CONVERGENCE TIME MONITORING ALGORITHM IN HYBRID SOFTWARE DEFINED NETWORKS

BY

YOUSEF ABDO NAJI ALTAREB

A dissertation submitted in fulfilment of the requirement for the degree of Master of Science (Communication Engineering)

Kulliyyah of Engineering
International Islamic University Malaysia

DECEMBER 2020

ABSTRACT

Network performance is extremely dependent on traffic monitoring. Therefore, Software Defined Network (SDN) technology is proposed to support the flow control and proper monitoring by providing a global view of the network. Unfortunately, replacing the entire traditional network to SDN is complex, which leads to the need of SDN switches deployment to the current network. Thus, a hybrid network environment has been emerged which consists of centralized controller, SDN switch and legacy routers. Hence, the advantage of the integration of traditional network and SDN have been taken place. The controller can collect SDN data instantly, while it waits for a long time to obtain the legacy network data. Consequently, failure detection and traffic management cannot be recognized in real-time. This research proposes a monitoring algorithm to monitor path state. It aims to reduce infrastructure cost in terms of replacing minimum number of legacy routers and minimizing the convergence time of collecting path load data. Significant paths are chosen by reconstructing load matrix using Singular Value Decomposition (SVD). SDN switches are then installed to cover these critical paths. As a result, critical paths can be directly addressed by the controller. On the other hand, the rest of the paths cannot be processed directly by the controller. Therefore, legacy path load data is estimated for the past time to support the controller for obtaining the current data. The proposed algorithm has been implemented over ISP topology of 24 nodes and 72 paths using Mininet emulator with Quagga routing software and Python-based open-source OpenFlow (POX) controller. The convergence time of the proposed algorithm takes only 12% more convergence time than the full SDN. Therefore, the proposed algorithm provides replacing one-third of legacy routers (8 out of 24) to SDN switches where the infrastructure does not need to be fully replaced which reduce the infrastructure cost.

خلاصة البحث

يعتمد أداء الشبكة وجودتها إعتمادا كبيرا على مراقبة حالة حركة المرور. ،وهكذا تم اقتراح تقنية الشبكة المعرفة برمجيا SDN لمراقبة التدفقات وادارتها بشكل افضل مركزيا. بالمقابل، إن تغيير الشبكة التقليدية بالكامل معقد بسبب قضايا التكلفة والصيانة، حيث يؤدي ذلك إلى الحاجة إلى نشر أجهزة SDN على الشبكة الحالية. بالتالي تنشأ لدينا بيئة شبكة هجينة تتكون من المتحكم المركزي وأجهزة تقليدية و SDN. الشبكة التقليدية تستفيد بشكل ملحوظ من دمجها مع اجهزة SDN. المتحكم يستطيع أن يجمع البينات من اجهزة SDN مباشرة ولكنه ينتظر وقتا اطول للحصول على معلومات الروابط التقليدية. كنتيجة فإن هذا يسبب تأخير في معرفة مشاكل الروابط وادارة حركة مرور الشبكة في الوقت الفعلي. هذا البحث يقدم خوارزمية تقوم بمراقبة حالة الروابط بالكامل. تهدف هذه الخوارزمية إلى تقليل تكلفة البنية التحتية من حيث استبدال الحد الأدنى من عدد أجهزة التوجيه القديمة وتقليل وقت التقارب لجمع بيانات تحميل الروابط. يتم اختيار اهم الروابط التي تحمل اكثر بيانات عن طريق اعادة تركيب مصفوفة حمولة الروابط. ومن ثم يتم تركيب اجهزة SDN لتغطية هذه الروابط المهمة. كنتيجة فإنه يمكن معالجة هذه الروابط مباشرة عن طريق المتحكم. بالمقابل فإن بقية الروابط لايمكن معالجتها مباشرة من المتحكم، وبالتالي التنبؤ ببيانات الروابط التقليدية للفترة الماضية تساعد المتحكم للحصول على معلوماتهم في الوقت الحالي. تم تقييم وقت التقارب للخوارزمية المقدمة بحيث سجلت 0,8 ثانية. تم تنفيذ الخوارزمية المقترحة باستخدام محاكي Mininet مع برنامج توجيه Quagga ووحدة تحكم POX مفتوحة المصدر تعتمد على Python. تستغرق هذه القيمة وقت تقارب أكبر من SDN الكامل بنسبة 12٪ فقط. وبالتالي ، توفر الخوارزمية المقترحة استبدال ثلث أجهزة التوجيه (8 من 24) بمفاتيح SDN حيث لا تحتاج البنية التحتية إلى الاستبدال الكامل مما يقلل من تكلفة البنية التحتية.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Master of Science (Communication Engineering).

…………………………………..
Aisha Hassan Abdalla Hashim
Supervisor

…………………………………..
Othman Omran Khalifa
Co-Supervisor

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Master of Science (Communication Engineering)

…………………………………..
Md. Rafiqul Islam
Internal Examiner

…………………………………..
Malik Arman Morshidi
Internal Examiner

This dissertation was submitted to the Department of Electrical and Computer Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science (Communication Engineering)

…………………………………..
Mohamed Hadi Habaebi
Head, Department of Electrical and Computer Engineering

This dissertation was submitted to the Kulliyyah of Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science (Communication Engineering)

…………………………………..
Sany Izan Ihsan
Dean, Kulliyyah of Engineering

DECLARATION

I hereby declare that this dissertation is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Yousef Abdo Naji Altareb

Signature .... Date ....

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

CONVERGENCE TIME MONITORING ALGORITHM IN HYBRID SOFTWARE DEFINED NETWORKS

I declare that the copyright holders of this dissertation are jointly owned by the student and IIUM.

Affirmed by Yousef Abdo Naji Altareb

……..……………………..	………………………..
Signature	Date

ACKNOWLEDGEMENTS

In the name of Allah, the Most Merciful and Most Compassionate.

Praises are due to Allah S.W.T, the Exalted, for His guidance, provisions, mercy, and blessing. Peace and blessings be showered upon our Holy Prophet Muhammad PBUH, his family, and companions.

First, I would like to express my gratitude to Allah S.W.T for His mercy and blessing that I managed to complete my master dissertation.

I submit my heartiest gratitude to my supervisor, Prof. Dr. Aisha Hassan Abdalla Hashim for her guidance and continuous support that has been given. I also would like to thank my co-supervisor, Prof. Dr. Othman Omran Khalifa. for his advice and support for me to complete my dissertation.

Moreover, it is my utmost pleasure to dedicate this work to my dear parents, Mr. Abdo Altareb and Mrs. Fuziah Yahya and my family, who granted me the gift of their unwavering belief in my ability to accomplish this goal: thank you for your support and patience.

In addition, I sincerely appreciate my best friends Amjad Hussain Sufian and Taiba Majid for their cooperation and time, which enabled me to complete my research. Without their help, it is far beyond possible to complete this research

Finally, I humbly extend my thanks to my friends for their various contributions and sincere encouragement throughout completing this dissertation. May Allah S.W.T bless all of them and grant them with Jannah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FNSS	Fast Network Simulation Setup
FCFS	First Come First Serve
Gbps	Gigabit per second
GUI	Graphic User Interface
IoT	Internet of Things
IPv6	Internet Protocol version sixth
LBP	Learning Bridge Protocol
LSP	Link State Packet
MPLS	Multiprotocol Label Switching
NOS	Network Operating System
ONF	Open Networking Foundation
OSPF	Open Shortest Path First
QoS	Quality of Service
SDN	Software Defined Network
SNMP	Simple Network Management Protocol
STP	Spanning Tree Protocol
SBL	Sparse Bayesian Learning
SAQR	Simulated Annealing based QoS-aware Routing
SVD	Singular Value Decomposition

LIST OF SYMBOLS

s	Second

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Nowadays, the steady improvement of Internet speed with a data transfer rate of tens to hundreds gigabits per second (Gbps) is noticed (Benzekki, El Fergougui, & Elbelrhiti Elalaoui, 2016). Big servers and cloud computing solved partially storing and processing massive data problem. Besides, the sixth version of Internet addresses (IPv6) has solved the shortage Internet addresses shortage. However, for a reliable, secure, simple, and low-cost network, the infrastructure of the existing network has to be changed. From this point of view, researchers started looking for alternative solutions, then the emerging technology known as software-defined networking (SDN) was found. It is the next generation of infrastructure in network engineering that supports a traditional network (Karakus & Durresi, 2017). Therefore, cloud computing, Internet of Things (IoT), big data, and the increasing demands for networks explain the need of SDN.

Traditional networks were not designed to handle large amount of data even with increased supply in processing power of the ethernet speed, wireless speed, 4G and 5G technologies. The traditional network infrastructure has not been changed due to its scalability and stability until SDN emerged in the past few years. However, it encounters some obstacles such as human errors during manual configuration, delay of packets because of distributed control etc. In addition, the cooperation between two different vendors does not exist (S. Huang, Zhao, & Wang, 2016). For example, Cisco protocols are not compatible with Juniper or Huawei protocols.

SDN solves the previous problems by the centralization of the controller that provides a global view of the network (Chen, Wang, Chung, & Chou, 2017). Moreover, the centralized point can manage the entire network to minimize errors and provides consistency during configuring rules on the network devices. The SDN centralized controller improves the security and quality of service (QoS) through enforcing some safe network rules that limit risks. For example, if any link in the network goes down, the controller instantaneously recognizes the problem, unlike the traditional networks that take a long time to detect the problem (Amin, Reisslein, & Shah, 2018). SDN is an effective scheme to save time and effort as well as provides high quality and reliable networks. However, practically it is difficult to move straightforward to SDN because of high cost of changing the entire infrastructure of the existing network (Lin, Wang, & Deng, 2017). Consequently, hybrid network has emerged with the combination of SDN and conventional networks. The hybrid network improves the network performance and facilitates its management where it takes the advantages of both networks (Sinha & Haribabu, 2017).

The traditional network lacks QoS due to multiple control sources. In the beginning, the Best-Effort approach was released, which depends on the First Come First Serve (FCFS) basis. But this approach is not suitable for real-time applications such as VoIP and video conferencing (Sinha & Haribabu, 2017). Multiprotocol Label Switching (MPLS) and DiffServ mechanisms were then established to simplify network management and to improve the QoS in real-time (Salman, Elhajj, Chehab, & Kayssi, 2017). In contrast, the quality of network performance is difficult to be achieved due to the distributed routers' control. Therefore, these challenges are solved by the controller centralization, which has a global view of the entire network (Amin et al., 2018). As mentioned earlier, it is not possible to move directly to SDN. Thus, it is necessary to monitor the path load state and avoiding any danger before it takes place to reach the best performance in the hybrid SDN (Tsai, Tsai, Hsu, & Yang, 2018). The controller finds it complicated when it comes managing the legacy devices over a large-scale network because gathering the whole network information takes a long time. Therefore, convergence time is an essential measure to help the controller to obtain the entire network status. Convergence time is the required time for routers to capture each other's information, such as links, devices, and routing tables information. There are several research efforts in the area of monitoring the traditional network (Li, Gao, Dong, & Chen, 2018) and pure SDN (Luong, Outtagarts, & Hebbar, 2016). However, less attention is given to hybrid network. Therefore, this research focusses on addressing hybrid networks issues and solutions.

1.2 PROBLEM STATEMENT

As mentioned earlier, SDN is the optimal solution for traditional network shortcomings. However, it is very costly to migrate from the traditional network to SDN immediately. Therefore, SDN devices are deployed gradually into legacy devices. There are numerous studies being carried out about monitoring on pure SDN and traditional networks. However, the monitoring in hybrid networks is still under-research due to a lack of studies in this field.

It is not easy to achieve full traffic management such as load balance, congestion, and latency for hybrid SDN because the entire network is not under single control. The centralized controller of SDN can easily manage OpenFlow switches and recognize the paths status in real time. Due to multiple Interior Gateway Protocol (IGP) routers in the traditional network part, collecting path load state data for large-scale network topology takes long time. Thus, paths monitoring is a critical issue. It is important to speed up the convergence time to help the controller to make decisions promptly when dealing with any changes in the network.

1.3 RESEARCH OBJECTIVES

This research aims to enhance the convergence time and reduce infrastructure cost associated with hybrid SDN. While the specific objectives are:

1) To propose a monitoring algorithm for real-time communication with less infrastructure cost.
2) To implement the algorithm using Mininet emulator and POX controller.
3) To evaluate the performance of hybrid SDN and compare it with a Full SDN.

1.4 RESEARCH METHODOLOGY

The process of network convergence time improvement consists of four steps. First, gathering paths status data from the traffic matrix. Second, determining the most critical paths that may cause latency. Third, replacing required legacy routers to SDN switches to cover the critical paths. Finally, analysing the past period legacy paths data to estimate the state of paths in the current period. Each of these steps will be explained in Chapter 3. This study involves simulation technique by using Mininet emulation and POX controller.

The approach followed in this research is shown in Figure 1.1.

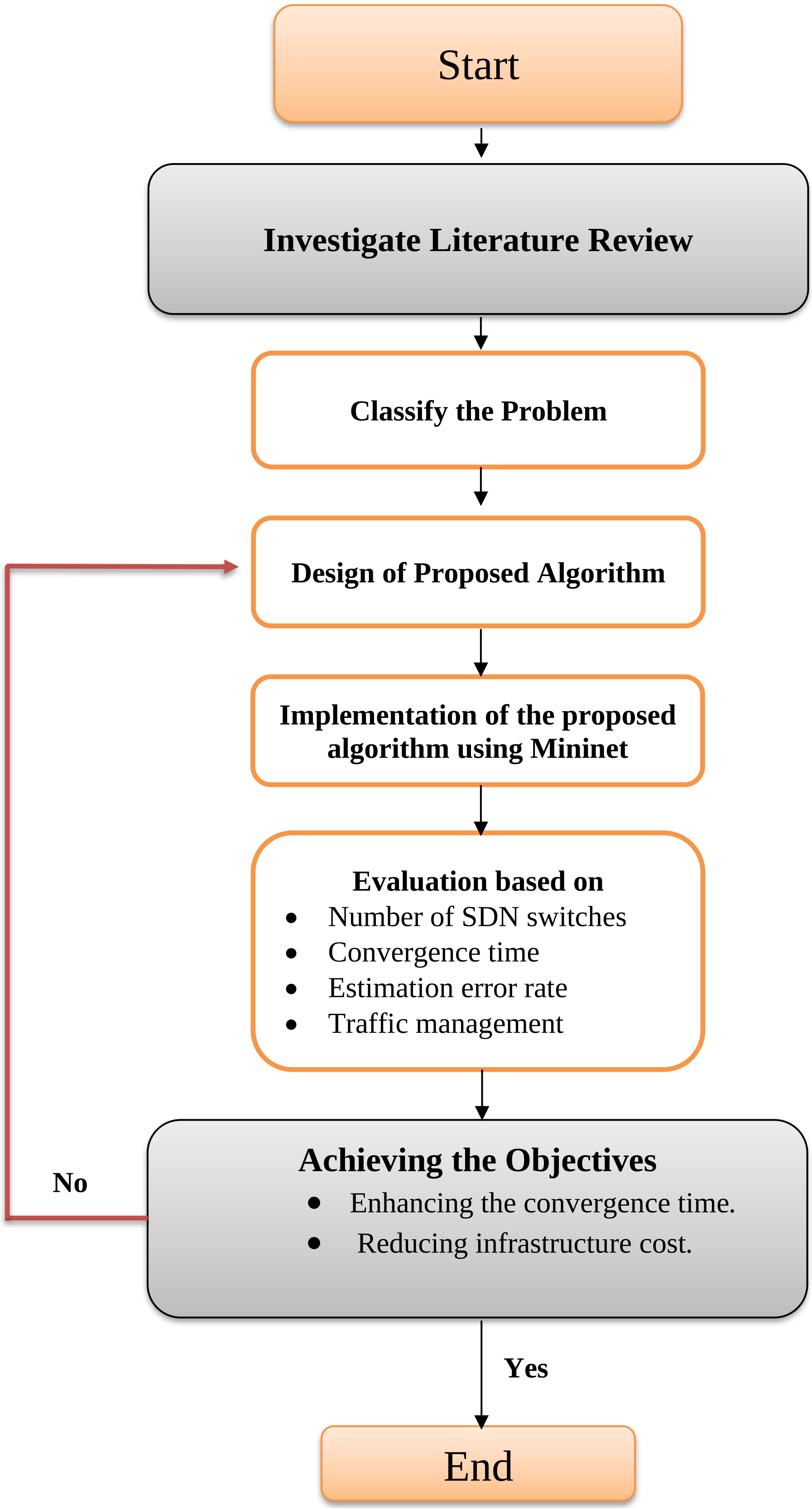


Figure 1.1: Methodology of research

1.5 SCOPE OF RESEARCH

In this research, a timely manner paths load data gathering has been implemented in hybrid SDN. This study proposes a method concentrating on choosing the most critical paths data and cover it by SDN switches. This study covers layer three convergence time, such as Open Shortest Path First (OSPF) convergence time. However, layer two convergence time, such as Spanning Tree Protocol (STP), is not taken into consideration because this issue has been addressed by many researchers.

1.6 THESIS ORGANIZATION

The remaining chapters in this research are organized as follows:

Chapter Two discusses the background of network monitoring involving hybrid SDN, full SDN, and traditional networks. In addition, related works to our study also are briefly discussed.

Chapter Three presents the design and simulation setup of the proposed algorithm using Mininet emulation and POX controller. Figures are used extensively to illustrate the outcomes of these steps and to show the build-up process towards achieving the research objectives.

Chapter Four describes the evaluation of the proposed algorithm. It presents results analysis and discussion.

Chapter Five presents the conclusion, contribution of the study and the future recommendation.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This section consists of three subsections: i) Monitoring network background ii) Hybrid network, and iii) Related Works. The first two subsections present basic terminologies used in this study, which mainly deals with the management of hybrid SDN. The latter section discusses related works to our research.

2.2 MONITORING NETWORK BACKGROUND

To achieve efficient network management, traffic status monitoring is an essential pillar to obtain high quality and stable networks (M Coates, HERO, Nowak, & Yu, 2002). Network monitoring shows the network behavior status via traffic statistics. With the increased Internet usage and growing a large number of applications, the traditional network suffers from some network performance weaknesses due to a lot of application requirements (Tsai et al., 2018). Traffic matrices are useful for planning and monitoring of network behavior. Therefore, there are two popular methods to monitor traffic flows in real networks, NetFlow application and Simple Network Management Protocol (SNMP). Moreover, SDN is an ideal solution to simplify network monitoring via centralized controller that has a global network view.

Network monitoring is divided into two parts: data measurement and data processing (Lee, Levanti, & Kim, 2014). Measuring data is based on three initial stages, then it is sent to the next two stages in the data processing part. First, the data collection stage, which collects the network information. Second, the preprocessing stage puts the gathered data in a statistical format. Next, this format is sent to the analysis unit by the transmission stage. In the fourth stage, the analysis stage analyzes the network status information. Finally, the stage of presentation, that announces and displays the results. Figure.2.1 shows a summary of network monitoring stages.

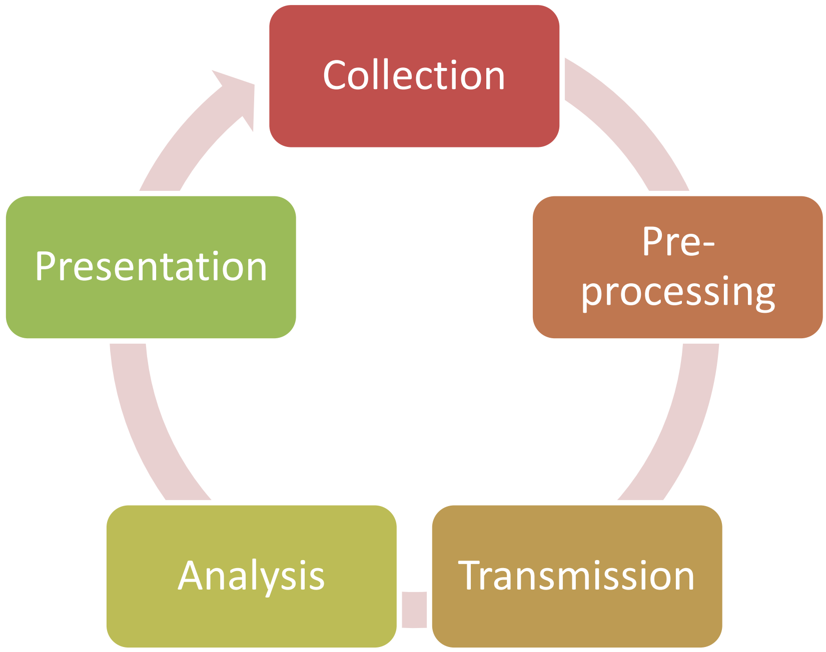


Figure 2.1: Network monitoring stages

2.2.1 Software Defined Network

Network devices are typically consisting of the data plane and control plane. A data plane is the hardware responsible for forwarding packets while the control plane provides the network intelligence such as selecting the best path, setting priorities, and policies. In addition, each device in the traditional network is managed individually and manually, such as a router or switch. However, SDN architecture took out network devices intelligence (control plane) to form a central controller. Therefore, there is a separation between the control and data planes of SDN (Karakus & Durresi, 2017). The control plane is responsible for decision-making such as packet routing, while the data plane executes control plane instructions. The result of separation is that the management function concentrates on a single network device known as the controller, which monitor the network easily.

Migrating to SDN transforms network management from running services on each device of traditional networks to simply programming services on management and control devices. The centralized controller activates services as well run and control them in all required network devices automatically. The programmable controller facilitates the network to be more flexible in design, application, and management. Besides, it improves security, time, speed, reliability, and troubleshooting (Amin et al., 2018).

2.2.1.1 SDN Architecture

Open Networking Foundation (ONF) organization, which is supported by some big companies such as Google, Facebook, and Microsoft, aims to develop networks by SDN and OpenFlow protocol (Ijaz Ahmad et al,2015). ONF divided the SDN network into three layers (Xinli Huang et al., 2018), as shown in Figure 2.2.

1. **Application layer:** the application layer is the first SDN layer that provides users' services and applications. It can communicate with the control layer through the north -API.
2. **Control layer:** the second SDN layer represents the network controller, which controls, manages, and supplies instructions to all network devices. The controller can communicate with the data-layer via OpenFlow protocol (south-API).
3. **Data layer:** the last SDN layer which contains several types of devices. It executes the controller's instruction, such as routing.

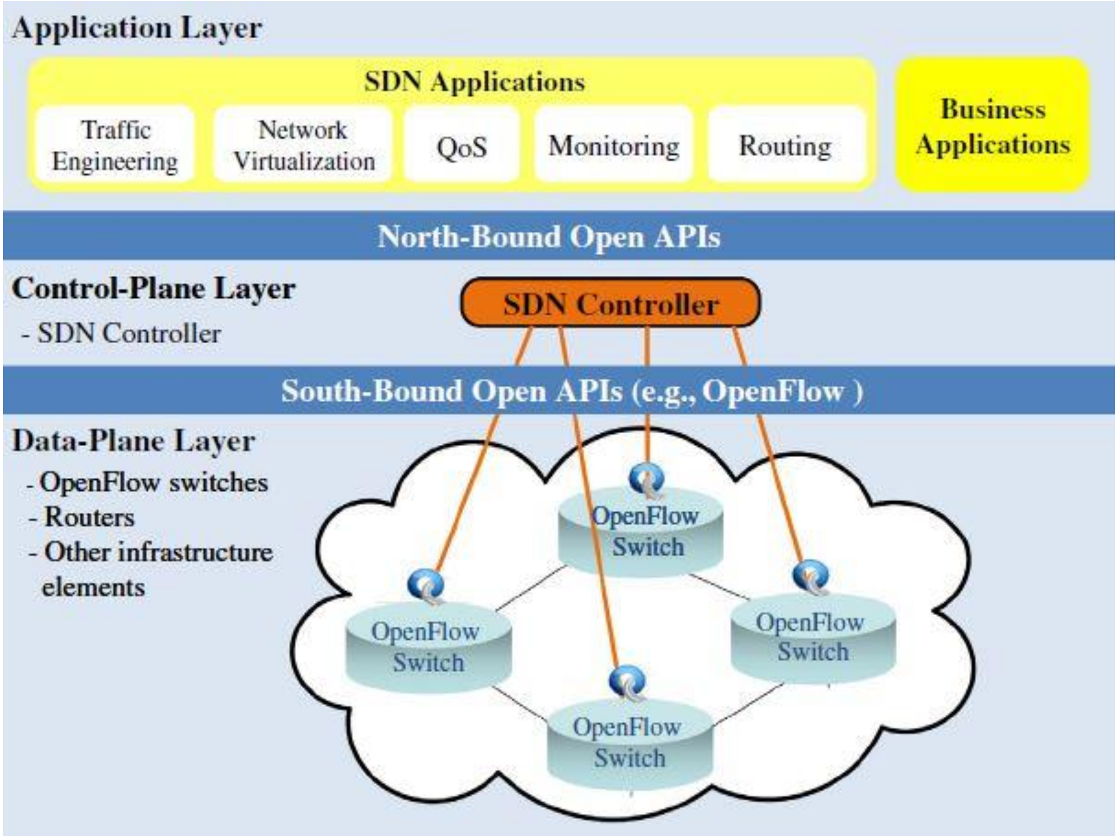
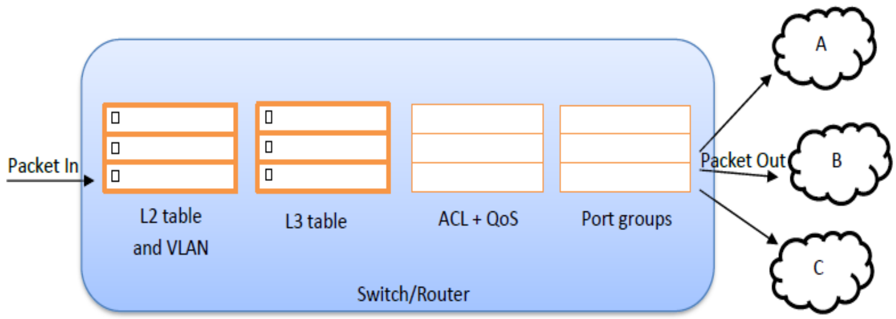


Figure 2.2: SDN architecture. (Akyildiz, Lee, Wang, Luo, & Chou, 2014)

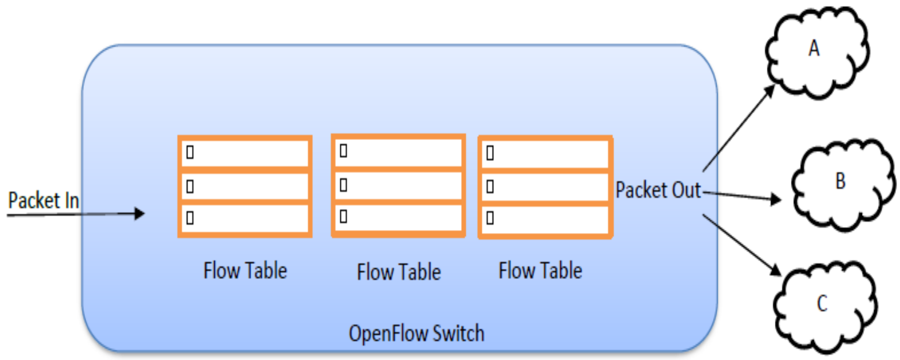
2.2.1.2 OpenFlow Protocol

OpenFlow protocol is the most popular protocol that is used in SDN (Diego Kreutz et al.,2015). It allows control layer to contact with data layer and exchange packet with each other. It has been developed to suit all vendor types. Figure 2.3 shows the difference in the internal structure of traditional and SDN devices. SDN is a more modern approach to manage actions and services. The network design is one of the prominent advantages that supports different vendors' devices which can interact with each other.

If the new packet does not match flow table entries, packets are handled by the controller. The controller handles these entries by creating new actions, then update flow tables. The flow table is a key element of the OpenFlow process to improve network performance as well as avoiding congestion and collision (Benzekki et al., 2016). Traditional networks rely on IP addresses as well as multiple sources of control and management. In contrast, SDN controls all devices in a data plane by a single centralized controller. Thus, when both networks are integrated, legacy and SDN systems must be appropriate to communicate with each other, either through software updating or hardware deployment.



a) Tables of actions in traditional devices



b) Flow tables in OF-Switch

Figure 2.3: The difference in the internal structure of traditional and SDN devices (Amin et al., 2018)