CHAPTER I
INTRODUCTION

1.1 Background

Internet usage is currently very high and will continue to increase every year, according to APJII, data on internet users in Indonesia in 2023 reached 221 million people out of a total of 278 million people (79.5%) and compared to the previous year there was an increase of 1.4% [1], meaning that the internet has become an important part and is difficult to separate from daily activities. However, there are times when the internet does not work properly or what we commonly call internet connectivity disruptions.

For example, it happened in 2022 where one provider's internet disruption occurred in Jabodetabek, Bandung, Garut, Bali, and Papua. Another case related to internet connectivity disruptions also occurred in the areas of East Java, Bali, Nusa Tenggara, Kalimantan, Sulawesi and Papua in 2021, caused by disruption of the submarine cable communication system resulting in a decrease in network quality and also did not occur only in one operator, other operators also experienced internet disruptions that occurred in 2024 where the network signal bar on the device was full but there was no internet connectivity [2–4]. In these 3 cases it can be seen that the handling of interference by operators is very slow so that it can have a detrimental impact on activities that require internet connectivity.

When there is a network disruption, it is necessary to handle it to restore the network to normal conditions and one of the techniques carried out is by rerouting. However, existing rerouting techniques take quite a long time and also sometimes rerouting is still done manually so that it is inefficient. Therefore, to overcome this, there is a fast rerouting technique that can do rerouting quickly and more efficiently.

There are various types of fast rerouting techniques such as Multi-Protocol Label Switching (MPLS) fast rerouting, fast rerouting in programmable networks (control plane and data plane fast rerouting) and others [5, 6]. MPLS is a protocol that utilises the Label Distribution Protocol (LDP) to establish neighbour relationships and create Label Switched Paths (LSPs). Within MPLS, there is a fast rerouting mechanism called MPLS Traffic Engineering (TE). MPLS TE is often combined with other routing protocols such as OSPF and IS-IS to determine routing paths. MPLS fast rerouting is capable of quickly recovering the network when the primary path experiences a failure, with a recovery time of ≤ 50 ms [7]. This recovery time is sufficiently fast to handle network faults effectively. However, as the number of LSPs increases, traffic loss can be affected for up to 5 seconds, or 5000 ms, which can significantly impact network performance [8]. Additionally, the MPLS fast rerouting technique currently in use has the disadvantage of relying on fixed mechanisms.

On the other hand, according to the ITU-T vision for 2030, network architectures will become highly flexible and programmable, encompassing devices such as routers, switches, and other network elements [9]. Therefore, to address the increasingly complex and dynamic network challenges, data plane fast rerouting techniques based on P4 (Programming Protocol-independent Packet Processors) programming can be employed to enhance network flexibility and programmability. This approach not only improves adaptability and dynamic network configuration, but also speeds up recovery time by up to 1 ms and improves overall network stability [10]. However, the existing P4-based fast rerouting technique uses a mechanism that requires each device to support the P4 programming language [11]. So that at the time of implementation it is necessary to replace each device with one that supports the P4 language and this will cause reconfiguration on each device and take a long time.

This research develops the P4 Adaptive Fast Rerouting (P4AFR) technique, a fast reroute method that enables rapid and adaptive decisions for selecting both primary and backup paths. The technique is designed for integration with conventional network and is implemented directly on the data plane using the P4 programming language. P4 is an open-source programming language that enables the flexible configuration of packet processing within a network [12, 13]. In addition to P4, this system also incorporates Python programming, which enables the sending of probing packets, cost calculation, and other distributed control functions to support efficient and adaptive path selection decisions.

1.2 Statement of Problem

There are several issues raised in this thesis:

1. Existing fast rerouting techniques such as MPLS fast rerouting are not flexible and adaptive in the face of network protocol developments and changes.

   The current MPLS fast rerouting technique is still fixed where its features and configurations are limited and cannot be modified as needed. The development of technology and networks today is so fast that it requires a system model that is able to keep up with technological developments, for example Named Data Networking (NDN) which is said to be the network of the future [14, 15]. NDN networks have different types of header packets than IP networks and with current MPLS technology, it certainly cannot recognize these types of packets so that it will cause problems at some point in the future. In addition, the only type of network fault that can be resolved is link failure.

2. Existing P4 programming-based fast rerouting (InFaRR) techniques cannot be integrated with common network devices.

   One example of a fast rerouting technique on the data plane uses a pushback mechanism [11]. The pushback mechanism is a technique where if there is a link failure on the network, the packet will be sent back to the previous switch until a switch is found that has a normal alternative path, of course, with this technique, general devices cannot do it. So it cannot be integrated and needs to replace each device to support the entire system.

3. Limited to detecting and avoiding only a single network issue.

   MPLS Fast Reroute, InFaRR, Swift, and others can only detect and avoid network problems in the form of link failures. In addition, other fast rerouting techniques detect and avoid congestion only [10]. However, network problems are not only link failures but also other issues such as high RTT and Congestion.

1.3 State of the Art

Fast Rerouting (FRR) is a technique for rapidly redirecting packet routes when network disturbances occur, such as Link Failures, high Round-Trip Time (RTT), or congestion. One of the most common methods is MPLS Fast Reroute. Initially, MPLS Fast Reroute selects a primary route and a backup route defined manually; however, over time, MPLS has been integrated with routing protocols such as OSPF or IS-IS. Although MPLS-OSPF Fast Reroute is reliable and fast, capable of switching routes within ≤ 50 milliseconds, MPLS has limitations: it only handles Link Failures, lacks flexibility, and struggles to adapt to evolving network conditions [16].

Additionally, other methods, such as Swift, employ a predictive approach based on IP tagging for fast reroute [17]. However, Swift risks routing loops when multiple failures co-occur and has convergence times in the order of seconds, making it less suitable for real-time requirements.

To address these limitations, a novel approach utilizing programmable data planes, specifically with the P4 language, has emerged. P4 offers high flexibility in packet processing, strong adaptability, and can switch routes in ≤ 1 millisecond [6]. Several studies on P4-based fast rerouting have been conducted, such as InFaRR and BLINK. InFaRR provides a fast reroute mechanism without generating routing overhead and employs a pushback mechanism [11]. However, this method manages registers with predetermined values to simulate link failure, which is not an actual failure. It requires all devices to support P4, which limits its practical deployment in heterogeneous networks. Meanwhile, BLINK utilizes data plane timeout-based detection, which is effective only for link failures and exhibits a detection latency of up to 1 second [18].

Therefore, this research proposes the P4 Adaptive Fast Reroute (P4AFR) method as a solution that combines the advantages of P4-based rerouting and dynamic control, implemented using Python. P4AFR is designed to handle multiple types of network disturbances beyond a single failure, including link failure, high RTT, and congestion. Furthermore, P4AFR maintains compatibility with legacy devices while delivering extremely fast recovery times.

1.4 Objective

The objective of this research is to achieve several specific goals:

1. In this thesis, it uses the P4 programming language to propose a more flexible and adaptive fast rerouting technique, called P4AFR.

2. The proposed method (P4AFR) can be integrated with common network devices.

3. The proposed method (P4AFR) is capable of detecting and avoiding network problems such as high RTT, link failure, and congestion.

1.5 Research Method

In this research, we propose a P4AFR method that can reroute quickly when network faults occur and can be integrated with devices that do not support P4 (legacy switches). To create P4AFR, we use two methods, namely network fault detection and rerouting on the data plane and generate a control function for route discovery, cost calculation and others. Both methods will be written in P4 and Python programming languages.

Network fault detection in the data plane helps increase the detection speed in real-time, such as in networks with high RTT and congestion, which indicate a fault on the network. The rerouting technique in the data plane is used to reroute quickly because it is at the bottom layer of the OSI layer (Data-Link Layer).

1.6 Hypothesis

In previous research, fast rerouting techniques were able to solve the problem of connectivity disruption in the network with a fast response time. However, the current fast rerouting technique is still relatively traditional and limited in the face of network development. As a solution to this, fast rerouting in the data field based on P4 programming is used. By using P4 programming, fast rerouting in the data plane will become more flexible and adaptive. However, the research that has been done requires each device to be replaced with a device that supports P4 programming. Therefore, in this research, we create a P4 Adaptive Fast Rerouting (P4AFR) technique that will run on the data plane based on P4 programming that is able to provide flexibility and overcome several types of network errors and can be integrated with existing devices.

1.7 Research Methodology

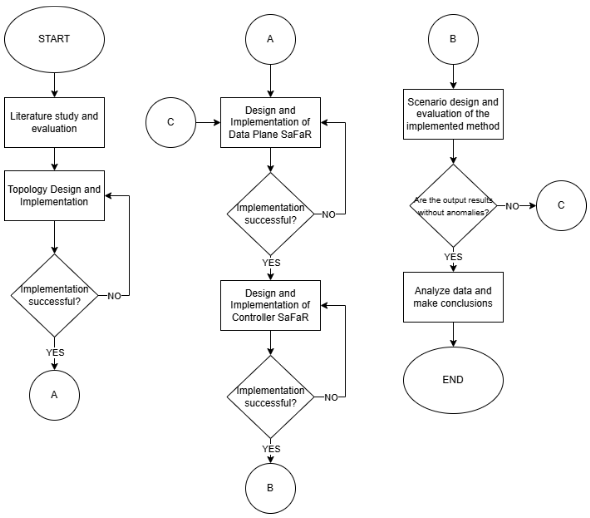

Fig. 1.1 Flowchart of research methodology.

The research stage series can be seen in Figure1.1. The research begins with a literature study to gather information and evaluate the system to find gaps that can be improved. Next, the topology design is carried out, which will be used for implementation and evaluation. If the implementation is unsuccessful, the topology design will be redesigned to repair the previous error. Then, after the topology implementation is successful, the network fault detection and rerouting functions in the data plane will be designed and implemented using the P4 programming language. However, a redesign will be carried out if the implementation is unsuccessful.

After the successful implementation, the controller design will be carried out to help the data plane carry out the control function. If the implementation is successful, an evaluation of the method used will be carried out, as well as analysis and conclusion-making, to determine whether the technique has been successfully carried out and whether the results are better when compared to the previous method. After the analysis and conclusion, the research is completed.