ADAPTIVE FAILURE RECOVERY AND PATH OPTIMIZATION FOR MULTI-HOP SOFTWARE-DEFINED EXCHANGE POINT NETWORKS

ABDIJALIL ABDULLAHI MOHAMED

UNIVERSITI SAINS MALAYSIA

2025

ADAPTIVE FAILURE RECOVERY AND PATH OPTIMIZATION FOR MULTI-HOP SOFTWARE-DEFINED EXCHANGE POINT NETWORKS

by

ABDIJALIL ABDULLAHI MOHAMED

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

July 2025

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to "Allah," our Creator, for bestowing upon me his blessings and the opportunity to earn my Ph.D.

Therefore, I would like to express my sincere gratitude to my supervisors, Associate **Prof. Dr. Selvakumar Manickam** (main supervisor), and **Dr. Shankar Karuppayah** (co-supervisor), for their support, creative ideas, and valuable assistance throughout my entire of PhD at the **Cybersecurity Research Centre, Universiti Sains Malaysia**, which is a prestigious institution. I would like to express my gratitude to the individuals who have directly or indirectly assisted me and contributed to the development of my academic career.

In particular, I would like to thank **SIMAD University** for providing me with a scholarship and all the financial support I needed to pursue this course of study. In addition, I would like to thank my wives. My life has undoubtedly been constructed over many years on the unshakeable foundation of their love, support, and encouragement. Their unwavering love and dedication are demonstrated by their tolerance of my usually unpleasant moods. I'm grateful to my mother for having faith in me and letting me pursue my ambitions. She kept a close eye on me, and that gave me the motivation and courage to face obstacles head-on. I would also like to thank my siblings, especially my exceptional brother, Mohamed Hassan, for providing me with the necessary support when I needed it. Finally, I would like to dedicate this work to my children: Suheylo, Mohamed, Abdullahi, Sumayo, and Abdirahman.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

IXP	Internet Exchange Point
SDN	Software-Defined Networking
SDX	Software Defined Exchange Point
BGP	Border Gateway Protocol
ISP	Internet Service Provider
CDN	Content Delivery Network
ELFR	Enhanced Link Failure Rerouting
FRR	Fast Rerouting
ARPANET	Advanced Research Project Agency Network
CSNET	Computer Science Network
NSFNET	National Science Foundation Network
NAP	Network Access Point
ARP	Address Resolution Protocol
AS	Autonomous System
iSDX	Industrial Software-Defined Exchange Point
RS	Route Server
OF	Openflow
P4	Programming Protocol-Independent Packet Processors

PEMULIHAN KEGAGALAN ADAPTIF DAN PENGOPTIMUMAN LALUAN UNTUK RANGKAIAN TITIK PERTUKARAN DITAKRIFKAN PERISIAN BERBILANG LONJAKAN

ABSTRAK

Tesis ini membentangkan mekanisme Enhanced Link Failure Rerouting (ELFR) yang dibangunkan untuk menangani cabaran dalam pemulihan kegagalan rangkaian Titik Pertukaran Ditakrifkan Perisian (SDX) berbilang hop. SDX, iaitu evolusi kepada Internet Exchange Points (IXPs) tradisional, menggunakan pendekatan Rangkaian Ditakrifkan Perisian (SDN) untuk membolehkan kawalan trafik yang boleh diatur cara. Namun begitu, pelaksanaan SDX berbilang hop masih menghadapi beberapa kekangan ketara seperti kelewatan pemprosesan paket, pengiraan laluan yang perlahan serta penggunaan memori yang tinggi semasa berlakunya kegagalan. Bagi menangani isu-isu ini, ELFR telah direka bentuk dengan dua modul bersepadu: Modul Pemprosesan Paket – membolehkan pengelasan dan pemulihan segera dalam suis menggunakan pengepala yang ditentukan oleh P4; dan Modul Pengiraan Laluan – yang secara proaktif mengira laluan sandaran berasaskan jiran bagi mengurangkan beban kepada pengawal. Keputusan menunjukkan bahawa ELFR mengatasi mekanisme Flow-Aware Pro-Reactive (FAPR) dengan mengurangkan Masa Pemulihan (RT) sebanyak 46.97%, Masa Pengiraan (CT) sebanyak 34.62%, dan Overhed Pengiraan (CO) sebanyak 44.44%. Ia juga berjaya mengurangkan Penggunaan Memori (M) sebanyak 37.5%, Kelewatan Pemprosesan Paket (PPD) sebanyak 28%, dan Nisbah Kehilangan Paket (PLR) sebanyak 63.16%. Penemuan ini membuktikan bahawa ELFR ialah solusi cekap dan tahan lasak untuk menangani kegagalan pautan dalam rangkaian SDX boleh atur cara.

ADAPTIVE FAILURE RECOVERY AND PATH OPTIMIZATION FOR MULTI-HOP SOFTWARE-DEFINED EXCHANGE POINT NETWORKS

ABSTRACT

This thesis presents the Enhanced Link Failure Rerouting (ELFR) mechanism, developed to address challenges in failure recovery within multi-hop Software-Defined Exchange Point (SDX) networks. SDX, an evolution of traditional Internet Exchange Points (IXPs), uses Software-Defined Networking (SDN) to enable programmable traffic control. However, multi-hop SDX deployments still face significant limitations, including high packet processing delays, slow path computation, and excessive memory consumption during failures. To tackle these issues, ELFR was designed with two integrated modules: the Packet Processing Module, which allows fast in-switch classification and recovery using P4-defined headers, and the Path Computation Module, which proactively calculates neighbor-based backup paths to lessen controller overhead. The architecture was implemented with Mininet, BMv2, P4Runtime, and the POX controller, and tested across four realistic failure scenarios. Results show that ELFR outperforms the Flow-Aware Pro-Reactive (FAPR) mechanism by reducing Recovery Time (RT) by 46.97%, Computation Time (CT) by 34.62%, and Computational Overhead (CO) by 44.44%, while decreasing Memory Usage (M) by 37.5%. Additionally, it minimizes Packet Processing Delay (PPD) by 28% and Packet Loss Ratio (PLR) by 63.16%. These findings confirm ELFR as an efficient, scalable, and resilient solution for recovering from link failures in programmable SDX infrastructures.

CHAPTER 1

INTRODUCTION

In 1990, the Internet was utilized by only a few million users, and commercial enterprises have just started integrating this innovative distributed infrastructure (Basit et al., 2020). Similarly, the Internet had a hierarchical structure that managed a small group of transit providers (Tier-1) that offered global connectivity. The process of routing network traffic from one geographical region to another is commonly reliant on large transit providers, known as Tier-1 providers. The Tier-1 entities occupy the highest position inside the hierarchical structure formed by a few thousand autonomous systems (ASes) that constitute the network of networks (da Silva et al., 2022). However, network operators have begun to change the structure of the Internet topology and create a flatter network via direct peering connections. Due to the Internet's topology, interconnection between Internet service providers (ISPs), transit providers, and Internet content providers (ICPs) is organised in a hierarchical structure. However, the Internet topology requires a significant transformation from hierarchical to flat (Hoeschele et al., 2021; Marcos et al., 2020).

Significant transformations have occurred since the initial stages, in which smaller autonomous systems, ASes, would resign to larger ones in exchange for connectivity (Böttger et al., 2019). The reliance on intermediaries has increased, resulting in higher transit fees, more complex routes, longer round-trip times, and a general lack of control over service quality (Alam et al., 2021). The use of direct peering links as a means to bypass intermediaries became a clear and logical response, leading to the emergence of Internet Exchange Points (IXPs) as the primary infrastructure for creating peering connections (Alam et al., 2021; Ó Briain, 2019).

To do Therefore, Internet Exchange Points (IXPs) have emerged to change the structure of the Internet topology and reduce interconnection costs. In recent decades, IXPs have become a vital component in the Internet peering ecosystem. As shown in Figure 1.1, IXPs are leveraged dramatically by Internet operators, which results in an increase in the number of providers participating in IXPs across the globe and growth trend of peering after the establishment of first IXP.

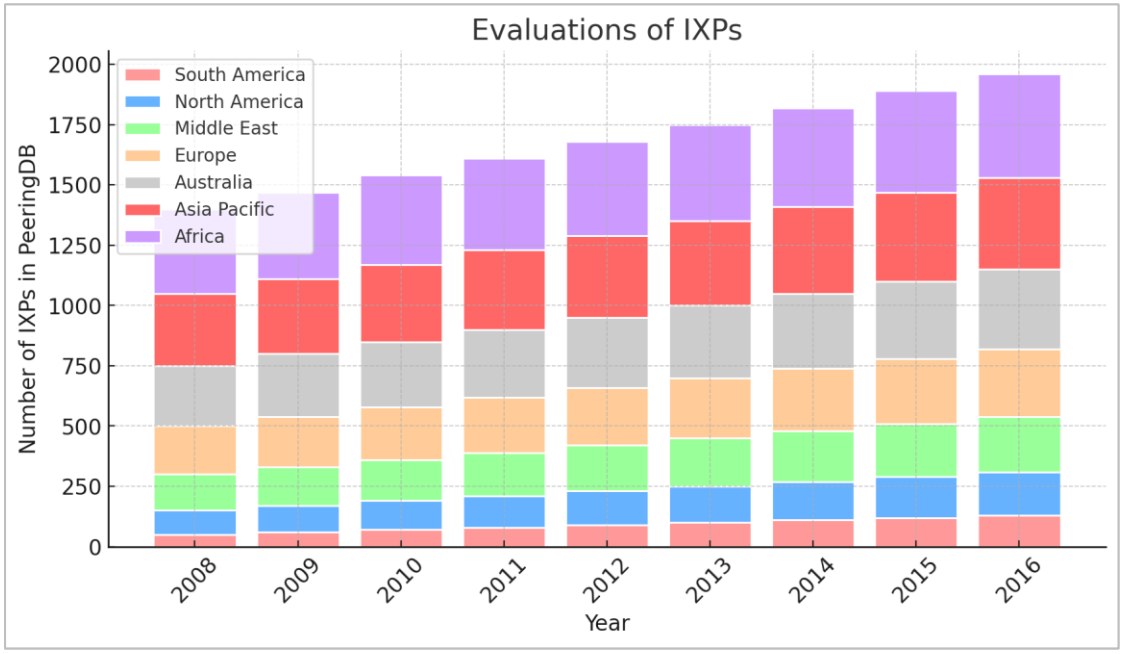

Figure 1.1 Evolution of the Number of IXPs Per Region (Böttger et al., 2019)

One technique to maximize network capacity is to use Internet Exchange Points (IXP). It enables many providers, such as content delivery networks (CDNs) and Internet service providers (ISPs), to share traffic across a single point. Additionally, it reduces the number of connections between various Internet service providers and the various network problems (Bertholdo et al., 2022). Additionally, IXP simplifies the localization of domestic traffic, eliminating the need to send it to the international link. This localization results in reduced numbers of connections, traffic routed to specific links, traffic delays, and increased costs associated with international routing (Emran et

al., 2019). IXP also optimizes the bandwidth and helps to improve network security (Lv et al., 2018; Subramani et al., 2020).

Domestic traffic routing internationally typically raises the possibility of jeopardizing data security. Therefore, by routing traffic to specific local ISPs and CDNs, IXP reduces the risks. In a similar vein, IXPs are a crucial part of the high-performance, scalable connection of today's network. IXPs essentially function as a layer-2 switch's counterpart, linking hundreds of network providers and enabling direct traffic exchange between them. It enables the following outcomes: increased local traffic, high bandwidth, reduced latency, and lower transit costs. For example, all other member networks can receive traffic directly from the content delivery networks linked to an IXP (M. V. B. da Silva et al., 2020; Masoud et al., 2017).

Setting up an Internet Exchange Point (IXP) entails a significant initial financial investment in infrastructure, as well as ongoing operational expenses. The management of an IXP is a complex undertaking that necessitates technical expertise, particularly in handling intricate peering arrangements and maintaining robust network operations (Giachone et al., 2018). As traffic through an IXP increases, the infrastructure must scale accordingly, which poses substantial challenges and requires further investment. Disputes over peering policies can lead to conflicts among members, potentially affecting the traffic flow and the efficiency of the IXP (Chiesa, Demmler, et al., 2017). Security is another concern, as centralizing traffic through an IXP can create a focal point for cyber threats, including DDoS attacks (Müller et al., 2019). Maintaining high levels of uptime is critical, as any downtime can disrupt a wide array of networks that rely on the IXP for traffic exchange (J. Böttger, 2017). Furthermore, IXPs may face regulatory and political obstacles that can limit their operations. They also operate in a

competitive market against other IXPs and private peering setups, which can influence their traffic volume and relevance. Lastly, IXPs are most beneficial for localizing traffic; networks with a significant amount of international traffic may not see as much benefit from using an IXP (Chiesa, Demmler, et al., 2017; Stocker et al., 2021).

Metric	With IXP	Without IXP
Latency	Lower	Higher
Packet loss	Less	More
Path length	Shorter	Longer
Cost	Lower	Higher

Figure 1.2 Leverages of IXP on Internet Providers

As shown in Figure 1.2, with an IXP, latency is drastically reduced, bandwidth is optimized, and local internet connectivity is strengthened. This means quicker and smoother data transfers that save costs and enhance the overall user experience. In contrast, the inexistence of IXPs resulted in a more decentralized network. Internet providers connect in a dispersed manner, potentially leading to longer, less efficient data routes, increased latency, and higher costs. Without an IXP, managing multiple point-to-point connections can be complex and expensive. Therefore, IXP has significant leverage over Internet providers, such as ISPs and CDNs.

IXP providers establish peering relationships between various network providers using the standard inter-domain routing protocol known as Border Gateway Protocol, or BGP. Two of their border routers must establish a direct BGP session to link the two networks (members). The primary requirement is for two ASes of the IXP to establish a bilateral (BL) BGP peering session at the IXP, enabling the exchange of traffic across the IXP's switching fabric. BGP, the Border Gateway Protocol, may promote Internet expansion and has a large capacity for scalability (Dabone et al., 2022; Rosa, 2018).

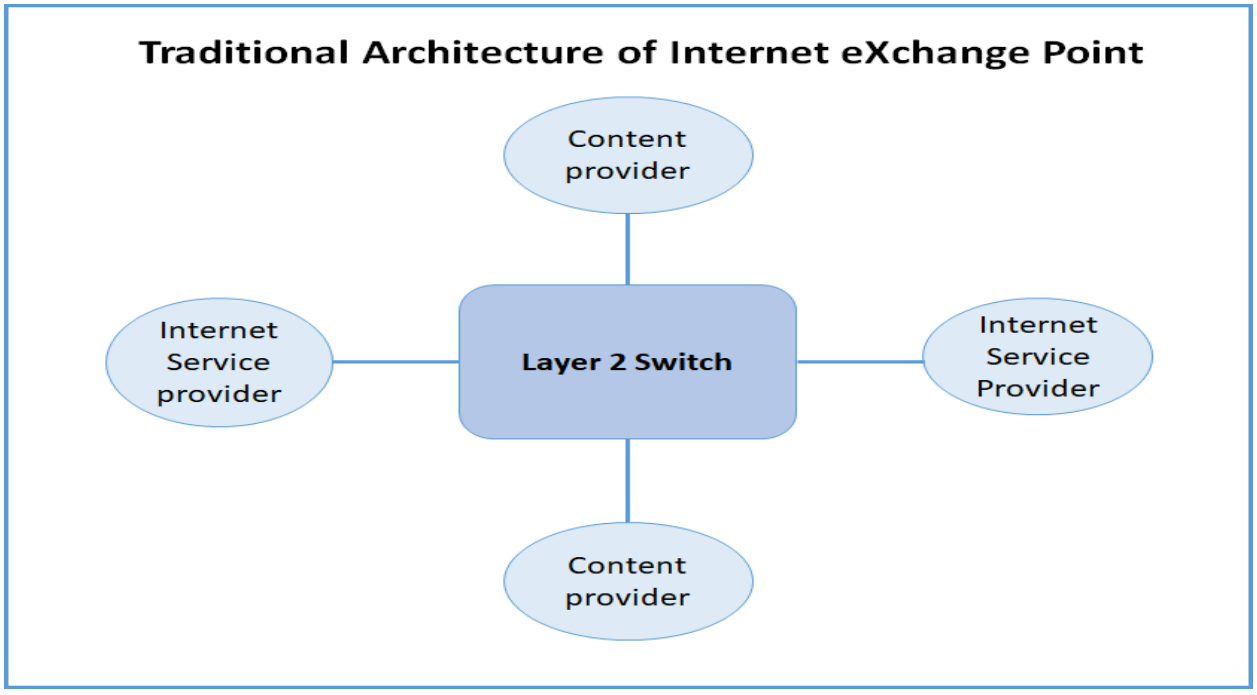

Figure 1.3 Traditional IXP Architecture

To establish a layer-2 switching fabric, a traditional IXP architecture typically requires at least one switch. As shown in Figure 1.3, this fabric connects various Internet service providers and content providers to share traffic with the IXP switch fabric by the peering agreement terms specified by the IXP.

1.1 Background

This section provides an explanatory overview of Internet Exchange Points (IXPs), Software-Defined Networks (SDNs), and Software-Defined Exchange Points (SDXs).

1.1.1 Internet Exchange Point

Internet Exchange Points (IXPs) serve as essential hubs in the global internet Infrastructure, facilitating effective data interchange among Internet Service Providers (ISPs), Content Delivery Networks (CDNs), and various other networks. Their primary role is to localize internet traffic, thereby reducing latency and improving network performance. By reducing the distance data packets must traverse, IXPs improve speed and reliability while simultaneously lowering operational costs for ISPs—savings that may ultimately benefit end users (Prehn et al., 2022; Visser et al., 2021). Moreover, IXPs are regulated by a consortium of stakeholders, comprising operators, member networks, and, in certain instances, governmental and international regulatory authorities. These structures of governance ensure compliance with established technical and security standards, often informed by entities such as the Internet Engineering Task Force (IETF), to maintain consistent and secure traffic exchange (Brooks, 2021; Cavalcanti, 2017; Lui, 2021).

As internet usage and service requirements continue to grow, IXPs are undergoing a substantial transition. To maintain scalability and resilience, numerous Internet Exchange Points (IXPs) are incorporating developing technologies like Software-Defined Networking (SDN), which enhances flexibility in traffic management and fault tolerance (Shukla & Stocker, 2019). The geographical proliferation of IXPs and the establishment of cooperative peering agreements are transforming the global connectivity architecture. These advancements underscore the

increasing importance of IXPs not only as infrastructural nodes but also as dynamic platforms that shape the future of internet growth and accessibility (Barbieri et al., 2021; Giotsas et al., 2020)

1.1.2 Software-Defined Network

Software-Defined Networking (SDN) transforms network architecture by facilitating logically centralized control over distributed, programmable switches. Unlike the conventional Border Gateway Protocol (BGP), which has historically been the standard for inter-domain routing, Software-Defined Networking (SDN) offers precise control over network traffic and improves programmability. BGP provides restricted flexibility, forwarding packets primarily based on destination IP prefixes, while SDN enables network administrators to execute dynamic and context-sensitive routing decisions (Chiesa et al., 2016; Foerster et al., 2018; Tang et al., 2016). These capabilities make SDN especially useful in data centers and inter-domain settings, where efficiency and adaptability are crucial. Recent research indicates that Software-Defined Networking (SDN) has been successfully implemented in domains such as traffic engineering and network automation (Q. Li et al., 2019; Prajapati et al., 2018; Tsai et al., 2018).

SDN fundamentally separates the control plane from the data plane, enabling centralized management of network intelligence. This architecture enables the SDN controller to govern routing decisions and flow policies autonomously from the devices involved with packet forwarding (V. Muthumanikandan & Valliyammai, 2017). SDN switches function primarily as data-plane devices, directing traffic according to flow rules established by the controller. At the initiation of a new flow, the switch transmits the initial packet to the controller, which then installs the corresponding flow rule. This mechanism decreases the necessity for continuous controller intervention, optimizing

packet forwarding and improving performance (Barolli, 2019; Haque & Abu-Ghazaleh, 2016; Mousa et al., 2016; Rawat & Reddy, 2016; Shu et al., 2016). This separation of responsibilities not only facilitates network management but also establishes a basis for scalable, policy-driven management.

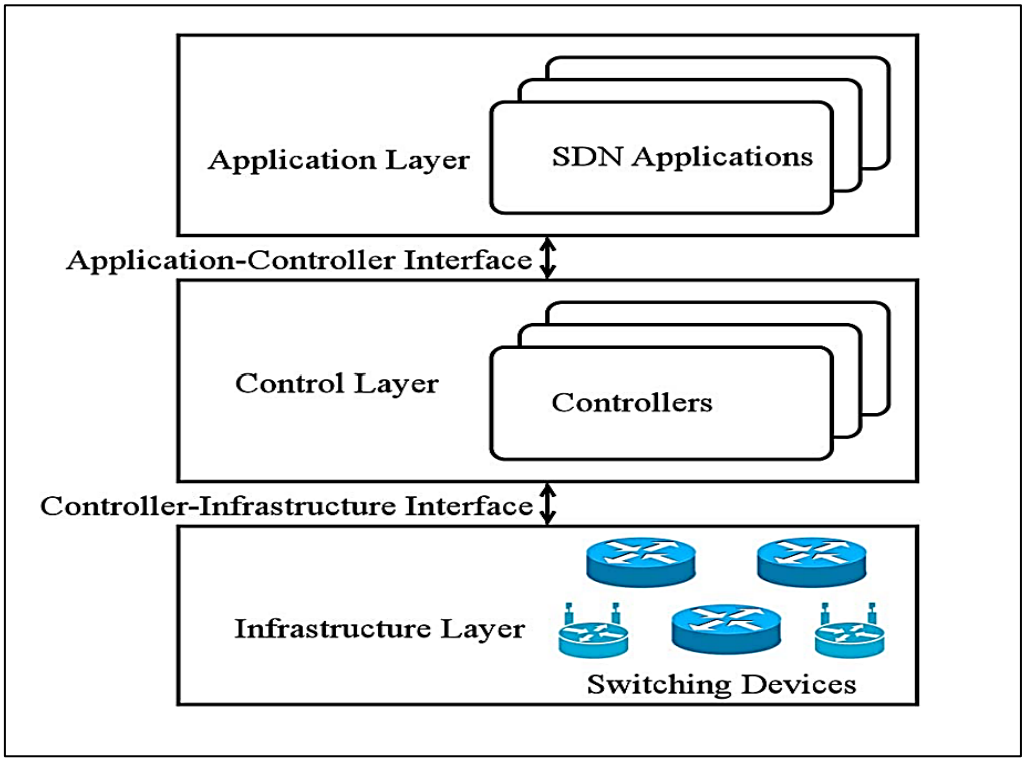

Figure 1.4 SDN Reference Model (Shin et al., 2016; Singh & Jha, 2017)

The architecture of SDN facilitates flexibility through its layered structure, comprising the infrastructure layer, control layer, and application layer as shown in Figure 1.4 (Shin et al., 2016; Singh & Jha, 2017). The infrastructure layer comprises network devices that are responsible for packet forwarding and providing real-time network status. The control layer serves as an intermediary, facilitating communication downward with the infrastructure through southbound interfaces, such as OpenFlow, and upward with the application layer via defined application programming interfaces (APIs). These APIs enable SDN applications to retrieve network information, facilitate informed decision-making, and implement flow rules on network devices. This design facilitates centralized policy enforcement and adaptive service supply. Furthermore,

SDN applications facilitate sophisticated capabilities such as network virtualization, dynamic access control, and load balancing, rendering SDN an effective instrument for contemporary network management (Xia et al., 2015).

1.1.3 Software-Defined Exchange Point

Software-Defined Exchange Points (SDX) represent the highly sophisticated improvement of Internet Exchange Points (IXPs), providing precise control over traffic routing and inter-domain policy implementation. In contrast to conventional Internet Exchange Points (IXPs), which utilize BGP-based routing that offers restricted and fuzzy forwarding capabilities, Software-Defined Exchanges (SDX) employ Software-Defined Networking (SDN) to facilitate accurate, programmable traffic management informed by real-time conditions and user-defined policies (Griffioen et al., 2016; Hermans et al., 2016). This dynamic capacity enables network operators to adjust traffic routes based on congestion levels or application-specific demands. Furthermore, SDX promotes multi-domain collaboration by enabling autonomous administrative entities to share network, storage, and computational resources while specifying diverse and extensive routing regulations within a shared exchange infrastructure (Chung et al., 2017; Mambretti et al., 2017).

From an operational standpoint, SDX diminishes complexity and expenses by facilitating automated configuration and dynamic policy implementation. This design facilitates sophisticated use cases, including inbound traffic engineering, service chaining via middleboxes, and application-specific peering, thus broadening the functional capabilities of IXPs beyond simple packet transit hubs (Chung et al., 2016). Moreover, SDX incorporates essential security features within the exchange infrastructure. notable among these are Distributed Denial of Service (DDoS) mitigation, which safeguards against massive attacks, and prefix hijack prevention via

the implementation of Resource Public Key Infrastructure (RPKI), enabling providers to authenticate route advertisements and prevent both malicious and accidental hijacking (Nakauchi & Nishinaga, 2016; Ricci & Feamster, 2016).

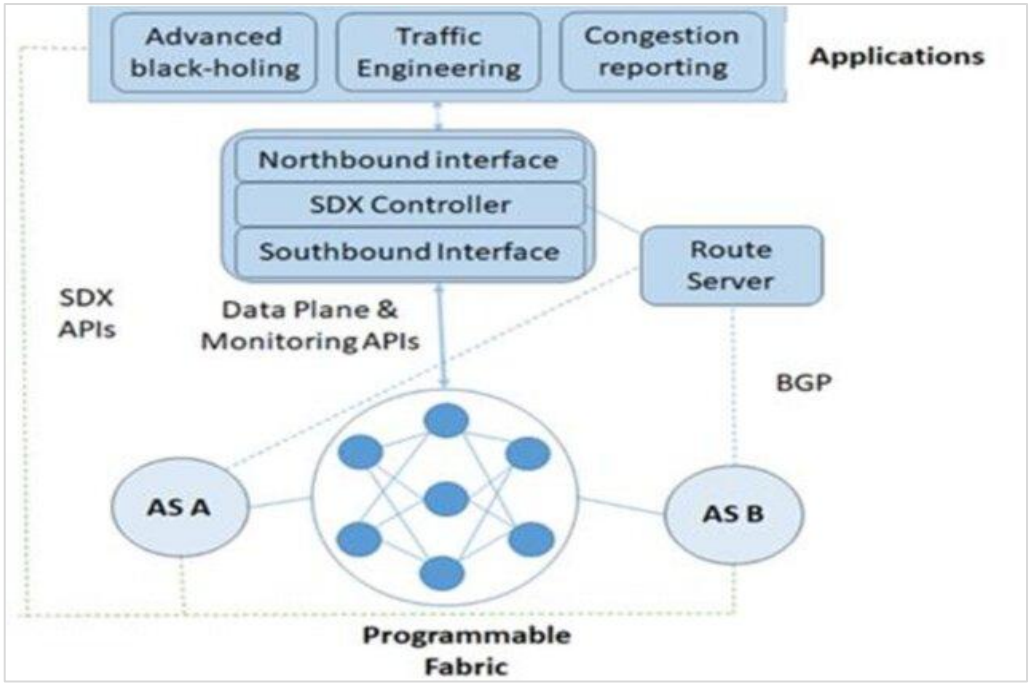

Figure 1.5 SDX Architecture

SDX implements a programmable architecture that enhances inter-domain traffic management by incorprating Software-Defined Networking (SDN) principles into traditional Internet Exchange Points (IXPs). The SDX architecture primarily consists of an SDN controller, a programmable switching fabric, a BGP route server, and management applications, as shown in Figure 1.5. The SDX controller provides each Autonomous System (AS) with the abstraction of a dedicated virtual switch, facilitating precise match-action policies for inbound and outbound traffic. This replaces default BGP forwarding while maintaining adherence to route advertisement rules (Gupta, Macdavid, et al., 2016; Mandal et al., 2017). This architecture enables the conversion of high-level policies into low-level forwarding rules, implemented via an OpenFlow-enabled fabric. APIs provide sophisticated management functionalities, enabling SDX participants to execute dynamic, secure, and efficient traffic engineering

across administrative boundaries, hence improving routing flexibility, performance, and security (Tsai et al., 2021; Wang et al., 2016).

1.1.4 Challenges and Considerations in SDX

This section typically provides a taxonomy of SDX challenges, categorizing key obstacles such as interoperability, fault tolerance, redundancy, and security in these advanced networks. It addresses how these issues impact the efficient management and evolution of software-defined IXPs. Here is the explanation of subsections:

1.1.4(a) Interoperability

Interoperability in SDX is crucial for ensuring that the new, more flexible network infrastructures can seamlessly work with existing systems. This challenge includes ensuring compatibility with legacy systems, which might not be designed for the dynamic and programmable nature of SDX (Liu et al., 2019). Additionally, cross-platform integration is crucial, as SDX environments frequently involve equipment and software from multiple vendors. Ensuring that these diverse components can communicate and function together effectively requires standardized interfaces and protocols (Themistocleous et al., 2023).

1.1.4(b) Complex Configuration and Management

The advanced capabilities of SDX introduce a higher level of complexity in network configuration and management. Technical personnel need specialized skills and knowledge to handle these sophisticated systems (Dzeparoska et al., 2018). Furthermore, the dynamic nature of SDX demands advanced automation and orchestration tools. These tools must efficiently manage and adapt network configurations in real-time, aligning with changing network conditions and requirements (Ahmad & Mir, 2021).

1.1.4(c) Dynamic Network Topology

SDX environments are characterized by their dynamic network topologies. This dynamism requires the network to adapt in real-time to changes, such as varying traffic patterns or the addition and removal of nodes (Bagci & Tekalp, 2019). Scalability is also a critical concern; as traffic volume and network size increase, the SDX must scale effectively without compromising performance or reliability (Huang et al., 2019).

1.1.4(d) Quality of Service (QoS)

Maintaining a high quality of service is essential in SDX. This involves ensuring consistent network performance, particularly in bandwidth and latency, across various applications and services (Tomovic & Radusinovic, 2019). Traffic prioritization becomes a significant challenge, requiring the network to differentiate and prioritize different types of traffic, such as critical data packets to maintain service levels (Biczok et al., 2017).

1.1.4(e) Security Considerations

Security in SDX involves multiple layers, with the protection of the centralized control plane being paramount. This control plane is a potential target for cyberattacks, and robust security measures are necessary to safeguard it (Mei, 2017). Additionally, ensuring data privacy and integrity as data traverses through the SDX network is crucial for maintaining user trust and complying with regulatory standards (L. Yu et al., 2017).

1.1.4(f) Latency and Packet Processing

Minimizing latency is crucial in SDX, particularly for applications that require real-time processing. This challenge involves optimizing the network to reduce delays in data transmission (Q. Wang, 2021). Efficient packet handling is also crucial; the

network must efficiently route and forward packets, which requires sophisticated algorithms and network design strategies (H. Yu et al., 2019).

1.1.4(g) Monitoring and Visibility

Effective monitoring and visibility are key to proactive network management in SDX. Real-time network analytics are needed to gain insights into network performance and traffic patterns (Cevik et al., 2022). Additionally, complete visibility across the network, including both the control and data planes, is vital for detecting issues, optimizing performance, and ensuring security (R. K. Shukla, 2021).

1.1.4(h) Protocol Evolution and Standardization

Keeping pace with the evolution of network protocols is a significant challenge in SDX. The network must be adaptable to new protocols as they emerge(Alotaibi et al., 2022). Furthermore, adherence to industry standards is crucial for interoperability and compatibility across different network environments (Bannour et al., 2017). Standardization ensures that SDX solutions can effectively communicate and operate within the broader internet infrastructure (Latif et al., 2020).

1.1.4(i) Fault Tolerance and Redundancy

Building resilient SDX systems capable of withstanding failures is essential for network reliability. This includes addressing link failures, a critical issue as they can disrupt traffic flow and lead to service outages (Bannour et al., 2017). Ensuring network redundancy and implementing effective failover mechanisms are necessary to quickly recover from such failures and maintain uninterrupted network services (Cevik et al., 2022).

1.1.5 Link Failure

Regarding fault tolerance and redundancy, this section highlights the emphasis of the research on addressing the specific issue of link failure within SDX environments.

The quantity of devices connected to the network is always increasing. Delivering packets to the destination more quickly has become a major challenge. Many problems, including as congestion, scarcity of bandwidth, and network and node failures, have been studied to transport packets faster. According to the previous studies (Muthumanikandan & Valliyammai, 2017b; Tan et al., 2019; Yasin et al., 2022), link failures are a serious problem in modern networking that requires solutions.

The academic and industrial domains are particularly interested in network failure recovery because of how important reliability is. Failure recovery has long been a major challenge in network management, especially software defined networks. There are a few proposed algorithms and strategies to guarantee network dependability in SDNs. These technologies allow one to plan ahead and create backup routes in case of anticipated network interruptions. However, because they ignore the restricted resources (such flow tables) of SDN switches or the failure recovery requirements of different flows, they are typically impracticable for deployment (Daha et al., 2021; Q. Li et al., 2019; Semong et al., 2023).

The issue of route interruption is the most prevalent network breakdown constraint. This issue can cost you a significant sum of money or your reputation. According to a study, the cost of delay caused by interruption is $8000 per minute on average. In contrast, this loss can be $100,000 or more for websites that conduct business online. Interruptions in routing can occur for a variety of reasons, including attacks, software flaws, and interface problems. According to a study, even in a network that is well-maintained, interruptions cannot be totally prevented, making it even more

important to restore interrupted services in a short time. How to replace an interrupted path with an alternate route is a key issue in interruption recovery. One of the most common techniques used in modern networks is rerouting. The interruption issue is addressed by rerouting techniques that permit routing protection (Ali et al., 2020a; Chen et al., 2018; Chu et al., 2015; He et al., 2019; Petale & Thangaraj, 2020).

1.2 Research Motivation

In the realm of network infrastructure, Internet Exchange Points (IXPs) have historically contributed in resolving numerous challenges within the industry. However, it is essential to acknowledge that existing IXPs face their own set of challenges and limitations.

Traditional IXPs, for example, have found it difficult to offer a wide range of data plane rules, such as efficient traffic redirection to intermediary devices, wide-area server load balancing, application-specific peering, and confined to inbound traffic engineering. Furthermore, as IXPs interconnect a growing number of network providers and facilitate the exchange of substantial traffic volumes, they face management and infrastructure challenges. These challenges encompass issues such as layer-2 forwarding limitations, susceptibility to broadcast storms, lack of traffic isolation, vulnerability to DDoS attacks, and limited mechanisms for ensuring accountability (Cho & Son, 2022; Kientega et al., 2021; Kotronis et al., 2016; Martins et al., 2018; Nawrocki et al., 2019; Nomikos et al., 2018; Sermpezis et al., 2017; Trusin et al., 2022; Warraich et al., 2020a).

Additionally, traditional IXPs rely on the Border Gateway Protocol (BGP) as the standard inter-domain routing protocol to establish peering relationships among diverse network members. Regrettably, BGP exhibits its own set of limitations,

including constrained routing to specific destination IP prefixes and indirect path selection mechanisms. While BGP has significantly contributed to the evolution of the Internet, it has also encountered considerable inflexibility and stagnation, characterized by slow convergence, complex network management, and a lack of end-to-end quality of service guarantees (Chiesa, di Lallo, et al., 2017; Dolnák, 2016; Nomikos et al., 2017; W. J. A. Silva & Sadok, 2018) To address these challenges and pave the way for network programmability, simplified management operations, and enhanced IXP services, the concept of software defined networking, SDN, has emerged (Mueller et al., 2020; Nomikos et al., 2018; Warraich et al., 2020).

IXP providers have adopted the SDN paradigm in response to the aforementioned difficulties, successfully severing the data plane from the control plane.

According to study (Lapeyrade et al., 2016), SDN provides sophisticated policies and robust functions that enable providers to manage their services more effectively and provide their clients with better options. The idea of the Software Defined Exchange Point (SDX), which is an application of SDN principles to Internet exchange points, is a significant advance in this regard. With SDX, network operators may now regulate forwarding processes with never-before-seen granularity thanks to a novel technique. Operators can develop and adapt their services at significantly shorter timescales, enabling more effective traffic exchange compared to conventional methods (Knob et al., 2016; Lapeyrade et al., 2016).

While SDX and SDN have undoubtedly mitigated many challenges encountered by IXP operators, it is important to acknowledge that certain persistent issues remain. These obstacles include things like packet processing delays and link failure recovery, which calls for the use of cutting-edge methods to improve the operational effectiveness of multi-hop IXP providers even further.

1.3 Problem Statement

Internet Exchange Points (IXPs) are essential to the architecture and functionality of the worldwide Internet. By facilitating direct connections among numerous autonomous networks, they enhance high-speed data transmission and promote the effective distribution of inter-domain traffic. Despite their strategic importance, the operational framework of conventional IXPs is predominantly inflexible and reactive, primarily due to their reliance on the Border Gateway Protocol (BGP). This protocol, which directs traffic exclusively based on destination prefixes and neighbor-advertised paths, constrains IXPs from employing dynamic or context-sensitive routing methods. As a result, legacy IXPs frequently experience configuration-related vulnerabilities, including routing loops, broadcast storms, and congestion, especially during connection failure scenarios.

In response to increasing needs for scalability and flexibility, Internet Exchange Points (IXPs) have progressively adopted software-defined networking (SDN) approaches, leading to the emergence of Software-Defined Exchange Points (SDXes). These facilitate centralized policy enforcement and programmability, especially in single-switch situations. The transition to more realistic multi-hop SDX topologies, in which traffic passes via numerous switches, presents a new diversity of operational challenges. In such systems, the breakdown of a single connection might propagate through numerous nodes and disrupt essential forwarding channels. Ensuring high availability and service continuity in this setting requires precise coordination and rapid rerouting capabilities, which conventional mechanisms often fail to deliver.

A primary difficulty in multi-hop SDX architectures is the latency and overhead related to connection failure recovery. Upon failure, rerouting generally requires either

controller intervention or the application of statically precomputed pathways, both of which signifies performance and scalability limitations. Controller-dependent rerouting increases latency from control-plane communication, but static precomputed routes may lead to excessive switch memory use and restricted adaptability to dynamic topological changes. The complexities of these challenges become worse in wide topologies featuring numerous interconnected switches, where real-time coordination becomes progressively complicated.

Moreover, the constraints of conventional southbound protocols like OpenFlow restrict switches from independently executing failover decisions. These protocols frequently exhibit insufficient flexibility to adaptively modify forwarding behavior at the data plane level, resulting in a significant dependence on the centralized controller during failure situations. Consequently, packet loss, routing delays, and inefficient traffic engineering are commonly observed during failures in extensive SDX networks.

The lack of effective mechanisms for low-latency, resource-efficient, and scalable link failure recovery severely limits the reliability and performance of multi-hop SDX systems. The increasing need for real-time services, edge computing, and robust inter-domain communication highlights the significant risk posed by the inability to swiftly and autonomously address problems, challenging the future sustainability of programmable exchange infrastructures.

In order to ensure the scalability, resilience, and performance of advanced multi-hop Software-Defined Exchange Point (SDX) infrastructures, it is essential to tackle a series of enduring and technically intricate difficulties related to link failure recovery. These problems are especially crucial in dynamic and high-throughput contexts, where

even minor disturbances can substantially limit network performance and service availability. The main difficulties requiring focused looking at are as follows:

- Lack of adequate mechanisms for decentralized and prompt failover coordination among distributed switches, resulting in delayed rerouting responses and strengthened reliance on the centralized controller, which can negatively impact the speed and reliability of failure recovery in multi-hop SDX environments.

- Excessive rerouting latency and substantial memory usage stemming from conventional controller-based or statically precomputed recovery methods, both of which challenge the network's responsiveness and scalability.

- Existing data plane protocols, such as OpenFlow, exhibit limited flexibility, constraining programmable switches from performing intelligent in-switch recovery operations independently of controller intervention.

1.4 Research Objectives

The primary objective of this research is to develop an enhanced approach for link failure recovery, tailored explicitly to the multi-hop architecture of Software-Defined Exchange Point (SDX) networks. The following specific objectives direct the research to achieve this primary goal:

1) To design an adaptive link failure recovery mechanism to minimize packet processing delays and path computation in multi-hop software-defined exchange point networks.

2) To propose an efficient path computation method to reduce rerouting delays and prevent excessive switch memory usage in multi-hop software-defined exchange point networks.
3) To propose an optimized packet processing method to reduce the recovery time of link failure recovery in multi-hop software-defined exchange point networks.

1.5 Scope and Limitations

This research is primarily focused on designing and evaluating an enhanced link failure recovery mechanism tailored for multi-hop Software-Defined Exchange Point (SDX) environments. The study centers on two critical modules: path computation and packet processing, both of which play a vital role in ensuring fast, reliable rerouting when link failures occur. The scope of the research is clearly defined and constrained by the technical and architectural parameters in Table 1.1.

Table 1.1 Research Scope

No	Items	Scope of Research
1.	Environment	IPv4
2.	Target layer	Network Layer
3.	Programming Languages	Python
4.	Topology	Multi-hop Topology
5.	Network Simulator	Mininet
6.	Performance Metrics	Recovery Time, Calculating Time, Computational Overhead, Packet Processing Delay, Packet Loss Ratio.

1.6 Research Steps

All the achievements made in this research can be categorized into four significant stages that are necessary to accomplish the primary goals as mentioned in

Section 1.4. Figure 1.6 provides an overview of all the research steps involved in this study.

The **first phase** consists of the literature review section. This phase investigated the backgrounds of Internet exchange points, followed by existing challenges and issues in IXPs. Then, investigated the architectural developments of Internet exchange points (IXPs), considering the paradigm of software-defined networking (SDN), followed by the concepts of the OpenFlow protocol.

The **second phase** comprises the analysis; it begins by analysing the previous challenges that exist in software-defined exchange points, SDXes. Then reviewed existing approaches and frameworks for link failure recovery, categorising them to identify the strengths and limitations of each. Additionally, reviewed the related work of all approaches and frameworks to address the research problem.

The **third phase** comprises module design and development. This phase describes how to develop the suggested method and goes over the path computation packet processing modules to enhance link failure recovery performance in the SDX multi-hop topology, considering the information obtained and investigated in earlier phases.

Performance review is the **fourth stage**. To assess the effectiveness of the suggested mechanism, this phase covers its implementation and experimental setup. This begins with the construction of the mechanism and ends with an analysis that considers performance metrics like Computational Overhead (CO), Recovery Time

(RT), and Calculating Time (CT), Packet Processing Delay (PDD) and Packet Loss Ratio (PLR).

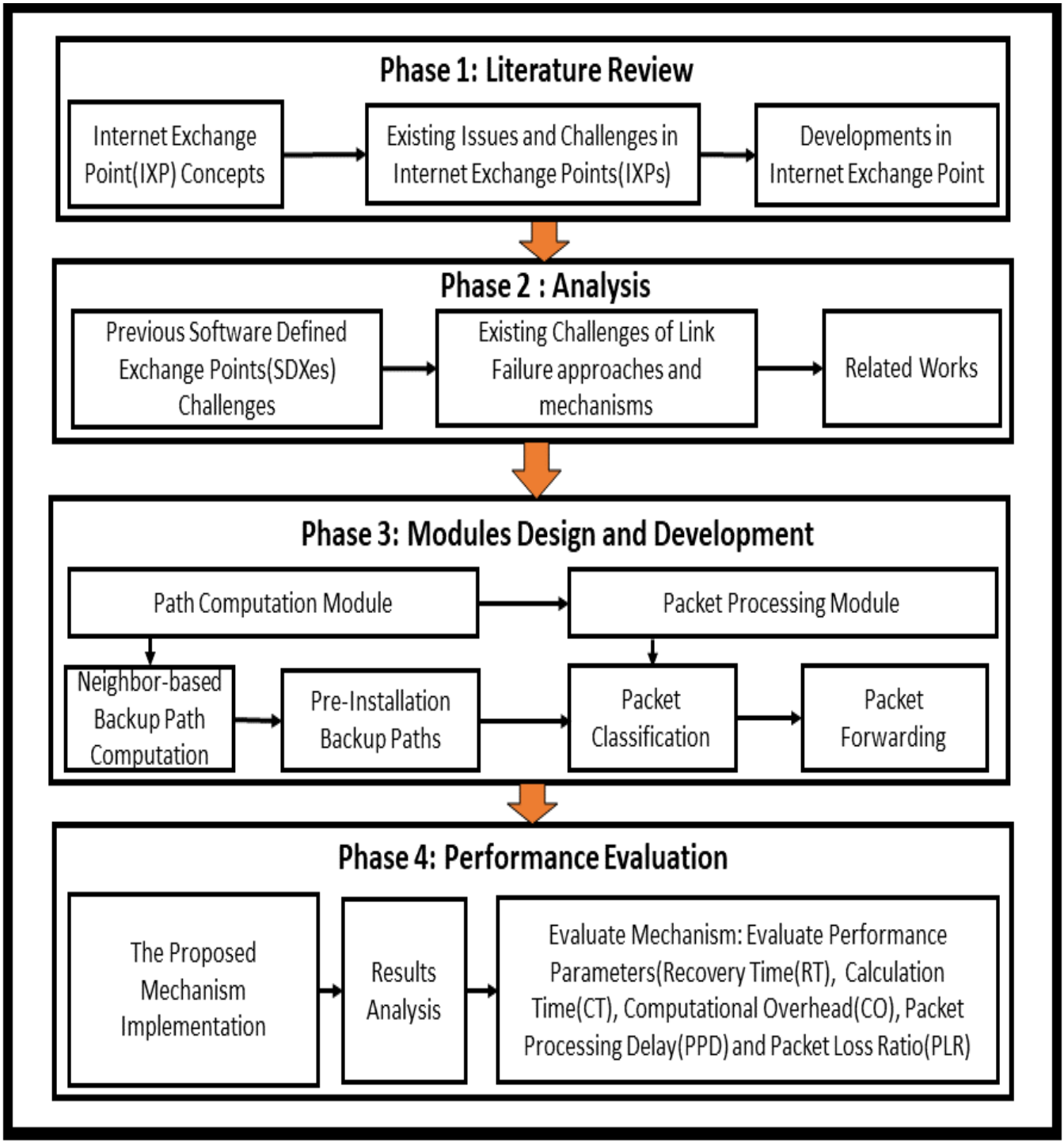

Figure 1.6 Research Steps

1.7 Research Contribution

This research enhances robust Software-Defined Exchange Point (SDX) networks by presenting a novel mechanism—Enhanced Link Failure Rerouting (ELFR)—tailored to tackle the complexities of link failure recovery in multi-hop SDX topologies. This study's main contributions are as follows:

- The design of an adaptive link failure recovery mechanism that integrates proactive control-plane intelligence with data-plane awareness to reduce packet processing delays and improve path computation efficiency in multi-hop SDX environments.
- The development of an efficient path computation algorithm that minimizes rerouting latency and reduces switch memory overhead by computing localized backup paths suitable for distributed SDX architectures.
- The implementation of an optimized packet processing algorithm that enables rapid in-switch classification and forwarding of recovery packets, thereby reducing the overall recovery time and ensuring seamless failover without controller intervention.

1.8 Chapter Organization

The structure of this thesis is organized as follows:

Chapter 1 introduces the research by presenting its background, motivation, objectives, scope, and key contributions. It defines the problem domain and establishes the relevance of the proposed work within the context of Software-Defined Exchange Points (SDXs).

Chapter 2 provides a comprehensive literature review, examining foundational concepts, related studies, and existing mechanisms for link failure recovery. It categorizes and analyzes the strengths and limitations of both SDX-based and fast reroute (FRR)-based frameworks, culminating in a critical evaluation that highlights the research gap addressed in this thesis.

Chapter 3 presents the architectural foundation of the proposed Enhanced Link Failure Rerouting (ELFR) mechanism, detailing its core modules—Packet Processing and Path Computation—along with their respective operational processes designed for failure recovery in multi-hop SDX environments.

Chapter 4 explains on the design logic, development tools, and implementation procedures used to realize the proposed mechanism. It provides a structured explanation of how the two primary modules are integrated and deployed within a simulated SDX environment using programmable data plane technologies.

Chapter 5 presents the experimental results and provides an in-depth analysis of the performance achieved by the proposed ELFR mechanism across multiple failure scenarios. It includes a comprehensive comparison with the baseline mechanism, highlighting ELFR's improvements in recovery efficiency, control overhead, and scalability.

Chapter 6 concludes the research by summarizing the key findings and outlining future research directions to further enhance the robustness and applicability of the proposed solution in real-world SDX deployments.