SOFTWARE-DEFINED NETWORKING BASED ON CENTRALIZED CONTROL FOR SMART GRID COMMUNICATION

By

ELISHA INDARJIT

Thesis submitted in fulfilment of the requirements for the degree Master of Engineering in Electrical Engineering

In the faculty of Engineering and the Built Environment

At the Cape Peninsula University of Technology

Supervisor: Dr. Marco Adonis

Co-supervisor: Dr. Angus Brandt

Bellville

Date Submitted: 18/05/2022

DECLARATION

I, Elisha Indarjit, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

	18 May 2022
Signed	**Date**

ABSTRACT

Communication networks are growing, and there is a need for improved planning and evolution of network scaling between *Smart Grid* (SG) and communication networks. The research explores the application of a *Software-Defined Network* (SDN) to the Smart Grid, for control, intelligence, and management. The same applies to *Smart Grid* environments; scalability, reliability, and intelligence become requirements.

The *Smart Grid* plays a significant role in providing electrical power to users and enterprises; the requirements are not phased within a location but rely on remote connectivity. This study measures the concept of a *Software-Defined Network* by the *centralized* SDN *controller* and the performance within an integrated network.

Further to the application, Software-Defined Network is applied to an SDN Smart home, SDN Automation plant, and SDN Electrical distribution system. Design phases are set out to evaluate the testing upon use case and integrated architecture of SDN and SG.

The study proves the successful integration of SDN+SG by a Network Exposure layer added to the target architecture to meet the advanced needs of connectivity to the end-user. Each use case directs the evolution of technology for automation and computing.

ACKNOWLEDGEMENTS

I wish to thank:

- Dr. M Adonis, for the guidance and motivation to achieve breakthrough results, the passion for research, and collaboration during the entire process.
- Dr. A Brandt, for your great wealth of technical insight, shaping the nature of my research and the testbed.

ABBREVIATIONS	**ACRONYMS**
AAA	Authentication, Authorization and Accounting
ABNO	Application Based Network Operation
ACL	Access Control List
API	Application Programmable Interface
APN	Access Point Network
BDDP	Broadcast Domain Discovery Protocol
BGP	Border Gateway Protocol
BGP-LS	Border Gateway Protocol - Link State
BOD	Bandwidth on Demand
CCN	Contents Centric Networks
CO	Central Office
CPE	Customer Premises Equipment
CPS	Cyber Physical System
CRP	Cloudcasting Rendezvous Point
DHCP	Dynamic Host Configuration Protocol
DOS	Denial of Service
DPI	Deep Packet Inspection
DPID	Data Path ID
EDFA	Erbium Doped Fiber Amplifiers
eMMB	enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
FA	First Available
FTTH	Fibre to the Home
GMPLS	Generalized Multi-Protocol Label Switching
GTR-VNE	Global Topology Resource - Virtual Network Embedding
HAN	Home Area Network
HTTP	Hypertext Transfer Protocol

HTTPs	Hypertext Transfer Protocol secure
IBGT	Insulated -Gate Bipolar Transistor
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IoT	Internet of Things
IP	Internet Protocol
IPSec	IP Security
IPTV	Internet Protocol Television
LLDP	Link Layer Discovery Protocol
LNR	Local Node Ranking
LSRP	Link State Routing Protocol
MAC	Media Access Control
MEC	Multi-Access Edge Computing
MIFaaS	Mobile IoT Federation as a Service
mMTC	Massive Machine Type Communication
MPLS	Multi-Protocol Label Switching
NAC	Network Access Control
NAN	Neighbourhood Area Network
NAT	Network Address Translation
NFV	Network Function Virtualization
NFVIPOP	Network Function Virtualization Point of Presence
NSO	Network Service Orchestrator
ODL	OpenDayLight
OF	OpenFlow
ONOS	Open Network Operating System

ONOS	Open Network Operating System
OOB	Out Of Band
OSI	Open System Interconnection Model
OSPF	Open Shortest Path First
OSPF-TE	Open Shortest Path First - Traffic Engineering
OTN	Optical Transport Network
OVPN	Optical Virtual Private Network
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
PLC	Programmable Logic Controller
POF	Protocol Oblivious Forwarding
POX	Python
QoS	Quality of Service
RAN	Radio Access Network
REST	Representational State Transfer
RIP	Routing Information Protocol
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
SCADA	Supervisory Control and Data Acquisition
SDN	*Software-Defined Network*
SDWSN	Software-Defined Wireless Sensor Network
SG	*Smart Grid*
SGCM	*Smart Grid* Conceptual Model
SGIM	*Smart Grid* Investment Model
SGIMM	*Smart Grid* Interoperability Maturity Model
SGMM	*Smart Grid* Maturity Model
SNMP	Simple Network Management Protocol
SSH	Secure Socket Shell

TCP	Transmission Control Protocol
TE	Traffic Engineering
TLS	Transport Layer Security
TLP	Transport Layer Protocol
TLV	Type Length Value
TTL	Time To Live
UDP	User Datagram Protocol
UPS	Uninterrupted Power Supply
URLLC	Ultra-Reliable Low Latency
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPDN	Virtual Private Dial Up Network
VPN	Virtual Private Network
VRF	Virtual Routing Forwarding
WAN	Wide Area Network
WSN	Wireless Sensor Network

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

CHAPTER 2: LITERATURE REVIEW

APPENDICES

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Networks are growing and need to adapt to changing environments such as Smart Home, Automation Plant, and Electrical Distribution Stations. Xu (2016, p. 1) found the increase of intelligence and control within communication and electrical networks applied to different environments can enhance and improve a system. There is a demanding need to meet and understand the customers' data analytics and system solutions requirements. Equally important is to guarantee the confidentiality, integrity, and availability of the information flow in a *Smart Grid* network discussed by Braeken & Kumar (2020).

The Internet has created a digital society, where many systems are connected and is accessible. With the rapid growth and demand of the traditional IP networks, there is a concern for their complexity and management. When looking at a traditional data network, there are many hardware devices such as routers, switches, and firewalls. The devices mentioned include hardware connectivity, which moves the data through them, and software configured to control data movement based on rules and regulations. Current data networks are limited to the following: flexibility, high availability, modularity, resiliency, scalability, security highlighted in Froom et al. (2010).

1.2 Background to the problem

Looking into the past years, we have seen many network changes, which have driven the technology. Koshiba, Shin & Sebe (1996) identifies the main driver for change is the size of the network, which has grown massively due to the number of devices that need centralized intelligence.

The size of a network and when comparing the rate of configuration change has created a significant amount of complexity in administering devices. Each device vendor uses its configuration language and specification, user interface, and syntax to add to the complexity, making it a highly skilled and time-consuming task.

All connected devices are allocated an IP address; when a packet is sent through the network, the source and destination IP address is included in the packet. Routers consist of a forwarding table that contains the IP address (Kurose et al., 2013).

Currently, data networks are vertically integrated: Riccardi et al. (2018) discusses the control and data planes that are bundled together on a device. Network Operators must configure each network device separately using low-level and often vendor-specific commands. In addition to the configuration complexity, network environments have to endure the dynamics of faults and adapt to load changes. In the current IP networks, automatic reconfiguration and response mechanisms are virtually non-existent.

*Software-Defined Network*s develop rapidly in academics and the telecommunications industry due to essential characteristics, including centralized control, scalability, network availability, and creation of services. It has a significant advantage over the traditional network, such as reducing Capex and operational expense (Sahoo, Sahoo & Panda, 2015).

The controller is a key enabler for network control logic under the policies defined by network providers. Software-Defined Networks are mainly used in emerging deployments within the communication network.

Controllers are diverse and developed according to Service Provider's specifications and environment; a domain controller will be placed either in Core Data Network, or Transmission or Radio Access Network. There are over 30 types of controllers proposed by industry and academics; the deployment of controllers cater to run time technologies, different programming languages, and feature sets. Now that controllers are becoming mature in their state and Service Providers are considering the implementation, it becomes time to re-investigate the performance (Shaikh & Darekar, 2018).

The concept of *Software-Defined Network*ing is relatively new in the industry and taking its leap. In contrast to a traditional network, SDN provides centralized control and implements APIs for programmable networks. As companies expand their network, it creates complexity and poor management of several network devices. It calls for centralized control, simplification, security, and scalability. The vital purpose of implementing Software-Defined Network – Centralized control is to prove the ability and features of the SDN controller, Liyanage, Gurtov & Ylianttila (2015) also pay reference to the feature sets.

A traditional electric distribution network consists of distribution transformers, circuit breakers, feeders, copper cabling, voltage regulators, etc. The electric distribution network has no intelligence or data collection techniques. The current distribution network is statically controlled, with limited improvements based on its infrastructure. The increase of developing and expanding locations requires a network that meets the demand for scalability, reliability, Quality of Service, and service management. Added to the demand, in today's digital society, Ford, Siraj & Eberle (2014) discuss the imperativeness to increase security and overcome to a degree transmission sequence failure, fraud, terrorism, cyber-attacks, vandals, and thefts.

Developed and non-developed countries serve to benefit from Smart Grid deployment. *Smart Grid* serves as the proposed way of smart energy to provide smart energy management and usage. Some of the key benefits of *Smart Grid* are uninterrupted power supply, decrease in transmission loss and increase in renewable energy, flexibility, and on-demand management discussed by Sachs (2010) and Guo, Pan & Fang (2012). The current aims of *Smart Grid* are policy implementation and its cost-effectiveness.

Smart Grid is a system that consists of intelligent electricity distribution devices, communication networks, sensors, metering, enhanced reliable performance; the *Smart Grid* will facilitate greater penetration in the energy sector.

Electricity is essential and used widely in all industries, homes, and devices. Conservation and demand management can form part of the solution to use electricity wisely. Electricity outages affect businesses and economies enormously, the call for better long-term solutions to manage the grid becomes a requirement. Yeung & Jung (2013) investigates *Smart Grid* as one of the solutions to handling the ever-increasing demand for power energy.

The increase of technology and integration into the ecosystem, including the communication backbone network, compel vendors to look into the security issues of *Smart Grid,* and the communication networks face. However, Li, Huang & Wu (2017) explains the communication network used for the energy grid suffers from the complexity and massiveness of energy-related data due to the *Smart Grid* being large-scale and remote-based.

A report from the Global *Smart Grid* Federation presented by Ramakrishna Kappagantu et al*.* (2018) provides insight into the existing power grid networks stating the current infrastructure will not meet the demand of the 21st-century parameters in quality, efficiency, reliability, ecology, and economy. The articles based on the research provide current technical and miscellaneous challenges to the state, inadequacies in grid infrastructure, cyber security, storage concerns, data management, communication issues, stability concerns, power theft, and energy management highlighted in Ramakrishna Kappagantu et al. (2018).

- This research explores the concept of Software-Defined Networking applied to Smart Grid networks compared to a traditional network.

This work is done by:

- Analyzing the protocols used on traditional networks
- Developing a base architecture,
- Identifying the areas that need improvement and
- Provide use cases of how SDN can manage networks.

1.3 Problem Statement

Authors Zhang et al. (2013) and Dong et al. (2015) have tried to link up *Smart Grid* to a communication network, Zhang et al. (2013) looks at opportunities and use cases for SDN enabled *Smart Grid*, while Dong et al. (2015) have proposed architecture and high-end resiliency. Da Silva et al. (2016) define how current communication networks suffer from network failures and related link failures; downtime equates to revenue loss and no connectivity. The need for a centralized system that brings both the communication network and Smart Grid together is needed.

The Internet has created a digital society, where many systems are connected and accessible. However, with the rapid growth and demand, of the traditional IP networks, there is a concern about the complexity and difficulty of managing the network.

When looking at a traditional data network, there is a range of hardware devices; such as routers, switches, and firewalls. The devices mentioned include hardware connectivity, which moves the data through them, and software, which is configured to control the movement of data based on rules and regulations. Current data networks are limited to the following: flexibility, and high availability.

Looking into the past years, we have seen many changes in networks, which have driven the technology. The main driver for change is the size of the network, which has grown massively due to the number of devices, which needs intelligences.

The size of a network and when comparing the rate of configuration change has created a great amount of complexity in administering devices. To add to the complexity, each device vendor uses its own configuration language and specification, user interface, and syntax making it a highly skilled and time-consuming task.

Network Operators are required to configure each network device separately using low-level and often vendor-specific commands. In addition to the configuration complexity, network environments have to endure the dynamics of faults and adapt to load changes. In current IP networks, automatic reconfiguration and response mechanisms are virtually non-existent.

Software-Defined Networking is adopted to build a resilient network for SDN and Smart Grid to overcome the challenges.

This research asks the question:

- Can an SDN controller centrally control Smart Grid components to enhance features such as security, policy implementation, and management?

The following chapters will explore the advantages and challenges of implementing SDN and show how the OpenFlow protocol was integrated into the traditional *Smart Grid*.

1.4 Research Aim and Objectives

This research aim explores the hypothesis that the SDN controller can control Smart Grid components providing benefits such as security, policy implementation, and device management.

In addressing the research question stated in **1.3**, the following objectives were identified:

Literature Review:

- To list the advantages and challenges faced.
- To provide resolutions to these challenges.
- To provide a detailed explanation of SDN and its packet operation within *Smart Grid*.
- To perform simulation and analyze the performance in control and packet flows.

- To identify SDN use cases and migration strategies.
- To utilize software tools by integration and configuration to demonstrate and prove the intelligence of SDN applied to the *Smart Grid* by use cases.
- To define use cases and testing metrics,
- Research into each of the current security solutions, the most significant security challenges, and security threats in networks,
- Investigate SDN security at each layer,

1.4.1 Test Bed Objectives and Outcomes:

- To prove the control of SDN into *Smart Grid* networks.
    - By using a SDN controller to deploy policies.
- To define how SDN can be applied/integrated to the *Smart Grid* environment
    - By selecting the most appropriate SDN controller for this *Smart Grid* implementation.
    - Define the roles of each technology and create a standard architecture,
    - Define the protocols to be used and policies.
    - To identify software tools that can be used to perform simulation
    - To build the concept by using models and logic to prove the use case.
- To outline how to implement SDN in a Smart Home, Automation Plant, and Electrical Distribution environment—providing the features and functionality with a design model.
    - To explain each integrated environment mentioned,
    - Listing the features and functionality of each design model.
    - To explain the packet movement and define the application slice per environment.
- To showcase management of the SDN controller within a network.
    - By the implementation of Use case and results.

1.5 Research Steps

*Software-Defined Network*ing technology describes a layered network topology. Inherited network architecture is found in network standards and protocols. Network design engineers abide by these standards and protocols. However, *Software-Defined Network*ing takes the presence to define new standards and protocols. They give network design engineers more freedom to scale and build their sized architecture. The separation of the forwarding and control plane allows for better programmability. This separation enables the brain or thinking ability role to be placed within the SDN controller, depending on the environment of the communication network. Further network layers can be connected to the architecture depending on the situation, De Puga, Salvador & Pellicer (2019) provides a more in-depth study of the orchestrator and hierarchical controller.

- The *Smart Grid* is built around many scaling environments; our focus will be a smart home, automation plant, and electrical distribution for this research. Each environment is created on its proprietary system and is de-centralized. The methodology is to use *Software-Defined Network*ing architecture and apply it to each *Smart Grid* environment. The intent is to analyze each *Smart Grid* environment and assess the performance against SDN use cases.
- The study was conducted in line with Denyer & Tranfield (2009), using a systematic analysis shown in Figure 1.1.

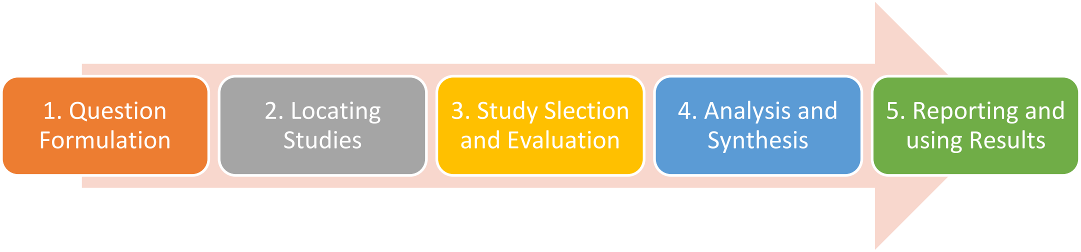

1. Question Formulation
2. Locating Studies
3. Study Slection and Evaluation
4. Analysis and Synthesis
5. Reporting and using Results

Figure 1.1: Steps followed in the Systematic Literature Review (SLR)

- The first stage involved framing the focus on the study, taking a high-level view of what current systems and future roadmaps become a requirement within SDN, and SG questions stemmed from the research gaps and what has been deployed in the real world. This stage incorporated brainstorming and forecasting future interest developments in the next five years.
- The second stage involved locating the studies. The search began by querying citation databases using keywords and selecting journal papers. The choice of peer-reviewed journal articles becomes a recommendation.
- The third stage consisted of refining the search and basing the study on its particular purpose. Our focus on the selection of journal papers is on the following: (1) Recent *Software-Defined Network*ing deployment; (2) *Smart Grid* evolution; and (3) Security within SDN.
- The fourth stage incorporated the classification of research methods used and addressed each *Smart Grid* environment's communication network. From this, the study could lay out the process of the integration of SDN and SG, adding the security framework to use.
- Finally, the fifth stage represented the documentation and Results in the next sub-section presented.

Chapter 2 provides the literature review of researchers and the work done on *Software-Defined Network*s and *Smart Grid*. Chapter 3 introduces the concepts and provides integrated architecture, an overall figure of SDN technology, system modelling, and OpenFlow protocol to define the Network exposure layer proposed between the Service Provider's communication network and *Smart Grid* network. Chapter 4 provides the design phases, testbed, and results. Use cases are identified to prove each scope of the control ability and management from the SDN controller to the network. Chapter 5 gives the conclusion.

The work scope brings *Software-Defined Network* and *Smart Grid* together, defining an integrated architecture for control and intelligence. Simulation by software compilation and a testbed is presented to prove by use case the scope. SDN also plays a role in the management of devices and programmability. The testbed will show the management of the *Smart Grid* components in Chapter 4.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The concept of *Software-Defined Network*ing is relatively new in the industry and taking its leap. As companies expand their network, it creates complexity and poor management of several network devices. It calls for centralized control, simplification, security, and scalability. The vital purpose of implementing *Software-Defined Network* – Centralized control is to prove the ability and features of the SDN controller.

Previous literature reviews from Chapter 1 have focused on different aspects of SDN and SG, providing brief technical information. They fail to give new researchers a concise overview of research outcomes and what still needs development when applying the concept of SDN to the *Smart Grid* on a deployment level. Furthermore, the limitations of SDN lie in the level of security within the architecture and protocols.

The proposed topic is to integrate intelligence and control into the *Smart Grid* environments when looking into an SDN Smart Home, SDN Automation Plant, and SDN Electrical Distribution. The intelligence mentioned will focus on a new architecture, security at each layer, functionality, and overcoming challenges in traditional communication and distribution networks. SDN will allow companies and their customers to control their systems, understand the network, and calculate the forecast growth. Furthermore, the ability to develop services relevant to a smart world in a short-term timeframe and save cost.

The SDN architecture in Figure 2.1 by Sibylle & Dave (2017) shows the SDN controller as a feedback node; resources can be assigned and controlled depending on the hardware and software, while the controller can enforce policies based on the principle of programmability of network services.

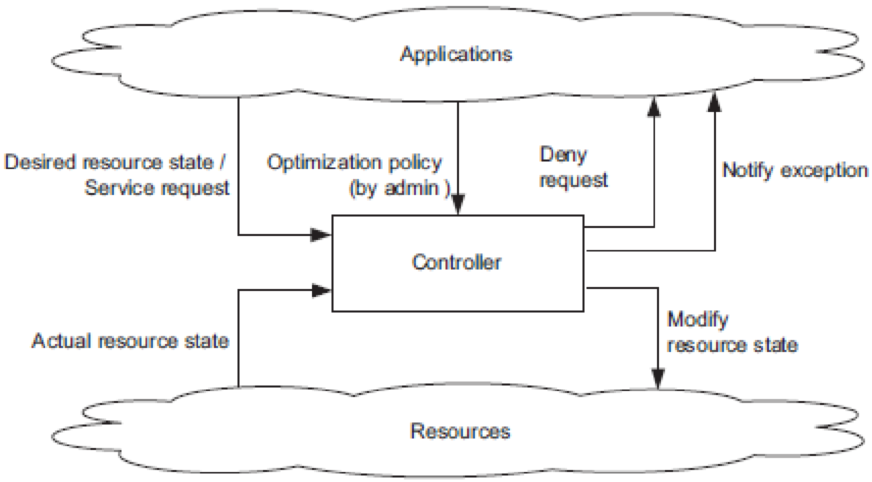


Figure 2.1: SDN controller as a feedback node

(Sibylle & Dave, 2017)

The remainder of Chapter 2 is organized as follows. Section 2 presents the literature review for *Software-Defined Network*; Section 3 presents the literature review for *Smart Grid*. Security is discussed in Section 4, where the gaps are detected, and future research lines are proposed.

2.2 *Software-Defined Network*

The concept of *Software-Defined Network* (SDN) brings a new mindset. It changes the network framework, Ruaro, Caimi & Moraes (2020) explore ways of managing network components by a centralized control system and creating an open, standardized way of deploying network services. Network Operations require a flexible network to provide network services and troubleshoot to avoid new project costs and timeous rollouts. SDN is a technology that caters to self-managing network capacities by policy-driven techniques, self-managing network availability, and thus avoiding unnecessary expansion expenses.

The primary concept to deploy SDN is for network automation and to cater for network programmability. SDN networks can self-heal and self-manage when a failure occurs. The centralized control plays a vital role in managing updated tables of links, nodes, application slices, and configurations. Other important features are mentioned below:

- The architecture consists of open standard interfaces to accommodate 3rd party *Smart Grid* applications.

- Controller to be allocated for centralized control of automation and programmability.
- Separation of the control plane from the data plane infrastructure or virtual infrastructure.
- They are built on similar architecture standards across fixed, mobile, cloud, enterprise, and security domains.

Figure 2.2 shows a traditional network diagram consisting of a proprietary operating system and each vendor-specific operating system designed with its applications. Network engineers require a variety of skills to manage their environment.

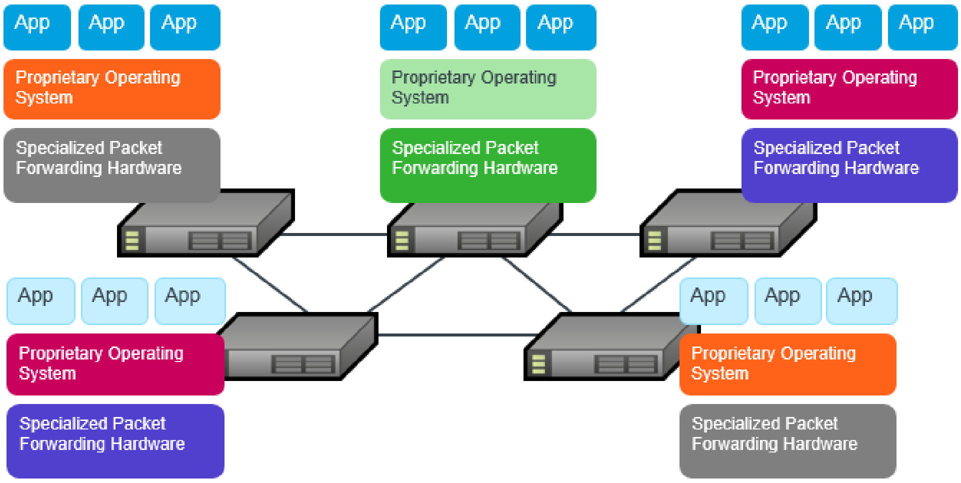

App
App
App
Proprietary Operating System
Specialized Packet Forwarding Hardware
App
App
App
Proprietary Operating System
Specialized Packet Forwarding Hardware
App
App
App
Proprietary Operating System
Specialized Packet Forwarding Hardware
App
App
App
Proprietary Operating System
Specialized Packet Forwarding Hardware
App
App
App
Proprietary Operating System
Specialized Packet Forwarding Hardware

Figure 2.2: Traditional network consisting of proprietary operating systems

According to section 1.1, the objectives serves to discuss the benefits of *Software-Defined Network*ing:

The model of SDN serves to address several issues listed from Yan et al. (2016) below:

- Network configuration is prone to human error, costing the network capacity links, consisting of a manual basis balancing the network, which becomes exhausting. To ensure high peak traffic is balanced correctly.
- New network services take months to deploy. Several departments form a Service Provider consisting of Radio Access Networks, Core Data Networks, Transmission,

and other departments; each network service requires all teams to be on board and to audit then scale their environment.

- A wide variety of management systems within the Network Management Group leads to segmented skills and an uncentralized platform.
- Correction of a network configuration change can take hours and keep the network in low recovery states.
- Product and vendor differences mean mixed networks, and it becomes challenging to plan and manage.
- A wide variety of skills are required for each Operating System for each independent node.
- SDN offers further benefits and becomes the platform to design and implement a scalable and efficient network.
- Orchestration – is a component, or the heart of the network, mainly to manage the network services. The term is applied to automating network processes. And deployment and management of the entire infrastructure over a network and network within networks.
- Abstraction – the task of hiding complexity and providing a simplified view. It entails multiple layers of abstraction within the Service Provider's network. Figure 2.3 shows the layers of abstraction.

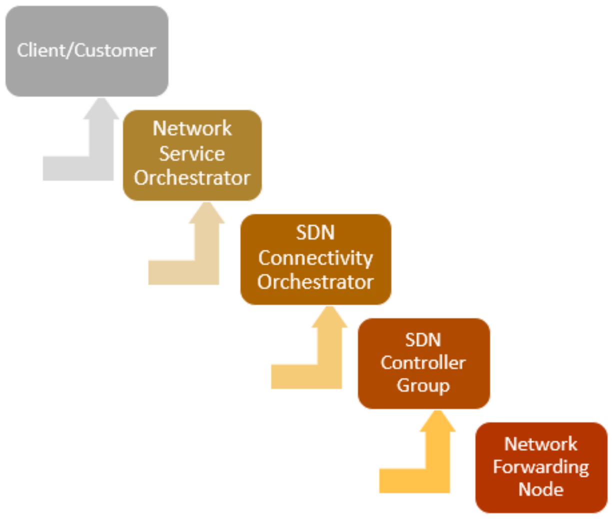


Figure 2.3: Abstraction layers of a communication network

Summary of Software-Defined Network benefits:

- Efficient use of resources (links, NE). The SDN network contains an SDN controller that knows the network's links and nodes. The SDN network can detect high traffic and load balance traffic.
- The concept moves from expensive proprietary equipment to cheap, fast commodity hardware.
- Providing better network visibility on network state so that capacity may be used more efficiently.
- Reduce network complexity and operational overhead.
- Allows Service Providers to add new services - New revenue source. To accommodate smaller companies and their requirements.
- SDN allows feature implementation and faster deployment.
- Partitioning of resources for safe experimentation.
- Implementation of well-known simple systems.
- Simplified operations, programming, etc., with centralized control.
- New possibilities

  - User plane/control plane decoupling allows new decision algorithms and Hardware uses.
  - Enable application-level programming of network.
- Vendor choice
  - Hardware/Software from different vendors, more diversity by giving the Service Provider more control of their strategy and roadmap.

Related Work:

SDN is a technology rather than an implementation or deployment which drives a new approach. Authors Ruaro et al. (2018), Berestizshevsky et al. (2017) give accredited research on centralized control. Authors Velloso et al. (2019) and Cong, Wen & Zhiying (2014) offer using SDN to improve the scalability of networks in a straightforward approach. Authors Berestizshevsky et al. (2017) and Sandoval-Arechiga et al. (2015) provide a genetic SDN model without the specification of a standardized approach. Focus is placed on the advantages and disadvantages of the SDN paradigm. On the other hand, Scionti, Mazumdar & Portero (2018) explain power-saving techniques by powering off links, not in use. The previous paper's concern is the focus placed on the SDN communication network, mainly the SDN framework, and not proposed how to integrate other environments, such as *Smart Grid*, or control *Smart Grid* components.

The SDN research moves to showcase a new architecture and prove the ability of control and intelligence from the SDN controller and the mediation used between two different environments. Previous proposals do not focus on security within the SDN architecture, especially defining security per layer, considering four layers: (i) securing the controller; (ii) securing the infrastructure; (iii) securing the application; and (iv) securing APIs and communication. The research will expand on Ellinidou et al. (2019) using Chiplet design, performing software compilation of execution using fewer interactions between the SDN controller and layer two environments.

2.2.1 Scalability

Federation, Hierarchical, and clustering are the principles to establish a scalable and high-performance network. To scale the network requires the addition of SDN controllers developed per network; however, a hierarchical orchestrator is needed to talk to another network. For the scope of the research, the design will focus on the SDN controller level.

Further to note:

A Federated system serves to establish multiple instances operating in a collaborative arrangement, a hierarchical system is to establish multiple instances operating in a tiered arrangement, and clustered system to establish multiple instances operating in a fault-tolerant mode and one standard model.

2.2.2 Performance

The question arises of how the SDN controller will handle the network processing; the SDN controller will be placed separately from the routers and handle control plane data, which reduces the capacity processing limitation, not to say, every device will have to pertain to a capacity threshold limit. The need for the designer to cater to future expansions is a requirement.

2.2.3 Security

Security techniques will be placed at each architecture layer and discussed in section 2.4.

Table 2.1 below shows the capabilities of the SDN controller,

Table 2.1: Genetic SDN controller capabilities

Feature	Description
Abstraction	Northbound – provides a simplified view, Southbound – translate between different network elements and the internal controller.
Auto Discovery	The controller supports dynamic network discovery of nodes, links, devices, services, etc. The controller will contain an up-to-date network topology table.
Connectivity Manager	Responsible for configuration and protocols.
Flow Manager	Responsible for pushing flows down to network elements and the correct direction of flow.
Local database	The controller should manage a complete database of the local domain environment.
Network Optimisation	The controller needs to have intelligent routing decision-making on the layer of network cost and network availability.
Orchestration	Fulfilment of an end-to-end service provisioned network.
Path Computation	The function contains all elements responsible for creating, managing, optimizing, and reporting paths through the infrastructure.
Plug-in manager	Dynamically informs the applications and internal modules of changes to adjust capabilities.
Policy management	Creation, adjustment, and deployment of domain rules
Reporting function	Gather and store statistics.
System Security	Allows security of the controller and connections. May include application control, encryption, authentication, and integrity checking.

Table 2.1 defines the capabilities of the SDN controller, but there are more needed feature capabilities listed in Table 2.2.

Table 2.2: Required feature for SDN controller

(Dinh & Park, 2021)

Feature	**Description**
Audit	Controller to audit the network and point out packet drops.
Controller Implementation	The controller shall be implemented on one or more VM instances.
Controller Module	Controller to support upgrade and rollback on network changes, Controller to be modular and support clearly defined interfaces, and controller to handle management framework.
Comprehensive logging capability	Ability to analyze problems.
High Availability architecture	Controller to contain flow table of links up and down, re-route traffic when a link failure occurs, and ensure an availability rate of greater than 99,99%.
Integration	Controller to support standard-based interfaces, including orchestration.
Internal monitoring	Controller to assess internal processes.
Multi-vendor	Controller to support a host of vendors.
Transaction oriented	Controller to guarantee the data integrity of the system.
Scalable and performance	Controller to allow scalability by adding more controllers or even accommodating more nodes. To build a network of federation, hierarchy, and clustering.
Security	Controller to provide strong authentication and integrity validation capabilities.

2.2.4 Comparison Analysis of *Software-Defined Network* with OSPF Protocol and Content-Centric Networks.

Nugroho, Dian, and Setyawan (2017) take a practical view to compare the performance of *Open Shortest Path First* (OSPF) protocol and SDN technology. The analysis of measuring SDN performances by a virtualisation software tool named GNS3. From the analysis of SDN, the results generated show the delay range of 0,3 ms to 6 ms and 0 % packet loss indicating SDN performance is greater then traditional networks.

The traditional Smart Grid or *Internet Protocol* (IP) network infrastructure uses a low-level configuration and specific syntax for each vendor. A network engineer will require a host of skills to manage network routers and switches, adding a level of management complexity onto the network.

The centralization and use of a controller or multiple controllers within a communication network become the building blocks for automation and network programmability on an SDN network.

SDN technology caters to network flexibility, helping the Service Provider to work more efficiently with different vendors and allowing the company to develop its hardware and feature requirements.

The architecture of *Software-Defined Network*ing is shown in Figure 2.4. Some of the aspects of SDN from www.opennetworking.org (2020) are listed below:

- Separation of the data and control plane.
- The practice of standard interfaces to be able to program network devices.
- Auto-discovery feature allowing devices on the network to be polled.
- Establishment of virtual platforms.

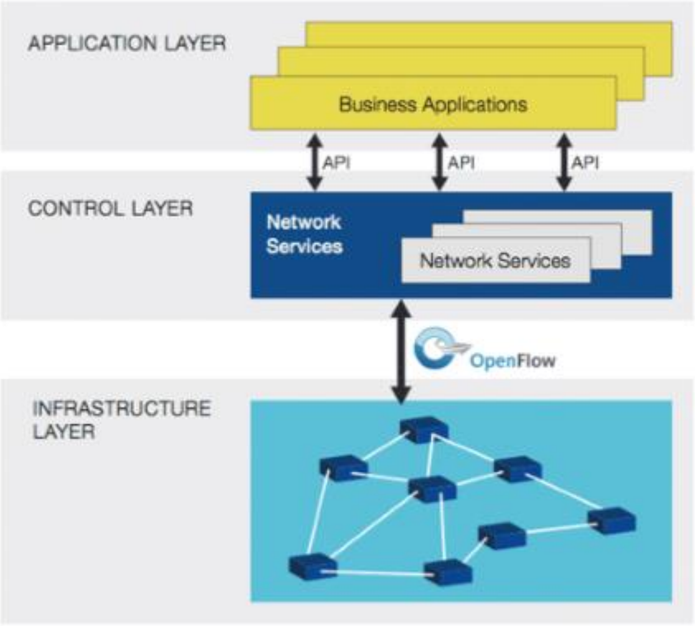


Figure 2.4: SDN Architecture

(Open Networking Foundation, 2020)

Table 2.3: Comparison of routing technology assessment on five devices taken from Nugroho, Dian & Setyawan (2017) shows the test results,

- The first is Open Shortest Path First (OSPF), with cost configured on each link within the network,
- The second, using SDN on condition of loss within each link, and third,
- Using SDN on condition of no loss calculated.

Table 2.3 Comparison of routing technology assessment based on jitter and packet loss on five devices

(Nugroho, Dian & Setyawan, 2017)

Parameter	**5 Device**		
	OSPF	**SDN**	**SDN - no loss**
Delay without load (ms)	57,3	0,3	0,3
Jitter without load (ms)	86,7	0,2	0,1
Packet Loss without load (%)	0	19,5	0
Delay Load 1 (ms)	58,5	0,7	0,5
Jitter Load 1 (ms)	102,5	1,7	0,5
Packet Loss Load 1 (%)	0	58,4	0
Delay Load 2 (ms)	75,6	4,1	2,3
Jitter Load 2 (ms)	75,4	372,8	2,3
Packet Loss Load 2 (%)	0	98,9	0
Delay Load 3 (ms)	99,5	∞	4,3
Jitter Load 3 (ms)	101,8	∞	52,6
Packet Loss Load 3 (%)	0,5	100	0

The results indicate a simulated relation for each delay parameter, jitter, and packet loss are related to each other. Table 2.3, it’s identified from the results of non-loss SDN. For all topologies, it indicates that SDN does not have any loss. The results show delay and jitter values, which are in the excellent category according to ETSI standards. The results prove that SDN condition without loss has comparable results and is suitable for packet delivery.

The main advantage of SDN technology is its practicality in building a topology. Unlike OSPF, which uses link-state principles and can rearrange traffic paths in terms of link failure, SDN looks at a topology that is not connected entirely between devices. Further to the state, SDN is excellent in QoS parameter results when compared to OSPF network (Nugroho, Dian & Setyawan, 2017).

2.2.5 The congestion control mechanism in ***Software-Defined Network*ing by traffic re-routing and SDN processing.**

Srikanth et al. (2018) study *Software-Defined Network*ing, enabling the user to program the network elements. With everyday increasing network traffic and vital congestion problems, the need for SDN becomes more pressing. The solution uses the shortest path algorithm from the SDN framework.

Congestion on the network amounts to many packets within a specific location due to the lack of capacity resources. The effect of congestion resultant in the customer having a poor experience on the network. A network node consists of a data and control plane. The data plane serves to route user traffic, while the control plane takes care of routing functionality.

The purpose of the SDN is to separate the data and control plane. OpenFlow is a software interface used to program the data plane switches, which helps manipulate the forwarding tables.

*Software-Defined Network*s help program network elements by separating the data and control planes. Algorithm 1 shows the decision steps taken for the control plane when there is a possible link failure. The system learns the network parameters, like link cost, and performs traffic re-routes according to the threshold set. This in-effect will evaluate the network topology and load balances the traffic. The algorithm will minimize the traffic on a single link by choosing alternate paths to balance the traffic.

Algorithm 1 Congestion control algorithm by Srikanth et al. (2018, p. 57)

```
Initialize Host 1, Host 2, Host 3
Set the information about hosts switch, Mac, ports, and switch paths.
Initialize the graph to the topology.
Get the statistics like Bandwidth from the REST API.
Get the Response about using the URL and Option.
If Option is Host Details then Get the Host Details
Else If Option is Switch Con then Get the Switch Connections
Else If Option is Transmission Link then
Compute the link transmission Cost
```

```
End If
Get all the paths that lead to a switch
for paths in shortest paths Cost
for node in paths
Add node to the path list
End for
Get the links cost of all the paths between the Hosts.
Get the best path using the flow rules from the REST API
If Congestion Occurs, then
Re-route the traffic in the next
Shortest best cost path
End If
End
```

Comer & Rastegarnia (2019) explore the SDN controller processing feature but establish the processing externally; the controller contains flow tables that have rules and policies that send and receive packets. The SDN controller aggregates all control plane subsystems into one program. Programmers will need skills on each specific programmer interface to develop applications.

The SDN controller is limited to modularity and does not support non-disruptive updates. Another challenge of the SDN controller is the reuse of SDN modules; if a module is used to collect topology information and requires to be used in another SDN environment, it will need to be re-coded. Furthermore, external management needs to support external applications when an SDN change is encountered. The external system needs to be proactive in supporting any state.

To assist with the mentioned processing challenges, the SDN controller needs to be re-designed to support the external environment and the outsourcing of packet processing. The re-design phase of the SDN controller is to allocate and divide into many sets of services instead of a centralized base. And allowing programmers to select an arbitrary programming language instead of force programming language.

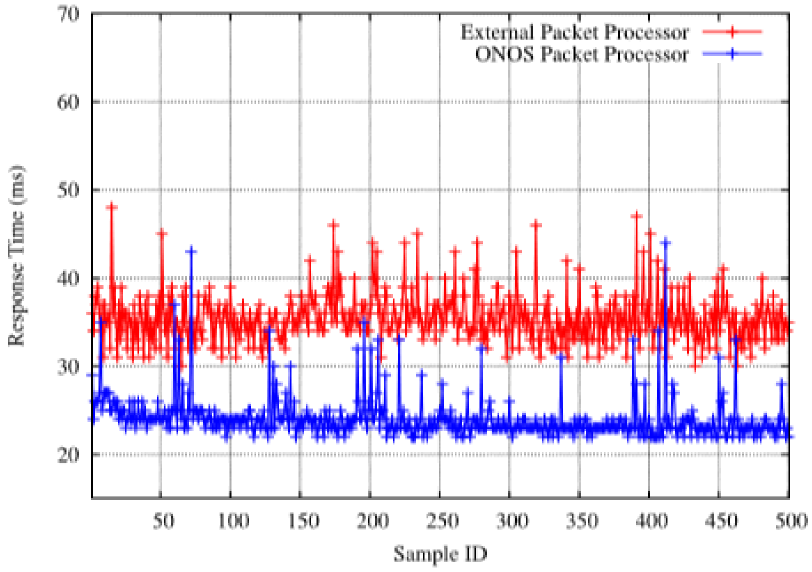


Figure 2.5: Showing the results of ONOS vs. External packet processing

(Comer & Rastegarnia, 2019, p. 127)

A test analysis was performed to showcase the results of packet processing of an ONOS controller versus external package processing. The experiment in Figure 2.5 was run 500 times by Comer & Rastegarnia (2019); the results show 24ms to 35 ms compared insignificant to the benefits of external packet processing.

2.2.6 Software-Defined Transport Network: Fundamentals, findings, and futures

The 21st century demands greater network dynamicity and on-demand connectivity that guarantees the customer capacity, lower latency, controlled jitter, availability, and control. With the growth of the network and the number of data centres, there is a need for high-capacity cloud-based applications. The applications include multimedia content, office automation platforms, and gaming content distribution.

Within the WAN layer, which connects to the data centres via multiple 10GE/40GE/100GE links, any failure of links will affect the entire WAN, which directly degrades service on the customer experience. When the failure occurs, the control plane will dictate to the user plane to recover from the failure.

King et al. (2016) provide insight into the communication network and the demand to make the network more responsive to services and more efficient. To create virtual instants that allow dedicated slots for the customer. Traditionally, an operator may use dedicated fiber links to cater to Generalized Multi-Protocol Label Switching (GMPLS) to meet customers' demands. Using *Software-Defined Network*ing (SDN), the network can be controlled and cater to the customer's needs, which can load balance the traffic on the provided links. SDN uses programmatic flow-based technologies, like OpenFlow and cloud DC interconnection, to communicate in a multi-vendor environment.

Software-defined transport networking consists of the network controller managing the WAN links and the physical transmission/switching nodes. A survey conducted by Forrester Consulting on behalf of Juniper Networks, January 2014, identified the critical demands from a cloud infrastructure classified as bandwidth, performance, reliability, and automation/programmability.

King et al. (2016) highlight the adoption of SDN; it will require incorporating other technologies and providing resource orchestration capabilities to span its domain and operation, which offers end-to-end connections. The IETF’s SDN framework, Application-Based Network Operations (ABNO), is a reference architecture built with the following architectural principles mentioned in Table 2.4.

Table 2.4: Reference architecture resources

(King et al., 2016)

Resource	**Description**
Loose Coupling	It provides a better-defined and standard-based Application Programmable Interface (API) and protocol mechanisms for faster development.
Low Overhead	Ensures no repeats within the management and control functions, reducing the overhead.
Modular	Referred to the integration of new capabilities in existing devices.
Intelligent	Designing the framework that includes Path Computation Element (PCE) and Traffic Engineering (TE).
Resource Management	The framework was built to discover and management for various networks and nodes.
Dynamic Management	Considering the SDN controller, it provides dynamic control based on application.
Policy Control	Deals around specifying connection requirements to implement policy management.
Technology Agnostic	Allowing for a wide variety of forwarding mechanisms self-managed.

Table 2.4 helps the designer when planning a network with the integration of SDN.

2.2.7 SDN Challenges

It allows new, improved architecture, control, management, and operation, enabling improved routing and topology control protocols to reduce the computational process.

*Software-Defined Network*ing is adored for its flexible operations and programmability layers; it provides room for virtual platforms that allow the selection of hypervisors and the fundamentals of Network Function Virtualisation (NFV).

The careful approach for SDN requires measurements calculated for the applications and programs running on the computer. Blenk et al. (2018) review the research study of virtualization on SDN and the hypervisor placement. There are three limitations of the research to mention:

- Facility location problem,
- SDN Controller placement, and
- Virtual network embedding.

Facility location problem tasks are to find the best facility located in a multi-vendor network. The problem is indicated generally by Heller, Sherwood & Mckeown (2012); tenant controllers need to connect to the hypervisor instance, while the hypervisor instance needs to connect to the SDN switches on the lower end.

SDN Controller placement, when the focus is on Controller Placement Problem, the underlying problem results from the number of controllers required per network. In SDN controller placement, Yao et al. (2014) recommend careful planning of the controllers, whiles Sallahi & Hilaire (2015) raise interest in building the network with controller resiliency.

The embedding of the virtual to a physical network resource is a crucial part of network virtualization. Many algorithms solve the issue; Yu et al. (2008) propose flexible path splitting. To apply Virtual Network Embedding to the control plane, which will solve the mapping of virtual SDN resources.

Customers leverage higher-quality video services, such as video calls, gaming, IPTV, and video services. The centralized control layer is known for its flexibility and enablement of bandwidth efficiency ways.

The deployment of large-scale multicast services requires smart group membership services and bandwidth reservation with guaranteed QoS. Soni et al. (2017) consider a layered ISP network in Figure 2.6, the deployment of NFV allows mini-datacentres stemming from network aggregation points at Central Offices (CO). The NFV CO allows multiple networks functions to run, such as NAT, firewall, or cache from commercial off-the-shelf hardware. The metro layer interconnects the Central Offices and uses more extensive capacity links. Also, the core network layer interconnects central offices that serve as Points of Presence and includes NFV infrastructure called NFVIPoPs. Finally, the SDN controller is responsible for programming packet forwarding in its domain

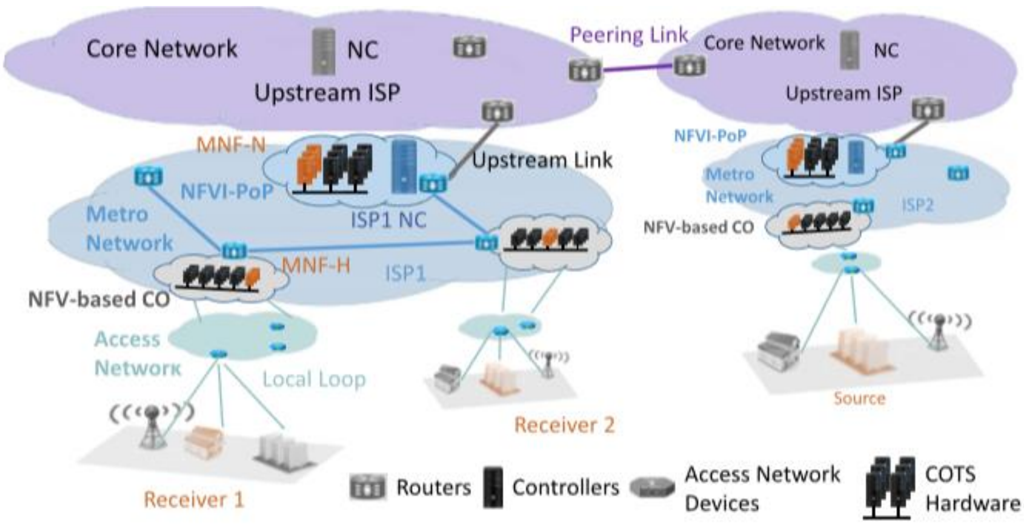


Figure 2.6: Software-Defined ISP network

(Soni et al., 2017)

2.2.8 Use Case of SDN

Table 2.5 shows the use of cases and critical characteristics.

Table 2.5: SDN Use Cases

Control method	References	Key characteristics to be identified
Service and network auto-discovery	(Talarico, Makhijani & Pillay 2016)	The mobile network consists of various applications and business models. 5G Networks aim to increase services, built on high speed, high bandwidth, and low latency networks. There has been shared interest in *Software-Defined Network*ing based service chains of Virtual Network Functions in these past years. Moreover, the concept of service slicing has gained momentum. The concept is characterized by different physical hardware, which is shared, to obtain multiple instances, which are isolated. Furthermore, a 5G network slice will be composed of many services, each for a specific purpose; for example, mobile data, which will

		have mobility, has one of its network function requirements. Compared to vehicle driving slice, which needs to meet low latency requirements. Talarico, Makhijani & Pillay (2016) proposed a Cloudcasting network, used to automate service discovery based on a shared IP network, used by many tenants to route traffic within a virtual network. The protocol describes Virtual Extensible Networks (VXNs) to sound the membership interests to a centralized designated authority, called Cloudcasting Rendezvous Point (CRP).
Bandwidth on Demand	(Bernstein et al., 2006)	The user's demand for high IP bandwidth or traffic engineering requires network protocols configuration over the wide-area network. Bandwidth on Demand services consists of shorter hold times provisioned on the communication flow. The single-layer approach to IP traffic management and bandwidth on demand stems from overall high capacity but not over-provisioned IP network. The selection of the Interior Gateway Protocol is an effective way of adjusting its parameters for the traffic behaviour of the network. Link weight values are distributed to the routers “in charge” of links from the IGP point of view. The multi-layer networking for IP TE and BoD at the IP layer need to take into account the following: • The WDM layer network topology, • The intermediate layer network topology, • The resources available in the IP layer, and • Allocation of IP bandwidth to the appropriate IP flows.
Real-time network evaluation	(Andreoli et al., 1996)	The increase of services on the network, such as video conferencing, Video on Demand, Interactive TV, and telephony over the internet.

		Created the employment of the Internet Engineering Taskforce to enhance the evolution of IP The internet is connectionless to support best-effort data communication of services for real-time services, a framework built on the Integrated Services on Internet model. The model lies in Resource Reservation. With the use of extended RSVP, there are two methods revised; the destination is an IP multicast group address, and the destination is a unicast IP address. However, RSVP signalling is not interoperable, and further requirements are needed.
Service migration and maintenance windows	(Jaumard et al., 2016)	Network swop migration could take months to years; the process becomes complicated when incorporating another service provider's nodes. The process starts with a massive load of planning, followed by lab testing. Network/Service migration describes a technical way of moving to a more efficient network. Jaumard et al. (2016) describe an optimization model that estimates the number of maintenance windows required for network migration. The proposed optimized model on the concept of shift configurations. Where a shift configuration is defined as a potential set of migrated circuit endpoints during a maintenance window.
Energy efficiency	(Wiatr et al., 2015)	Wiatr et al. (2015) study energy conservation within the telecommunications network. The study estimates the communication industry uses 1,8 % of electricity consumption, moreover, a 10 % increase annually. Its knowns to switch devices on idle mode to save electricity usage. Which is not economical today. Wiatr et al. (2015) discuss two strategies:

		the maximum allowance lifetime decrease and minimum time a device should be kept off to save enough energy to compensate for the reparation costs of a single failure. The above strategies affect the performance of the network. Erbium-Doped Fiber Amplifiers (EDFA) used within the system possess better performance than both approaches in the presence of energy efficiency.
Network configuration and setup	(Graur 2017)	The internet is made of physical and virtual components; sensed data is collected and processed. Which automatically triggers actions or services that may or may not include human interaction. The proposed solution brings a security challenge that will require a multi-layer protection scheme. The articles use the SDN controller to communicate its SDN switches, monitor the network for faults, and perform new configurations. The SDN Controller is used to reconfigure the SDN network devices. The controller integrates two modules, NetModel and ModFloodlight. NetModel builds an in-memory representation of the network description. While ModFloodlight, Application-Controller Plane Interface is for communicating with the Floodlight through the controller's REST API.
Co-ordinated restoration reversion	(Ruepp et al., 2008)	The restoration process is dependent on wavelength routing. Optical networks are not eliminated from failures; in this, traffic is still required to reach its destination on the free available wavelength. Wavelength Conversion is often the bottleneck for connection restoration. Ruepp et al. (2008) investigate the nodal stub-release method to solely release the wavelength along the stub path while keeping the span resources occupied in terms of achieved restoration percentage. The procedures

		mentioned no stub-release, full stub-release, and Nodal stub-release. Simulation results show that the method performs well in both dense and sparse topologies.

2.3 *Smart Grid*

Smart Grid is an electrical network comprising control, communication, and advanced monitoring. The *Smart Grid* consists of various physical components and cyber systems that encompass challenges mentioned by Chren, Rossi & Pitner (2016). Figure 2.7 shows the *Smart Grid* members, composed of smart appliances, energy storage, greenhouse gas reduction, and information and communication systems. The *Smart Grid* requires a new thinking mind of the interactions between the users, the cyber influence, and the power network. The outcome is to control components of the *Smart Grid* from the SDN controller.

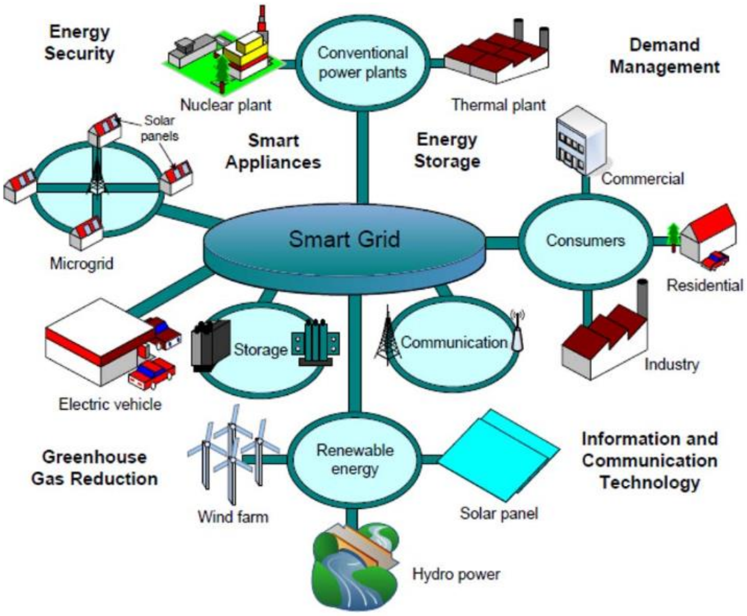


Figure 2.7: *Smart Grid*

(Yu, 2011, p. 1059)

When integrating physical and cyber systems, some challenges arise, as mentioned below,

2.3.1.1 Architecture and Design

For seamless integration into the communication network, Du et al. (2021) mentions careful planning of the communication design and deployment for control and requirements for computation. The process will encompass the capacity calculations of power networks and their integration into communication protocols—equipment in its ability to plug and play fashion. The area of cyber security becomes a point for development, considering the evolution of technology, smartphones, smart meters, and intelligent features. The scope needed will define new standards and procedures to protect the customer and the provider's assets. Section 3.2 presents the integrated architecture of Smart Grid and SDN communication network.

2.3.1.2 Information Science and Engineering

The combination of intelligent systems adds value to physical and cyber systems for data processing, data sensing, the need for control, and security enforcement. In terms of smart meters, the cost has reduced and increased in its processing ability. This feature allows real-time communication, which calls for a data network with the following properties (Barai, Krishnan & Venkatesh, 2015):

- Ultra low latency,
- High availability, and
- Layered security.

The real-time communication requirements ensure data is transmitted and received at the highest speed and maintain data integrity. Ultra-low latency is a given on a 5G network due to the frequency spectrum advantages and network architecture. Based on research and disruptive technology advancements, Multi-Edge Computing(MEC) becomes the solution, allowing Gi LAN functionality to become features and sit at the access network's edge discussed in **Appendix 1**.

Benefits of *Smart Grid*

Colak, Ayaz & Ahmed (2021) provide the benefits of the *Smart Grid* migrates to a more resilient network that consists of communication equipment to enable monitoring, control, and intelligence built into the systems.

- Improving the quality of power, the demand of the customer/industry needs, and troubleshooting for a self-healing network.
- Increase in bandwidth to enhance the traditional model.
- Control and monitor the placement of a *Smart Grid* Controller to provision the network.
- Embrace distribution and substation automation.
- Caters for the facility as a Service, new framework requirements.
- The improvement of *Smart Grid* security and security techniques.
- Customers are enabled to provision and manage their network.
- Exploration of new services and technology.
- Energy storage options for Electric Vehicles and modern storage.

2.3.2 Challenges and Solutions of *Smart Grid*:

Many countries face the challenge of the correct time to invest in *Smart Grid*, which depends on the growing energy demand versus infrastructure productivity. Duan, Zhao & Guo (2020) take careful decisions that are required to establish a resilience *Smart Grid* that caters to the increasing demand for communities, factories, and investment opportunities. Power Service Providers continuously scale and upgrade their grid, ensuring Key Performance Indicators are met. Various departments plan the distribution and supply of power to their customers, the constant battle of repairing faults while maintaining a solid-state provision of power.

2.3.2.1 Traditional infrastructure

The traditional or old equipment is not compatible with an advanced communication network. The principle of each device on the network requires its unique IP address. For the markets to cater to this, it drives businesses to use IPV6 network addressing mentioned by Mollah et al. (2021), providing billions of IP addresses. So that all devices or infrastructure can be connected to the network. The investment will require smart appliances, smart meters, storage, transmission infrastructure, and advanced software. This directs the strategy to SDN technology; one of the characteristics of SDN is building the platform on white boxes that run various Operating Systems. According to pre-announce policies, the *Smart Grid* will have robust metrics in taking up actions and resolving faults. Another solution is forming a strategy team to prepare a forecast of power growth, future projects, and new scoping metrics, a new process that will assist with new investment for the intelligent grid.

2.3.2.2 Quality Supply to Households

To protect lower current devices at the end state. Nazirov et al. (2021) evaluates he grid needs to cope at peak times and provide quality supply to the customers. There are many techniques for improving the supply, such as power factor correction. Intelligent optimized planning is required to ensure that redundancy is built into the grid when a failure occurs. This takes a deeper consideration when supplying a hospital or life support home.

2.3.2.3 Loss of Transmission and Distribution Energy

Due to the weak grid, the losses incurred causes a loss in revenue. Other losses include power theft; the Service Provider needs to take more significant measures to deal with threats and pressure the government to order and mitigate injustice by Rui et al. (2021). Serious, effective measures are required to be put in place. Reduction measures will include a risk audit of each system, analysis of payment vs. usage, improved methods, and employee company privacy of information.

2.3.2.4 Renewable Energy Integration

The fluctuating and unpredictable nature of renewable energy sources such a solar and wind power proposes technical challenges. It requires complex technology mentioned by Valencia-Calvo, Olivar-Tost & Garcia-Ortega (2020)

—the need for protective devices to serve as the middle ground when challenges occur. For a *Smart Grid*, the need for high power processing Integrated Circuits (IG) is mandatory. IBGTs are known for their fast switching ability, which leads to integrating a communication network to the grid. On the one hand, engineers frequently embark on technology solutions to reduce latency.

Filters circuits to produce a reliable and stable flow of electric power. In contrast, engineers embark on faster switching techniques to cater to the market in the power sector. With correct product selections, the integration of renewables will simplify the process of its smart characteristics.

2.3.2.5 Interoperability and cyber security

Interoperability standards are developed by The National Institution of Standards and Technology(NITS), which refers to Advanced Meter Infrastructure(AMI) end-to-end security, revenue models, inter-control communication, building security, management applications, and many more. Before, companies planned their grid, and today there are well-defined standards to shape the grid into a global grid. Cyber security should be built into the grid on planning, and during testing phases, the use case of self-recovery should be tested, considering different scenarios. This way, the engineers can thoroughly test the performance of the grid. Another technique is a planned attacks onto the grid, introducing a lockout state to the attacker. Would good planning and testing the grid will be able to perform even under attack.

Smart Grid development is a continuous process, the result of providing quality flow to the customer.

Vineetha et al. (2014) highlight the benefits of the *Smart Grid*, to list a few, Uninterrupted Power Supply for all households, reduced transmission, and distribution loss.

Further add, high penetration of renewable energy sources, cyber secured electrical grid, large scale energy storage, flexibility to consumers to interact with the electricity market, market-based electricity pricing, and demand-side management.

The *Smart Grid* concept design is to have intelligence on the grid to handle the non-forecasted load and distributed resources using information and communication technology with the utility of smart meters and a control system.

In addressing the benefits of *Smart Grid*, there is also a list of limitations and need for tremendous developments,

- It becomes a need to create a new communication infrastructure, which requires far higher advanced features on the system and component-wise.
- Another issue is the supply quality; due to the power demand, it becomes more crucial to improving the quality.
- The concern is placed on the fluctuating and unpredictable nature of renewable energy sources like solar photovoltaic and wind energy to be integrated into the power grid.

- Furthermore, the limitation of cyber security to be integrated includes advanced metering infrastructure and *Smart Grid* end-to-end security, revenue metering information model, building automation, inter-control centre communications, substation automation and protection, application-level energy management system interfaces, information security for power system control operation.

2.3.2.6 Failure, Metrics, and Costs

The topic of network link failure is a constant topic that requires consistent planning. Network engineers audit the network to ensure each link has a set threshold to avoid traffic congestion on an interface. Service Providers ensure their management system is updated continuously; the role of the Network Management Group compiles a list of nodes and links on their network; they perform the routine checks on the physical layer. Section 4.3.2.1 shows how SDN can perform better during routing and failure scenario.

The view of failure, when applied to internetworks, raises concern about how failure can affect the next network dependent on the other. Rastegarfar et al. (2015) emphasize the transmission network that interconnects the Core Data Network. The main emphasis points to a failure of the controller, which will affect the routing of packets increase packet loss. A failure can be categorized as hard or soft failure; a hard failure is known when a node is dead without its control. On the other hand, a soft failure model is when a node is dead but prevents reconfiguration and leads to static operation.

Soni et al. (2017) developed a testbed to show the effects of high traffic and the performance of the network; Figure 2.8 shows when workload traffic increases, the L2BM increases the percentage of Critical links by 2-3%, but in parallel increases the Bandwidth Acceptance Ratio by 8-10%. Besides, the number of Critical links is increased. L2BM can increase guaranteed-bandwidth multicast requests better than other algorithms by using the threshold-based technique. Hence, it can more efficiently utilize the allocated network bandwidth on the links.

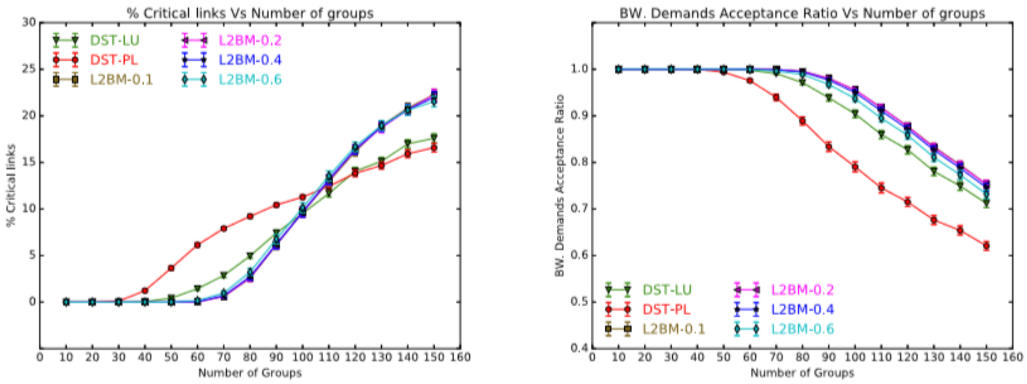


Figure 2.8: Percentage of Critical Links and Bandwidth Acceptance Ratio

(Soni et al., 2017)

With *Software-Defined Network*, the ability to automate the network is becoming more accessible and more programmable, simplifying network operations to human-readable language. The move from manual network configuration to a more automated way of network policies takes ground, creating room for change. OpenFlow allows the programmability of devices, known as a protocol that establishes the standard for the SDN controller to communicate to devices defined by Mckeown et al. (2008). Lara & Ramamurthy (2016) propose OpenSec, an OpenFlow-based security framework that allows the implementation of policies on network devices. OpenSec is a piece of software running on top of the SDN controller and multiple security services running, firewall, Intrusion Detection System (IDS), Deep Packet Inspection (DPI). A policy consist of a description of the flow, a list of security policies, and how to react to malicious content.

Bossart & Bean (2011) approach to determine metrics, cost, and benefits on *Smart Grid* projects. By the American Recovery and Reinvestment Act provides resources to fund field projects and demonstration projects. The data collected from *Smart Grid* projects shows that the benefits of *Smart Grid* greatly outweigh the cost of implementation.

Metric are reports calculated on the baseline, project-level metrics, and system-level metrics. The benchmark should reflect the specification, which includes historical performance data. Project-level metrics point to the technologies used and their effect on the operation, while System-level metrics point to-rated existing technologies.

Utilities can benefit from *Smart Grid* by improved operations, including more accurate and automated metering and billing, better outage management, reduced electrical losses, better assets maintenance, improved maintenance, and an improved planning process.

There are some significant challenges when performing the analysis of metrics, costs, and benefits. These challenges include:

- Establishing baseline data for the performance review;
    - Collecting data on location;
    - Determining societal benefits;
- Monetizing benefits;
    - Interpreting *Smart Grid* data to electrical distribution;
    - Comparison differences between the Service Provider and consumer; and
    - Using appropriate assumptions and calculation methods.

2.3.3 Smart Metering in *Smart Grid* framework and software model.

Due to the consumer's demand, there is a rise in greenhouse gas emissions and carbon footprint, leading to climate change and further environmental issues. The conventional grid consists of electromechanical components; however, the *Smart Grid* brings the communication network integrated within the power grid; there is one-way communication in the existing power grid. In contrast, the *Smart Grid* provides two-way communication, allowing the consumer to access the data.

Bansal & Singh (2016) propose to use information technology to overhaul the electric grid. Using solar and wind power, the consumer can have the features to combine energy efficiency with their power supply. The *Smart Grid* allows all the devices and equipment used to transmit power connections in the network. The *Smart Grid* allows each component and system to be self-monitored and to a level of troubleshooting a self-healing network.

Singhal & Saxena (2012) propose a new *Smart Grid* Monitoring Model to assist in understanding the *Smart Grid* deployment and capabilities within electric utility companies. The main characteristics built on this model are mentioned below:

- New products and services – the *Smart Grid* intends to introduce opportunities for new products and services.
- Power Quality – describes availability, voltage stability, resiliency, and self-healing features in more significant terms.

- Generation and storage options – in terms of wind, solar, and geothermal sources with the power sector.
- Consumer participation – the level of involvement from the community and users.
- Operational resiliency against disaster – long-term scoping for potential hazards.
- Asset optimization and operational efficiency – ability to monitor the real-time basis of each system.
- Response to disturbances – The amount of time minimized to responding to system outages.

Furthermore, the study of Singhal & Saxena (2012) introduces the *Smart Grid* Interoperability Maturity Model (SGIMM), *Smart Grid* Investment Model (SGIM), *Smart Grid* Maturity Model (SGMM), and *Smart Grid* Conceptual Model (SGCM).

SGIMM provides features like status/progress measuring statistics, gap analysis, and prioritization of efforts to improve the current.

SGIM provides feature sets:

- The complete framework for quarterly details costs and benefits computation,
- Forecast of impacts of *Smart Grid* implementation program on customers and the end-user,
- Guidelines for better intelligent grid investment analysis, and
- Suggestions regarding *Smart Grid* strategies, which are cost-effective.

Whiles SGMM provides features:

Developing a shared *Smart Grid* vision and guidelines, communicating with different stakeholders using a common platform,

- Assigning different tasks as per proper precedence,
- Monitoring and measuring progress in various domains, and
- Developing new and modified plans if changes are required.

And finally, SGCM provides features like:

- Analysis of standards,
- A process of interactions in different domains and
- Pure focus on cyber security , network management, data management, and application integration.

2.3.4 Investment-benefit analysis and evaluation model of the *Smart Grid*.

Jianming et al. (2010) analyze the benefits of the *Smart Grid*, which is based on a stable grid and supported by communication and information platforms; the integration caters to power flow, information flow, and business flow using intelligent control. The integration was built within the components of power generation, transmission, substation, distribution, transfer, users, and communication information of the traditional grid.

The 'smart' will add many benefits to the power grid for the business and the consumer; however, it comes with excellent capital investment.

The *Smart Grid* has a few economic advantages in the following aspects mentioned below:

- The improvement of the operating performance uses improved technology; the service provider can take better control of a complex power system, reduce the necessary investment costs, reduce operating costs, reduce maintenance costs, and increase the service life of the equipment.
- Promoting the consumption demand for electric power. The consumer has created a demand for power quality and improved power supply reliability; the *Smart Grid* caters to autonomy, self-healing, and defence.

Concerning this relation, Yuezin et al. (2010) propose an investment cost evaluation model, when

TC is the total investment for intelligence;

sum TC_i is basic investment costs reduced by intelligent power grid;

r is interest rate;

A is fixed assets;

c_I is the ratio of maintenance costs of capable fixed assets.

$$\overline{TC} = ({}^{TC}\!/_{t} - \Delta TC_i) + TC * r + A * c_I \tag{1}$$

2.3.5 Application of Power Line Communication in smart power consumption.

State Grid Corporation of China built the strategic goal for *Smart Grid* to enhance the power grid's comprehensive service capabilities, improve power efficiency, and promote energy conservation and emission reduction.

The *Smart Grid* can cater to the two-way communication channel between electricity users and the service provider to obtain their user data. Jianming et al. (2010) state the communication requirements for the *Smart Grid*, and there is a demand for faster speed internet connectivity. As a result, households migrate to FTTH (Fiber To The Home). In contrast, wireless communication is known for its easy installation but requires higher levels of security. While Power Line Communication uses the existing power line resources to communicate, it possesses the strong anti-jamming capability and adequate data transmission security, making it easier for two-way communication.

The State Grid Corporation of China has built two smart power consumption pilot projects in Beijing, located in Lianxiangyuan District and Yard No. 95 Fucheng Road. Figure 2.9 shows the Lianxiangyuan Pilot Project that consists of the collecting master station, the property management master station, concentrator, collector, smart interactive terminal, smart sockets, home security equipment, and other telecommunication value-added service.

The project scope is to collect power consumption information using a family smart interactive terminal built on wireless technology to further achieve other results in meter reading, water, and gas. The system collects and controls water heaters, air conditioners, rice cookers, and other home appliances.

The system incorporates home security, which combines emergency calls, gas leak detectors, smoke detectors, and infrared detectors.

The technology mentioned features high bandwidth, the meter reading occurs every 15 minutes on the power collection system, and the collector can make real-time responses.

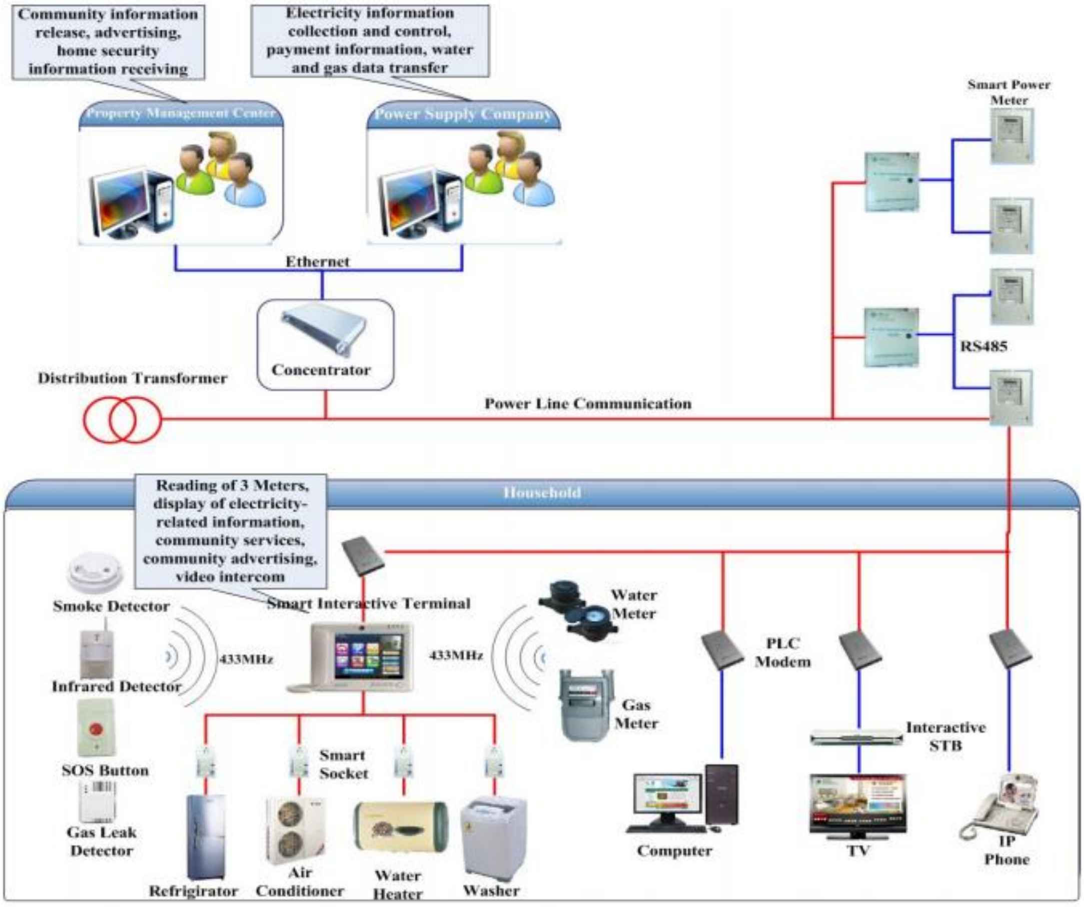

Community information release, advertising, home security information receiving
Electricity information collection and control, payment information, water and gas data transfer
Property Management Center
Power Supply Company
Smart Power Meter
Ethernet
Concentrator
Distribution Transformer
Power Line Communication
RS485
Household
Reading of 3 Meters, display of electricity-related information, community services, community advertising, video intercom
Smoke Detector
Smart Interactive Terminal
Water Meter
433MHz
433MHz
Infrared Detector
PLC Modem
Gas Meter
SOS Button
Smart Socket
Interactive STB
Gas Leak Detector
Refrigirator
Air Conditioner
Water Heater
Washer
Computer
TV
IP Phone

Figure 2.9: Lianxiangyuan Pilot Project

(Jianming et al., 2010)

The traditional network architecture's current requirements are to overcome ossification, to support dynamic network services and applications. Network virtualization is considered the primary solution to addressing ossification in the light of resource allocation and management. Zong et al. (2018) created three topologies and created different algorithms to measure the performances on each network. The First Available (FA) computing resource and Local Node Ranking (LNR) produced the benchmark results.

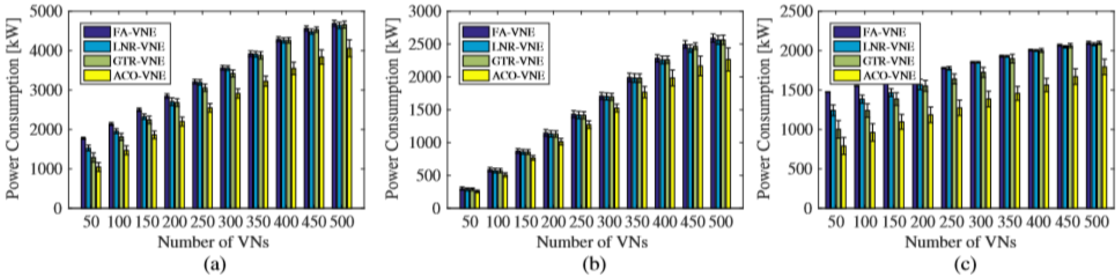


Figure 2.10: Comparison of power consumption. (a) Total power consumption, (b) Network power consumption, and (c) DC power consumption

(Zong et al., 2018)

The test shown in Figure 2.10 was performed on several Virtual Nodes to calculate the power used from the four algorithms. The results showed that Global Topology Resource – Virtual Network Embedding GTR-VNE obtained 9.3% and 5.1% improvement compared to the benchmark results. The property of the GTR node is to assist in decreasing power consumption.

2.3.6 *Smart Grid* oriented smart substation characteristics analysis and capacity planning.

Jin-Lun et al. (2012) provide an in-depth definition of a smart substation divided into three layers: process, spacer, and control layer. The process layer consists of intelligent equipment, merging unit, capable terminal, substation power distribution, transmission, transformation, measurements, control, protection, and other related functions. The spacer layer is built of relay protection, frequency, and control device. The station control layer contains an automation system, station domain control, communication system, complete supervisory control and data acquisition (SCADA), information management, and other related functions.

Within the Telecommunications network, we see higher peak traffic; it becomes a challenge building the capacity network and catering for full-redundancy—the time taken to plan and deploy fiber capacity compared to the demand of service applications. While over-provisioning, the network leads to an increase in Operational Expenses. Alvizu et al. (2017) proposed a phased approach to introduce machine learning and promote dynamic bandwidth provision.

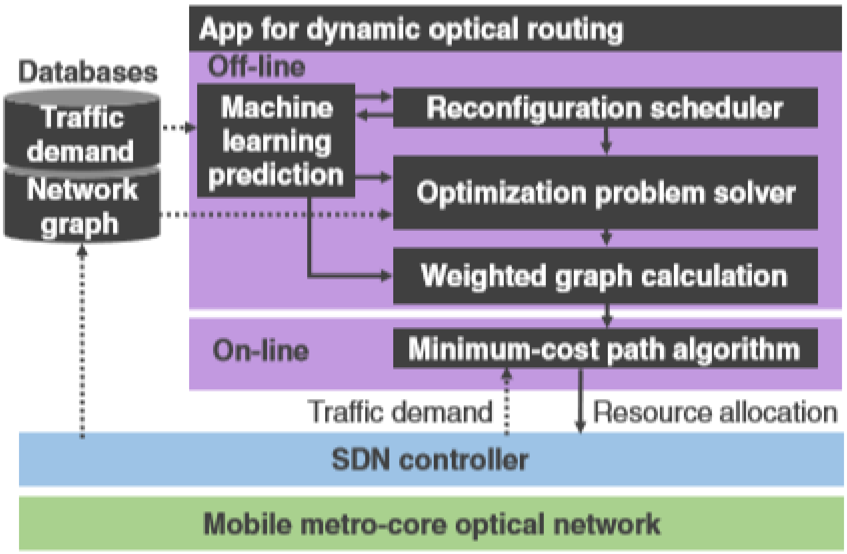

App for dynamic optical routing
Databases
Traffic demand
Network graph
Off-line
Machine learning prediction
Reconfiguration scheduler
Optimization problem solver
Weighted graph calculation
On-line
Minimum-cost path algorithm
Traffic demand
Resource allocation
SDN controller
Mobile metro-core optical network

Figure 2.11. Dynamic optical routing metaheuristic for the software-defined mobile metro-core network

(Alvizu et al., 2017)

From Figure 2.11, we identified the following phases:

- Off-Line Scheduling: Use to estimate the network traffic and plan reconfiguration of nodes,
- Off-Line Planning: Used to calculate and predict the reconfiguration interval by the adjustment of network weights, and
- On-Line Routing: Used to build a physical topology to compute the online routing decision.

The results have shown the reduced power consumption on the network nodes.

2.4 Security

2.4.1 Threats

The communication network gained more significant influence during the world pandemic of COVID-19, where technology serves as the best gateway of virtual communication and safety measures. The rate of threats increases, equivalent to the high traffic peak on a service providers' network. Under the assumption, the person/s behind the threats must be identified. Research shows that an attacker's purpose is to gain a reputation for bringing down a particular system, showing their skills, and resulting in revenue loss.

Another attacker could be the end-user, the customer seeking to interrogate the network by using the system and getting away from being billed. Another attacker is labelled to jeopardize the billing system, causing less revenue and higher traffic volumes. The form is a voucher system that allows the customer to benefit from free rewards in gigabytes.

Employees who intend to make mistakes on the core network purpose create many disruptions of network services. Finally, the power sector can bring down the power of a site, allowing a no signal or connection approach.

- The attacking approach is careful and targeted at peak times to ensure the network fails and proves the attacker's ability.
- Message spoofing, sending false messages to the end-user, intends to draw the customer to a winning item and request the customer's privacy pins.
- Baseline response replay, replaying authenticated messages back to the master.
- Direct slave control, the ability to remove access from an authorized entity.
- Network scanning, ability to request network information, seemingly from a trusted network.
- Response delay, purposefully delaying a key message on the master control.
- Rouge interloper, attacking a machine with correct port allocation.

From the 5G reference architecture in Choi, Kim & Park (2016), Figure 2.12 shows the different interfaces on a 5G network prone to attacks.

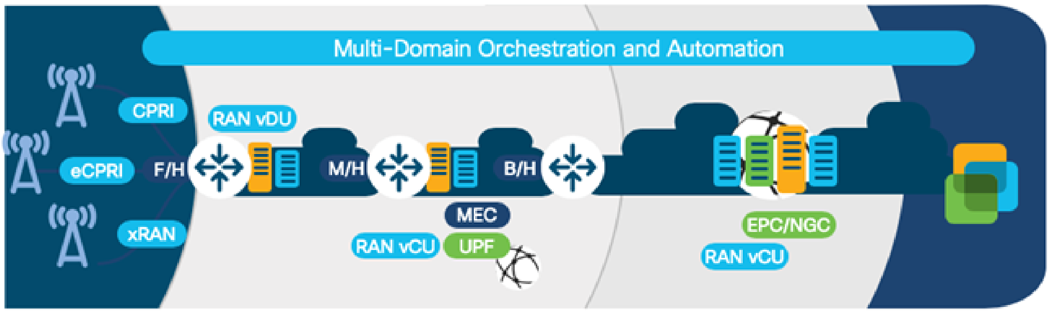


Figure 2.12: Reference 5G Architecture

A 5G network is also prone to attacks listed below; within each, a solution is required:

- Attacks with physical access to xRAN and eCPRI,
- Attacks by mobile endpoints created by DOS flooding,

- Attacks on the radio interface designed by jamming,
- Attacks with physical access to the transport network created by Man in the middle attacks,
- Insider attacks made by data modification and data leakage,
- Attacks from other mobile networks created theft of service or eavesdropping, and
- Attacks from external systems created a compromise of the network element.

Table 2.6 shows the effect of an attack in a network; it will result in a loss of revenue.

Table 2.6: Effects of an attack and identified threats

(Shu et al., 2020)

Category	**Threat**	**Description**
Loss of link availability	Flooding an interface	DDoS/TDos via mobile end-points
	Crashing a network element	DoS/DDoS via rogue media streams and malformed.
Loss of confidentiality	Eavesdropping	Eavesdropping via sniffing the Gm interface.
	Data leakage	Unauthorized access to sensitive data on the IMS.
Loss of integrity	Traffic modification	Man-in-the-middle attack on the Gm interface.
	Data modification	SIP messaging impersonation via spoofed SIP messages.
Loss of control	Control the network	SPIT (Spam over Internet Telephony)/unsolicited voice calls resulting in Voice-SPAM/TDos.
	Compromise of network elements	Compromise of network elements via attacks from external IP networks.

Malicious Insider	Insider attacks	Malicious insider makes unauthorized changes to IMS configuration.
Theft of Service	Service free of charge	Theft of Service via SIP messaging impersonation.

Security challenges in a virtual environment are depicted in Figure 2.13; Kim et al. (2020) expand on the resource connection between the virtual and physical environments. The below mention network vulnerabilities within a software-driven virtual environment.

- Hypervisor vulnerability,
- API security,
- Orchestration vulnerability,
- Virtual monitoring, limited visibility to mobility, and EPC interfaces.
- Virtualized firewalls,
- Secure boot,
- Secure crash,
- User/tenant authentication and accounting
- Topology validation and enforcement
- Performance isolation,
- Authenticated time service,
- Private keys within cloud images,
- Detection of attacks on resources in virtualization infrastructure,
- Security monitoring across multiple administration domains – lawful interception.

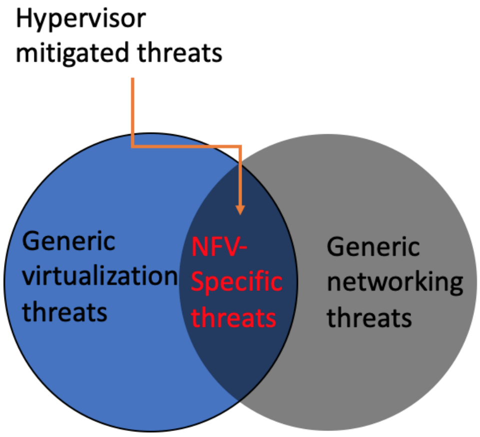


Figure 2.13: Virtualization resource relational dependency.

Figure 2.14 shows the components from a virtual environment that require management in terms of security threat protection.

To reduce security attacks, the following measures need to be considered and continuously applied,

- Conduct security scans and apply security patches,
- Ensure the hypervisor is hardened and minimized, providing vulnerable ports are closed,
- Assuring access to the hypervisor is controlled via User Access Management.

Malware compromises Virtual Machine undertook by VM/Guest Operating System manipulation, data exfiltration/destruction. A hacker exploits a vulnerability in the open-source code and infects the hypervisor with malware.

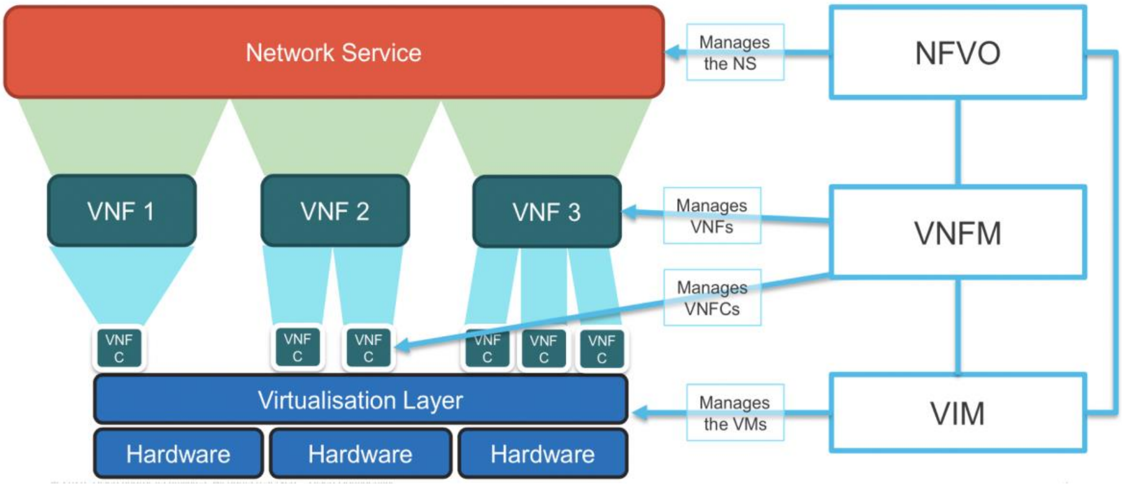


Figure 2.14: Virtualization environment layers

The trust model is established for decision-making and communication between intelligent devices, with its ability to sense when to trust and when not to trust. The intelligence came for an added security layer within the protocol stack and was configured among standards. Security systems in terms of Intrusion Prevention Systems and Intrusion Detection systems become the pillar to solve problems.

2.4.1.1 SDN Network Security Solutions.

According to section 1.4 objectives the proposed approach of Securing the SDN layers follow:

A protection measure is to use an Out Of Band (OOB) network to control traffic. Theodorou & Mamatas (2020) examine how easier and less costly to construct an OOB network in a data centre than across an enterprise WAN. Using an OOB network for the northbound and southbound communications could help secure the protocols for controller management.

Using TLS or SSH or another method to secure northbound communications and secure controller management would be considered a best practice. The communication from the application and services requesting services or data from the controller should be secured using authentication and encryption methods.

Secure coding practices for all northbound applications requesting SDN resources should be a best practice. Not only are certain coding practices beneficial to the security of public-facing Internet web applications, but they are also applicable to the northbound SDN controller. Some SDN systems can validate the network against controller policy.

2.4.1.1.1 Securing the Controller

Access to the SDN controller must be controlled to prevent unauthorized activity. Role-based access policies that are audited and revised consistently should be used. Any unauthorized attempts should fire up alerts to the Network Fault Management Group(NFMG). Also, configuration changes require to be audited and reviewed regularly.

Best practices for hardening and patching the system should be in place. If the best procedure or security standard is not followed, the risk and potential impact should be documented. It is essential and essential to plan with high-availability controller architecture to prevent distributed denial-of-service (DDoS) attacks. The benefit will allow the testing and updates in a live environment. As well as an immediate failover if the change does not work correctly.

Karmakar et al. (2020) recommend controller security techniques mentioned below:

- Management security,
- Authentication, Authorisation, and Accounting,
- Strong passwords,
- Transport Layer Security, and
- Physical security.

Underlying Operating Software Security

- System patches and fixes,
- Strong password,
- Disable unnecessary protocols, ports, and devices,
- Authentication, Authorisation, and Accounting, and
- Enable host-based firewall and only allow required communication ports.

2.4.1.1.2 Securing the infrastructure

This layer is crucial in protecting the equipment and preserving secure operations; network engineers will need to keep the focus on the following:

Operational

- Keep the Operating device System up to date,
- Centralize log collection,
- Configuration management, and
- Physical security.

Management Plane

- Use secure protocols to manage infrastructure: SSH, HTTPS, SNMPv3 with ACL to restrict access.
- Control management and monitor sessions with AAA.
- Use an encrypted local password.
- Disable unused services or interfaces on shutdown mode.
- Authenticate tunnel endpoints and secure tunnelled traffic.

2.4.1.1.3 Securing the Application

- Application Security
- Digital signing of code.
- Certification on the process.
- AAA.

Underlying platform security

- System patches and fixes.
- Strong passwords.
- Disable unnecessary protocols, ports, and devices.
- Authentication, Authorisation, and Accounting.

- Enable a host-based firewall and only allow required communication ports.

2.4.1.1.4 Securing the API and Communication

The single most effective defence against parameter manipulation and injection attacks is to validate all incoming data against a strict schema –virtually describing what is considered permissible inputs to the network. Schema validation should be as restrictive as possible, using typing, ranges, sets, and detailed whitelisting listing whenever possible.

Good schema validation can protect again anyway injection attacks but also consider explicit scanning for common attack signatures. SQL injection or script injection attacks often betray themselves by following common patterns that are easy to spot by scanning raw data. Consider also; attacks may take other forms, such as a denial of service. Extensive messages, heavily nested data structures, or overly complex data structures can result in an effective denial-of-service attack that needlessly consumes resources on an affected API server. Leverage networking infrastructure to spot and mitigate network level DoS attacks and check for DoS attacks that exploit parameters.

Finally, the critical security benefits of SDN serve and encourage the strategy of *Smart Grid* vendors to incorporate and build an efficient network, providing the following: - untethers policies from the physical perimeter, policy management, and enforcement for diverse multi-tenant environments traffic steering and path management that accelerates detection and isolation of threats programmability -enables automation and adaption to mitigate risks, and open interfaces to foster multi-vendor interoperability.

2.4.2 Cyber-security in *Smart Grid*: Survey and challenges.

Mrabet et al. (2018) mention some of the significant shortcomings of the electricity grid, namely high cost and expensive assets, time-consuming demand response, high carbon emission, and blackouts. A study conducted by Berkeley National Laboratory in 2004 showed that power interruptions cost the American economy around $80 billion per year discussed in Knapp & Samini (2013). The critical problems cannot be addressed in the current system. At the same time, the *Smart Grid* promises to provide flexibility and reliability by integrating new power resources and enabling corrective capabilities such as renewable wind energy and solar energy. The *Smart Grid* defines as a system to collect information or data in the generation and delivery—nevertheless, Mrabet et al.

(2018) raise the risk that exists in the *Smart Grid*; any interruptions in power generation could disturb *Smart Grid* stability and will impact people living within that location. In addition, there is an exchange of valuable data, theft, or alteration of this data that will disturb consumer privacy.

More precisely, Rawat & Bajracharya (2015) provide details on the *Smart Grid* as a prime target for cyber terrorism. As a result, cyber security is gaining more attention from governments, energy industries, and consumers. According to the National Institute of Standards and Technology (NIST) conceptual model for *Smart Grid*, communication networks connect power system components shown in Figure 2.15. (NIST special publication 1108, 2010). There are seven logical domains: Service Provider, Operations, Markets, Customer, Distribution, Transmission, and Bulk Generation. The bottom four deal with power and information flows, and the top three deal with data collection and power management in the *Smart Grid*.

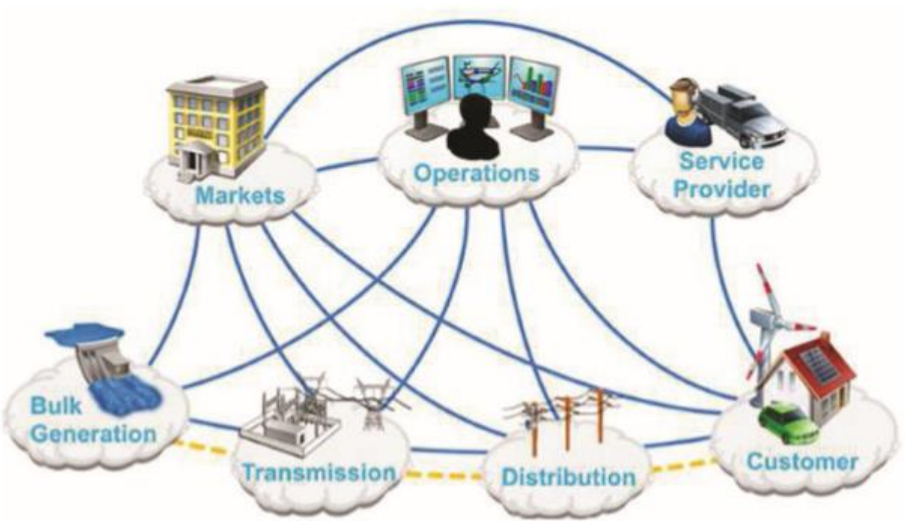


Figure 2.15: The NIST conceptual model for *Smart Grid*

(NIST special publication 1108, 2010)

The *Smart Grid* consists of various components installed in the plant generation, distribution sites, and at the customer premises. These components are fall under high and low power devices, generation and distribution, measurements, and communication. Each part is connected to the next element in a *Smart Grid* to operate, monitor, and control the power flow. The traditional equipment lacks intelligence and has not been upgraded to meet cyber security attack levels. To provide an example, a malicious user gains access to a customer’s data on the network. There is a need to increase the intelligence within each device and overall the electrical and communication network.

The system provided encrypted authentication for the user and was built with a selected hash key generation to maintain a control session. Because of the problems stemming from the ever-increasing, Rawat & Bajrachachaya (2015) introduce the smart meter and its ability to have a PIN/passwords set, the PIN can be set once-off. If a password or authentication is required, the controller will require a change.

Based on the purpose of retrieved, Rawat & Bajrachachaya (2015) provide a comprehensive study of challenges in a *Smart Grid* security environment. The information will present the requirements for a *Smart Grid* system. A comparison between the traditional network and *Smart Grid* is required.

The *Smart Grid* system is dependent on generation, transmission, distribution, and consumer, and each is connected by either a Wide Area Network (WAN), Neighbourhood Area Network (NAN), or Home Area Network (HAN). A *Smart Grid* relies on wired and wireless communication networks, and each offers its security vulnerabilities. Rawat & Bajrachachaya (2015) present the requirements of a *Smart Grid* based on the challenges in the communication network.

- Latency requirements: *Smart Grid* networks require real-time communication and low latency. The network is required to meet a minimum throughput speed rate. Another option is to enrich the packet header to ensure each packet travelling end to end in the network has selected priority.
- Data size and flow: Due to the nature of the grid's high intelligence, information sent to the consumer requires to receive in real-time, even during peak times. To this, the option will introduce Quality and Class of Service.
- Password/PIN requirements: The correct type of password authentication needs to be configured, using private and encrypted public key hashing to ensure user data is uncompromised.
- Layered network architecture: Smart requires a revised network architecture to accommodate services and to provide centralized control.

- The *Smart Grid* requires being secure and complying with policies to secure information using Confidentiality, Integrity, and Availability, also known as the CIA triad developed by the *Smart Grid* Interoperability Panel.
- Confidentiality refers to user access that is provided to authorized people only. Privacy is one of the crucial issues for the customers and to prevent the misuse of information.
- The integrity of information in a *Smart Grid* is required to maintain the accuracy and consistency of their data. This feature is built to provide robust monitoring systems.
- Availability in the *Smart Grid* requires that the information must be available to authorized parties when needed without comprising security policies. Power systems are expected to be available 100% of the time.

Other security requirements for the *Smart Grid* include the physical security of grid assets.

- Self-healing and resilience operations in the *Smart Grid*, The grid requires to have features to self-heal from a cyber-attack. Thus the network must perform profiling and estimating to monitor the data flow and detect any abnormal incidents.
- Authentication and access control; each device connected to the network requires access control and is limited to several authorized users. Each user is required to follow the security processes in place.
- Communication efficiency and security, the network is required to be met the demands of real-time monitoring; design engineers are required to ensure minimum latency and improve the system using network protocols.

2.4.3 *Smart Grid* Cyber Security: Challenges and Solutions.

Shapsough, Qatan & Aburukba (2015) highlight the threats and challenges exposed to the *Smart Grid*; the article addresses various cyber security challenges, like connectivity, trust, customer privacy, and software vulnerabilities.

- Connectivity: There is a large number of devices connected on the network, each device plays a role in the process and controls the flow of data, one of the issues faced by a large number of devices, and it's challenging to manage and de-centralized. The system requires a high level of protection against attacks and vulnerabilities.
- Trust: There is a level of trust required from the consumer to abide by the respected policies and agreements.
- Customer's privacy: Ensuring consumers' privacy is an essential aspect in any system, including the *Smart Grid* that needs to be protected and stored. Large amounts of data will collect the customer's usage and actions. The data needs encryption upon encryption, so it will not be readable to an unauthorized party if the information is extracted.
- Software vulnerabilities: Software may suffer from weaknesses that include malware, Supervisory Control, and Data Acquisition (SCADA) systems are not excluded from the risk of malware and malicious updates. It opens up a gap to acquire software engineers having the skills set to minimize the risk and further recommendations to have the software update tested in a laboratory environment first.

Shapsough, Qatan & Aburukba (2015) provide details of network security and detections used. Denial of Service (DoS) is the most common attack in the *Smart Grid* network. When a DoS occurs, the user and administrators cannot perform any task on the network. DoS Detection and DoS Mitigation can handle DoS. Below is listed and explained the types of DoS Detection methods:

- Using flow entropy: The technique presented suggests sampling packets and measuring flow entropy to detect an attack.
- Using signal strength: A jamming attack can occur in two forms, a continuous amplified signal that will jam the link or a noise-like signal that will perform the same. Using signal strength will add a detector and set a threshold or use a decoder to compare the signal.
- Using transmission failure count: A transmitter or a receiver is used to detect jamming signal attacks.
- Once a DoS attack is identified, the engineers need to take corrective measures in a short time space to ensure the risk gets minimized for the protection of the devices on the network and data. Below is listed and explained the types of DoS Mitigation methods:
- Pushback: This method will block all traffic that matches the pattern of the attack by mapping its characteristics.
- Rate limiting: Once the attack has been identified, the router will limit the data rate for that user. The method uses a detector to identify the users.
- Filtering: The router uses filters to detect attacks and compare them against a detector's blacklist based on the source IP address. If a filter is positive, the active will perform a block.

Shapsough, Qatan, & Aburukba (2015) complete the article by sharing on network security protocols, which are internet-based protocols for secure communication such as Internet Protocol Security (IPSec) and Transport Layer Security (TLS). However, network security protocols require to be selected based on the design of the network architecture topology.

2.4.4 Overview of the *Smart Grid* cyber-security state of the art study.

Dari & Essaaidi (2015) mention conventional power systems are based on conventional resources, namely, oil, coal, and gas, to produce energy. The resources are consumed rapidly and becoming in time scare. According to Lu & Song (2010), US statistics show consumption and energy production have increased by two and three factors. These resources are becoming scarce in the future. They also emit carbon dioxide levels (C02), which is one major threat to the green environment.

Lara & Ramamurthy (2016) analyzed OpenSec, taking the opportunity to design a network testbed and create algorithms. Measurements are based on the time factor.

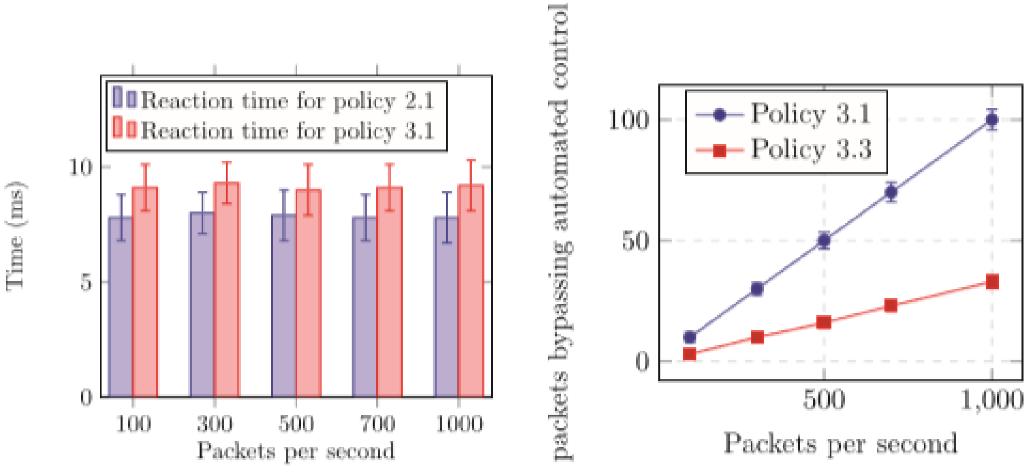


Figure 2.16: Detection and blocking rate

(Lara & Ramamurthy, 2016, p. 39)

Figure 2.16 shows the time difference of detection blocking of malicious traffic. The time taken by OpenSec to detect and block is equal to the time reaction in comparison. The linear graph shows the number of packets that pass the switch after being stopped. From this, OpenSec is more effective in the detection and blocking of malicious traffic. The contribution hides the complexity of security setup and management.

2.4.5 On the security of *Software-Defined Network*s.

Prasad, Koll & Fu (2015) discussed a variety of attack scenarios that reflect the presence of malicious hosts, switches, and controllers in an SDN environment. An attacker may cause harm to the network in the following ways,

Malicious hosts: A hostile host can perform attacks that include host location hijacking, link fabrication, DoS, or Man-In-Middle attacks. An example of a DoS attack is to flood the switch with a high volume of packets, effectively increasing the flow rules.

In the case of Malicious switches: A malicious switch creates an illusion of the network by changing the topology, which diverts network traffic to an unwanted destination—allowing the system to learn a non-existing link between two switches by manipulating the LLDP packets—leading to packet loss and poor network experience to the customer.

On the other hand, malicious controllers: A malicious controller may completely lose the network. A hostile controller can install flow rules on switches to re-route traffic, leading to a significant packet loss.

Each of the mentioned attacks requires a solution. Prasad, Koll & Fu (2015) identify ways of detection and mitigation. Topoguard verifies the IP address is legitimate to detect a compromised host while checking LLDP packet integrity and switch port properties mentioned in Hong et al. (2015). When dealing with a compromised switch, SPHINX creates a flow graph for every flow in the network, using the updated and original comparison to verify against policy change mentioned in Dhawan et al. (2015). A compromised controller is still in investigation.

Satasiya & Raviya (2016) perform research on Software-defined firewalls; a traditional firewall is software or hardware used to look at network traffic and the feature to allow traffic and block traffic. The firewall is placed between the private network and at the border of the public network. The firewall has robust policies in denying traffic. A careful security engineer desires the Access Control List, which allows specific traffic and blocks the other.

Compared with the Software-defined firewall, the critical design feature has a centralized control system to manage the traffic and provides flexible traffic control. The design of SDN is to establish an automated network, self-healing, self-monitoring, and self-blocking.

Satasiya & Ravya (2016) performed test simulation cases, used the POX controller and Openflow switch based on the IP address, MAC address, and source/destination address within layers 2, 3, and 4 of the OSI model. The results show the restriction of traffic and also raised the limitations of SDN below:

- Open access,
- Fraudulent flow rules, and
- Non-registered 3rd party access.

Further to the security vulnerabilities, Table 2.7 shows the treads occurrence on the control and data planes.

Table 2.7: Security attacks on control and data plane

(Bhardwaj, Panda & Datta, 2020)

Control Plane	**Data Plane**
DoS attacks – Centralized control is opposed to attacks.	Fraudulent flow rules – The policies defined require closing the loop and denying identified fraudulent flow rules.
Unauthorized controller access – The controller is used by a 3rd party and retained open access.	Flooding attacks – Each flow table has several flows.
Scalability and availability – The controller requires a design to be able to scale another layer.	Controller comprising – the data layer needs to be secure to disallow the hijacking of the controller.
	Man-in-the-Middle attack – SDN network has a complex TLS setup.
	TCP – Level attacks – Absence of TLS within layer 4.

Each proposed solution mentioned is subjected to limitations. The technical network team will require a higher level of knowledge to deal with issues considered.

2.5 Integration of *Software-Defined Network* and *Smart Grid*.

A *Smart Grid* system provides intelligent control within its unique system with its communication system extending to automation and machine learning from Vineetha & Babu (2014). The communication systems are either wireless or wired, which opens the window for security levels discussion.

Open-source applications are available for every common enterprise software type in databases, applications, network monitoring tools, security software, and web servers. In all the mentioned, mature commercial software alternatives also exist. An open-source application can be more secure than its commercial equivalents. Open source communities may seem chaotic and occasionally fractious, but they can be remarkably agile and cohesive when it comes. They've repeatedly shown they can do an excellent job discovering, characterizing, and patching security vulnerabilities. Besides, these community open source security practices are often backed by supplies that provide mature commercial support and indemnification. In summary, Open Source software has been proven to offer better value, lower costs, and improved security, addressing today's most critical enterprise considerations.

The research showcases a new architecture to integrate a communication network and *Smart Grid* network, perform control from the SDN controller, and serve for intelligence.

The success of a controller is going to be dictated by the following concepts:

- Multiple Southbound interface protocol support,
- Well-defined Northbound API support,
- Programmability,
- High availability and performance, and,
- Security.

To be applicable in a multivendor environment, multiple southbound protocols need to be supported. The northbound interface needs to be well defined so that applications can be quickly built and stacked on top of the controller. The accessible infrastructure and agile programmability of stability and performance are crucial points that need to be addressed for live large production networks. Typically, no single point of failure should be deployed, so controllers need to be installed on separate redundant physical appliances connected to the network forwarding devices through a secure and redundant management network. Security is essential, so the controller and the infrastructure connections need to be encrypted using standard protocols.

Table 2.8 shows the SDN controller considerations regarding performance and skills from Zhao, Iannone & Riguidel (2015).

Table 2.8: SDN Controller Considerations

	NOX	POX	Ryu	Floodlight	ODL OpenDayLight
Language	C++	Python	Python	JAVA	JAVA
Performance	Fast	Slow	Slow	Fast	Fast
Distributed	No	No	Yes	Yes	Yes
OpenFlow	1.0/1.3	1.0	1.0/1.4	1.0	1.0/1.3
Learning curve	Moderate	Easy	Moderate	Steep	Steep
		Research, experimentation, demonstration	Open source Python controller	Maintained Big Switch Networks	Vendor applicaton support

2.5.1 Network Discovery

Network discovery can mean a couple of things in a *Software-Defined Network*, including, but is not limited to, switches, routers, links, and hosts (Gu, Li & Yu, 2020). Figure 2.17 shows an enterprise network data flow consisting of network discovery, default path, and high availability.

When the switch establishes a Transmission Control Protocol (TCP) connection to a controller, the controller sends a feature request message to the switch and waits for a reply. When the reply reaches the controller, the controllers get informed about the switch feature, for instance, the datapath, list of ports, etc.

When a switch connects to a controller, the controller periodically commands the switch to flood Link Layer Discovery Protocol (LLDP) and Broadcast Domain Discovery Protocol (BDDP) packets through all of its ports. A discovery protocol packet typically contains the DPID of the sender along with the port of the switch that the message originates. The reserved set of destination MAC addresses and other types of discovery protocol packets lets the controller differentiate them from the other data packets. LLDP is used to discover links between switches, and BDDP is used to find the switches in the same broadcast domain. Using a combination of LLDP and BDDP packets, the controller discovers the switches' direct and indirect connections. Further, the controller also keeps an eye on the liveliness of the connections regularly with periodical checks.

2.5.2 Network Default Path

Traditional routing protocols such as Open Shortest Path First (OSPF) and Routing Information Protocol(RIP) use different computation algorithms to determine the best path through the network that a packet must travel. Routing Information Protocol (RIP) is a dynamic protocol used to find the best route or path from end to end over a network using a routing metric or hop count algorithm. This algorithm is used to determine the shortest time over the shortest distance.

Open Shortest Path First (OSPF) from Tao et al. (2021) is a Link State Routing Protocol (LSRP) that uses the shortest path first network communication algorithm to calculate the shortest connection path between known devices. OSPF performs an algorithm by first calculating the shortest path between the source and destination based on link bandwidth cost and then allows the network to send and receive IP packets. OSPF finds the best network layout by calculating the shortest device connection paths using the first algorithm's shortest track.

2.5.3 High Availability Controller – State Replication

The approach involves the setup of a master/slave cluster of controllers above the switches in a network. The entire cluster of controllers is represented by a single virtual IP that would be the primary controller's IP. This is responsible for network data flows, and other slave controllers will keep synchronized with the primary controller. If a failure occurs, failure detection will kick in.   After failure recovery, heartbeat messages get used to detect the failure of the primary controller. In contrast, the primary controller's failover to the secondary slave controller, controller network services, and application restoration and control network interfaces is up (Suartana, Anggraini & Pramudita, 2020).

2.5.4 High Availability Controller – Switch Point of View

This approach involves the setup of a cluster of controllers above the switches in a network. The entire cluster of controllers is represented by a single virtual IP that would be the primary controller's IP. The primary controller will be selected based on the priorities assigned to the individual controllers. The primary controller will be responsible for network data flows, and the other controllers will keep synchronized with the primary controller (Suartana, Anggraini & Pramudita, 2020).

If a failure occurs, the election algorithm is executed to determine the next primary controller.

- Initial Mode: The system starts with data flow through the network. The primary controller takes control of the network. Other controllers in the cluster establish a connection to the primary controller via the proposed Northbound API. Other controllers request the current network view and network interfaces list for control channels connection, current states of network services, and applications to the primary controller.
- Operational Mode: In this mode, the primary controller processes OpenFlow messages from network devices and controls network data flows; the standby controllers monitor the primary controller state and synchronize with it. The controller state includes the network topology view, network services, and application, and data synchronization.
- Failure detection – heartbeat messages are used to detect the failure of the primary controller.
- Recovery – this will include the election of a new primary controller using the election algorithm. The new primary controller informs about this change to the other controllers and the virtual IP change. Controller network services and application restoration.

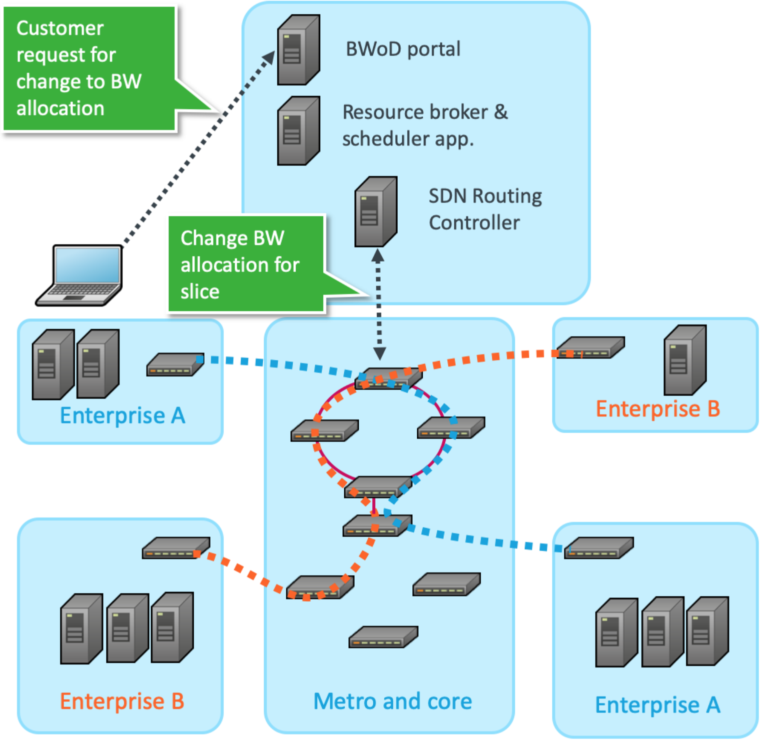

Customer request for change to BW allocation
BWoD portal
Resource broker & scheduler app.
SDN Routing Controller
Change BW allocation for slice
Enterprise A
Enterprise B
Enterprise B
Metro and core
Enterprise A

Figure 2.17: Enterprise Network data flow

2.5.5 SDN Controller types

Table 2.9 shows a single instance SDN controller, and table 2.10 shows multiple instance SDN controllers when looking at language and original author (Y.B.P. Gautam & Sato, 2020)

Table 2.9: Single Instance SDN controller

Name	Language	Original Author	Notes
OpenFlow Reference	C	Stanford/Nicira	Example only
NOX	Python, C++	Nicira	No longer actively developed
Beacon	Java	David Erickson	Runtime modular, web UI, framework, regression test framework.
Maestro	Java	Zheng Cai	
Trema	Ruby, C	NEC	Include an emulator, regression test framework.
Floodlight	Java	BigSwitch Networks	Fork of Beacon
POX	Python	Murphy McCauley	
(Mc)Nettle	Haskell	Andreas Voellmy	
RYU	Python	NTT	
MUL	C	Kucloud	

Table 2.10: Multiple Instance SDN Controller

Name	Language	Original Author	Notes
Onix	C ++	Nicira	Described in an academic paper
Big Network Controller	Java	BigSwitch Networks	Built on Floodlight
ProgrammableFlow		NEC	
OpenDaylight	Java	Consortium	Built on Beacon
ONOS	Java	Open Networking Lab	Built on Floodlight
Cisco XNC	Java		
OpenContrail	Java	Contrail/Juniper	
HP VAN	Java	HP	

2.5.5.1 Controller Descriptions

NOX

NOX is a piece of the *Software-Defined Network*ing ecosystem. Specifically, it's a platform for building network control applications. While SDN grew from several academic projects (SANE & Ethane), the first SDN technology to get real name recognition was OpenFlow. And NOX was initially developed at Nirica Networks side by side with OpenFlow – NOX was the first OpenFlow controller. Nirica donated NOX to the research community in 2008, and since then, it has been the basis for many and various projects in the early exploration of the SDN space.

To a developer, NOX provides a C ++ OpenFlow 1.0 API, provides fast and asynchronous Input/Output, designed for topology discovery, learning switch, and network-complete switch.

POX

Excellent for diving into SDN using Python on Windows, Mac OS, or Linux. It's mainly targeted at research and education and defines fundamental abstractions and techniques for controller design.

In a way, POX is NOX's younger sibling. At the core, it's a platform for rapid development and prototyping of network control software using Python, Meaning, at a fundamental level, it's one of the growing numbers of frameworks including NOX, Floodlight, and Trema.

POX is under active development. Its primary target is research, and many research projects are relatively short-lived. POX features include Pythonic OpenFlow interface POX performance graph, reusable sample components for path selection, topology discovery, supports the same GUI and visualization tools as NOX, performs well compared to NOX applications written in python.

Open Network Operating System (ONOS)

The Open Network Operating System is the first open-source SDN network operating system targeted specifically at the Service Provider and mission-critical networks. ONOS is purpose-built to provide the high availability, scale-out, and performance these networks demand. ONOS has also created useful Northbound abstractions and APIs to enable easier application development and southbound abstractions and interfaces to control OpenFlow-ready and legacy devices. The ONOS will bring carrier-grade features such as scalability availability and performance to the SDN plane, enable web style agility, help service migrate their existing networks to white boxes, and lower server provider CapEx and OpEx.

Open Daylight

OpenDaylight (ODL) is a highly available, modular, extensible, scalable, and multi-protocol controller infrastructure built for SDN deployments on modern heterogeneous multi-vendor networks. It provides a model-driven service abstraction platform that allows users to write applications quickly across various hardware and southbound protocols.

A model-driven service abstraction layer means the controller does not have to account for all equipment installed in the network, allowing it to manage a wide range of hardware and southbound protocols.

Floodlight

Known for enterprise-class, Apache-licensed, Java-based OpenFlow controller, supported by Big Switch Networks, supports a broad range of virtual and physical OpenFlow switches.

RYU

Component-based SDN framework supported by NTT, deployed in NTT cloud data centres, RYU supports OpenFlow, Netconf, OF-configs. They are used for the continuous testing environment with various OpenFlow v1.3 and v1.4 switches.

*Software-Defined Network*ing and *Smart Grid* play a vital role in the evolution of communication and electrical efficiency. The Chapter provides a view of the advantages and limitations of each section. Research shows the need for new technology to fill-full current and future requirements. Our approach is to design and explore the deployment options of integrating SDN and SG. The integration, however, will add further challenges to the market. In this, the correct protocol and configurations will serve to address those arising challenges.

From Chapter 2, research shows the scalability of the grid to cater to the communication needs, considering the expansion of the network. Detail analysis is required to ensure the safety and privacy of the customer's data and protection of the network components.

Chapter 2 expresses the value of Software Defined Networking, and in comparison to traditional networks and current protocols, an in-depth study on Smart Grid is provided, which flows into the study on Security applied to layers of the architecture. A short section on the integration of both environments is provided and Chapter 2 ends with research on different SDN controllers used for this thesis.

When looking into the digital world, with vast improvements and collaboration of vendors, the final output intends to produce an SDN Architecture that can be applied to an SDN Smart Home, SDN Automation Plant, and SDN Electrical Distribution. The following chapters will define new processes that can be used effectively with technology work environments—producing a final design built on the SDN model, which entails simulation and real-world components—finally testing the performance of different use cases.

CHAPTER 3

SYSTEM MODELING

The research focuses on integrating *Software-Defined Network* (SDN) and *Smart Grid* (SG) environments; the aim is to provide control of Smart Grid components from the SDN controller and add intelligence. Chapter 3 covers a proposed integrated architecture of SDN+SG, Capacity Forecast Model, OpenFlow modelling, Link Layer Discovery Protocol, proposed Flows, proposed OpenFlow Groups, proposed SDN within Automation Plant, proposed SDN within Smart Home, proposed SDN within Electrical Distribution system, and finally Migration strategies and methodologies.

SDN and SG's proposed architecture provides a new layer called the Network Exposure layer that fits between the communication network and *Smart Grid* applications. OpenFlow protocol was selected and used as the model for the design re-engineering of OpenFlow Groups and packet Flows. The Chapter provides a proposed architecture for each environment by bringing the model to real-world environments for SDN Smart Home, SDN Electrical Distribution system, and SDN Automation plant. To showcase the integration factor and intelligence from the SDN controller to the *Smart Grid* components.

A Layer 2 topology needs to model the node, port, trunking, VLAN, and various Layer 2 tree topologies. This Layer can become quite complicated in a large environment where VLANs are allowed only on specific ports, and Layer 2 network protocols decide how frames (data "packets") move between devices (i.e., which ports become blocked). Layer 1 topology where ports G1/1 carry VLANs 10 and all interfaces are configured as trunks. Layer 2 addresses are introduced and become an essential part of the Layer 2 inventory. A Layer 3 topology introduces IP addresses, routing policies/protocols, and the beginnings of service differentiation using Quality of Service (QoS) and Traffic Engineering techniques shown in Figure 3.1. SDN technology requires layer two and layer three nodes to be SDN-enabled, providing application slices based on requirements.

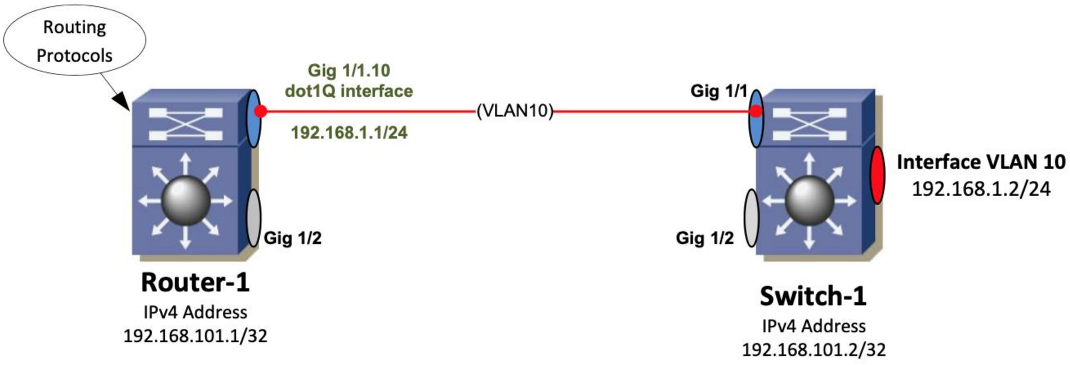


Figure 3.1: An example of Layer 3

3.1 Capacity Forecast Model

Capacity Management is a set of work processes associated with the provisioning and managing infrastructure resources used to support business processes cost-effectively. These work processes include monitoring, reporting, tuning, planning, and predictive modelling.

Network models help to define what needs to be managed. The three main management areas are (Bastos, 2019):

- Traffic / Service Performance management,
- Capacity growth management, and
- Network planning.

Network models are representations of systems, processes, and interactions and can exist in various levels of abstraction. Ideally, the network model is accurate enough to answer queries considering individual elements and up to the network as a whole. However, a model is only as good as the information it contains and how well it represents what it is trying to model.

In terms of computer networks, there are two types of network models:

- a physical topology, and
- a logical topology.

Application models consist of different software used for devices to communicate and perform an action; Application Programming Interface (API) is used to communicate between software. The application model can be built to overlay the physical and logical topology models.

Capacity growth management observes the network over a daily to monthly period and asks: “how is the network growing, and how do we ensure sufficient capacity? This allows alarms to be raised as well as identify areas of concern.

Resource capacity management is generalized as follows:

- A resource is a generic term and can be a link, class of service, disk, port module, CPU, interface density or availability, memory, etc.
- Considers peak-hour rates and maximum daily utilization for resources without a daily profile.
- Build sufficient capacity for single resource failures.
- Consider upgrade delays when ordering additional capacity.

The network must be designed with redundancy proposed in Figure 3.2, showing a forecast model recommendation. This means:

- The network must be configured for single link failure, inter-regional, regional, and site-local.
- Any link must carry all the traffic of its link pair in a failover scenario.
- Links should therefore never carry more than 50% of their link capacity under non-failover scenarios.
- This means that the Critical Upgrade Threshold for any link is 50% of link capacity.
- According to my current planning scope, the Upgrade Threshold is dependent on the Upgrade Delay but is recommended at a maximum of 40% of link capacity.

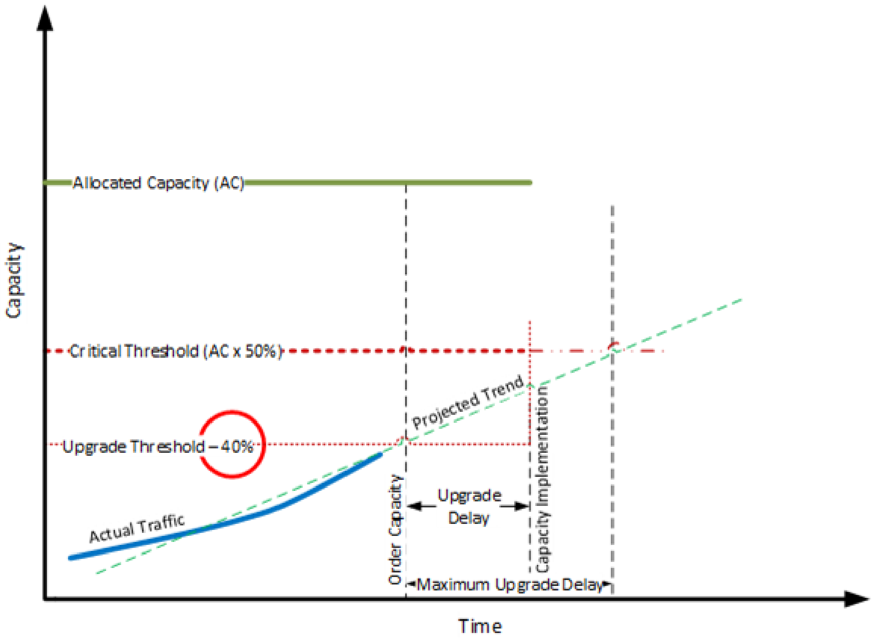


Figure 3.2: Capacity Forecast Model

The Capacity Forecast model shows the planning required to avoid link congestion in a network. To accommodate failures, the model becomes critical with the growth of traffic and the influence of unplanned events. The model currently used is a manual analysis; the expectation is to have the *Software-Defined Network*(SDN) controller perform the calculations of the Capacity Forecast model on a real-time basis.

3.2 Architecture of *Software-Defined Network* and *Smart Grid*

SDN technology plays a role in several sectors; the research will explore a Smart Home, Automation Plant, and Electrical Distribution system in **3.4.1**, **3.4.2**, and **3.4.3**. An integration model will be presented for each environment to showcase the architecture and algorithms on the SDN controller.

The aim is to design the network on centralized control and implement the intelligence in each environment. A new architecture will be presented for each environment to showcase the SDN concept's advantages. The purpose of SDN in transport networks provides the following from Sahoo, Sahoo & Panda (2015):

- Cost savings through virtualization,
- Accelerate the introduction of new services across the whole network,

- Automate the workflow processes to reduce operational costs and increase scalability,
- Optimize resource consumption.

When the focus is given to a Service Provider’s communication network, the packet world and transport world can be expanded from Argibay-Losada et al. (2015) in Table 3.1—showing different scaling of network design.

Table 3.1: Packet World and Transport World

Packet World	Transport World
Connectionless	Connection-oriented
Enterprise origins	Service Provider origins
Dynamic flows	Static pipes
Innate control plane	NMS + Cross-connect paradigm
Numerous distributed Control plane solutions	Nascent CP (GMPLS)
Monolithic, closed systems	Open, programmable systems

My research aims to integrate the communication network into a *Smart Grid* network, allowing the SDN controller to establish centralized control of both environments. Figure 3.3 shows the integration, where each environment belongs to a Pod. The implementation caters in Figure 3.3 for an SDN Electrical Distribution system, SDN Automation network, and SDN Smart home,

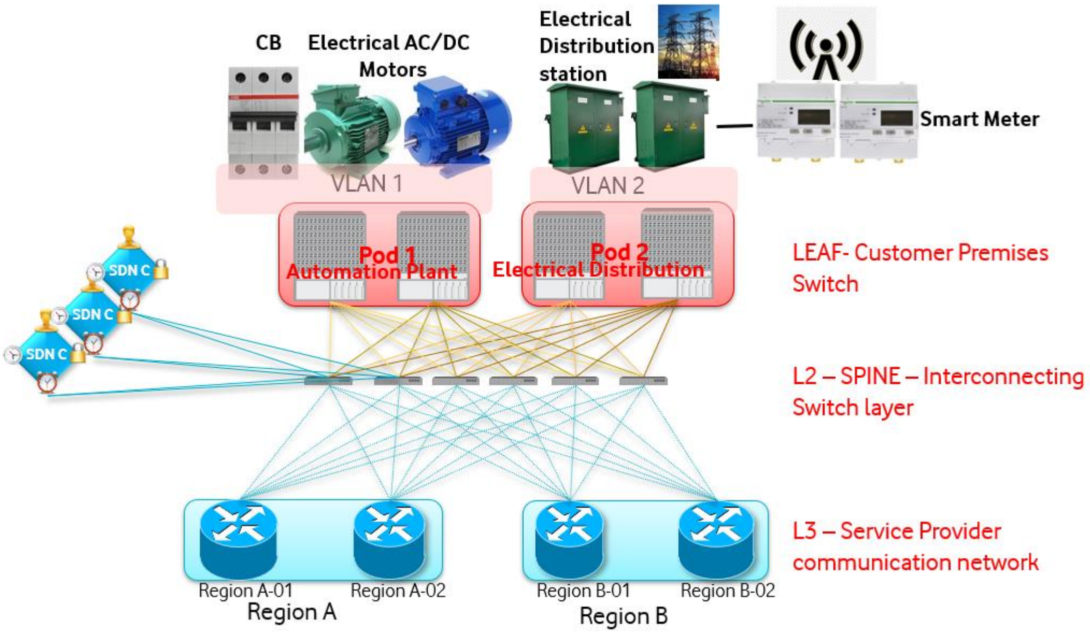

CB
Electrical AC/DC Motors
Electrical Distribution station
Smart Meter
VLAN 1
VLAN 2
Pod 1 Automation Plant
Pod 2 Electrical Distribution
SDN C
SDN C
SDN C
LEAF- Customer Premises Switch
L2 – SPINE – Interconnecting Switch layer
L3 – Service Provider communication network
Region A-01
Region A-02
Region A
Region B-01
Region B-02
Region B

Figure 3.3: Architecture of *Software-Defined Network*(SDN) and *Smart Grid*(SG) component integration

The architecture described in Figure 3.3 integrates SDN and SG components. To bring understanding to the architecture, the architecture is explained:

- The communication network belongs to a Service Provider. The solution is not to re-design a new network but rather to add intelligence and control using the SDN concept; the communication network is labelled ‘Region A’ and ‘Region B.’ This can scale to the core network for inter-regional connectivity.
- The *Smart Grid* contains AC/DC components; in Figure 3.3, ‘Automation Plant’ and ‘Electrical Distribution’ are placed in pods 1 and 2, respectively. The Pod is off-the-shelf equipment, known as white boxes, with software running on it; its purpose is to keep the intelligence within the SDN controller (policies and commands). The Pod will reside on the customer’s premises.
- The entire system is designed with redundancy. The pure reason is to provide high availability to the user, considering real-world outages/faults—loss of connectivity equates to revenue loss. The idea around the design is to build a platform that fits in the real world, in the country of load shedding.

- The interconnecting switches will reside in the layer two environments of the Service Provider. Based on the protocol set, for each service, in terms of Automation Plant or Electrical Distribution. A VLAN ID will be set, so all data (user and control) will not be accessed by the other, e.g., Automation Plant data will not be able to see Electrical Distribution data.
- A tapping aggregation-probing system that will provide analytics to the user and vendor. The Smart meter will form a minor role in the research, and connectivity will be wireless to a 4G/5G network. Smart meters are considered virtual Smart Meters.

The SDN controller contains policies and pushes policies to the *Smart Grid* environment. A policy resembles the recipe in baking a cake; there is a list of ingredients and steps to follow. The list of ingredients is the *Smart Grid* component's parameters, and the steps are the systematic operation. The policies applied to *Smart Grid* can be replicated for the need to scale the SG environment and not a rework.

Figure 3.4 shows the Network Exposure layer that defines the mediation between the communication network and the *Smart Grid* network. The Exposure layer routes the path for the intercommunications, control, and management via the API catalogue, which consists of an open-source protocol to communicate to different platforms on one standard Exposure layer.

The communication network is built on *Design, Deployment, and Provisioning*. It is built on a virtualization platform that shares CPU, Storage, and NIC resources. The application layer contains application slices, each for its unique ability to serve on latency, mobility, and automation. The Ultra-Reliable Low Latency application slice can serve the Electrical Distribution system. The enhanced Mobile Broadband will serve a Smart Home, and massive Machine Type Communication serves Automation Plant.

The Exposure Layer is built on algorithms to separate different application slices on the standard layer, and each environment will be configured to each virtual LAN proposed:

SDN-ED: VLAN 1

SDN-SH: VLAN 2

SDN-AP: VLAN 3

To cater for each client, a sub-interface is allocated, shown:

SDN-ED: VLAN 1.1000

SDN-SH: VLAN 2.1000

SDN-AP: VLAN 3.1000

The research proposes the Network Exposure layer to model the integration of *Software-Defined Network* and *Smart Grid*. Which can be viewed as a highway with dedicated lines with Virtual Local Area Network (VLAN) of sub-interfaces. Each VLAN is unique and contains control and user data within each. The Network Exposure layer becomes the standardized interface for each *Smart Grid* environment to communicate to the network. The Electrical Distribution system contains proprietary software from a Smart Home; the Network exposure layer allows different proprietary software to communicate on a standardized interface, allowing control and bringing intelligence from the communication network.

The VLAN provides a logical network for each *Smart Grid* environment that shares the network's resources; each VLAN group will be inserted a range from 1000 to 1999, depending on the planned scalability of the network.

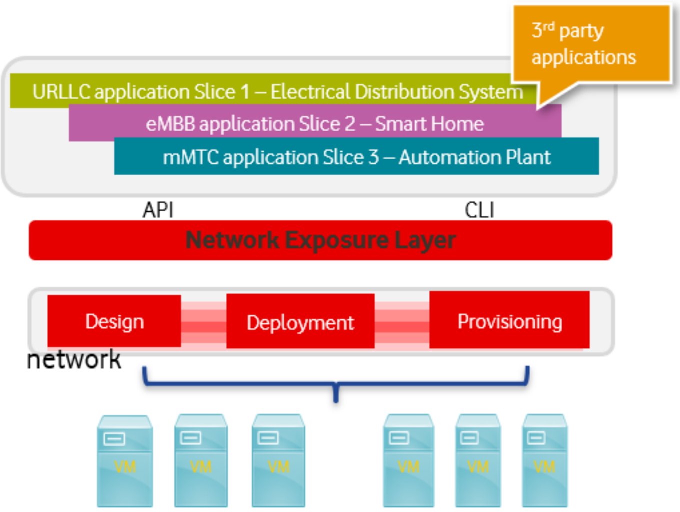

3rd party applications
URLLC application Slice 1 – Electrical Distribution System
eMBB application Slice 2 – Smart Home
mMTC application Slice 3 – Automation Plant
API
CLI
Network Exposure Layer
Design
Deployment
Provisioning
network
VM
VM
VM
VM
VM
VM

Figure 3.4: Network Exposure layer

For the *Smart Grid* to communicate to the Service Provider's network, a detailed configuration system is set up to create each application slice shown in Figure 3.5. The tenant labeled SDN, and VRF 01 is set to carry *Smart Grid* traffic. Each Application Slice will serve per service, like SDN Electrical Distribution or SDN Smart Home is an application slice, which is unique to each other. Finally, the VLAN shown is also unique to each other and each End Point Group contains the VLANs for each customer's application slice. The Service Provider's network sits on the shared virtual platform.

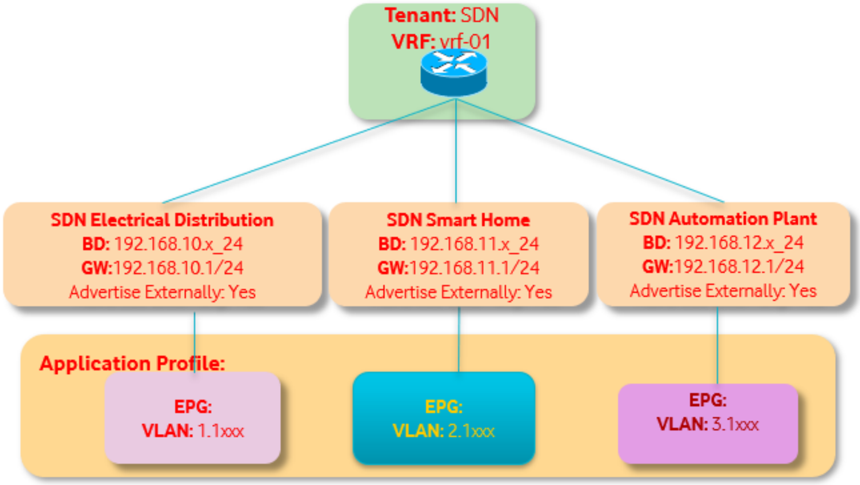


Figure 3.5: VRF/VLAN per Application Slice

SDN Electrical Distribution control and user plane traffic cannot be seen by SDN Smart Home, likewise SDN Automation Plant. Within the Application Profile, each *Smart Grid* environment is configured within a different VLAN; for example, Electrical Distribution is within VLAN 1.1, Smart Home is within VLAN 2.1, and Automation Plant is within VLAN 3.1.

3.3 OpenFlow Modeling

OpenFlow is a protocol that enables the programmability of the forwarding plane across the network as a whole. OpenFlow is leveraged at the Southbound Interface between the SDN controller and OpenFlow switch. OpenFlow attempts to abstract the implementation details of a network; there are many ways to view OpenFlow:

- As a protocol,

- As an instruction set,
- Or as an architecture.

It can combine all; OpenFlow defines an interface between an SDN controller and the switch. The OpenFlow component in the SDN controller is responsible for communicating instructions to the switch across the secure channel.

An OpenFlow switch interface defines the following:

- State – The network packet condition, how the state can be checked, and a condition matched based on the *Smart Grid* operation.
- Control Interface – Given a state, how can the switch forward or modify packets. Containing control plane data to execute actions.
- Behaviour – What software can program the switch and report to the controller to create the desired results from the state and control interfaces.

3.3.1 OpenFlow Secure Channel

Openflow uses Secure Channels between the communication of the controller and switch; the research selected OpenFlow to configure on the testbed and for the design model.

The Secure Channel is used for:

- Processing instructions and configuration flow between the controller and the switch, notifications from the switch to the controller, and packets from processing to or from the controller.
- The connections operate over Transmission Control Protocol (TCP), and both switch and controller listen on port 6653. Initially, OpenFlow used port 6633.
- The connection is encrypted using over Transport Layer Security (TLS). It can operate clearly, but TLS is recommended and is considered more secure than Secure Socket Layer. The research also serves to build security into the SDN and SG architecture.

- A switch can connect to more than one controller and have multiple connections between the switch and the controller. Various connections can offer redundancy and load balancing and improve the overall connection performance between the switch and the controller. When there are multiple connections, one connection is the main, and the others are known as auxiliary connections. If the primary connection is down, the auxiliary connections are also removed. The architecture in Figure 2 shows the multiple connections to cater for a redundant and load balance network.
- Auxiliary connections must use the same source IP address as the primary connection. Depending on the switch configurations, they can still use a different transport layer, such as TLS, TCP, DTLS, or UDP. Note, OpenFlow does not provide ordering or delivery guarantees on connections using UDP or DTLS. If messages must be processed in sequence, they must be sent over the same connection.

The controller uses set OpenFlow messages to manage the pod’s switch. Manipulating messages, the controller not only can add, modify or delete flow table entries, but it can also query the switch for features and statistics, configure the switch, set switch port properties, and send packets out a specified switch port.

Asynchronous Messages are sent from the switch to the controller. These messages can be packet or packet header that does not match any flow entry and therefore needs to be processed at the controller, a notification of a change in a flow state, or an error message.

Table 3.2 shows OpenFlow message types and examples.

Table 3.2: OpenFlow message types

Message Types	**Description**	**Examples**
Controller to Switch	The controller initiates it.	Read State Modify State Packet Out Configuration Barrier Features Role Request
Asynchronous	Initiated by the OpenFlow switch without solicitation from the customer.	Error Packet In Flow Removed Port Status
Symmetric	It was initiated in either direction without solicitation.	Hello Echo Experimenter

Each flow entry contains a set of instructions that are executed when a packet matches the entry. The actions will collaborate with *Smart Grid* component actions. The below shows the instructions and action sets:

- Apply Actions
- Clear Actions
- Write-Metadata
- Stat Trigger
- Required Instructions
- Write Action
- Goto Table

Actions include:

- Copy TTL/Decrement TTL

- Pop/Push Tags (MPLS/PBB/VLAN)
- Set Fields
- QoS Actions
- Output

The switch may support arbitrary action execution order through the list of actions of the Apply Action instruction. The following takes place:

- Copy TTL inwards: apply copy TTL inward actions to the packet,
- Pop: apply all tag pop actions to the packet,
- Push-MPLS: apply MPLS tag push action to the packet,
- Push-PBB: apply PBB tag push action to the packet,
- Push-VLAN: apply VLAN tag push action to the packet,
- Copy TTL outwards: apply decrement TTL action to the packet,
- Set: apply all set-field actions to the packet,
- QoS: apply all QoS actions, such as meter and set the queue to the packet,
- Group: if a group action is specified, apply the actions of the relevant group bucket(s) in the order specified by the list,
- Output: if no group action is specified, forward the packet to the port specified by the output action.

The output action in the action set is executed last.

3.3.2 Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is an open and extendable part of the Internet Protocol suite used in IEEE 802 to advertise its identity and abilities and other devices connected within the same network. LLDP is used primarily in wired Ethernet-connected devices to facilitate network resources management and simplify networking tasks for administrators in a multi-vendor network. In addition, LLDP plays a significant role within SDN. The protocol's multi-vendor support enables network discovery of devices/*Smart Grid* components and management tools like Simple Network Management Protocol

(SNMP) in a network made up of devices from different manufacturers. LLDP makes it unnecessary to use a high number of proprietary protocols to support a multi-vendor network.

On an interval schedule, an LLDP device sends its information in Ethernet frames. A frame starts with the required Type-Length-Value TLVs of Chassis ID, Port ID, and Time-To-Live (TTL). Next, the frame contains the device's destination MAC address, a multicast address that is not forwarded outside a network, assuming 802.1D compliance.

LLDP is also known as Station and Media and Access Control Connectivity Discovery, as specified in IEEE 802.1AB. Similar proprietary protocols include Cisco Discovery Protocol, Extreme Discovery Protocol, Foundry Discovery Protocol, Microsoft Link-Layer Topology Discovery, and Nortel Discovery Protocol.

LLDP information is sent from each of its interfaces at a fixed interval in the form of an Ethernet frame. Each frame contains one LLDP Data Unit (LLDPDU) sequence of Type-Length-Value (TLV) structures shown in Figure 3.6.

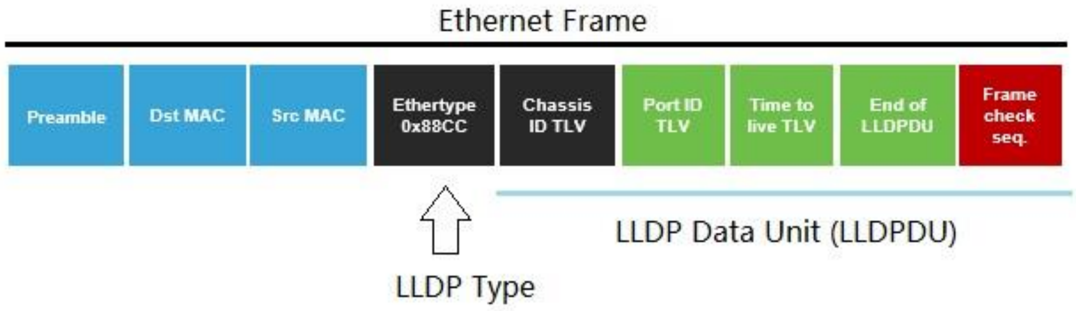


Figure 3.6: Ethernet Frame – LLDP Data Unit

The ethernet frame used in LLDP has its destination MAC address typically set to a particular multicast address that 802.1 D compliant bridges do not forward. Other multicast and unicast destination addresses are permitted. Each LLDP frame starts with the following mandatory TLVs: Chassis ID, Port ID, and Time-to-Live. Any number of optional TLVs follows the mandatory TLVs. The frame ends with a particular TLV, named end of LLDPDU, in which both the type and length fields are zero.

The next section shows what makes up a flow table and provides an example. In conjunction, packet-in/packet-out OpenFlow pipeline processing figures are presented to show the matching ability. And the summary of the section provides a detailed framework on OpenFlow Groups.

3.3.3 Flow table

Each flow table may not support every match field, instruction, action, or set field defined by the specification, and different switch flow tables may not support the same subnet. The table features request enables the controller to discover what each table supports.

Its match fields and priority identify a flow table entry: the match fields and priority are taken together to identify a specific flow table's unique flow entry. The flow entry that wildcards all fields and when priority is equal to zero is known as the table-miss flow entry.

To make compatible the *Smart Grid* components.

Each flow entry contains:

- Match fields: to match against packets. These consist of the ingress port and packet headers and other pipeline fields such as metadata.
- Priority: matching precedence of the flow entry according to speed or *Smart Grid* application slice.
- Counters: updated when packets are matched.
- Instructions: to modify the action set or pipeline processing of the *Smart Grid*.
- Timeouts: the switch expires the maximum amount of time or idle time before.
- Cookie: opaque data value chosen by the controller. The controller may use it to filter flow entries affected by flow modification and flow deletion requests.
- Flags: flags alter how flow entries are managed; for example, the flag OFPFF_SEND_FLOW_REM triggers flow removed messages for that flow entry.

When a packet is presented to a table for matching, the input consists of the packet, the ingress port's identity, the associated metadata value, and the associated action set. A flow table may include a table-miss flow entry, which renders all Match Fields wildcards with the lowest priority (priority 0). The following shows the processing steps:

- Find the highest-priority matching flow entry. If there is no match on any entry and there is no table-miss entry, the packet is dropped. If there is a match only on a table-miss entry, then that entry specifies one of three actions.
- Send a packet to the SDN controller. This action will enable the controller to define a new flow for this and similar packets or decide to drop it.
- Direct packets to another flow table further down the pipeline.
- Drop the packet.

The matching flow could be based on the following examples:

- Match on a sector, like SDN_SMART_HOME, or

  SDN_ELECTRICAL_DISTRIBUTION, or SDN_AUTOMATION_PLANT.
- Match on incoming traffic or outgoing traffic.
- Match on fraud detection techniques.
- Match on specific groups.
- Match on-site, region, pod, or device.

Each flow entry has an idle timeout and a hard timeout associated with it, configured through the OpenFlow controller. The idle timeout is the number of seconds after which a flow entry is removed from the table and the hardware provided because no packets match it. The hard timeout is the number of seconds after which the flow entry is removed from the flow table and *Smart Grid* hardware, whether or not packets match it.

The flow tables of an OpenFlow switch are sequentially numbered, starting at 0 from ONF Switch Specification 1.5; pipeline processing happens in two stages, ingress processing and egress progressing. The first egress indicates the separation of the two stages; all tables with a number lower than the first egress table can be used as an ingress table.

Suppose the outcome of the ingress processing is to forward the *Smart Grid* action to an output port. The OpenFlow switch may perform egress processing in the context of that output port. Egress processing is optional; a switch may not support any egress tables or may not be configured to use them. If no valid egress table is configured as the first egress table, the packet must be processed by the output port, and in most cases, the packet is forwarded out of the switch. If a valid egress table is configured as the first egress, the packet must be matched against flow entries of that flow table, and other egress flow tables may be used depending on the outcomes of the match in that flow table.

If a packet does not match a flow table, this is a table miss. The behaviour on a table miss depends on the table configuration. The instructions included in the table-miss flow entry in the flow table can flexibly specify how to process unmatched packets; useful options include dropping them, passing them to another table, or sending them to the SDN controller over the control channel packet-in messages. The following section presents OpenFlow Groups.

3.3.4 OpenFlow Groups

An OpenFlow group is an abstraction that facilitates more complex and specialized packet operations that cannot easily be performed through a flow table entry. Each group receives packets as input and performs any OpenFlow actions on these packets. A group cannot perform any OpenFlow instructions, so it cannot send packets to other flow tables or meters. Furthermore, packets are expected to be matched appropriately before entry to a group, as groups do not support matching on packets – groups are merely mechanisms to perform advanced actions or sets of activities; a bucket defines the parameters and actions shown in Figure 3.7.

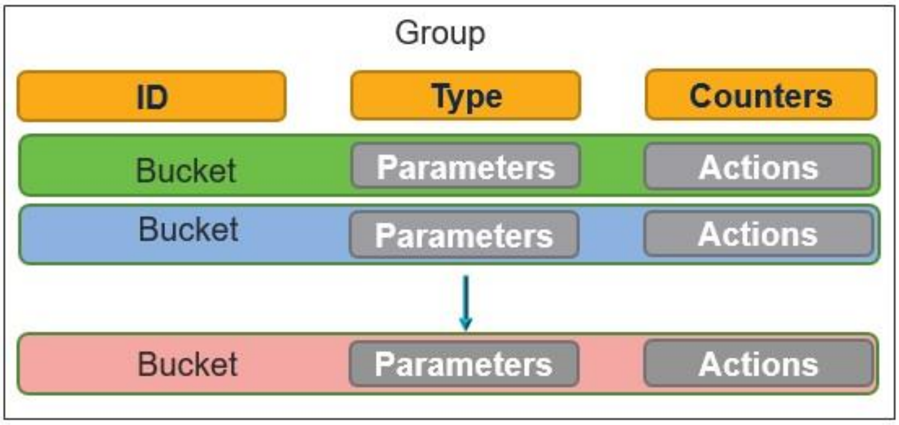


Figure 3.7: OpenFlow Group – Parameters and Actions

A bucket can serve the *Smart Grid* domains, like the automation plant, the parameters will define the settings for a PLC system, and the actions will define what outputs will occur.

3.3.4.1 OpenFlow Group Tables – All

The ALL group approach will take any packet received as input and duplicate it to be operated on independently by each bucket in the bucket list, which can be effective for Automation Plant or Electrical Distribution systems. In this way, an ALL group can replicate and then operate on separate copies of the packet defined by each bucket's actions. Thus, different and distinct actions can be in each bucket, allowing various operations to be performed on additional packet copies in Figure 3.8. The sample of deploying a motor torque of 10 policy is sent to each configured region, in KZN, WES, LIM and EC.

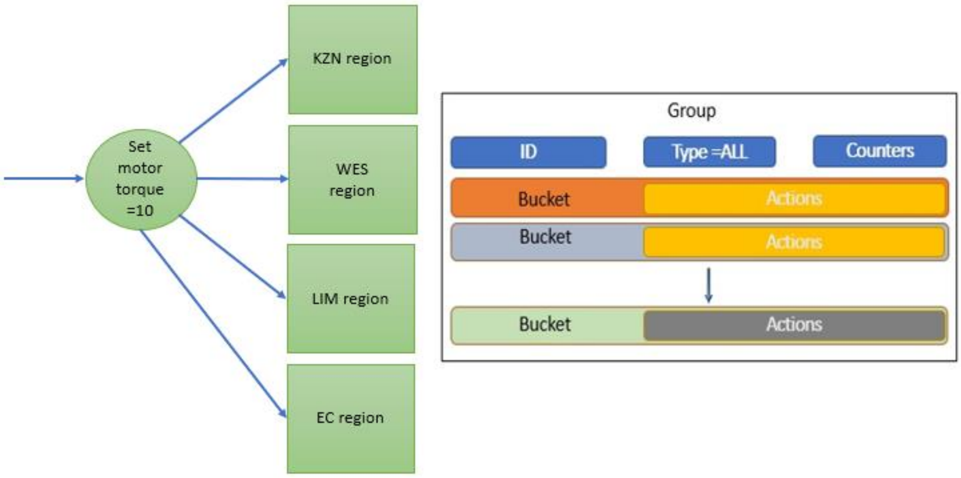


Figure 3.8: OpenFlow Group Table- All

3.3.4.2 OpenFlow Group Tables – Indirect

The Indirect group approach can be challenging to comprehend as a ‘group’ since it contains only a single bucket where all packets received by the group are sent to this lone bucket. In other words, the Indirect group does not include a list of buckets but a single bucket instead.

The purpose of the Indirect group is to encapsulate a common set of actions used by many flows. For example, suppose flow A, B, and C match different packet headers but have a common set or subset of activities. In that case, these flows can send a packet to a single Indirect group instead of duplicating the list of common actions for each flow. The Indirect group is used to simplify an OpenFlow deployment and reduce the memory footprint of a set of similar flows in Figure 3.9. Each flow will define Motor parameters, Temperature settings and Valve adjustments separately for a particular region or location.

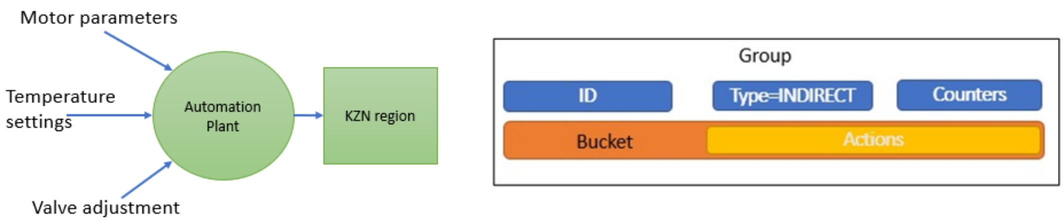


Figure 3.9: OpenFlow Group Tables - Indirect

3.3.4.3 OpenFlow Group Tables – Select

The Select group approach is primarily designed for load balancing. Each bucket in a Select group has an assigned weight preferred for the Electrical Distribution system, and each packet that enters the group is sent to a single bucket. The bucket selection algorithm is undefined and is dependent on the switch's implementation; however, weighted is the most straightforward choice of packet distribution. The weight of a bucket is provided as a particular parameter to each bucket. Each bucket in a Select group is still a list of actions, so any actions supported by OpenFlow can be used in each bucket, and like ALL groups, the bucket need not be uniform in Figure 3.10. The weight 100, 200, 300 or 400 needs to be matched for the system to enable load balancing and thus prevent over surge.

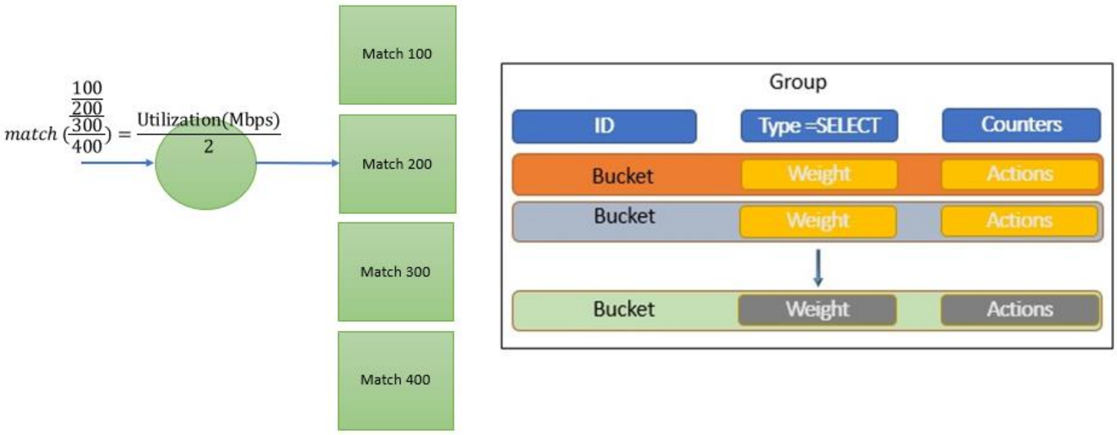


Figure 3.10: OpenFlow Group Tables - Select

3.3.4.4 OpenFlow Group Tables – Fast Failover

The Fast failover group approach has a list of buckets. In addition to the list of actions, each bucket has a watch port and/or watch the group as a particular parameter. The watch port/group will monitor the liveness or up/down status of the indicated port/group. If the liveness is deemed to be down, then the bucket will not be used. If the liveness is determined to be up, the bucket can be used. Only one bucket can be used at a time, and the bucket in use will not be changed unless the liveness of the currently used bucket’s watch port/group transitions from up to down. When such an event occurs, the Fast failover group will quickly select the next bucket in the bucket list with a watch port/group that is up.

There is no guarantee on the transition time to select a new bucket when a failure occurs. The transition time is dependent on search time to find a watch port/group that is up and on the switch implementation. However, the motivation behind using a Fast failover group is that it is almost guaranteed to be quicker than consulting the control plane to handle the port down the event and inserting a new flow or set of flows. Fast failover group will serve Electrical Distribution systems and cater to failed sites. With Fast failover groups, link failure detection and recovery occur entirely on the data plane in Figure 3.11. The system will allow a high (UP) or low (DOWN) bucket approach, to enable an always-On network, and building intelligences into the system.

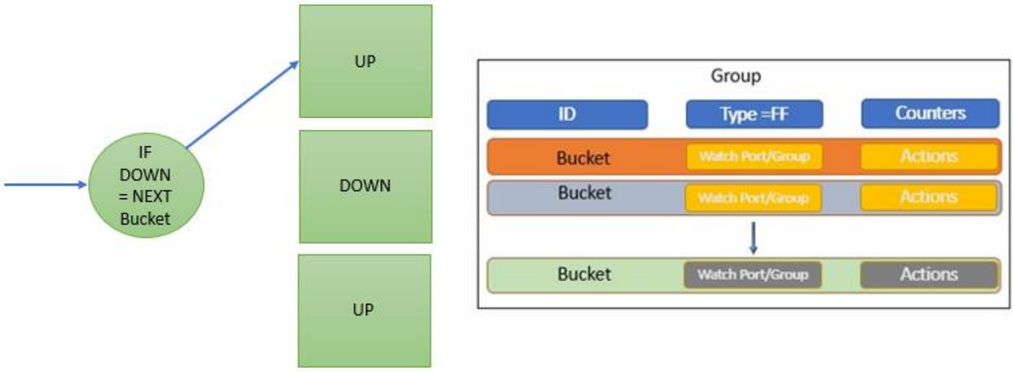


Figure 3.11: OpenFlow Group Tables – Fast Failover

3.4.1 SDN within an Automation Plant

Traditional large and flat layer 2 data centers have scalability and provisioning limitations. However, due to high server virtualization, the increasing requirement to have direct layer two amongst geographically diverse locations can solve problems. One of the basic premises behind server virtualization is resource allocation – which needs to distribute Virtual Machines across many physically different machines.

Automation plants are widely spread in regions and continuously require new design specifications and parameters. To either improve the existing products or create new products. The process can be exhausting and complicated; the improved solution proposes open-source software tools, not vendor proprietary software. An SDN Automation Plant presented can allow operators to deliver Infrastructure as a Service automate the delivery from software control.

The Software-Defined automation plant approach forces IT organizations to adapt. Architecting Software-Defined environments require rethinking many IT processes – including automation, metering, and billing – executing service delivery, service activation, and product assurance.

To make this possible, the incorporation of virtual switches can be programmatically controlled by an external SDN controller via a standardized interface like OpenFlow. Furthermore, multiple overlay networks can be created between these switches. This will overcome the plant's location and gain control via remote and fast provisioning.

The creation of a virtualization layer sees all the traffic entering and leaving the network. All the services are implemented at this virtualization layer and abstract the physical network and its topology. This creates a broader and scalable architecture. The vSwitch-based edge becomes the programmatically controlled virtualization layer connected to the SDN controller. This will allow communication between the tenants and the controllers.

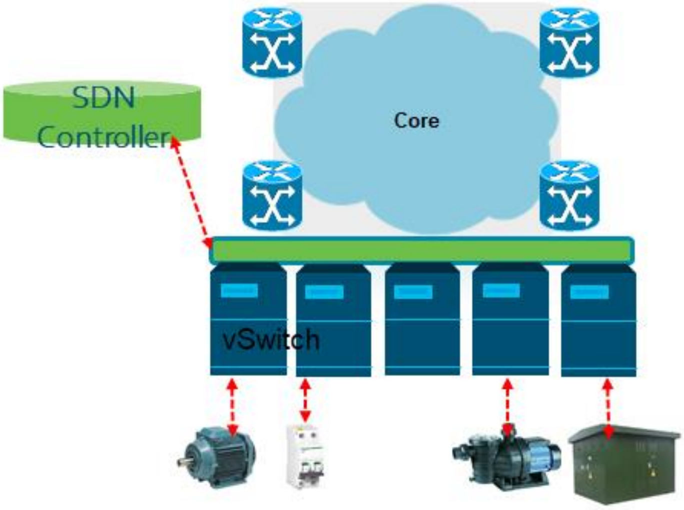


Figure 3.12: SDN Automation Plant

The vSwitch is at the Edge of a network in Figure 3.12, and the programmatically controlled virtualization layer, the Core, is the fabric providing simple IP connectivity.

The management of devices ensures the operation of a device on safety regulation. The SDN controller manages the devices connected to the network. The process of automation is required, allowing each device to operate according to its datasheet specification. Each device is labelled a serial number. Each serial device is unique; it becomes a fair task to record its data specification and attach it to a tagged serial number. The Service provider should do this process. This way, the SDN controller understands the device's operating ability.

For example, a data sheet for motor ABB 3-phase induction motor, serial code – 3GBA 182 410-ADCIN, with a maximum torque of 3.5 T_{max}/T_N. The maximum torque will be recorded within the SDN controller, and if reached, it will engage a limit switch to cut off the power. Specifications can be amended according to the product owner.

In terms of the motor's operation as a service or relay as a service, the execution of service will be required to be inserted into a traffic path to be identified and direct particular traffic types across several network elements. Data integrity is a priority ensuring the packet is delivered to the correct destination to be executed. The data packet is treated like any other packet, with firewall requirements, Access control List, Authentication, Authorization, Accounting, and proxy configuration. The product owner can set these features and protect the assets and revenue of the domain in Figure 3.13.

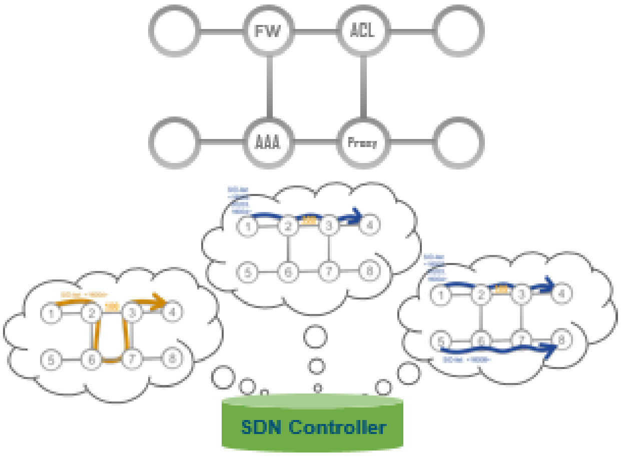


Figure 3.13: Security enablement

SDN Automation Plant plays a significant role within the *Smart Grid* environment; the management and control factor of the Automation Plant serve important and to be managed remotely. The need for security techniques is built into the platform to work remotely of the Automation Plant. The factor of automation scales onto the buckets of the OpenFlow Group Tables, enabling configuration/parameters to be distributed across VLAN services. The creation of these services can be scaled out and built from templates.

The management of the Automation Plant is performed from an open-source GUI to communicate to all OpenFlow enabled virtual/switches. The GUI is hosted on a web page, showing each component's live topology and status. Another area is the naming convention of each Automation Plant component; it preferred to have a standard description shown below:

VLAN () – Service/application slice () – component ()

VLAN 1 – Automation Plant 01 – Motor 01

Within the configuration of each service, we have a sub-interface showing the port number and further configuration details. The idea is to ensure that the SDN controller has a complete populated list of components within its routing tables. From the description, the SDN controller can track and record each action performed on the system, using a unique description that also includes a unique IP address.

SDN Automation Plant adds intelligence and manages control from a virtual station; amongst the virtual platform is the requirement for the site owner to scale and forecast new growth areas.

3.4.2 SDN within a Smart Home

A Smart Home is defined by the house's critical components; the components are a virtual smart meter and an integrated wireless communication unit to measure and perform hard stop features.

The smart meter is a platform that controls and monitors the home; its intelligence comes from the SDN controller to manage and control the home network. The smart meter will manage the household appliances and meters, from the kitchen appliances to each room and outdoor supply. The customer will have complete control to switch on or off the supply of electrical components. For example, the owner will control the timer for the pool pump and switch the geyser when not requiring hot water. The control features are relatively basic but efficient to protect a family and help the owner manage their usage. The monitor feature will provide stats linked to a mobile application, purely on use, fraud detection, and theft.

An extensive number of owners struggle to manage the usage payment with most appliances requiring electrical power. A basic meter cannot justify incorrect usage, primarily when a high bill shock is issued to an owner after months. For this, a system needs to store data and be available at a rate of 99,999%. The need for control is becoming a greater demand in larger cities. The smart meter will serve as the guard at the main gate. The smart meter, or virtual smart meter, will be downloaded on a mobile application regarding its flexibility. And have features to set a threshold alarm on usage, providing live notification to the owner.

Fraud or loss of revenue is becoming a more pressing topic. The smart meter will have features to monitor and prompt when usage patterns exceed the reference levels. In addition, algorithms will channel when fraud is suspected and notify a security team. The higher the level of fraud, the more significant revenue loss, Service Providers strive to secure their network from fraudulent activities to protect the customer and protect their network.

Two different network planning elements are noted. The first task is to build a redundant supply to the customer, and the second is to identify the location and cause of the theft. The last aims are to detect theft, preserve the customer's supply, and mitigate theft activities. To promote always On network for network efficiency.

The Smart Home consists of a virtual smart meter, fridge, router, pool pump, and electrical machines shown in Figure 3.14. The fundamental scope is to have the ability to control and manage your network from the SDN controller. The role of the Network Exposure layer will explore the ability to allow different proprietary devices to communicate to the OpenFlow switches. For research, the OpenFlow protocol will be configured to serve as the network exposure layer. Each device within the Smart Home needs to be OpenFlow enabled.

The SDN Smart Home proposes the SDN controller to take routing information and determine the optimal OpenFlow path based on the topology created from the information received from all the OpenFlow-enabled devices on the network. This path would start from the location where the packet enters to the location where it exits the network. Both directions would determine the optimal path for the call packets to flow through the network.

This may require directing packets out links different from the traditional pipeline would have selected. This flow matching information, along with the actions necessary, is pushed to each OpenFlow-enabled switch. The OpenFlow entry ages out when the call ends due to inactivity, freeing up resources for other actions or high-priority application flows.

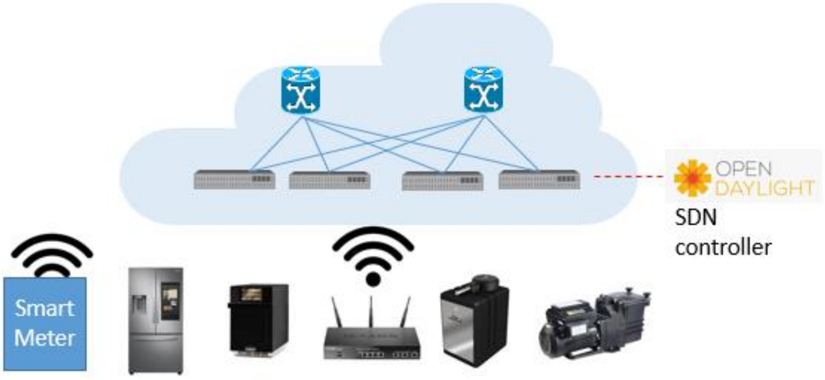


Figure 3.14: SDN Smart Home

The SDN controller can identify the newly connected devices, check what's the new device is and what it needs to access, and push the flow settings back to the SDN-enabled switches. Initially, the end-user device gets the IP address from DHCP under the SDN controller's supervision; the SDN controller knows that the new device is connected to the network.

Considering a Smart appliance connected to the home router. SDN controller has a collection of multiple SDN applications. Once the new device is connected, the SDN controller runs the Network Access Control Application.

The Network Access Control (NAC) Application in Figure 3.15 publishes the base flows from the newly connected user to the network. The flows include accessing the Active directory for login, DNS, and DHCP. The NAC Application can perform multiple checks by calling the database to identify the user's device, performing checks against the Antivirus server, and using Active Directory to ensure the user's login to the domain. Once the NAC Application receives the valid responses from sources, it pushes the additional flows from the specific user to the whole network. The flows are based on the user's access and placed in different switches to allow controlled access.

The process is transparent to the user, the user logs in to their PC, and the NAC application checks and enforces the policies to execute *Smart Grid* Smart Home actions.

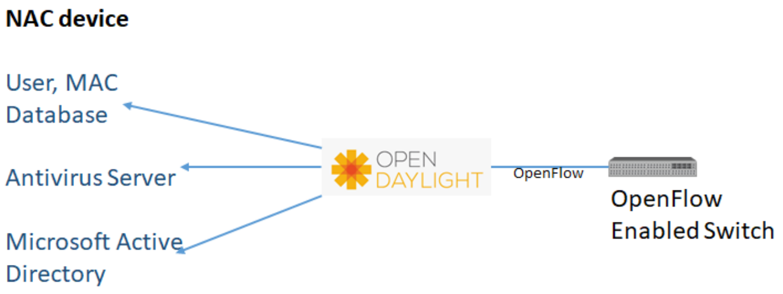


Figure 3.15: Network Access Control Application

Smart Home networks require logically partitioned networks, each with its policy. Currently, solutions such as MPLS from Porwal, Yadav & Charhate (2008) or BRF-Lite create logical network slices over a single physical network. Deploying and managing these technologies is static, time-consuming, and very cumbersome. SDN/OpenFlow-enabled switches allow logical networks to be created on-demand in a matter of minutes and instead of weeks. The logical network will be configured with a VLAN, which will be unique on the network and belong to a Smart Home. VLAN 1 cannot see the actions of VLAN 2. These switches can enforce flexible policies to control and limit interaction among logical and *Smart Grid* networks.

The benefits of SDN in a Smart Home,

- Operational savings: SDN can lower operating expenses with simplified management and better infrastructure utilization. With cost-competitive options for hardware, it can also reduce expenses on proprietary Smart Home brands.
- Higher performance: SDN can support dynamic allocation of bandwidth and resources as needed by variable user application loads.
- Improved Uptime: SDN reduces configurations and deployment errors.
- Better Management: Centralized management reduces time spent on application deployment and routine maintenance. A tapping aggregation system can be used to stream traffic for analytics purposes.
- Resource flexibility: SDN offers a broad choice of innovation network applications, services, and custom development using standard tools to connect smart appliances.

Within the SDN controller, it becomes useful to set a power usage threshold; aside from the control of devices, the need for management serves importance to meet the SDN concept. How does automating a threshold level propose that the SDN controller record each appliance and set an average?

The example of a kettle is used to attain a reference level; the SDN controller will calculate the amount of time the kettle had taken to boil and make a record, then an average is recorded, and a 20% value is applied to set the threshold stability.

KR() Kettle Record

$$\left(\frac{(\mathrm{KR1}+\mathrm{KR2}+\mathrm{KR3}+\mathrm{KR4}+\mathrm{KR5})}{5}\right) \times \left(1+ \left(0,2 \times \frac{(\mathrm{KR1}+\mathrm{KR2}+\mathrm{KR3}+\mathrm{KR4}+\mathrm{KR5})}{5}\right)\right) = \text{Threshold level} \qquad (2)$$

The exact process will apply to other appliances. To ensure safety and management of an SDN Smart Home.

3.4.3 SDN within the Electrical Distribution System

The concept brings a new thinking pattern to the Service Provider of electrical power that introduces an intelligent automated network interconnected to the *Smart Grid*. The approach uses interconnecting switches configured with OpenFlow protocol to send and receive the state of an Electrical Distribution system.

Using SDN and OpenFlow, Traffic Engineering (TE) has a logically centralized control plane and a clear separation of the networking hardware in the data plane(Electrical Distribution System) from the control plane's networking software. This logically centralized control plane is expected to view the networking infrastructure’s resource usage globally.

The global view allows SDN TE applications to optimize the topology more deterministic, predictable, and efficient.

The proposed Mirror Production of traffic system uses OpenFlow capable switches instead of dedicated tap aggregation equipment to filter and store infrastructure data.

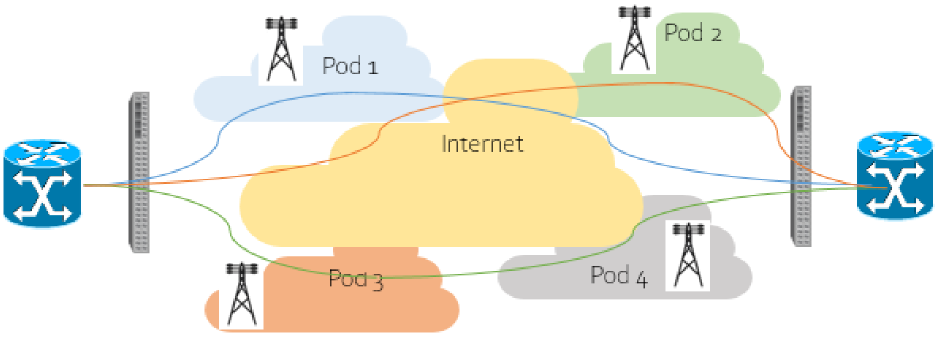


Figure 3.16: SDN Electrical Distribution network

The proposed Elephant Flow Optimization is the ability to control network infrastructure in the data centre to ensure that critical business applications continue to run with low latency while co-existing with large data set. When considering traffic flow across each pod domain in Figure 3.16, the latency will differ in different applications; for this review, the applications will be labelled critical business applications. Within the Electrical Distribution system, each Pod belongs to a region. Using the SDN controller, which contains knowledge of active devices on the network, electrical power will be diverted according to its destination when a failure of a pod occurs. The SDN controller will determine the connection path. This creates dynamics to run high bandwidth applications on the SDN/SG network while maintaining performance for legacy business applications by taking live paths of routes on the resource availability.

SDN allows big data applications that co-exist and interact with applications on a single network. SDN can use the appropriate QoS and flow rules across different ports on the network to ensure optimal use of resources based on the type of application flows that the network is seeing.

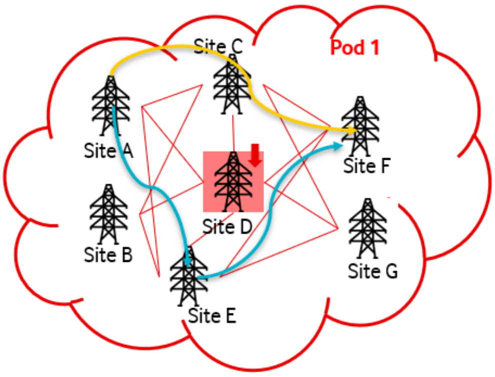


Figure 3.17: Pod 1 power diversion

Figure 3.17 shows the Electrical Distribution network of seven sites, Site D is down. The ability of the SDN controller allows the decision-making to route electrical power via other interconnected sites. In this case, electrical power is diverted from Site A to Site F.

Table 3.3: Failure scenario of power availability

Electrical power flow	**First Failure – Site D down**	**Second Failure – Site C down**
Site A – Site C – Site F	Alternate Path – Site A – Site E – Site F	Alternate Path – Site A – Site E – Site F
Site A – Site D – Site F	Alternate Path – Site A – Site C – Site F	
Site A – Site E – Site F		

Table 3.3 shows a second failure on Site C. The alternate path of electrical power flow is from site A to E to Site F. The topology defines a new way of connecting the customer and placing value on an always On electrical network that ensures a site has a diverse route. Each site will be built on a protocol to identify its neighbouring site and apply an equation to the site's amount of load. The SDN controller will continuously assess the current load added to the neighbour site load and compare the maximum load.

$$(L_{current} + L_{neighborsite)} = (L_{max)} \quad (3)$$

The advantages of SDN in the Electrical Distribution System:

- **Network performance** – SDN provides optimized application routing, reducing the need for costly MPLS networks based on configured network costing. The increase of capacity in a network increases proportionally to the increased cost. Using the thinking ability of the SDN controller, resources can be used wisely when considering traffic flow and power flow. Chapter 5 provides further details.
- **Network Reliability** – the ability to failover without delay of Border Gateway Protocol, without waiting for a complete site/node failure on rapid recovery times.
- **Manageability lower** – lower admin costs and better control, with an unparalleled level of network visibility.
- **Security Encrypted** – encrypted connectivity with frequent critical changes at configured intervals.
- **Flexibility and Scalability** – The traditional network must be upgraded consistently; the failure to increase capacity due to unplanned network growth/promotions creates congestion and a bad experience for the customer. SDN provides the ability to scale bandwidth up or down at a moment’s notice, redirecting electrical power to a failed destination. Redistribute bandwidth to accommodate new applications or Electrical Distribution services.
- **Financial Performance** – SDN eliminates the cost of costly MPLS networks.

- **Fast Office Moves or Adds** – The ability to create services or applications in minutes instead of weeks and months with MPLS. Using instant deployment allows moves, adds, and changes to the network from the controller.

3.5 Migration Strategies and Methodology

Figure 3.18 shows steps towards migration; the purpose of the research is not to build or redesign a new network but to integrate intelligent components that can perform functions from a control plane level. The key steps involved in an SDN *Smart Grid* migration are:

- Identify and prioritize the core requirements of the target network. For example, not all traditional starting network requirements may be met initially by the target Software-Defined *Smart Grid* network.
- Prepare the starting network for migration. For example, the starting network might need to be moved to a clean intermediate standard state from which the rest of the migration can proceed.
- Implement a phased network migration approach. Migrating individual devices will necessitate device-specific drivers and methods.
- Validate the results. Once the migration is completed, the target network must be validated against a documented set of requirements or expectations.

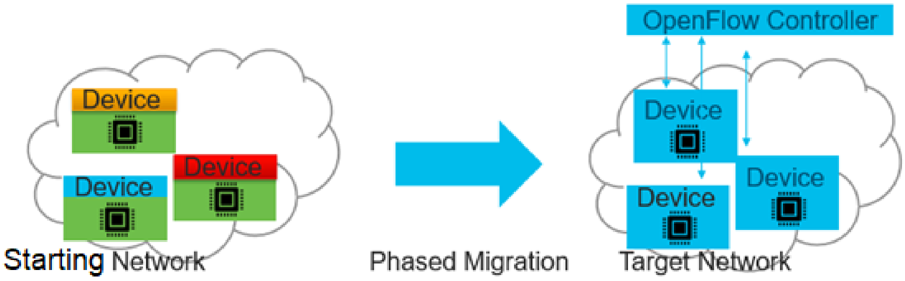


Figure 3.18: Phased Migration

There are several migration methods:

- **Direct Upgrades** are the direct method of upgrading existing networking equipment with OpenFlow agents and decommissioning the control machine favouring OpenFlow controllers and configurators.
- **Phased Upgrade** is when OpenFlow devices are deployed in conjunction with existing devices. Network operations are maintained by both the existing control machine and by OpenFlow controllers and configurations. Once services have been migrated to the OpenFlow target network, the starting network is decommissioned, including the devices and control machine.
- **Greenfields deployment**, the Greenfield deployment, is one where there is either no existing deployment or legacy network is upgraded to become OpenFlow enabled, and the control machine is replaced with an OpenFlow controller.
- **Mixed deployment**, this migration approach assumes that new OpenFlow devices are deployed and will co-exist with other traditional switches/routers and communicate with legacy control machines. The new OpenFlow controller and the traditional devices will need to exchange routing information via a legacy control machine.

In this case, Hybrid Network Deployment, both mixed network deployment and hybrid devices with both legacy and OpenFlow functionality, can coexist. In this scenario, the Hybrid devices communicate to the OpenFlow controller and the legacy control machine.

Chapter 3 provided the architecture and connectivity of Software Defined Networking and Smart Grid integrated, the need for a model that brings the two environments together to control and manage. The Chapter provides the OpenFlow model into Group Tables specifically designed for each use case. Focus is placed on SDN within Automation Plan, SDN within Smart Home, and SDN within the Electrical Distribution system. Each shows the connectivity and operating application. The chapter ends on migration and deployment recommendation, for smooth integration of Smart Grid components.

Chapter 4 shows three design phases to meet the methodology and architecture shown in Chapter 3. To prove the concept of SDN control and SDN management, use cases have been technically analyzed.

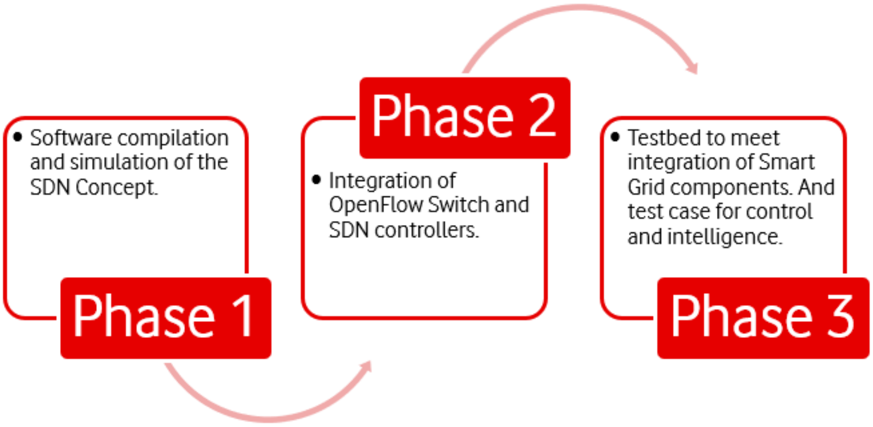
Chapter 4 presents the simulation to evaluate SDN, while section **4.1.3** shows the testbed used to control the Smart Grid component, for the testbed, electrical DC motors were used to show control output.

CHAPTER 4

TESTBED AND RESULTS

4.1 Introduction

The design consisted of three phases to showcase the methods and approaches used to prove and test the *Software-Defined Network* (SDN) in control of *Smart Grid* (SG) components and bring intelligence between a communication network and *Smart Grid*. Figure 4.1 shows the three design phase's purpose.



Software compilation and simulation of the SDN Concept.
Phase 1
Phase 2
Integration of OpenFlow Switch and SDN controllers.
Testbed to meet integration of Smart Grid components. And test case for control and intelligence.
Phase 3

Figure 4.1: Design phases of *Software-Defined Network* and *Smart Grid*

Phase 1 consisted of software to simulate network topologies and prove the SDN controller's concept of centralized control.

Phase 2 integrated an OpenFlow switch on Raspberry Pi to showcase real-time remote performance using the Floodlight GUI to prove the management of devices.

Phase 3 integration of *Smart Grid* components to prove the ability of control factor.

Each phase is built on open-source software applications used to model use cases for *Smart Grid* applications and prove motor operation control. The scope of the testbed shows the intelligent management of the communication and *Smart Grid* network and proven control from an SDN controller of the *Smart Grid* component. The SG component may belong to SDN Smart Home, SDN Automation Plant, or SDN Electrical Distribution system. Each design phase is a step approach to build use cases for testing and achieving results.

4.1.1 Phase 1: Simulation of centralized control

Phase 1 of the design project consisted of a simulation of network topology to establish the SDN Controller's ability to provide centralized control.

Service Provider' networks are growing, and the need to scale and manage its network becomes an overwhelming task. Traditional networks consist of routers and switches; each is configured and managed individually, resulting in more excellent times to bring up new services, human errors, and various proprietary Operating Systems. In addition, engineers need to keep up with the new release of firmware versions and code bug restoration.

Phase 1 shows the ease of performing configurations via the SDN Controller, being the network's centralized controller. For this analysis performed on a Windows PC, a Virtual Machine (VM) was built on Ubuntu 14.04 and Mininet to create network topologies. For secure connectivity, Open Source Putty was used to access the VM. To bring up hosts, Xming software was installed on Windows PC.

Use Case: Emulation of Network Topology Simulation

The Use Case is to emulate a network configuration without a controller, and to emulate a network configuration with a controller. Both network topologies were tested using three hosts and five host networks shown in Figures 4.2 and 4.3.

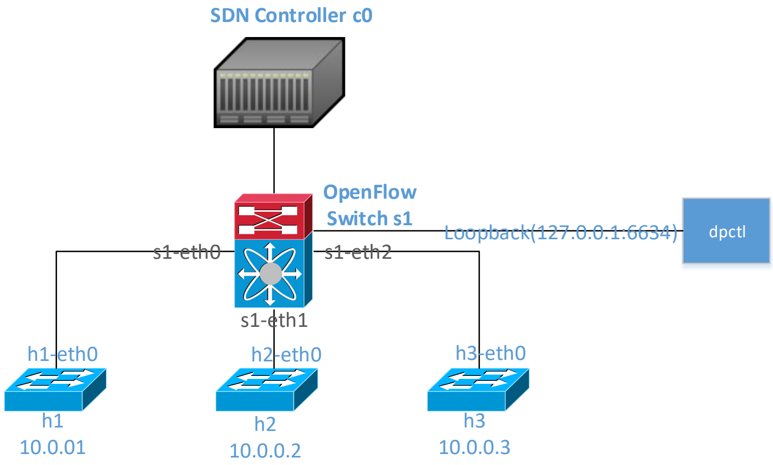


Figure 4.2: 3 Host network

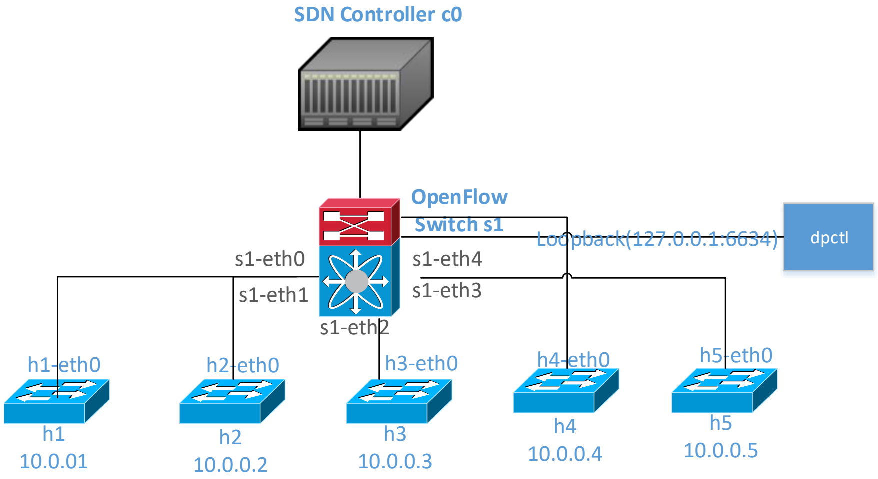


Figure 4.3: 5 Host network

The test performed proves the ability of the SDN controller able to transmit packets faster than a traditional network by configuring a 3 Hosts and 5 Hosts network; the configuration installed an OpenFlow Switch and connected to the SDN Controller. The OpenFlow Switch allows different hosts/devices to communicate to the SDN Controller. Each host was assigned a network IP address and configured on port eth0. From the SDN controller, a ping test served to learn the network topology environment.

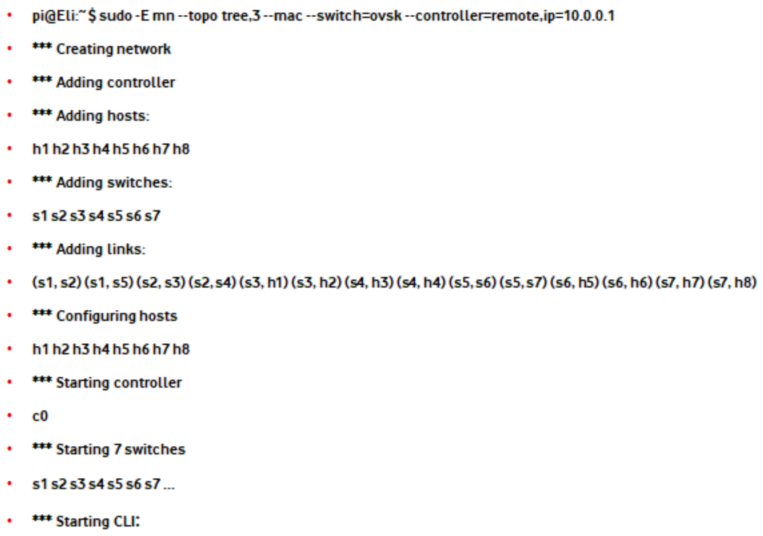
4.1.2 Phase 2: SDN Management

Each company has contractual requirements to perform management and visibility of their network devices on a real-time monitoring system. Phase 2 introduces the Floodlight Controller configured on Raspberry Pi and Floodlight GUI attached to the network's OpenFlow switch; the design phase testbed consists of Raspberry Pi acting with an OpenFlow switch connected to the internet.

Use Case Network Topology creation with SDN

The Raspberry Pi has installed Mininet to simulate network topologies and can be accessed by a putty session by the command line or from the Floodlight OpenFlow controller GUI. The aim is to prove the management ability of the SDN Controller. For security purposes, the Raspberry Pi is assigned a Username and Password.

Figure 4.4 shows the command and output for network topology, consisting of 7 switches and 8 hosts, and starting with the SDN controller.



```
pi@Eli:~$ sudo -E mn --topo tree,3 --mac --switch=ovsk --controller=remote,ip=10.0.0.1
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6) (s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
```

Figure 4.4: Network topology setup seven switches and eight hosts

The "***sudo -E mn***" command points the network setup to network IP address 10.0.0.1, which can be accessed from a web browser to showcase the live real-time network status shown in Figure 4.5. The aim is to have visibility and management of communication and *Smart Grid* devices.

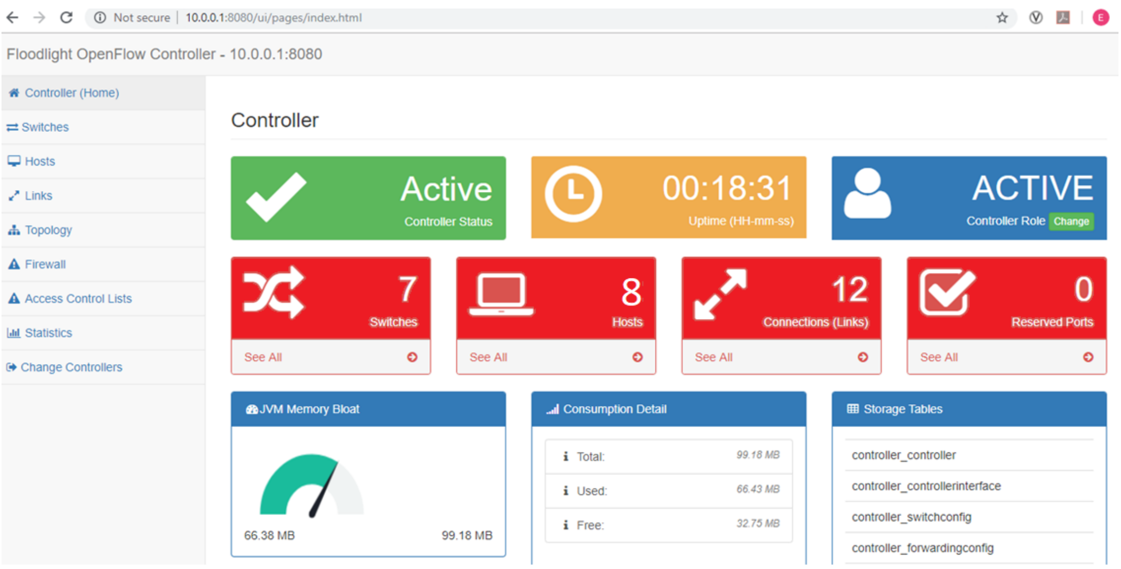
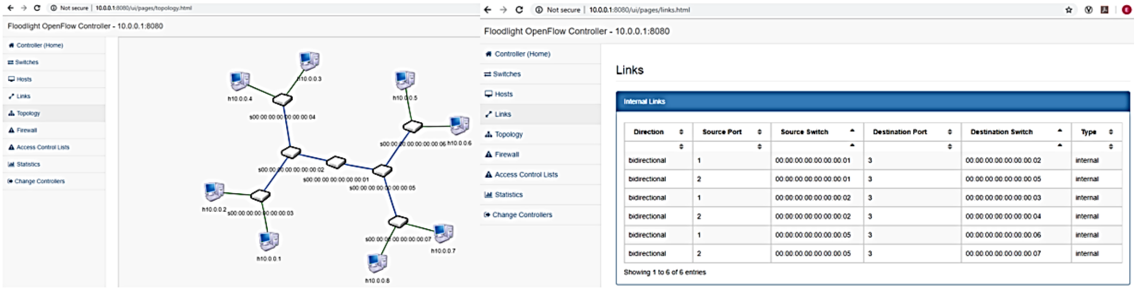


Figure 4.5: Floodlight OpenFlow Controller GUI

Figure 4.6 shows the network topology created and the allocation of MAC addresses against each port. Each category of devices is shown, indicating the status and health state. The controller's state is shown from the simplified Floodlight GUI, and on the left panel, each category of the device indicates the configure features and further detailed information within. Thus, the engineer can make changes to the Floodlight OpenFlow Controller remotely.



Links

Internal Links

Direction	Source Port	Source Switch	Destination Port	Destination Switch	Type
bidirectional	1	00:00:00:00:00:00:00:01	3	00:00:00:00:00:00:00:02	internal
bidirectional	2	00:00:00:00:00:00:00:01	3	00:00:00:00:00:00:00:05	internal
bidirectional	1	00:00:00:00:00:00:00:02	3	00:00:00:00:00:00:00:03	internal
bidirectional	2	00:00:00:00:00:00:00:02	3	00:00:00:00:00:00:00:04	internal
bidirectional	1	00:00:00:00:00:00:00:05	3	00:00:00:00:00:00:00:06	internal
bidirectional	2	00:00:00:00:00:00:00:05	3	00:00:00:00:00:00:00:07	internal

Showing 1 to 6 of 6 entries

Figure 4.6: Network topology and assignment of port information

4.1.3 Phase 3: Use Case Design Testbed

Design phase 3 focussed on integrating *Smart Grid* component to *Software-Defined Network* to meet section 1.2 objective. To achieve the aims, can a SDN controller within a network control a *Smart Grid* component, managing and bringing intelligence. In this analysis, DC motors were used, which connected to a drive module. The SDN controller contains policies or even modify templates that have already been created. A policy is a set of code that includes parameters and specifications to operate a device. When a policy is called, it will read the function and execute it in python. The policies are the parameters and settings for the *Smart Grid* component.

The information presented in Figure 4.7 shows the software and applications used to prove the concept of *Software-Defined Network*ing applied to the *Smart Grid* for the research mandate. The software is open-source and simulates different network topologies to control DC motors.

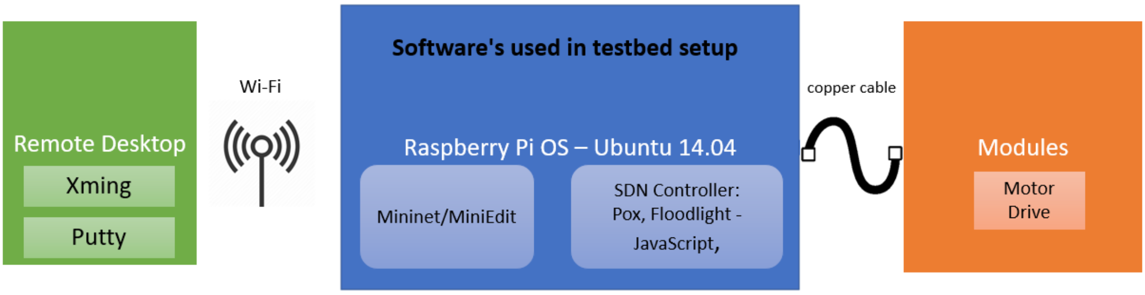


Figure 4.7: Softwares within Design Testbed

Xming

Xming is developed for Microsoft Windows to allow Linux graphical applications to run remotely. The tool allows the opening of CLI windows for hosts, switches, and controllers for configurations and mainly CLI.

Putty

Putty is an open-source terminal emulator, which supports different network protocols like SSH, designed for Windows. The emulator enables to connect to a network by safe, secure SSH session; in this case, the connection is to the Raspberry Pi Operating System.

Mininet

A tool used to create a real-world emulated network environment, creating a network setup that includes an SDN controller using OpenFlow protocol. The tool is used to set up a real-world scenario of an SDN network and communication between hardware devices.

POX

POX is a python based open source SDN controller/OpenFlow controller. To allow the controller's connectivity with switches and host and perform packet movement network operations.

Floodlight

Floodlight uses OpenFlow protocol to orchestrate the network's traffic flow. It is part of the SDN controller family and is used to provide the GUI with status information of the network topology.

Motor Drive

Basic open-source software to allow/enable motor operation for a high state or off state.

Layered architecture of *Software-Defined Network* and *Smart Grid*:

The network moves from configuring a device to provisioning services for a customer. Services range from mobility, broadband, and latency. Each service is an application slice that belongs to a specific application and is channeled to meet latency Service Level Agreements or remote connectivity. The *Smart Grid* introduces an advanced application slice level upon the communication network to overcome remote location and real-time challenges.

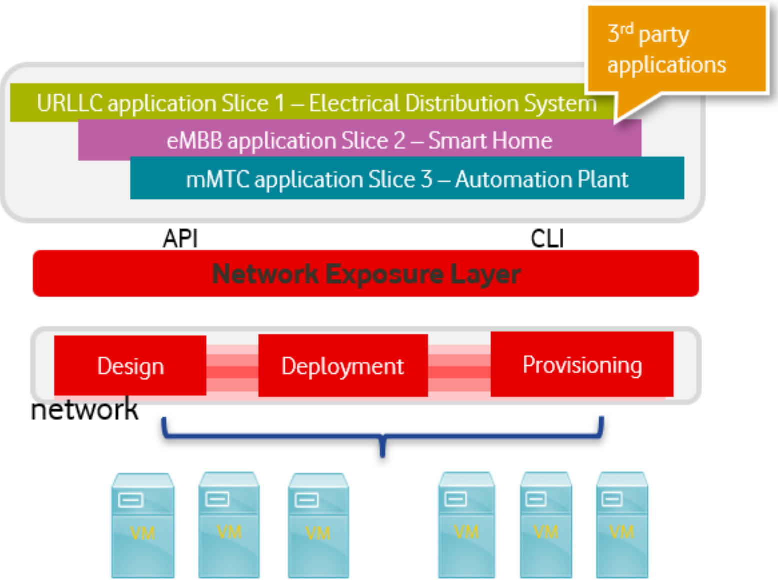

3rd party applications
URLLC application Slice 1 – Electrical Distribution System
eMBB application Slice 2 – Smart Home
mMTC application Slice 3 – Automation Plant
API
CLI
Network Exposure Layer
Design
Deployment
Provisioning
network
VM
VM
VM
VM
VM
VM

Figure 4.8: Services of Smart Grid Applications

The architecture describes the network for design, deployment, and provisioning on SDN. On top of the network is the Network Exposure Layer shown in Figure 4.8. The Network Exposure layer interconnects the communication network and the *Smart Grid* application slice. The exposure layer will contain a catalogue of policies and APIs depending on each *Smart Grid* environment. In addition, the exposure layer enables a standardized catalogue to accommodate proprietary software in Table 4.1.

The policy contains a set of code or instructions for the operation of the *Smart Grid* environment, to control and manage the network's components. The controller contains these policies or even modify templates that have already been created. When a policy is called on the controller, the controller will read the function and execute it in python. The policies are the parameters and settings for the *Smart Grid* component.

Table 4.1: API Standardized Catalog

Standardized Catalog	
Smart Home API	Policy Smart Home
Electrical Distribution API	Policy Electrical Distribution
Automation Plant API	Policy Automation Plant

The layered architecture in Figure 4.9 incorporated a new sub-layer, network exposure layer, which exposes the Application Programmable Interfaces (APIs) on a standardized protocol. The layer serves to integrate the *Smart Grid* components by using OpenFlow protocol.

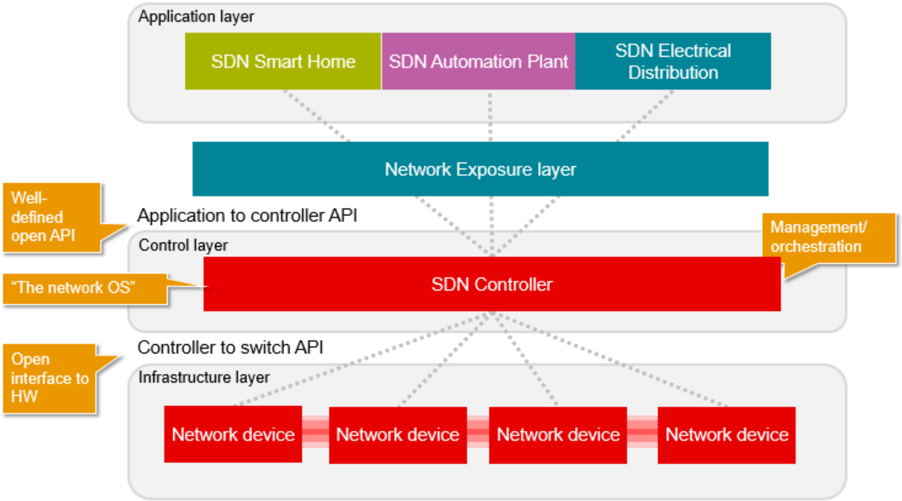


Figure 4.9: SDN architecture incorporating a network exposure layer

Mininet, an open-source tool, was used to create network topologies. A Raspberry Pi was configured to act as an OpenFlow Switch in the design. Any network hosts and controller types can be selected when creating the network topology.

The General Purpose Input/Output (standard interface) on the Raspberry Pi connects microcontrollers to other electronic devices for this analysis. For example, it can be used with sensors, diodes, displays, and system-on-chip modules. In addition, the testbed connects to an external power supply and remote control of connected devices.

The Raspberry Pi has wifi capability, which allows the user to configure and connect to a Service Provider's network; this allows the user to remotely configure and execute files/commands/services from his PC. No need for an Ethernet cable is required or to be on site.

The Raspberry Pi requires installing different software; for programmability Java, Floodlight, POX, Xming, GPIO, and Mininet are necessary.

Installation code for Java: **Appendix 2**

Installation code for Floodlight: **Appendix 2**

Figure 4.10 shows a traditional communication system and separate *Smart Grid* Automation plant system. Both systems are unique and do not share the same communication network. The communication network consists of different vendors, Cisco and Nokia systems; an adaptor is designed to communicate with each other on a service level. Proprietary software is needed to execute configurations on either Cisco or Nokia individually. The *Smart Grid* Automation plant consists of Programme Logic Controller, only catered for a specific Automation plant, connected to drive motor and DC motors. The PLC contains its communication module, developed to cater for logic high or low on inputs and outputs. Both systems lack centralized intelligence, and their control plane and forwarding plane are combined.

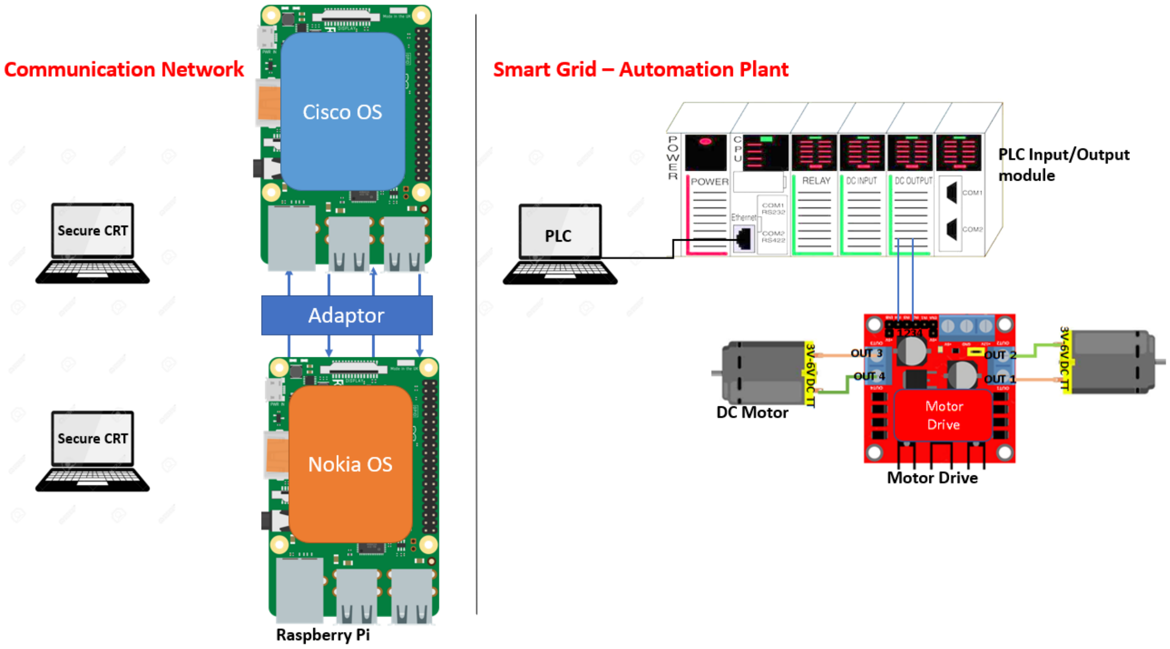


Figure 4.10: Traditional Communication and *Smart Grid* Automation plant systems

Figure 4.11 shows the Testbed schematic of the Raspberry Pi, Drive module, and motors connected.

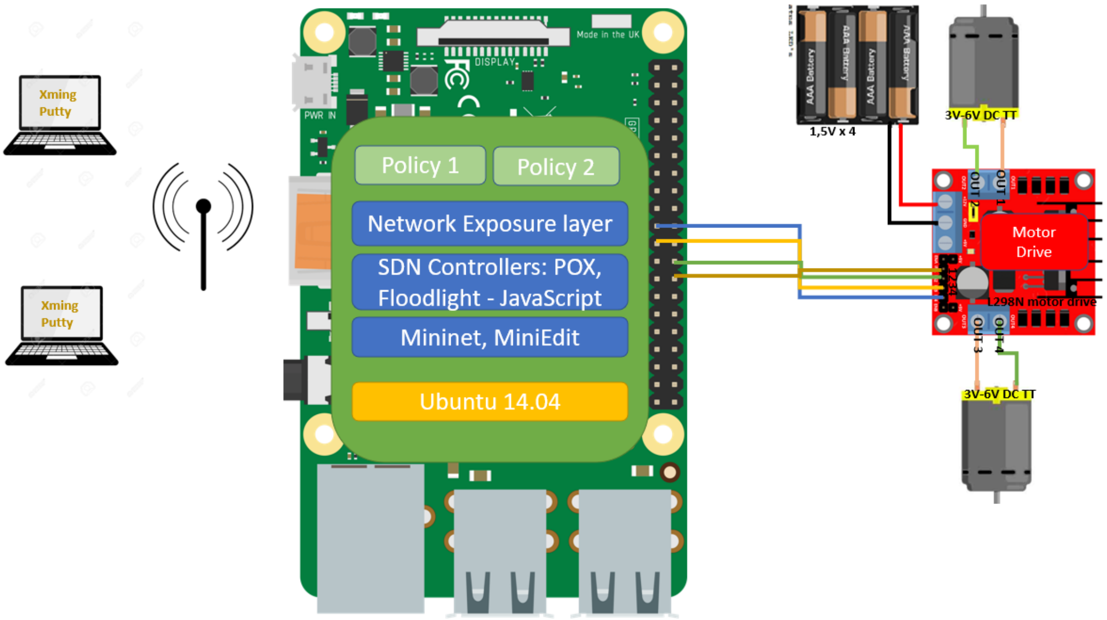


Figure 4.11: Testbed: *Smart Grid* components control and management by SDN controller

Once the Raspberry Pi is powered on, the User will pick up the Raspberry Pi's IP address. A mobile phone will identify Pi's IP address. The file Operating System configured on the Raspberry Pi is Ubuntu 14.04. Using Putty to connect securely, the user will be able to SSH into the Raspberry Pi. Logins will be required as a security measure. In this analysis, different Open Source controllers were tested; the first recommended step is to call the controller, in this case, Floodlight.

cd/floodlight

sudo java –jar ./target/floodlight.jar

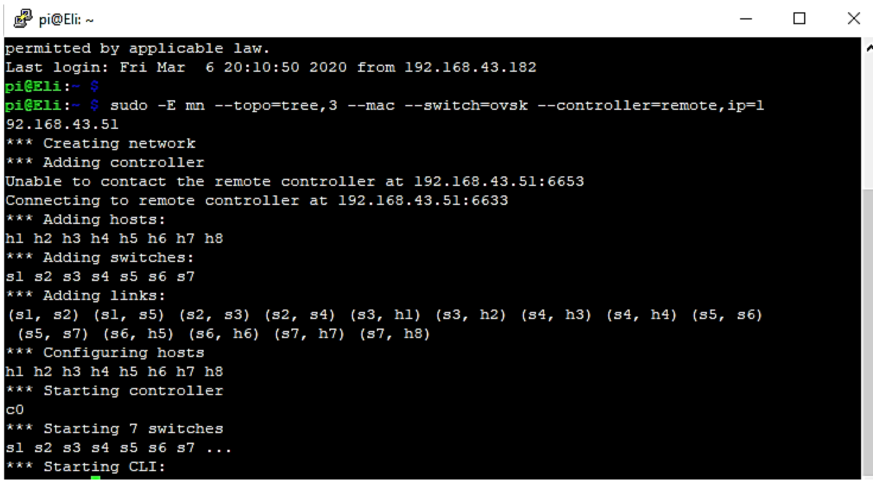
From a web browser, open a new session and type (ip):8080/ui/index.html. The command will connect the network device to the Floodlight GUI.

For the analysis:

198.168.43.51:8080/ui/index.html

The next step is to create the topology: *sudo –E mn –topo=single/tree,() –mac –switch=ovsk –controller=remote,ip=198.168.43.51*

Figure 4.12 shows the creation of the successful topology, and Figure 4.13 shows a sample of Floodlight GUI showing the real-time status of devices.



```
permitted by applicable law.
Last login: Fri Mar  6 20:10:50 2020 from 192.168.43.182
pi@Eli:~ $
pi@Eli:~ $ sudo -E mn --topo=tree,3 --mac --switch=ovsk --controller=remote,ip=1
92.168.43.51
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.43.51:6653
Connecting to remote controller at 192.168.43.51:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6)
 (s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
```

Figure 4.12: Creation of network topology

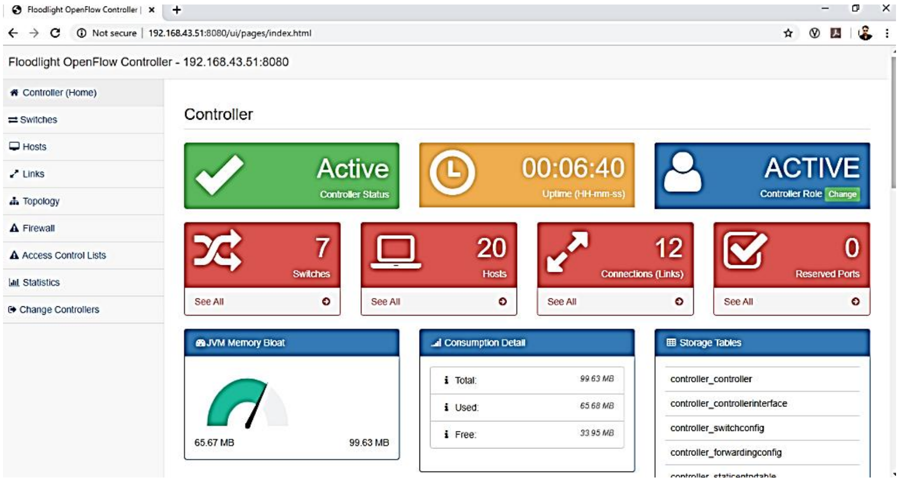
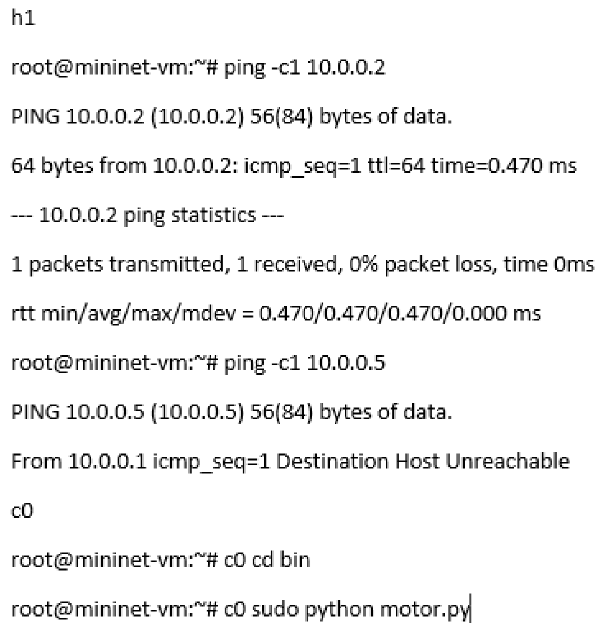


Figure 4.13: Floodlight GUI shows an active state of the controller

The next step uses xterm to bring up the controller and host/s; for this software to communicate, an open-source tool Xming had to be downloaded and configured. Xming was configured with settings for a Windows PC.

The configuration system had many errors in terms of disk/module capability and Windows security parameters. With the use of open-source code, restoration techniques assisted in fixing bugs. After bringing up xterm for the controller (c0) and host (h1), the user can call a parameter file or policy and execute it. The SDN Controller plays a role in management; for this scope, a VLAN was configured to create a logical network within the network topology; for this case, we can manage which host can receive network packets and block host 5. Figure 4.14 shows the results after setting up a VLAN network that enables a VLAN per service or application slice. The service or application slice could be for SDN Smart Home or SDN Automation Plant. The controller's capability provided the hosts' management, allowing manual traffic blocking to a specific host and Access Control List setup.



```
h1
root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.470 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.470/0.470/0.470/0.000 ms
root@mininet-vm:~# ping -c1 10.0.0.5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
c0
root@mininet-vm:~# c0 cd bin
root@mininet-vm:~# c0 sudo python motor.py
```

Figure 4.14: h1 allowing traffic on the specific host

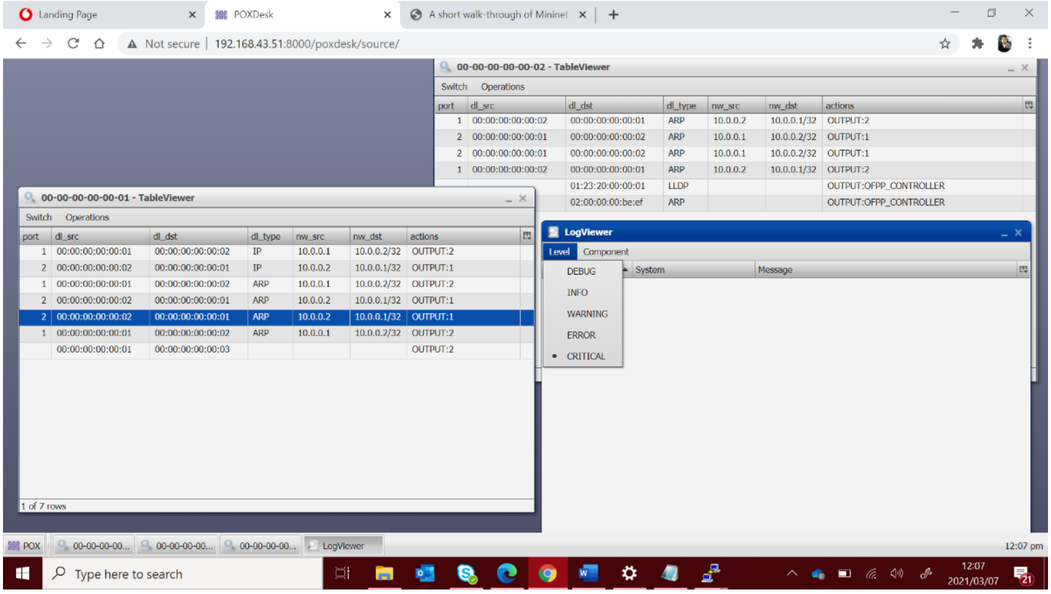
A motor file was called and executed on the c0 – controller for this specific analysis. According to the parameters set, the motor-operates show using a centralized controller to manage and control the *Smart Grid* component. In addition, a driver software(python rpi.gpio) had to be loaded to control power levels from the domain controller, enabling the GPIO pins. The Network Exposure Layer sets up new communication streams. It allows different vendor-specific software to communicate on the OpenFlow controller/switch platform, which carries the policy's parameters to be executed. The same exposure is also applied to a 5G network, exposing its capabilities and services. And can be applied to a service: operation of plant components or substation switching. The file code is pasted below:

```
File: Motor.py
    Import RPi.GPIO as gpio
```

```
      Def init():
gpio.setmode(gpio.BCM)
gpio.setup(17,gpio.OUT) # dependent on port
      gpio.setup(22,gpio.OUT)
      gpio.setup(23,gpio.OUT)
      gpio.setup(24,gpio.OUT)
   def forward(sec)
init()
gpio.output(17,True) # dependent on port
gpio.output(22,False)
      gpio.output(23,True)
      gpio.output(24,False)
      time.sleep(sec)
      gpio.cleanup
   def reverse(sec):
       init()
      gpio.output(17,False)
      gpio.output(22,True)
      gpio.output(23,False)
      gpio.output(24,True)
      time.sleep(sec)
      gpio.cleanup
   print "forward"
   forward(10) # timer set for motor to run forward
   print "reverse"
   reverse(5) # timer set for motor to run in reverse
   quit()
```

Communication to *Smart Grid* Component/host.

The test analysis was performed using a POX controller to showcase the actions and functions; The setup consisted of a single controller and 3 OpenFlow switches allocated to each *Smart Grid* domain, the domain of Smart Home, the Electrical Distribution system, and the Automation Plant. The actions per event are recorded and seen in the below Figure 4.15. The TableViewer displays the actions upon each action in the network setup; the action indicates the protocol and packet movement from its source to the destination. Each switch will show its communication with the other device and its OpenFlow output, such as OFPP_CONTROLLER.



00-00-00-00-00-02 - TableViewer

port	dl_src	dl_dst	dl_type	nw_src	nw_dst	actions
1	00:00:00:00:00:02	00:00:00:00:00:01	ARP	10.0.0.2	10.0.0.1/32	OUTPUT:2
2	00:00:00:00:00:01	00:00:00:00:00:02	ARP	10.0.0.1	10.0.0.2/32	OUTPUT:1
2	00:00:00:00:00:01	00:00:00:00:00:02	ARP	10.0.0.1	10.0.0.2/32	OUTPUT:1
1	00:00:00:00:00:02	00:00:00:00:00:01	ARP	10.0.0.2	10.0.0.1/32	OUTPUT:2
		01:23:20:00:00:01	LLDP			OUTPUT:OFPP_CONTROLLER
		02:00:00:00:be:ef	ARP			OUTPUT:OFPP_CONTROLLER

00-00-00-00-00-01 - TableViewer

port	dl_src	dl_dst	dl_type	nw_src	nw_dst	actions
1	00:00:00:00:00:01	00:00:00:00:00:02	IP	10.0.0.1	10.0.0.2/32	OUTPUT:2
2	00:00:00:00:00:02	00:00:00:00:00:01	IP	10.0.0.2	10.0.0.1/32	OUTPUT:1
1	00:00:00:00:00:01	00:00:00:00:00:02	ARP	10.0.0.1	10.0.0.2/32	OUTPUT:2
2	00:00:00:00:00:02	00:00:00:00:00:01	ARP	10.0.0.2	10.0.0.1/32	OUTPUT:1
2	00:00:00:00:00:02	00:00:00:00:00:01	ARP	10.0.0.2	10.0.0.1/32	OUTPUT:1
1	00:00:00:00:00:01	00:00:00:00:00:02	ARP	10.0.0.1	10.0.0.2/32	OUTPUT:2
	00:00:00:00:00:01	00:00:00:00:00:03				OUTPUT:2

1 of 7 rows

Figure 4.15: POX controller and TableViewer

Figure 4.16 shows the communication of messages between the devices. Between the *Smart Grid* Pod, forwarding element/switch, and SDN controller. The analysis confirms actions for experimental purposes. For example, the execution of motor policy will result in an action showing the OUTPUT executed and the IP address the action occurred on.

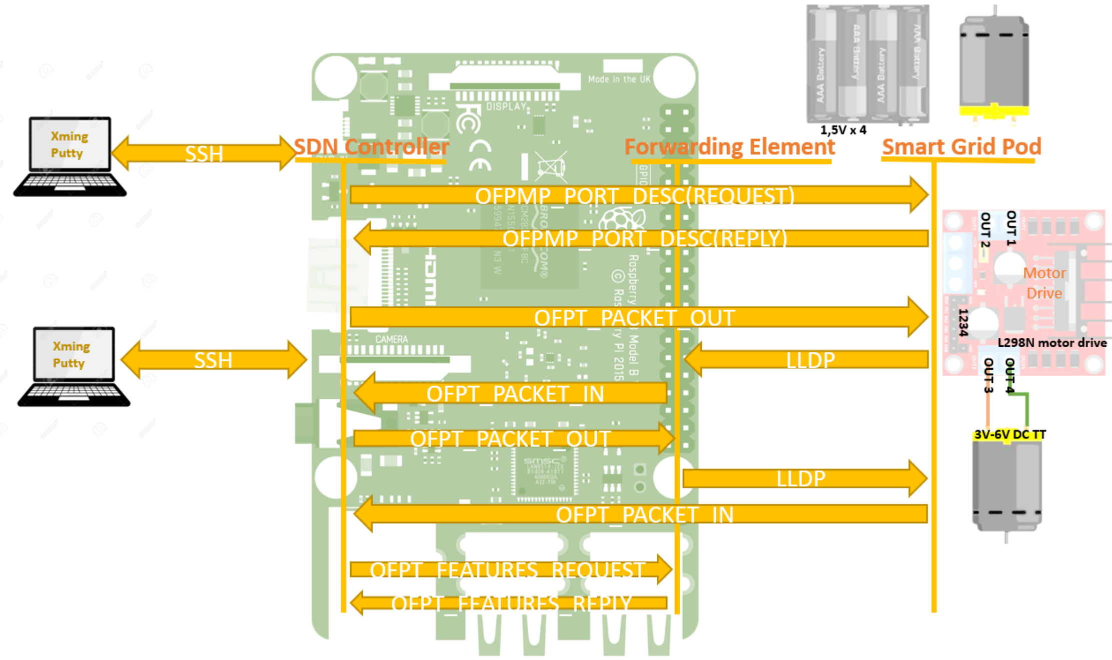


Figure 4.16: OpenFlow request and reply actions between SDN controller, Forwarding Element, and *Smart Grid* Pod

4.1.3.1 Data packet exchange between the SDN controller, Forwarding Element, and *Smart Grid* node

The steps below illustrate how OpenFlow works between an SDN controller and an OpenFlow switch:

Step 1: Connection setup between the Forwarding element and SDN controller

Step 2: Proactive flow programming

Step 3: Topology discovery via Layer Link Discovery Protocol (LLDP) of *Smart Grid* component

Step 4: Control plans maintenance and reactive flow programming

Step 1: Connection setup between Forwarding Element and SDN Controller

The Forwarding Element may optionally accept TCP/TLS connections from the controller. In addition, the switch may allow the controller to initiate the connection. In this case, the Forwarding element should accept incoming standard TLS or TCP connections from the SDN controller, using either a user-specified transport port or default OpenFlow transport port 6653. Connections initiated by the Forwarding element and the controller behave the same once the transport connection is established.

- First, a TCP handshake occurs.
- The official IANA registered port for OpenFlow is 6653 as of the OpenFlow 1.3. However, most vendors use 6633 by default, which was suggested by the original OpenFlow 1.0 specification.
- Next, the Forwarding element and SDN controller exchange an OFPT_HELLO packet. The OFPT_HELLO contains the supported OpenFlow version.
- When negotiating the version, if both the forwarding element and controller send the same version, the connection will be established. The protocol is also arranged via the version bit map, assuming both forwarding element and controller send the OFPHET_VERSIONBITMAP. Otherwise, both devices must accept the lower version number.
- Lastly, set up messages on the feature negotiation occurs OFPT_FEATURES_REQUEST in Figure 4.17.

The SDN controller requests the switch to declare its necessary capabilities. The switch may declare its DPID (data-path ID) to identify its uniqueness in the OpenFlow network in this message and max buffer size and several tables supported.

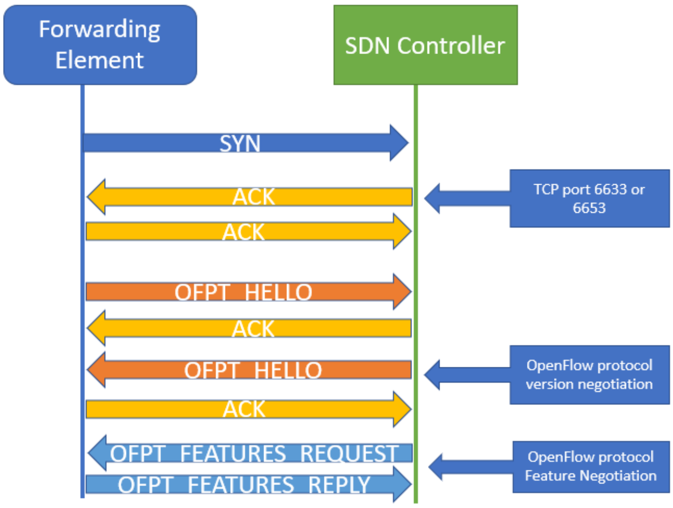


Figure 4.17: Step 1: Connection setup between OpenFlow switch and SDN Controller

Step 2: Proactive Flow Programming

The exchange of 'proactive' flows onto the Forwarding Element. For example, the OFPT_FLOW_MOD packet is generated by the SDN controller instructing the Forwarding element to add flows into its flow table. These flows are proactive. They are not responding to any packets received by the SDN controller (e.g., reactive flows). A flow is an instruction set to the *Smart Grid* component.

The OFPT_BARRIER_REQUEST message to have the Forwarding Element validate the change was made to confirm that the flows were successfully added. This ensures that the state in the OpenFlow switch's forwarding table is what the SDN controller expects. It also ensures messages are processed in the proper order. This shows the action for the *Smart Grid* component was successful or not. In other words, the Forwarding element must complete all operations received before the BARRIER_REQUEST message shown in Figure 4.18, as indicated by the transaction ID.

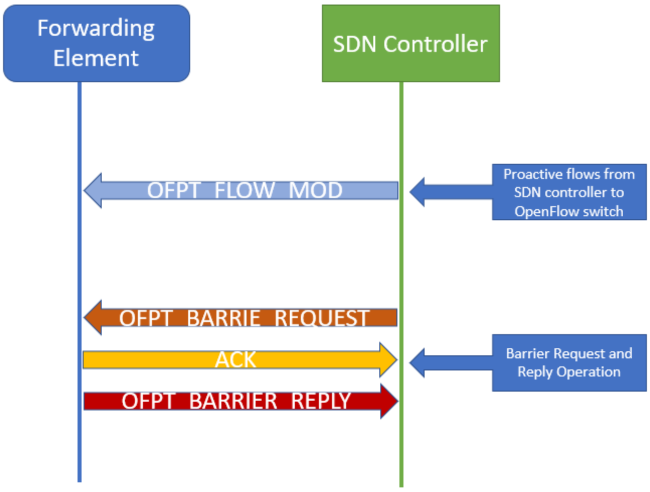

Forwarding Element
SDN Controller
OFPT_FLOW_MOD
Proactive flows from SDN controller to OpenFlow switch
OFPT_BARRIE_REQUEST
ACK
OFPT_BARRIER_REPLY
Barrier Request and Reply Operation

Figure 4.18: Step 2: Proactive Flow Programming

Step 3: Topology Discovery via LLPD for *Smart Grid* component

The port description request OFPMP_PORT_DESCRIPTION enables the controller to describe all the standard ports of the Flowing element. This structure is the standard port structure representing ports and includes the port number, port config, and port status. The port description reply must consist of all the standard ports defined in the

Forwarding element or attached, regardless of their state or configuration. The *Smart Grid* component will be described in a real-world deployment.

When the controller wishes to send a packet out through the data path, it uses the OFPT_PACKET_OUT message. On the other hand, when a packet is received by the data path and forwarded to the controller, they use the OFPT_PACKET_IN message shown in Figure 4.19.

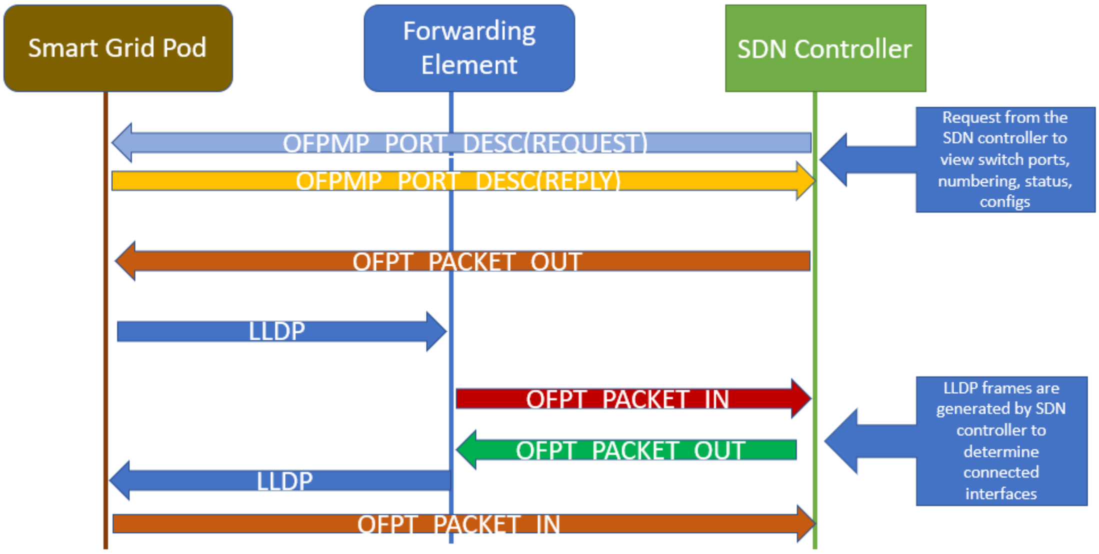


Figure 4.19: Step 3: Topology Discovery via LLPD of *Smart Grid* component

Step 4: Control Plane Maintenance and Reactive Flows

The control plane maintenance to ensure connectivity between the Forwarding elements using OFPT_ECHO_REQUEST/REPLY for control plane maintenance.

To ensure the maintenance of the control plane session between the forwarding elements and SDN controller. Forwarding element and SDN controllers may use the OFPT_ECHO_REQUEST_REPLY messages shown in Figure 4.20. The forwarding element or controller may initiate this process; reactive flows can be programmed based on the operation. Reactive flows typically happen when the SDN controller learns about a station connected to the OpenFlow network. For example, considering the Smart Gird Pod switch, each port is connected to a *Smart Grid* component; the same learning occurs on each interface or host. The discovery method varies depending on the vendor; one example may be based on a PACKET_IN of an ARP message from a host. When the SDN controller discovers a host, it may program a forwarding rule on the entire *Smart Grid* infrastructure for a specified forwarding behaviour depending on the application.

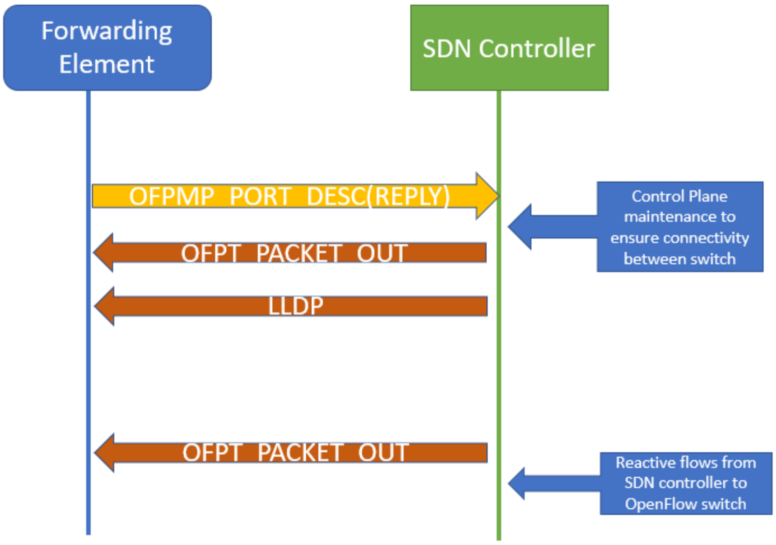

Forwarding Element
SDN Controller
OFPMP_PORT_DESC(REPLY)
OFPT_PACKET_OUT
LLDP
OFPT_PACKET_OUT
Control Plane maintenance to ensure connectivity between switch
Reactive flows from SDN controller to OpenFlow switch

Figure 4.20: Step 4: Control Plane Maintenance and Reactive Flows

4.2 Simulation and Testbed Results

Section **4.2** presents a use case to meet section **1.2** objectives and prove the control ability within the Smart Grid environment.

Use Case: Emulation of Network Topology Results

The ping results shown in Figure 4.21 calculate the time taken to reach the destination device on the traditional network and an SDN network. The same was applied to both the 3 Host and 5 Host networks. The result shows positive on the SDN network, host 10.0.0.2-3Host-SDN, 10.0.0.3-3Host-SDN, 10.0.0.2-5 Host-SDN and 10.0.0.5-5Host-SDN show an average time of 0.476m.s, indicating a ratio of 97% improvement speed. Figure 4.21 shows the results of a network without an SDN controller and with an SDN controller. The SDN network results end with 'SDN' on the description. The results show that the more hosts in a network, the decrease in speed when looking at a network without the SDN controller; this becomes the opposite of the results that include an SDN controller in the network. The speed increases when using an SDN controller in reaching its network destination.

Within any network, it’s essential to test network performance on speed and packet loss to reference and improve the customer’s experience. Therefore, the network's speed, availability, and latency are tested on drive testing. Drive testing is the process of measuring a network’s performance based on KPIs. For example, *Smart Grid* Energy plants are usually positioned at remote locations with high winds for wind power turbines. From the evidence of ping testing, the SDN concept proves its faster speed and reachability; the SDN Controller maintains an updated route table of hosts.

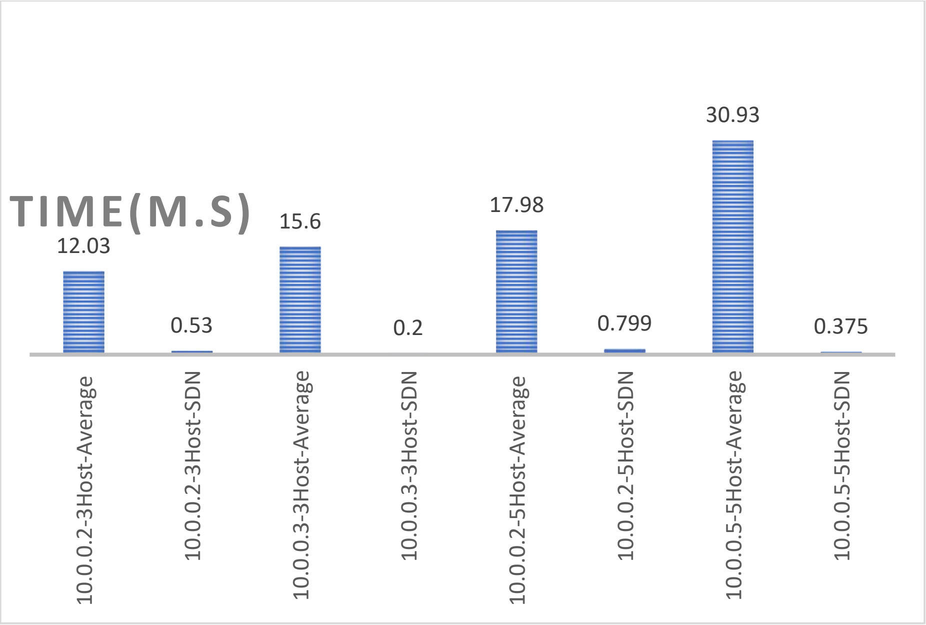
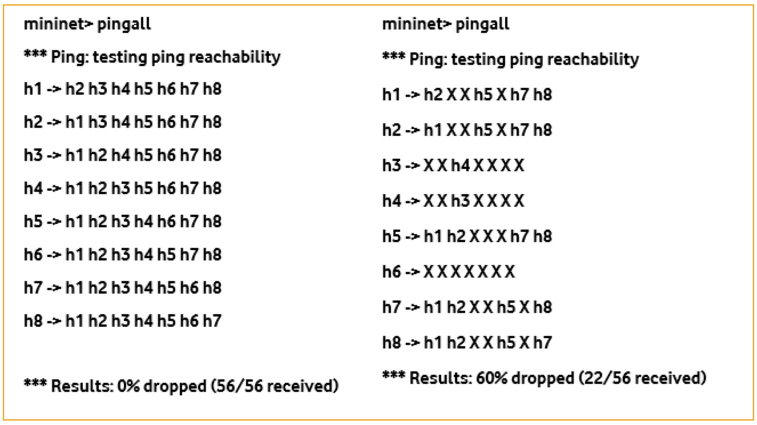


Figure 4.21: Ping test of host scenario using SDN Controller and without SDN Controller

Use Case Network Topology creation and SDN 0% drop rate Results

The analyses tested each host's reachability to measure the speed of the network topology; Figure 4.22 shows two sets of ping results; the first shows a 0% drop rate, and the second shows a 60% drop rate when a link is down. The results again show that when a link or site is down, traffic can still re-route upon the SDN controller’s route table. Thus, showing the re-routing of packets to reach their host. Finally, phase 3 will discuss the Cube analysis to perform routing between regions.



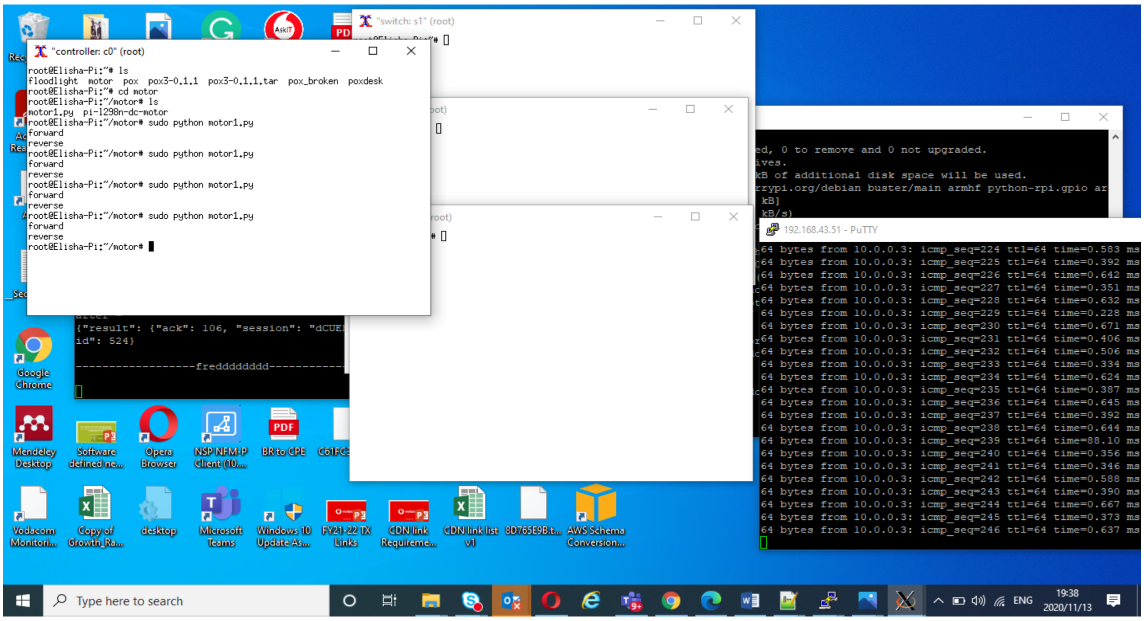
```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7

*** Results: 0% dropped (56/56 received)
```

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X h5 X h7 h8
h2 -> h1 X X h5 X h7 h8
h3 -> X X h4 X X X X
h4 -> X X h3 X X X X
h5 -> h1 h2 X X X h7 h8
h6 -> X X X X X X X
h7 -> h1 h2 X X h5 X h8
h8 -> h1 h2 X X h5 X h7
*** Results: 60% dropped (22/56 received)
```

Figure 4.22: Ping results with and without SDN controller on a link failure

Use Case Design Testbed Results

Figure 4.23 shows the controller command line, using policies stored within a directory, a python file is called, to be executed; for this exercise, the 'motor1.py' file was introduced and executed.



"controller: c0" (root)

```
root@Elisha-Pi:~# ls
floodlight  motor  pox  pox3-0.1.1  pox3-0.1.1.tar  pox_broken  poxdesk
root@Elisha-Pi:~# cd motor
root@Elisha-Pi:~/motor# ls
motor1.py  pi-l298n-dc-motor
root@Elisha-Pi:~/motor# sudo python motor1.py
forward
reverse
root@Elisha-Pi:~/motor# sudo python motor1.py
forward
reverse
root@Elisha-Pi:~/motor# sudo python motor1.py
forward
reverse
root@Elisha-Pi:~/motor# sudo python motor1.py
forward
reverse
root@Elisha-Pi:~/motor# █
```

```
{"result": {"ack": 106, "session": "dCUE
id": 524}

-------------------freddddddddd-----------
```

```
ed, 0 to remove and 0 not upgraded.
ives.
kB of additional disk space will be used.
rrypi.org/debian buster/main armhf python-rpi.gpio ar
 kB]
 kB/s)
```

192.168.43.51 - PuTTY

```
64 bytes from 10.0.0.3: icmp_seq=224 ttl=64 time=0.583 ms
64 bytes from 10.0.0.3: icmp_seq=225 ttl=64 time=0.392 ms
64 bytes from 10.0.0.3: icmp_seq=226 ttl=64 time=0.642 ms
64 bytes from 10.0.0.3: icmp_seq=227 ttl=64 time=0.351 ms
64 bytes from 10.0.0.3: icmp_seq=228 ttl=64 time=0.632 ms
64 bytes from 10.0.0.3: icmp_seq=229 ttl=64 time=0.228 ms
64 bytes from 10.0.0.3: icmp_seq=230 ttl=64 time=0.671 ms
64 bytes from 10.0.0.3: icmp_seq=231 ttl=64 time=0.406 ms
64 bytes from 10.0.0.3: icmp_seq=232 ttl=64 time=0.506 ms
64 bytes from 10.0.0.3: icmp_seq=233 ttl=64 time=0.334 ms
64 bytes from 10.0.0.3: icmp_seq=234 ttl=64 time=0.624 ms
64 bytes from 10.0.0.3: icmp_seq=235 ttl=64 time=0.387 ms
64 bytes from 10.0.0.3: icmp_seq=236 ttl=64 time=0.645 ms
64 bytes from 10.0.0.3: icmp_seq=237 ttl=64 time=0.392 ms
64 bytes from 10.0.0.3: icmp_seq=238 ttl=64 time=0.644 ms
64 bytes from 10.0.0.3: icmp_seq=239 ttl=64 time=88.10 ms
64 bytes from 10.0.0.3: icmp_seq=240 ttl=64 time=0.356 ms
64 bytes from 10.0.0.3: icmp_seq=241 ttl=64 time=0.346 ms
64 bytes from 10.0.0.3: icmp_seq=242 ttl=64 time=0.588 ms
64 bytes from 10.0.0.3: icmp_seq=243 ttl=64 time=0.390 ms
64 bytes from 10.0.0.3: icmp_seq=244 ttl=64 time=0.667 ms
64 bytes from 10.0.0.3: icmp_seq=245 ttl=64 time=0.373 ms
64 bytes from 10.0.0.3: icmp_seq=246 ttl=64 time=0.637 ms
```

Figure 4.23: Controller c0 motor1.py execution

Figure 4.23 shows the execution of motor1.py and the communication of the motor running forward 'forward' and 'reverse.'

The network topology presented in Figure 4.24 shows a new network created; each device is allocated to a network IP address. We can monitor and configure domains for each function or output from the Floodlight GUI. The host also can be seen as *Smart Grid* components within its domain.

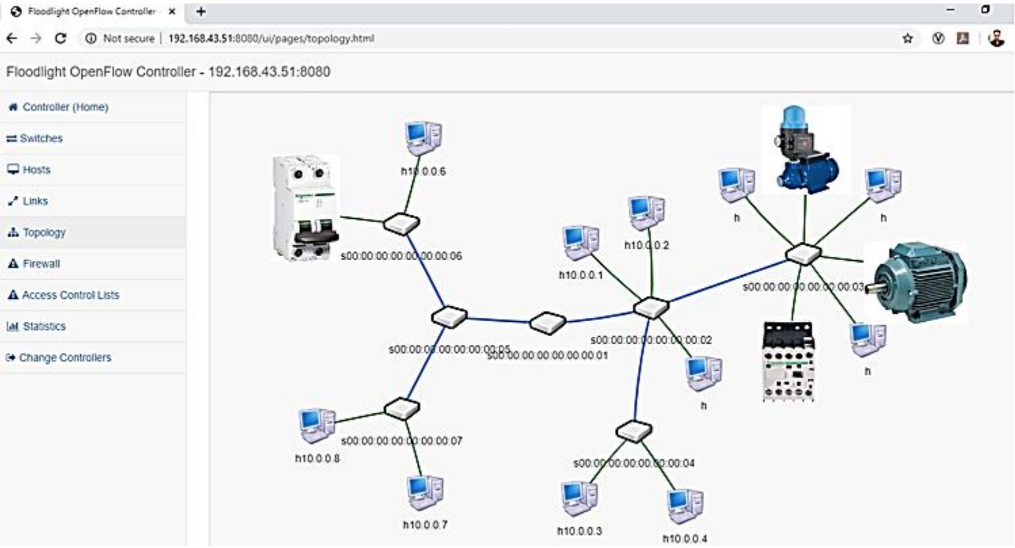


Figure 4.24: Network topology with *Smart Grid* components

4.2.1 Use case Result 1

POC: Control of *Smart Grid* component via SDN Controller

The testbed topology consisted of hosts and *Smart Grid* components, each with its unique network's IP address. The test proves the SDN controller's ability to call policies within the SDN controller and execute them on a live real-time network. Use Case 1 shows the execution of policy motor1.py on motors 1 and 2.

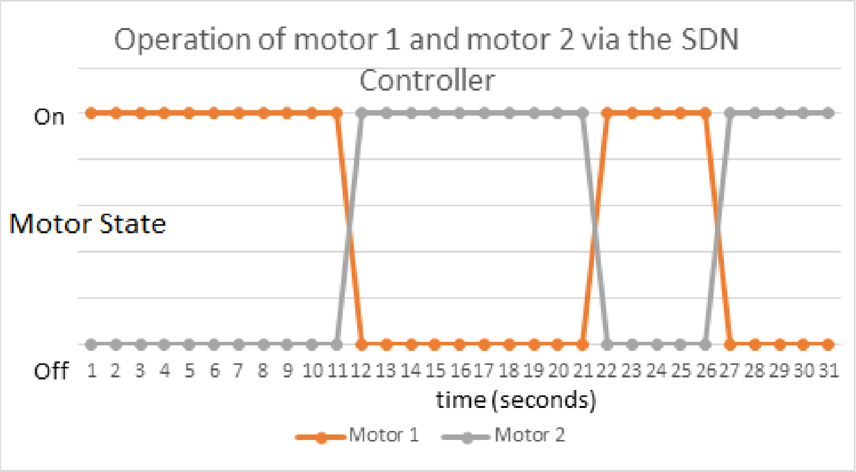


Figure 4.25: Network policy executed by SDN controller to control motor 1 and motor 2

Figure 4.25 shows the execution of motor one and motor 2; Motor 1 runs for 10 seconds and then goes to Off state; at 11 seconds, motor 2 starts operation and ends at 22 seconds. The operation of motor one and motor two states is within the policy that is run from the SDN controller has proven the ability to control remotely *Smart Grid* components.

4.2.2 Use Case Result 2

4.2.2.1 Automation of SDN network test analysis

The analysis consists of a simulated network using Miniedit connected to the CLI; from this network setup, automating the network functions in the recovery of link failure and assessing how the controller can re-route the traffic. The aim is to achieve zero loss of packets upon a link failure. The controller will contain routing tables that learn the network environment and execute policies. Presently, the traditional network does not have these features built-in. Only upon manual configuration the traffic is re-routed to another path, which the network will drop packets and result in traffic congestion.

The test case shown in Figure 4.26 created a new network topology with multiple hosts/switches/controllers. The network consisted of 10 hosts, eight switches, and three controllers. Each device is required to be unique in IP addresses.

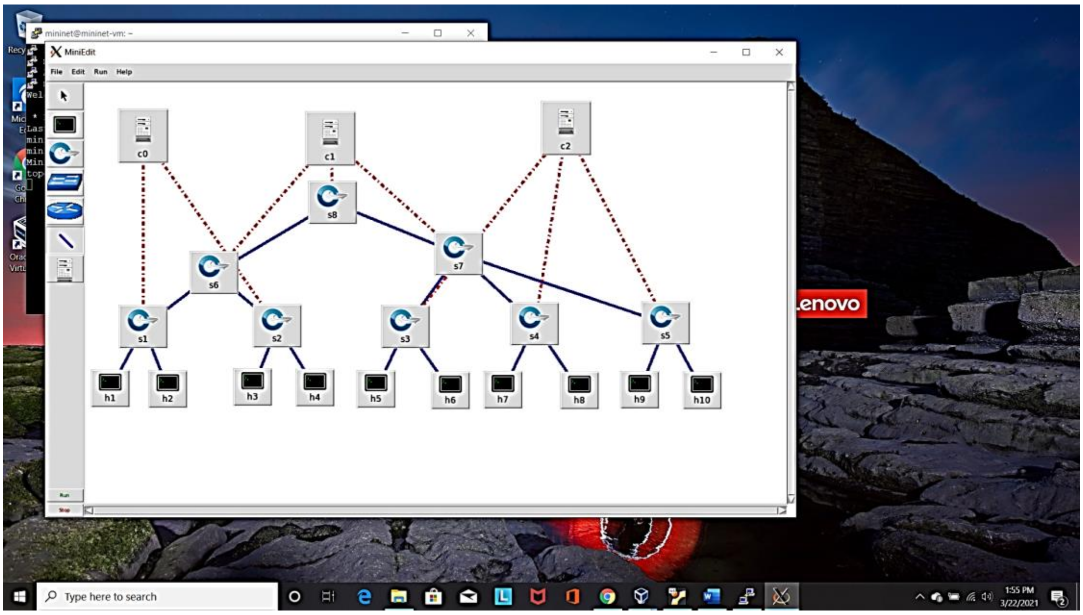
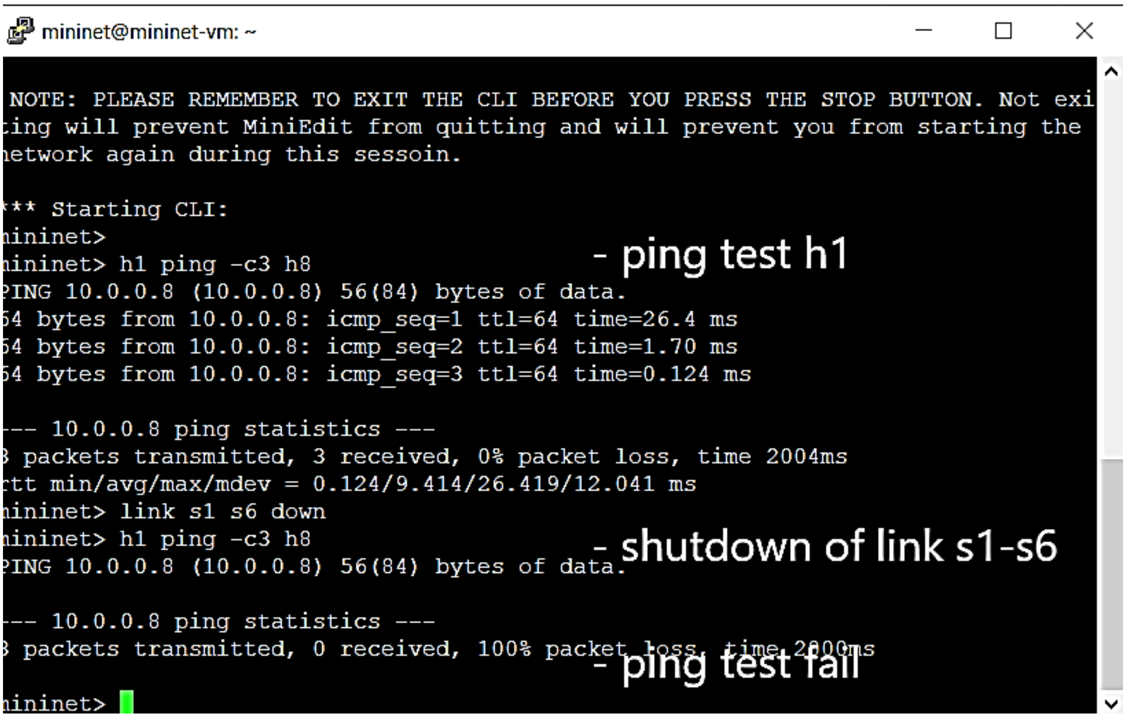


Figure 4.26: Simulation of network topology on MiniEdit

The next step was to assess connectivity when a link s1-s6 is down, and non-enablement of controllers (c0, c1, c2); the ping test indicated failed in Figure 4.27.



```
 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exi
ing will prevent MiniEdit from quitting and will prevent you from starting the
etwork again during this sessoin.

** Starting CLI:
ininet>
ininet> h1 ping -c3 h8
ING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
4 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=26.4 ms
4 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=1.70 ms
4 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.124 ms

-- 10.0.0.8 ping statistics ---
 packets transmitted, 3 received, 0% packet loss, time 2004ms
tt min/avg/max/mdev = 0.124/9.414/26.419/12.041 ms
ininet> link s1 s6 down
ininet> h1 ping -c3 h8
ING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.

-- 10.0.0.8 ping statistics ---
 packets transmitted, 0 received, 100% packet loss, time 2000ms

ininet>
```

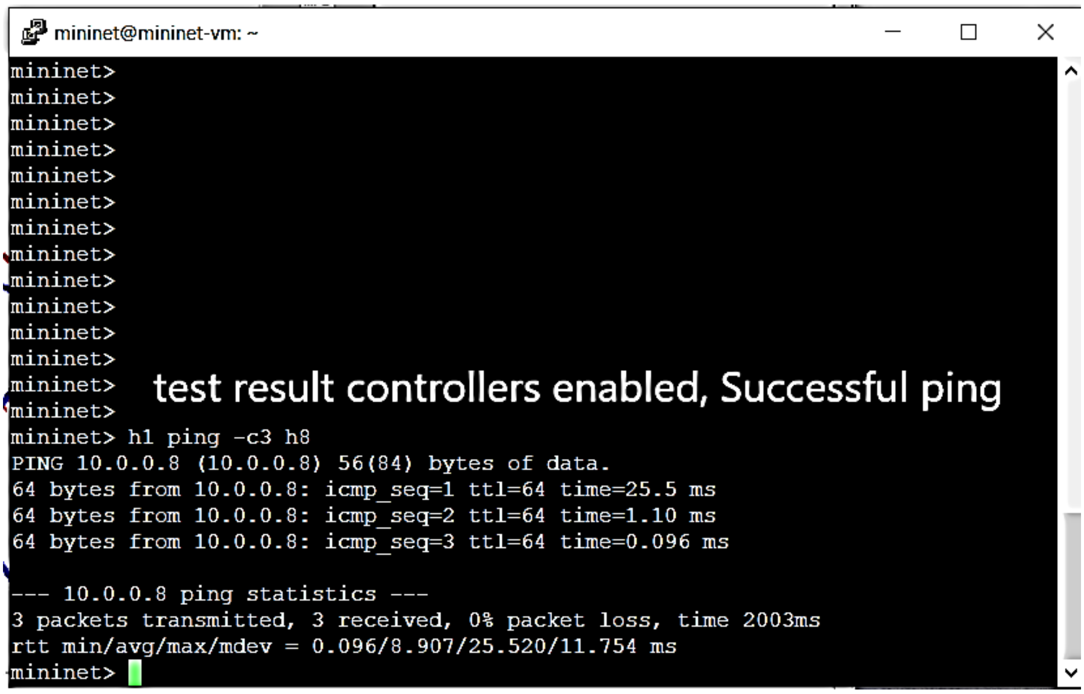
- ping test h1

- shutdown of link s1-s6

- ping test fail

Figure 4.27: CLI ping testing

The next step enabled the controllers and tested the connectivity while the s1-s6 link was down; results showed good and successful pings to its destination in Figure 4.28.



```
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> h1 ping -c3 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=25.5 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.096 ms

--- 10.0.0.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.096/8.907/25.520/11.754 ms
mininet>
```

Figure 4.28: Successful packets received upon Enabled SDN Controllers

From the results, the controller can show the ability to re-route traffic, which is the brain of the network. The results show the automation factor of network routing upon failed links or the attacker’s ability to threaten the network.

4.2.3 Use Case Result 3

4.2.3.1 Test case 3a

From section 2.2.4, the objectives are to prove the fastest routing path and intelligence of the SDN controller, a simulation network using live nodes across the Vodacom network and configured capacities to test routing of policies from a *Smart Grid* pod to another. The Vodacom network uses a Cube topology for inter-regional routing. The Cube analysis shows a real-world example of a communication network interconnecting into the *Smart Grid*. The Cube scenario uses Pod 1 and Pod 2 to showcase the connectivity and latency requirements. For the practicality of the analysis, Service Providers configure their network not on latency or distance, but packet flow is based on a protocol Open Shortest Path First (OSPF). Considering the main cities of South Africa, each path shown below has a configured OSPF number. The rule of thumb is to sum the OSPF numbers, and by this, the sending router will decide the route of traffic on the lowest OSPF total. The entire Cube in Figure 4.29 needs to be managed and controlled, meaning that the Network Management Group's support monitors the network and operational team to perform network changes when a threshold is exceeded.

The cube analysis is used to compare the results and conclusion from Comparison analysis of *Software-Defined Network* and OSPF protocol using virtual media (Nugroho, 2017, p. 110). The defined paper identifies the effectiveness of SDN and measures its performance on an OSPF network. Whiles the OSPF network calculates the costing and chooses the shortest path, based on cost configured on the network, which considers latency only on a planning level. The Cube analysis and results from the paper prove the SDN Controller's ability to make smart routing decisions in freedom of automation.

Table 4.2 shows the traffic routing paths when data is sent from Pod 1 Vlan 1 to Pod 2 Vlan 2; the data contains operator *Smart Grid* components policies. Each route shows the total cost. For example, from Table 4.2, Route 3 calculates the least 'Total Cost' amount and will be the routed path from Pod 1 Vlan 1 to destination Pod 2 Vlan 1.

Table 4.2: Traffic Routing of Cube

Route:	**Path**	**Total Cost**
Route 1	JFL-PRS-DMO-DNE	300
Route 2	JFL-DMO-DNE	450
Route 3	JFL-CFO-CTE-DNE	225
Route 4	JFL-MTP-CTE-DNE	300
Route 5	JFL-MTP-DNE	300

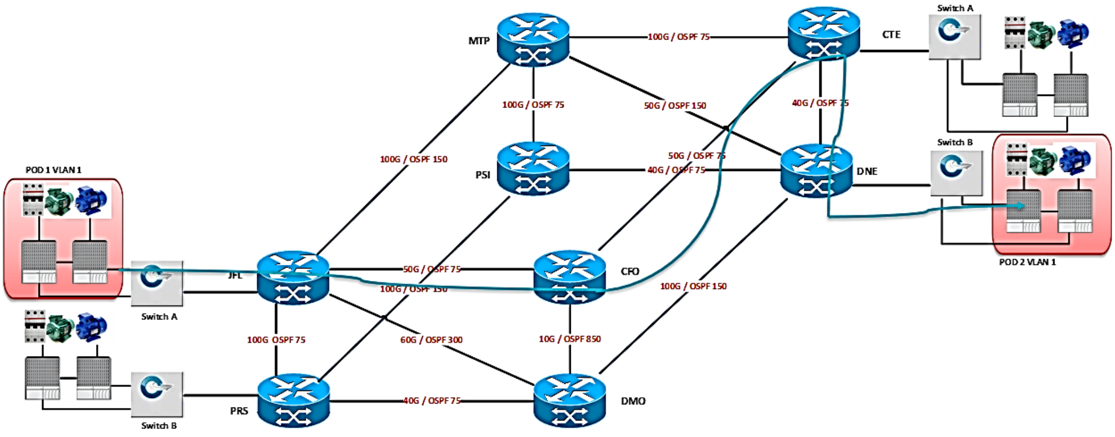


Figure 4.29: OSPF Cube of South Africa

Figure 4.30 shows SDN Controller 1 and 2 positioned between Switch A and Switch B; the SDN Controller contains routing tables and understands the network's capacities and utilization of paths. The SDN controller identified the shortest path upon configured capacities.

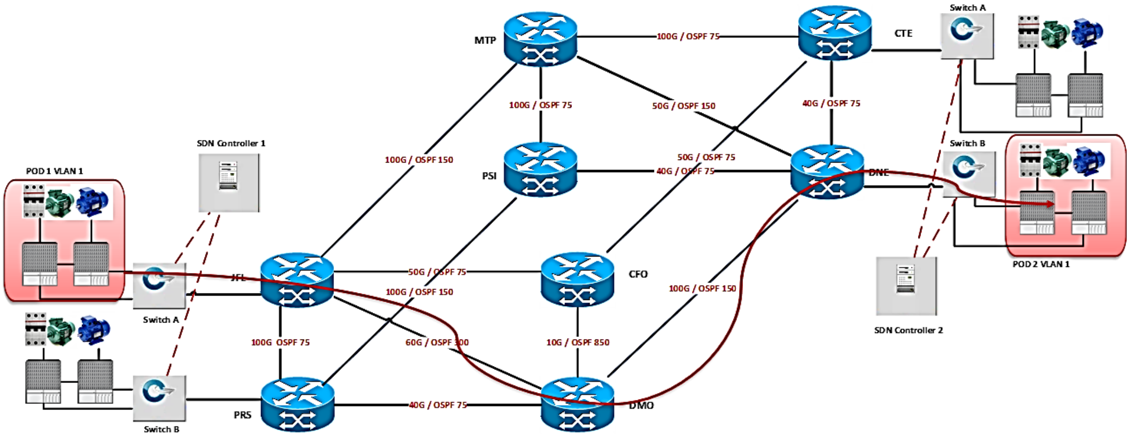


Figure 4.30: SDN applied to Capacity Cube of South Africa

When SDN is applied to the Cube, the chosen routing path is more favourable in using configured capacities and finding the shortest route to its destination. Furthermore, the SDN controller manages the cube network and does not rely on human intervention. The conclusion from this analysis stands in agreement with the results and conclusion from Comparison analysis of *Software-Defined Network* and OSPF protocol using virtual media (Nugroho, 2017, p. 110). Also, the test performed from the paper on the seven devices shows the lowest latency on an SDN network. Thus, both analyses proved the SDN concept's ability to provide smart control for network traffic routing on intelligence and speed.

Also, the study conducted by Rehmani *et al.* (2018, p. 4) shows the effectiveness of the SDN controller based on two scenarios, a centralized approach and a random approach. The centralized approach has algorithms set up to direct traffic flow upon a sudden attack of links, whiles the random approach, to see how the network will react to a sudden link attack. The random approach showcases a worst-case scenario. The traditional network setup will rely on a traffic engineering protocol to re-route traffic, taking a higher latency or highly congested paths. Thus, the role of the SDN controller is built with intelligence to understand a network on capacities and real-time operational states.

Furthermore, Ma *et al.* (2020, p. 872) simulate a 5G IP+Optical power communication topology to show the convergence speed based on bandwidth, delay, and risk equilibrium. The study provides a detailed account of the best power routes on a mesh power *Smart Grid*. The SDN controller can route based on IP+Optical power to its destination between a failure or planned site. Thus, providing an Always On power network. From Ma *et al.* (2020, p. 873), the AC3 algorithms achieve a further average target rate of 2.8% for network delay.

4.2.3.2 Test case 3b

A local area network was used for the test case to show recovery time after a network switch when Down and to compare to an SDN network when the same type of failure occurs. The setup consisted of two routers and two switches connected to show the primary and secondary sides. The test case used a shutdown command on SW1 of Figure 4.31; the result recorded showed in Table 4.3, the SDN network contains a recovery time below 10 ms.

Each SDN controller contains routing tables and decisions of best routing paths considering capacity resources and down paths. Kurtz *et al*. (2017, p. 4) designed a testbed to measure the performance of failure detection and delay on the recovery process. The testbed setup shows the connection from the SDN controller to the Bay switch and Substation switch. The route tables are updated on the state of the switch ports. Compared to this paper's proposed architecture and testbed setup, the SDN controller is connected to switches in domains or pods depending on their application. The design will prove affected based on the results. The result from Kurtz et al. (2017, p.6) shows a decrease in the SDN controller's recovery time shown in Figure 4.31, which points to the intelligence of the SDN controller's ability to make decisions faster based on the network services. Network services are *Smart Grid* applications of operations. The traditional network of the local area network depends on Hot Standby Router Protocol to failover traffic to the next node. The time taken will depend on the expiry time of each node to failover. The SDN controller can think ahead and calculate the best efficient routes for smooth failover.

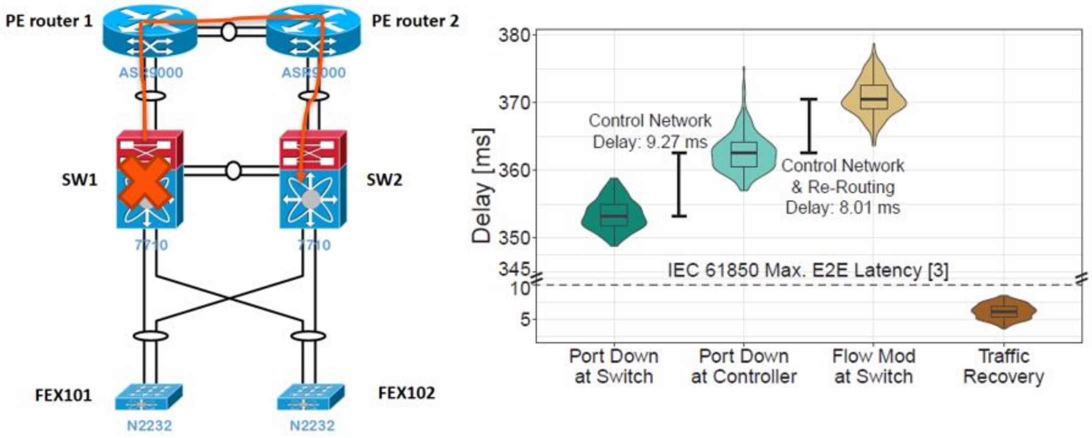


Figure 4.31: Traffic Recovery delay upon failed node
(Kurtz, 2017)

Al-Rubaye *et al*. (2019, p. 274) propose a network upgrading scenario to increase the number of SDN switches for dual redundancy. The further increase in infrastructure will carry a higher cost that can be disagreed to solving the traditional network resource issues. It can be stated the more nodes or switches, the greater the number of hops, which indicates a higher latency or higher traffic recovery rate. The other purpose of the SDN controller is to build a network with its current resources and plan for smart resiliency without exhausting the budget or having a duplicated network.

Table 4.3: Test result showing recovery times

Local Area Network	**SDN network**
352 ms	<10 ms

From the test of each use case that assessed the capability of the SDN controller within the SG domains, the results proved the ability to control the *Smart Grid* components and the centralized management of all devices on a real-time event. In addition, the results have shown positive in network speed, availability, and security of executing policies. When considering (a) the scalability of the SDN controller to other *Smart Grid* domains .

(a) The concern arises of scalability and integrating a mixture of different provider's nodes, for example, when a network contains Cisco devices and Nokia devices. The case calls for adapter development for Nokia to Speak to Cisco. From understanding adapter development, each service will require an adapter. The complexity builds up when introducing a further 3rd node provider, adding more adaptors on layer 2 or 3 ISO environment. The solution of the SDN concept overcomes the scalability issue by introducing a hierarchical controller that binds together node providers—and is further used for policy enforcement, policy application, hierarchical quota, and hierarchical observability.

4.3 Summary

Chapter 4 presents the simulation and testing to prove the concept of SDN and apply the control ability to the *Smart Grid* environment. The integrated architecture is presented. The design consisted of three phases to showcase the methods and approaches used to prove and test the *Software-Defined Network* (SDN) in control of *Smart Grid* (SG) components and bring intelligence between a communication network and *Smart Grid*. The chapter aims to present the following:

- Control ability of the SDN controller,
- Management of the SDN+SG network by the SDN controller,
- Routing performance compared to a traditional network, and
- Recover time after a network node failure.

CHAPTER 5

CONCLUSION

5.1 Summary of work

In this thesis, a comprehensive study was conducted to evaluate the open-source *Software-Defined Network* and *Smart Grid*. The use cases were set up to be easily reproduced.

The agreed approach selected three design phases in **Chapter 4** to prove the concept of a *Software-Defined Network* within a *Smart Grid* environment. Each design phase built a use case to show if the SDN controller improves centralized control, improves Smart Grid control, and improves network management from Figures **4:13, 4:14, 4:21,** and **4:22**. The approach was not to re-design a new architecture but rather build on previous researchers and real-world scoping using industry knowledge and collaboration efforts to re-think and re-analyze by **Figure 2.4**.

Smart Grid is a phased deployment and not a one-step approach, which benefits energy organizations, the economy, and the production industries.

The traditional communication network is built on proprietary brands and has no relation to the *Smart Grid*, the need for reduced latency, speed, mobility, and massive machine type signalling. It is becoming the fundamental platform needed as the 5G and 6G network prepares the way with the aforementioned qualities. Another area of concern is the need for remote connectivity into routers, switches belonging to the communication network and pods, *Smart Grid* devices belonging to the *Smart Grid*.

The main question arose how to control *Smart Grid* devices from the communication network? Section **4.1.3**. How to integrate the communication network of a Service Provider and *Smart Grid* environment? Section **4.1.3**. How to manage, automate and maintain the integrated network? Section **4.2.2/3.** How to build security and protect customer data and vendor assets? Section **2.4.1.1.**

The proposed framework is scalable, secure, and flexible shown in **Figure 3.4**. The approach builds a new thinking pattern, allowing the control ability to sit within the SDN controller and forwarding plane hosting layer two and layer one operational devices. The SDN framework can be extended to multi-cloud provider networks using orchestration.

To achieve the above, the research focussed on critical components to test over use cases; the SDN controller introduces the centralization of both environments within the network. The SDN controller contains policies that are the operational parameters of

Smart Grid components. When a policy is called, it will read and execute the algorithm accordingly.

In this research, the SDN concept is applied to a Smart Home, an Automation Plant, and Electrical Distribution plant in sections **3.4.1, 3.4.2,** and **3.4.3**; each contains similar circuit breakers, relays, motors, etc. The research discussed details of the application of SDN within each environment. The aim ensured that each environment within the *Smart Grid* arena proposed a redefined integrated architecture connecting to the communication network and enabling remote control.

5.2 Methodology Applied

The steps included an analysis of the SDN network and applying the concept over the *Smart Grid* network; the complication arose on integrating both environments. My research proposed a Network Exposure layer that exposes the APIs of the *Smart Grid* environment. Each *Smart Grid* environment will be allocated an API naming convention and a VLAN. So that, each environment uses the shared network but belonging to its own, like SDN Smart Home cannot see what's happening in SDN Automation Plant, and verse versa. Also, the purpose of the research is based on centralized control; the SDN controller showed the ability to contain the devices within the flow tables.

Careful evaluation and network setup looking at an OSPF network in section **2.2.4** and in comparison to an SDN network, the benefits show great flexibility within a communication network and ability to route packets to a *Smart Grid* sector.

The successful integration in section **4.2** was performed using an open-source management tool to control and monitor the live environment from a Floodlight GUI and configure policies from the SDN controller via OpenFlow protocol. Test design phase three proved SDN's control and intellectual ability when applied to the SG component. The test cases proved the ability to control and manage the *Smart Grid* network components. The results showed the management of traffic engineering when a link failure occurs.

It was proposed in Al-Rubaye et al. (2019) a network upgrading scenario to increase the number of SDN switches for dual redundancy. The further increase in infrastructure will carry a higher cost that can be disagreed to solving the traditional network resource issues. It can be stated the more nodes or switches, the greater the number of hops, which indicates a higher latency or higher traffic recovery rate. Another purpose of the SDN controller is to build a network with its current resources and plan for smart resiliency without exhausting the budget or having a duplicated network.

Finally, MEC's deployment across a communication network and *Smart Grid* network will enable a further scalable, efficient, and resilient *Smart Grid* network. To provide network application slices for each service of SDN Smart Home, SDN Automation Plant, and SDN Electrical Distribution system.

The research discussed the concept of *Software-Defined Network*ing. It made the comparison between a traditional network in Chapter **2** by analyzing the current traditional network, the protocols used, and base architecture, providing the areas that need improvement and stating how SDN can be used efficiently.

Providing an analysis of SDN Security and addressing security issues faced on the power grid. Research into the current security solutions, the most significant security challenges, and security threats in networks investigate SDN security at each layer and produce practical solutions on an SDN secure network.

In conjunction with section **1.4**, Chapter **3** highlighted the advantages and challenges of implementing SDN, explained the OpenFlow protocol in each *Smart Grid* pod and listed the advantages, challenges, and solutions by conducting detailed research on SDN. Defining the roles of each technology and creating a standard architecture, defining the protocols to be used and policies. Building the concept by using models and logic to show its effectiveness.

To outline how SDN is implemented in a Smart Home, Automation Plant, and Electrical Distribution environment—providing the features and functionality with a designed model—explaining each integrated environment mentioned, Listing the features and functionality of each design model. Chapter **3** concludes with migration strategies.

Chapter **4** shows the software tools that are used to perform simulation and selection of SDN controller by use case—defining how SDN is applied/integrated to the *Smart Grid* environment. An explanation of packet movement and define packet flow, utilizing software tools by integration and configuration to demonstrate and prove the intelligence and control of SDN applied to the *Smart Grid* by use cases.

5.3 Outcome of Results

Chapter 3 presented the methodology used to integrate both communication networks and Smart Grid, adding intelligence by centralized SDN controllers; the design prepared the way by use case to prove the control and intelligent abilities. The successful results in section **4.2** showed the control of the Smart Grid component via the SDN controller. And to acknowledge the management of SDN within the network.

The outcome of the thesis stated in section **1.4.1** has been successfully fulfilled on:

- In proving the control of SDN into *Smart Grid* networks, successfully performed in section **4.1.3** and based on the testbed use case.
- To identify software tools that can be used to perform simulation by **Figure 4.7**.
- To select the most appropriate SDN controller for this *Smart Grid* implementation, provided in section **3.3**.
- To define how SDN can be applied/integrated to the *Smart Grid* environment provided in section **3.2**.
- In defining each technology's roles and creating a standard architecture, presented in section **2.2**.
- In defining the protocols to be used and policies, they are presented in sections **3.3.1, 3.3.2, 3.3.3, & 3.3.4.**
- To build the concept using models and logic to prove the use case, as shown in section **4.2**.
- To outline how to implement SDN in a Smart Home, Automation Plant, and Electrical Distribution environment—providing the features and functionality with a design model—presented in sections **3.4.1, 3.4.2, & 3.4.3** showcasing the integration per environment.
    - In explaining each integrated environment mentioned,
    - Listing the features and functionality of each design model.
    - In explaining the packet movement and defining the application slice per environment.

The successful research and testbed can be scaled out and proved the flexible integration of SDN into the SG environment, with a high level of intelligence, management, and automation driving the technology markets into the new SDN thinking paradigm.

Future Recommendations are presented in **Appendix 4**.

REFERENCES

Abdullah, N.A.S., Noor, N.L.M. & Ibrahim, E.M.N. (2013). Resilient organization: Modelling the capacity for resilience. *2013 International Conference on Research and Innovation in Information Systems (ICRIIS)*, pp. 319-324. doi: 10.1109/ICRIIS.2013.6716729.

Al-Rubaye, S., Kadhum, E., Ni, Q,. et al. (2019). Industrial Internet of Things Driven by SDN Platform for *Smart Grid* Resiliency. in *IEEE* Internet of Things Journal, vol. 6, no. 1, pp. 267-277.

An, Braeken. & Pardeep, Kumar. (2020). Secure and Efficient Privacy-preserving Scheme in Connected *Smart Grid* Networks. *IoT Security: Advances in Authentication*, pp.247-264, doi: 10.1002/9781119527978.ch13.

Andreoli, C., Blefari-Melazzi, N., Listanti, M., et al. (1996). Mobility management in IP networks providing real-time services. *Proceedings of ICUPC* - 5th International Conference on Universal Personal Communications, pp. 774-777.

Argibay-Losada, P.J., Yoshida, Y., Maruta, A. et al. (2015). Performance of fixed-length, variable-capacity packets in optical packet-switching networks. in *Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 609-617. doi: 10.1364/JOCN.7.000609.

Bachman, F., Bass, L., Brown, B,. et al. (2009). Advanced Security Acceleration Project (ASAP) – *Smart Grid*. Security Profile for Advanced Metering Infrastructure.

Bansal, P., & Singh, A. (2016). Smart metering in *Smart Grid* framework: A review. *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 174-176.

Barai, G.R., Krishnan, S., & Venkatesh, B. (2015). Smart metering and functionalities of smart meters in *Smart Grid* - a review. *2015 IEEE Electrical Power and Energy Conference (EPEC)*, pp. 138-145. doi: 10.1109/EPEC.2015.7379940.

Bastos, J. (2019). Forecasting the capacity of mobile networks. *Telecommunication Systems*., vol. 72, pp. 1-12. doi: 10.1007/s11235-019-00556-w.

Berestizshevsky, K., Even, G., Fais, Y. et al. (2017). SDNoC: Software de_ned network on a chip. *Microprocessors Microsyst.*, vol. 50, pp. 138-153.

Bernstein, G.M. (2006). IP Bandwidth on Demand and Traffic Engineering via Multi-Layer Transport Networks. *2006 IEEE First International Workshop on Bandwidth on Demand,* pp. 44-48.

Bhardwaj, S., Panda, S.N., & Datta, P. (2020). Layer-Based Attacks in the Ternary Planes of Software-Defined Networking. *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 292-295, doi: 10.1109/WIECON-ECE52138.2020.9398012.

Bossart, S.J. & Bean J. E. (2011). Metrics and benefits analysis and challenges for *Smart Grid* field projects*. IEEE 2011 EnergyTech*, pp 1-5.

Bozakov, Z. *et al.* (2017). A NEAT framework for enhanced end-host integration in SDN environments. *2017 IEEE Conference on Network Function Virtualization and Software-Defined Networks (NFV-SDN)*, pp. 1-7. doi: 10.1109/NFV-SDN.2017.8169828.

Brown, B., Singletary, B., Willke, B,. et al. (2008). Advanced Security Acceleration Project (ASAP) – *Smart Grid*. Advanced Metering Infrastructure(AMI) System Security Requirements.

Choi, Y., Kim, J. & Park, N. (2016). Revolutionary direction for 5G mobile core network architecture. *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 992-996. doi: 10.1109/ICTC.2016.7763350.

Chren, S., Rossi, B. & Pitner, T. (2016). *Smart Grid*s deployments within EU projects: The role of smart meters. *2016 Smart Cities Symposium Prague (SCSP)*, pp. 1-5, doi: 10.1109/SCSP.2016.7501033.

CIP-002. Cyber Security-Critical Cyber Asset Identification. (2009). North American Electric Reliablity Corporation(NERC). pp. 003-009.

Cloud Computing – Information Assurance Framework. (2009). European Network and Information Security Agency (ENISA).

Colak, A., Ayaz, M.S., & Ahmed, K. (2021). Long Term Benefits of Advanced Communication Techniques in Smart Grids," *2021 9th International Conference on Smart Grid (icSmartGrid)*, pp. 283-288, doi: 10.1109/icSmartGrid52357.2021.9551259.

Comer, D., & Rastegarnia, A. (2019). Externalization of Packet Processing in Software Defined Networking, in *IEEE Networking Letters*, vol. 1, no. 3, pp. 124-127, doi: 10.1109/LNET.2019.2918155.

Cong, L., Wen, W., & Zhiying, W. (2014). A con_gurable, programmable and software-de_ned network on chip. in *Proc. IEEE Workshop Adv. Res. Technol. Ind. Appl. (WARTIA)*. pp. 813-816.
CPNI. Good Practice Guide, Process Control and SCADA Security. Centre for the Protection of National Infrastructure (CPNI):2005.

D’Lima, A., Peirce, B., Louie, B. et al. (2018). UtilityAMI. Home Area Network System Requirements Specification.

D’Souza, D., & Marinos, N. (2019). U.S. Government Accountability Office (U.S. GAO). Cyber security for Critical Infrastructure Protection.

da Silva, E.G., da Silva, A.S., Wickboldt, J.A. et al. (2016). A one-class nids for sdn based scada systems. in *Computer Software and Applications Conference (COMPSAC)*, 2016 IEEE 40th Annual, vol. 1. IEEE, pp. 303–312.

Dari, E.Y. & Essaaidi, M. (2015). An overview of *Smart Grid* cyber-security state of the art study. *3rd International Renewable and sustainable energy conference*, pp. 1-7.

Das S. (2012). Unified control architecture for packet and circuit network convergence. PhD Thesis, Standford University.

de Puga, J.S., Salvador, C.E.P. & Pellicer, A.B. (2019). Architecture and Use Case for an IoT Deployment with SDN at the Edge and Dual Physical and Virtual Gateway. *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1-6. doi: 10.1109/ICCCN.2019.8847095.

Denyer, D., & Tranfield, D. (2009). Producing a systematic review. The Sage handbook of organizational research methods. London: *Sage Publications Ltd*, pp. 671-689.

Desai, M., & Nandagopal, T. (2010). Coping with Link Failures in Centralized Control Plane Architectures. *Proceedings of 2nd International Conference on Communication Systems and Networks* (COMSNETS), pp. 1–10.

Dhawan, M., Roddar, R., Mahajan, K., et al. (2015). SPHINX: Detecting Security Attacks in Software-Defined Networks*. 22nd Annual Network and Distributed System Security Symposium*, pp. 8-11.

DHS, August 2008. 36 NIST. System Protection Profile - Industrial Control Systems. National Institute of Standards and Technology (NIST): 2004. pp. 1.

DHS. (2009). Cyber Security Procurement Language for Control Systems. U.S. Department of Homeland Security.

DHS. (2011). Catalog of Control Systems Security: Recommendations for Standards Developers. U.S. Department of Homeland Security.

Dinh, P.T. & Park, M. (2021). BDF-SDN: A Big Data Framework for DDoS Attack Detection in Large-Scale SDN-Based Cloud. *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, pp. 1-8. doi: 10.1109/DSC49826.2021.9346269.

DOE. 21 steps to Improve Cyber Security of SCADA Networks. Office of Energy Assurance. U.S. Department of Energy. 2002.

Dong, X., Lin, H., Tan, R. et al. (2015). *Software-Defined Network*ing for *Smart Grid* resilience: Opportunities and challenges. in Proceedings of the 1st ACM Workshop on Cyber-Physical System Security. *ACM*, pp. 61–68.

Du, J., Lu, Y., Li, C. et al. (2021). Model Quality Evaluation of Advanced Distribution Management System Based on Smart Grid Architecture Model, *2021 China International Conference on Electricity Distribution (CICED)*. 688-691, doi: 10.1109/CICED50259.2021.9556844.

Duan, J., Zhao, B., & Guo, S. (2020). The Design and Implementation of Smart Grid SOC Platform," *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, pp. 264-268, doi: 10.1109/ICIBA50161.2020.9277373.

Ellinidou, S., Sharma, G., Rigas, T. et al. (2018). SSPSoC: A secure SDN-based protocol over MPSoC. *Secur. Commun. Netw.*, vol. 2019, pp. 1-11.

Ford, V., Siraj, A. & Eberle, W. (2014). *Smart Grid* energy fraud detection using artificial neural networks," *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*, pp. 1-6, doi: 10.1109/CIASG.2014.7011557.

Fox, G., Kamburugamuve, S. & Hartman, R.D. (2012). Architecture and measured characteristics of a cloud based Internet of Things. in Proc. *IEEE* Int. Conf. CTS, pp. 6–12.

Froom, R., Erum, F., & Balaji, S. (2010). *Implementing Cisco IP switched networks (SWITCH) foundation learning guide*. Indianapolis. IN. Cisco Press.

Gautam, Y., Gautam, B.P. & Sato, K. (2020). Experimental Security Analysis of SDN Network by Using Packet Sniffing and Spoofing Technique on POX and Ryu Controller. *2020 International Conference on Networking and Network Applications (NaNA)*, pp. 394-399, doi: 10.1109/NaNA51271.2020.00073.

GB/T 22239-2008. (2008). Information security technology – Baseline for classified protection of information system. National Standard of the People's Republic of China.

Goran, N. (2010). Cyber security and power system communication-essential parts of a *Smart Grid* infrastructure. *IEEE* Transaction on power delivery, vol. 25, no. 3, pp. 1501-1507.

Graur, F. (2017). Dynamic network configuration in the Internet of Things. 2017 5th International Symposium on Digital Forensic and Security (ISDFS), pp. 1-4.

Gu, Y., Li, D., & Yu, J. (2020). Im-OFDP: An Improved OpenFlow-based Topology Discovery Protocol for Software Defined Network. *2020 IFIP Networking Conference (Networking)*, pp. 628-630.

Haeberlen, T., Dupre, L., Catteddu, D,. et al. (2012). European Network and Information Security Agency (ENISA). Cloud Computing – Benefits, risks and recommendations for information security.

Hong, S., Xu, L., Wang, H., et al. (2015). Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures. *Proceedings of 2015 Annual Network and Distributed System Security Symposium* (NDSS'15).

Hu, Q., Wu, C., Zhao, X. et al. (2018). Vehicular Multi-Access Edge Computing With Licensed Sub-6 GHz, IEEE 802.11p and mmWave. *IEEE Access*, vol. 6, pp. 1995-2004.

IEC. IEC 62351 1-8, Power System Control and Associated Communications – Data and Communication Security.

IEEE. IEEE 1686-2007, IEEE Standard for Substation Intelligent Electronic Devices(IED) Cyber Security Capabilities.

ISO/IEC. (2013). Information technology – Security techniques – Information security management systems.

Jansen, W., & Scarfone., K. (2008). Guidelines on Cell Phone and PDA Security. National Institute of Standards and Technology(NIST).

Jaumard, B., Pouya, H., Fahim, R., et al. (2016). Planning network migration. *2016 IEEE International* Conference on Communications (ICC), pp. 1-6.

Jianming, L., Bingzhen, Z., Jiye, W., et al. (2010). Application of power line communication in smart power Consumption. *ISPLC2010*, pp. 303-307.

Jin-Lun, C., Chun, H., Zhao-Xin, Z., et al. (2012). *Smart Grid* oriented smart substation characteristed analysis. *IEEE* PES Innovation *Smart Grid* Technologies, pp. 1-4.

Karmakar, K.K., Varadharajan, V., Tupakula, U. et al. (2020).Towards a Dynamic Policy Enhanced Integrated Security Architecture for SDN Infrastructure. *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. pp. 1-9, doi: 10.1109/NOMS47738.2020.9110405.

Kempf, J., Bellagamba, E., Kern, A., et al. (2011). Scalable fault management for OpenFlow in *IEEE* Int. Conf. on Communications (ICC), pp. 6606–6610.

Kim, H., Ben-Othman, J., Mokdad, L. (2020). Research Challenges and Security Threats to AI-Driven 5G Virtual Emotion Applications Using Autonomous Vehicles, Drones, and Smart Devices. in *IEEE Network*, vol. 34, no. 6, pp. 288-294, doi: 10.1109/MNET.011.2000245.

King, D., et al. (2016). The Software-Defined Transport Network: Fundamentals, findings and futures. *18th International Conference on Transparent Optical Networks (ICTON)*, pp. 1-4.

Kissel, R.L., Stine, K.M., & Scholl, M.A. et al. (2008). Security Considerations in the Information System Development Life Cycle. National Institute of Standards and Technology (NIST).

Knapp, E.D. & Samani, R. (2013). *Applied cyber security and the Smart Grid: implementing security controls into the modern power infrastructure*. Syngress. Elsevier.

Koshiba, R., Shin, S. & Sebe, N. (1996). Relationship between decentralized and centralized designs in mixed sensitivity problem. *Proceedings of 35th IEEE Conference on Decision and Control*. pp. 7-8 vol.1, doi: 10.1109/CDC.1996.574236.

Kurose, J.F. & Ross. K.W. (2013). *Computer Networking*. Pearson Education Limited. Harlow, England.

Kurtz, F., Dorsch, N., Bektas, C. et al. (2017). Synchronized measurement concept for failure handling in software-defined *Smart Grid* communications. *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 1-6.

Lara, A., & Ramamurthy, B. (2016). OpenSec: Policy-Based Security Using Software-Defined Networking. in *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 30-42. doi: 10.1109/TNSM.2016.2517407.

Li, X., Huang, Q. & Wu, D. (2017). Distributed Large-Scale Co-Simulation for IoT-Aided *Smart Grid* Control. *IEEE Access*, vol. 5, pp. 19951-19960, 2017, doi: 10.1109/ACCESS.2017.2753463.

Liyanage, Madhusanka., Gurtov, Andrei. & Ylianttila, Mika. (2015). Software Defined Networking Concepts. *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*, pp.21-44, doi: 10.1002/9781118900253.ch3.

Lu, G. & Song, W.Z. (2010). SmartGridLab: A laboratory-based *Smart Grid* testbed. *Smart Grid* Communication. *2010 First IEEE International Conference*, pp. 143-148.

Ma, Q. et al. (2020). Co-Allocation of Service Routing in SDN-driven 5G IP+Optical *Smart Grid* Communication Networks based on Deep Reinforcement Learning. *2020 International Wireless Communications and Mobile Computing (IWCMC),* pp. 868-873.

Mangos, W. (2008). Wireless Standards. International Standards of Auditing(ISA).

Mau, D.O., Taleb, T. & Chen, M. (2015). MM3C: Multi-source mobile streaming in cache-enabled content-centric networks. in Proc. *IEEE* Globecom, pp. 1–6.

Minimum Security Requirements for Federal Information and Information Systems. (2006). National Institute of Standards and Technology (NIST).

Mogull, R., Arlen, J., Gilbert, F,. et al. (2021). Cloud Security Alliance (CSA). Security Guidance for Critical Areas of Focus in Cloud Computing.

Mollah, M. B., Zhao., J. Niyato, D. *et al.* (2021). Blockchain for Future Smart Grid: A Comprehensive Survey," in *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 18-43, doi: 10.1109/JIOT.2020.2993601.

Mrabet, Z., Kaabouch, N., Elghazi, H., et al. (2018). A study on cyber security of *Smart Grid* on public networks. *2013 IEEE Green Technologies Conference (GreenTech)*, pp. 301-308.

Narantuya, J. *et al.* (2019). SDN-Based IP Shuffling Moving Target Defense with Multiple SDN Controllers. *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S)*, pp. 15-16. doi: 10.1109/DSN-S.2019.00013.

National Institute of Standards and Technology(NIST). Guidelines for *Smart Grid* Cyber Security, *Smart Grid* Cyber Security Strategy, Architecture, and High-Level Requirements: 2010. pp. 1.

National Institute of Standards and Technology(NIST). Recommended Security Controls for Federal Information Systems (including those for Bulk Power System)(NIST SP 800-53). 2010.

Nazirov, K.B., Ganiev, Z.S., Dzhuraev, S.D. et al. (2021). Experimental Evaluation and Analysis of Electric Power Quality in Electric Networks Municipal-Households. *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp. 1491-1494, doi: 10.1109/ElConRus51938.2021.9396351.

Nikolai. (2014). MobileGuardian: A security policy enforcement framework for mobile devices. *2014 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 197-202. doi: 10.1109/CTS.2014.6867564.

NIST Special Publication 1108. NIST framework and roadmap for *Smart Grid* Interoperability standards: 2010.

Nugroho, A.S., Dian Safitri, Y. & Setyawan, T.A. (2017). Comparison analysis of *Software-Defined Network* and OSPF protocol using virtual media. *2017 IEEE International Conference on Communication, Networks, and Satellite (Comnetsat)*, pp. 106-111.

Orsini, G., Bade, D. & Lamersdorf, W. (2015). Computing at the mobile edge: Designing elastic android applications for computation offloading. in Proc. *IEEE* 8th IFIP Wireless Mobile Netw. Conf. (WMNC), pp. 112–119.

Padma, V., & Yogesh, P. (2015). Proactive failure recovery in OpenFlow based *Software-Defined Network*s. *3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1-6.

Porwal, M.K., Yadav, K. & Charhate, S.V. (2008). Traffic Analysis of MPLS and Non MPLS Network including MPLS Signaling Protocols and Traffic Distribution in OSPF and MPLS. *2008 First International Conference on Emerging Trends in Engineering and Technology*, pp. 187-192. doi: 10.1109/ICETET.2008.58.

Prasad S, Koll D, Fu X. On the Security of *Software-Defined Network*s," *2015 Fourth European Workshop on Software-Defined Networks*: 2015. pp. 105-106.

Qi, K. (2020). Computer Network Information Security Analysis and Management Research Based on Improved Wavelet Neural Network. *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*. pp. 129-132, doi: 10.1109/TOCS50858.2020.9339745.

Ramakrishna, K. S. & Arul, D. (2018). Challenges and issues of *Smart Grid* implementation: A case of Indian scenario*. Journal of Electrical Systems and Information Technology*, no. 3.

Rawat, D.B. & Bajracharya, C. (2015). Cyber security for *Smart Grid* systems: Status, challenges and perspectives. *SoutheastCon, pp*. 1-6.

Rehmani, M.H., Akhtar, H., Davy, A. et al. (2018). Achieving Resilience in SDN-Based *Smart Grid*: A Multi-Armed Bandit Approach. *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 366-371.

Rehmani, M.H, Davy, A, Jennings, B. et al. (2019). Software Defined Networks-Based Smart Grid Communication: A Comprehensive Survey. in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2637-2670, thirdquarter, doi: 10.1109/COMST.2019.2908266.

Riccardi, E. P., Gunning, Ó. G., de Dios, M. et al. (2018). An Operator view on the Introduction of White Boxes into Optical Networks. *Journal of Lightwave Technology*, vol. 36, no.15, pp. 3062-3072.

Ruaro, M., Caimi, L.L., & Moraes, F.G. (2020). A Systemic and Secure SDN Framework for NoC-Based Many-Cores. in *IEEE Access*, vol. 8, pp. 105997-106008.

Ruaro, M., Medina, H.M., Amorynd. A.M. et al. (2018). Software-de_ned networking architecture for NoC-based manycores, in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*. pp. 385-390.

Ruepp, S., Buron, J., Andriolli, N. et al. (2008). Nodal Stub-Release in All-Optical Networks. in *IEEE* Communications Letters, vol. 12, pp. 47-49.

Rui, Z., Kecheng, L., Xiaoming, L. et al (2021). The Research on Multi-Dimensional Evaluation Index System of Distribution Network Loss. *2021 China International Conference on Electricity Distribution (CICED)*, 2021, pp. 747-751, doi: 10.1109/CICED50259.2021.9556681.

Sachs, G. (2010). A principle based system architecture fame work applied for defining, modelling & designing next generation *Smart Grid* systems. *Massachusetts Institute of Technology*.

Sahoo, K.S., Sahoo, B. & Panda, A. (2015). A secured SDN framework for IoT. *2015 International Conference on Man and Machine Interfacing (MAMI)*, pp. 1-4. doi: 10.1109/MAMI.2015.7456584.

Sandoval-Arechiga, R., Vazquez-Avila, J.L., Parra-Michel, L. et al. (2015). Shifting the networkon-chip paradigm towards a software de_ned network architecture," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, pp. 869_870.

Scionti, A., Mazumdar, S., & Portero, A. (2018). Towards a scalable software de_ned network-on-chip for next generation cloud," *Sensors*, vol. 18, no. 7, pp. 1-24.
Security Guidelines for the Electricity Sector. (2019). Vulnerability and Assessment. North American Electric Reliability Corporation (NERC).

Sgambelluri, A., Giorgetti, A., Cugini, F., et al. (2013). OpenFlow-Based Segment Protection in Ethernet Networks*. Journal of Optical Communications and Networking*, no. 5, pp. 1066 -1075.

Shaikh, M.Z. & Darekar, S.H. (2018). Performance Analysis of Various Open Flow Controllers by Performing Scalability Experiment on Software Defined Networks. *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, pp. 783-787, doi: 10.1109/ICICT43934.2018.9034343.

Shapsough, S., Qatan, F. & Aburukba, R. (2015). *Smart Grid* cyber security: challenges and solutions. *2015 International conference on Smart Grid and Clean energy Technologies*(ICSGCE), pp. 170-175.

Sharma, S., Staessens, D., Colle, D., et al. (2010). Enabling Fast Failure Recovery in OpenFlow Networks. *Proceedings of 8th International Workshop on the Design of Reliable Communication Networks* (DRCN), pp. 164–171.

Sharma, S., Staessens, D., Colle, D., et al. (2011). *Software-Defined Network*ing: Meeting Carrier Grade Requirements. *Proceedings of 18th IEEE Workshop on Local Metropolitan Area Networks* (LANMAN), pp. 1-6.

Shu, F., Chen, S., Li, F. et al. (2020). Research and implementation of network attack and defense countermeasure technology based on artificial intelligence technology. *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 475-478, doi: 10.1109/ITOEC49072.2020.9141751.

Sibylle, S., & Dave, H. (2017). *Software-Defined Network*ing architecture standardization. *Computer Standards & Interfaces*, pp. 197-202.

Simmons, G., Armstrong, G.A. & Durkin, M.G. (2011). An exploration of small business website optimization: Enablers, influencers and an assessment approach. Int. *Small Bus*, vol. 29, no. 5, pp. 534–561.

Singhal, A. & Saxena, R. (2012). Software models for *Smart Grid*. *2012 First International Workshop on Software Engineering Challenges for the Smart Grid (SE-SmartGrids), pp*. 42-45.

Srikanth, A., Varalakshmi, P., Somasundaram, V., et al. (2018). Congestion Control Mechanism in *Software-Defined Network*ing by Traffic Rerouting. *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 55-58.

Standard of good practice for information security. (2020). Information Security Forum (ISF).

Stouffer, K., Falco, J. & Scarfone, K. (2007). Guide to Industrial Control Systems (ICS) Security Special Publication 800-82. Second public draft, National Institute of Standards and Technology.

Suartana, I.M., Anggraini, M.A.N. & Pramudita, A.Z. (2020). High Availability in Software-Defined Networking using Cluster Controller: A Simulation Approach. *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*, pp. 1-5, doi: 10.1109/ICVEE50212.2020.9243173.

Sun, X. & Ansari, N. (2016). EdgeIoT: Mobile edge computing for the Internet of Things. *IEEE* Commun, vol. 54, no. 12, pp. 22-19.

Sydney, A., Ochs, D., Scoglio, C., et al. (2014). Using GENI for experimental evaluation of *Software-Defined Network*ing in *Smart Grid*s. *Computer Networks*, pp. 5-16.

Takahashi, N., Tanaka, H. & Kawamura, R. (2015). Analysis of process assignment in multi-tier mobile cloud computing and application to edge accelerated web browsing, in Proc. 3rd IEEE Int. Conf. Mobile Cloud Comput. Services Eng. (MobileCloud), pp. 233–234.

Talarico, S., Makhijani, K. & Pillay-Esnault, P. (2016). Efficient service auto-discovery for next generation network slicing architecture. *2016 IEEE Conference on Network Function Virtualization and Software-Defined Networks (NFV-SDN)*, pp. 26-32.

Taleb, T., Dutta, S., Ksentini, A. et al. (2017). Mobile edge computing potential in making cities smarter. *IEEE Commun. Mag.*, vol. 3, no. 55, pp. 38–43.

Tao, J., Yuan, R., Liu, Q. et al. (2021). Research and Implementation of a Network Based on SDN and Multi Area OSPF Protocol. *2021 IEEE 9th International Conference on Information, Communication and Networks (ICICN)*, pp. 134-138, doi: 10.1109/ICICN52636.2021.9673836.

The International Society of Automation(ISA). ANSI/ISA–99 Security for Industrial Automation and Control Systems. Part2: Establishing a Manufacturing and Control Systems Security Program.

The International Society of Automation(ISA). ANSI/ISA–99.00.01–2007. Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models.

Theodorou, T., & Mamatas, L. (2017). CORAL-SDN: A *Software-Defined Network*ing solution for the Internet of Things. *2017 IEEE Conference on Network Function Virtualization and Software-Defined Networks (NFV-SDN)*, pp. 1-2.

Valencia-Calvo, J., Olivar-Tost, G., & García-Ortega, G. (2020). Model and Simulation of a Renewable Energy Market: Integration of Renewable Energy Sources with the Conventional Generation System. *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1-7, doi: 10.23919/CISTI49556.2020.9140987.

Velloso, A., Ruaro, M., Jantsch, N. et al. (2019). Distributed SDN architecture for NoC-based many-core SoCs. in *Proc. 13th IEEE/ACM Int. Symp. Netw.-on-Chip*. pp. 1_8.
Vineetha, C.P. & C. A. Babu, "*Smart Grid* challenges, issues and solutions," *2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, 2014, pp. 1-4, doi: 10.1109/IGBSG.2014.6835208.

Wang, Y., Zhang, B., Lin, W., et al. (2011). *Smart Grid* information security - a research on standards. *2011 International Conference on Advanced Power System Automation and Protection*, pp. 1188-1194.

Wen, Tan. & Bair, X. (2012). Application research base on system engineering for analyzing *Smart Grid* standards. *IEEE PES Innovative Smart Grid Technologies.* Tianjin, pp. 1-3.

Wiatr, P., Chen, J., Monti, P., et al. (2015). Energy efficiency versus reliability performance in optical backbone networks. IEEE/OSA Journal of Optical Communications and Networking, no. 5, pp. A482-A491

Xu, H. (2016). Finite horizon optimal control and communication co-design for uncertain networked control system with transmit power constraint. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-7, doi: 10.1109/SSCI.2016.7849834.

Yan, Q., Yu, F.R., Gong, Q. et al. (2016). *Software-Defined Network*ing (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602-622.

Yeung, P., & Jung, M. (2013). Improving Electric Reliability with Smart Meters, *White Paper*, Silver Spring Networks.
Yu, R., Zhang, Y., Gjessing, S. et al. (2013). Toward cloudbased vehicular networks with efficient resource management. *IEEE* Netw., vol. 27, no. 5, pp. 48–55.

Yu, Y., Xin, L., Shanzhi, C., et al. (2010). A framework of using OpenFlow to handle transient link failure. in Int. *Conf. on Transportation, Mechanical, and Electrical Engineering*(TMEE), pp. 2050–2053.

Yuanxiong, G., Miao, P., & Yuguang, F. (2012). Optimal power management of residential customers in the *Smart Grid*. *IEEE Transactions on Parallel and Distribution Systems,* vol. 23, pp.1593-1606.

Yuexin, L., Haoqing, X., Hanwu, L., et al.(2010). Investment-benefit analysis and evaluation model of the *Smart Grid*. *CICED 2010 Proceedings*, pp. 1-5.

Zhang, J., Seet, C.B., Lie, T.-T. et al. (2013). Opportunities for *Software-Defined Network*ing in *Smart Grid*. in Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on. *IEEE*, pp. 1–5.

Zhao, Y., Iannone, L. & Riguidel, M. (2015). On the performance of SDN controllers: A reality check," *2015 IEEE Conference on Network Function Virtualization and Software-Defined Network (NFV-SDN)*, pp. 79-85. doi: 10.1109/NFV-SDN.2015.7387410.

APPENDICES

Appendix 1

Multi-access Edge Computing (MEC)

Multi-access Edge Computing (MEC) provides a cloud computing platform at the edge of the Radio Access Network(RAN). MEC offers storage and computational resources. Facilitating multi-service and multi-tenancy by allowing authorized 3rd parties to make use of storage and network node processing. Also, MEC is a crucial enabler to support M2M and IoT services for smart city services, the energy sector, and automotive. Table 1A shows the MEC use case and MEC solution.

Table 1A: MEC Use Case and MEC Solutions

Use Case:	MEC Solution:	Reference:
Computation Offloading	RAN-aware content optimization breaks down applications into segmented components and creates an offloading online strategy based on optimization parameters. Creating an NFV-enabled MEC architecture for video and gaming.	Orsini et al. (2015)
Distributed Content Delivery and Caching	Hybrid caching enables the option of requesting content from other nearby edge platforms, Ensuring optimum QoS.	Taleb et al. (2017)

Web Enhancement	Reduction in access time and web-page loading acceleration:
	• Content Optimization,
	• Accelerated Browsing,
	• Web Acceleration.
		Simmons et al. (2011);
		Takahashi et al. (2015);
		Mau et al. (2015).

Big IIoT data	Local IoT gateway functionality can perform GTP aggregation and big data analytics for *Smart Grid*, e-health, or alarm notification. IoTCloud/mMEC - Cloud-based open-source controller and architecture with an API enable scalable sensor-centric and smart manufacturing applications.	Sun & Ansari (2016); Fox et al. (2012).
Smart City Services	Resource management, safety, and VM mobility: Content distribution and	Yu et al. (2013); Hu et al. (2018).

	processing of car-to-car, car-to-infrastructure. Integration of licensed Sub-6 GHz band + pure IEEE 802.11p-based V2V communications.	

The characteristics(Porambage, 2018 et al.) of MEC applied to the real world can benefit in several sectors highlighted in Table 1B.

Table 1B: Characteristics of MEC when applied to sectors

Characteristics of MEC	Smart Home	Smart City	Remote Surgery	Autonomous Vehicle	AR	VR	Gaming	Retail	Wearable IoT	Farming	Smart Energy
Low Latency		x	x	x	x	x	x		x	x	x
Increase Bandwidth	x	x	x	x	x	x	x	x	x	x	x
Content Awareness	x	x	x		x	x	x	x	x	x	x
Fast Inter-RAT handoff	x	x	x		x	x	x	x	x		
Caching	x	x			x	x	x	x	x		
Edge Analytics	x	x	x		x	x	x	x	x		
Security	x	x			x		x			x	x
Fast Mobility	x	x	X	x	x		x		x	x	x

The challenges of MEC are stated below (Dao, 2017 et al.) and recommended solution:

- Green and costly infrastructure which exhausts the networks' bandwidth.
- Engineer to plan according to the site and traffic utilization/forecast process.
- Resource management – high computation requirements.
- Use the theory of stochastic geometry, queueing, and parallel computation for provisioning.
- Mobility management – Real-time management required.
- Introduce Mobile-IoT-Federation-as-a-Service(MIFaaS) for dynamic cooperation.
- Security control - privacy, integrity, and confidence in the customer data processing.

- Review/Implement Cyber-Physical System(CPS) layered approach based on IoT and MEC.

The deployment of MEC across a network will enable a further scalable, efficient, and resilient network.

Appendix 2

Installation code

Installation code for Java:

```
sudo apt install default-jdk
java -version
```

Installation code for Floodlight:

```
$ git clone git://github.com/floodlight/floodlight.git
$ cd floodlight
$ git submodule init
$ git submodule update
$ ant

$ sudo mkdir /var/lib/floodlight
$ sudo chmod 777 /var/lib/floodlight
```

Appendix 3:

Testbed

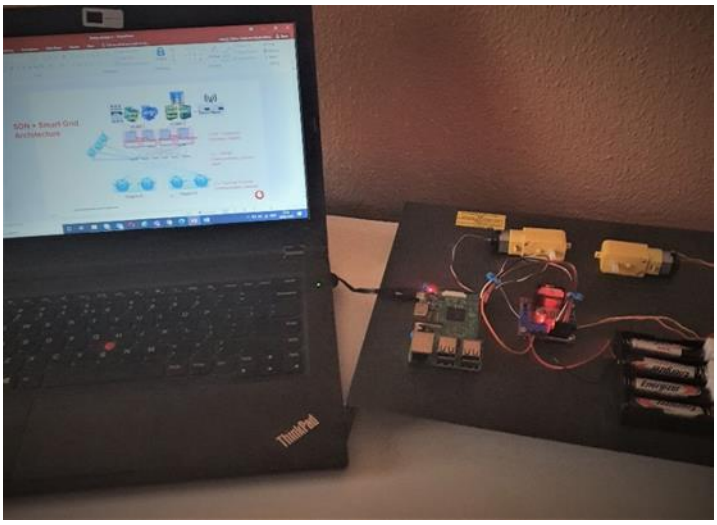


Figure 3A: Testbed

Appendix 4: IP SDN/Orchestration Provisioning – Future recommendations.

The information presented shows the future expansions and technologies that work with SDN technology.

IP only – Layer 3, VPN Service provisioning end to end.

Layer 3 Service creation creates a secure connection for a customer from their site to their corporate site. The operation of the service request is via the RESTful interface and received by the Network Service Orchestrator(NSO), containing service level agreements. The first step of the NSO is to check if the customer is registered and authorized. The NSO forwards the request to the Hierarchical orchestrator with the service access points depending on the network parameters. The Hierarchical IP WAN controller will look at source and destination domains to formulate all Autonomous System Boundary Router to establish connectivity. Figure 4A shows a 5G network with SDN controllers within the cloud; a 5G network can latency enhanced mobile broadband and mobility from different network slices. The addition of orchestration will need further research and how to implement the full solution in the future.

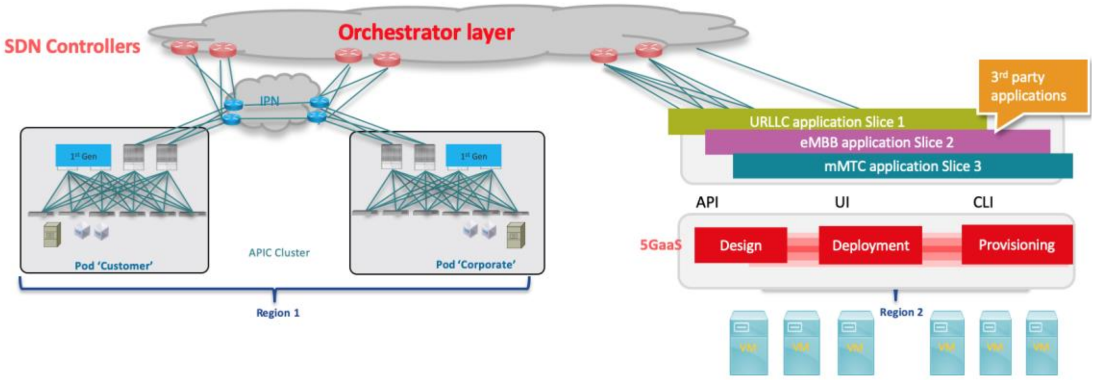


Figure 4A: Network architecture on a 5G connectivity flow

Figure 4B shows the process involved when a service request is presented.

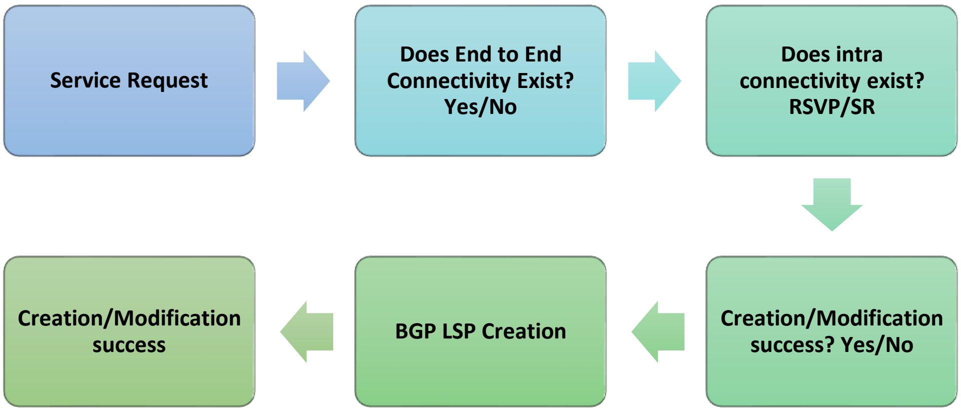

Service Request
Does End to End Connectivity Exist? Yes/No
Does intra connectivity exist? RSVP/SR
Creation/Modification success? Yes/No
BGP LSP Creation
Creation/Modification success

Figure 4B: Layer 3 VPN Service provisioning

Multi-vendor multi-domain Orchestration

The SDN multi-vendor and multi-domain orchestration allow the setup of a layer one service across international and national domains end to end. The hierarchical controller performs inter-domain service connectivity. Taking into account each domain belonging to another vendor. Supports full open Southbound APIs and extensions of WAN optical controller architecture. The ability allows for connection of high availability of 99.99% and guaranteed bandwidth. When a service is requested, the Layer 1 interconnection forwards the request to the Service Connectivity Orchestrator. The connection is from the Data Centre to the Corporate site. And, thus, the requirements are more severe. The Service Connectivity Orchestrator queries the Global inventory database to define Layer 1 service endpoints. To obtain original terminating Optical Node IDs and their corresponding port interfaces.

Bandwidth on Demand/Optical VPN

Optical-Virtual Private Networks provide large Industries or Carriers access to an SDN network, referring to capacity, provisioning, and monitoring of the network slice. O-VPN supports the creation of virtual/intelligent/dedicated provisioning of the Service Provider's network. The bandwidth visualization provides a secure dedication to the user plane capacity based on requirements. The customer is then able to manage their network within a multi-domain network. Figure 4C shows site a and b, belonging to different domains; the SDN controller is the brain and policies' formation.

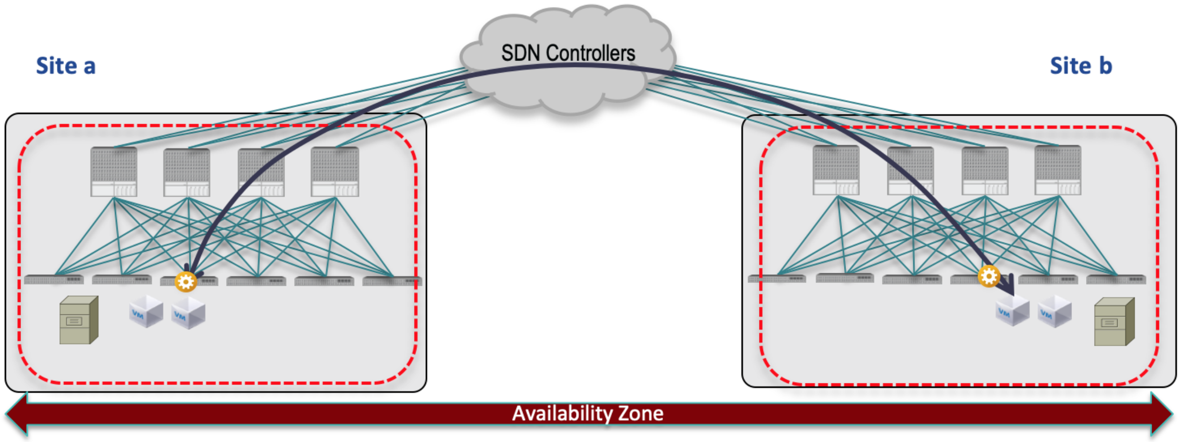


Figure 4C: SDN Controller layer over Site a and b

Network Evaluation

The use case presents a far automated approach or less contention in the management and analysis of the network. The aim is to facilitate and automate some of the operational functions on real-time network topology. One of the feature sets is on-line provisioning, allowing the creation of service on a live network while not disturbing the rest of the live services—the ability to restore in case of failure given by policy. Another feature is Network survival; the on-line service survival performs live troubleshooting when a failure occurs—the process between the Application, E2E Service Orchestrator, and Computation Engine.