software. Thus, the author's co-simulation architecture follows that of a master-slave strategy with an event-based synchronization methodology.

Bergmann et al. [67]: This work presents a layered co-simulation architecture, where the NETOMAC power system simulator and ns-2 are combined. The smart grid applications (i.e. control component) are separate from the power and communication system simulators. Two different interfaces were constructed in order to couple the various simulation units; one interface coupled the power simulator to the communication network simulator, and the other interface coupled the communication network simulator to the control component containing the smart grid applications. An interesting aspect of this work's architecture is that the communication network simulator can be interpreted to be a part of the interface between the power system and control components. NETOMAC has an interface called ExternBridge, which can be used to access a Java Native Interface API to ship data from the NETOMAC environment in an xml format to ns-2, where the data traverses a simulated communication network on its way to receptive socket(s) of the various smart grid applications in the control component.

VPNET [68]: This smart grid simulator combines Virtual Test Bed (VTB) and OPNET simulators via a dedicated synchronization architecture; the synchronization component of the co-simulator (aka their "Co-Simulation Coordinator") was developed by the authors in C#. VPNET uses an ad-hoc simulation framework that leverages existing software interfacing features present in both VTB and OPNET, and a fixed point-based synchronization methodology. Common Object Model based APIs are present in VTB, which is used by VPNET to send power system state information to the Co-Simulation Coordinator so it can then be sent to the OPNET environment; allow the Co-Simulation Coordinator to edit the VTB simulation parameters; and allow the Co-Simulation Coordinator to stop and start the execution of the VTB solver. The various nodes simulated

in OPNET use OPNET's "external system module" (aka Esys) interfaces for the bidirectional exchange of information to/from OPNET's execution controller, which can communicate with the Co-Simulation Coordinator using a Windows Socket.

GridSim [69]: GridSim is a real time smart grid simulator intended for the simulation of WAMPAC applications. The article describing GridSim did not put emphasis on the details on how the authors interfaced the different simulation units for power network, substation, communication network, and control centre components of their design, and instead described the WAMPAC applications capable of being simulated in GridSim (hierarchical state estimation and oscillation and damping monitoring) in addition to the author's proposed smart grid "data delivery middleware" (GridStat) designed to provide communication network QoS guarantees for such applications. It is unclear if conventional communication network protocols can be modeled with the communications component of their co-simulator, or if the fidelity of this component is high enough to model the dynamics of lower-layer communication protocols.

PowerNet [70]: PowerNet Combines Modelica and ns-2 simulators using a master-slave architecture, dynamic point-based synchronization methodology, and ad-hoc simulation framework. The master simulation unit is ns-2, which is programmed with an orchestration mechanism that can read and write information from/to the Modelica environment, and control the execution of Modelica. Such orchestration commands are event-driven, making the synchronization time step dynamic. The two simulation units are interfaced through the use of UNIX pipes.

Babazadaeh et al. [71]: This work presents a real time WAMPAC testing platform developed with OPAL-RT and Simulink for the power system SU, SoftPMU to emulate phasor measurement units (PMUs), OPNET for the communication system SU, OpenPDC for the simulation of a phasor data concentrator (PDC), and PowerIT for the smart grid application implementation platform. The co-simulator was only demonstrated to perform uni-directional data flow from PMUs and video surveillance cameras to control centre applications in [71]. Synchrophasor data is modeled to be generated from power system measurements and encoded in an IEEE C37.118 format by the SoftPMU module, where it is sent in real time to the OPNET environment via TCP/UDP protocol. The OPNET simulator can recreate the real time traversal of data packets through some communication network model, and has a "system in the loop" feature that allows for various network components to be replaced with actual hardware. After traversing the communication network, the data is fed into a PDC simulated by OpenPDC so the data can be aggregated and accessed by the various smart grid applications written in PowerIT.

Greenbench [72]: Greenbench is combines PSCAD and OMNeT++ simulators to create a co-simulator capable of simulating the impact of cyber attacks on a FREEDOM-based smart microgrid. The authors implement a master-slave simulation architecture, where an orchestration mechanism (named the "interactor") is built in OMNeT++ through the help of OMNeT++'s "self event" artifacts that are generated by simulation components after uniform increments in time as an implementation detail. These self events can be exploited to implement a fixed point-based synchronization mechanism for the orchestrator. An ad-hoc simulation framework is used to interface the simulation components with "buffer files", where information that needs to be sent by one simulation component is stored in a binary file that is accessible by the other. Such buffer files are used to exchange simulation information at every synchronization step.

Garau et al. [21]: In this work, the authors' main objective was to test the feasibility of LTE-enabled fault location, isolation, and restoration (FLISR) applications. In the process of doing so, a co-simulator was constructed in order to simulate the coupled dynamics of power systems and LTE communications in particular. DIgSILENT PowerFactory and SimuLTE (executed in OMNeT++ environment) were combined in a dedicated synchronization architecture, with an ad-hoc simulation framework seemingly inspired by HLA, as the authors refer to their synch component as a run-time interface (RTI). The RTI was implemented in Python, allowing for easy interfacing with DIgSILENT PowerFactory, which comes with a Python API. TCP sockets were used to interface the RTI with OMNeT++. A fixed point-based synchronization methodology was used by the RTI.

2.4 Potential Use Cases for D2D in Smart Grids

2.4.1 On Smart Grids: The FREEDM Architecture

Traditional power systems have a centralized architecture with unidirectional power flow from producer to consumer. Such systems involve large power plants feeding power over long distances via high voltage (usually 115kV or higher) transmission systems, whereupon the power is disseminated to transmission substations that step this voltage down to a medium level (usually from 4kV to 42kV). From there, the power flows through a distribution system to be consumed by the users of the electricity (e.g. a home or office building). Before the customer consumes the power, its voltage is stepped down one more time to the standardized secondary side voltage of 120V/240V. Traditional grids have limited means for interconnecting distributed energy resources at the low and

medium voltage levels of the power system, have limited communication infrastructure, and forbids islanding during outages [73].

In recent years, the rising needs for increased penetration of renewable energy resources, more intelligent coordination of dispatch of both supply and consumption of power to mitigate overloading infrastructure, and grid resiliency have driven a movement to modernize the power system in order to meet these needs.

The result of modernizing traditional power infrastructure is a "smart grid", essentially a power system made "smart" by the integration of information and communication technology and advanced power conversion infrastructure that can result in the system being capable of handling high penetration of distributed energy resources in addition to other applications that aim to make the grid more cost effective and resilient for the benefit of all its stakeholders. In a smart grid, system components can monitor their operation state and communicate to other components as required to uphold not only nominal, but optimal operation of the grid. A popular manifestation of smart grid is demand response, an activity where the consumer is involved in altering their power consumption scheduling in real time in response to the needs of the grid infrastructure among other things such as electricity pricing. Another way smart grid is used is to facilitate the coupling of micro grids to the main distribution system of a local distribution company (LDC) to not only ensure a safe grid tie, but to also provide a myriad of ancillary services to the larger power ecosystem, such as reactive power supply for voltage and frequency regulation, peak shaving, and energy imbalance compensation [73].

The FREEDM system is a model smart grid infrastructure proposed by [74], which envisions the essential grid components that would create an easily extensible (i.e. "plug and play") distributed power system, whereupon distributed energy resources such as storage devices and renewable power sources like photovoltaic panels can be coupled to the grid with the same ease as connecting a phone to a battery charger. Fig. 2.5 visualizes the FREEDM model. The architecture proposed has the same base structure as a traditional distribution system (primary voltage bus, power feeders, and distribution transformers to step down voltage for consumer consumption), but adds intelligent communication devices throughout the system. Intelligent Fault Management (IFM) devices are proposed to quickly locate faults and respond by isolating the afflicted primary feeders from the rest of the power system, whereas Intelligent Energy Management (IEM) devices are to be used to communicate to other IEMs, head office, and IFMs on behalf of the secondary side distributed energy resources they are managing [74]. Another novel addition to traditional distribution infrastructure is the introduction of the Solid State Transformer (SST), which is to replace analogue distribution transformers

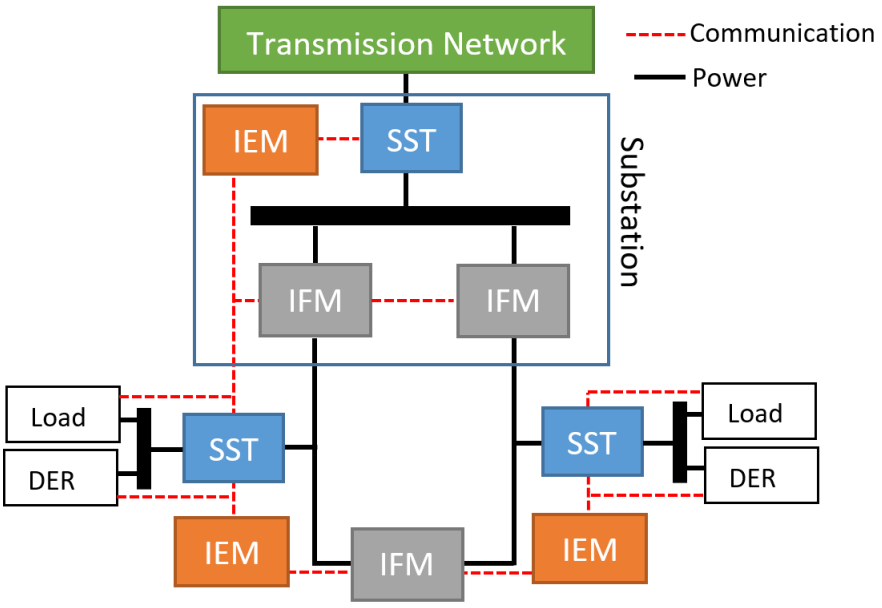

FIGURE 2.5: FREEDM architecture diagram.

(at least where DER is being coupled to the grid), acting as a node to which both secondary side DC and AC buses can be implemented to connect various DER to [74]. The SST is a power electronic device, and is to be capable of inverting and rectifying power waveforms as required; also, the authors identify how the SST can maintain unity power factor at the primary side of the SST to assure that no interconnection problems would arise when DER are coupled to the grid [74]. This requirement seems slightly dated today, as there is now talk on using DER for VAR injection to provide voltage and frequency support to the distribution system [73].

2.4.2 Applying D2D Technology to Smart Grid Applications

The work of Kalalas *et al.* [6] identifies mobile communications as a promising technology choice for smart grid ICT infrastructure due to its technological maturity, pre-existing infrastructure maintained by dedicated entities, ease of deployment (low up front and maintenance costs), and its usage of licensed spectrum that allows for QoS provisioning. Nonetheless, the authors did identify that for the massive machine type communications that are associated with smart grid communications in addition to the bursty protection and control messaging that demands highly reliable and low latency communication, LTE standards require some improvement before being able to provide all the communication needs of future smart grids.

Kalalas *et al.* identifies that D2D communications can help with some of the shortcomings of LTE applied to smart grid communications as it has the potential to boost throughput, improve reliability and latency performance, and prevent the occurrence of

LTE network overloading. As identified in Section 2.2.1.1, underlay D2D communication is capable of boosting the throughput of a cellular network via frequency reuse, and is a potential solution to the large data rate requirements for the alarm and protection functions of substation smart grid applications [6]. Moreover, with the smart grid devices being able to communicate without the base station, it will result in less traffic congestion on the LTE core network and base station infrastructure. Communication reliability can be improved via D2D communications as well, as transmitting data over both a direct link in addition to conventional cellular channels introduces redundancy to message transmission efforts. Lower latency can be achieved through direct communications as well, as such information streams avoid the traversal of the core LTE network.

The authors identify three types of smart grid applications in which D2D communication can be deployed at a neighbourhood area network scale to improve the feasibility of LTE enabled smart grids in the ways previously mentioned, namely microgrid distributed management and control, smart substation automation, and active demand response [6].

Microgrid distributed management and control, as the name implies, refers to the applications required for the management and control of microgrid power system dynamics for the continuity of electrical servicing while using a distributed control architecture. As already mentioned, LTE is an ideal technology choice for smart grid communications from an economics and assets management standpoint. However, the essential control mechanisms required for a microgrid (e.g. protection mechanisms that handle system disturbances such as abrupt load changes or faults) require millisecond-scale latency performance, making the use of conventional LTE communications unfeasible from a technical standpoint. Thus, the authors [6] claim how various papers suggest that by shifting the conventional centralized and hierarchical microgrid control architecture to a distributed multi-agent based one, low latency D2D communications can be used, thus improving the feasibility of LTE-enabled microgrid control.

Smart substation automation applications are part of the cyber physical system that manages various aspects of power system performance at distribution substations, where IEDs at substations process information sensed from various devices such as current and voltage transformers in order to attain awareness of the grid state required to perform actuation commands as required to handle grid contingencies and maintain continuity of electrical servicing. Such actuation commands could open circuit breakers, shift the settings of a transformer's load tap changer, dispatch capacitor banks for VAR injection, etc. The authors [6] claim that such substation automation communications are limited to IEDs within a single substation, but the reach and power of such applications can be

extended if IEDs belonging to different substations could communicate to one another. However, the wired communication currently used in substation LANs is not scalable from a cost standpoint to implement inter-substation communications, where LTE communication is, but is not suitable for mission critical applications. It is proposed that low latency D2D communication can be used to make LTE communication possible, unlocking unprecedented levels of distribution system operation automation.

Active demand response allows consumers to have access to real time power pricing and consumption information in order to have more insight into the optimal dispatch of their electrical loads in addition to any potential distributed energy resources in their possession for their economic benefit (e.g. sell energy to the grid when prices are high, defer schedule-able power loads like washing machines when the price of electricity is cheapest, etc.). Moreover, distribution system operators can send price signals to consumers to influence their power consumption habits in order to stay within various distribution system operation constraints, e.g. raising the price of electricity during peak hours to reduce electricity demand to keep transformers from being overloaded. The authors identify D2D communication as a useful enabler for demand response, citing studies that advocate for D2D enabled mesh networks to be used to send pricing and power consumption information to demand response stakeholders, and implementing relaying mechanisms to extend the communication range of the mesh via additional single link hops.

2.4.3 Literature Review - Potential D2D Driven Smart Grid Use Cases

Cheng et al. [75]: In this paper, the authors identify that the large data requirements generated by the advent of the smart grid that may put the cellular network at risk of being overloaded with smart grid device data flows. To alleviate stress on a cellular network enabling a smart grid, the authors propose to offload the delay-tolerant smart grid information to a heterogeneous D2D network, where the smart grid devices transmit data directly to passing vehicles via outband WiFi-enabled or inband underlay cellular D2D communication, depending on what the author's transmission mode selection algorithm deems optimal for any given scenario. Then, the vehicles will relay the smart grid devices' messages to the next roadside WiFi access point they happen to come within range of, which will then send the information to its final destination through the core network.

The authors formulated two different data forwarding schemes that help smart grid devices choose a statistically optimal vehicle to choose to send its data to, and also develop mode selection and resource allocation algorithms. The data forwarding problem

was formulated as an optimal stopping problem, and the authors provided a "time-oriented" solution, and a "vehicle-oriented" solution to the same problem. The time-oriented optimal forwarding scheme divided time into discrete slots, where at every time slot, the device would consider if the vehicle it currently has the opportunity to send its data to will be less likely to result in higher communication delays than those consequential to waiting for the next time slot for a better forwarding opportunity. In the vehicle-oriented optimal forwarding scheme, an optimal rule was defined that determines a threshold time duration that the expected delay in message delivery from vehicle to roadside access point should be below to be worth forwarding; otherwise, from a communication delay perspective, it is statistically favourable to wait for the next vehicle to arrive, and evaluate the forward opportunity it has to offer.

Concerning the mode selection problem, the driving metrics that motivated the mode selection process were monetary cost for the communication services. Outband D2D WiFi was assumed to be free to transmit data with, while inband underlay LTE D2D was more expensive. Moreover, the author's communication network model assumed that if data sent via D2D communication is not received on its first attempt, the smart grid device will re-transmit using conventional cellular communications with an eNB. This conventional cellular communication mode is modeled to have the highest cost among all available transmission modes. The successful data transmission probabilities for outband and inband D2D communication were determined as a function of many communication and device mobility model parameters. These probabilities were used in their mode selection scheme to determine in any given scenario if it is statistically cheaper to transmit via outband or inband D2D. Generally speaking, outband D2D is cheaper if it is successful in getting the information to its destination, but has a higher probability of failure than inband D2D, meaning this scheme has a higher risk of needing to re-transmit using the most expensive communication mode, namely conventional cellular communication. This risk-reward trade off is what makes the probability of transmission success so vital to the mode selection decision.

Finally, a centralized "fair delivery-ratio maximization resource allocation" scheme was proposed to maximize the lowest "delivery ratio" among all smart grid nodes. The delivery ratio is defined as the ratio of data transmitted by a smart grid device via cheaper D2D communications to the total data transmitted in the network. The main novelty behind their resource allocation scheme was the author's implementation of virtual queues for each smart grid device, where the length of a device's queue evolves over discrete advancements in time, where the difference between minimum delivery ratio desired and the delivery ratio achieved over the previous time interval is added to the length of the queue. The queue lengths are then used as weights in a sum of the achieved delivery ratios of all smart grid devices in the network, which the author's

resource allocation algorithm maximizes. This way, the algorithm allows devices who have historically been allocated the lowest delivery ratios in the past are given priority for the current time interval in which target delivery ratios are being allocated.

Ultimately, the author's vehicle-assisted offloading scheme resulted in the ability to offload 58% of delay-tolerant smart grid data to vehicles via D2D communications, resulting in fairly allocated cost savings for all smart grid devices at the same time.

Piacentini [76] and Fey et al. [77]: These are two early works that presented various ideas revolving around distributed smart grid energy management and control. Piacentini identified in 2011 that cost and performance improvements being made in FPGA technology was facilitating the installation of smart nodes throughout a power system that can handle computational burdens required to perform smart grid applications. It was also identified by Piacentini that the use of distributed smart grid intelligence can result in a reduced burden on wireless communication networks, and by discarding the need for a centralized, hierarchical SCADA system that is sensitive to a single point of failure, a more robust smart grid can be realized.

Fey *et al.* aimed to find a solution to making heterogeneous communications infrastructures that various smart grid applications use in an inter-operable fashion. The authors present an agent-based smart grid control system, where control nodes in the same low voltage grid must coordinate their actions to achieve some common goal, e.g. voltage control. However, to facilitate the communication required among agents in a distributed microgrid control scenario, the authors identified that the agents need to be able to discover one another's presence in the multi-agent system at "first-plugin"; provide locational information; and advertise smart grid services, all while upholding the basic needs of QoS and communication security. Ultimately, with these base functionalities, the applications of various agents in a power grid can communicate directly to one another, making multi-agent control feasible. This direct application-to-application functionality is what the authors define as device-to-device communication, where such communications are direct only in the logical sense, and in reality are logical overlays to an underlying multi-hop heterogeneous network. Although Kalalas *et al.* made it sound like Fey *et al.* was identifying the need for the cellular D2D communication as defined in this thesis for distributed multi-agent microgrid control schemes, this is not true, as Fey *et al.* did not define D2D in the same way. In light of this, one must consider whether or not multi-agent based power system control strategies even lend themselves to cellular D2D communications any more than centralized schemes do. In reality, it is not the control methodology that influences the feasibility of D2D communications as a solution for information exchange in a smart grid, but the proximity between nodes that require communication with one another. For example, if some multi agent control architecture

has agents that are several kilometers apart in order to cover an entire distribution system, a D2D connection may very well be not be feasible, whereas a centralized microgrid control strategy may have a its controller within a few hundred meters of all other nodes in the system, making D2D possible.

Ci et al. [78]: This work aimed to model the communication delays of a wireless mesh network used to implement a community microgrid load sharing application, and simulate the impacts of such delays to observe their impact on the power system. Moreover, a novel load sharing control scheme was developed to make the application more robust to communication delays. Load sharing is the practice of synchronizing the voltage and frequency magnitudes of parallel power inverters connected to a common AC bus that distributes power to loads on the microgrid. These power inverters are used to convert the power output by different kinds of distributed energy resources (e.g. wind, solar, batteries) to a commonly acceptable form. The authors modeled the communication delay of a wireless communication link as the summation of two random delays, namely the "service time" for data packet to traverse the communication link, and the queuing delay experienced. The service time is modeled to follow a geometric distribution, and the queuing delay is based on previous works in queuing theory, and makes use of the service time probability distribution's first and second order moments. Information loss probabilities and retransmission schemes are also modeled, but the details are not included in the paper.

It was shown that the load sharing control scheme that synchronizes the output of a power inverter interfacing a distributed energy resource to the microgrid relies on the power output information of the other parallel inverters in the microgrid in order to generate the controller's proper instantaneous reference voltage values. As such, the delayed power output information from other inverters results in sub optimal voltage references for the local inverter's controller, which can cause power system instabilities that can harm the loads on the microgrid. The authors reduced the impact of the instabilities observed in simulation caused by their modeled communication delays in a conventional load sharing controller implementation approach with a new control strategy. The key idea of this new strategy was to delay the local power output feedback loop by a time delay that is randomly distributed according to the same distribution as the communication latency from the remote power output feedback messages. The simulation results showed that the author's approach significantly reduced the drop in active power output of the inverter in comparison to the conventional control scheme when both were exposed to message delays in the microgrid.

This study shows that wireless communication can be a feasible solution to microgrid load sharing control systems, as communication delay effects can be handled by altering

the control scheme. With this being said, microgrid load sharing appears to be an appropriate use case for D2D communications, and due to its low latency potential, may even be capable of eliminating the need for altering conventional control schemes to mitigate latency impacts in future microgrids, yielding simpler control designs.

Cao et al. [79]: In one study, D2D communications was applied in a smart grid context in order to reduce the information loss rates of a FREEDM-based energy system [79]. The article identified the importance of latency in smart grid operations, as there are many time critical operations that take place in energy systems. Also, it described the hierarchical nature of communication infrastructure commonly proposed for smart grid systems, involving random-access based communication based LANs for communication of devices on the same low voltage bus managed by a single IEM, followed upstream by a WAN implemented using cellular networks. For an IEM to generate a report to a distribution system's control room, data would have to be collected from all devices on the LAN by the IEM, which would then report as required to head office through the WAN. To improve the spectral efficiency of the WAN, it was proposed that IEMs can share their allocated cellular resources in order to leverage the resources that are unused by IEMs that have less information to send than others [79]. For example, an IEM experiencing fault conditions will require higher bandwidth than nominally required to send all the fault reporting messages to head office; meanwhile, another IEM not undergoing a fault may have been allocated resources that are sitting idle. The authors proposed the concept of D2D-assisted relaying to help tap in to such idle resources [79]. Traditional relaying involves a UE sending data to an intermediary node that relays its signal to a base station in order to boost spectral efficiency by leveraging spatial diversity [79]. However, the authors took this concept a step further and considered the possibility of a IEM using radio channels of another IEM to form a D2D connection to a relay node while still using their own channels [79]. The optimal pairing of IEMs (for D2D relaying) in addition to the transmission mode of each IEM (i.e. IEM to base station, IEM using traditional relaying, IEM using D2D assisted relaying) was modeled as a 2-stage stochastic programming problem and solved [79]. The programming problem was stochastic due to the random latencies experienced in LAN communications [79]. In the simulations conducted by the authors, it was demonstrated that the usage of D2D-Assisted relaying in smart grid communications reduced information loss rates in comparison to communication "frameworks" that either did not use relaying at all or only used traditional relaying [79].

Chen et al. [80]: In this paper, the authors present a way to dynamically form microgrids with available distributed generation available after a natural disaster has disconnected the distribution system from centralized power sources fed by the transmission system. This microgrid formation mechanism intends to restore the most critical

loads on a distribution system in a timely manner with the distributed energy resources available. The authors consider a radial power distribution system that is outfitted with remotely controllable switches that can automate the reconfiguration of the system topology to form islanded microgrids, with each distributed generation resource powering one microgrid. The problem of determining the optimal formation of microgrids was formulated as a mixed integer linear programming problem (MILP), where the loads in the distribution system are selected to be connected to a microgrid in a manner that maximizes the priority-weighted sum of power servicing being restored post-disaster. The authors were mindful of the constraints of the problem, ensuring that the load flows at each node balanced, and that line impedences were taken into account to calculate the corresponding voltage drops to ensure that voltage levels of nodes stay above an acceptable level. Moreover, the effects of the disaster can be introduced as constraints in the problem, by making certain switch states unchangeable by the optimization as a result of the need to isolate the faults introduced to the distribution system. The MILP formulated is solvable with commercial software.

In addition to the load restoration optimization, the authors identified the need for global state information of the microgrid must be collected by a node so that the optimization can be performed. It is assumed that there are communication nodes distributed over the power system that have partial information about the overall system, i.e. information pertaining to a handful of switches and loads that it is close to. The authors proposed that information can be shared among nodes by applying the average consensus method, a distributed means to disseminating information throughout a network. At a high level, the average consensus method works by iteratively exchanging information with local nodes, where in every step, a node observes the differences in its recorded values of each grid element parameter (e.g. a load's nominal power demand) with those recorded by its neighbouring nodes. These differences are scaled by a learning rate, and added to the values of the previous iteration. The average consensus method, after sufficient iterations using a properly selected learning rate, results in each node's recorded values of each grid element parameter being equal to the average of the initial values for such elements held by all nodes in the system. Thus, if all nodes collectively know all information elements of the distribution system, and if all nodes hold information on disjoint sets of elements, with the element values they do not have access to being initialized to zero, global information can be recovered from the average initial values found by average consensus method.

The authors demonstrated a proof of concept with two case studies, where they implemented their load restoration strategy on both IEEE 37 and 123 node test distribution systems, where the 37 node model had 3 distributed generators, and the 123 node system had 8. The generation capacities and load demands were chosen randomly. Successful

formation of microgrids was demonstrated in addition to convergence of the average consensus algorithm used to disseminate information.

The communication requirements of the load restoration scheme was identified by the authors to only need short range wireless networks, as general consensus method only requires communication among local nodes. As such, the authors suggest that WiFi or ZigBee can be used. However, it seems that cellular D2D communications would also be a good fit for this application, and would even benefit from its access to licensed spectrum and improved feasibility for longer communication ranges. In fact, these advantages of D2D communications over other wireless communications options are the main source of this thesis' inspiration for developing measures to make D2D accessible to many different smart grid application archetypes.

Chapter 3

D2D LTE Enabled Smart Grid Simulator Design

3.1 Introduction

As mentioned in Chapter 1, due to the widespread use of Simulink's Simscape Power Systems to facilitate power engineering research [27–33], it is believed that extending its functionality into the realm of LTE communications without needing software outside of the Matlab/Simulink environment would be of value to the academic community. Nonetheless, there is little work done in developing a smart grid simulator solely in the Matlab/Simulink environment despite its powerful software packages for simulating the LTE physical layer, discrete event dynamics, and conventional power systems. Elkhorchani and Grayaa [81] tested their proposed smart grid wireless communication architecture on top of their power-flow model of a renewable-integrated power system in the Matlab/Simulink environment. The authors modeled ZigBee, WIMAX, and WiFi physical layer communication protocols, albeit assuming constant SINR levels, and do not seem to model a communications protocol stack up to the application layer. Shlebik et al. [82] simulates a PLC-enabled AMI application in Matlab, modeling the transmission of smart meter data via various modulation schemes. Moreover, despite the wide choice of protocols and architectures that can be simulated with the communications simulators used by state of the art smart grid co-simulators [61], such simulators (e.g. ns-3, OPNET) do not offer a high fidelity characterization of the dynamics present at the physical layer of a protocol stack. More specifically, such communications simulators are considered "layer 2" simulators, meaning that they ignore or heavily simplify physical layer ("layer 1") dynamics, e.g. fading effects, physical resource allocation, etc. Conversely, our co-simulation tool has the advantage of being capable of simulating the

propagation of time domain electromagnetic waveforms that have been modulated with digital information compliant to LTE specifications, making highly detailed analysis on effects of signal attenuation, fading, and interference possible.

This chapter describes the use of powerful simulation packages from the Matlab/Simulink environment and develops a novel interface software that allows simulation of NAN-scale smart grid applications enabled by LTE-compliant D2D communications. The readily available packages are LTE System Toolbox, SimEvents, and Simscape Power Systems. The interface software that allows power system and ICT dynamics to be simulated at the same time is described. Platform testing is achieved by simulating a simple fault location, isolation, and restoration (FLISR) application, and confirming that its power and communications dynamics interact with one another as expected.

The remaining sections of this chapter are organized as follows: Section 3.2 provides an overview of the Matlab/Simulink implementation of this model. Section 3.3 describes the FLISR application devised for testing the simulator. Section 3.4 presents the test results, showing successful execution of the FLISR application on the power system side, with its control signal actuation timing being consistent with that of the messaging dynamics observed in the LTE network.

3.2 Integrated Simulator Model

As mentioned in Chapter 2, the co-simulation of power and communication systems for the simulation of smart grid applications is not a trivial task. This is due to the difficulty in properly synchronizing the event-based dynamics of the communication simulator with the continuous time dynamics of the power system. The main goal of this section is to provide a system design perspective of our co-simulator's CT-DE dynamics and its Simulink-Matlab interface. The purpose of this approach is twofold. First, being able to characterize the joint dynamics between our power system simulator (CT based) and LTE network simulator (DE based) with mathematical equations which generalizes our solution to the problem of merging such simulation units. The consequence is that future practitioners can refer to our generalized system design as a guide to implementing their own custom co-simulation of some continuous time system coupled with a communication network that uses a time slot based medium access control scheme. Note that, in our co-simulator, we have chosen LTE as the communication facilitator however any communication technology can be integrated using our methodology.

Secondly, a mathematical characterization of our simulator's Simulink-Matlab interface unambiguously describes the dynamics of our co-simulation architecture, providing clarity and reassurance. This is an essential step for encouraging practitioners to consider utilizing our hybrid simulation approach. We desire to promote the legitimacy of this approach as an alternative to more complicated co-simulation architectures proposed in the literature [83, 84] based on the leading High Level Architecture (HLA) and Functional Mockup Interface (FMI) co-simulation frameworks, whose brief descriptions can be found in Müller et al. [61].

3.2.1 Software Architecture

The proposed solution to the comprehensive smart grid simulator combines the LTE System Toolbox, Simscape Power Systems, and SimEvents software packages that are available in the Matlab/Simulink environment. In addition to using these software packages, original code was developed to model the relevant 4G LTE protocol layers that the LTE System Toolbox does not support.

Fig. 3.2 provides a high-level system diagram of the simulation platform. In the Simulink environment, both Simscape Power Systems and SimEvents discrete event simulation tools are used. Simscape provides easy to use block components for power system devices, modeling their dynamics and providing the user with a simple interface to modify their characteristics. Depending on the state of the power system constructed using Simscape, sensors monitoring state variables, e.g. line voltages and currents, may trigger an "event" in the SimEvents domain. SimEvents allows one to simulate system behavior that responds to intermittently occurring "event" inputs. For example, such an "event" could be a timer within a smart meter expiring indicating it is time for the device to report its measurements to another device in the smart grid via a communications message. These events produced in SimEvents proceed to call Simulink functions that in turn evoke the applications of smart grid communication devices, which are implemented as Matlab scripts.

In the simulator, smart grid device application scripts construct data packets to send through a communication network in the form of a binary array. This data is passed to the LTE protocol stack. The LTE protocol stack was simplified, only capturing the features relevant to the flow of data through the communication system. As such the packet data convergence protocol (PDCP) layer, which manages header compression and encryption, was ignored for simplicity. However, the radio link control (RLC) layer is implemented in our simulation platform, as it fulfills the key role of concatenating and segmenting internet protocol (IP) packets (technically PDCP packets) buffered by

the application into RLC protocol data units (PDUs). Fig. 3.1 visualizes this process. This figure also indicates the header fields added to the RLC SDU; as this data plane is used for D2D nodes, the Unacknowledged Mode (UM) of the RLC layer is being used. As such, UM mode of RLC has FI, E, SN, E, and LI header fields. The FI field is a two bit field that indicates whether or not the beginning and end of the PDU's payload consists of a segmented upper layer packet. The first E field is a single bit that indicates if the RLC PDU has an extended header. The SN field is a sequence number used in the reassembly of segmented packets. The second E field is only present if the RLC PDU has an extended header, which occurs when RLC SDUs are being concatenated. As such, the second E field is indicates if a third E field is needed, and so on. This mechanic allows for a dynamic number of LI fields, which indicate the size of its corresponding RLC SDU, or RLC SDU segment. The sizes of the RLC PDUs are determined independently from the sizes of the IP packets that are being used to fill the PDU. This process ensures that the data packets being sent over the air interface can be encoded within the interval of frequency spectrum allocated to it for the given Transmission Time Interval (TTI) it is being sent at.

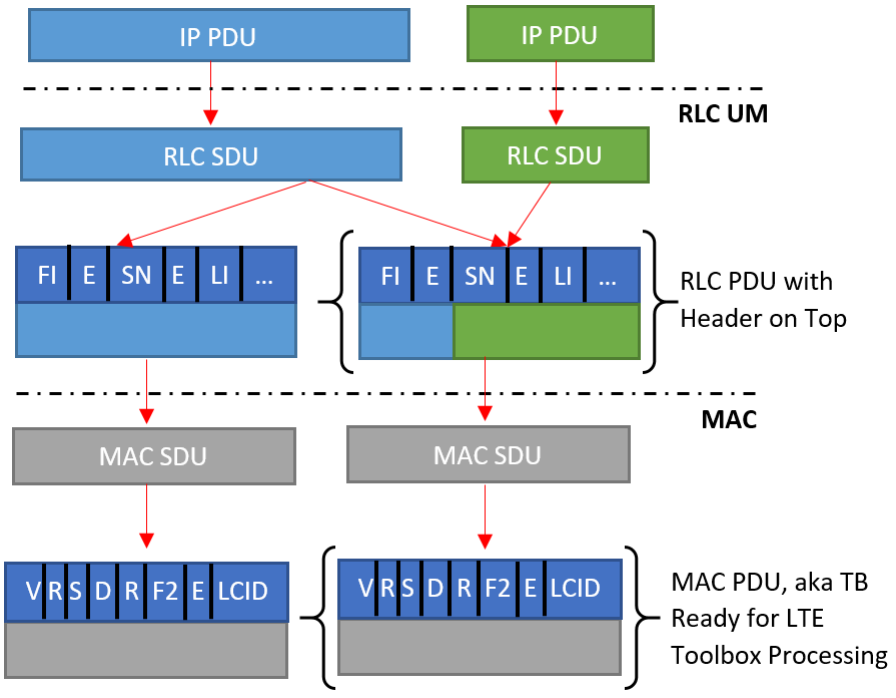

FIGURE 3.1: Implementation of LTE D2D data plane.

These RLC PDUs are then given a medium access control (MAC) layer header, which specifies information such as source and destination IDs of sender and receiver in the case of D2D MAC headers, in addition to other pieces of information like the Logical Channel ID (LCID) that specifies the logical channel of the data for data plane messages, or the type of control element (CE) being sent for control plane messages. It should be noted

that fig. 3.1 does not have a CE header field shown in the MAC header. This is because SL-SCH, i.e. sidelink data plane messages do not need control elements. However, there are several other fields for this type of message, namely the V, R, Source ID (S), Destination ID (D), F2, E fields in addition to the aforementioned LCID field. V is a "version number" field that is intended to be used in the future. R indicates"reserved" bits that are currently not being used for anything. The Source and Destination IDs are specified by higher layer protocols, and are used to identify the D2D node that is transmitting data, and a "Group ID" that identifies the group of receivers receiving the data [36]. In this work, these ID values were arbitrarily set. F2 indicates the format of the MAC header, and E indicates if there is another MAC SDU in the PDU aside from the one the current header fields are describing. In this work, it was never needed to combine MAC SDUs, as our communication nodes only ever had one stream of data coming from one application at a time. Once MAC headers are added, the data has been refined into transport blocks, and are ready to be input into LTE Toolbox functions, which modulate the binary data into a time domain waveform. On the receiving end, the reverse of this process occurs, ultimately resulting in a different smart grid entity application script receiving a message that results in the occurrence of an "event" in SimEvents, which in turn causes some form of actuation to be performed within the power system, e.g. a circuit breaker opening, or the inputs to some power controller changing.

3.2.2 DEVS Model of ICT-Power System Interface and its Implementation

The interface between the power and communications system simulator dynamics can be modeled with the collection of DEVS atomic models characterized in this section. Fig. 3.3 is a diagram of the interacting DEVS models. For any given D2D-enabled smart grid application, it can be modeled as having N "message generator" nodes that accept power system sensor readings and/or commands from "message receiver" nodes, then generate appropriate communications messages that get queued within the atomic model in response. An "abstract LTE network" block's dynamics is synchronized with the power system, and manages the N message generators, making forwarding and message dropping decisions every subframe the simulation advances. The N generators forward their messages to the appropriate sub-collection of N message receiver blocks, which process the incoming message(s) and send outputs to its corresponding message generator and/or power system actuation mechanism as required. The input/output coupling of the message generator and receiver atomic models can be modeled as a bipartite digraph, where the message generator nodes have one or multiple unidirectional

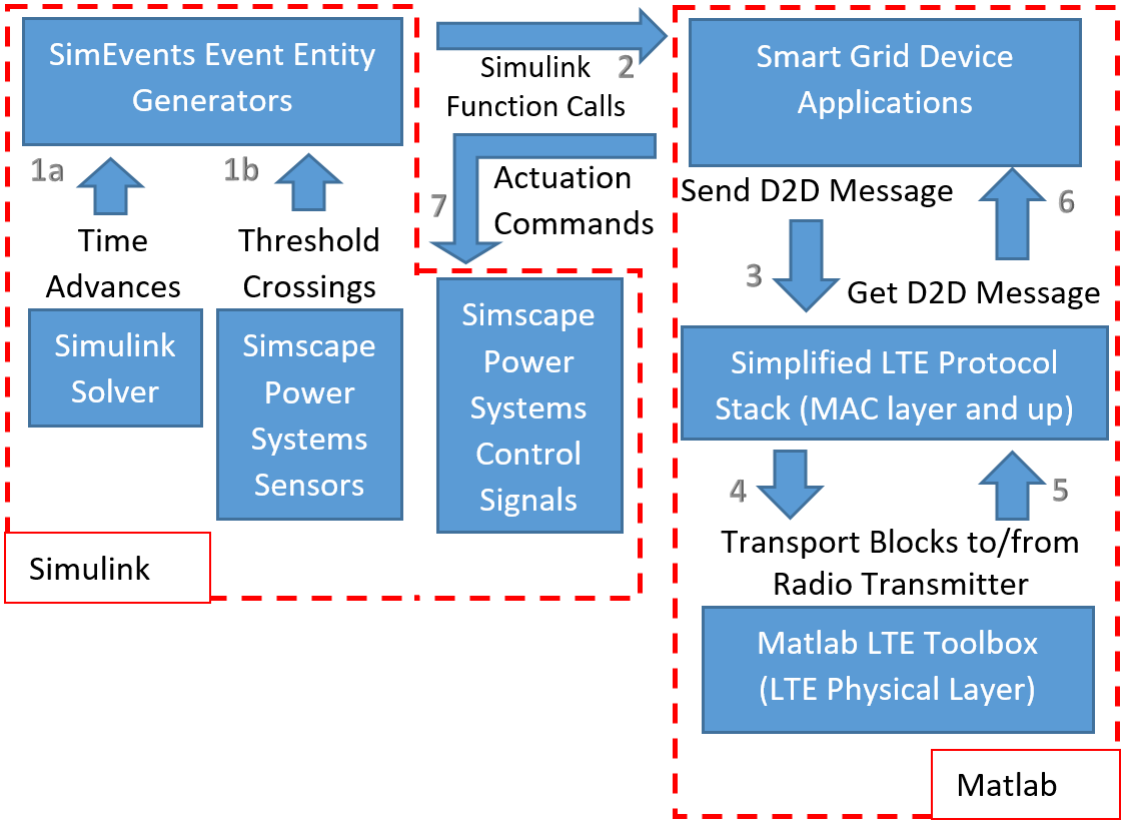

FIGURE 3.2: High-level system diagram of smart grid simulator.

connections to receiver nodes, and every receiver node has a single unidirectional connection to a single generator node. This graph structure captures how D2D nodes transmit by multicasting messages to a group of receivers, while a D2D node's receiver can only send information to its own transmission entity.

3.2.2.1 Message Generator Atomic Model

The message generator atomic model accepts either instantaneous measurements of power system state variables; signals from a message receiver block; or new queue state and data forwarding instructions from the abstract LTE network atomic model as potential inputs.

Should the input be from a receiver entity or the power system, the message generator generates appropriate binary messages, and stores them in a queue that is expressed as part of the atomic model's state variable. After enqueuing the appropriate messages, the new state of the queue is output from the model to be accepted by the abstract LTE network atomic model, so the LTE simulator is aware of the new information.

Should the input be from the abstract LTE network, the message generator updates the state of its internally documented data queue with the new input information, and then

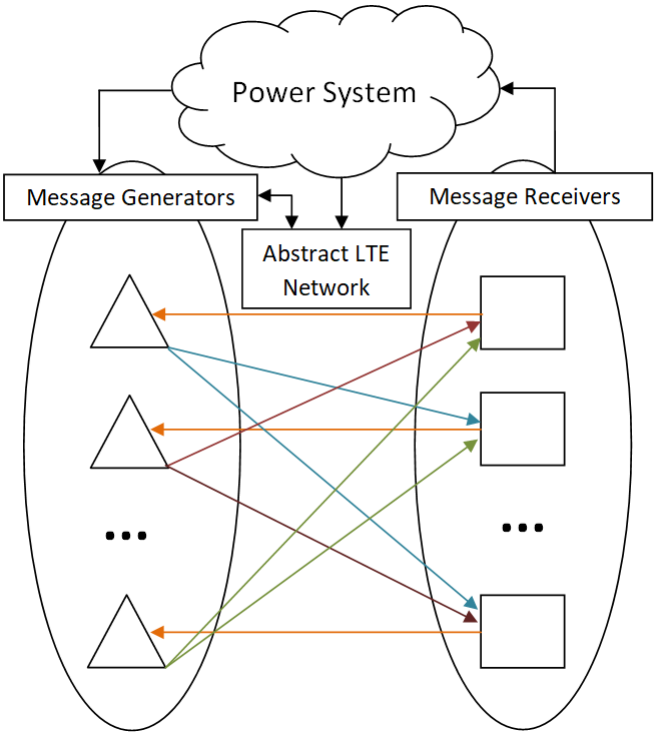

FIGURE 3.3: Input-output coupling diagram of DEVS atomic models.

outputs the first f messages in the updated queue. f is included in the input from the abstract LTE network.

The DEVS tuple is characterized as follows:

- X: The set of possible inputs is defined in terms of three sets and an "*enqueue* label". The first two sets are a set Γ of integer m-tuples with n or fewer digits (i.e. a set of measurements or signals from the power system or message receiver blocks) and a set Δ of binary word q-tuples, where each word has k or fewer bits (i.e. a data queue):

$$\Gamma = \bigcup_{m=1}^{M} \{x_a | x_a \in \mathbb{Z}, log(|x|) + 1 \leq n\}^m \tag{3.1}$$

$$\Delta = \bigcup_{q=1}^{Q} \{x_b | x_b = a_1 a_2 ... a_k, a_i \in \{0, 1\}, 1 \leq i \leq k\}^q \tag{3.2}$$

  m and q are variables that change with the number of measurements/signals present and the number of messages in the queue respectively. M and Q are the largest possible values for m and q respectively.

The third set is the set Θ of natural numbers from 0 to F, where F is the maximum number of messages that the message generator can output to receiver blocks in a single output:

$$\Theta = \{x_c | x_c \in \mathbb{N}, x_c \leq F\} \tag{3.3}$$

X is defined as the union of the Cartesian product of all possible m-tuples and an *enqueue* label and the Cartesian product of all possible q-tuples and the set of natural numbers that can represent how many messages shall be output:

$$X = \Gamma \times enqueue \quad \bigcup \quad \Delta \times \Theta \tag{3.4}$$

- Y: The set of all possible outputs is defined in terms of three sets, and two labels, namely $toLTE$ and $toRXers$. The first set $\Lambda \subset \Delta$ represents all possible f-tuples of messages that can be forwarded to a receiver block:

$$\Lambda = \bigcup_{f=1}^{F} \{y_a | y_a = a_1 a_2 ... a_k, a_i \in \{0,1\}, 1 \leq i \leq k\}^f \tag{3.5}$$

The second set is all possible values that can represent a message generator's data queue, namely Δ as previously defined. The third set is the set Π of natural numbers from 1 to N, where N is the number of D2D-enabled smart grid nodes in the network:

$$\Pi = \{y_b | y_b \in \mathbb{N}, 1 \leq y_b \leq N\} \tag{3.6}$$

Y is defined as the union of the Cartesian product of Λ and $toRXers$ and the Cartesian product of Π, Δ, and $toLTE$:

$$Y = \Lambda \times toRXers \quad \bigcup \quad \Pi \times \Delta \times toLTE \tag{3.7}$$

- S: The state has four sub-states $s_{ied} \in S_{ied}$, $s_q \in S_q$, $s_{out} \in S_{out}$, and $s_{app} \in S_{app}$. The set of possible states is defined in terms of four sets. The first set consists of three sub-states that dictate whether the message generator is either idle in operation, enqueuing new data messages, or dequeuing data messages:

$$S_{ied} = \{idle, enqueuing, dequeuing\} \tag{3.8}$$

The second set is the set of possible queue states of the message generator, namely Δ:

$$S_q = \Delta \tag{3.9}$$

The third set is Π, the set of possible numbers of messages that can be forwarded in a single output:

$$S_{out} = \Pi \tag{3.10}$$

The fourth set is S_{app}, which is the set of all combinations of smart grid application-specific state variables that characterize the mapping from input signals and measurements to binary messages. Naturally, there is no general way to characterize S_{app}. S is defined as the Cartesian product of all defined sub-state sets:

$$S = S_{ied} \times S_q \times S_{out} \times S_{app} \tag{3.11}$$

- δ_{ext}: The external transition function is defined when the input sub-state $s_{ied} = idle$. This function also makes use of some function $runapp_{mode}(), mode \in \{0, 1\}$, which models the smart grid applications to be designed by power systems engineers, and can be any computer algorithm that accepts measurements/signals and returns a set of binary messages which are then enqueued for transmission while both taking into account and updating the state variables of S_{app} as required. For the ease of the characterization of δ_{ext}, a *mode* subscript is included for this function that allows one to choose whether $runapp_{mode}()$ returns the new state application-specific state variables, or the new queue state of the message generator. In reality, this function would do both simultaneously. $runapp_{mode}()$ can be characterized in general terms as a mapping from Cartesian product of X and S to Δ:

$$runapp_{mode} : \begin{cases} X \times S \mapsto \Delta, & mode = 0 \\ X \times S \mapsto S_{app}, & mode = 1 \end{cases} \tag{3.12}$$

$\delta_{ext}(s, x)$ is a piecewise function that either enqueues data messages or updates the queue state and the number of messages that are to be output as per the instructions of the abstract LTE network:

$$\delta_{ext}([idle, s_q, s_{out}, s_{app}], x) = \begin{cases} [enqueuing, runapp_0(x, s), 0, runapp_1(x, s)], & x_2 = enqueue \\ [dequeuing, x_1, x_2, runapp1(x, s)], & otherwise \end{cases} \tag{3.13}$$

- δ_{int}: The internal transition sets s_{ied} to $idle$, removes the first s_{out} messages (i.e. any messages that may have been output from the queue by λ), sets s_{out} to 0, and leaves s_{app} unchanged:

$$\delta_{int}(s) = [idle, s_{q-D}, 0, s_{app}], \tag{3.14}$$

  $D = \{1, 2, ..., s_{out}\}$ is a set of indices of queue s_q, and s_{q-D} denotes the vector containing the set of elements of s_q (i.e. the set of binary messages of s_q) that do not have the indices present in index set D.

- λ: The output function either returns a vector whose elements consist of the node identifier $id \in \Pi$ of the message generator; the queue of the message generator; and a $toLTE$ label, or a vector whose elements consist of a vector of output binary messages and a $toRXers$ label:

$$\lambda(s) = \begin{cases} [id, s_q, toLTE], & s_{ied} = enqueuing \\ [s_{qD}, toRXers], & s_{ied} = dequeuing \end{cases} \tag{3.15}$$

  $D = \{1, 2, ..., s_{out}\}$ is a set of indices of queue s_q, and s_{qD} is the vector containing the elements of s_q that have the indices present in index set D.

- t_a: The time advance function is defined as follows:

$$t_a(s) = \begin{cases} \infty, & s_{ied} = idle, \\ 0, & otherwise \end{cases} \tag{3.16}$$

  This way, the message generator waits in an idle state until an input arrives, upon which it will instantaneously call the output function λ.

3.2.2.2 Abstract LTE Network Atomic Model

This atomic model treats the discrete advances in time of the power system simulator as event inputs in order to synchronize the timing of events in the ICT side of the simulator to the time advancement of the power system dynamics. Moreover, this atomic model dictates how long the queuing delay will be for each message moving from one device to another, and whether or not a message will make it to the receiver, or be lost due to communication failure.

In order for the LTE network simulator to have access to the messages being generated by the smart grid applications, this atomic model also accepts queue information updates

from the message generator atomic models as inputs. The DEVS tuple is characterized as follows:

- X: The set of all input events are the same, namely a the union of the subframe advance event, and the set of possible outputs from the message generator atomic model:

$$X = advance\_subframe \bigcup \Lambda \times toRXers \bigcup \Pi \times \Delta \times toLTE \tag{3.17}$$

- Y: There are N outputs to this atomic model, each one coupling to each of the N message generators implemented in the simulation. As such, Y can be modeled as a set of N-dimensional ordered pairs, where each element is either a two dimensional vector, or a "null" output. The two dimensional vector space is the Cartesian product of the set of possible queue states a message generator atomic model can have and the number of messages that should be forwarded at the time of the current subframe advance event. The "null" element is used for particular queues if they are empty, in which case no output is required:

$$Y = (\Delta \times \Theta \bigcup null)^N \tag{3.18}$$

- S: The state has four sub-states $s_t \in S_t$, $s_{iju} \in S_{iju}$, $s_Q \in S_Q$, and $s_{LTE} \in S_{LTE}$. s_t is the number of $advance\_subframe$ inputs received since the beginning of the simulation, representing the time of simulation (for both the power and communications systems) at the resolution of milliseconds. As such, S_t is the set of natural numbers ranging from 0 to the time duration of the simulation in milliseconds t_{sim}:

$$S_t = \{s_t | s_t \in \mathbb{N}, 0 \leq s_t \leq t_{sim}\} \tag{3.19}$$

S_{iju} consists of "idle", "judgment", and "queue update" states:

$$S_{iju} = \{idle, judgment, queue\_update\} \tag{3.20}$$

s_Q is a N-tuple containing the states of all message generator atomic model queues in the simulation, which includes both the message queues and the number of messages to be forwarded upon the advancement of a subframe. Such information is needed to determine the simulation trace of the LTE network simulator, which in turn determines when messages can be forwarded to receiving smart grid application nodes. S_Q is defined as follows:

$$S_Q = (\Delta \times \Theta)^N \tag{3.21}$$

Finally, S_{LTE} is the set of all combinations of LTE network simulator-specific state variables that characterize its discrete event dynamics. S is defined as the Cartesian product of S_t, S_{iju}, S_Q, and S_{LTE}:

$$S = S_t \times S_{iju} \times S_Q \times S_{LTE} \tag{3.22}$$

- δ_{ext}: The external transition function increments s_t, changes s_{iju} to $judgment$, and updates the state of all message generator queues if the input is a subframe advance event. The message generator queues and LTE simulator state variables get updated by the LTE network simulator, which also has access to the current queue states of all the message generators. This functionality is modeled by a generic function $runLTE_{mode}()$, which maps the elements of the Cartesian product of X and S to the set of queue state N-tuples, i.e. S_Q if $mode = 0$; if $mode = 1$, $X \times S$ is mapped to S_{LTE} instead:

$$runLTE_{mode} : \begin{cases} X \times S \mapsto S_Q, & mode = 0 \\ X \times S \mapsto S_{LTE}, & mode = 1 \end{cases} \tag{3.23}$$

The design of $runLTE_{mode}()$ is independent of the co-simulation interface characterized in this section, and is in reality instantiated by the ICT simulation unit chosen for the co-simulator.

In the case that the input is a queue information update from one of the message generators, the external transition function changes s_{iju} to $queue\_update$, and replaces the i^{th} element of s_Q. Thus, the characterization of δ_{ext} is as follows:

$$\delta_{ext}([s_t, idle, s_Q, s_{LTE}], x) = \begin{cases} [s_t + 1, judgment, runLTE_0(X,S), runLTE_1(X,S)], \\ \quad x = advance\_subframe \\ [s_t, queue\_update, s_Q^{'}, s_{LTE}], \quad x_3 = toLTE \end{cases} \tag{3.24}$$

where

$$s_Q^{'} = (s_{Q1}, s_{Q2}, ..., s_{Qi}, ..., s_{QN}), \quad i = x_1 \tag{3.25}$$

- δ_{int}: The internal transition function changes s_{iju} to "idle":

$$\delta_{int}([s_t, s_{iju}, s_Q, s_{LTE}]) = [s_t, idle, s_Q, s_{LTE}] \tag{3.26}$$

- λ: The output function is defined for when $s_{iju} = judgment$, and outputs s_Q:

$$\lambda([s_t, judgement, s_Q, s_{LTE}]) = s_Q \tag{3.27}$$

- t_a: The time advance function is defined as follows:

$$t_a([s_t, s_{iju}, s_Q, s_{LTE}]) = \begin{cases} \infty, & s_{iju} = idle, \\ 0, & otherwise \end{cases} \tag{3.28}$$

  This way, the LTE network increments its state and dictates the advancement, delay, or destruction of messages every subframe at the same instant the power system advances a millisecond.

3.2.2.3 Message Receiver Atomic Model

The message receiver atomic model accepts sets of one or more binary messages as inputs, and generates appropriate numerical signals to their associated generator entity. The DEVS tuple is characterized as follows:

- X: The set of all possible inputs consists of the set of all possible outputs from a message generator atomic model (although the inputs with $toLTE$ tags are ignored):

$$X = \Lambda \times toRXers \bigcup \Pi \times \Delta \times toLTE \tag{3.29}$$

- Y: The set of possible outputs consists of the Cartesian product of the set of all m-tuples of integers with n or fewer digits, where $m \leq M$ and an *enqueue* tag, which is exactly $\Gamma \times enqueue$:

$$Y = \Gamma \times enqueue \tag{3.30}$$

- S: The state has two sub-states $s_{ia} \in S_{ia}$ and $s_{app} \in S_{app}$. S_{ia} consists of *idle* and *active* states:

$$S_{ia} = \{idle, active\} \tag{3.31}$$

S_{app} is the collection of all possible combinations of smart grid application specific state variables that are used to determine the appropriate mapping from input messages to output signals to be sent to the message generation atomic model. Naturally, there is no general way to characterize S_{app}. S is defined as the Cartesian product of S_{ia} and S_{app}:

$$S = S_{ia} \times S_{app} \tag{3.32}$$

- δ_{ext}: The external transition function is defined when the input sub-state $s_{ia} = idle$, and the second element of $x = toRXers$. When the message receiver atomic model receives an input, s_{ia} gets set to $active$, and the application specific state variables are updated based on the input and current state of the message receiver. Updating the application specific state variables in reality is done by algorithms coded by the smart grid application developer, and the process varies from application to application. However, they can all be modeled in general as some function $updateVariables()$ that maps the elements of X and S_{app} to the elements of S_{app}:

$$updateVariables : X \times S_{app} \mapsto S_{app} \tag{3.33}$$

By making use of $updateVariables$, the external transition function can be defined as follows:

$$\delta_{ext}([idle, s_{app}], x) = [active, updateVariables(s_{app}, x)], \quad x_2 = toRXers \tag{3.34}$$

- δ_{int}: The internal transition function changes s_{ia} from $active$ to $idle$:

$$\delta_{int}([active, s_{app}]) = [idle, s_{app}] \tag{3.35}$$

- λ: The external transition function is up to the smart grid application developer to design, as it specifies the mapping from the state variables of the smart grid application to the set of possible outputs to be sent to the smart grid device's message generator:

$$\lambda([active, s_{app}]) : S_{app} \mapsto Y \tag{3.36}$$

- t_a: The time advance function is defined as follows:

$$t_a([s_{ia}, s_{app}]) = \begin{cases} \infty, & s_{ia} = idle, \\ 0, & otherwise \end{cases} \tag{3.37}$$

This way, the message generator waits in an idle state until an input arrives, upon which it will instantaneously call the output function λ.

3.2.3 Remark on Generalization of Co-simulation Interface

Future practitioners can refer to the equations in this section as a guide to implementing their own custom co-simulation of some continuous time system coupled with a communication network that uses a time slot based medium access control scheme. For example, any CT simulation unit that can be wrapped in the augmented DEVS tuple presented in Equation (3) of Nutaro, et al. [66] can have its inputs and outputs seamlessly connected to the outputs and inputs of the DEVS tuples constructed in this paper.

3.3 Microgrid Simulation: Fault Location, Isolation, and Restoration (FLISR) Application

To test the simulator, a fault location, isolation, and restoration (FLISR) application was developed using the simulator. As depicted in fig. 3.4, we model a 1 km, 5 kV loop distribution feeder with three 5 kW spot loads separated by "smart switches" that relays information to a main feeder circuit breaker (CB). Let the smart switch between Loads 1 and 2 be named Switch A, and the one between Loads 2 and 3 be named Switch B. Also, let the smart switch between Load 3 and the backup feeder (supply #2) be named the Tie Switch. The breaker can exchange information from these switches and perform FLISR operations, ultimately facilitating remote control over the actuation of such switches to reconfigure the feeder topology in response to faults occurring. For example, fig. 3.4 shows that in the event of a fault occurring at Load 2, the FLISR application would reconfigure the circuit topology to disconnect the portion of the power line in which the fault is located in a manner that keeps as many loads energized as possible.

Fig. 3.5 depicts the messages that the various smart grid devices need to send to one another. These messages have been numbered on the left-hand side of the diagram so they can be referenced by the labeling in fig. 3.7d. When the fault is detected by the circuit breaker (CB), it opens to de-energize the feeder, and queries Switches A and B to see if they detected the fault. Based on the response of the switches, the CB can then deduce the segment of line in which the fault occurred. Based on where the fault occurred, the CB then sends appropriate actuation commands to Switches A and B in addition to the Tie Switch. In order to assure the line is only re-energized after the fault has been isolated, Switches A and B must execute their actuation commands first, then

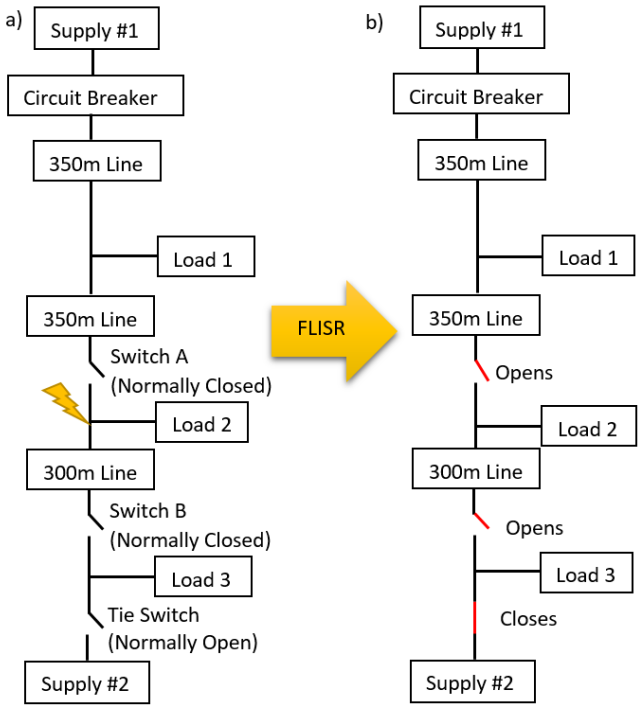

FIGURE 3.4: Depiction of loop feeder topology and its switch configurations in a) pre-fault event and b) post-fault event states.

send an acknowledgement (ACK) message to the CB and Tie Switch to notify them that it is safe to carry out their actuation commands.

3.4 Results

The FLISR application defined in Section 3.3 was put through a series of tests by applying single phase to ground faults at the various loads of the aforementioned power feeder topology, and then observing the resultant power system waveforms and communication messaging. Further tests were conducted to observe the simulation traces produced by the co-simulator when communication link failures occurred between the various devices of the FLISR application.

In the LTE simulator, a 5 MHz bandwidth Scheduling Access Period was used, within which contains the sidelink control and data resource pools used for device-to-device communication. The transmission power and antenna gains of each device in the network were set to 20 dBm and 10 dB respectively. Concerning fundamental large scale fading parameters, the simulator's path loss model that characterizes the frequency dependent attenuation of a propagating signal as a function of distance was set to a commonly accepted model published in section 5.3.2.2.2 of a Third Generation Partnership Project (3GPP) technical report [85]. Moreover, the penetration loss and noise (present in the

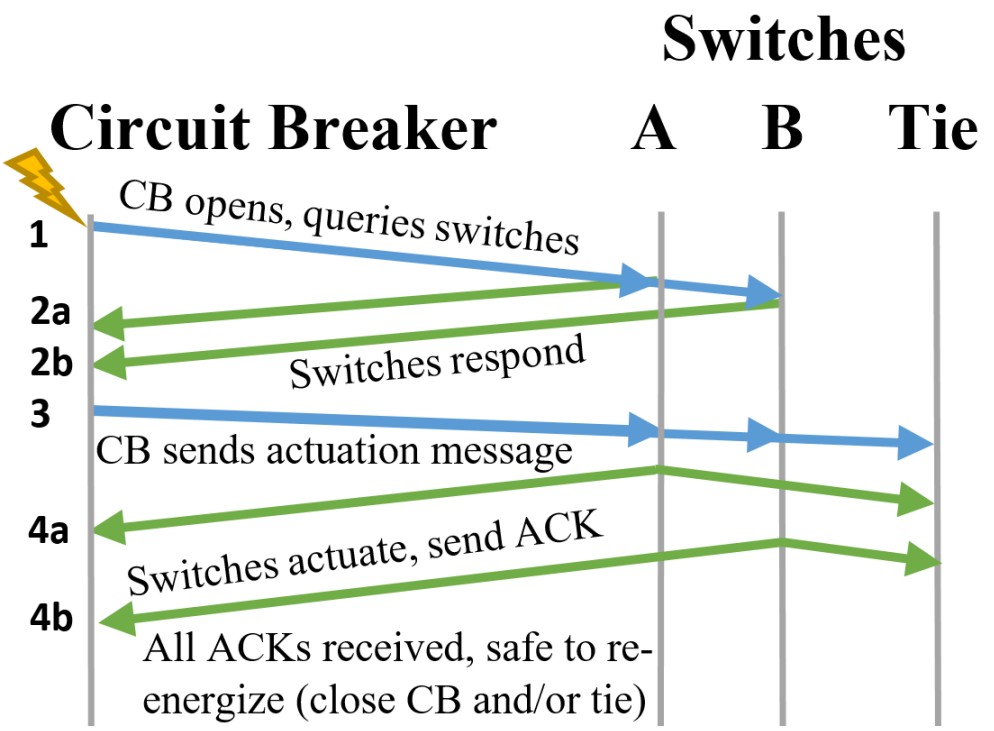

FIGURE 3.5: FLISR application message sequence timing diagram.

communication devices' electronics) was set to 5 dB. Finally, shadowing was modeled as a zero mean Gaussian random variable with a 2 dB variance [86].

Figures 3.6, 3.7, and 3.8 visualize the power and ICT system dynamics of the FLISR application handling faults occurring at Loads 1, 2, and 3 on the distribution feeder respectively. Figures 3.9, 3.10, and 3.11 observe the impact of the joint power system ICT dynamics when communication failures occur between the CB and Switches A and B; the CB and the Tie Switch; and the Switches A and B and the Tie Switch respectively.

As can be seen in figures 3.6d, 3.7d, and 3.8d, the message sequencing is consistent with the FLISR application model shown in fig. 3.5; the changes in the relay signals occur immediately after actuation commands have been received by Switches A and B, as well as after the ACK messages have been received by the CB and Tie Switch.

In fig. 3.6a, one can see how a fault occurring at Load 1 results in the substation egress feeder shutting off its current supply indefinitely. This is to be expected, as the feeder breaker and Switch A must open in order to isolate a fault at Load 1. Moreover, in fig. 3.6b, Loads 2 and 3 get restored by the backup feeder, while Load 1 remains isolated from the power system. This is also expected, as the FLISR application was designed to close the tie switch, re-close Switch 2, and open the CB and Switch 1 in this particular situation, which is observed to be the case in fig. 3.6c.

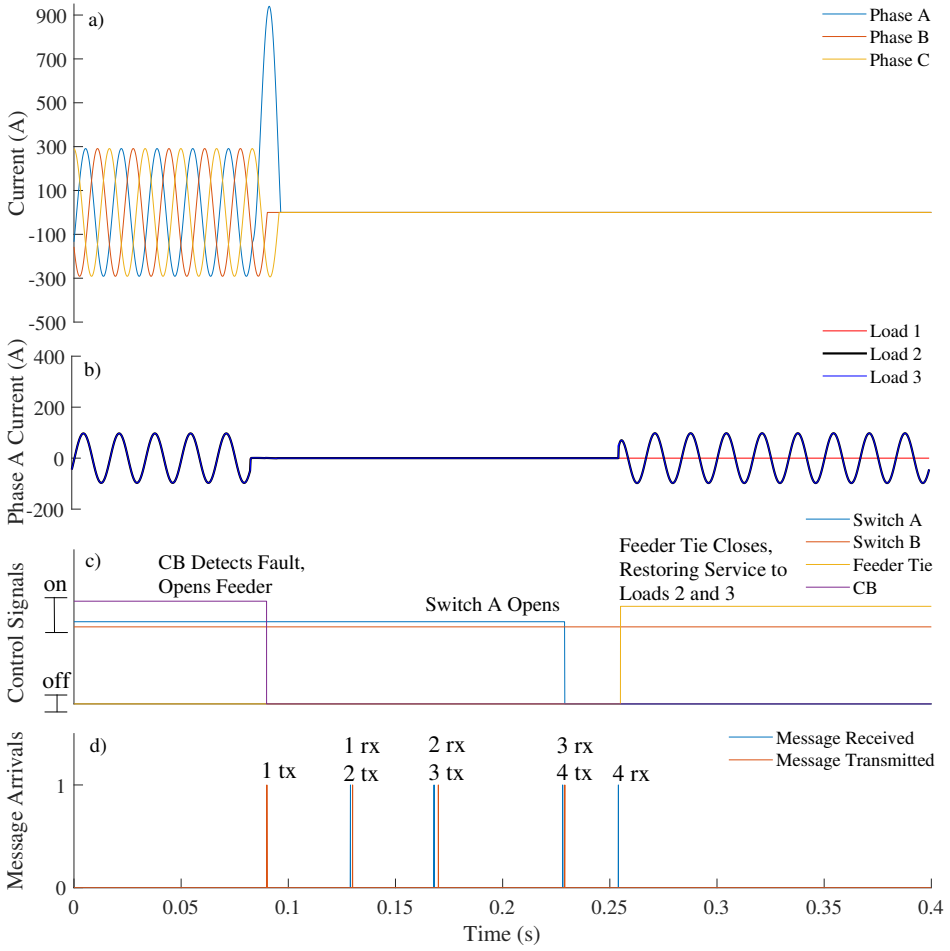

FIGURE 3.6: Co-simulation traces for FLISR application when a fault occurs at Load 1. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

In fig. 3.7a, one can see how a fault occurring at Load 2 results in the substation egress feeder shutting off its current supply while the FLISR application determines how to safely restore power to as many loads as possible while isolating the fault, then gets turned back on. The peak egress current is a third of its original value upon restoration. Such a result is as a consequence of Load 2 being isolated from the system, and Load 3 being restored by the backup feeder in lieu of its original feeder, making Load 1 the only remaining demand left on the original feeder. Fig. 3.7b is consistent with the explanation of fig. 3.7a, as Load 2's current output remains zero after the FLISR event, while Loads 1 and 3 get restored. The control signaling in fig. 3.7c is also consistent with the other sub plots, as it shows Switches A and B opening (i.e. their control signals lowering), the Tie Switch closing, and the CB re-closing.

In fig. 3.8a, one can see how a fault occurring at Load 3 results in the substation

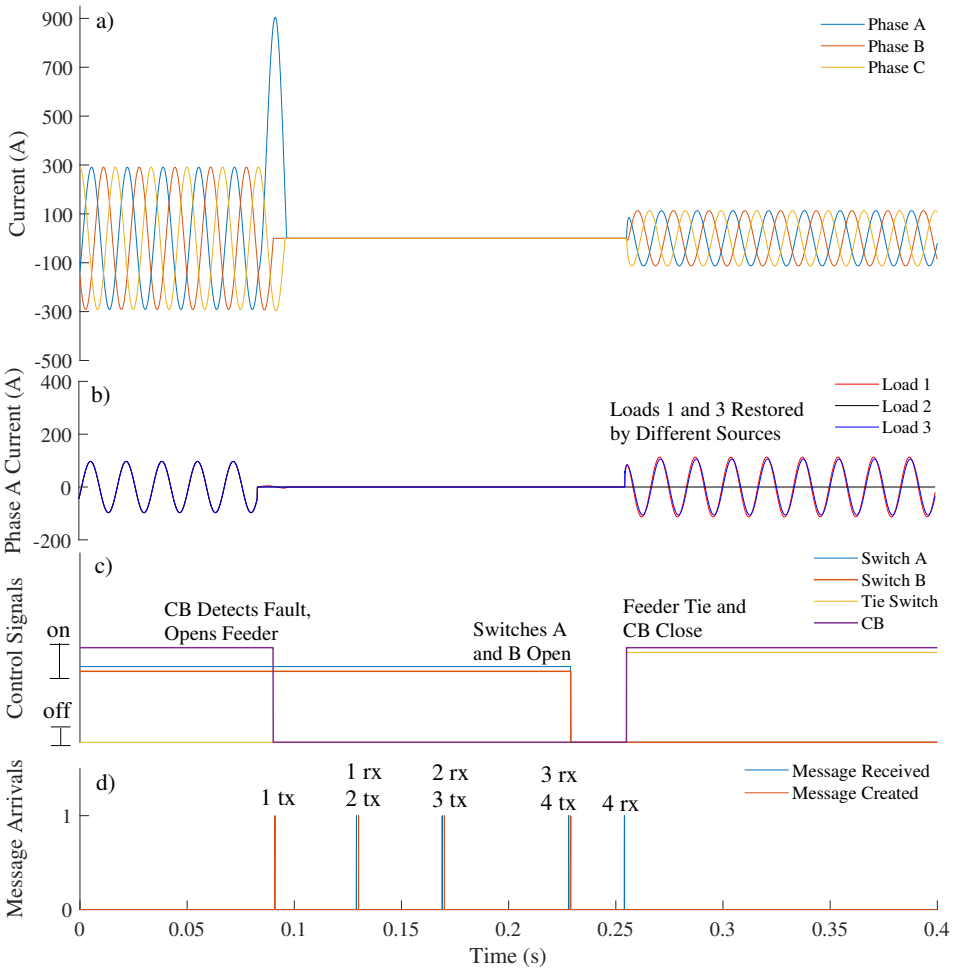

FIGURE 3.7: Co-simulation traces for FLISR application when a fault occurs at Load 2. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

egress feeder shutting off its current supply while the FLISR application determines how to safely restore power to as many loads as possible while isolating the fault, then gets turned back on. The peak egress current is two thirds of its original value upon restoration. Such a result is as a consequence of Load 3 being isolated from the system, while Loads 1 and 2 are able to be restored by the original feeder since the fault is downstream of these loads. Fig. 3.8b is consistent with the explanation of fig. 3.8a, as Load 3's current output remains zero after the FLISR event, while Loads 1 and 2 get restored. The control signaling in fig. 3.8c is also consistent with the other sub plots, as it shows Switch B opening, the Tie Switch remaining open, and Switch A as well as the CB re-closing.

In fig. 3.9, a FLISR application is tested when there is a single line (phase A) to ground fault occurring at Load 2. However, for this test, a communication failure between the

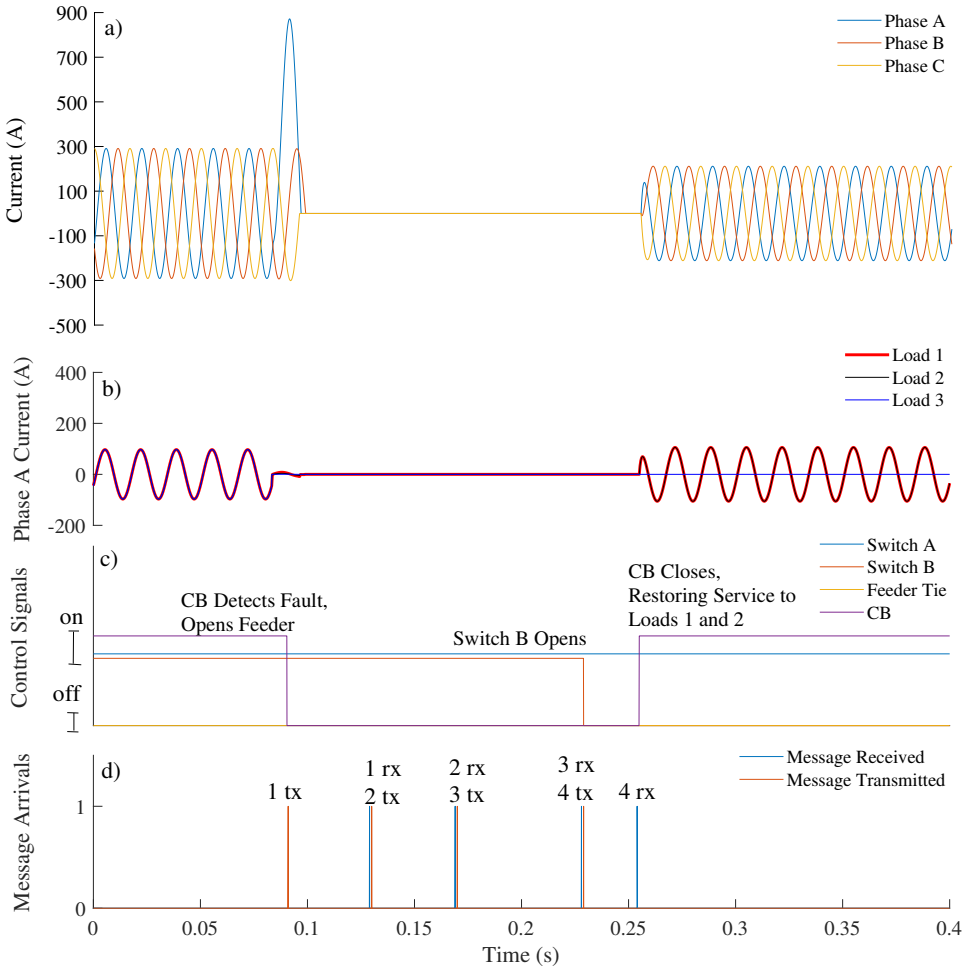

FIGURE 3.8: Co-simulation traces for FLISR application when a fault occurs at Load 3. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

CB and Switches A and B was simulated as well to observe its impact on the performance of the FLISR application. To see the difference that the simulated communication failure made in the power system's simulation trace, fig. 3.9 can be compared to fig. 3.7. One can see that the communication link failure between the CB and Switches A and B results in a complete failure of the FLISR application as expected. Aside from the main feeder breaker shutting off due to the local detection of high current (which does not require any communication messaging), no other control signals change in fig. 3.9c, which is consistent with the absence of communication network activity observed in fig. 3.9d. Because the CB cannot initiate the FLISR application, clearly nothing happens, and none of the loads are restored in 3.9b.

In fig. 3.10, the FLISR application is being tested for a fault occurring at Load 1 while there is a communication failure between the CB and Tie Switch. One can see that

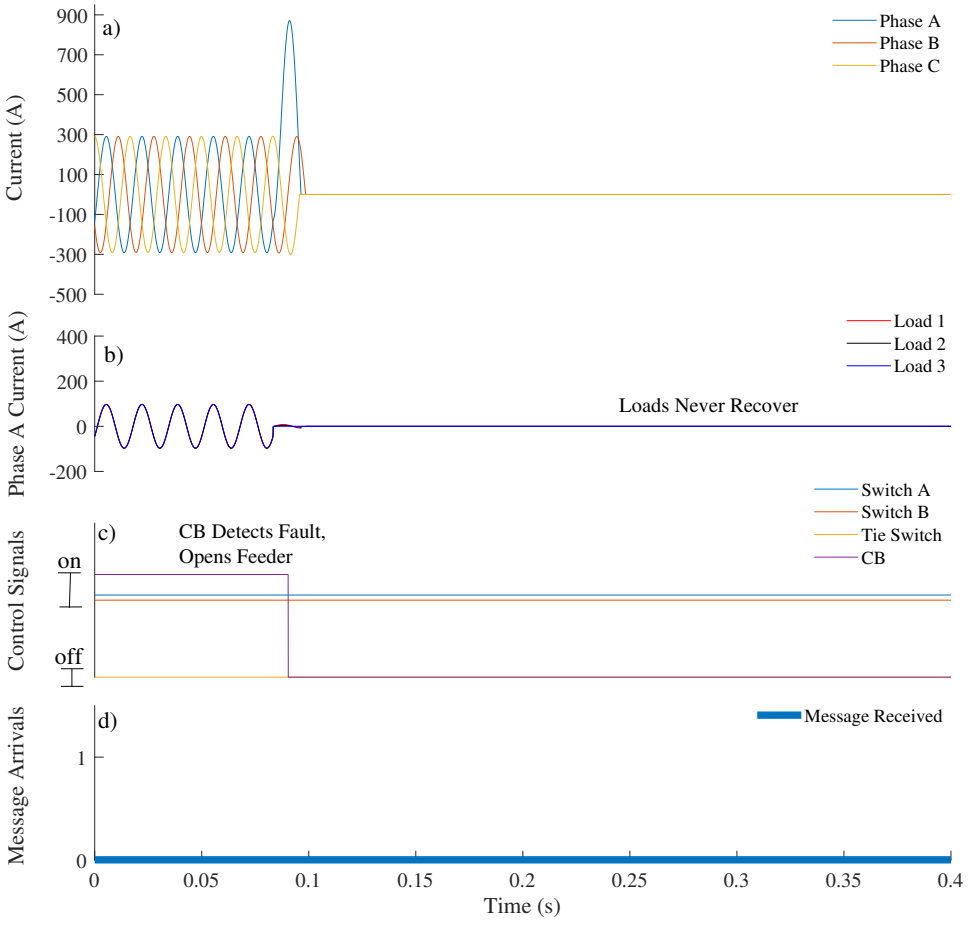

FIGURE 3.9: Co-simulation traces for FLISR application when a fault occurs at Load 2, but a communication failure occurs between the CB and Switches A and B. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

the communication link failure between the CB and the Tie Switch results in a partial failure of the FLISR application as expected. Despite the control signals of Switches 1 and 2 working properly in fig. 3.10c, the Tie Switch fails to close due to communication failure, which is consistent with the communication network activity observed in fig. 3.10d. Because the Tie Switch never hears the actuation command sent by the CB, the Tie Switch never closes to connect Loads 2 and 3 to the backup feeder, and thus none of the loads are restored in 3.10b.

In fig. 3.11, the FLISR application is being tested for a fault occurring at Load 2 while there is a communication failure between Switches A/B and the Tie Switch. One can see that the communication link failure between Switches A and B and the Tie Switch results in a partial failure of the FLISR application as expected. Despite the control signals of Switches 1 and 2 working properly in fig. 3.11c, the Tie Switch fails to

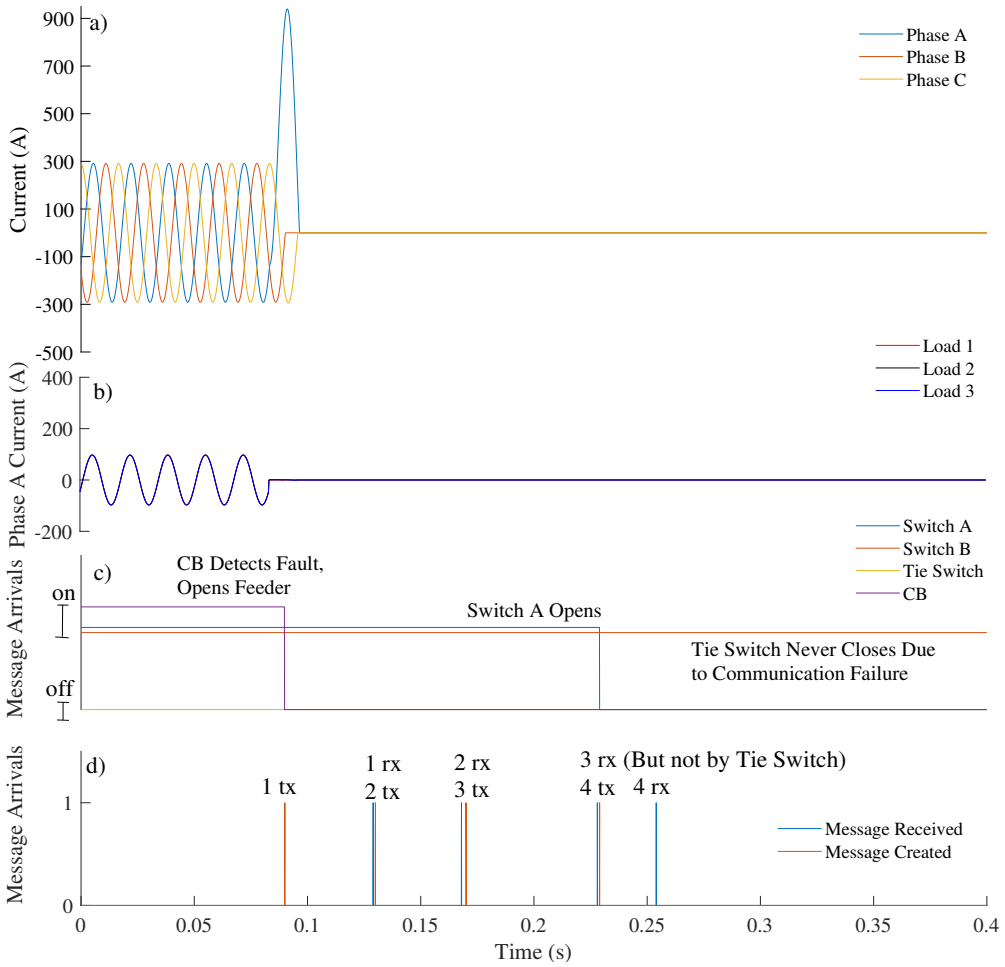

FIGURE 3.10: Co-simulation traces for FLISR application when a fault occurs at Load 1, but a communication failure occurs between the CB and the Tie Switch. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

close due to communication failure, which is consistent with the communication network activity observed in fig. 3.11d. Because the Tie Switch never hears the acknowledgement messages sent by Switches A and B, the Tie Switch never closes to connect Load 3 to the backup feeder, and thus only Load 1 is restored in 3.11b.

For further simulation testing, fig. 3.12 plots the signal to interference plus noise ratio (SINR) as a function of antenna gain (for transmission and reception) for the D2D connections established between the feeder breaker (CB) and the smart switches. As per fig. 3.4, Switch A is closest to the CB, followed by Switch B, then the Tie Switch. This spatial distribution of the smart grid devices is thus consistent with the data in fig. 3.12, as higher SINRs are achieved using the same antenna gain for communication links that are closer together. Moreover, the SINR values at which the messages start

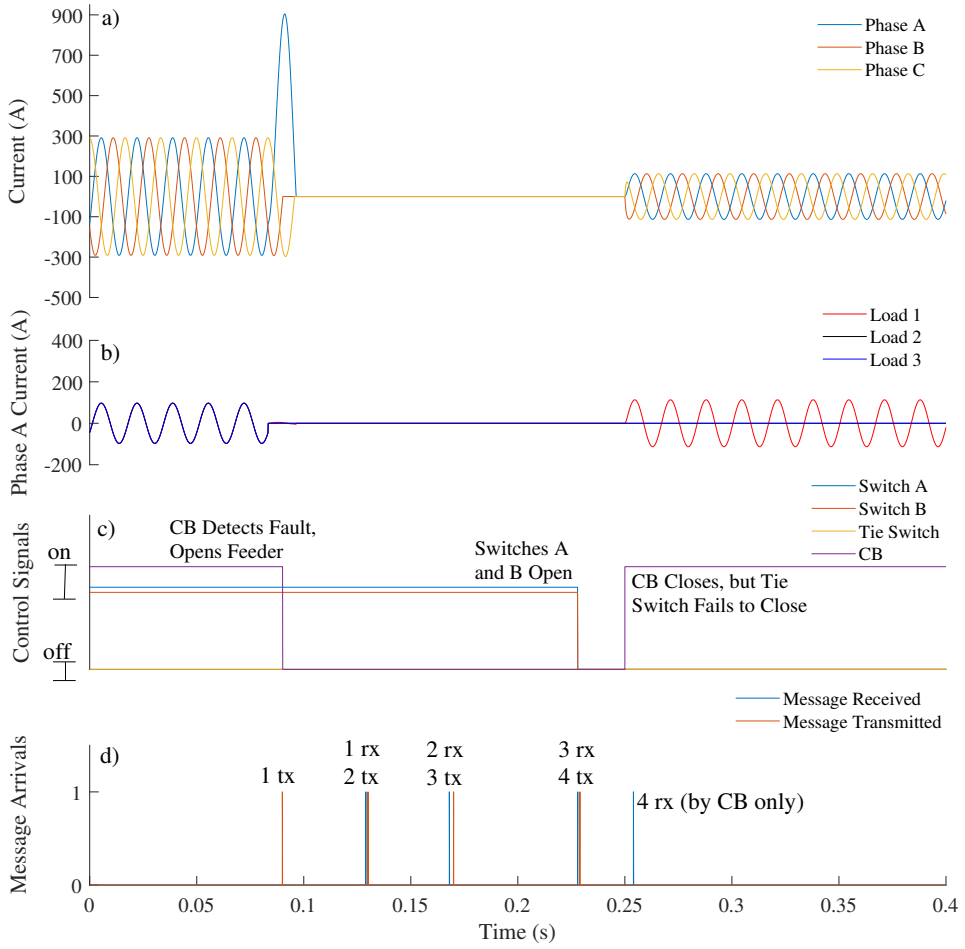

FIGURE 3.11: Co-simulation traces for FLISR application when a fault occurs at Load 2, but a communication failure occurs between the Tie Switch and Switches A and B. a) Is the three phase substation egress current; b) is the phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. c) Shows the control signals of all the switches and main breaker of the feeder. d) Shows events where a message has been received by a smart grid device.

being lost are similar to those that correspond to a rise in block-error ratio observed in the example sidelink throughput conformance testing script created by Mathworks [87].

FLISR applications in general require very little communication resources on the data plane, as their message sizes can be made on the order of hundreds of bytes or lower, and are infrequently transmitted. Thus, the test results presented in this paper do not investigate scenarios of high network congestion. However, for analyzing the impacts of using different LTE cellular resource allocation algorithms on the packet drop rates observed in larger scale networks that push the network into high congestion scenarios, one can refer to the results of Chapter 4 of this thesis.

In summary, this chapter introduced a way to combine existing simulation software packages available in the Matlab/Simulink environment to create a self-contained LTE D2D

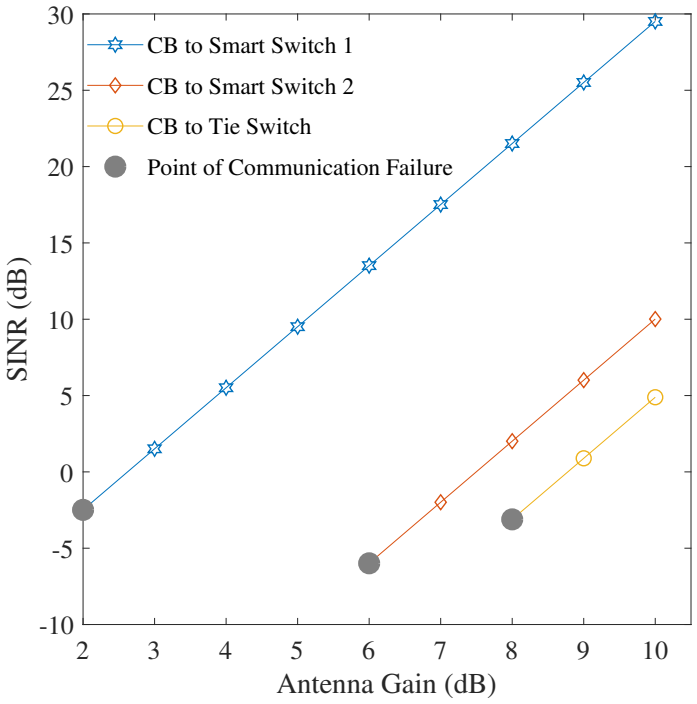

FIGURE 3.12: SINR plotted as a function of antenna gain for the D2D connections established between the feeder breaker (CB) and the smart switches.

enabled smart grid simulator with detailed ICT and power system dynamics alike. A discrete-event simulation based interface was mathematically characterized and implemented to facilitate our simulation. The simulator was tested by using it to simulate a FLISR application on a simple loop feeder for various fault locations and communication failure scenarios.

Chapter 4

High Reliability Q-Learning Scheduling for D2D Microgrid Communications

4.1 Introduction

To make smart grid solutions economically feasible, there is interest in implementing mobile wireless communication technology, i.e. Fourth Generation Long Term Evolution (4G LTE) and in the coming years Fifth Generation New Radio (5G NR) to establish a network of communication links over a distribution system with minimal investment in physical infrastructure [1–3]. Nevertheless, the data transfer delay (latency) in existing mobile communication technology is not guaranteed to be satisfactory for latency-critical smart grid services (e.g. synchrophasor applications), and addressing this problem is an ongoing area of research [3–5].

Device-to-Device (D2D) communication is a promising means to improve communication performance at a neighbourhood area network scale by allowing direct data exchange between users, which in certain transmission modes can be controlled directly by such users [6, 7]. Moreover, such transmission modes also have the benefit of being usable even when devices are not within the coverage of a cell tower. Among the transmission modes that allow users to self-allocate resources are Transmission Mode 2 (TM-2) introduced in LTE Release 12, and TM-4 introduced in LTE Release 14. TM-2 was originally designed for public safety applications and prioritized prolonging battery life [24], but requires that D2D agents self-allocate cellular resources in a random manner and thus is not currently suitable for high reliability and low latency communications. On the other hand, TM-4 was designed for high reliability and low latency communication

in addition to being able to manage fast-moving devices to facilitate vehicle-to-vehicle (V2V) communications, but disregards battery life considerations [24]. As such, neither transmission mode has been designed as an optimal smart grid communications solution. It can be argued that based on the communication requirements of smart grid applications, some combination of elements from both transmission modes would be best. We propose that the resource allocation mechanism of TM-2 be upgraded to meet the quality of service (QoS) requirements of smart grid applications in order to have a less complex and more power-efficient solution to D2D-enabled smart grid communications than TM-4.

A critical issue with TM-2 that impedes its communication performance is how it randomly self-allocates cellular resources, which leads to high packet drop rates (PDR), making it inadequate for certain smart grid applications sensitive to information losses. For example, a typical frequency stability synchrophasor application is sensitive to message failure rates exceeding 0.33% based off data loss sensitivity metrics published by the North American Synchrophasor Institute [25]. To address this problem, a multi-agent high-reliability Q-learning (HRQ) resource allocation scheme is introduced in this chapter to replace the random allocation mechanism. Naturally, HRQ is powered by Q-learning, which is a decision-making algorithm that considers the situation it is in, and chooses the best action to take based on past experience. Formally, that which takes the action is referred to as the "agent", and this agent interacts with its "environment" by taking "actions". The agent's situation is considered the "state" of the environment, and there are a finite set of actions that can be taken in any given state. The best action for a given state is the one with the largest expected "reward" associated with it, which is based on an action's influence on the environment [26]. HRQ divides the LTE resource grid into orthogonal partitions of resources from which a scheduling agent can select, and the agent is rewarded or penalized depending on whether or not it takes the same scheduling action as one or more other agent(s).

Concerning existing works on distributed resource allocation algorithms, few integrate the specifications of the LTE D2D TM-2 standard into their solution. Seemingly only Shih *et al.* [43] does so, offering a resource allocation algorithm to reduce packet drop rates. HRQ alternatively does not require randomly altering the operation of the D2D transmitter at the physical layer to sense control channel messages of other transmitting nodes in order to function. Moreover, because Shih *et al.* rely on being able to locally sense the interference of other D2D transmitters in the network, their approach is susceptible to the classic "hidden terminal problem" faced by wireless local area networks. Other works either do not take into account the need to be implementable under LTE protocol architecture [79], have a centralized allocation scheme that is not applicable in out of coverage scenarios [45, 88], or focus instead on using D2D communications for

frequency reuse in order to boost channel throughput instead of minimizing PDR for the D2D nodes [42, 44, 54, 55, 89–91].

The rest of the chapter is organized as follows. Section 4.2 presents the system and traffic models, as well as the problem formulation. In Sections 4.3 and 4.4, the HRQ and LTE random resource allocation schemes are presented respectively. Section 4.5 contains the simulation results that show how the HRQ-enabled agents can achieve orthogonal self-organization of resources, causing lower latency and zero packet drop rate for low to moderate network traffic.

4.2 System Model

4.2.1 Microgrid Communication Network Model

Fig. 4.1 was produced by Medhat Elsayed, published in our work [92]. Fig. 4.1a visualizes the communication network topology modeled in this study. A set $\aleph$ of D2D nodes is considered where each D2D node is $i \in \aleph$. A subset of D2D nodes $\Re \subset \aleph$ represent microgrid devices that have certain quality of service requirements. In addition, a set $\Im$ of auxiliary D2D nodes, where $\Im \subset \aleph$ are considered in order to overload the network with traffic and assess the performance of our algorithm. This auxiliary traffic is stochastic with message generation rates following a Poisson process, and message payloads are generated with exponential random variables. Cellular nodes are not included in the model, and instead an out of network coverage scenario is considered. However, interference among D2D nodes remains due to the distributed and uncooperative resource allocation process.

Fig. 4.1b-d depicts the data plane architecture of the LTE TM-2 standard used in this study. As can be seen in fig. 4.1b and fig. 4.1d, the lowest 3 layers of the LTE protocol stack has been modeled. The radio link control (RLC) entities of the D2D devices have been modeled to operate in unacknowledged mode as per TM-2 specifications [36]. At the physical layer, TM-2 D2D communication channels, i.e. sidelink channels, are organized into repeating segments of the LTE resource grid known as scheduling access periods (fig. 4.1c). It should be noted that a "scheduling access period" is the exact same thing as a "PSCCH period" as it was described in Chapter 2. For a lengthy description of the implementation and joint ICT-power system dynamics of our network model, consult Chapter 3.

The scheduling access period architecture is further partitioned into a sidelink control channel, and a sidelink data channel. The sidelink control channel precedes the sidelink

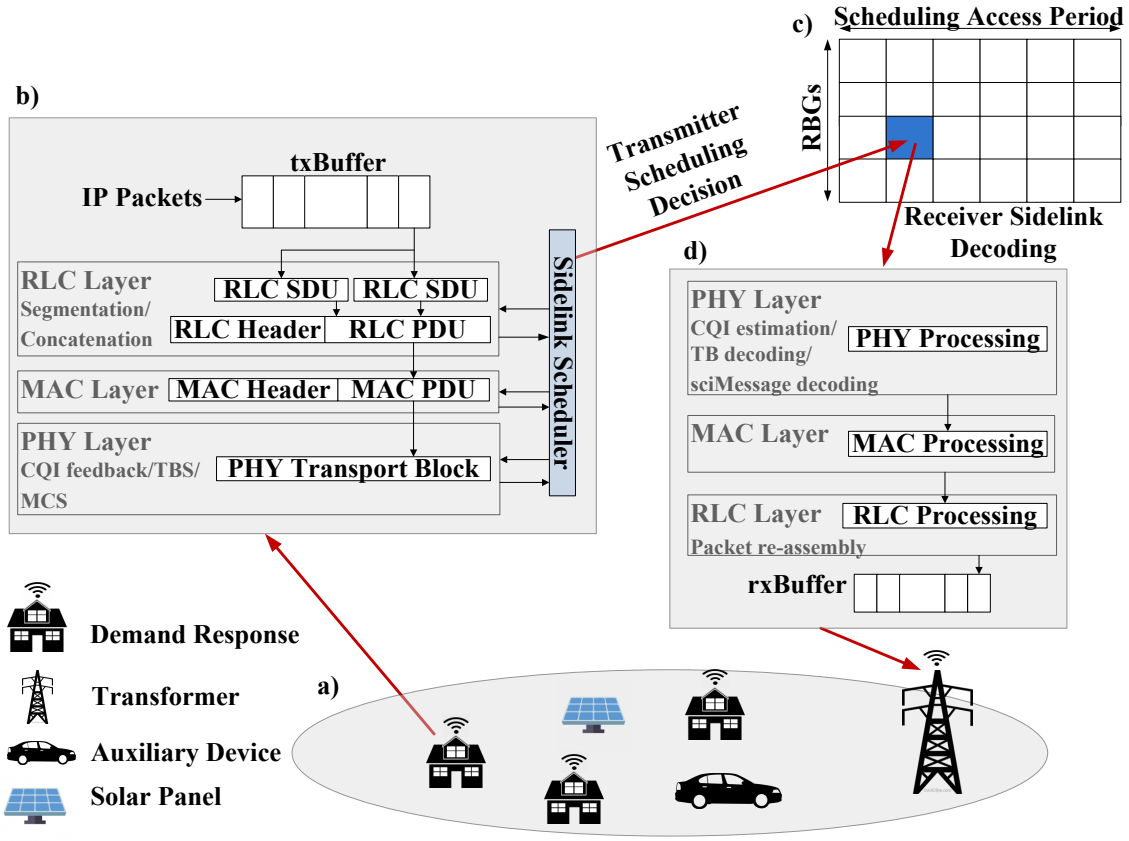

FIGURE 4.1: Network model of 3GPP LTE-compliant Release-12 D2D TM-2 communication. Concerning the acronyms in this figure, MAC is medium access control, PHY is physical, SDU is service data unit, PDU is protocol data unit, CQI is channel quality indicator, RBGs is resource block groups, MCS is modulation and coding scheme, tx is transmission, and rx is reception.

data channel in time, and D2D nodes make scheduling decisions once per scheduling access period, encoding a sidelink control information (SCI) message that the receiver can use to determine which resources to listen to in the upcoming sidelink data channel. The physical sidelink control channel, or PSCCH, is two subframes long. Two copies of SCI messages are transmitted whenever a D2D node transmits information, one on each subframe of the PSCCH. The physical resource blocks used for both SCI messages are chosen randomly to minimize chances of interference. Moreover, SCI messages are always sent at the most robust modulation and coding scheme that the LTE standard has available, i.e. QPSK with the largest amount of coding redundancy available. However, for this study, the focus is to observe the impact of how our resource allocation algorithm that controls the data channel resources has on the QoS of smart grid applications. Improving the allocation of control plane resources is not in the scope of the thesis, and as such, due to the small amount resources that the SCI messages occupy; the fact that they are transmitted twice; and that they always adhere to a very robust MCS, it is assumed that SCI messages are always successfully received by the D2D receivers.

TABLE 4.1: D2D Traffic Settings and QoS Requirements.

D2D application	Inter-arrival time [ms]	Packet size [b]	Max latency [ms]	Max PDR [%]
DR	20	2000	500-minutes	1-9
Solar	20	1120	300-2000	0
PMU	16	592	50	0.33
Auxiliary	20	2000	N/A	N/A

4.2.2 Traffic Model of the Microgrid Applications

As presented in Table 4.1, the traffic characteristics of three different microgrid applications, namely demand response, solar generation forecasting, and phasor management unit (PMU) communications have been modeled. This table also includes some typical QoS requirements that could be expected for the aforementioned application archetypes. For the solar panel and demand response traffic, all parameters except their tolerable PDRs were derived based on the communication requirements specified by the US Department of Energy [4, 93]. For DR applications, it is assumed that 1-9% PDR is manageable based on Kong's analysis of dynamic pricing applications [94]. For distributed energy resource generation forecasting communication, it has been assumed that PDR should be close to 0 [95]. Concerning the PMU traffic model, Table 4.1 was populated using the PMU applications requirements published by the North American SynchroPhasor Institute (NAPSI) [25] in addition to the IEEE C37.118.2 standard for PMU-generated synchrophasor data transmission [96, 97].

4.2.3 Problem Formulation

The proposed solution aims to reduce the PDR of the D2D smart grid devices by performing efficient resource allocation in time and frequency. To improve reliability, signal-to-interference plus noise ratio (SINR) is used, where improving SINR improves the probability of successfully decoding the transmitted packets. SINR is formulated as follows:

$$\gamma_{u,k} = \frac{b_{u,k}\ p_{u,k}\ h_{u,k}}{\omega_k\ N_0\ + \sum\limits_{m \in I} b_{m,k}\ p_{m,k}\ h_{m,k}}, \tag{4.1}$$

where $\gamma_{u,k}$ is SINR of u^{th} D2D pair, and I is the set of interfering D2D pairs, i.e. the D2D pairs that use same resource blocks as transmitting u^{th} D2D pair. For the remaining terms, $b_{u,k}$ is allocation indicator of u^{th} D2D pair; $p_{u,k}$ is the transmit power of the u^{th} D2D pair on the k^{th} resource block; $h_{u,k}$ is the channel coefficient of k^{th} resource block; ω_k is bandwidth of k^{th} resource block; N_0 is additive white Gaussian noise (AWGN) single-sided power spectral density; $b_{m,k}$ is allocation indicator of m^{th} D2D interfering

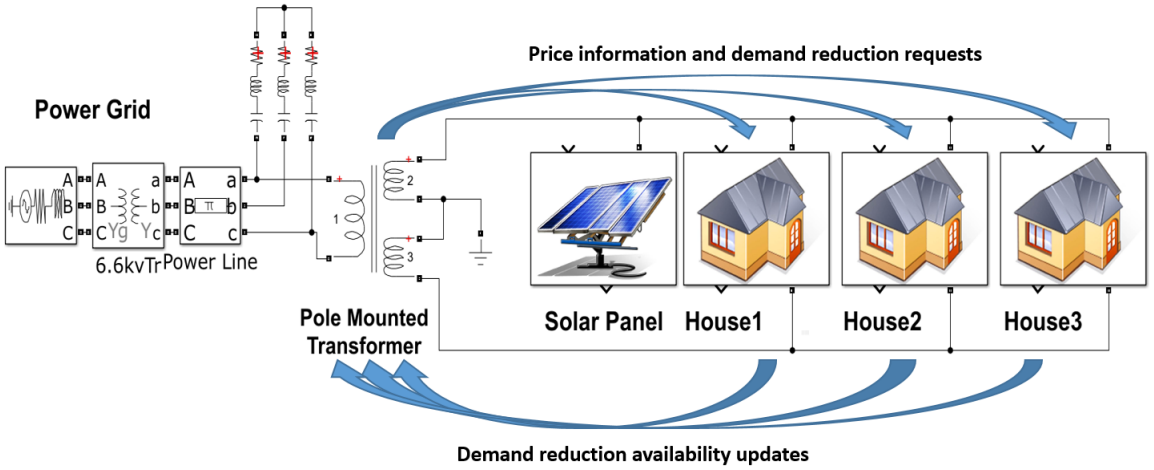

FIGURE 4.2: Diagram of power system modeled in the integrated simulator.

pair; $p_{m,k}$ is transmit power of interfering m^{th} pair; and $h_{m,k}$ is the channel coefficient of m^{th} interfering pair. Here, the optimization problem aims to maximize the aggregate SINR of smart grid D2D pairs as follows:

$$\max_{b_{u,k}} \quad \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{u,k}, \tag{4.2}$$

where N is the number of D2D pairs within the network, and K is the total number of resource blocks available. It should be noted that the objective function is solved subject to fixed power transmission. Moreover, the effects of device mobility are assumed negligible due to the assumption that the D2D nodes are stationary smart grid devices.

4.2.4 Demand Response Enabled Power System Model

Smart grids are a combination of two sophisticated systems, namely a traditional power system and a communications network. To test the HRQ algorithm's impact on the performance of the communication network in relation to the LTE standard, it is only necessary to model the data traffic of the smart grid applications considered in this study. However, the demand response (DR) application was modeled on the power systems side as well in order to witness the impacts of communication system performance on the power system dynamics of a DR-enabled microgrid. The power system environment within which this application is being simulated can be visualized in fig. 4.2 [87]. The details of the simulation environment is provided in Section 4.5. In the model, if the power flow through a transformer crosses a certain threshold level P_{limit}, a "demand response event" is initiated. If such an event happens, the smart meter continuously recalculates the minimum demand reductions required to keep the power consumption

from exceeding P_{limit}, and sends messages to the households containing new power reduction requests. Also, households were set to reduce their power consumption to a constant "base comfort" level as long as the price of electricity surpasses a particular value C_{max}. It is assumed that households have instantaneous control over their power consumption.

4.3 High-Reliability Q-Learning (HRQ) Scheme

HRQ is a multi-agent distributed Q-Learning algorithm performed by each D2D transmitter to efficiently allocate resource blocks every scheduling access period for reliability maximization. Before characterizing HRQ, two subsections derive the applicability of our chosen Q-Learning methodology. In the first subsection, the problem formulation in Section 4.2.3 is recast as maximizing the long-term return of a Markov Decision Process, thus making it possible to apply reinforcement learning algorithm to our problem. In the second subsection, the most important "dimensions" of reinforcement learning are explained and used as a lens through which to observe the various available methodologies and select the one appropriate to the current problem. The notion of reinforcement learning dimensions used in this thesis is consistent with the works of Sutton *et al.* [98], which developed this intellectual framework to provide means to having a more timeless and robust understanding of reinforcement learning principles that are not defined in terms of specific methods. Aside from justifying both the use of reinforcement learning and the Q-Learning algorithm in particular for this problem, these two subsections also provide essential background knowledge on reinforcement learning.

4.3.1 Observing Optimization Problem from a Reinforcement Learning Perspective

Ultimately, reinforcement learning methods all assume that the problem they are trying to solve is maximizing the return of a Markov Decision Process (MDP). A MDP specifies the dynamics of an agent interacting with an environment characterized by a state $s \in S$ with an action $a \in A$ that results in receiving a reward $r \in R$. The state and action sets can be created in any way that makes sense for the specific MDP under consideration. In contrast, the reward set is always a set of scalar values as per the Reward Hypothesis [98]. In addition to reward generation, the environment-agent interactions result in the environment transitioning to a new state $s' \in S$; this new state may be the same as the previous state. This process is sequential in nature, and understanding the sequencing dynamics of the process is crucial to understanding any reinforcement learning algorithm.

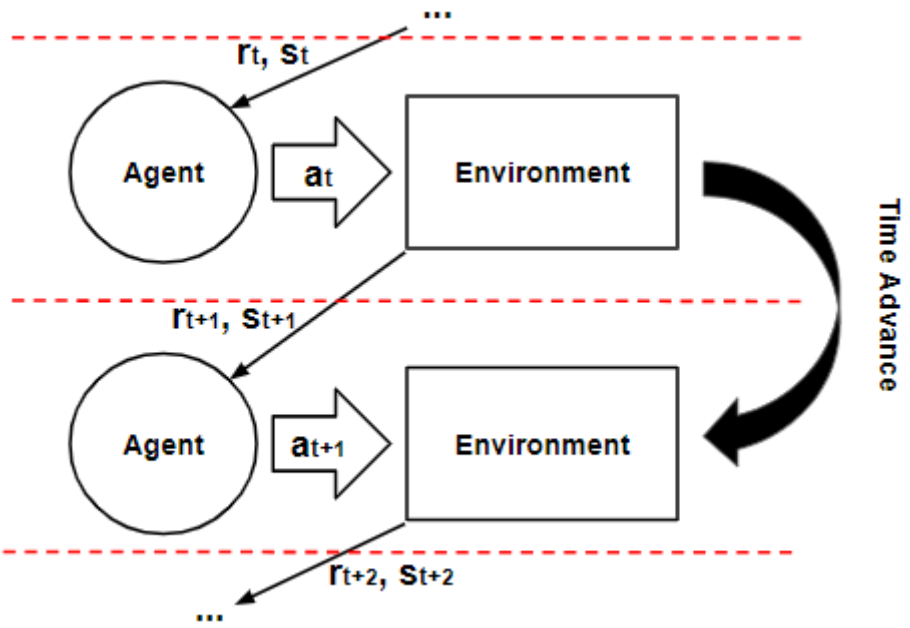

FIGURE 4.3: Visualization of MDP sequence dynamics.

Fig. 4.3 shows that starting at time t, the environment is in state s_t, and action a_t is taken. Only at time step $t+1$ is the reward r_{t+1} induced by action a_t while in state s_t received. This reward is known once the environment has transitioned to s_{t+1}.

Given the state of the environment, the sets of feasible actions in addition to their expected rewards are subject to change. As alluded to when mentioning *expected* reward, it is common for the reward for taking a particular action in a certain state to be non-deterministic. Thus, if a state is revisited in the MDP and the same action is taken as before, depending on the nature of the environment, the agent may not receive the same reward as last time. Moreover, it also may very well be that the same action does not result in the environment transitioning to the same state. Consequentially, as defined in Sutton *et al.* [98], the environment of a MDP is characterized by a joint probability distribution $p(s', r|s, a)$ of next state, immediate reward pairs (s', r) conditioned on the current state-action pair under consideration. Note that we characterized the MDP's environment dynamics so that they are only based on the current state of the MDP and the next action to be taken. As a consequence, the visitation of past states has no impact on what will happen beyond the current state. This characteristic is known as the Markov Property [98], and in order to formulate a problem as an MDP, the state space must be such that any $s \in S$ contains all information required to determine state transition dynamics.

To successfully manage a MDP, one is challenged with developing an action selection policy $\pi(a|s)$ that maximizes the long term "return" received by an agent. Note that "return" is a function of the rewards received over the course of the MDP, and is generally

not the same as a single reward received at any given point in time. How the return is defined depends on the dimensions of the reinforcement learning problem. As an example, consider a MDP that starts at some state s, and comes to an end after 5 state transitions. This is an example of an "episodic" MDP, where there is a definitive end to the process, opposed to the process continuing indefinitely. In this case, the "return" associated with state s is commonly defined as the sum of expected rewards received after all state transitions (so 5 total reward values for this example) of the episode. Since the policy $\pi(a|s)$ influences the actions that will be taken for a given state, and since these chosen actions then influence the rewards obtained throughout the MDP, the expected return associated with a given state depends on the characterization of $\pi(a|s)$. $\pi(a|s)$ is characterized as the distribution for choosing any given action a that is conditioned on state s.

From these ideas, the "value function" $v_\pi(s)$ can now be introduced, which is a function of state that outputs the expected return of a given state assuming that some policy $\pi(s|a)$ is followed in all future decisions made by the agent in the MDP. When $v_\pi(s)$ is defined in this way, it can be expressed recursively, i.e. the expected return of the current state can be expressed in terms of the expected reward to be gained from the next state transition plus the expected output of the value function evaluated at the next state. When written recursively, one obtains the Bellman equation for the value function, which is presented in Sutton *et al.* [98] as

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S, r \in R} p(s', r|s, a)[r + \gamma v_\pi(s')], \quad \forall \ s \in S. \tag{4.3}$$

The γ term is a discount factor used to generalize the equation so that it can handle defining return in a way that wants to devalue expected rewards received the further into the future they are. If less than one, then the value of future states is scaled down accordingly. Note if $v_\pi(s')$ is replaced by its own Bellman equation, then another γ is injected into the equation overall, making the reward two steps ahead get scaled by γ, while the "$v_\pi(s'')$" term is scaled by the square of γ. This process can be done indefinitely. As a matter of opinion, the common way of expressing the reward term in the Bellman equation is rather misleading. It is confusing to not have any notation indicating which time step the reward term is associated with. Since s and s' denote the state of the environment at the current and next time steps respectively, it naturally leads the reader of the equation to believe that r is associated with the current time state, when in reality it is produced at the next time state.

Return values can also be assigned to state-action pairs, and a function similar to $v_\pi(s)$ can be defined, namely the "action-value function" $q_\pi(s, a)$. A Bellman equation also

exists for the action-value function. The Bellman equation representation of value and action-value functions are important to the process of finding the actual values of such functions, as one can either solve them through policy iteration techniques, or if the size of the state-action space sufficiently small, it may be even possible to solve a system of linear equations for all state values [98]. Once the (action)-value function is known for a given policy, it provides a figure of merit with which we can evaluate how good that policy is relative to other policies. Moreover, new strictly superior policies can be built using the valuation of old policies. For more details on this, consult the sections of Sutton *et al.* on "General Policy Iteration".

To complete the idea of the MDP, one must have a quantitative description of the optimal behaviour of an agent interacting with its environment. The Bellman optimality equation for $q_*(s,a)$ provides this [98]:

$$q_*(s,a) = \sum_{s' \in S, r \in R} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')], \tag{4.4}$$

where

$$q_*(s,a) = \max_{\pi} q_\pi(s,a), \quad \forall \ s \in S, a \in A. \tag{4.5}$$

Equation 4.5 expresses the ideal objective of the agent in a MDP, namely to produce a policy $\pi(a|s)$ that optimizes the action values for every state action pair. Equation 4.4 is a recursive form of 4.5, and is foundational to developing methods to find optimal policies in practice.

Now that the architecture of a MDP has been established, it can be shown how the resource allocation problem specified in 4.2.3 can be recast as a MDP, and thus a reinforcement learning problem. Resource allocation is merely a sequential decision making process based on the current state of the network, where each action taken is the allocation of time and frequency resources for a device to transmit data with. The resource allocation mechanism is the agent, and the cellular network is the environment. After each resource allocation action is taken, the state of the environment changes, and the resultant QoS performance of the network can be transformed into a scalar reward value. In our case, we care about maximising SINR, so the reward would be proportional to the SINR achieved. If the state of the agent is defined as the current QoS performance of the network, the agent has all information that will influence what the next action will result in concerning next state and reward. In other words, the QoS of the network in the past has no bearing on what the current QoS is, nor the decision that must be made to either

maintain or improve it. As a consequence, our resource allocation problem if presented in this way obeys the Markov Property, and is thus a valid MDP. Finally, since we can construct a valid MDP, we now have access to reinforcement learning methods to solve our original problem. In the next subsection, we will traverse the salient dimensions of the reinforcement learning problem space to arrive at the method chosen for this paper, namely a particular flavour of Q-Learning.

4.3.2 Choosing an Appropriate Reinforcement Learning Algorithm

To determine which reinforcement learning method is appropriate to apply to our MDP, we must analyze the various "dimensions", or characteristics of the reinforcement learning problem we are faced with. The most relevant reinforcement learning dimensions that will be considered in this analysis are the following:

- **Existence of Model of Environment:** Whether or not one has access to the $p(s', r|s, a)$ probability distribution of the environment determines if it is possible to use Dynamic Programming methods that involve using the Bellman equations directly to find optimal policies [98]. Although Dynamic Programming is perhaps the most mathematically grounded approach to finding an optimal policy, it is seldom used in practice, as it is rare to have the transition dynamics of the environment available. In our case, we indeed do not have access to a model of the environment, which eliminates the possibility of using Dynamic Programming. Fortunately, there are many alternative methods that can be used that do not require a model of the environment, and instead learn the values of states and actions by interacting with the environment. Using this trial and error approach is analogous to sampling $p(s', r|s, a)$ at every time step, so the sample mean of the environment dynamics will be the same as $p(s', r|s, a)$. After the agent interacts many times with the environment, reliable estimates of the value of policies used can be attained.

- **Real Experience Vs. Simulated Experience:** Another dimension that exists when sampling experience from the environment is whether or not a simulator is used to recreate the environment the agent is acting in. In our case, deploying a remote smart grid network to test our algorithm is not feasible due to both cost and time constraints, so a simulation must be used. When simulators are used in lieu of reality, the policy learned with simulated experience may not translate perfectly when transferred into the real world. The extent to which the simulated learning can be redeployed in reality depends on the level of fidelity of the simulator, i.e. how realistic it is. It is for this reason that considerable effort was taken

in Chapter 3 of this thesis to develop a high fidelity simulation environment of a D2D communications enabled smart grid. Unlike most communication network simulators, ours captures LTE-compliant communication dynamics down to the physical layer, so it is believed that the simulated experience using our simulator would have minimal issues if applied to a real smart grid.

- **Depth of Update and Width of Update:** There are many ways to update the valuation of a given state or state-action pair of a MDP. The "depth" and "width" dimensions help describe any given value update strategy. The depth of an update corresponds to how many future state values after the state it is updating are considered by the agent while updating its value. For example, an approach that is trying to iteratively solve equation 4.3 as it is written would be a low-depth approach to update $v_\pi(s)$. However, an approach that implements recursion and re-injects equation 4.3 into itself to replace $v_\pi(s)$ with a Bellman term containing $v_\pi(s'')$ is considered to be one step "deeper" than the previous approach. The width of an update is simpler to describe; a value update is wide if it considers all possible (s', r) outcomes weighted by their respective probabilities, for each possible action that the policy may allow in the state being valuated. This is the case in equation 4.3. However, a wide update strategy is only possible if a model of the environment is known. If sample experience is used, like in our case, then only one possible outcome will be reflected in each value update. Regarding depth, our problem is believed to desire a low depth update. The main reason for this is related to speed of convergence; low depth temporal difference based methods have been observed to have lower variance value update trajectories compared to high depth Monte-Carlo based methods, which usually corresponds to faster convergence to a stable value estimate/policy [98]. This is desired in most online learning settings such as this one, since it is clearly desired for the agent to start performing adequately as soon as possible. Thus, the method that would be appropriate for our problem would have both low width and low depth, which points to a single bootstrapped return temporal difference based online learning method.

- **Agent Planning:** A common strategy in reinforcement learning to make the most use out of experience obtained from the environment is to have the agent execute a "planning" step in between each interaction with the environment, where the agent updates its state-action values and/or policy based on some model of the environment. We have already established that pre-existing model of our environment is not available, however it is still possible for an agent without access to $p(s', r|s, a)$ to learn a model of the environment through experience. The most common way of doing this is to save the experienced state transitions and reward values that

were generated by the actions taken by the agent. Then in the planning step, these saved (s', r, s, a) tuples are sampled according to some sampling strategy, and used to update the agent's state-action values and/or policy. Nonetheless, a planning step was not used in this problem due to the highly non-stationary communication network environment. It is believed that past experience in this problem becomes obsolete very quickly due to the ever-changing configuration of the network, both in regards to the traffic characteristics of an individual node, as well as the very number of nodes active in the network at any given point in time. It is also because of this reason that an aggressive learning rate is used in the Q-Learning algorithm.

- **Continuous or Episodic:** An episodic task is one where there is a distinct end, or "terminal state" inherent to the MDP, such as a game ending in a win or loss. Conversely, a continuing task runs indefinitely. Our problem is a continuing task. In order to have finite return values for continuing tasks, one can introduce discount factors to the values of future states, or use "differential returns", which are returns that are relative to the average reward associated with a state under a particular policy. Differential returns do not go to infinity because it is equally likely for a differential return to be positive or negative. We decided to use a discounted return due to the non-stationarity of the communication network environment; there is definitely no value in looking infinitely far into the future like differential returns do, as the policy being learned for the present network configuration will need to adapt in response to changes in the network. It should also be noted that since our problem is continuous, it makes methods designed for episodic tasks such as Monte-Carlo based methods not applicable. Thus, this dimension also points us in the direction of temporal difference based methods.

- **On or Off Policy Learning:** Another key decision that must be made in the reinforcement learning algorithm design process is whether or not the policy your using to explore the environment is the one that is being valuated by the algorithm. For example, if an "on-policy" approach used an epsilon greedy policy (where with probability $(1 - \epsilon)$ the action with the highest expected value is chosen) to choose the actions the agent is taking in the environment, then the update rule of the algorithm must also assume subsequent actions taken by the agent will also adhere to an epsilon greedy policy. Conversely, this does not need to be true for an "off-policy" approach. An off-policy approach is free to choose any policy desired to explore the environment in order to learn the values of a completely different policy.

  Mission critical smart grid applications will rely on our prospective resource allocation algorithm to offer reliable delivery of information, implying that a reliable, deterministic policy is ideal to obtain through learning. In other words, we would

not want the resource allocation algorithm to suddenly take a poor random action in the middle of an important smart grid control process. However, the agent must start with a non-deterministic policy in order to explore the state-action space to come up with a deterministic policy. This problem can be easily solved with an off-policy learning approach; in the first few moments of the algorithm, the agent explores with a stochastic policy such as epsilon greedy, but updates the state-action values of the MDP according a deterministic policy, then when a sufficient amount of exploratory actions have been taken, the action can switch to the deterministic policy that is greedy with respect to the action values learned.

- **Value Function Approximation:** In our resource allocation problem, we have initially assumed that only a moderate number (less than 12) communication nodes will be active at a time in the D2D network, which is not unreasonable in for the remote community setting we are targeting. As such, we will be partitioning our action space into 12 discrete selections of different orthogonal regions of the LTE resource grid every scheduling access period. Moreover, our state space, which provides an indication of the current channel quality of an agent, has been made small as well. Only two distinct states are defined, one where ideal CQI is achieved, and another where it is not. As will be seen in the results, this simple state-action space was sufficient to achieve an effective distributed self organization of cellular resources among users. Nonetheless, the natural question that arises from this is how scalable is our HRQ algorithm. With such a small state-action space, the time to completely explore the state-action space is rather short, but what if we wanted to deploy this in a larger D2D network? In this case, we would need a more flexible state action space, where it would become feasible for tens, perhaps hundreds of devices to communicate at once. This would imply that the number of disjoint resource groups that can be chosen amongst users would have to grow to at least the size of the users in the network.

  One option would be able to continuously increase the size of the state-action table, but after a certain point, exploration of all possibilities would become infeasible. At a certain scale of network, it would become a necessity to replace the tabular based method used in this thesis with a function approximation method. Instead of learning the value of each state-action pair separately, the agent can instead learn a smaller set of function parameters that transforms the state action space into a new representation that can generalize across different states and actions. This generalization reduces the need for exhaustive exploration of all possibilities, as the function can provide value estimates for states that it has not seen before. This is because the function's parameters get tuned by similar previously visited states to generate a reasonable output for those that have never been explicitly experienced.

In fact, this approach is especially useful when there are infinite state possibilities, as is the case when the state variables are continuous. A popular version of value function approximation is deep reinforcement learning, where the state is input into a neural network, which through nonlinear regression, estimates the value of each action an agent can take; each node in the output layer represents the estimated value of a different action. However, if the dimensionality of the state space is small, linear function approximation methods like tile coding are simpler yet effective options as well.

As a summary of the analysis performed by considering the various dimensions fundamental to every reinforcement learning problem, it has become apparent that a tabular, model free, off-policy, temporal difference based learning algorithm is suitable for the scope and nature of the problem we are trying to solve. This leaves us with two popular reinforcement learning methods as options, namely Expected SARSA and Q-Learning. Q-Learning updates its action values by computing the Bellman optimality equation directly using estimates of the next-state's action values to approximate some of its terms [98]:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right]. \tag{4.6}$$

The terms α and γ are scalar learning rates and discount factors respectively. S_t and A_t are the state and actions experienced by the agent at time t. S_{t+1} and R_{t+1} are the observed next state and reward that resulted from the previous state-action pair.

Expected SARSA is a generalization of Q-Learning, where instead of having learn the greedy policy's action values, the temporal difference update's target value is generated by the sum of all possible next-state action values weighted by the probability of any given policy [98]:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\right]. \tag{4.7}$$

Expected SARSA allows one to learn the value of an "epsilon-soft" policy, for example one based on a softmax function of the set of action values. This makes Expected SARSA capable of finding the value of a stochastic policy. However, in this work, we are not interested in learning a stochastic policy, as we are interested in finding a deterministic policy that can reliably offer high QoS to the nodes in the cellular network. Moreover, Expected SARSA is more computationally expensive than Q-Learning (when they are

not the same thing), as it must compute a weighted average of all possible next-state action values. As such, Q-Learning was the reinforcement learning algorithm chosen for this problem. The next subsection characterizes the details of the Q-Learning tuple used for HRQ.

4.3.3 On Other Machine Learning Alternatives to Reinforcement Learning

One may pose the question that asks why, among machine learning algorithms, should we choose reinforcement learning to implement our solution. Reinforcement learning methods are particularly apt to navigating sequential decision making processes where it must influence a stochastic, potentially non-stationary environment to gain favourable returns, which is exactly the same problem that a distributed cellular resource allocation strategy faces.

Supervised learning methods, say recurrent neural networks, could also potentially be used to map input historical time series data related to the evolution of network state to some new resource allocation action to be taken in the next time step. However, to train such algorithms, it requires extensive data sets with labeled "correct actions" to make given the input time series data. Due to the non-stationarity of the cellular network, what the "correct action" is for a given state is likely to change as time goes on, making training neural networks or other supervised learning constructs before the environment changes difficult. Another alternative is the use of genetic algorithms to make resource allocation decisions, which is a heuristic approach already thoroughly explored in the literature [99–101].

4.3.4 HRQ Characterization

HRQ models reliability in terms of SINR of D2D devices as per equation (4.1), as low SINR is a root cause of PDR. In particular, efficient resource block allocation allows devices to select disjoint actions, i.e. distinct resource blocks, which lessens interference and increases SINR. As such, improving SINR increases the probability of successfully decoded packets which achieves higher reliability. Besides reliability, interference mitigation enables devices to transmit on channels of higher quality that lead to allocation of higher transport block size (TBS). As such, large packets are less prone to segmentation at the radio link control layer, leading to reduced latency. It should be noted that this algorithm manages the allocation of resource blocks, i.e. physical spectrum, and is not designed to manage transmitter power settings.

In HRQ, we address the reliability by formulating the Q-Learning tuple as follows:

- **Agents:** D2D transmitting nodes.
- **States:** Channel quality is used to represent HRQ states, where CQI_{ideal} is defined as the best channel quality that drives devices to allocate disjoint resources, hence achieving highest reliability.
- **Actions:** HRQ performs actions that manifest as the selection of cellular resources in time and frequency every scheduling access period. To minimize the action space, contiguous sets of subframes and resource blocks are aggregated to form larger (but still orthogonal) sets of resources that agents can select as their scheduling decision. In frequency, every four contiguous resource blocks are combined to form a "resource block group", whereas in time, the pool is partitioned into two 16 subframe intervals. The network model was set to a 5 MHz bandwidth and 40 ms scheduling access period, which consists of 25 resource blocks and 32 subframes for data transmission. Under such conditions, 12 actions populate the action space.
- **Reward function:** The reward is defined as follows:

$$\tau_u = \begin{cases} -1 & CQI_u < CQI_{ideal}, \\ 1 & otherwise, \end{cases} \tag{4.8}$$

where τ_u is the reward of u^{th} D2D pair, CQI_u is the channel quality of u^{th} D2D pair, and CQI_{ideal} is the ideal channel quality. The rationale behind the reward function is to reward the agent when achieving the ideal QoS, i.e. ideal channel quality, whereas it penalizes the agent otherwise. The reward and punishment values are of equal magnitude for two main reasons. The first reason is that it makes the Q values easier to interpret; when the magnitudes of reward and punishment are equal, positive Q values automatically indicate that the action is more often successful in achieving ideal QoS, and vice versa. The second reason is that unequal magnitudes would result in asynchronous adaptation speeds of the Q values in the TD process. As an extreme example to illustrate the point, say one rewards the agent 1000 units should they find an action that achieves ideal QoS, and -1 units otherwise. Should the agent receive the 1000 unit reward for an action, and then have the environment change where the action is now no longer a good one to take, it will take a long time for the agent to learn that things have changed, as the new rewards of -1 would take many time steps to influence the Q value that had been updated with a value of 1000. Setting the reward and

punishment values to be equal magnitude should assure a minimal number of TD updates no matter how the true action values change.

- **Q-value update:** The Q-values are updated according to the temporal difference (TD) equation [26]:

$$Q(S_u, a_u) = Q(S_u, a_u) + \alpha[\tau_u + \gamma \max_{a \in \mathbf{A_u}} Q(S_u^{'}, a) - Q(S_u, a_u)], \tag{4.9}$$

where α is a learning rate, γ is a discount factor, $\mathbf{A_u}$ is the action-space of u^{th} D2D pair, and $Q(S_u^{'}, \mathbf{A_u})$ are the Q-values at next state $S_u^{'}$ and all actions $\mathbf{A_u}$. The new action $a_u^{'}$ is selected based on a modified version of ϵ-greedy approach:

$$a_u^{'} = \begin{cases} random & \epsilon, \\ random(arg \max_{a \in \mathbf{A_u}} Q(S_u^{'}, a)) & 1 - \epsilon, \end{cases} \tag{4.10}$$

where ϵ is the exploration probability. The modified ϵ-greedy accounts for situations where multiple state-action pairs are tried for having the largest Q-value during exploitation iterations. Thus, during the exploitation phase, the scheduler chooses among the set of actions with the highest Q-value at random instead of the conventional approach of selecting the first Q value in the list returned by a max function.

Algorithm 1 presents the steps of HRQ. TM-2 currently has no means to feed back channel quality metrics to the transmitters, but a new sidelink control information message that can be sent by the receiver back to the sender would be a simple solution to this. The algorithm terminates after T scheduling access periods.

This study compares the performance of the proposed HRQ scheduling algorithm to that of the allocation scheme prescribed by the LTE standard for D2D communication operating in TM-2 [102].

4.4 LTE Random Resource Allocation Algorithm and Simulation Implementation Details

4.4.1 Baseline Algorithm: LTE Random Allocation

TS 36.321 Release 12 Section 5.14.1.1 specifies to randomly select the time and frequency resources for sidelink shared channel (SL-SCH) and sidelink control information of a

Algorithm 1 HRQ

Initialization: Q-table $\leftarrow$ 0, α, γ, and ϵ.
for scheduling access period $t = 1$ to T **do**
  Step 1: Update channel quality value based on the last transmission.
  Step 2: Compute the reward τ_u and observe the new state $S_u^{'}$ due to last action execution.
  Step 3: Update the Q-value as in (4.9).
  Step 4: Transit to next state $S_u^{'}$.
  Step 5: Select the next action $a_u^{'}$ based on modified ϵ-greedy policy as in (4.10).
  If Exploitation and tie **then**
    Select action Randomly
  End If
end for

sidelink grant from the resource pool configured by upper layers with equal probability [102]. This specification is implemented as **Algorithm 2** in our simulator.

Conceptually, the scheduling algorithm calculates how much data is queued in the device's transmission buffer, and then determines all feasible combinations of I_{trp} (time resource pattern index) and RIV (Resource Indication Value) sidelink control information scheduling parameters that correspond to the smallest amount of bandwidth capable of sending all the existing data queued over the course of the next scheduling access period. Finally, the scheduler randomly selects among all feasible scheduling options that were found this way with equal probability.

The variables of **Algorithm 2** have been defined as follows:

- $TBSdesired$: The TBS that would result in the scheduler being able to clear the data buffer of the agent over the course of the next scheduling access period. It should be noted that any data that arrives after the scheduling decision must wait until the next scheduling decision in order to be allocated resources.
- $nprbMax$: The largest number of PRBs a scheduler can self-allocate given the bandwidth of the scheduling access period. In this study, a 5MHz scheduling access period was used, so $nprbMax$ was set to 25 PRBs.
- I_{TBS}: An index that maps a particular Modulation and Coding Scheme (MCS) to a row in an LTE lookup table (for example TS 36.213 Release V12.5.0 Section 7.1.7.2 Table 7.1.7.2.1-1) that ultimately yields TBS values for given combinations of MCS settings and the number of PRBs being used to map the TB to the resource grid.

- $nTBs$: A feasible number of transport blocks that can be sent by an agent over the course of a scheduling access period. For the given network configuration in this study, $nTBs$ can equal 1, 2, 4, or 8.
- $nPRBs$: The number of PRBs required to be allocated in order to have access to TBs with a size equal to or greater than $TBSdesired$ given a particular $nTBs$ value.
- $nTBnPRBcombos$: A list of all feasible $[nTBs, nPRBs]$ combinations that yield enough bandwidth to clear the agent's data buffer.
- RIV: A "Resource Indication Value" parameter specified by the LTE standard for D2D communications that specifies both the number of PRBs to be scheduled in the next scheduling access period in addition to which set of contiguous PRBs to use.
- I_{TRP}: A Time Resource Pattern Index specified by the LTE standard for D2D communications that specifies the number of subframes and their indices in the next scheduling access period. This ultimately controls how many TBs can be sent in a single scheduling access period by an agent and when.

4.4.2 Complexity Analysis

The complexity of the LTE random allocation was compared to the proposed Q-learning algorithm, HRQ. As shown in **Algorithm 2**, random allocation performs a search in TS 36.213 Release V12.5.0 Section 7.1.7.2 Table 7.1.7.2.1-1 [103] to find the required number of resource blocks $nPRBs$ to allocate per transport block that will be transmitted in a single scheduling access period. Recall from Section 4.4 that $nPRBs$ should be large enough to fit all data desired to be transmitted over the upcoming scheduling access period. Naturally, this number depends on the number of transport blocks that will be sent; if fewer are sent, then each transport block must contain more PRBs than when more are sent. Thus, $nPRBs$ must be calculated up to $max(nTBs)$ times, where $max(nTBs)$ is the number of transport blocks that can be placed in the data channel of a single scheduling access period. $max(nTBs)$ is a quarter of the number of subframes present in the PSSCH, which in a 40 subframe scheduling access period with a 32 subframe PSSCH, is equal to 8 transport blocks. One must divide the number of subframes by 4 because of how the open loop HARQ process of TM-2 sidelink communications requires transmitting 4 copies of every transport block sent. However, the LTE standards generally do not allow any number from 1 to $max(nTBs)$ to be set as the number of

Algorithm 2 Implementation of 3GPP Release 12 D2D TM-2 Random Allocation Scheduling

Initialization: $nprbMax$, I_{TBS}.
for scheduling access period $t = 1$ to T **do**
  $k = 1$
  for $nTBs$ = Each possible number of transport blocks to transmit next scheduling access period **do**
    Step 1: Compute $TBSdesired$.
    Step 2: Iterate through TS 36.213 Release V12.5.0 Section 7.1.7.2 Table 7.1.7.2.1-1 [103] until it is discovered how many PRBs are required to generate a transport block with a TBS equal to or greater than $TBSdesired$:
    for $i = 1$ to $nprbMax$ **do**
      Read the TBS value corresponding to I_{TBS} and i PRBs.
      If $TBSvalueread \geq TBSdesired$
        $nPRBs \leftarrow i$.
        $success \leftarrow$ true
        Break from innermost loop
      end If
    end for
    If $success$
      $nTBnPRBcombos(k) \leftarrow [nTBs, nPRBs]$
      $k \leftarrow k + 1$
    end If
  end for
  Step 3: For every set $[nTBs, nPRBs]$ in $nTBnPRBcombos(k)$, determine all $[RIV, I_{TRP}]$ pairs possible, and out of ALL pairs across ALL $[nTBs, nPRBs]$ sets, choose one at random as the final scheduling decision for the scheduling access period.
end for

transport blocks allowed for a D2D node to be sent in a single scheduling access period. Depending on the network configuration, a specific subset of possible quantities of transport blocks are allowed to be sent per scheduling access period. In the network configuration chosen in this study, a D2D node could only transmit either 1, 2, 4, or 8 transport blocks in a single scheduling access period. If the data payload is very large, then the algorithm can iterate up to $nprbMax$ times before realizing that for the number of transport blocks to be sent, it cannot fit all data in the D2D node's buffer. Thus, a conservative estimate for the computational complexity of the LTE random allocation algorithm would assume that any number of transport blocks up to $max(nTBs)$ is allowed, and the algorithm iterates $nprbMax$ times for each number of transport blocks considered. Assuming linear search, this worst case scenario requires $nprbMax$ iterations every $max(nTBs)$ iterations, making the Big-O complexity of random allocation $O(NM)$, where $N = nprbMax$, and $M = max(nTBs)$.

Concerning HRQ, for a specific bandwidth configuration the number of resource block groups is defined as $nRBGs = \lfloor \frac{nPRBs}{sRBG} \rfloor$, where $sRBG$ is the size of a resource block

group in resource blocks. $nRBGs$ represents the number of disjoint sets of PRBs that an HRQ agent has to choose from in its resource allocation process. From the perspective of the Q-learning algorithm, $nRBGs$ is one dimension of the action space. Time is the second dimension of the action space, where the HRQ agent must decide which among a finite group of disjoint sets of subframes in the resource grid it should choose to transmit transport blocks within. Let the number of different groups of subframes the HRQ agent can choose be defined as $nSFGs$, i.e. the "number of subframe groups". In this study, we let $nRBGs = 6$, and $nSFGs = 2$.

As an aside, the configuration of $nRBGs$ and $nSFGs$ was chosen based on the number of smart grid nodes that were desired to be simulated, their bandwidth requirements, and the size of the PSSCH. The desire was to be able to test the performance of HRQ for low, medium, and high levels of traffic congestion. Since 6 smart grid D2D nodes are sending information in our study, we wanted to make the size of the PSSCH resource grid partitions such that these 6 devices would occupy half the available bandwidth of a single scheduling access period. As such, if we do not simulate any other communication nodes, a "light" congestion scenario can be simulated. Then, by incrementally introducing additional "auxiliary" devices to the network, we can evaluate the HRQ algorithm's performance for moderate to severe levels of network congestion. Since only 6 free grid partitions remain after the smart grid nodes are introduced into the network, it does not require many more nodes to simulate a stressed communication network, minimizing the computational burden for our experiments.

Under the same assumption of linear search used while evaluating the LTE random resource allocation algorithm, the complexity of HRQ relies mainly on the maximum search performed in eq. (4.9) and (4.10). Therefore, complexity of HRQ can be identified using number of actions at a desired state, i.e. a row in the Q-table. Number of actions is determined by considering the possible allocation units in time and frequency direction which are $nSFGs$ and $nRBGs$ respectively, hence the total number of actions becomes the product of these two values. As such, the Big-O complexity of HRQ is also $O(NM)$, but in this case $N = nSFGs$, and $M = nRBGs$.

It should be noted that which algorithm is more computationally complex is decided by the size of the action space of the HRQ algorithm. One can tune the granularity of the state action space of the HRQ algorithm to either outperform, match, or be worse than LTE Random allocation; the implementer of the HRQ algorithm must then consider the computational resources they have at hand when specifying the state action space of the HRQ algorithm, as there is a fundamental trade-off between computational speed and the flexibility of the HRQ algorithm.

TABLE 4.2: Network settings

Network Settings	
Transmission bandwidth	5 MHz
Number of resource blocks	25 (12 subcarriers / resource block)
scheduling access period	40 msec
Time-to-transmit interval (TTI)	1 msec
D2D resource block pool	25
D2D subframe pool index range	8-39
Microgrid radius	50 m
Number of microgrid D2D pairs	6
Number of auxiliary D2D	0-11
Channel Settings	
Transmit power	20 dBm
Tx/Rx antenna gain	10 dB
Pathloss	3GPP pathloss model [104]
Penetration loss	5 dB
Noise Figure	5 dB
Shadowing	$\sim$ LOGN(0, 2(dB))
Q-Learning Settings	
Learning rate (α)	0.5
Discount factor (γ)	0.9
Exploration probability (ϵ)	0.1

4.5 Results

This section is divided into three subsections. Subsection 4.5.1 presents results on the latency and PDR performance of both scheduling algorithms, where 20-40% reductions in PDR and >10 ms drops in latency were observed for all smart grid applications. Subsection 4.5.2 shows the convergence of the HRQ algorithm; HRQ was able to converge as long as the number of agents did not exceed the size of the action space. Subsection 4.5.3 presents the results collected on the observed changes in power system dynamics of the simulated DR-enabled microgrid as a function of resource allocation mechanism selection. The power fluctuations observed were reduced by two orders of magnitude when HRQ was used in lieu of random self-allocation.

For the simulations, the D2D communication protocol stack was implemented on top of MATLAB's LTE Toolbox which was then combined with a microgrid simulator implemented using Simscape Power Systems and SimEvents software packages available in the Matlab/Simulink environment. The performance and convergence of the HRQ algorithm was tested using the mobile communications portion of the developed simulator, whereas all power and communication elements were leveraged in simulating the DR-enabled microgrid. The simulation parameters are given in Table 4.2:

4.5.1 Comparing the performance of HRQ and LTE TM-2 scheduling strategies under varying traffic intensities

The HRQ and LTE schedulers were tested and compared under various levels of network traffic. The network traffic was adjusted by how many "auxiliary" D2D devices were injected into the communication network. At each level of network traffic tested (i.e. for 0, 3, 5, 6, 7, 9, and 11 auxiliary devices present in the network), the communication network was simulated for 15 s, over which time 375 scheduling access periods (and thus scheduling decisions) elapsed. Moreover, for every 15 s simulation at every level of network traffic, the simulation was repeated 12 times, and the average message latency and PDR values were plotted with 95% confidence intervals for both scheduling strategies.

Over these tests, the exploration time of the HRQ algorithm was set to 10 s, during which time an epsilon greedy exploration strategy was implemented by the scheduler. After 10 s, the HRQ algorithm switched to a purely greedy strategy, only executing its policy without exploration. This test compared the performance of HRQ with respect to LTE scheduling strategies after HRQ has completed its exploration phase. Thus, the performance metrics obtained are calculated based only on the final 5 s of the network simulation.

The results of latency and PDR performance for both schedulers have been presented in fig.s 4.4 and 4.5 respectively. Fig. 4.4 demonstrates the potential for D2D communication to provide strong latency performance; both scheduling options succeed in meeting the QoS requirements of the smart grid applications, which is due to the inherent nature of D2D communication. However, HRQ provides additional improvements in latency, as it proactively allocates communication resources every scheduling access period instead of scheduling resources for what has been buffered since the last scheduling access period.

One interesting observation in 4.4 is how the different smart grid applications have very similar latency performance when the random self allocation mechanism is used, while when HRQ is used, there is more variation in the message latencies among the different applications. It is believed that the lack of proactive scheduling in the random self allocation algorithm makes the latency performance across the different smart grid applications similar to one another, as their messages must always wait until the next SA period before they can be allocated resources. Conversely, when using HRQ, if a new message can make use of proactively allocated resources, the message can be transmitted in the same SA period that it was generated in. For the PMU application, which has the lowest data payload (592 bits), the latency is lowest, since it can fit all its data in the first SA period on a regular basis. Then, for the solar energy generation

forecasting application, its data payload was moderately large (1120 bits), and has an average communication latency that is between those of the PMU application and the DR application. The variance of the solar application is also higher than the other applications, which is believed to be caused by its particular payload size, where there is significant probability that the messages created could either be transmitted in the SA period they were created in, or in the following SA period. Finally, the DR application has the largest latency among all smart grid applications, but is still smaller than the latencies observed for the random self allocation scheduler. This application has the largest data payload (2000 bits), so it consistently needs to finish transmitting in the following SA period. However, due to the segmentation of application layer messages done at the RLC layer, some of the message can still be transmitted in the first SA period, meaning it has a lower payload to transmit in the second SA period, making it still faster than the random self allocation scheduled messages.

Concerning PDR performance, it can be seen in fig. 4.5 that for varying traffic intensities, the LTE scheduler fails to meet the QoS requirements for the smart grid applications. However, the PDR can be kept down to essentially 0 for low to moderate levels of network traffic when using the HRQ scheduler. The PDR starts to rise as the number of scheduling agents in the network approaches the number of orthogonal scheduling decisions available in the action space. The number of agents in the network reaches the number of orthogonal scheduling decisions when 6 auxiliary devices are injected in the network. This results from the inability of the HRQ algorithm to consistently self-organize the agent's scheduling decisions in the 10 s exploration time provided in this experiment, which is further discussed in Section 4.5.2. Naturally, the average PDR and uncertainty around the average PDR continues to increase as the number of agents in the network begin to exceed the number of actions available for the agents to take, as it becomes impossible to guarantee QoS with HRQ under such conditions. For example, if there are 12 D2D pairs each taking a unique action among the 12 available, if another D2D pair is introduced to the network, any action it chooses will be the same as what is already being taken. In fig. 4.5, this point of network saturation begins at 7 auxiliary devices. We further explored the effects of extreme network saturation by simulating 11 auxiliary devices at the network, which resulted in PDR performance falling below the QoS requirements of the archetypal smart grid applications under consideration. It should also be noted that the HRQ and LTE reliability performance begins to overlap, and as the level of traffic congestion becomes arbitrarily large, their PDR values are expected to become the same, as it would become impossible for either scheduling algorithm to avoid interference.

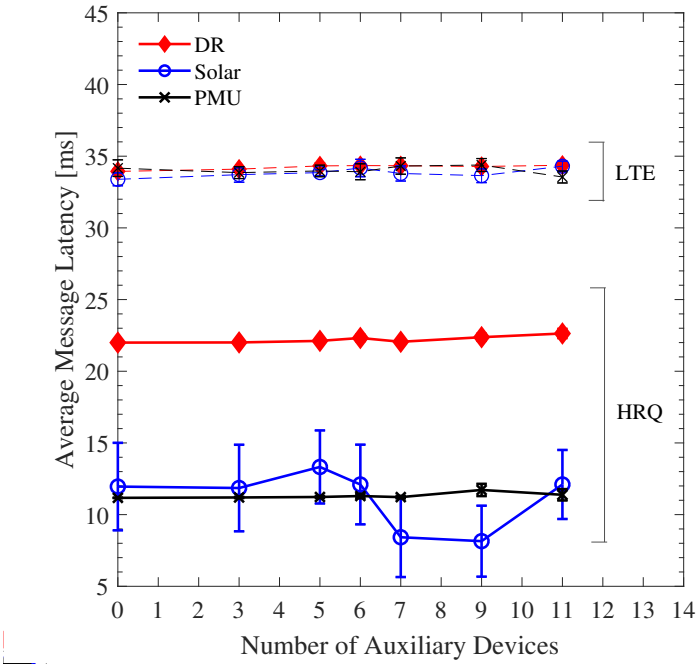

FIGURE 4.4: Average message latency for smart grid applications under varying traffic intensity.

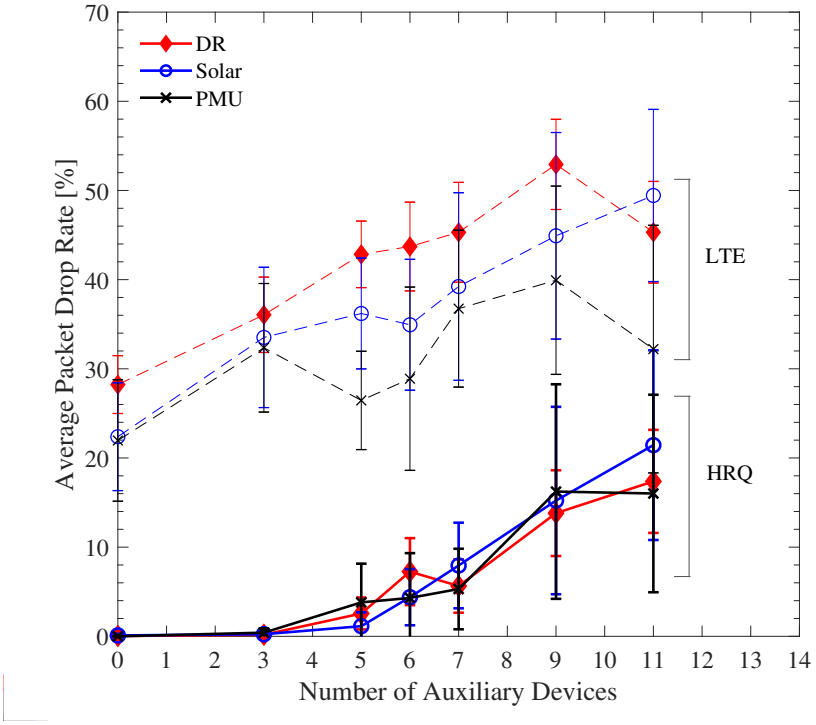

FIGURE 4.5: Average PDR for smart grid applications under varying traffic intensity.

4.5.2 Convergence of HRQ Algorithm

To demonstrate the ability for HRQ to reach an optimal policy, which corresponds to all agents making unique (and orthogonal) scheduling decisions, cumulative regret of all smart grid agents were aggregated and plotted with respect to time in fig. 4.6. The data points plotted in fig. 4.6 have a 1 ms time resolution, i.e. the cumulative regret was recorded after every elapsed millisecond-long subframe in the simulation. Regret is the difference between the reward of an action taken and the reward associated with the action that an optimal policy would have taken. Cumulative regret is the time integral of regret. Because the HRQ scheduler reward function has only two possible outputs, namely 1 and -1, every time an agent takes an action that results in a -1 (which happens when an agent makes the same scheduling decision as another agent), the regret is 2. Otherwise, the regret is 0. Thus, the point in the simulation where the smart grid agents are capable of self organizing their scheduling decisions so that they do not interfere with one another corresponds to the point in time where the cumulative regret of all agents stops increasing.

It can be observed that the HRQ algorithm is capable of converging to an optimal policy when the number of agents does not exceed the number of available orthogonal scheduling decisions in the algorithm's action space.

When there are 0 auxiliary devices, there are fewer possible actions for each agent that would result in interfering with another node. As such, the aggregate amount of negative rewards is lowest for this case. Since the cumulative regret reaches 0 at essentially the end of the exploration period, it is most likely the case that a 10 s exploration period is not necessary in low traffic conditions, and the algorithm could converge faster. This shows how the exploration period length is an important hyper parameter of our algorithm that will have a significant impact on its short-term performance. To avoid needing this hyper parameter, replacing epsilon greedy exploration with a more sophisticated policy such as a softmax based policy could be tried. A softmax function converts a set of numbers into a probability distribution whose density values are proportional to the relative differences in size of the input numbers. In our case, the set of numbers would be the action values in the Q-table that an agent can select at a given state. If no particular action value is significantly larger than any other, as it would in the beginning of the decision process, the softmax function would assign similar probabilities across all actions, thus encouraging exploration of the action space. However, as the agent learns which actions are best, the softmax naturally assigns higher probability that the historically good actions are chosen. Eventually, the action value of the optimal policy is so much larger than the alternatives, that the softmax distribution approaches a deterministic policy, thus ending the exploration of the agent. The large benefit to

this approach is that we do not have to arbitrarily decide when to stop exploring the environment, as it is now done autonomously by the agent. Nonetheless, a drawback to the softmax approach is that this policy only converges to a deterministic policy in the limit as the greedy action approaches infinity, so there will always be a small probability that the agent will continue to explore indefinitely, whereas in the approach we used with an exploration and exploitation phase, we are guaranteed to follow a truly deterministic greedy policy at a pre-specified point in time into the agent's life. It is expected that softmax would approach a near flat cumulative regret curve much faster than our approach, but our approach would achieve a truly flat curve faster than softmax. The difference between a "near flat" curve and a "truly flat" curve can be significant if the agent is enabling communication for a mission critical system, which can be the case in a smart grid context.

When 3 auxiliary devices are introduced to the network, the cumulative regret observed is similar in shape to that of the 0 auxiliary device case. The main difference is that the slope of its cumulative regret slope during the exploratory period is steeper for longer in the early period, then becomes roughly the same as the 0 auxiliary device case. In the first half second of the trials, fig. 4.6, all regret curves plotted appear to have the same steepness except for the 0 auxiliary device case. It is believed that this slope corresponds to the rate of regret experienced by an agent when it is still following an epsilon greedy policy, *and* its greedy action is failing to succeed in avoiding interfering with other agents. This is the point in time of the agent's operation where it is operating the poorest, because it is still taking random decisions, while also unable to perform well when it takes the greedy action. After 10 seconds when the policy becomes a deterministic "greedification" of the learned Q values, the slope still immediately decreases even when the agent has not yet converged to the optimal policy, as in the 6 auxiliary device case. This is because the agents no longer have to take random actions, and suffer their consequences. Returning to the comparison between the cumulative regret slopes of the 0 auxiliary device case to that of the 3 auxiliary device case, the point in time where they both have the same slope is believed to occur the moment that the 3 auxiliary scenario has discovered an optimal policy. This is because the 0 auxiliary device case is assumed to converge almost immediately, as it never had the initial very steep regret curve of all other trials, meaning its slope during the exploration phase is most likely induced solely by random actions taken by the agent. Under this assumption, it makes sense that the 3 auxiliary device case would converge when its cumulative regret slope is equal to that of the 0 auxiliary device case.

When 6 auxiliary devices are introduced to the network, the number of agents in the network matches the number of actions defined in the HRQ action space. In this case, the HRQ algorithm takes over 15 s to converge (recall that one subframe is 1 ms). This

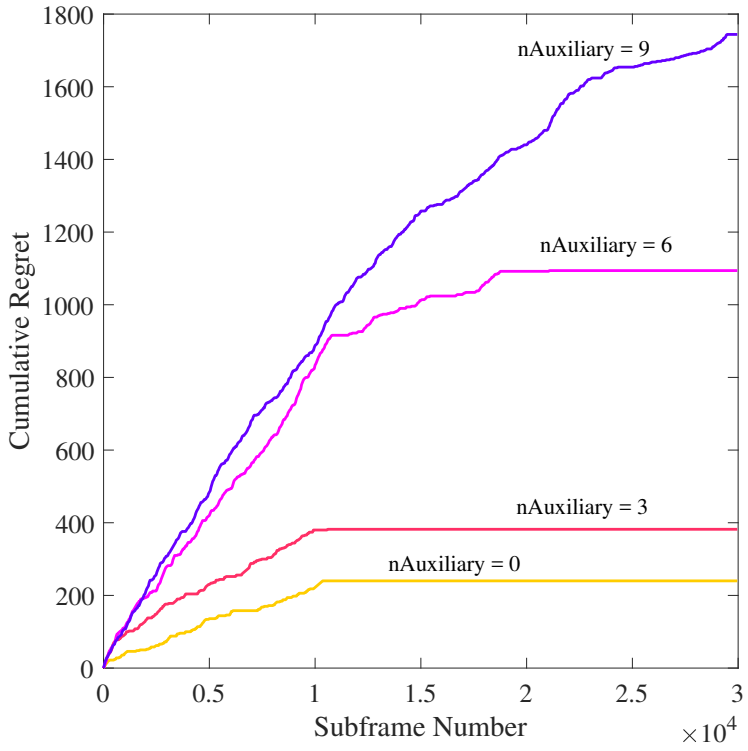

FIGURE 4.6: Total cumulative regret for HRQ scheduling algorithm for various numbers of auxiliary devices. Every subframe lasts one millisecond.

is why the average PDR was larger than 0% in fig. 4.5 even in cases where there were a sufficient number of actions for each agent to assign themselves a unique action.

When 9 auxilliary devices are introduced to the network, fig. 4.6 results suggest that a plateau in cumulative regret is about to be formed at the very end of the time axis. This is possible because the aggregate cumulative regret of the smart grid agents plotted does not include the cumulative regrets of the auxiliary agents. To elaborate, there is the possibility that the set of smart grid agents could all terminally select disjoint actions while the auxiliary agent's HRQ policies converge to taking actions (that are not all disjoint) among the remainder of the action space. If this happens, the cumulative aggregate regret of the smart grid nodes will stop increasing, but the auxiliary agents that are interfering with one another will still experience negative rewards at least some of the time. For example, say three auxiliary nodes have learned to share two disjoint resource sets; they could each take turns having a resource set all to themselves, while the other two share the other set, and experience interference.

4.5.3 Impact of scheduling technique on the performance of microgrid applications

Fig. 4.7 demonstrates the functionality of the application performing demand response activity due to both price signals and transformer overloading scenarios as described in Section 4.2.4. Fig. 4.7a plots the aggregate power curves of the households, and fig. 4.7b plots the transformer's power curves for multiple cases. The first case is where the demand response application is disabled (no DR), providing a baseline power curve that can be altered by the application. The next three cases correspond to the dynamics of the power system when demand response is active and facilitated by D2D LTE communications utilizing the LTE scheduling mechanism, HRQ, and an ideal scheduler (ideal data transfer). The ideal scheduler allows the smart grid devices to communicate under perfect channel conditions that ignore the effects of interference and noise. The power curve for the photovoltaic (PV) module and the real time price of electricity are plotted in fig. 4.7c and fig. 4.7d respectively. The transformer power flow curve is the difference of household aggregate demand less PV power generation. In cases where transformer power flow is positive, the distribution system is providing power to the microgrid; when it is negative, the power generated by the PV exceeds the aggregate household demand, and the surplus flows into the distribution system.

Concerning the price signaling dynamics of the three homes in the model (fig. 4.2), one home is programmed to have a C_{max} of 300 price units, whereas the other two homes are given C_{max} values of 400 price units. Based on this, one should expect power consumption to drop when the price curve increases, and fall back down again when the price decreases. Also, the transformer power flow exceeds $P_{limit} = 5000W$ shortly before the 6 s mark of the simulation, meaning the demand should decrease to keep the power flow through the transformer at or below $5000W$. As can be observed, the ideal data transfer and HRQ power curves do manage to behave in this manner, and are almost identical to one another. However, the power curve in the scenario where the application is being implemented with the LTE resource scheduling algorithm proves to deviate from the ideal case. For example, it can be seen at the 2 and 2.5 second marks, there are delays in the responsiveness of the households on the order of 100 ms when they are reacting to price changes. Such delays are not present in the other curves. This can only be a consequence of the higher PDR present when utilizing the LTE scheduler. These plots are intriguing in the sense that they display the ability of the integrated simulation platform to investigate how the performance of the communication network influences that of the power system.

In summary, this chapter introduced a multi-agent Q-Learning based resource allocation strategy targeted at reducing packet drop rates in LTE TM-2 D2D communication to

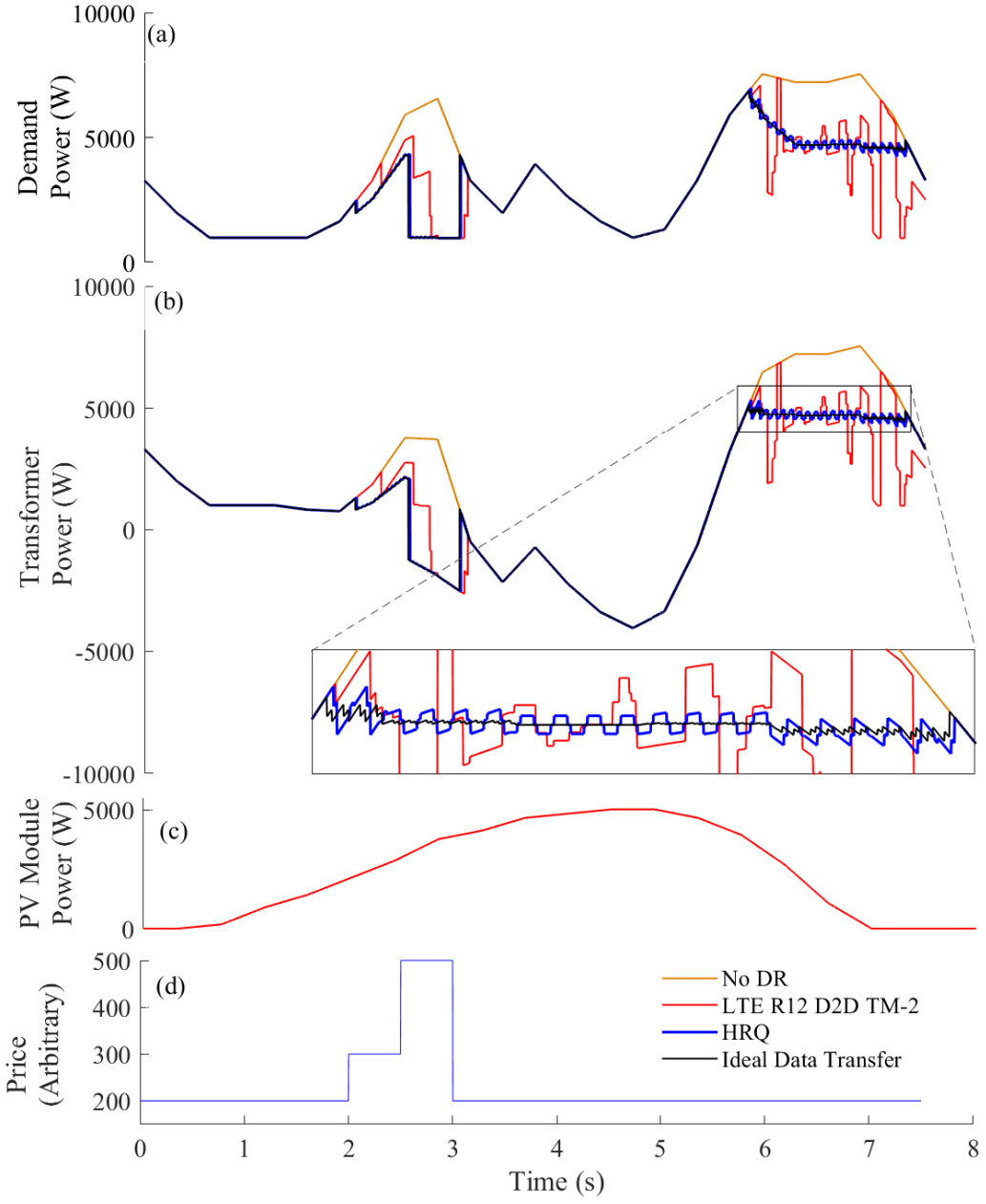

FIGURE 4.7: Demand response application performance for 3GPP, HRQ, and ideal mobile communication models: plots of a) Aggregate household demand; b) transformer power flow; c) microgrid PV generation; and d) price of electricity as a function of time.

increase its usability for smart grid applications. The QoS requirements and data traffic characteristics for various smart grid application archetypes were studied and modeled in an integrated Matlab/Simulink simulation environment to both test the performance of the scheduling mechanism and demonstrate the simulator's ability to design smart power systems and observe how the power and communications systems interact with one another. The performance of the HRQ scheduling algorithm was compared to that of the existing means of resource scheduling in TM-2 prescribed by the LTE standard. Our results show that HRQ outperformed the benchmark algorithm in both latency and PDR QoS metrics. Finally, a basic demand response application developed in the simulator was presented to demonstrate the impact the two different scheduling strategies have on its ability to properly manage transformer loading and price signaling activities. As a future work, it would be interesting to enhance the performance of HRQ with advanced exploration strategies and implement more sophisticated smart grid use cases.

Chapter 5

Conclusion

Smart grids use information and communication network technology to improve the performance of many different aspects of electric power systems. Such aspects include the economic efficiency of electricity markets; the dynamic stability of transmission and distribution networks; and the power system's capacity for the integration of renewable generation sources. LTE D2D enabled communication networks offer a means to the implementation of neighbourhood area network scale smart grid applications in remote locations without cell towers that interface users to a wide area network. This is the result of LTE D2D using licensed spectrum and having a globally standardized protocol architecture that permits distributed cellular resource scheduling, which allows for the QoS provisioning needed for mission critical smart grid applications.

Nonetheless, two barriers to the successful real-world use of LTE D2D technology for mission critical, neighbourhood area network scale smart grid applications have been identified at the research and development stage of the engineering process, and have been addressed in this thesis.

The first barrier is the lack of accessible simulation software for engineers to develop and test the feasibility of their D2D LTE enabled smart grid application designs. The ideal simulator would be able to have high fidelity for both the communication and power system networks, so the impact that the QoS of the communication network has on the power system dynamics can be fully understood.

The second barrier is the lack of a distributed resource allocation algorithm for LTE D2D communications that has been tailored to the needs of smart grid applications. On one hand, LTE TM-2 specified in Release 12 of the LTE standard a random self allocation strategy for transmitting devices, which results in high packet drop rates that are unsuitable for mission critical applications. This transmission mode was intended

for use in public safety applications, e.g. for communications among an emergency response team after a natural disaster, and prioritized energy efficient communications to conserve the battery life of devices. On the other hand, the LTE TM-4 specified in Release 14 of the LTE standard introduced a high reliability, low latency option for D2D communications tailored for vehicular communication. However, control messages must be processed every subframe for the resource allocation decisions to be made, as the resource allocation process of TM-4 requires historical channel use of nearby devices in the network. As such, TM-4 would require the communications device to perform more signal processing, etc., making it not ideal for use by IoT devices that may have small battery capacities, the very same that could be expected to be used in smart grid applications. Thus, a new transmission mode that can balance QoS performance with energy efficiency would be desirable for the needs of smart grid applications.

In this thesis, I addressed the first barrier by developing a co-simulation tool in Matlab/Simulink that combined the powerful LTE System Toolbox physical layer simulation software with Simscape Power Systems, a high fidelity power systems simulation package. To merge these two simulation tools, SimEvents was used to evoke event-triggered Matlab function calls from the Simulink environment in which the power system design is simulated. SimEvents thus acted as the primary orchestration mechanism of the co-simulator, allowing millisecond time advances in the Simulink solver to produce event triggers that evolve the discrete event based dynamics of the simulated communications network subframe by subframe, reproducing the desired co-simulation interface dynamics mathematically characterized by DEVS atomic models.

Also, the upper layers of the data plane protocol stack of the LTE standard had to be modeled in order to process application layer smart grid messages into transport blocks that can be input into LTE System Toolbox functions for modulation and encoding into electromagnetic waveforms. Upon studying the protocol stack of the LTE standard, it was observed that the PDCP layer was not required to be modeled in the co-simulation tool, as it deals with practical encryption and data header compression issues, which do not have a large impact on the flow of data through a network. However, the RLC layer, which handles data segmentation and concatenation of upper layer SDUs, effectively maps how many application layer messages can be fit into a single transport block of data sent in one subframe of time. As such, for a realistic simulation of the data flow from one device to another, the RLC layer was programmed, following LTE standards for implementing Unacknowledged Mode RLC reception and transmission entities. Also, the architecture of D2D MAC headers was taken into account, appending such control information fields prior to sending the packets to the physical layer processing functions of LTE System Toolbox.

The upper layer LTE network functionality and Simulink-Matlab interface was integrated with the physical layer LTE network simulation code previously developed by the NETCORE laboratory and tested with a simple Fault Location, Isolation, and Restoration application. It was observed that the power system dynamics were successfully triggering smart grid applications to send communications messages into the LTE network, while received messages resulted in the appropriate shifts in control signaling in the power system as expected.

I addressed the second barrier identified in this thesis by developing a multi agent reinforcement learning resource allocation algorithm that maximizes the reliability of the data transmissions through the D2D network. The design of the Q-learning tuple that characterizes this algorithm (HRQ) is compatible with the LTE Release 12 physical layer control and shared channel architectures. Thus, the inherent energy efficiency of the Release 12 D2D standard was kept, but the resource allocation algorithm was improved to boost its QoS performance to a more acceptable level. The algorithm ended up being considerably simple, where the action space of the Q-learning tuple divided the LTE resource grid into disjoint sets within which the transmitting node would allow itself to encode information. Should the observed CQI determined by the receiving node be below an acceptable threshold, the transmitting node would not be rewarded, but would otherwise be granted an optimal reward. Thus, by trial and error, the algorithm could determine which cellular resources should be used by a node without explicit knowledge of the scheduling decisions of the other users in the network.

HRQ resulted in significant improvements in the packet drop rates observed in the simulated network scenarios for low to moderate levels of traffic congestion, virtually eliminating packet losses while 20-40% of the data got corrupted when using the existing means for resource allocation prescribed by the LTE standard. A comparison of the performance of a real time demand response application while the nodes in the network were scheduling cellular resources using HRQ vs. random allocation was presented. The power curves of the microgrid for the random allocation case showed orders of magnitude higher deviations from the power curves realizable with an idealized communications network than what was observed for those while using HRQ.

The work presented in this thesis could be extended in many ways. Despite simply accepting that LTE Release 12 D2D communication consumes less power than Release 14, it would be an interesting study to compare the QoS-energy efficiency trade-off between the two versions before and after the upgrade of Release 12 to include HRQ. It would also be useful to know how successful the option of improving the energy efficiency of Release 14 architecture to be more suitable for IoT devices would be. Moreover, it would be interesting to adapt HRQ to V2X standards issued in Release 16 to see if it

can add value to the exciting field of vehicular communications in the next generation of mobile communication standards.

Another area of extended research could be exploring more advance methods for exploration vs. exploitation strategies in reinforcement learning for this particular algorithm. In this thesis, a slightly modified epsilon greedy approach was used, but this a rather naive approach to the exploration vs. exploitation dilemma. During the work conducted for this thesis, an original alternative to epsilon greedy was briefly tested, and it resulted in much faster convergence than epsilon greedy, but more work would need to be done to fully test and document this alternative.

Concerning the design of the Q-learning algorithm itself, future work could also be done to implement different reward functions and observe their performance. For example, the reward function and action space could be extended to introduce a "schedule nothing" action, and provide a reward of 0 should an agent take it. Perhaps this would improve the network capacity of the system.

Finally, scaling up the capabilities of the co-simulation tool developed to be able to simulate wide area communication networks would be useful. This would involve simulating the core network in addition to the communication interfaces between different eNBs. As of now, the co-simulator can only simulate neighbourhood area network scale smart grid applications, thus limiting the set of potential smart grid applications that it can test dramatically. Most of the smart grid applications that are concerned with the performance of the communication network are in the realm of wide area monitoring, protection, and control of a distribution network, which means the simulation tool developed would have to be scaled up to reach its full potential in academic and industrial communities alike. Moreover, further research on the scalability in terms of the number of agents operating within a D2D neighbourhood area network can also be performed, i.e. increasing the number of traffic flows that must select disjoint resources to avoid interference. This would require the dimensionality of the action space to increase (i.e. a finer resolution of resource sets to choose from), making it harder for HRQ to stabilize. Investigating how large the action space can get before tabular Q learning becomes an infeasible method, and then proposing ways to handle the increased number of agents would be desirable. For example, having agents shift to random self allocation schemes if their HRQ algorithm begins to operate poorly, or using action value function approximation methods to handle large action spaces could be tested to see if they can overcome scalability issues.

Bibliography

[1] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid — the new and improved power grid: A survey. *IEEE Commun. Surveys Tutorials*, 14(4):944–980, Fourth 2012. ISSN 1553-877X. doi: 10.1109/SURV.2011.101911.00087.

[2] M. Erol-Kantarci and H. T. Mouftah. Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues. *IEEE Commun. Surveys Tutorials*, 17(1):179–197, First quarter 2015. ISSN 1553-877X. doi: 10.1109/COMST.2014.2341600.

[3] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah. Delay critical smart grid applications and adaptive qos provisioning. *IEEE Access*, 3:1367–1378, 2015. ISSN 2169-3536. doi: 10.1109/ACCESS.2015.2466077.

[4] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke. A survey on smart grid potential applications and communication requirements. *IEEE Trans. Ind. Informat.*, 9(1):28–42, Feb 2013. ISSN 1551-3203. doi: 10.1109/TII.2012.2218253.

[5] F. A. Asuhaimi, J. P. B. Nadas, and M. A. Imran. Delay-optimal mode selection in device-to-device communications for smart grid. In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 26–31, Oct 2017. doi: 10.1109/SmartGridComm.2017.8340710.

[6] C. Kalalas, L. Thrybom, and J. Alonso-Zarate. Cellular communications for smart grid neighborhood area networks: A survey. *IEEE Access*, 4:1469–1493, 2016. ISSN 2169-3536. doi: 10.1109/ACCESS.2016.2551978.

[7] K. Shamganth and M. J. N. Sibley. A survey on relay selection in cooperative device-to-device (D2D) communication for 5G cellular networks. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 42–46, Aug 2017. doi: 10.1109/ICECDS.2017.8390216.

[8] X. Li, Q. Huang, and D. Wu. Distributed large-scale co-simulation for iot-aided smart grid control. *IEEE Access*, 5:19951–19960, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2753463.

[9] K. Mets, J. A. Ojea, and C. Develder. Combining power and communication network simulation for cost-effective smart grid analysis. *IEEE Communications Surveys Tutorials*, 16(3):1771–1796, Third 2014. ISSN 1553-877X. doi: 10.1109/SURV.2014.021414.00116.

[10] C. Steinbrink et. al. Simulation-based validation of smart grids - status quo and future research trends. In *Proceedings of 8th International Conference on. Industrial Applications of Holonic and Multi-Agent Systems*, pages 171–185, 2017. doi: 10.1007/978-3-319-64635-0_13.

[11] Giovanni Nardini, Antonio Virdis, and Giovanni Stea. Modeling network-controlled device-to-device communications in simulte. *Sensors*, 18(10), 2018. ISSN 1424-8220. doi: 10.3390/s18103551. URL https://www.mdpi.com/1424-8220/18/10/3551.

[12] Richard Rouil, Fernando J. Cintrón, Aziza Ben Mosbah, and Samantha Gamboa. Implementation and validation of an lte d2d model for ns-3. In *Proceedings of the Workshop on Ns-3*, WNS3 '17, pages 55–62, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5219-2. doi: 10.1145/3067665.3067668. URL http://doi.acm.org/10.1145/3067665.3067668.

[13] A. Asheralieva and Y. Miyanaga. Qos-oriented mode, spectrum, and power allocation for d2d communication underlaying lte-a network. *IEEE Transactions on Vehicular Technology*, 65(12):9787–9800, Dec 2016. ISSN 0018-9545. doi: 10.1109/TVT.2016.2531290.

[14] F. Guo, L. Herrera, R. Murawski, E. Inoa, C. Wang, P. Beauchamp, E. Ekici, and J. Wang. Comprehensive real-time simulation of the smart grid. *IEEE Transactions on Industry Applications*, 49(2):899–908, March 2013. ISSN 0093-9994. doi: 10.1109/TIA.2013.2240642.

[15] M. Manbachi, A. Sadu, H. Farhangi, A. Monti, A. Palizban, F. Ponci, and S. Arzanpour. Real-time co-simulation platform for smart grid volt-var optimization using iec 61850. *IEEE Transactions on Industrial Informatics*, 12(4):1392–1402, Aug 2016. ISSN 1551-3203. doi: 10.1109/TII.2016.2569586.

[16] C. Shum, W. H. Lau, K. L. Lam, Y. He, H. Chung, N. C. F. Tse, K. F. Tsang, and L. L. Lai. The development of a smart grid co-simulation platform and case study on vehicle-to-grid voltage support application. In *2013 IEEE International*

Conference on Smart Grid Communications (SmartGridComm), pages 594–599, Oct 2013. doi: 10.1109/SmartGridComm.2013.6688023.

[17] C. Shum, W. H. Lau, T. Y. Wong, T. Mao, S. H. Chung, C. F. Tse, K. F. Tsang, and L. L. Lai. Modeling and simulating communications of multiagent systems in smart grid. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 405–410, Nov 2016. doi: 10.1109/SmartGridComm.2016.7778795.

[18] Markus Mirz, Lukas Razik, Jan Dinkelbach, Halil Alper Tokel, Gholamreza Alirezaei, Rudolf Mathar, and Antonello Monti. A co-simulation architecture for power system, communication and market in the smart grid. *Complexity - Special Issue: Complex Optimization and Simulation in Power Systems*, pages 1–12, February 2018. doi: 10.1155/2018/7154031.

[19] I. Sychev, O. Zhdanenko, R. Bonetto, and F. H. P. Fitzek. Aries: Low voltage smart grid discrete event simulator to enable large scale learning in the power distribution networks. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, Oct 2018. doi: 10.1109/SmartGridComm.2018.8587538.

[20] P. Palensky, A. van der Meer, C. Lopez, A. Joseph, and K. Pan. Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems. *IEEE Industrial Electronics Magazine*, 11(2):6–21, June 2017. ISSN 1932-4529. doi: 10.1109/MIE.2017.2671198.

[21] Michele Garau, Emilio Ghiani, Gianni Celli, Fabrizio Pilo, and Sergio Corti. Co-simulation of smart distribution network fault management and reconfiguration with lte communication. *Energies*, 11(6), 2018. ISSN 1996-1073. doi: 10.3390/en11061332. URL https://www.mdpi.com/1996-1073/11/6/1332.

[22] G. O. Troiano, H. S. Ferreira, F. C. L. Trindade, and L. F. Ochoa. Co-simulator of power and communication networks using opendss and omnet++. In *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pages 1094–1099, Nov 2016. doi: 10.1109/ISGT-Asia.2016.7796538.

[23] Z. Pan, Q. Xu, C. Chen, and X. Guan. Ns3-matlab co-simulator for cyber-physical systems in smart grid. In *2016 35th Chinese Control Conference (CCC)*, pages 9831–9836, July 2016. doi: 10.1109/ChiCC.2016.7554916.

[24] R. Molina-Masegosa and J. Gozalvez. LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications. *IEEE Veh. Technol. Mag.*, 12(4):30–39, Dec 2017. ISSN 1556-6072. doi: 10.1109/MVT.2017.2752798.

[25] NASPI. PMU Data Quality: A Framework for the Attributes of PMU Data Quality and a Methodology for Examining Data Quality Impacts to Synchrophasor Applications. White Paper NASPI-2017-TR-002 PNNL-26313, North American SynchroPhasor Initiative (NASPI), 3 2017. Version 1.0.

[26] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2014. ISBN 0262028182, 9780262028189.

[27] D. Verma, S. Nema, and R. K. Nema. Implementation of perturb and observe method of maximum power point tracking in simscape/matlab. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 148–152, Dec 2017. doi: 10.1109/ISS1.2017.8389387.

[28] M. Fahmida, M. S. Alam, and M. A. Hoque. Matlab simscape simulation of solar photovoltaic array fed bldc motor using maximum power point tracker. In *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, pages 662–667, Jan 2019. doi: 10.1109/ICREST.2019.8644339.

[29] C. S. Salvador and A. A. L. Manalo. Planned renewable energy usage during power outage. In *2017 25th International Conference on Systems Engineering (ICSEng)*, pages 377–385, Aug 2017. doi: 10.1109/ICSEng.2017.79.

[30] O. Adeyemo, P. Idowu, A. Asrari, and J. Khazaei. Reactive power control for multiple batteries connected in parallel using modified power factor method. In *2018 North American Power Symposium (NAPS)*, pages 1–6, Sep. 2018. doi: 10.1109/NAPS.2018.8600686.

[31] A. Delavari, I. Kamwa, and P. Brunelle. Simscape power systems benchmarks for education and research in power grid dynamics and control. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pages 1–5, May 2018. doi: 10.1109/CCECE.2018.8447645.

[32] T. D. Castillo and J. R. Castro. Renewable energy source p-v conected to the grid through shunt active power filter based in p-q theory. In *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pages 1–5, Oct 2018. doi: 10.1109/ETCM.2018.8580274.

[33] J. Khazaei, M. Beza, and M. Bongiorno. Impedance analysis of modular multi-level converters connected to weak ac grids. *IEEE Transactions on Power Systems*, 33 (4):4015–4025, July 2018. ISSN 0885-8950. doi: 10.1109/TPWRS.2017.2779403.

[34] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed. A survey of device-to-device communications: Research issues and challenges. *IEEE Communications Surveys Tutorials*, 20(3):2133–2168, thirdquarter 2018. ISSN 1553-877X. doi: 10.1109/COMST.2018.2828120.

[35] A. Asadi, Q. Wang, and V. Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys Tutorials*, 16(4):1801–1819, Fourthquarter 2014. ISSN 1553-877X. doi: 10.1109/COMST.2014.2319555.

[36] Rohde and Shwarz. 1MA264: Device to Device Communication in LTE. White Paper 1MA264, Rohde and Shwarz, 9 2015. Version 0e.

[37] Release 12 sidelink pscch and pssch throughput - matlab & simulink. https://www.mathworks.com/help/lte/examples/release-12-sidelink-pscch-and-pssch-throughput.html. (Accessed on 02/19/2019).

[38] Andjela Ilic-Savoia. 5G Boot Camp Part 1: NR Technology Overview. Technical workshop, Keysight Technologies, 2018.

[39] IEEE Spectrum. 3GPP Release 15 Overview. Technical article, IEEE, 2020. URL https://spectrum.ieee.org/telecom/wireless/3gpp-release-15-overview.

[40] 3rd Generation Partnership Project. RAN Rel-16 progress and Rel-17 potential work areas. Technical article, 3GPP, 2019. URL https://www.3gpp.org/news-events/2058-ran-rel-16-progress-and-rel-17-potential-work-areas.

[41] Qualcomm. 5G NR based C-V2X. Technical presentation, Qualcomm, 2018. URL https://www.qualcomm.com/media/documents/files/5g-nr-based-c-v2x-presentation.pdf.

[42] Hao Ye and Geoffrey Ye Li. Deep reinforcement learning for resource allocation in V2V communications. *CoRR*, abs/1711.00968, 2017. URL http://arxiv.org/abs/1711.00968.

[43] M. Shih, H. Liu, W. Shen, and H. Wei. UE autonomous resource selection for D2D communications: Explicit vs. implicit approaches. In *2016 IEEE Conference*

on Standards for Communications and Networking (CSCN), pages 1–6, Oct 2016. doi: 10.1109/CSCN.2016.7785185.

[44] A. Asheralieva and Y. Miyanaga. An autonomous learning-based algorithm for joint channel and power level selection by D2D pairs in heterogeneous cellular networks. *IEEE Trans. on Commun.*, 64(9):3996–4012, Sept 2016. ISSN 0090-6778. doi: 10.1109/TCOMM.2016.2593468.

[45] S. Wen, X. Zhu, X. Zhang, and D. Yang. QoS-aware mode selection and resource allocation scheme for device-to-device (D2D) communication in cellular networks. In *IEEE International Conference on Communications Workshops (ICC)*, pages 101–105, June 2013. doi: 10.1109/ICCW.2013.6649209.

[46] X. Xiao, X. Tao, and J. Lu. A qos-aware power optimization scheme in ofdma systems with integrated device-to-device (d2d) communications. In *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5, Sep. 2011. doi: 10.1109/VETECF.2011.6093182.

[47] Guodong Zhang. Subcarrier and bit allocation for real-time services in multiuser ofdm systems. In *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, volume 5, pages 2985–2989 Vol.5, June 2004. doi: 10.1109/ICC.2004.1313079.

[48] G. Youjun, X. Haibo, T. Hui, and Z. Ping. A qos-guaranteed adaptive resource allocation algorithm with low complexity in ofdma system. In *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, Sep. 2006. doi: 10.1109/WiCOM.2006.62.

[49] H. Wang, G. Ding, J. Wang, S. Wang, and L. Wang. Power control for multiple interfering d2d communications underlaying cellular networks: An approximate interior point approach. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1346–1351, May 2017. doi: 10.1109/ICCW.2017.7962846.

[50] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang. A latency and reliability guaranteed resource allocation scheme for lte v2v communication systems. *IEEE Transactions on Wireless Communications*, 17(6):3850–3860, June 2018. ISSN 1536-1276. doi: 10.1109/TWC.2018.2816942.

[51] W. Sun, E. G. Ström, F. Brännström, K. C. Sou, and Y. Sui. Radio resource management for d2d-based v2v communication. *IEEE Transactions on Vehicular Technology*, 65(8):6636–6650, Aug 2016. ISSN 0018-9545. doi: 10.1109/TVT.2015.2479248.

[52] Y. Pan, C. Pan, Z. Yang, and M. Chen. Resource allocation for d2d communications underlaying a noma-based cellular network. *IEEE Wireless Communications Letters*, 7(1):130–133, Feb 2018. ISSN 2162-2337. doi: 10.1109/LWC.2017.2759114.

[53] D. H. Lee, K. W. Choi, W. S. Jeon, and D. G. Jeong. Two-stage semi-distributed resource management for device-to-device communication in cellular networks. *IEEE Transactions on Wireless Communications*, 13(4):1908–1920, April 2014. ISSN 1536-1276. doi: 10.1109/TWC.2014.022014.130480.

[54] S. Maghsudi and S. Stańczak. Joint channel allocation and power control for underlay D2D transmission. In *IEEE International Conference on Communications (ICC)*, pages 2091–2096, June 2015. doi: 10.1109/ICC.2015.7248634.

[55] S. Nie, Z. Fan, M. Zhao, X. Gu, and L. Zhang. Q-learning based power control algorithm for D2D communication. In *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, Sept 2016. doi: 10.1109/PIMRC.2016.7794793.

[56] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. *ACM Comput. Surv.*, 51(3):49:1–49:33, May 2018. ISSN 0360-0300. doi: 10.1145/3179993. URL http://doi.acm.org/10.1145/3179993.

[57] Gabriel A. Wainer. *Discrete-Event Modeling and Simulation: A Practitioner's Approach*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2009. ISBN 1420053361, 9781420053364.

[58] Derek Rowell. State-space representation of lti systems. 08 2002.

[59] Francois E. Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387261028.

[60] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph, and K. Pan. Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling. *IEEE Industrial Electronics Magazine*, 11(1):34–50, March 2017. ISSN 1932-4529. doi: 10.1109/MIE.2016.2639825.

[61] IEEE Task Force on Interfacing Techniques for Simulation Tools, S. C. Müller, H. Georg, J. J. Nutaro, E. Widl, Y. Deng, P. Palensky, M. U. Awais, M. Chenine, M. Küch, M. Stifter, H. Lin, S. K. Shukla, C. Wietfeld, C. Rehtanz, C. Dufour, X. Wang, V. Dinavahi, M. O. Faruque, W. Meng, S. Liu, A. Monti, M. Ni, A. Davoudi, and A. Mehrizi-Sani. Interfacing power system and ict simulators:

Challenges, state-of-the-art, and case studies. *IEEE Transactions on Smart Grid*, 9(1):14–24, Jan 2018. ISSN 1949-3053. doi: 10.1109/TSG.2016.2542824.

[62] K. Hopkinson, Xiaoru Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury. Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. *IEEE Transactions on Power Systems*, 21(2):548–558, May 2006. ISSN 0885-8950. doi: 10.1109/TPWRS.2006.873129.

[63] H. Lin, S. S. Veda, S. S. Shukla, L. Mili, and J. Thorp. Geco: Global event-driven co-simulation framework for interconnected power system and communication network. *IEEE Transactions on Smart Grid*, 3(3):1444–1456, Sep. 2012. ISSN 1949-3053. doi: 10.1109/TSG.2012.2191805.

[64] H. Georg, S. C. Müller, C. Rehtanz, and C. Wietfeld. Analyzing cyber-physical energy systems:the inspire cosimulation of power and ict systems using hla. *IEEE Transactions on Industrial Informatics*, 10(4):2364–2373, Nov 2014. ISSN 1551-3203. doi: 10.1109/TII.2014.2332097.

[65] H. Georg, C. Wietfeld, S. C. Müller, and C. Rehtanz. A hla based simulator architecture for co-simulating ict based power system control and protection systems. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 264–269, Nov 2012. doi: 10.1109/SmartGridComm.2012.6485994.

[66] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar. Integrated hybrid-simulation of electric power and communications systems. In *2007 IEEE Power Engineering Society General Meeting*, pages 1–8, June 2007. doi: 10.1109/PES.2007.386202.

[67] J. Bergmann, C. Glomb, J. Gotz, J. Heuer, R. Kuntschke, and M. Winter. Scalability of smart grid protocols: Protocols and their simulative evaluation for massively distributed ders. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 131–136, Oct 2010. doi: 10.1109/SMARTGRID.2010.5622032.

[68] W. Li, A. Monti, M. Luo, and R. A. Dougal. Vpnet: A co-simulation framework for analyzing communication channel effects on power systems. In *2011 IEEE Electric Ship Technologies Symposium*, pages 143–149, April 2011. doi: 10.1109/ESTS.2011.5770857.

[69] D. Anderson, C. Zhao, C. Hauser, V. Venkatasubramanian, D. Bakken, and A. Bose. Intelligent design real-time simulation for smart grid control and communications design. *IEEE Power and Energy Magazine*, 10(1):49–57, Jan 2012. ISSN 1540-7977. doi: 10.1109/MPE.2011.943205.

[70] V. Liberatore and A. Al-Hammouri. Smart grid communication and co-simulation. In *IEEE 2011 EnergyTech*, pages 1–5, May 2011. doi: 10.1109/EnergyTech.2011.5948542.

[71] D. Babazadeh, M. Chenine, K. Zhu, L. Nordström, and A. Al-Hammouri. A platform for wide area monitoring and control system ict analysis and development. In *2013 IEEE Grenoble Conference*, pages 1–7, June 2013. doi: 10.1109/PTC.2013.6652202.

[72] M. Wei and W. Wang. Greenbench: A benchmark for observing power grid vulnerability under data-centric threats. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2625–2633, April 2014. doi: 10.1109/INFOCOM.2014.6848210.

[73] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez. Control of power converters in ac microgrids. *IEEE Transactions on Power Electronics*, 27(11):4734–4749, Nov 2012. ISSN 0885-8993. doi: 10.1109/TPEL.2012.2199334.

[74] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale. The future renewable electric energy delivery and management (freedm) system: The energy internet. *Proceedings of the IEEE*, 99(1):133–148, Jan 2011. ISSN 0018-9219. doi: 10.1109/JPROC.2010.2081330.

[75] N. Cheng, N. Lu, N. Zhang, T. Yang, X. (. Shen, and J. W. Mark. Vehicle-Assisted Device-to-Device Data Delivery for Smart Grid. *IEEE Transactions on Vehicular Technology*, 65(4):2325–2340, April 2016. ISSN 0018-9545. doi: 10.1109/TVT.2015.2415512.

[76] R. Piacentini. Modernizing power grids with distributed intelligence and smart grid-ready instrumentation. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–6, Jan 2012. doi: 10.1109/ISGT.2012.6175653.

[77] S. Fey, P. Benoit, G. Rohbogner, A. H. Christ, and C. Wittwer. Device-to-device communication for smart grid devices. In *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–7, Oct 2012. doi: 10.1109/ISGTEurope.2012.6465751.

[78] S. Ci, J. Qian, D. Wu, and A. Keyhani. Impact of wireless communication delay on load sharing among distributed generation systems through smart microgrids.

IEEE Wireless Communications, 19(3):24–29, June 2012. ISSN 1536-1284. doi: 10.1109/MWC.2012.6231156.

[79] Y. Cao, T. Jiang, M. He, and J. Zhang. Device-to-device communications for energy management: A smart grid case. *IEEE J. Sel. Areas Commun.*, 34(1): 190–201, Jan 2016. ISSN 0733-8716. doi: 10.1109/JSAC.2015.2452471.

[80] C. Chen, J. Wang, F. Qiu, and D. Zhao. Resilient distribution system by microgrids formation after natural disasters. *IEEE Transactions on Smart Grid*, 7(2):958–966, March 2016. ISSN 1949-3053. doi: 10.1109/TSG.2015.2429653.

[81] H. Elkhorchani and K. Grayaa. Smart micro grid power with wireless communication architecture. In *2014 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, pages 1–10, Nov 2014. doi: 10.1109/CISTEM.2014.7077037.

[82] T. Shlebik, A. Fadel, M. Mhereeg, and M. Shlebik. The development of a simulation-based smart grid communication management system using matlab. In *2017 International Conference on Green Energy Conversion Systems (GECS)*, pages 1–7, March 2017. doi: 10.1109/GECS.2017.8066153.

[83] Casper Thule, Kenneth Lausdahl, Cláudio Gomes, Gerd Meisl, and Peter Larsen. Maestro: The into-cps co-simulation framework. *Simulation Modelling Practice and Theory*, 92, 04 2019. doi: 10.1016/j.simpat.2018.12.005.

[84] Alberto Falcone and Alfredo Garro. Distributed co-simulation of complex engineered systems by combining the high level architecture and functional mock-up interface. *Simulation Modelling Practice and Theory*, 97:101967, 08 2019. doi: 10.1016/j.simpat.2019.101967.

[85] LTE and ETSI. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) requirements for LTE Pico Node B (3GPP TR 36.931 version 9.0. 0 Release 9 Section 5.3.2.2.2), 2011.

[86] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001. ISBN 0130422320.

[87] Simplified model of a small scale micro-grid - matlab & simulink. https://www.mathworks.com/help/physmod/sps/examples/simplified-model-of-a-small-scale-micro-grid.html. (Accessed on 02/07/2019).

[88] S. Alwan, I. Fajjari, and N. Aitsaadi. Joint routing and wireless resource allocation in multihop LTE-D2D communications. In *2018 IEEE 43rd Conference on Local*

Computer Networks (LCN), pages 167–174, Oct 2018. doi: 10.1109/LCN.2018.8638241.

[89] F. A. Asuhaimi, S. Bu, and M. A. Imran. Joint resource allocation and power control in heterogeneous cellular networks for smart grids. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2018. doi: 10.1109/GLOCOM.2018.8647668.

[90] A. Laya, K. Wang, A. A. Widaa, J. Alonso-Zarate, J. Markendahl, and L. Alonso. Device-to-device communications and small cells: enabling spectrum reuse for dense networks. *IEEE Wirel. Commun.*, 21(4):98–105, August 2014. ISSN 1536-1284. doi: 10.1109/MWC.2014.6882301.

[91] L. Melki, S. Najeh, and H. Besbes. Radio resource allocation scheme for intra-inter-cell D2D communications in LTE-A. In *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1515–1519, Aug 2015. doi: 10.1109/PIMRC.2015.7343538.

[92] K. Shimotakahara, M. Elsayed, K. Hinzer, and M. Erol-Kantarci. High-reliability multi-agent q-learning-based scheduling for d2d microgrid communications. *IEEE Access*, 7:74412–74421, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2920662.

[93] The United States Department of Energy. Communications requirements of smart grid technologies. *DOE*, Jan 2012.

[94] P. Kong. Effects of communication network performance on dynamic pricing in smart power grid. *IEEE Systems Journal*, 8(2):533–541, June 2014. ISSN 1932-8184. doi: 10.1109/JSYST.2013.2260913.

[95] Rafael Asorey Cacheda, Daniel Castro Garcia, Antonio Cuevas, Francisco Javier Gonzalez Castano, Javier Herrero Sanchez, Georgios Koltsidas, Vincenzo Mancuso, Jose Moreno, Seounghoon Oh, and Antonio Panto. *QoS Requirements For Multimedia Services*, pages 67–94. 01 2007. doi: 10.1007/978-0-387-53991-1_3.

[96] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, Dec 2011.

[97] S. R. Firouzi, L. Vanfretti, A. Ruiz-Alvarez, F. Mahmood, H. Hooshyar, and I. Cairo. An iec 61850-90-5 gateway for ieee c37.118.2 synchrophasor data transfer. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5, July 2016. doi: 10.1109/PESGM.2016.7741393.

[98] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

[99] H. Takshi, G. Doğan, and H. Arslan. Joint optimization of device to device resource and power allocation based on genetic algorithm. *IEEE Access*, 6:21173–21183, 2018.

[100] S. Lee, J. Kim, and S. Cho. Resource allocation for noma based d2d system using genetic algorithm with continuous pool. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 705–707, 2019.

[101] R. Bansod, A. Shastry, B. Kumar, and P. K. Mishra. Ga-based resource allocation scheme for d2d communcation for 5g networks. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 748–752, 2018.

[102] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification. Technical Specification (TS) 136.321, 3rd Generation Partnership Project (3GPP), 04 2015. Version 12.5.0.

[103] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures . Technical Specification (TS) 36.213, 3rd Generation Partnership Project (3GPP), 04 2015. Version 12.5.0.

[104] C. C. Coskun and E. Ayanoglu. Energy- and spectral-efficient resource allocation algorithm for heterogeneous networks. *IEEE Trans. Veh. Tech.*, 67(1):590–603, Jan 2018. ISSN 0018-9545. doi: 10.1109/TVT.2017.2743684.