UNIVERSITY of RWANDA

Website: www.aceiot.ur.ac.rw
Mail: aceiot@ur.ac.rw

COLLEGE OF SCIENCE AND TECHNOLOGY

AFRICAN CENTRE OF EXCELLENCE IN INTERNET OF THINGS (ACEIoT)

Title:

DESIGN OF A RELIABLE SDN-BASED COMMUNICATION FOR DISTRIBUTION AUTOMATION IN SMART GRID

A dissertation submitted in partial fulfilment of the requirements for the award of Masters of Science degree in Internet of Things: Wireless Intelligent Sensor Network (WISeNet)

Submitted By:

S Damascene UWIZEYEMUNGU (Ref. No: 221000414)

Supervised by:

Dr. Didacienne MUKANYILIGIRA

August, 2024

DECLARATION

I; S Damascene UWIZEYEMUNGU, Master's student from African Center of Excellence in Internet of Things, at the University of Rwanda; declare that this research thesis is my original work and has never been presented anywhere in the world.

S Damascene UWIZEYEMUNGU,

Ref: 221000414

Signed:

Date: **19/08/2024**

BONAFIDE CERTIFICATE

This is to certify that this submitted Research Thesis work report is a record of the original work done by S Damascene UWIZEYEMUNGU (**Ref. No: 221000414**), MSc. IoT-WISNET Student at the University of Rwanda / College of Science and Technology / African Center of Excellence in Internet of Things, the Academic year 2020/2021.

This work has been submitted under the supervision of Dr. Didaciene MUKANYILIGIRA

Main Supervisor: **Dr. Didaciene MUKANYILIGIRA**

Date:

Signature

The Head of Masters and Training

Dr. James Rwigema

Signature:

Date:

Signature…………………

Abstract

The transition from traditional to smart grids and increasing complexity requires advanced communication frameworks to ensure reliable and efficient distribution automation (DA). Considering its capabilities to decouple the control plane from the data plane, the Software-Defined Networking (SDN) technology has emerged as a promising approach to address stringent requirements of smart grid communications, offering enhanced control, flexibility, robustness, and programmable network management. This thesis focuses on the integration of SDN into distribution automation systems, aiming to enhance the reliability and performance of communication networks within smart grids to ensure reliable communication between different components and devices of the grids by minimizing latency, enhancing fault tolerance, and optimizing resource allocation. The study reviewed challenges and issues in communication existing systems used in traditional grids such as lack of flexibility, latency, and bandwidth utilisation, and the study presented an SDN-based architecture tailored for smart grid applications, leveraging the centralized control and dynamic configuration capabilities of SDN to mitigate these issues. Through a combination of simulation-based analysis, it demonstrates how SDN can be leveraged to dynamically adjust communication paths in response to network failures, thereby maintaining consistent and reliable data exchange between distribution automation elements and devices. A comprehensive evaluation of the proposed solution has been performed in a simulation environment developed using Mininet and NS-3 simulation tools, emulating near real-world situations to analyze the key metrics revealing substantial improvements in network performance compared to traditional communication setups. The results and findings have indicated that the SDN-based platform can significantly increase the reliability and efficiency in communication, thus highlighting the paramount potential of SDN to support high demanding requirements of distribution automation in modern grids. However, the proposed solution provides a good foundation for further research of SDN's potential in developing the capabilities of next-generation power distribution systems.

Keywords: SDN, smart grid, distribution automation, communication system, reliability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI: Artificial Intelligence

AMI: Advanced Metering Infrastructure

AMR: Automatic Meter Reading

API: Application program interface

BAN: Body Area Network

BW: Bandwidth

CLI: Command Line Interface

DA: Distribution Automation

DER: Distributed Energy Resources

DSL: Digital Subscriber Lines

E2E: End-to-End

EAM: Enterprise Asset Management

EAN: Extended Area Network

FAN: Field Area Network

FFR: Fast Failure over Recovery

FTP: File Transfer Protocol

FTPS: File Transfer Protocol Secure

GUI: Graphical User Interface

GW: Gateway

HAN: Home Area Network

ICT: Information Communication Technology

IED: Intelligent Electronic Device

IEEE: Institute of Electrical and Electronics Engineers

IoT: Internet of Things

IP: Internet Protocol

IPE: Internetworking Proxy Entity

KPI: Key Performance Indicator

LAN: Local Area Network

MAN: Metropolitan Area Network

MANO: Management and Orchestration

MB: Mega Bit/ Byte

ML: Machine Learning

NAN: Neighbourhood Area Network

NFV: Network Functions Virtualisation

NFVI: Network Functions Virtualisation Infrastructure

NS-3: Network Simulator-3

ONOS: Open Network Operating System

PC: Personal Computer

PLC: Power Line Communication

QoS: Quality of Service

RTL: Round Trip Latency
Kbps: Kilo byte per second

SAM: Substation Asset Management

SDN: Software-Defined Networking

SFTP: Secure File Transfer Protocol

SG: Smart Grid

SGAM: Smart Grid Architecture Model

SSL: Secure Sockets Layer

TCP: Transmission Control Protocol/

TLS: Transport Layer Security

VM: Virtual Machine

WAN: Wide Area Network

CHAPTER I: INTRODUCTION

1.1 Introduction

The energy grid; widely regarded as one of the 20th century's greatest technological inventions; has revolutionized the way electricity is generated, transmitted, and distributed, making improvements for socio-economic development as well as people's lives [1] [2]. However, with the high increase in energy demand, the reliability and sustainability of the electric energy and its management became a serious challenge. Nowadays, power outages may seriously impact and cause fatal problems to human society concerning security, health, and the economy. Therefore, to mitigate these challenges, new information communication technologies have been integrated into electric power networks to improve their overall performance [3].

Rapid technological advancements have led to the evolution of the traditional power grid into a smarter and more efficient system known as the Smart Grid (SG) [1]. The SG integrates advanced communication and control technologies to enhance power distribution reliability, sustainability, and efficiency. Distribution automation, a critical aspect of SGs, involves the deployment of intelligent devices; such as IEDs, sensors, and actuators, and algorithms to monitor and control power distribution networks [2]. Considering data generated by various components involved in SGs, traditional communication systems are not able to ensure bidirectional communication efficiently. To ensure seamless and reliable communication within the Smart Grid, the implementation of a Software-Defined Networking (SDN) based communication system becomes essential. SDN is a paradigm that allows the separation of the control plane from the data plane in a network, providing centralized and dynamic control and programmability [3]. By decoupling the control logic from network devices, SDN enables more flexible and efficient network management, offering significant benefits for distribution automation in the Smart Grid.

To design a reliable SDN-based communication system for distribution automation, several key considerations need to be addressed. First, the system must provide high availability and fault tolerance to ensure uninterrupted communication in the Smart Grid [4] [5]. Fault-tolerant architectures and redundancy mechanisms play a vital role in critical infrastructure systems to minimize disruptions and maintain continuous operation [6] [7]. Moreover, the SDN-based communication system should prioritize security to protect against potential cyber threats and attacks [8]. Access control, encryption, and intrusion detection systems are essential security measures in securing communication networks in the Smart Grid [9] [10].

Moreover, scalability is a crucial factor in accommodating the increasing number of connected devices and the growing volume of data in distribution automation [11]. Scalable SDN architectures can provide the flexibility and capacity to handle the expanding network requirements [12]. Also, the system should support real-time communication and low-latency requirements to enable fast decision-making and efficient power distribution [13]. Low-latency communication is essential for real-time applications in the Smart Grid, allowing quick responses to changing grid conditions and improving overall system performance [14]. Lastly, interoperability with existing communication protocols and standards is vital to ensure seamless integration with legacy systems [15]. Integrating SDN with existing communication protocols in the SG facilitates interoperability and smooth transition [16].

By considering these factors and leveraging the benefits of SDN, a reliable communication system can be designed for distribution automation in the Smart Grid. Such a system will facilitate efficient monitoring, control, and optimization of power distribution networks, leading to improved reliability, reduced outages which may result in power blackouts if not properly dealt with; as well as their duration once they happen; and enhanced overall performance.

1.2 Background

A power smart grid or intelligent grid refers to an advanced electricity network that incorporates digital communication technology, sensing devices, and automation systems to improve the efficiency, reliability, and sustainability of electricity generation, transmission, distribution substations and lines, and consumption. Unlike traditional electrical grids, which use one-way communication from power plants to consumers, smart grids enable two-way communication between various components of the grid, from generation to consumer through transmission and distribution. The smart grid concept emerged as a response to the challenges and opportunities the evolving energy landscape presents [17]. Figure 1 illustrates the traditional architecture network of the power grid from power generation to consumption through transmission and distribution.

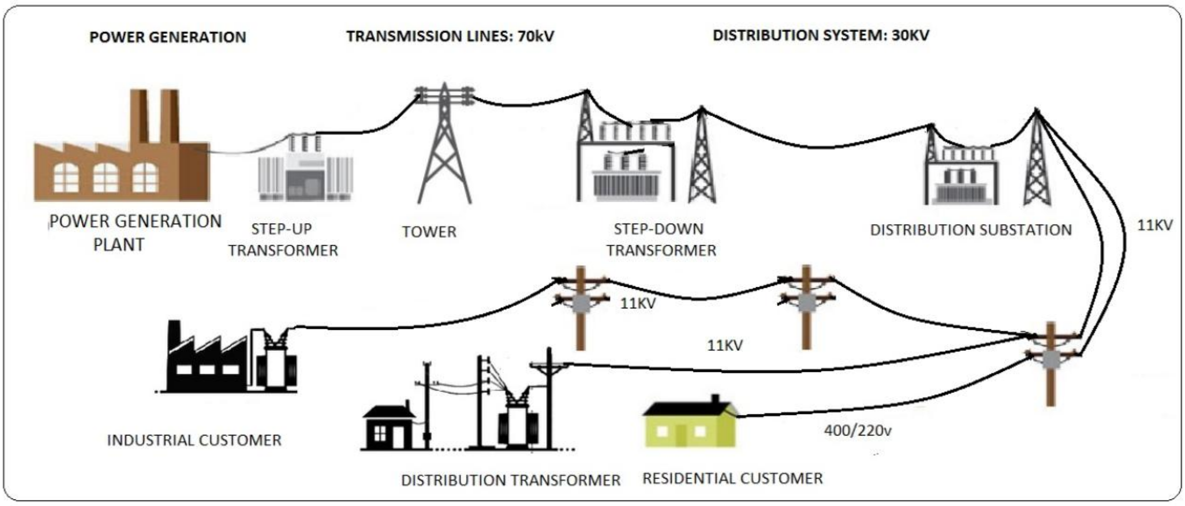


Figure 1: Illustration of the architecture of traditional power grid

Figure 2 shows the architecture of the Smart Grid while Figure 3 illustrates the interconnection between devices involved in SG network.

Figure 2: Architecture of Smart Grid (Cfr NIST)

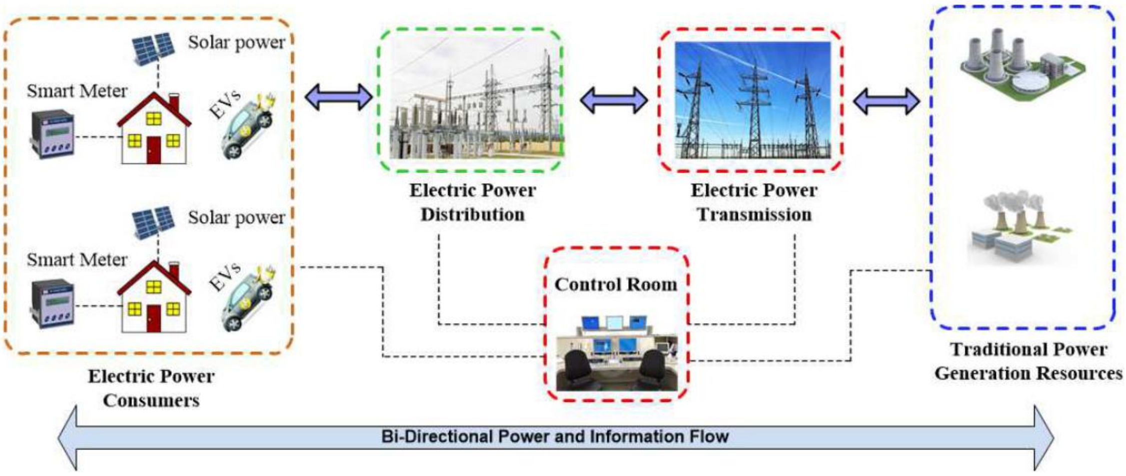


Figure 3: Integration of ICT into traditional power grid

To ensure seamless and reliable communication within the Smart Grid, the implementation of a Software-Defined Networking (SDN) based communication system becomes crucial. SDN offers a flexible and programmable network infrastructure by decoupling the control plane from the data plane, enabling centralized control and management of network resources [18]. Designing a reliable SDN-based communication system for distribution automation in the Smart Grid requires considering several key aspects. First, high availability and fault tolerance are essential to ensure uninterrupted communication. Fault-tolerant architectures and redundancy mechanisms play a vital role in achieving high system availability [19] [20] [50]. Figure 3 highlights the architecture model of SDN.

Security is another critical consideration in designing the communication system. With the increasing threat of cyber-attacks, protecting the Smart Grid communication infrastructure becomes paramount. Implementing robust security measures, such as access control, encryption, and intrusion detection systems, is essential to safeguard against potential vulnerabilities and attacks [21] [22]. Scalability is an important factor in accommodating the continuously growing number of connected devices and the increasing volume of data in distribution automation. The SDN-based communication system should be designed to handle large-scale networks efficiently and provide scalability to support future expansion [23].

Real-time communication and low-latency requirements are important for timely decision-making and efficient power distribution in the Smart Grid. Low-latency communication ensures that control commands and monitoring data are transmitted quickly, enabling rapid responses to the changing grid conditions [24] [25]. Finally, interoperability with existing communication protocols and standards is vital to ensure seamless integration with legacy systems. The SDN-based communication system should be designed to work harmoniously with different communication protocols and enable smooth communication between various components of the Smart Grid infrastructure [26].

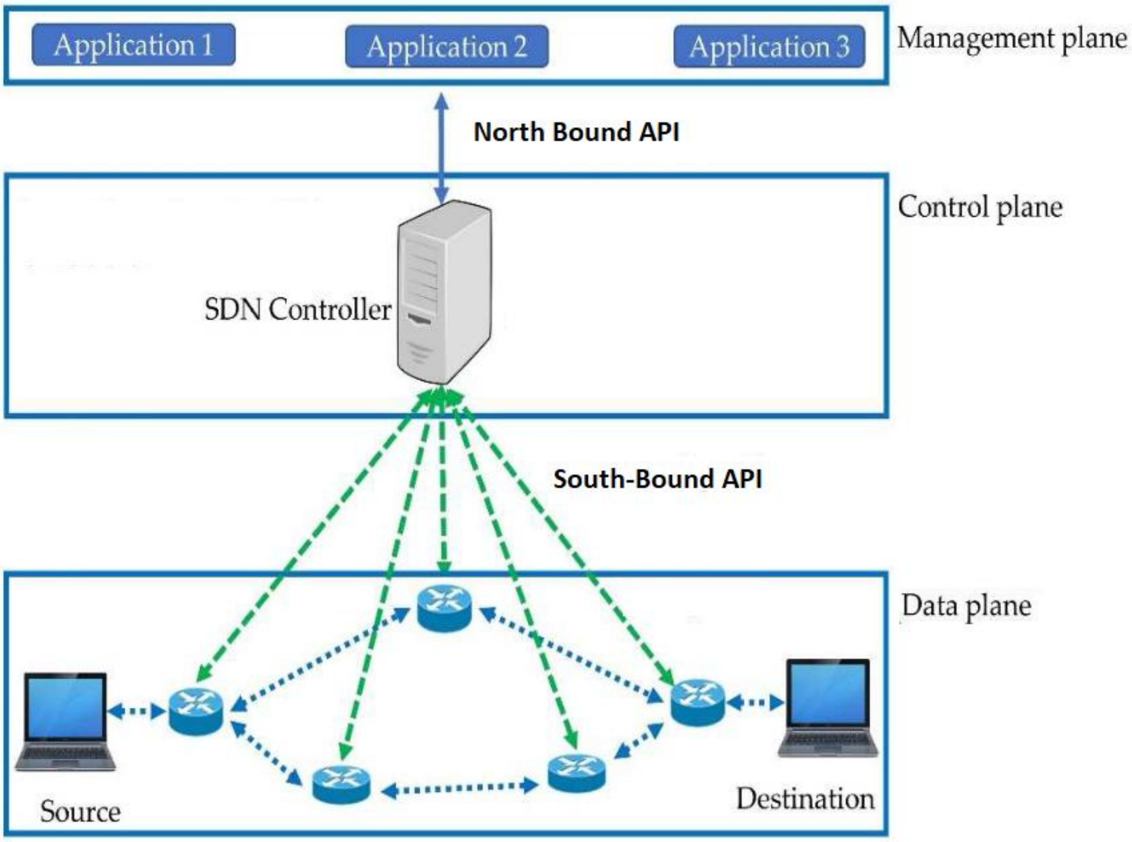


Figure 4: Architecture of Software-Defined Networking (SDN)

By addressing these considerations and leveraging the benefits of SDN, a reliable communication system is designed for distribution automation in the Smart Grid. Such a system will enable efficient monitoring, control, and optimization of power distribution networks, leading to improved reliability, reduced power outages, and enhanced overall performance.

1.3 Motivation

The motivation behind designing a reliable SDN-based communication system for distribution automation in the Smart Grid stems from the need to enhance the efficiency, reliability, and security of power distribution networks, which will be capable of achieving optimum performance of electrical distribution systems. Traditional power grids face challenges such as limited visibility, frequent power interruptions and outages for longer durations, and difficulties in real-time monitoring and control, which can result in inefficient energy distribution and compromised grid resilience [27].

By implementing distribution automation, which involves the deployment of new communication technologies, the utilities aim to overcome these challenges and improve overall grid performance. However, effective communication infrastructure is essential for the successful implementation of Smart Grid in general and distribution automation in particular. This is where the integration of SDN-based communication systems becomes of high importance and crucial. SDN offers several advantages that align with the requirements of distribution automation. The centralized control and programmability provided by SDN enable efficient management and orchestration of network resources, leading to enhanced reliability and flexibility [3]. Additionally, SDN facilitates real-time monitoring and control, enabling faster response to grid events and enabling optimized power distribution.

The motivation for designing a reliable SDN-based communication system lies in the potential benefits it offers. By leveraging SDN's capabilities, the system can provide high availability and fault tolerance, ensuring uninterrupted communication in the Smart Grid. This translates into reduced downtime and improved grid reliability. Moreover, robust security measures can be implemented to protect against cyber threats and attacks, ensuring the integrity and confidentiality of grid communication [21] [27]. Scalability is another motivating factor, as the Smart Grid can be expanded with the integration of other energy sources (such as renewable energy) and the proliferation of connected devices. SDN-based communication systems can handle the increasing data volumes and accommodate the growing number of devices, enabling efficient management of the evolving grid infrastructure [22]. The motivation also arises from the need for real-time communication and low-latency requirements in distribution automation. With SDN, it is possible to minimize communication delays, enabling fast decision-making, rapid fault detection, and efficient load balancing in the Smart Grid [24] [23].

Overall, the motivation to design a reliable SDN-based communication system for distribution automation in the Smart Grid stems from the desire to enhance grid performance, reliability, and security, by enabling efficient monitoring, control, and optimization of power distribution networks.

1.4 Problem Statement

The power grid infrastructure is undergoing significant transformations with the introduction of Smart Grid technologies. Distribution automation plays a crucial role in improving the efficiency, reliability, and sustainability of power distribution systems. However, the existing communication systems used in distribution grids often face challenges in meeting the dynamic and diverse requirements of modern smart grids. The problem lies in the limitations of traditional communication architectures, which are not designed to handle the complex and rapidly changing communication demands of distribution automation in a smart grid environment. Specifically, the problem to be addressed in this paper includes:

- How the communication system could be made flexible to adapt to the addition of newly connected devices and users as well as to related big data generated by these devices while keeping to standards, network efficiency, and performance? Scalability and network efficiency are key requirements for communication systems to be able to support the modern use of energy through Smart Grid implementation.

- What are the requirements and characteristics should communication systems have to provide high reliability and network availability in order to enable effective grid management, monitoring, and control thereby enhancing overall system performance? Reliability and network availability enable the distribution grid to ensure consistent and uninterrupted communication thus enabling effective monitoring, control, and optimization of the grid infrastructure.

- How will resilience and fault tolerance be guaranteed for distribution automation in the power grid? Resilience and fault tolerance enhance the reliability and availability of a communication system thus ensuring the continuity of critical operations of the grid even when faults or disruptions occur in the network.

- How would Software Defined Networking (SDN) be configured and implemented to enable and support automated big data collection needed for distribution automation systems? SDN-based communication systems offer centralized control, dynamic configuration, network resilience, enhanced security, and efficient traffic engineering capabilities, all of which contribute to improving the distribution automation of smart grids.

1.5 Objectives

The objective of the study is divided into two parts: general objective and specific objectives

1.5.1 General objective

To design and evaluate a reliable SDN-based communication system that ensures to enhancement of the overall performance of distribution automation within a smart grid environment. This system aims to provide robust communication channels between grid components and devices, optimize resource utilization, and allow rapid fault detection and recovery.

1.5.2 Specific objectives

- To develop a network architecture that leverages SDN to centralize control, improve visibility, and optimize routing within the smart grid's distribution automation network;
- Implement and evaluate the communication system's ability to support real-time data exchange and control signals required for effective grid management;
- Conduct system testing using simulation tools to evaluate the performance of the SDN-based communication system under various conditions, including normal operation, peak loads, and fault scenarios.

By achieving these goals, the thesis aims to contribute to the advancement of smart grid technologies and improve the efficiency and reliability of distribution automation systems through innovative SDN-based solutions.

1.6 Hypothesis

The hypothesis for designing a reliable SDN-based communication system for distribution automation in the Smart Grid is that implementing an SDN architecture with centralized control and programmable network elements can significantly improve the reliability, security, scalability, and efficiency of communication in the distribution automation system. By leveraging SDN technologies, such as dynamic network configuration, intelligent traffic management, and proactive fault detection and recovery, it is easily possible to enhance the overall performance and resilience of the Smart Grid.

This hypothesis assumes that by decoupling the control plane from the data plane and centralizing control in an SDN controller, will allow the achievement of better network visibility, flexibility, and adaptability. Moreover, the hypothesis posits that by employing

SDN-based communication, it is possible to effectively manage and coordinate various devices, systems, and protocols within the distribution automation system, leading to improved fault tolerance, security, and interoperability.

To validate this hypothesis, comprehensive testing and evaluation will be conducted, comparing the performance of the proposed SDN-based communication system with traditional communication approaches. Key metrics, such as network reliability, latency, scalability, security, and interoperability, will be assessed to determine the effectiveness and benefits of the SDN-based solution.

1.7 Scope

The scope of this study encompasses the design and implementation of a reliable Software-Defined Networking (SDN)-based communication system specifically tailored for distribution automation in the smart grid. The study focuses on developing a communication infrastructure that addresses various challenges faced by traditional approaches and provides a robust and adaptable solution for efficient distribution automation.

The scope of this study includes, but is not limited to, the following aspects:

i. *System architecture*: The study will focus on designing the overall system architecture of the SDN-based communication system for distribution automation. This involves determining the placement of SDN controllers, network switches, and other network elements, as well as defining the communication protocols and interfaces between different components.

ii. *Communication protocols*: The study will investigate and select suitable communication protocols for the SDN-based system, considering factors such as reliability, security, and interoperability. This may include protocols like OpenFlow, NETCONF, or RESTful APIs.

iii. *Network design and optimization*: The study will involve designing and optimizing the network topology and configuration to support efficient and reliable communication in the distribution automation system. This may include aspects such as network segmentation, traffic engineering, and load balancing.

iv. *Fault tolerance and recovery*: The study will explore mechanisms and strategies to ensure fault tolerance and quick recovery in case of network failures or disruptions. This may involve redundancy mechanisms, proactive fault detection, and fast rerouting techniques.

v. *Performance evaluation*: The study will evaluate the performance of the designed SDN-based communication system. This includes assessing metrics such as latency, throughput, reliability, and scalability under different operational conditions and traffic loads.

vi. *Integration with distribution automation components*: The study will consider the integration of the SDN-based communication system with other components of distribution automation, such as sensors, actuators, and control systems. It will explore the compatibility and interoperability requirements to ensure seamless communication and coordination.

The scope of this study does not include the design and implementation of other aspects of the Smart Grid, such as power generation, transmission, or specific automation applications beyond distribution automation. It specifically focuses on the reliable SDN-based communication system for supporting distribution automation in the Smart Grid context.

Overall, the study aims to develop a reliable and efficient SDN-based communication system that can enhance distribution automation in the Smart Grid, improve operational efficiency, and contribute to the overall advancement of Smart Grid technologies.

1.8 Significance of the Study

The design and implementation of a reliable Software-Defined Networking (SDN)-based communication system for distribution automation in the Smart Grid environment holds significant importance for the advancement of Smart Grid technologies. The study highlights the most significant benefits of using SDN-based communication to take advantage of the benefits offered by the electric Smart Grid.

A reliable SDN-based communication system enables efficient and real-time communication among Smart Grid devices involved in distribution automation. This enhances the overall automation process, allowing for quicker fault detection, isolation of the section where the fault is localized, and restoration of power distribution. It facilitates effective load balancing, fault management, and grid optimization, leading to improved reliability and resilience of the

Smart Grid. The use of SDN enables centralized control and programmability of the communication infrastructure, allowing for dynamic resource allocation and network optimization. This leads to enhanced grid efficiency, reduced power losses, and improved energy management. The SDN-based communication system enables flexible and scalable operations, accommodating changing grid conditions and demands effectively. SDN-based communication systems are inherently scalable, allowing for the addition and integration of new devices and technologies seamlessly. The designed system can accommodate future expansions, such as the integration of renewable energy sources, electric vehicles, and advanced grid control systems. It provides a future-ready infrastructure capable of supporting emerging smart grid technologies.

The study focuses on designing a reliable communication system that ensures fault tolerance, high availability, and minimal downtime. By implementing redundancy, failover mechanisms, and fault detection techniques, the SDN-based communication system can effectively mitigate communication disruptions and maintain reliable operation. Additionally, the integration of robust security measures protects against cyber threats and unauthorized access, ensuring the integrity and privacy of smart grid data.

CHAPTER 2: LITERATURE REVIEW

2.1. INTRODUCTION

The design and implementation of a reliable communication system with enough capacity to support distribution automation in a smart grid environment require a comprehensive understanding of past; as well as current; research and developments in the field. The literature review of the present work has been carried out mainly focussing on the smart grid domain, distribution automation systems, and SDN applications which provides us with an overview of key studies and works relevant to this project.

2.2. Smart Grid (SG)

In recent years, it has been realized that the traditional power grid infrastructures are no longer able to support the energy demand of the 21st century, which resulted in the emergency of the smart grid paradigm. The Smart Grid is well described by IEEE's Smart Grid Architecture Model presented in Figure 4 [47], and is defined as a modern electric power grid infrastructure for improved efficiency, reliability, and safety, through automated control and new communications technologies [47] with full-duplex communication and bidirectional power flow between utilities and customers [48]. In short, the Smart Grid is a fully automated electrical power network, whereby every user and node on the network can be monitored in real-time; and hence controlled; with a capacity of a bi-directional flow of energy and information [50].

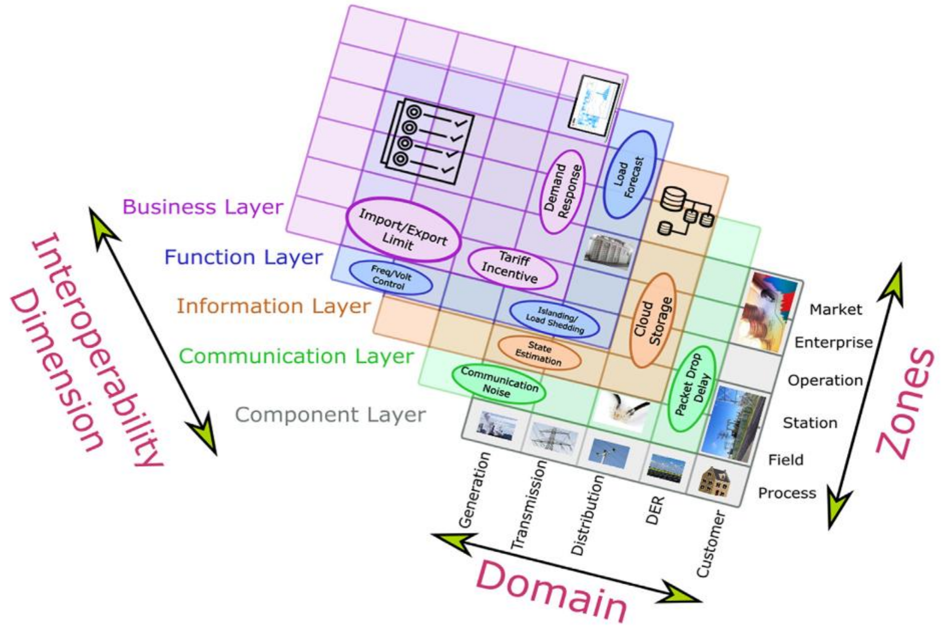


Figure 5: Smart Grid Architecture Model (SGAM) [47]

Yu, Xinghuo, and Yusheng Xue. in "Smart grids: A cyber–physical systems perspective." (*Proceedings of the IEEE* 104.5 (2016): 1058-1070), described challenges that traditional electric grids face: supply and use of energy in a more efficient, effective, economical, and environmentally sustainable way, before highlighting benefits of Smart Grid to mitigate them. The seamless integration and interaction of information sensing, processing, intelligence, and control as cyber systems with the power network infrastructure as the physical systems are essential to the success of SGs. Hence, the integration of enabling ICT and other cutting-edge technologies with massive power networks is crucial to the SG objectives in order to make electric energy generation, transmission, distribution, and usage more effective, efficient, affordable, and ecologically sustainable.

Farhangi, H. (2010) in "The path of the smart grid. IEEE Power and Energy Magazine, 8(1), 18-28”; defined the Smart Grid as a collection of all technologies, concepts, topologies, and approaches that allow the silo hierarchies of generation, transmission, and distribution to be replaced with an end-to-end, organically intelligent, fully integrated environment where the business processes, objectives, and needs of all stakeholders are supported by the efficient exchange of data, services, and transactions. Utilities started with the introduction of Automated Meter Reading (AMR); still a one-way communication system, shortly replaced by Advanced Metering Infrastructure (AMI); a two-way communication system to allow utilities to interact with customers’ service level. The emergence of AMI served as a transition to the emergence of the Smart Grid where the distributed command-and-control can fully be implemented over AMI backbone.

Gungor, Vehbi C., et al. "Smart grid technologies: Communication technologies and Standards." *IEEE Transactions on Industrial informatics* 7.4 (2011): 529-539; discussed the importance of communications system technologies in the implementation of the Smart Grid. The paper listed various communications technologies from ZigBee (a relatively low in power usage, data rate, complexity, cost of deployment) to Digital Subscriber Lines -DSLs (high-speed digital data transmission technology that uses the wires of the voice telephone network) via *mesh network* (a flexible network consisting of a group of nodes, where new nodes can join the group and each node can act as an independent router); Cellular Network Communication (good option for communicating between smart meters and the utility and between far nodes); and Powerline Communication -PLC (a technique that uses the existing powerlines to transmit high-speed data signals from one device to the other). In addition to insights into their related advantages and drawbacks, the authors dived into SG communications requirements, such as Security; System Reliability, Robustness, and Availability; Scalability; and Quality-of-Service (QoS) which govern the choice of communications technology.

2.3. Distribution automation

As one of the key components of the Smart Grid, Distribution Automation encompasses a suite of technologies and strategies that enable utilities to remotely monitor, control, and optimize the operation of their distribution networks. However, the nature of electrical distribution in traditional power grids is no longer able to support the increasing complexity of energy generation and consumption patterns, and the need for a more flexible, responsive, fault detection, and self-healing distribution system has become paramount. Hence, the notion of Distribution Automation.

The integration of Distribution Automation (DA) in SG provides significant advantages mainly in terms of reliability, efficiency, and resilience. Technological advancements in real-time data analytics, machine learning, as well as communication technologies, are the driving forces in the development of more complex DA systems, which enable the utility to monitor, control, and optimize the performance of electrical distribution systems in real-time.

Smith, J., Johnson, A., & Brown, C. (2018). "State-of-the-Art in Distribution Automation: A Review." *IEEE Transactions on Smart Grid.* This paper provides a comprehensive review of current technologies and methodologies in DA. It highlights the importance of advanced sensing, real-time data analytics, and renewable energy integration. The authors discuss various DA technologies, including automated switches, reclosers, and voltage regulators, and their impact on grid reliability and efficiency. The authors discussed advances in communication technologies, real-time data analytics, and machine learning as driving forces in the development of more sophisticated DA systems. However, challenges such as interoperability and optimal device placement; which are crucial for a successful DA deployment; could have been better addressed to fully realize the potential of DA in smart grids.

Kim, S., Lee, J., & Park, K. (2022). "Real-Time Data Analytics for Fault Detection and Diagnosis in Distribution Automation Systems." *IEEE Transactions on Industrial Informatics.* This paper investigates the application of real-time monitoring and data analytics for fault detection and diagnosis in DA systems. The authors propose machine learning-based algorithms and pattern recognition to accurately identify and localize faults, thereby improving outage response times and overall grid reliability.

Garcia, M., Martinez, L., & Lopez, P. (2020). "Optimal Placement and Sizing of Distribution Automation Devices in Smart Grids." *IEEE Transactions on Power Systems.* The authors of the study discuss and propose a methodology for determining the optimal placement and sizing of DA devices, depending on the network topology, load profiles, and fault types and scenarios. The study goes on to demonstrate that strategic deployment of DA devices can significantly improve grid reliability and efficiency.

2.4. Software-Defined Networking (SDN)

SDN has gained significant attention as a promising technology for communication systems in smart grids, particularly for distribution automation. SDN architecture provides a flexible and scalable framework for managing communication in Smart Grids by separating the control plane from the data plane, enabling centralized control and management of network devices and resources. [28].

SDN architecture provides enhanced reliable and available communication systems capable of ensuring efficient and uninterrupted communication between devices and systems involved in distribution automation [29] [30]. It also ensures strong security measures to protect the communication system from cyber threats and ensures the integrity, confidentiality, and availability of data [31] [32]. SDN allows the optimization of the network performance using its capability to ensure efficient resource allocation, bandwidth management, and traffic engineering. This helps to minimize latency, reduce congestion, and improve the overall efficiency of data transmission [33] [34]. Support real-time communication: Design the communication system to support real-time communication requirements of distribution automation. This includes minimizing communication delays, enabling fast data transmission, and facilitating rapid decision-making and control actions [35] [36].

Enabling a scalable and flexible communication system is another advantage guaranteed by an SDN-based communication system to enable the network to accommodate the growing number of devices and data volume in the Smart Grid. This involves the use of scalable SDN architectures, dynamic resource allocation, and flexible network configurations [37] [38]. Also, the system should enable seamless integration and interoperability between different devices, systems, and communication protocols within the Smart Grid. This allows for efficient communication and coordination among distributed automation components [39] [40].

Guo, S., Li, H., & Han, Z. (2018). Software-defined networking for smart grid: a comprehensive survey. IEEE Transactions on Industrial Informatics, 14(12), 5296-5306. This survey paper provides a comprehensive overview of the applications and benefits of SDN in the smart grid context. It discusses the challenges and opportunities of using SDN in various aspects of the smart grid, including distribution automation. The study highlights the advantages of SDN in terms of flexibility, scalability, and fault tolerance.

Wang, Z., & Lu, Z. (2018). A survey on communication technologies for smart grid systems. IEEE Transactions on Industrial Informatics, 14(5), 2083-2095. This survey paper provides an overview of communication technologies used in smart grid systems. It discusses various communication protocols, including SDN, and their applications in the smart grid domain. The study emphasizes the need for reliable and efficient communication solutions for distribution automation in the smart grid.

Wei, S., Yang, D., Wu, Z., & Zeng, P. (2019). Communication technologies for distribution automation in smart grid: A survey. IEEE Access, 7, 110781-110793. This survey paper provides an overview of communication technologies for distribution automation in the smart grid. It discusses various communication protocols and technologies used in distribution automation systems, such as wireless communication, power line communication, and SDN. The study highlights the benefits and challenges of these technologies and discusses their applications in distribution automation.

Yao, J., Yang, H., & Wang, B. (2019). Communication network architecture design for distribution automation in smart grid. Energies, 12(19), 3700. This paper focuses on the communication network architecture design for distribution automation in the smart grid. It discusses the requirements and challenges of communication networks in distribution automation and proposes a communication network architecture based on SDN. The study emphasizes the benefits of using SDN for dynamic network management and efficient communication in distribution automation.

Huang, L., Hu, Y., Ning, H., & Zhou, Z. (2019). SDN-based communication architecture for distribution automation in smart grid. IEEE Access, 7, 36349-36358. This paper proposes an SDN-based communication architecture specifically designed for distribution automation in the smart grid. It presents a comprehensive framework that integrates SDN principles into the communication infrastructure, enabling efficient and reliable communication among smart grid devices. The study highlights the advantages of SDN, such as centralized control and dynamic resource allocation, for distribution automation applications.

Wuttidittachotti, P., & Min, G. (2019). An SDN-based communication architecture for smart grid distribution automation. Energies, 12(18), 3496. This paper presents an SDN-based communication architecture for smart grid distribution automation. It proposes a novel architecture that leverages SDN principles to enable efficient and reliable communication among smart grid devices. The study highlights the advantages of SDN in terms of network management, traffic engineering, and fault recovery, contributing to the overall reliability of distribution automation in the smart grid.

CHAPTER 3: RESEARCH METHODOLOGY

3.1. Methodology

The methodology for designing a reliable SDN-based communication system for distribution automation in the Smart Grid, considering the fact that it is impossible to carry out tests in the grid and access to physical devices for the experiment, only simulation will be used, involves the following steps:

3.1.1. Requirement Analysis:

Identify the specific requirements and objectives of the distribution automation system in the Smart Grid. This includes understanding the communication needs, network architecture, scalability requirements, security considerations, and integration with existing distribution automation components.

3.1.2. Literature Review:

Conduct a comprehensive literature review to gather information on SDN-based communication systems, distribution automation technologies, and relevant simulation tools and models. This step helps identify best practices, existing simulation frameworks, and potential challenges in designing a reliable SDN-based communication system for distribution automation.

3.1.3. System Design:

Based on the requirements analysis and literature review, design the architecture and components of the SDN-based communication system. Define the network topology, select appropriate SDN controllers and switches, and design communication protocols based on the specific requirements identified in the earlier stage.

3.1.4. Simulation Tool Selection:

Identification and selection of a suitable simulation tool or framework for simulating the designed SDN-based communication system has been done taking into consideration of factors such as ease of use, availability of relevant modules and models, support for SDN simulation, and compatibility with the chosen network architecture and protocols.

It is in that regard that, the following tools have been selected:

NS-3: It is a discrete-event network simulator used to simulate and study network protocols and complex communication systems, allowing users to model various types of communication systems, including wired, wireless, and hybrid networks [59], [60]. NS-3 supports emulation, which enables simulated networks to interact with real-world systems, and provides visualization tools like PyViz that enable live visualisation and NetAnim for network topologies and events animation. NS-3 is an open-source software designed to work on Linux, Mac, and Windows. Furthermore, NS-3 supports numerous network protocols, namely IPv4 and IPv6; TCP and UDP; and Ethernet and Wireless.

The NS-3 tool advances the simulation time by processing events from an event queue, where each event has a well-defined and scheduled execution time, and the simulator progresses by executing events in chronological order. The simulator executes the scheduled events, updating the simulation state as it progresses. The simulator automatically stops when there are no more events in the queue or when a "*stop*" event is encountered.

Taking into consideration above mentioned features and capabilities provided by the tool, we adopted to use NS-3 simulator in our project as we aimed to test communications under predefined conditions such as link failure, propagation delays and jitter, and network reliability and resilience. Furthermore, being an open-source, NS-3 is suitable for academic research, providing possibilities like the addition of new SDN modules, customisation of technology generation, integration with external tools like Python scripts, OpenFlow-based SDN switches, and trace analysers.

Mininet: A network emulation tool running real Linux network software, Mininet allows the creation of a virtual network on a single machine which is commonly used for testing and experimenting SDN and related technologies. An open-source software, Mininet provides a Python-based API to enable users to create, test, and control networks programmatically [61]. It is also equipped with MiniEdit, a GUI application used for an intuitive drag-and-drop interface serving for designing network topologies, which makes it very suitable for SDN simulation, where it is required to create hosts, switches, routers, and links that connect them to SDN controllers like ONOS [62] [63]. An illustration of a simple Python script and a MiniEdit snapshot are shown in Figure 21.

Combined with MiniEdit, its Graphical User Interface (GUI), Mininet provides a robust and user-friendly platform required for simulating networks in a virtual environment. It is also more effective for low-cost SDN simulation and testing on a single machine.

The choice of Mininet emulator was made based on the score and objective of our project, which is to investigate communication resilience and failure over behaviour. The Mininet tool enabled us to simulate SDN controller failure, redundant paths, and switches' behaviour in the circumstances of an outage.

ONOS: Open-source software operating system serving as an SDN controller that offers scalability, high availability, and performance [62]. ONOS is a prevailing SDN controller able to provide the ability to control and manage networks using GUI, CLI, and API. Figure 22 illustrates an ONOS configuration file sample and a snapshot of graphical instances.

A powerful SDN controller designed for the needs of modern networks, ONOS offers scalability, high availability, and flexibility in managing carrier, enterprise, and data center networks, which makes it the most widely used tool for SDN projects. As controller, ONOS integrates SDN applications and manages switches and hosts [63]; its selection for our project was taken considering ONOS advantage to provide low-latency and better throughput in OpenFlow-based simulation [64].

3.1.5. Network Modelling and Configuration:

Develop a detailed network model of the SDN-based communication system using the selected simulation tool. Configure the network parameters, including SDN controller settings, switch configurations, link characteristics, and traffic patterns, to accurately represent the real-world scenario.

3.1.6. Simulation Scenarios and Experiments:

Define a set of simulation scenarios and experiments to evaluate the performance and reliability of the designed SDN-based communication system. This may include varying network loads, introducing fault scenarios, and testing different fault tolerance mechanisms. Design experiments to assess factors such as latency, throughput, scalability, network resilience, and fault recovery time.

3.1.7. Simulation Execution and Data Collection*:*

Run the simulation experiments using the selected simulation tool. Collect relevant performance metrics, network statistics, and other data to analyze the behavior and performance of the SDN-based communication system under different scenarios. Ensure that sufficient data is collected to enable a comprehensive evaluation of the system's reliability and efficiency.

3.1.8. Data Analysis and Evaluation:

Analyse the simulation results and evaluate the performance of the designed SDN-based communication system based on the collected data. Compare the results with the defined requirements and assess the effectiveness of the system in achieving the objectives of distribution automation in the Smart Grid. Identify any performance bottlenecks, limitations, or areas for improvement.

3.1.9. Documentation and Reporting:

Document the simulation methodology, system design, simulation experiments, results, and analysis. After a successful test, a comprehensive report summarizing the methodology, findings, and recommendations for further improvement or research will be prepared and compiled. Such a document could be useful for public and private institutions such as utility companies and regulatory bodies, and individuals to disseminate the knowledge and potential benefits of the designed SDN-based communication system.

By following this methodology, utilizing simulation-based approaches, the study ensures a controlled and repeatable environment to design and evaluate a reliable SDN-based communication system for distribution automation in the Smart Grid.

CHAPTER 4: SYSTEM ANALYSIS, DESIGN, IMPLEMENTATION AND EVALUATION

4.1. Requirement analysis

As the smart grid is an integrated system, it is worthful to be noted that each requirement in a specific smart grid operability layer; as described in the IEEE's smart grid architecture model [Figure 5]; may impact requirements in other layers and components. Hence, while designing requirements for a smart grid communication network, it is important to study an end-to-end view of the smart grid architecture. Figure 6 illustrates this interoperability layer interdependency across the smart grid components that are most prevalent in the distribution of electricity, highlighting the architectural elements that need to be considered.

Starting with the business layer, the main purpose of a distribution grid is to reliably, safely, effectively, and efficiently deliver electricity to customers. The required specifications for a communication network for distribution automation in a smart grid must start with the high-level requirements, taking into consideration the smart grid objectives. In Smart Grids, the energy and information flow are bi-directional, and consequently, these requirements need to take into account how the location and flow of information will impact the flow of electricity.

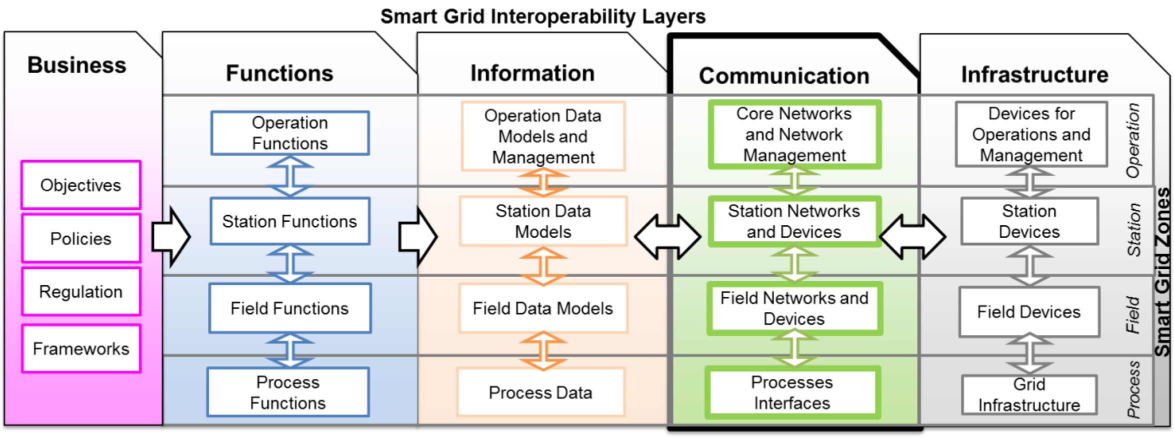


Figure 6: Interdependency between elements of the Smart Grid Architecture Model [57]

The current infrastructure of the power grids, their layout, and their operating environment greatly influence communication requirements and they may limit the options available for selection of communication systems. This project takes into consideration two approaches in order to meet our initial objectives.

Firstly, the development of a communication network capable of supporting a reliable smart grid system; which can accommodate existing Enterprise Asset Management (EAM) systems; with more focus on designing the communication layer to support and enable the functions layer to meet the power grid objectives. Secondly, the implementation of the network functions in the communication network design will also serve to improve the overall communication reliability of the grid for electricity distribution.

4.2. Communication Networks

The development of communication networks for distribution automation in a smart grid involves designing, implementing, and managing the infrastructure necessary for real-time monitoring, control, and optimization of power distribution. Moreover, the development of network communications should consider a system that will enable enterprise asset management (EAM).

The EAM system analyses grid asset data and uses derived information to predict possible threats in the relatively short, medium, and long term which helps in planning and scheduling adequate preventive maintenance. Since the EAM systems greatly depend on data acquisition and analytics, the transmission of these data will require a reliable network with minimum latency and sufficient bandwidth due to their massive amount and sensitivity, although some of the data are not necessarily processed in real-time. Table 1 illustrates the generally recommended end-to-end (E2E) communication latency, bandwidth, and reliability for smart grid systems.

Systems	**Examples**	**BW [Kbps]**	**Latency [ms]**	**Reliability [%]**
Mission critical	Substation Automation, Distribution Automation, Overhead Line monitoring, and Wide Area Situational Awareness.	9.6 -1500	15 -200	99-99,9999
Systems with lower data volume	Distribution Management, Home Energy Management, Distributed Energy Resources, AMI, and Outage Management.	9.6 -100	100– 2000	99-99,99
Systems with higher data volume	Demand Response Management, Electrical Vehicles, EAM and Meter Data Management	56 and above	2 and above	99-99,99

Table 1: Recommended communication requirements for smart grid systems [45].

The above recommendations for communication requirements, clearly show that communication networks to support EAM systems should ensure that communication reliability is kept in the range of 99% - 99.99%. This means that the designed communication network for the EAM system must ensure that all data generated by the grid devices reach their destination within the aforementioned range, that is at a minimum 99% of the time, and no data should be lost during transfer. The recommendations also highlight the need for sufficient bandwidth to efficiently accomplish data transfer of massive data samples from various grid equipment and devices.

The IEC 61850; which is an international standard that defines a communication protocol for intelligent electronic devices (IEDs) used in power utility automation systems such as distribution automation to enable interoperability and seamless communication between various devices in substations and power grids standard, specifies the types of messages to be used in substations communication in accordance with required E2E latency and the sizes of typical messages [46]. Table 2 illustrates the message types along with their size and their corresponding E2E latency requirements.

Message Type	**Required E2E Latency**	**Typical Message Size**
Fast messages	< 3 ms - 100 ms	1 bit
Medium speed messages	< 100 ms	1 bit -16 bits
Low-speed messages	< 500 ms	16 bits – 1024 bits
Raw data messages	< 3 ms - 10 ms	12 bits – 18 bits
File transfer functions	> 1000 ms	512 bits – 200 kilobits

Table 2: Message types used in substation communication [46].

Furthermore, knowing that the data in EAM systems are generally collected and transferred in bunches, the functions of data transfer should be taken into account for the system to function properly. In this regard, data from grid equipment and devices; which are constantly monitored by sensors attached to them; are stored for a relatively short period in storage facilities near collection points before the transmission to a centralized data center, eventually equipped with processing applications. Figure 8 illustrates the typical data flow in the EAM system deployed in distribution automation.

We note that network communication requirements must be defined in such a way that they meet all Smart Grid Architecture Model (SGAM) operability layers to be able to design a network with all capabilities required to handle data acquisition. Figure 7 illustrates the elements per layer to be considered for designing requirements for an EAM system.

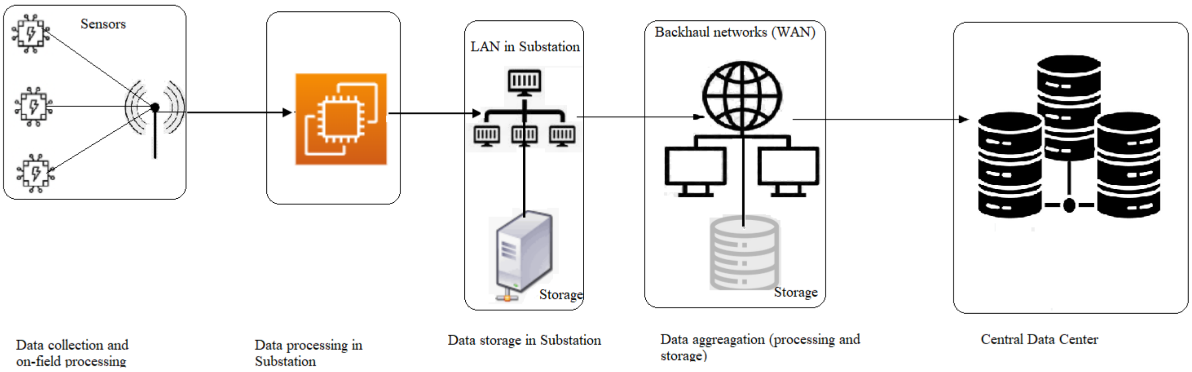


Figure 7: Data flow illustrating substation data acquisition

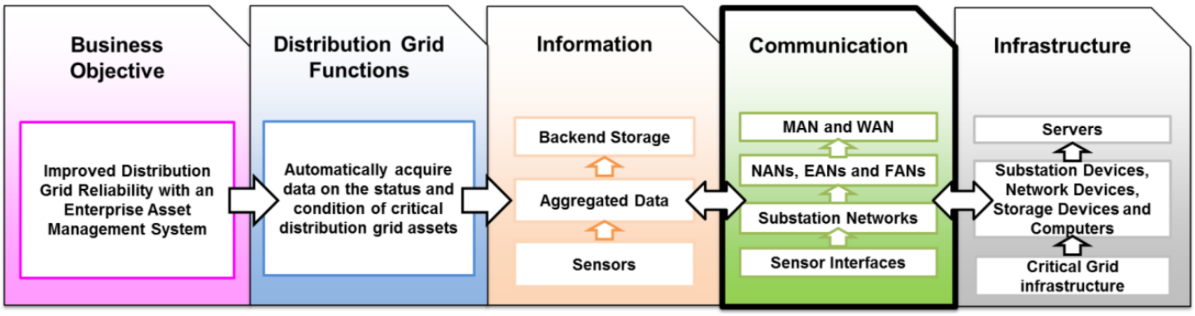


Figure 8: A smart grid architecture for an EAM system in a distribution grid.

The data collected by monitoring sensors attached to substation equipment are normally stored in various formats, with text files a basic one; used in capturing long strings of data; supported by substation technologies. After collection and temporary edge storage, the data need to be transferred from one device to another (i.e., from the substation to the control center). To efficiently transfer these files collected from field devices, the FTP (File Transfer Protocol); a standard network protocol used for transferring files between a client and a server over a TCP/IP network by providing a simple and reliable method for exchanging files over the Internet or within a local network; is considered a suitable option as it provides the possibility for transferring files of large size with improved reliability.

Moreover, for security purposes, FTP could be enhanced by the use of either FTP over Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) for File Transfer Protocol Secure (FTPS) or FTP over Secure Shell (SFTP) to offer ways to transfer the files more securely with relatively improved low latency. Additionally, the design needs to consider communication requirements for devices supporting data collection functions in order to accommodate the bidirectional communication for both electricity and data, whereby adoption of appropriate protocols.

The network should be designed in such a way that it provides the possibility to integrate an EAM system into a distribution grid, giving devices receiving sensor readings communication interfaces. Any part of the grid can be informed by sensor technologies, but it is important to carefully take this information into account to prevent data overload. Figure 9 shows the wireless sensor network in the substation.

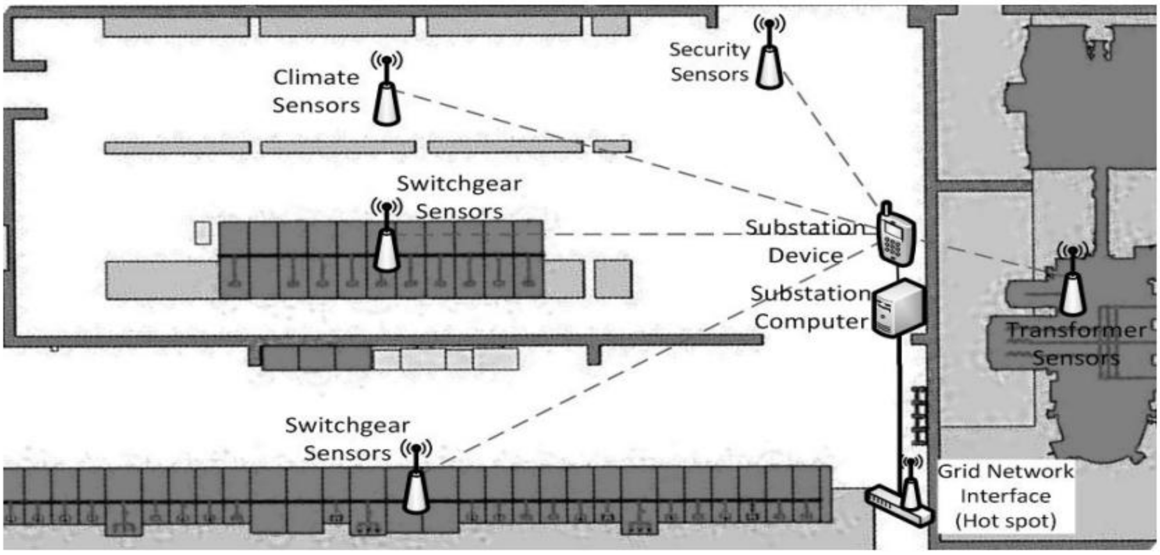


Figure 9: A smart grid architecture for an EAM system in a distribution grid.

To connect substation sensor networks to the NANs' router which directly connects to the distribution grid's MAN, network topologies need to be well-defined as each distribution network has its particularity. To deal with physical design constraints, the topologies must be developed to ensure reliable communication with sufficient backup flow pathways and adequate backup communication infrastructure. Figure 10 and Figure 11 present a suggestion for the NAN topology.

All routers in this NAN are linked to the main router in the Gikondo and Kiyovu Substations which are Primary Substations, forming a star topology. All routers have a minimum of three connections to the network thanks to a direct connection between each one and a router installed in a nearby substation. The two main substation routers must be connected to a backhaul network using TCP/IP-based Ethernet networks that can allow FTP file transfers as well as control and monitoring messages.

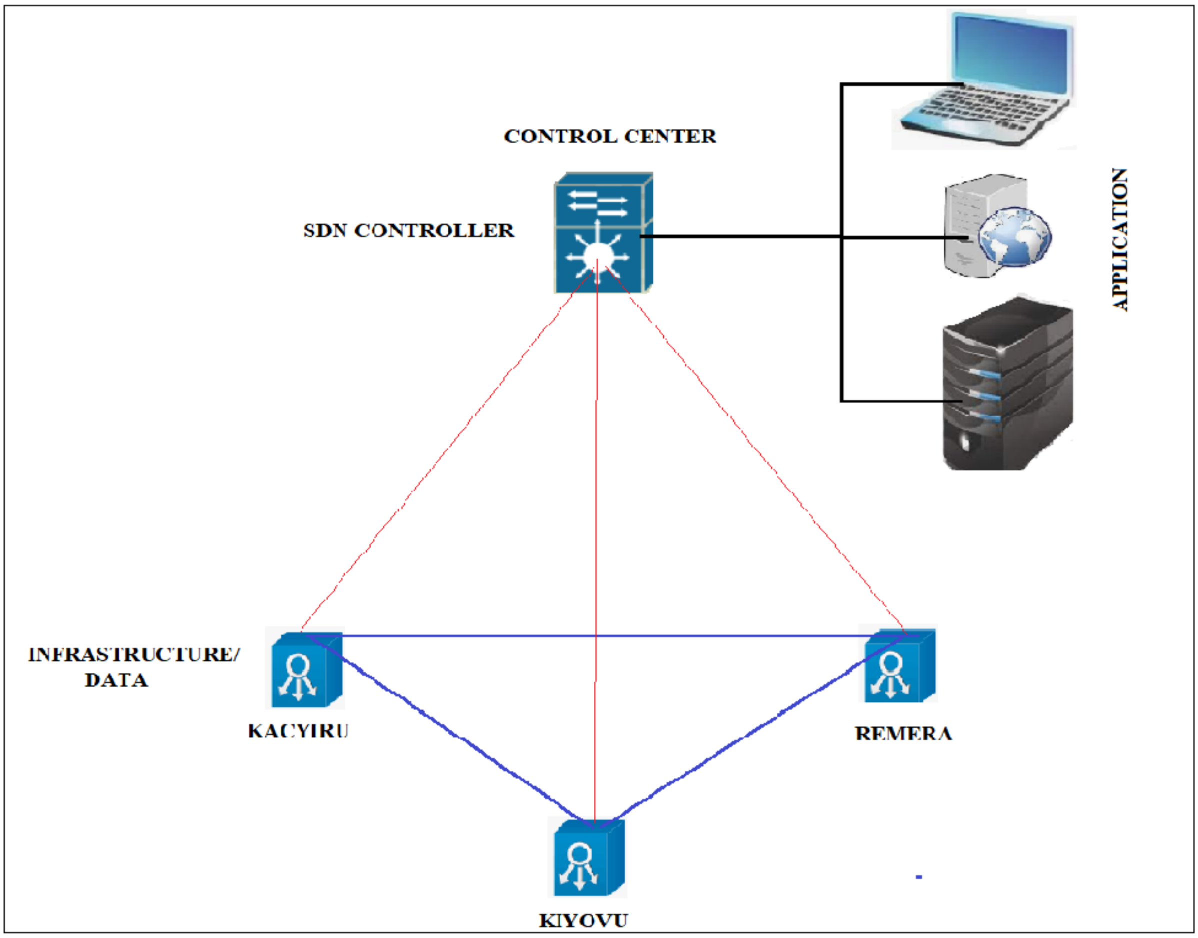


Figure 10: Proposed network topology for the Distribution Grid section.

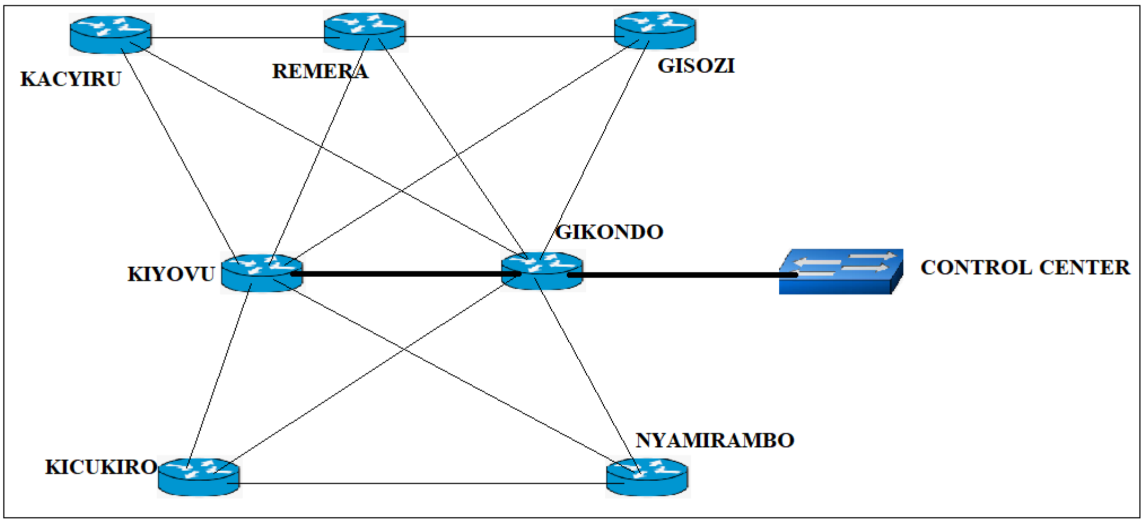


Figure 11: Proposed Network Diagram Interconnection of Sub-Stations

When creating a communication network for distribution grids, communication dependability is crucial. The reliability of the network's communication can be negatively affected by the current hardware-based approaches for constructing distribution grid communication networks, which demand significant capital expenditure. Its design, assessment, and deployment must be efficient while requiring the least amount of physical network setup and configuration, which calls for a flexible development platform. To meet the goals of the modern power grid, smart grids must facilitate two-way energy and information exchanges within the distribution grid. Figure 12 shows how communication reliability and power supply reliability are interdependent in a smart grid.

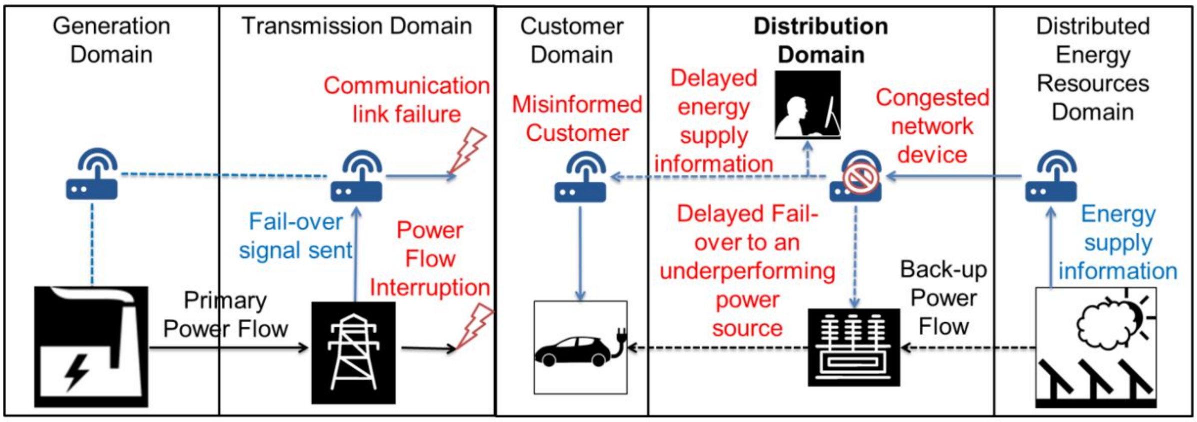


***Figure 12: Relationship between the reliability of power supply and communication network*.**

Considering the above setup, when a power supply interruption occurs, it is immediately detected by a sensor in a transmission substation and the disruption triggers a failover signal to another substation in a distribution grid, which enables it to switch over to an auxiliary power supply in accordance with the network configuration. However, a malfunctioning network device; generally, a switch; or network medium; results in a delay in the transfer of status information as well as command signal, resulting in a switch-over delay or failure causing power disruption. To improve the overall reliability of the power grid, mechanisms need to be put in place that reduce the probability of communication failures, delays, and mean time to repair communication problems.

To increase communication reliability in smart distribution grids, networks must integrate features that decrease the likelihood of failures at the communication source, destination, and communication channel. Failure of communication hardware or software on devices that send or receive data is the primary reason for communication issues at the source or destination of a message. The grid failover function would be ineffective if the transmission substation attempts to send the failover command but discovers that its network interface is unavailable. The data transmitting device connected to the power plant equipment should therefore be configured to store these data sets until its connection to the communication network is re-established. It should then reattempt to transmit these data sets. When any device that is supposed to receive status information from the power plant is disconnected from the network, transmissions will fail and the data from the transmitting device must be stored as well. Since many of these devices rely on limited amounts of embedded memory for storage, there is a risk that these devices might run out of storage space, which will result in data loss.

To overcome this problem, in-network data buffering functions must be included in the network design to provide additional storage capacity. These functions will act as temporary data repositories that take over the data transmission responsibilities from the source devices. In the network, data buffering will also only work for smart grid functions that make use of non-time-sensitive messages. For time-sensitive grid functions it may be advisable to automatically switch over data transmissions to back-up devices as soon as a communication problem is detected. If these switchovers can be performed proactively, it can reduce the risk of destination device unavailability during critical moments of data communication while simultaneously reducing the dependence on human intervention for these functions to execute. A backup power supply's data-transmitting equipment should be configured to store data sets until a connection to the communication network is established and then try to transmit them again. In-network data buffering features must be included in the network design to prevent data loss when any device gets disconnected from the network. To mitigate the risk of the destination device being unavailable during crucial times of data transfer, automatic switchovers to backup devices should be applied for time-sensitive grid tasks as soon as a communication issue is identified.

Network functions are required to provide network administrators with the ability to identify and remotely resolve network link and node failures or allow them to switch over traffic flows to backup flow paths. Multiple link and node failures can be caused by natural disasters or as part of attacks on the network, resulting in subsections of the communication network becoming isolated from the core networks. This would prevent the devices in this domain from exchanging any messages with other domains, and reduce the impact on time-sensitive grid functions.

To maintain some degree of grid functionality, distribution grid communication networks must be designed to maximize the available communication resources.

Communication network functions must provide automated failover methods and work to reduce the effects of any communication failures on the execution of grid operations to accomplish this. The network must be put through a variety of simulated communication failure scenarios to measure the interruptions to the established EAM data collection functionalities. Messages need to get to their destinations in time to perform the necessary grid function before it's too late, to reduce the possibility of communication delays. Communication delays in smart distribution grids can be caused by two factors: the distance a packet must travel in terms of the number of hops through the network's nodes and the amount of time each packet spends at a node before it is transferred. To reduce the probability of communication delays, network functions must ensure the use of the shortest flow paths between sources and destinations. Additionally, underperforming network devices can lead to increased time spent at a node.

Several factors may cause communication delays in smart distribution grids. However, the following two are the most critical to the E2E latency: the number of hops a packet needs to go through across the network to reach the final destination and the amount of time each packet spends at a node for processing before it is transmitted to the next hop. The adoption of the shortest flow channels between sources and destinations is required by network functions to decrease potential communication delays. Increased time spent at a node can also be caused by underperforming network components. Network devices can experience packet transfer delays due to overloading, which can be identified through real-time communication network monitoring and historical traffic trend analysis. Packet prioritization functions can help prioritize critical packets, but in some cases, the amount of network traffic may exceed the device's buffer capacity, resulting in dropped packets and retransmission.

Network functions need to be flexible, automated, and configured to address this issue without disrupting smart grid functions. Communication problems can be addressed by diverting network traffic away from the problem nodes and links, identifying alternative flow paths, and testing problematic devices and channels. Automated network functions can be used to address many of these communication problems, but network administrators still require a global view of the network configuration and status. Remote network control functions are required to allow them to perform repair tasks remotely, thereby eliminating the need to travel to field devices for repairs. In large-scale smart distribution grid communication networks, the scalability of a repair process is of particular interest due to the difficulty of locating a problem with multiple network devices and links.

4.3. System Design

Smart grids can be made better with the use of communication architectures based on SDN, Network Functions Virtualization (NFV), and Machine-to-Machine (M2M). The combination of these three technologies greatly and positively impacts modernizing and optimizing smart grids, enabling them to meet the growing demands for reliable, efficient, and sustainable energy distribution. A novel communication network design that is focused on supporting distribution grids is provided, and a suggested procedure to be followed when thinking about putting this architecture into practice using a virtual networking platform is also presented.

The functional architecture here proposed is designed with the electrical power distribution domain in mind, but it can also be used in other smart grid domains. The distribution grid infrastructure, which includes equipment, buildings, and electricity infrastructure, is represented by the data plane which is the process area. The process zone includes devices connected to the distribution infrastructure, such as sensors and actuators which are normally connected to field devices through copper wires which serve to the signals transfer.

Devices in a distribution grid's field zone translate signals into data so that they can be transferred using communication networks. These gadgets could be standalone computers, laptops, portable devices, or integrated devices. Internal substation networks (LANs), NANs, EANs, and FANs will be used by these devices to communicate with one another as well as with other station zone components. Then, by offering interfaces to the grid's backhaul networks, the NFVI in the station zone will connect the field components of the distribution grid to the MANs or WANs of the grid. The NFVI that hosts the domain-level SDN switches and controllers for the distribution grid's fundamental communication networks is located in the operating zone, and VMs hosted on this NFVI also house the M2M Backend services. The NFV MANO also functions in the operation zone, where it oversees the provisioning, updating, resource management, and termination of all VMs on readily accessible NFVI in the network connections. Figure 13 illustrates the architecture of the proposed solution.

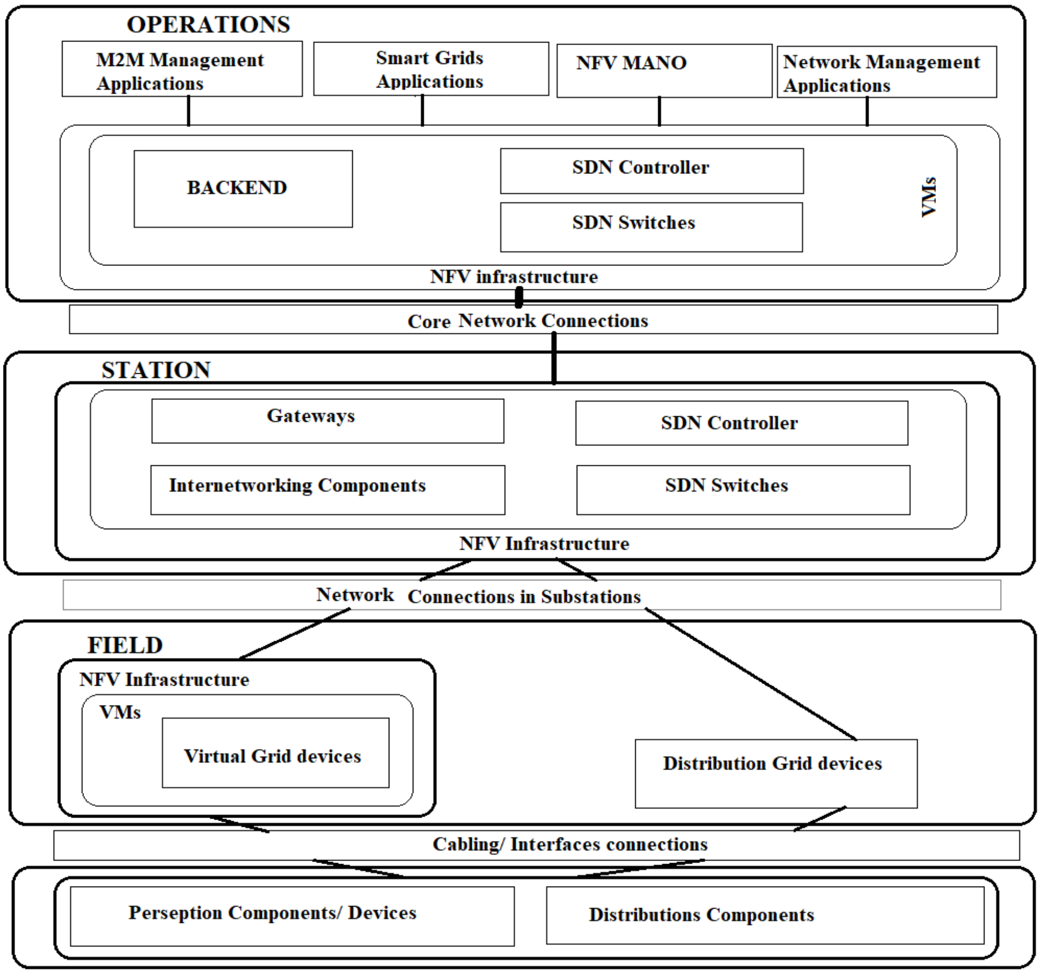


Figure 13: Architecture of the proposed solution.

4.4. Implementation and Evaluation

Figure 14 shows a process to follow when implementing the proposed architecture from design, and evaluation up to deployment. Virtual network design is a process that starts with creating a topology design using a virtual networking system. Network functions and services that support monitoring, control, and automation can be set up along with the required interfaces to smart grid applications and devices. Applications that monitor and report network activity should also be tested on these virtual platforms. After the evaluation, a decision can be made on whether the network design has met the requirements or needs to be improved. Virtual networking platforms make it easy to change designs on the fly and re-evaluate them with minimal effort.

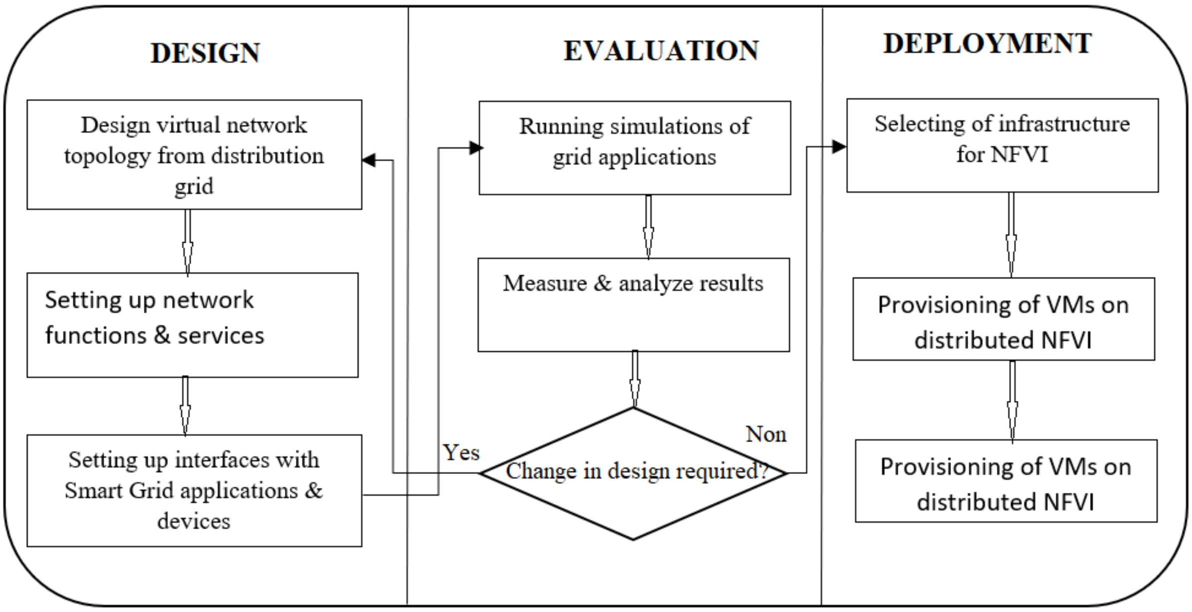


Figure 14: Process for designing, evaluating, and deploying networks using virtualization-based platforms.

A topology design is the first step in the virtual network design process, which is done with the aid of a virtual networking system. The necessary interfaces to smart grid applications and devices can be set up along with network functions and services that support monitoring, control, and automation. These virtual platforms should also be used for testing applications that track and report network activities. Depending on the results of the evaluation, it can be decided whether the network architecture is satisfactory or needs to be improved. Changing designs quickly and re-evaluating them with little effort is made possible by virtual networking platforms. Copies of the virtual machines and virtual network functions (VNFs) developed in the design platform can be provided to a specific grid NFVI when a virtual network design is ready for deployment in a real distribution grid environment. These VMs will need to be provisioned and managed by a robust and capable NFV MANO when deployed in a large-scale distributed environment. After deployment, the network could require an additional test to verify that the communication network is still able to achieve the required smart grid communication requirements. Virtual network devices and virtual hosts are among the virtual network topologies developed and assessed using this approach.

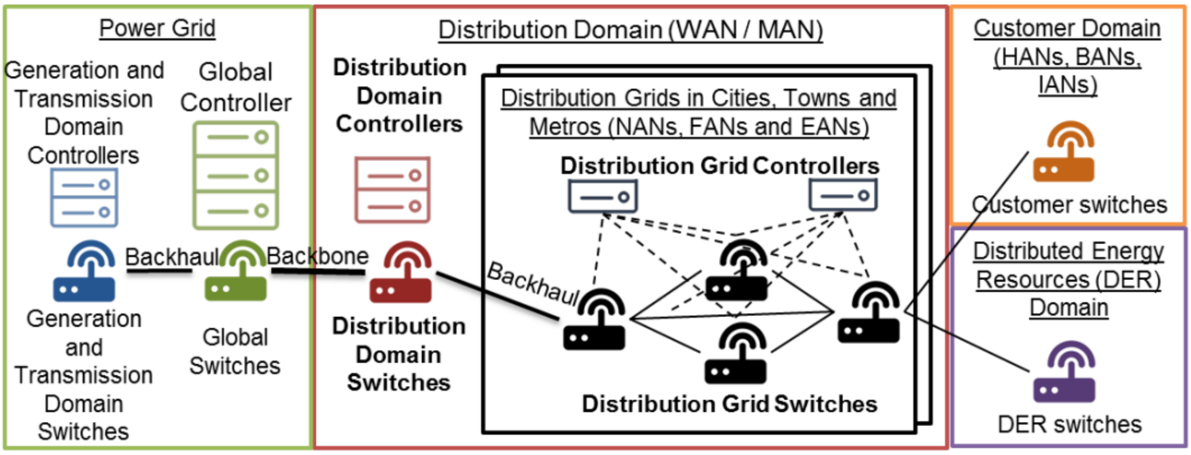


Figure 15: Reference model for designing network topologies for smart distribution grids.

Domain-level switches and domain-level controller deployment should be the first step in all distribution grid communication networks. The backhaul network linkages and the grid's backbone networks will be connected to the domain-level switches, which will also support the distribution grid's core network. The backhaul network links will link the domain-level switches to distribution grid switches, which in turn connect to more switches nearby to form the edge networks known as NANs, EANs, and FANs in the distribution grid.

Distribution grid controllers should be incorporated into these networks to bring control capabilities closer to edge devices and mitigate the risk of SDN switches losing all connectivity to an SDN controller in the event of a network failure. Distribution grid switches could also function as interfaces for networks in the consumer and distributed energy resource domains. Substation computers are installed in each primary and major substation to serve as the station and field zone NFVI in the proposed design for a software-based network that supports an EAM system. These substation computers will have direct connections to the sensors that gather data about the crucial grid assets for the EAM system, network connectivity via SDN switches, and substation asset management (SAM) capabilities. The distribution grid's primary substations will serve as the primary network interfaces for equipment at secondary substations, mini-substations, and other smaller field infrastructures that do not have their own substation computers. The main substation computers will also host M2M IPE services, which will convert all data obtained from distribution grid devices to a standard data model utilized by M2M data acquisition.

The low-level data aggregation and storage of received sensor data will be handled by the IPE. Along with other network service features, the M2M Gateway services will offer a higher level of data aggregation in the edge networks. The domain SDN controllers and the backend M2M services' virtual machines will be hosted by servers in the distribution grid's control center, which will also act as the NFVI in the operating zone. Along with the NFV MANO, EAM apps will also run on these servers. This design will feature network functions that automatically find the whole network topology and update the topology data when the network configuration changes. Applications that enable automatic FFR, traffic scheduling with flow aggregation, run-time network control, and real-time network monitoring, and are included in the proposed design are all included. For communication connection and node failures, FFR functions will serve as the primary recovery mechanism, and a hybrid strategy is advised for optimum performance and flexibility. Functions for automatic traffic scheduling will be developed as M2M services and set up to support automated failover. Users will receive live monitoring metrics of the state of the world network and records will be kept for analysis of previous trends thanks to real-time network monitoring and control functionalities. Figure 26 illustrates the architecture of the suggested design.

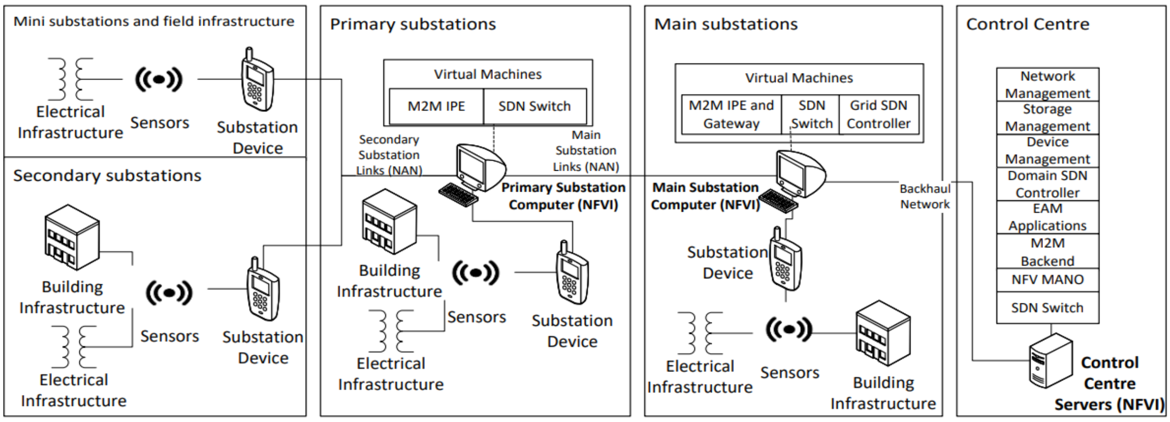


Figure 16: Architecture for a software-based network system in a distribution grid.

After discussing the requirements and design considerations for communication networks for improvement of the reliability of distribution grids and a presentation of the software-based networking architecture design whose goal is to provide these requirements; a design was also proposed for a reliability-focused communication network that will enable an EAM system in the electricity distribution grid. The proposal in this paper suggests a communication network design based on a virtual networking system and incorporates network features including fast failover recovery, automated host failover, and traffic scheduling. Furthermore, the paper discusses the testing techniques and evaluation tools to be used for the evaluation of the proposed design.

4.4.1. Implementation

A network emulation system running on a single personal computer (desktop or notebook) was used to build a virtual network to test the suggested design. The range of this implementation was restricted to the elements displayed in the architecture diagram in Figure 17 and the elements specified in Table 3.

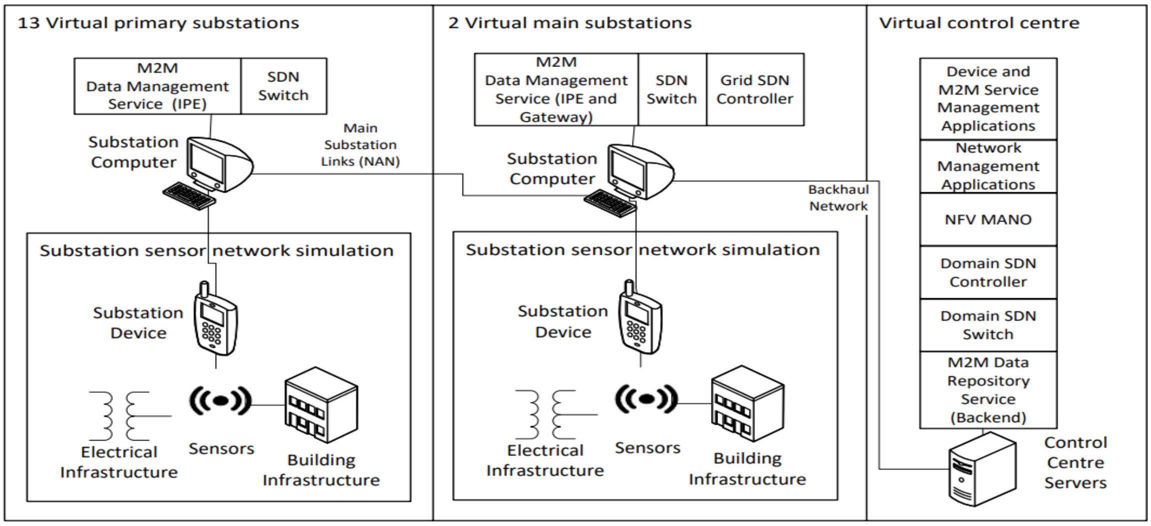


Figure 17: Communication network architecture for the distribution grid.

Layer	**Scope**
Infrastructure	Installation of two main substations and four randomly sampled substations
Devices	Connecting sensors to substation equipment and computers
Virtual devices	Installation of 3 SDN controllers, 7 virtual switches, and virtual hosts
Network	Virtual wired interconnections between virtual devices and equipment to form one NAN and connect it to MAN
Control	Installation of SDN application interfaces and M2M services to enable network tasks such as quick failover recovery, host failover, and traffic scheduling.
Applications	Implementation of Data acquisition and monitoring mechanisms at substations, and creation of non-functional applications for network monitoring, VM administration, and M2M service management.

Table 3: The scope of implementation

The purpose of the software-based communication network was to increase grid dependability by enabling and supporting big-data collection activities and exploiting network features. Initial tests were done to see if it could effectively transmit simulated sensor data from a substation without any data loss or service interruptions. Additional trials were run to see if the network functions could automatically identify and repair a variety of simulated network issues. A network emulation system known as Mininet leverages lightweight virtualization in establishing networks on Linux virtual machines. It is employed to manage a virtual network topology that is implemented programmatically with the help of the Mininet API and defined in a Python script.

Additionally, it makes use of process-based virtualization to set up numerous instances of virtual hosts in separate network namespaces that have access to the directory structures and programs running on the underlying VMs. A high-level architecture of the implemented testbed is shown in Figure 18.

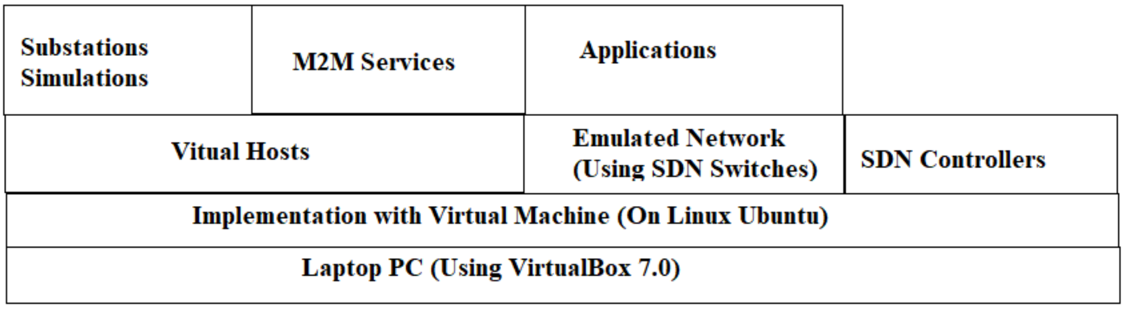


Figure 18: Architecture of implementation testbed

It is worth to assume that the grid architecture and equipment employed in this grid section would be standardized, whereby each substation would have the same sensors, substation equipment, and substation computers and would employ the same communication protocols. This assumption made it possible to develop a Substation Simulation script that could be readily executed each time a new substation was required for the modifications to the implemented design by removing the requirement to configure multiple physical interfaces to real devices. The outcomes of the scheduled tests wouldn't be affected by the usage of substation simulation scripts because the evaluation's main goals were to assist data collection functions and enhance communication dependability. The Substation Simulation Python script was created to generate sensor data, functioning as smart grid devices connected to various substation sensors, using the OpenMTC IPEsensors demo application. The method shown in Figure 13 will be carried out using the simulation script that was created.

Using a function-generated random number and a specified sensor list, the script begins by creating a new instance of a simulated substation device for which a random sensor is chosen. Based on the knowledge required to address the greatest threats to the dependability of Kigali's distribution system, the types of sensors on this list were selected. A fresh sensor reading is then produced using a second random number and predetermined parameters as inputs to a data production function, depending on whether the chosen sensor gives a value or status measurement. To support various scenarios, these input settings can be changed to make adaptability to different cases. The generated sensor reading is then appended to a text string based on the data model which contains the date and time stamp, the unique identification for the device, and the unique identifier for the sensor. The embedded memory of the emulated device momentarily stores each text string.

The usage of a Substation Simulation script simplifies the setup of the testbed and any later modifications to the implemented design by removing the requirement to configure multiple physical interfaces to real devices. The outcomes of the scheduled tests wouldn't be affected by the usage of substation simulation scripts because the evaluation's main goals were to assist data collection functions and enhance communication dependability. The method shown in Figure 13 will be carried out using the simulation script that was created.

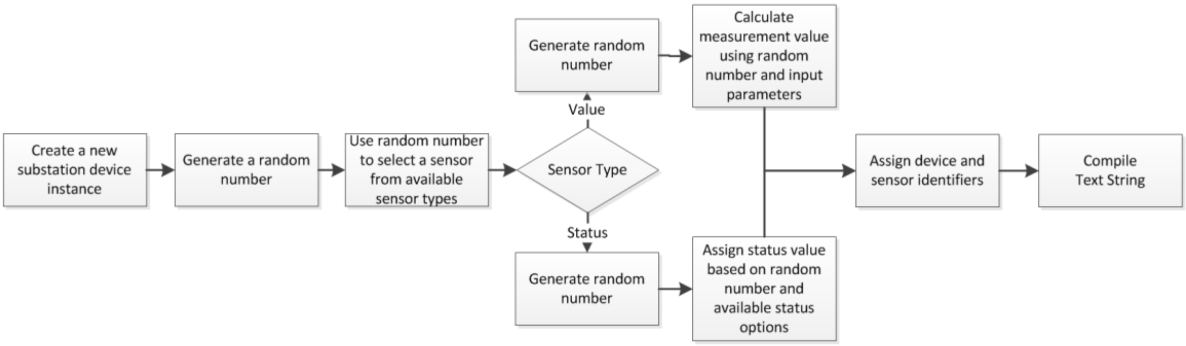


Figure 19 The procedure followed for the Substation Simulation script

Using M2M service platforms, these data sets were automatically combined and transferred to a central data repository. These services were also created as Python scripts with functions for data organization and storage, data transmission using FTP, traffic planning, and automated failovers to backup devices. To implement these M2M services, two different Python scripts were created. 7 virtual hosts representing the key substations in the Kigali distribution grid were used to run the first script, which was termed the key Substation Data Acquisition script. When a text string is indicated to be available, this script performs the duties of an M2M IPE by directly accessing the text string outputs produced by the substation simulation scripts. Once the device's unique identification has been extracted from the text string, the IPE script checks to verify if the source device is included in a list of recognized devices. The script adds the new device to its list of recognized devices if the data come from a new device. It then requests the construction of a new directory structure for the newly discovered device using the file management capabilities of a virtual host acting as a substation Gateway.

The script then makes a distinction between value and status measurements before adding the received text string to one of two different text files for future differentiation. Once a predetermined file size threshold has been achieved, these text strings are then added to text files. A new service request will be made to the substation Gateway service upon reaching the threshold using event-based scheduling. The Gateway substation computer is to be instructed to produce copies of its locally stored text files in the appropriate directory structures as per the second request. The IPE service will erase its local copies to make enough space for the upcoming batch of sensor readings after receiving confirmation that the copies were correctly produced. The data aggregation procedure will be repeated after the file threshold counter has been reset.

To schedule and transfer any acquired data to a backend repository hosted on the control center servers where it can be accessed by EAM applications, the second Python script, referred to as the Main Substation Data Acquisition script, is executed on the hosts that represent the main substation computers. A timer function is set up in this script's time-based scheduling system to wait a predetermined length of time before each data transfer. The script creates an FTP connection between the virtual servers that represent the M2M Gateway network and the M2M Backend whenever the timer approaches zero.

4.4.2. Evaluation tools and test plan

Tools that could monitor communication both under normal operating settings and under fault conditions that expose the distribution grid's dependability were required to assess the deployed communication network. Since the ONOS GUI offers a clear high-level overview of the network and presents real-time network status information in an easy-to-use topology style, it was selected as the primary evaluation tool for overall communication monitoring. For in-depth packet monitoring and packet dissection, the network packet analyser Wireshark was also deployed in the testbed. The data management Python scripts were configured with status logging messages that notified users of each function the script executed and warned them of any issues with the execution of these functions to monitor the effects of simulated communication problems on the scripts that enabled the data acquisition functions. By administratively deactivating particular network interfaces, the Mininet API was utilized to replicate a variety of communication breakdowns. Commands entered through their ONOS CLI were used to simulate SDN controller failures. The Substation Simulation scripts' parameters were changed to make the text files larger than the network's maximum capacity to replicate high volumes of traffic that overloaded the system's resources.

Several tests were run to assess the network that strives to increase grid dependability as well as the deployment of the platform that aims to simplify the network installation process. Every test's outcomes were noted for analysis. The first test was designed to determine how long it would take to set up a network of this kind and whether the suggested Gikondo substation communication network design could be installed on a single desktop computer. The second test assessed the competence of the developed M2M data management service to enable these data acquisition tasks, as well as the implemented network's ability to support automated big-data acquisition functions needed by an EAM system. To increase the reliability of communication, the last five tests assessed the functionality of the M2M services and network functions that were put into place

To obtain a rough idea of the effort required to set up a software-based network for a portion of a distribution grid, the initial test checked the amount of time needed to complete the network preparation chores. These included determining the overall network topology, configuring the network topology, getting the SDN controllers ready, instantiating the network, establishing the best network flow rules, and turning on the necessary network features and services. There were three phases to the network preparation. The first stage concentrated on the core network, which was composed of three SDN switches, three virtual hosts, and three 62 remote SDN controller interfaces that were configured in MiniEdit.

The ONOS CLI and GUI programs, along with the necessary ONOS services, were launched, resulting in the instantiation of three ONOS controller instances. While stage three concentrated on configuring the 10 edge network hosts, stage two was devoted to creating the Gkdo NAN, which was made up of 10 switches and their corresponding network lines. The designed network was instantiated in Mininet at each iteration, and flow rule updates and automated network discovery were tracked using the ONOS GUI. Additionally, console windows were opened to run the built data-collecting services. The entire amount of time needed to prepare the network was computed by adding the time estimates for each task and stage. The test's success criteria included having a software-based network on a desktop computer that was ready to handle a smart grid application and be able to be continuously monitored.

The objective of the second test was to transport the data produced by the substation simulations over the established network, making sure that the data sets ended up in the appropriate directory structure in the grid's central data repository. The transfer must be completed without human interaction or unexpected problems, and they needed to leverage the deployed M2M data management services. On the 10 primary substation hosts, the Substation Simulation and the Primary Substation Data Aggregation script were run to start the test. After that, the main substation hosts launched the Main Substation Data Aggregation script, setting the timed parameters to delay data transmissions by few seconds for each new file that was collected. Terminal windows were launched for each substation host, and these scripts were monitoring their feedback messages while they ran. Both the network activity and the file management tasks carried out by the M2M services were observed through the ONOS GUI and the Linux file explorer, respectively.

For the last five tests, the following requirements had to be met to be successful: Any network function that was put in place had to immediately start responding to the corresponding difficulties they were designed to address once a particular communication issue was simulated. Next, to minimise the impact of the issues in communication and the proper operation of the data-collecting applications, the network functions had to take the necessary steps. SDN controller failures, link failures, node failures, gateway host disconnections, substation disconnections, and NAN isolations were among the failure scenarios that were simulated. The data-collecting procedure had to be uninterrupted and these functions were to be carried out without any data loss. If there were some inevitable delays in data transmission during the testing of the data buffering network capabilities, then communication would immediately continue once the network issue was fixed. Improved round trip latency (RTL) for packets transmitted over congested networks was a necessary outcome of traffic scheduling operations being executed.

A primary substation host was set up to run both the primary substation data aggregation script and the Substation Simulation script for each of these tests. As a result, there would be regular data transfer between the main substation host and the host assigned to serve as the main substation gateway. At predetermined intervals, the main substation data aggregation script was also run to send the collected data to the backend host. The link between the two switches in an active flow path was deactivated to simulate a link failure. We disabled a particular network switch in an active flow path to simulate a node failure. The implemented FFR routines are required to handle recovery from both the link and node failure.

By turning off the connection to particular hosts, it was possible to mimic their disconnection. While an IPE host was disconnected from the network, the host failover mechanisms implemented in the Python scripts had to verify that no data was lost and handle recovery from a disconnected Gateway host. During a simulated NAN isolation, when the backhaul network link was turned off, the buffering routines also had to guard against data loss. Using the Mininet API, all of these errors were replicated. Using the ONOS CLI, one of the ONOS controllers was given the "shutdown" command, simulating an SDN controller failure. The remaining backup SDN controllers had to make sure this failure was handled automatically. The ONOS GUI served to track network activity for every simulated failure, and feedback messages from the Python scripts that were running in the host's terminal windows were used to track the impact on the data acquisition scripts that were in progress.

The substation simulation scripts were modified to enable the storage of thousands of sensor readings per text file prior to each transfer, simulating an overload of the network resources in the active flow pathways. The high volume of readings would cause text files to exceed 10 MB in size, which would cause network congestion to rapidly accumulate in these flow channels due to the 10 Mbps bandwidth limit on the NAN lines. Next, a 10-second transmission delay was added to the scheduling parameter.

On a different substation host, a terminal window was opened, and a ping command was run to see how this congestion affected the RTL of the transferred packets. Using the shared and crowded network resources, this instructed the host to send 64 brief messages to another host. This host was able to function as an additional smart grid application that used brief, time-sensitive messages because each message required a response from the other host. To determine the RTL for every ping message, the time stamps of each transmitted and received reply were recorded.

CHAPTER 5: RESULTS AND ANALYSIS

The previous chapter was about the implementation of a reliable communication system; based on SDN technology; that could make more efficient automated distribution in Smart Grid. The simulation was run using NS-3 as a simulation tool, network emulator Mininet and MiniEdit, and ONOS as an SDN controller. The simulation was performed based on a model designed considering the Kigali distribution network. In this section the results, their evaluation and analysis of the simulation are presented

5.1. Simulation setup of the network

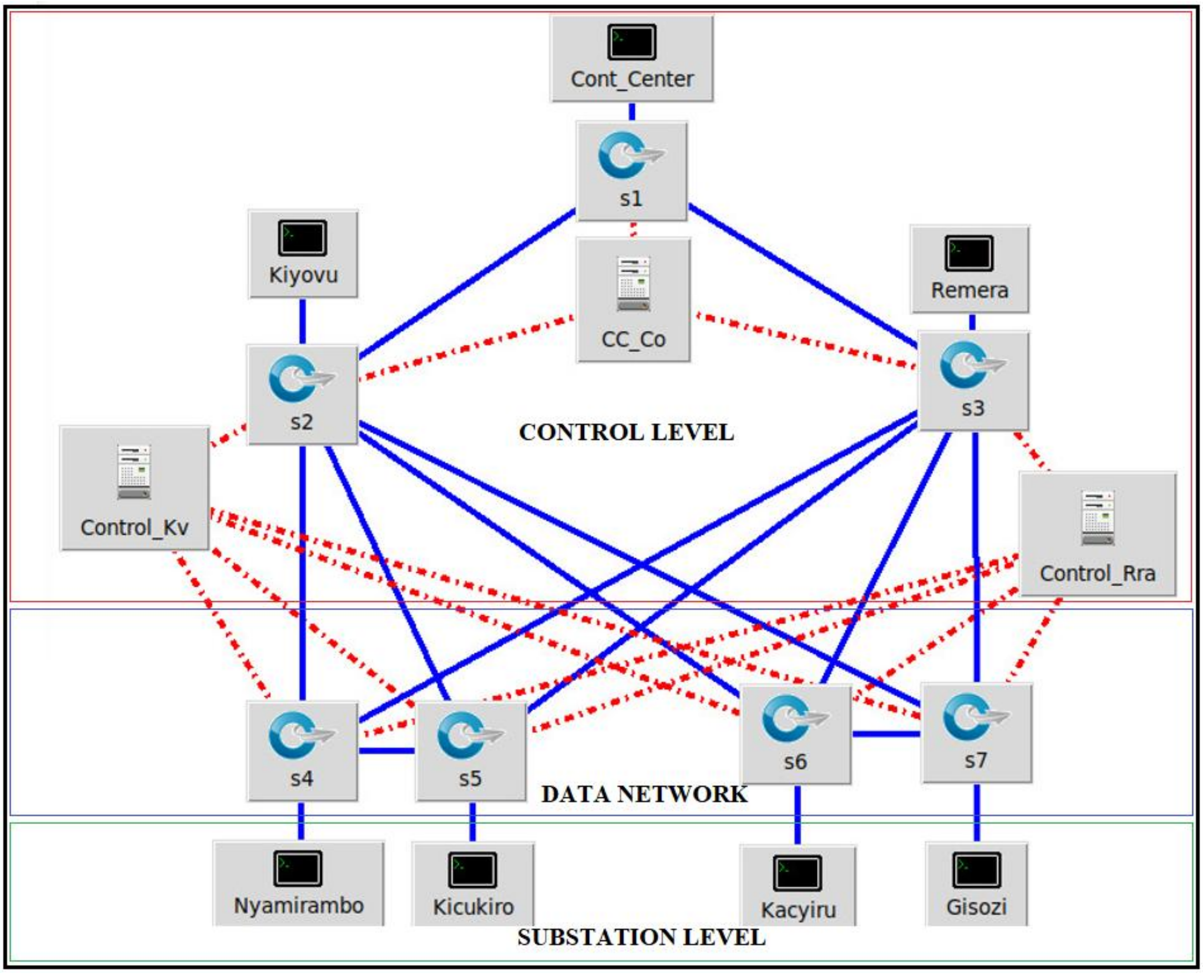


Figure 20: The topology of the communication system emulated in Mininet by MiniEdit

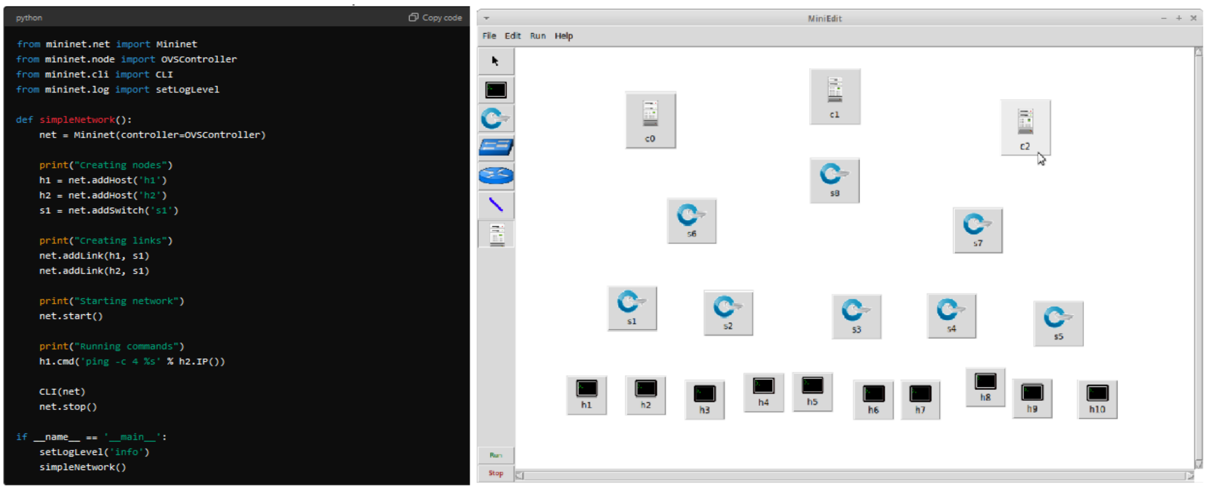
There are multiple processes involved in setting up a simulation environment for a Smart Grid's SDN-based communication system. NS-3, Mininet/MiniEdit and ONOS tools have been used for simulation to test and assess the system for KPIs such as throughput, latency, and failure recovery. These metrics have been analysed for evaluation of the reliability, scalability, and efficiency of the designed system.

5.1.2. Simulation Tools Review

The following tools have been used in the composition of the simulation environment:

NS-3: It is a discrete-event network simulator used to simulate and study network protocols and complex communication systems allowing users to simulate various types of communication systems including wired, wireless, and hybrid networks. NS-3 supports emulation which allows simulated networks to interact with real-world systems, and includes visualization tools like PyViz that enable live visualisation and NetAnim for network topologies and events animation. NS-3 is an open-source software designed to work on Linux, Mac, and Windows.

Mininet: Mininet is a network emulation tool that allows to create a virtual network, running a real kernel, switch, and application code on a single machine, mostly Linux platform [61]. It allows users to simulate, prototype, and test Software-Defined Networking (SDN) environments with real OpenFlow switches, hosts, and controllers, with minimal resources [62]. Below a simple Python script and MiniEdit snapshot are shown in Figure 21.



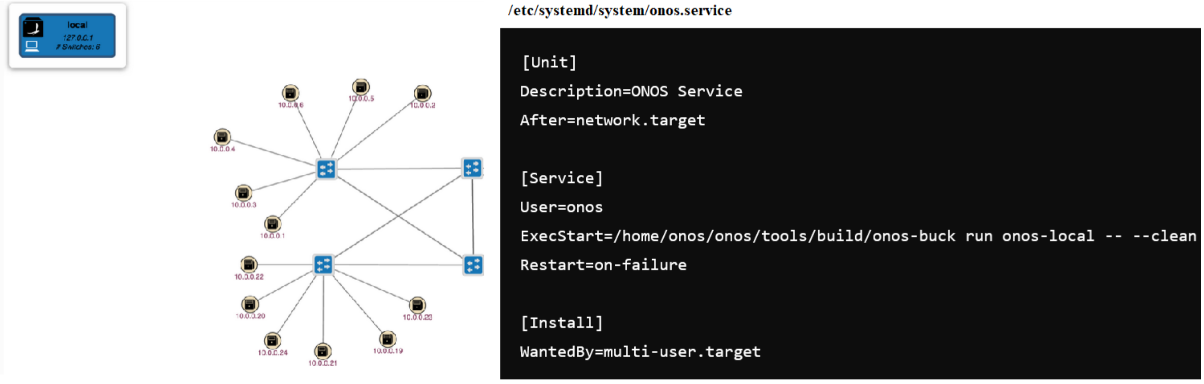
```python
from mininet.net import Mininet
from mininet.node import OVSController
from mininet.cli import CLI
from mininet.log import setLogLevel

def simpleNetwork():
    net = Mininet(controller=OVSController)

    print("Creating nodes")
    h1 = net.addHost('h1')
    h2 = net.addHost('h2')
    s1 = net.addSwitch('s1')

    print("Creating links")
    net.addLink(h1, s1)
    net.addLink(h2, s1)

    print("Starting network")
    net.start()

    print("Running commands")
    h1.cmd('ping -c 4 %s' % h2.IP())

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    simpleNetwork()
```

Python script **MiniEdit GUI**

Figure 21: Simple Python script of Mininet network topology and MiniEdit GUI snapshot

Combined with MiniEdit, Mininet provides a robust and user-friendly platform required for simulating networks in a virtual environment.

ONOS: Open-source software operating system serving as an SDN controller that offers scalability, high availability, and performance. ONOS is a prevailing SDN controller able to provide the ability to control and manage networks using GUI, CLI, and API. Figure 22 illustrates an ONOS configuration file sample and a snapshot of graphical instances.



/etc/systemd/system/onos.service

```
[Unit]
Description=ONOS Service
After=network.target

[Service]
User=onos
ExecStart=/home/onos/onos/tools/build/onos-buck run onos-local -- --clean
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Snapshot of ONOS GUI

ONOS configuration file

Figure 22: Illustration of ONOS graphical instance snapshot and configuration file

5.2. Results presentation

The results presented in this paper have been recorded from a network topology environment setup with 3 controllers. Named CC_Co, the 1st controller was placed in the main Control Center to control the entire network, and it is the master in our setup. The other two; Control_Kv and Control_Rra; were placed in two primary substations, which are assumed to interconnect multiple sections of the Kigali electric grid, namely Kiyovu and Remera. The network consisted of 7 switches, whereby 2 were installed in primary substations and one in the Control Center. To each switch, we connected one host device, which served to take all required data for performance evaluation and analysis.

Through MiniEdit, the backhaul network was used to create the network topology that is illustrated in Figure 20. After the substation hosts were positioned and linked to their appropriate substation switches, the Kiyovu NAN was mapped out next. Ultimately, three instances of the ONOS controller were started, as well as the necessary controller operations. The three controllers were being monitored from the ONOS, launched immediately after the ONOS controllers reported successful activation. To configure the controller interfaces for the network, MiniEdit was utilised to obtain the IP address of each controller.

The Python script required to instantiate the created network in Mininet was compiled and run by MiniEdit upon the issuance of the "Run" command. The network links, flow pathways, and newly generated network switch instances were then reported by the ONOS GUI. The control of each switch by one of the three controller instances was shown by boxes of varying colours when the switches were displayed in a graphical topological view. Grey lines that changed colour based on the volume of network traffic on each link at any one time were used to represent the network links connecting these switches. Real-time reports were also provided on the volume of network traffic on each link. Every host was instructed to broadcast a ping message to the network to identify itself using the Mininet CLI. The ONOS GUI indicated that both the number of hosts and flow pathways found had increased with each ping message. On the topology, these hosts were shown as a circle icon with their IP address underneath. Figure 23 displays the entire network topology that was found and provided by the ONOS GUI.

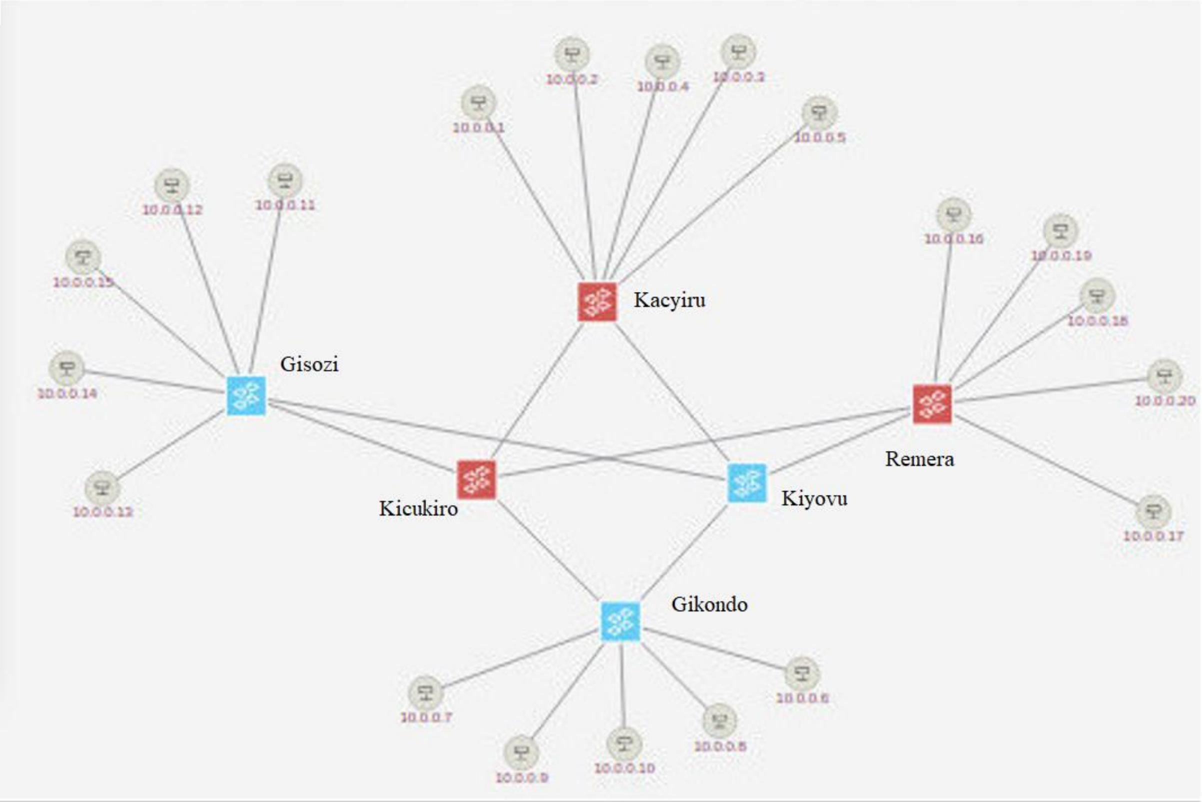


Figure 23: Connected devices displayed by ONOS GUI

The primary goal of the project is to improve communication reliability, and the tested key performance metrics were traffic scheduling, automated host failover, and FFR. A network underwent a series of failure simulations to test certain network operations. During these simulations, the network had to autonomously recover without interfering with the data-collecting process. The network that was put into use for the first two tests was used for all of these tests.

5.2.1. Fast Failover Recovery (FFR)

From the MiniEdit GUI, we were able to disable any link. However, the Mininet "link down" command can also be used to simulate a link failure between two switches in an active flow path between two communicating hosts. Once the link was stopped, all network traffic on the flow path with the broken link was redirected to an alternate flow path, as reported by the ONOS GUI, and the ONOS controller had identified the link failure. An animation of the failing link turning red and fading, followed by a spike in network traffic on another link, illustrates the failed link and diverted traffic on the network topology.

After reactivation of the network connectivity that had been disabled, the traffic flow did not instantly return to the original flow path, despite the ONOS GUI reporting the discovery of the re-established link. It took a few seconds for the flow to automatically revert to the quickest path. Three snapshots depicting network traffic on the original flow channel, traffic diverted following a connection failure, and traffic flowing following the re-establishment of the failed link are shown in Figure 24.

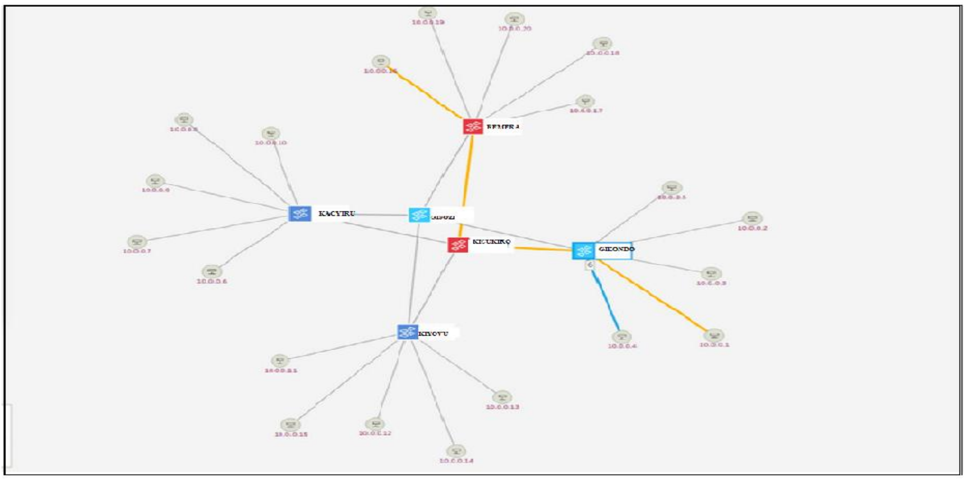
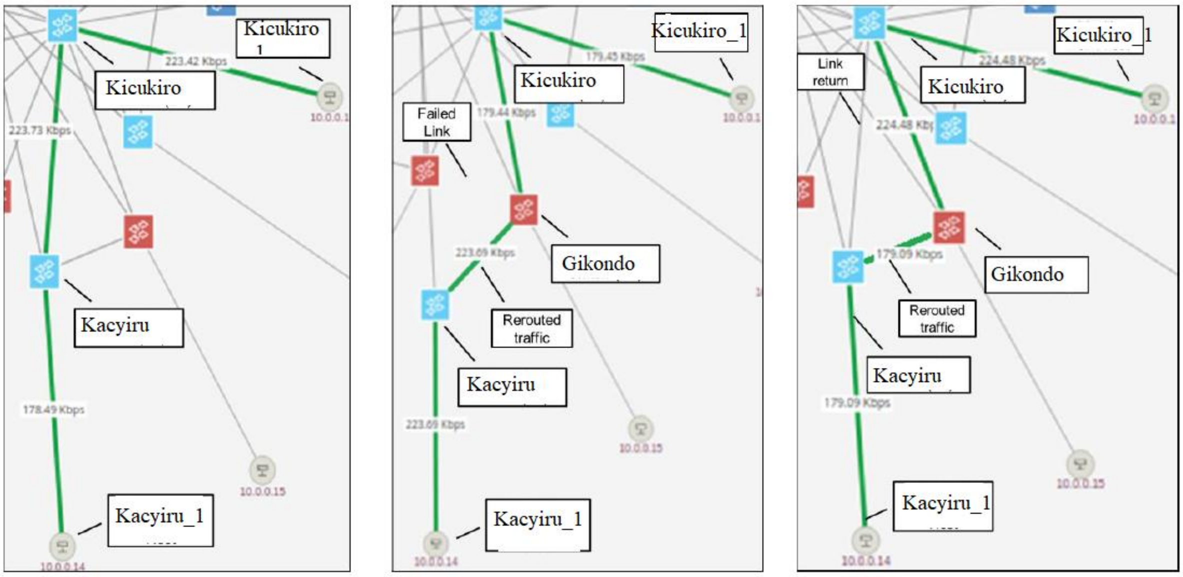


Figure 24: Caption of established connections for network traffic

When a network switch was disabled to simulate a node failure, similar outcomes were seen. The automated rerouting of impacted traffic to an alternate flow path and the node failure were reported by the ONOS GUI after the switch was disabled. The ONOS GUI indicated a topology change brought on by the returning node when the node was re-enabled. Once again, traffic continued to follow the shortest path via the local switch rather than returning to the original one. Figure 24 displays the traffic flows at the beginning of the test, the node failure that was discovered, and the traffic that was rerouted based on the information provided by the ONOS GUI.



Traffic path before connection-cut

Alternative after failure

Traffic flow (retained) after link recovery

Figure 25: Illustration of traffic flow during a network failure

Three times, both FFR tests were administered, and each time, comparable outcomes were seen. These failovers happened without the switches contacting the SDN controllers since proactive FFR features were installed in this network. As a result, the failovers always happened in a matter of milliseconds. Upon identifying that a link connected to one of its ports was unreachable, a switch employed its group tables to locate the appropriate substitute flow and then redirected traffic to another port that was accessible. The switches in question had to wait for their flow entries to expire since no mechanisms were in place to compel the SDN controllers to automatically optimise the network's flow paths on regular intervals. This resulted in a request for new flow entries from the SDN controller, which it fulfilled based on the shortest paths that, if necessary, redirected traffic flows. Since the switches could be turned off while they were buffering and processing packets, there was a chance that data would be lost with each node failure. As a result, the packets would not get to where they were supposed to. Thankfully, retransmission requests were supported by the communication protocols in this network when packets failed to arrive within the fixed period. The data acquisition applications would not malfunction as a result of these retransmissions because the data transmissions for the implemented functionalities were not time-sensitive.

5.2.2. Host Failover Functions Automation

During the execution of this script, all connections to the Kacyiru host were turned off to test the host failover features included in the Primary Substation Data Acquisition scripts. The host with IP address 10.0.0.1 was disconnected from the network topology as soon as these links were stopped, according to the ONOS GUI. As it tried to send its next batch of text files, a warning notice then showed up on the terminal window of the Gikondo substation server. The notification told users that because the chosen gateway was now unavailable, the IPE host was unable to connect to it. The script's attempt to connect to a different gateway host was reported in a subsequent message. Additionally, a successful connection was recorded to the IP address 10.0.0.2 of the Gikondo server. The ONOS GUI displayed an increase in network traffic between the Kiyovu host and the Gikondo host, indicating a transfer of text files, as soon as the alternate gateway approved the connection credentials required by the file management service. The links to the Gikondo gateway server were reactivated following a few successful broadcasts to the backup gateway host. The terminal window for the Kiyovu host now showed no alerts and verified that it had successfully connected to the host with IP address 10.0.0.1. While the Kiyovu substation data was split briefly between two gateway hosts, a review of the backend directories verified that these files were eventually transferred to the same central location via the central repository. Figure 26 displays screenshots from this test of the terminal window of the IPE host and the network traffic to the other gateway that was seen in the ONOS GUI.

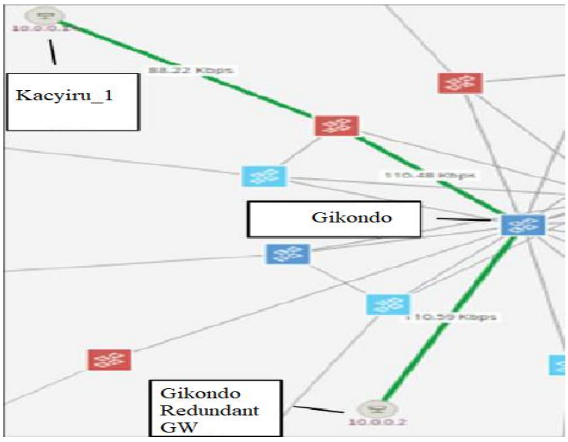


Figure 26: Illustration of links for host failure over automation

This function was scripted to try to connect to the primary Gateway host of the substation before switching over to the backup, so each time the primary host was unavailable for a file transfer attempt, a few seconds of delay was added. Nevertheless, these results showed how effective the host failover function applied was. Though it increased the chance that a gateway host could fail while an IPE host was busy uploading data, this delay had no effect on the entire data collection process. By guaranteeing that any unsuccessful file transfers will be identified and reported as an exception, the FTP protocol is used to mitigate this risk. If necessary, a Python script can automatically handle these kinds of exceptions using comparable failover routines as a possible future improvement.

5.2.3. Traffic Scheduling

Apart from being resilient to errors, a reliable communication network should also reduce the likelihood of preventable delays in communication, which are typically brought on by overworked network infrastructure. By changing the settings of the substation simulation scripts, very large text files were generated for network transmission, which was used to evaluate the traffic scheduling features incorporated in this network. Text files larger than 10 Megabytes in size were created when the file threshold option was altered to record 100,000 readings prior to each file transfer. Additionally, a 10-second transmission delay was added to the scheduling option. The 10 Mbps limit imposed on the NAN's network lines thus caused the network to become overloaded every time a file transfer was carried out. The ONOS GUI made it easy to see that the network resources were overloaded since it turned red on the overloaded connections in the topology and indicated that the maximum 10 Mbps of available bandwidth was being used for each file transfer.

A terminal window was opened on the Kicukiro substation host, and a ping command was run on it to see how this congestion affected the RTL of the transferred packets. The RTL for each ping signal was recorded throughout ten file transfer cycles using Wireshark, which was used to analyse packet exchanges. Each cycle produced a comparable pattern for the ping message RTL measurements, as demonstrated by the graphing of these data sets. Up to a maximum of roughly 0.6 seconds, the trend for each cycle demonstrated a consistent increase in RTL for the pings. Subsequently, data less than 0.05 milliseconds showed a sharp decline in RTL. Figure 29 displays a graph that illustrates the trend for one of these cycles.

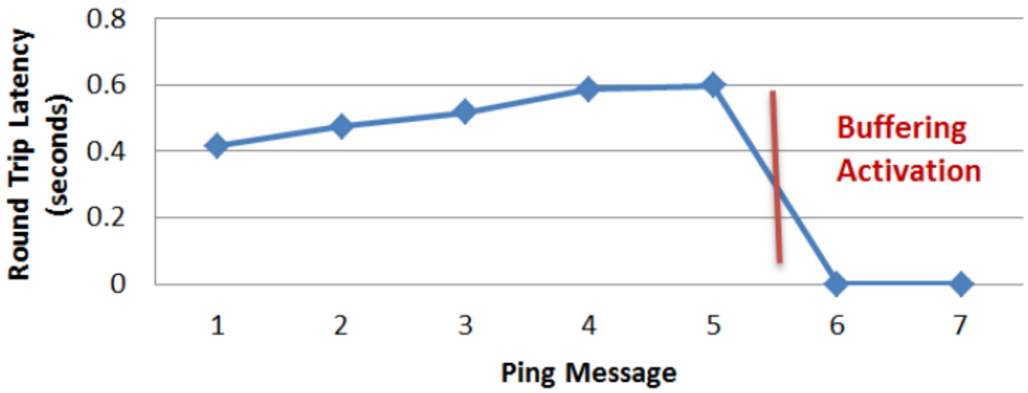


Figure 27: Graph illustrating the latency changes with traffic prioritization

Text file transfers were halted by the traffic scheduling function, which was activated at the same time as these abrupt drops in RTL. Multiple FTP packet exchanges occurred on the network as a result of the traffic scheduling function's ability to stop data from buffering on the host and allow the transfer of text files to continue after its timer expired. Ping messages and other network packets were processed slowly as a result of the flood of FTP packets that overflowed the switches' network buffers. Upon completion of the file transfers, the scheduling mechanism blocked any additional file transfers for the following ten seconds and reset the timer. By doing this, the FTP packets were eliminated from the network and a significant quantity of network traffic was decreased, enabling faster ping message transfers. The amount of time the network is free of congestion can be changed manually by modifying the scheduling parameters. However, longer buffering times led to bigger backlogs building up on the buffering hosts. Thus, more time was needed to clear these backlogs. The time needed to clear these backlogs could end up being longer than the length of time the hosts spend in a buffering state for systems that depend on the transfer of massive volumes of data. As a result, the host's storage capacity will eventually run out and it will eventually overflow. The maximum bandwidth of the network will need to be upgraded in these cases.

Overall, the test results confirmed that the implemented network could recover from the following events: hosted disconnections using automated host failover functions, SDN controller failures using automated controller failover functions, link and node failures using FFR functions, and substation or network isolation events using data buffering functions. These outcomes additionally showed how the traffic scheduling functions included in the current architecture might lessen the possibility of huge data transfers overflowing network capacity. All these aspects are going to contribute to the network's overall increased reliability of communication. These functions were executed without interfering with any of the data collection functions; rather, they served to assist them by making sure they could continue, even in situations when they would otherwise fail. These features offered automated recovery methods that prevented data loss during any of the tests that were carried out, even in situations when delays and communication failures were inevitable. Table 4 presents an overview of the results obtained from the communication reliability tests that were undertaken.

Functions tested	Simulation scenarios	Results	Comment
Fast failover recovery	Link Failure in a distribution grid	Failure detected and traffic diverted to an alternative flow path	Functions executed without interruptions or any data loss.
	Node Failure in a distribution grid		
Automated host failover	Disconnected Gateway host in a distribution grid	Data transfers rerouted to a backup Gateway host	Functions executed without interruptions or any data loss. Slight delay in data transfers.
Traffic scheduling	Overloaded network resources caused by large file transfers	Amount of network traffic reduced with scheduled data buffering, resulting in improved RTL for other network traffic.	Functions executed without interruptions or data loss. Delay in data transfers during buffering.

Table 4. Summary of communications reliability test

CHAPTER 6: **CONCLUSION AND RECOMMENDATIONS**

6.1. Conclusion

The present paper demonstrated, through simulation, that SDN-based communication can significantly improve the reliability and sustainability of distribution automation in the Smart Grid. Effective and efficient communication systems offer the possibility to utility companies the ability to monitor and control the power grid in real-time and hence easily locate faults for quick intervention and energy restoration after power interruption.

This research aimed to study ways to achieve a communication system strong enough to make the implementation of smart grid a success. The results obtained from simulation, are supportive to previous studies whereby utilisation of SDN technology proved to be efficient by providing vital advantages in terms of flexibility and control over complex networks in other fields like Data Centers. These network deal with the transfer of big data from and to various and numerous nodes. The centralisation of the control with SDN controller (ONOS in our case), the system can quickly detect and respond to network failures by dynamically rerouting the traffic around failed nodes or links which maintains data transfer and minimizes communication downtime. Furthermore, the use of multiple redundant connections proved the improvement in fault tolerance, as the network maintains operational integrity even during unexpected failures.

In terms of traffic management and low latency, the tests carried out during simulation experiments have shown that centralised control by SDN controller ensures optimization of the traffic management. To ensure low-latency communication for critical grid data traffic, control signals for automated switches and fault detection can be prioritized over other network traffic which also emphasises the value of SDN in the smart grid environment.

6.2. Recommendations

As the goal of this dissertation was to increase distribution grid reliability through the design and implementation of a software-based communication network, the scope was narrowed due to the field's nature and requirements, and there is still a lot to do and room to enhance and expand this design based on present findings and results.

In modernized power grids, network security should be considered equally significant as reliability. However, the study in this paper has been limited to reliability and hence, we recommend further researches be conducted to improve the security of the grid’s architecture here proposed. Security measures provided by Machine-to-Machine (M2M) such as intrusion management and encryption can be explored to complement this work to further enhance the smart grid security.

We also recommend future research to consider the accommodation of multiple network protocols to improve the design here presented, as it only focussed on Ethernet protocol. This would encompass our proposed model (SDN-based architecture) to support the integration of heterogeneous platforms and devices from various vendors which are required to provide functions and services to perform network and data model translation and ensure systems interoperability. Such improved architecture will serve to leverage the benefits of monitoring and control by SDN-based systems which incorporate different communication protocols and standards.

List of References:

[1] Smil, Vaclav. "Energy in the twentieth century: resources, conversions, costs, uses, and consequences." *Annual Review of Energy and the Environment* 25.1 (2000): 21-51.

[2] https://justenergy.com/blog/history-of-electricity-electric-power/

[1] Farhangi, H. (2011). The path of the smart grid. IEEE Power and Energy Magazine, 8(1), 18-28.

[2] Kundur, P. (2007). Power system stability and control. McGraw-Hill Professional.

[3] Kreutz D, Ramos FMV, Verissimo PE, Rothenberg CE, Azodolmolky S. "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2015.

[4] Li, Y., Li, W., & Tian, F. (2019). A survey on fault-tolerant control for power systems: Approaches, implementations, and future directions. IEEE Transactions on Industrial Electronics, 67(6), 5150-5163.

[5] Papachristos, D., & Kyriakidis, M. (2018). Fault-tolerant control in smart grids: Techniques, methodologies, and applications. IEEE Transactions on Industrial Electronics, 65(8), 6709-6720.

[6] Brancucci Martínez-Anido, C., Schneider, K. P., Blevins, B., Veselka, T., & Zhang, W. (2013). Design principles for resilient power delivery systems. IEEE Transactions on Smart Grid, 4(2), 752-761.

[7] Wang, J., Vittal, V., Kezunovic, M., Li, F., & Rahman, S. (2016). Toward resilient power grids: A survey of current and future research trends in cyber-physical systems. IEEE Transactions on Smart Grid, 8(3), 1216-1227.

[8] Peña-Reyes, C. A., & Monroy, R. (2018). A survey of cyber-physical system security for the critical infrastructure. IEEE Access, 6, 20408-20420.

[9] Luo, F., Sun, Y., Sun, Y., Yu, F. R., & Liang, X. (2019). Security in software-defined networking: Threats and countermeasures. IEEE Communications Magazine, 57(9), 66-72.

[10] Alhomoud, A., & Rayes, A. (2020). A comprehensive survey on security in software-defined networking. Journal of Network and Computer Applications, 152, 102564.

[11] Li, Y., Xiao, Y., Wang, P., Wei, Y., & Xu, Y. (2019). Scalable software-defined networking: A survey. IEEE Access, 7, 118383-118398.

[12] Li, J., Chen, X., Zhang, Z., & Ma, C. (2017). Scalable architecture design for software-defined networks. IEEE Network, 31(5), 18-24.

[13] Islam, S. M. R., Zavodovski, A., Yap, A. D., Yau, K. L., & Hossain, E. (2020). Real-time data communication in smart grid: Challenges, requirements, and solutions. IEEE Transactions on Industrial Informatics, 17(8), 5623-5632.

[14] Islam, S. M. R., Mahmud, M. A., Hossain, M. S., & Nasser, N. M. (2019). Low-latency wireless communication for smart grid applications: A survey. IEEE Communications Surveys & Tutorials, 21(4), 4008-4033.

[15] Kolehmainen, S., Partanen, J., & Al-Salami, M. (2015). Survey of communication and networking technologies for distribution automation systems. IEEE Transactions on Industrial Informatics, 11(2), 394-403.

[16] Fang, X., Misra, S., Xue, G., & Yang, D. (2013). Smart grid—The new and improved power grid: A survey. IEEE Communications Surveys & Tutorials, 14(4), 944-980.

[17] Huang C, Liang Q, Jing J. "Distribution Automation for Smart Grids: An Overview." IEEE Transactions on Smart Grid, vol. 3, no. 3, pp. 1194-1207, 2012.

[18] Lin X, Lu R, Liang X, Shen X, Shen XS. "A Survey on Fault-Tolerant Control Strategies for Smart Grids." IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2171-2189, 2015.

[19] Lin X, Tang Y, Wen S, Liao X, Vasilakos AV. "Fault-Tolerant Control for Smart Grids: Challenges, Approaches, and Future Directions." IEEE Communications Magazine, vol. 52, no. 7, pp. 96-104, 2014.

[20] Huang C, Liang Q, Jing J. "Secure Communication for Smart Grid Distribution Systems with SDN." IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1420-1432, 2019.

[21] Alharbi S, Kandeepan S, Loo J, Maharatna K. "Security Challenges and Solutions for Software-Defined Networking in Smart Grid." IEEE Access, vol. 7, pp. 68582-68595, 2019.

[22] Niazi M, Zaman N, Gidlund M. "Scalable SDN Architectures for Large-Scale Networks: A Review." IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 643-677, 2019.

[23] Liu Z, Khatibi S, Li W, Tang X. "Low-Latency Communication for Real-Time Applications in Industrial IoT." IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2498-2508, 2021.

[24] Chen M, Hao Q, Xie Y, Li W. "Low-Latency Data Collection for Smart Grid with Software-Defined Networking." IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1467-1476, 2019.

[25] Huang C, Liang Q, Jing J. "An SDN-Based Communication Architecture for Integrating Distributed Generations into Smart Grids." IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1687-1695, 2018.

[26] Huang C, Liang Q, Jing J. "Distribution Automation for Smart Grids: An Overview." IEEE Transactions on Smart Grid, vol. 3, no. 3, pp. 1194-1207, 2012.

[42] Xiao Y, Xiao Y, Yi J, Liu H, Lei Y. "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges." IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 602-622, 2016.

[43] Ab Rahman NH, Hashim F, Ismail Z, Ismail MH, Sani SM. "A Cost Analysis of Software-Defined Network (SDN) Deployment." Procedia Computer Science, vol. 59, pp. 346-355, 2015.

[44] Ahmed SI, Kim J, Lee S. "Cost-Effective Architecture of Software-Defined Networking for Green Computing." Sustainability, vol. 10, no. 11, Article no. 4109, 2018.

[45] Gungor, V. Cagri, et al. "A survey on smart grid potential applications and communication requirements." IEEE Transactions on industrial informatics 9.1 (2012): 28-42.

[46] CODE, PRICE. "Communication networks and systems in substations–Part 5: Communication requirements for functions and device models." ed (2003).

[41] Farhangi, H. (2010). The path of the smart grid. IEEE Power and Energy Magazine, 8(1), 18-28.

[47] Yang, L., & Song, Y. (2012). Smart grid communication networks: Challenges and opportunities. IEEE Communications Surveys & Tutorials, 14(4), 1054-1078.

[48] Cokic, Mita, and Ivan Seskar. "Software-defined network management for dynamic smart GRID traffic." Future Generation Computer Systems 96 (2019): 270-282.

[49] Faria, P., & Vale, Z. (2016). Integration of renewable energy sources in smart grids: A survey on the state-of-the-art. IEEE Communications Surveys & Tutorials, 18(2), 1047-1070.

[50] Kundur, Prabha S., and Om P. Malik. Power system stability and control. McGraw-Hill Education, 2022.

[51] Demirci, Sedef, and Seref Sagiroglu. "Software-defined networking for improving security in smart grid systems." 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA). IEEE, 2018.

[52] Gungor, Vehbi C., et al. "Smart grid technologies: Communication technologies and standards." IEEE transactions on Industrial informatics 7.4 (2011): 529-539.

[53] Gungor, Vehbi C., et al. "Smart grid technologies: Communication technologies and standards." IEEE transactions on Industrial informatics 7.4 (2011): 529-539.

[54] Iqbal, Sana, et al. "A comprehensive review on residential demand side management strategies in smart grid environment." Sustainability 13.13 (2021): 7170.

[55] Yu, Xinghuo, and Yusheng Xue. "Smart grids: A cyber–physical systems perspective." Proceedings of the IEEE 104.5 (2016): 1058-1070.

[56] Thomas, Mini S., Seema Arora, and Vinay Kumar Chandna. "Distribution automation leading to a smarter grid." ISGT2011-India. IEEE, 2011.

[57] Liu, Xin, et al. "Electric power grid resilience with interdependencies between power and communication networks–a review." IET Smart Grid 3.2 (2020): 182-193.

[58] Liu, Xin, et al. "Electric power grid resilience with interdependencies between power and communication networks–a review." IET Smart Grid 3.2 (2020): 182-193.

[59] Zong, Yu, Zengwei Zheng, and Meimei Huo. "Improving the reliability of HWMP for smart grid neighborhood area networks." *2016 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. IEEE, 2016.

[60] https://www.projectguideline.com/doing-software-defined-networkingsdn-simulation-under-ns-3/

[61] Wang, Shie-Yuan. "Comparison of SDN OpenFlow network simulator and emulators: EstiNet vs. Mininet." *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014.

[62] Sheikh, Mohammad Nowsin Amin, et al. "A qualitative and comparative performance assessment of logically centralized SDN controllers via Mininet emulator." *Computers* 13.4 (2024): 85.

[63] Starke, Allen, et al. "Toward resilient smart grid communications using distributed sdn with ml-based anomaly detection." *International Conference on Wired/Wireless Internet Communication*. Cham: Springer International Publishing, 2018.

[64] Singh, Avtar, Navjot Kaur, and Harpreet Kaur. "Extensive performance analysis of OpenDayLight (ODL) and open network operating system (ONOS) SDN controllers." *Microprocessors and Microsystems* 95 (2022): 104715.