Software Defined Networking:

A case study in College of Science and Technology (CST)

[1]Tshering Dorji, [2]Kencho Bidha, [3]Berandra Rai, [4]Ngawang Choden, [5]Yeshi Wangchuk

Department of Information Technology, College of Science and Technology, Royal University of Bhutan

Abstract—As a means to ease the functionality of networking, Software Defined Networking, commonly abbreviated as SDN, emerged as an effective and efficient architecture that helps to optimize the network resources via dynamic and automated SDN programs along with providing ease of management and configuration. This paper consists of a series of steps that were taken to conduct the feasibility of implementing SDN at CST by studying and analyzing the existing network topology of the college through simulation. The selection of the software tools for the simulation were made based upon the literature review conducted, out of which, GNS3 as the network simulator and OpenDayLight as the SDN controller stood out to be a fine choice. The simulation was held in two phases whereby the first phase included simulating the existing network topology of the college using GNS3 network simulator and the second phase included simulating the existing network topology using OpenDayLight SDN controller. In both the phases, VMware Workstation pro was used as the hypervisor on which the simulation was carried out efficiently. The purpose of two simulations was to help realize the bottleneck in the current networking system of the college and move towards better networking architecture.

Index Terms—SDN, ODL, GNS3, Simulation, Controller, Ubuntu, Mininet, Legacy network

---

I. INTRODUCTION

Most of the developing and underdeveloped nations were still using the legacy network despite knowing the fact that new technologies and approaches have already been surfaces out. However, most companies and institutes were found to be reluctant to migrate towards trending technologies since they cannot afford to dump the already existing infrastructures and simply invest in the new technologies. According to the systems and network operator of the College of Science & Technology (CST), the college still used the legacy network in the campus. By legacy network, it means that the configuration must be done on each and every device manually. There were a handful of disadvantages associated owing to which most companies and institutes in the other parts of the world have migrated to other alternatives of networking. One such alternative is Software Defined Networking. Introduced in 2011, it is a technology in the field of Networking, which works with the use of 'software-based controllers or APIs for interacting with the base physical infrastructure and routing the internet traffic' [12]. SDN allows the administrator to control the entire network from a centralized working area without the involvement of any manual procedures. For the purpose of checking the feasibility of implementing SDN in CST, a simulation of the existing network topology was carried out using (Graphical Network Simulator) GNS3 simulator. From the simulation, a redefined network topology of CST was proposed and recommended.

II. LITERATURE REVIEW

One of the most potential future Internet possibilities has been identified as software-defined networking. It distinguishes itself by two characteristics i.e., provision of programmability in network application development and distinguishing the control plane from the data plane. Legacy systems that rely on human setup of proprietary equipment are known to be time-consuming and prone to mistakes [13]. SDN was introduced in order to simplify and secure network. It separates the control plane and the data plane in order to achieve scalability and agility in managing the network. To fulfill its infrastructure aims, it relies on elastic cloud architectures and dynamic resource allocation [8]. The two types of newly created networking paradigms that help in transforming the way legacy networks work are SDN and Network Function Virtualization (NFV). SDN is based on the principle of centralization, which means that all network-control decisions are made in a single location whereas, NFV works through the virtualization of network functions as software operating on a high-end server [9]. SDN is a novel concept that attempts to make networking devices more efficient and faster in order to enable a variety of applications. The main goal lies in separating the coexisting control and data plane. This gives the control plane more access and intelligence in order to handle various network devices efficiently and with little labor [10]. SDN is still in its early phases, but it is a dynamic, controlled, cost-effective, and adaptable architecture. This method separates network control and forwarding processes which in turn allows the abstraction of the network infrastructure which can then be used by the network services and applications. The protocol is critical to the development of SDN systems.

III. METHODOLOGY

The problem statement was framed based on the current networking status of the college campus whereby it was found that the college used a legacy network which required manual configuration and a lot of cost were incurred for the maintenance. To address this problem, as a better alternative of networking, SDN was suggested. To supplement the alternative suggested, a literature review on relevant papers and findings related to the project were carried out.

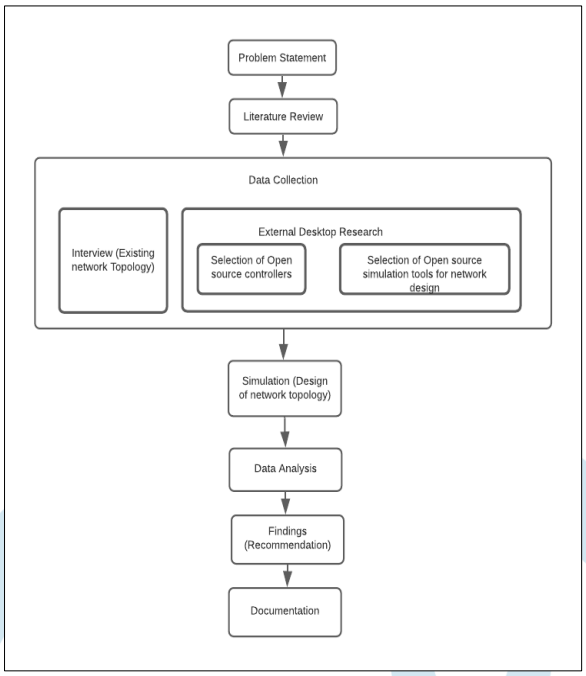

Figure 1:Methodology

The data collection process included interviewing the ICT officer of the college from time to time and exploring various open-source software required for the project using the external desktop research. From this external desktop research, GNS3 and OpenDayLight (ODL) were selected as the tools that were to be used for this project.
With the tools selected during the data collection, the simulation of the existing network topology of the college was conducted in two different phases. The first phase included simulating using the GNS3 network simulator. The second phase included simulating using the ODL SDN controller which was an SDN specific simulation.
The existing design of the CST network topology and its shortcomings were analyzed by comparing with the simulated designs that were produced. After having collected and analyzed the data, the new design of network topology for CST was recommended. Based on the research carried out, the outcomes and the findings were recommended for the reliable and robust network system. Finally, the findings were compiled into the documentation.

IV. COMPARATIVE ANALYSIS FOR NETWORK SIMULATOR

In order to select the best network simulator for this project, a number of comparison parameters were set. The following are the parameters based on which the comparison among the network simulators were carried out as shown in fig 2.

Open source

Open source network simulators can be used for simulation of both wired and wireless networks. One of the main uses of it would be that the source code comes with affiliated packages and no limitations have been set on its interfaces which will keep it open for future improvements as well [6].

Suitable for wireless networks

Wireless network means network connection done without wired connection. Wi-Fi is an example of a wireless network. This parameter verifies to see if the network simulators have support for wireless network connectivity.

Emulates real network interface

The purpose of emulation in general is to test and validate a simulation model with the real-world counterparts, or vice versa, in due course of development of a real-world implementation, to test it in a simulated model [6].

User friendly interface

An interface which allows the users to easily understand and navigate intuitively through the application efficiently is said to be a user-friendly interface [3].

Parameter	Packet Tracer	GNS3	NS3	Mininet	OMNeT++	NS2	JisT
Open Source	No	Yes	Yes	Yes	Yes	Yes	Yes
Program Language	Javascript	Python	C++/Python	Python	C++	C++	Java
Suitable for wireless network	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Emulates real network interface	Yes	Yes	No	Yes	Yes	Yes	Yes
User Friendly Interface	Yes	Yes	No	Yes	Yes	No	Yes
Platform	Linux, Windows, Mac	Linux, Mac, Windows	Linux, Mac	Linux	Windows	Linux, Mac	Linux, Mac, Windows
License	Proprietary	GPL	GPL	BSD	GPL	GPL	Apache

Figure 2: Comparison table for Network Simulation tools

The number of network simulators chosen for this project was narrowed down to top seven which were then compared based on the comparison parameters set. The parameters were set based on the needs of the project. Upon comparing the network simulators, it was found that GNS3 qualified the maximum number of parameters among other network simulators. Hence, GNS3 was selected as the network simulator for this project.

The GNS3 is a network simulation tool which allows users to view and import software images from many manufacturers. The GNS3 VM was known to be the most popular method for importing images. The simulator's most exciting feature was that it acted as a whole network by bringing in a network device.

V. COMPARATIVE ANALYSIS FOR SDN CONTROLLER

The following are the parameters based on which the comparison among the SDN controllers were carried out and shown in fig.3.

GUI

In contrast to the traditional command-line or text-based interfaces, GUI allows the users to develop interactions, which can be accessed with the help of visual indicators which is inclusive of menus and icons [11].

Programming Language

The controllers are built either by using just one programming language or by using a combination of multiple programming languages in their core and modules. Under certain conditions, a controller that is built using multiple languages was said to have higher performance with efficient memory allocation and the ability to be executed on multiple platforms.

Architecture

The architecture of a controller can either be centralized or distributed whereby a centralized controller was popular among the networks of smaller scale while a distributed controller was accessible across numerous domains. The architecture of a controller can be further classified as flat or hierarchical. The controller with a flat architecture had equal responsibilities assigned to all its controller instances whereas a hierarchical controller consisted of a root controller.

Programmable Interface

Controllers can have different Application Programming Interface (APIs) each with different functions. Northbound API (NBI) contributed to the controllers while the Southbound API (SBI) enabled the interactions between the controller and the SDN that enabled the routers and switches whereas East-West API used in a distributed or hierarchical environment to form peers coming from different domains of multiple controllers.

Platform and Interface

The compatibility of the controllers with specific operating systems during the implementation is defined by platform and interface properties. Some of the controllers have graphical interfaces while others have web-based interfaces for the use by administrators to configure and view statistical information.

Threading and Modularity

Controllers can be single-threaded or multiple-threaded based on the suitability for deployments. For lightweight SDN deployments, a single-threaded controller were preferred whereas in case of commercial purposes multi-threaded controllers were more suitable. In terms of modularity, which is the ability of a controller to allow different applications and functionalities to be integrated, a higher modularity was preferred since a controller with higher modularity permitted faster execution of tasks in a distributed environment [14].

License, Availability, and Documentation

When it comes to licensing, some of the controllers discussed here were licensed as Open Source while others contained a proprietary license. However, the source code of the controllers were available online, further changes could be made as per the

requirements by anyone. It could be observed that the majority of the controllers had a lack of proper documentation while assessing them online. Nevertheless, the few controllers that were updated on a regular basis were found to include current and detailed documentation for all available versions, as well as community-based support [14].

Parameters	ODL	FloodLight	RYU	ONOS	NOX	POX
Introduction Year	2013	2013	2013	2014	2009	2009
GUI	Web based	Web based Java	Python	Java	Python +qt4	Python +qt4
Programming Language	Java	Java	Python	Java	C++	Python
Platform support	Linux, Mac, Windows	Linux, Mac, Windows	Linux	Linux, Mac, Windows	Linux	Linux, Mac, Windows
Documentation	Good	Medium	Medium	Good	Medium	Poor
Architecture	Distributed Flat	Centralized	Centralized	Distributed Flat	Centralized	Centralized
North-Bound API	Rest, RestConf,XMPP ,NetConf	Rest, Java RPC,Quantum	Rest	Rest,Neutron	ad-hoc	Rest
South-Bound API	Open Flow	Open Flow	Open Flow	Open Flow	Open Flow	Open Flow
License	EPL	Apache	Apache	Apache	GPL	Apache
Multithreading	Yes	Yes	Yes	Yes	Yes(Nox-MT)	No

Figure 3:Comparison table for SDN controller

As in selecting a network simulator, the same method of comparison was used to choose the SDN controller which was based on required parameter, which included parameters such as GUI, programming language, platform support, documentation, architecture, and two APIs, license, and multi-threading, which were determined based on the project specifications. As a simplified method, the parameters were compared in the tabulated form, which made things simpler to determine the best GNS3 controller. According to the comparative table, ODL supported all of the required parameters. As a result, ODL was chosen as the SDN controller, which satisfied the requirements of the project.
It provides open NBI that can be used by applications and SBI to control the underlying devices. Those same applications simply used controllers to collect network data, ran analytics algorithms, and created new network rules using it. It is a standalone software device with its own Java Virtual Machine (JVM). This means it worked on any Java-enabled computer and operating system.

VI. COST BENEFIT ANALYSIS

The reason for sorting out devices was to carry out a Cost-Benefit Analysis on the implementation of SDN in CST. The cost-benefit analysis is an assessment to check whether the estimated costs exceed the estimated income and other benefits.
As of 2022, more than 50% of the networking devices in CST were found to be supporting SDN. The following are the list of SDN supportive and SDN non-supportive networking devices along with their number and cost per device [7].

SDN supported devices (Allied Telesis)			
Model Number	Cost per Device (Nu.)	No. of devices	Total Cost (Nu.)
X230-10GP	59497.97	6	356987.82
X230-18GT	50310.06	7	352170.42
X230-18GP	68326.83	3	204980.49
X230-28GT	58624.62	3	175873.86
X510-28GSX	341499.54	3	1021198.62
Grand Total of Allied Devices			2111211.21

SDN Non-supported devices (Cisco and D-Link)			
Model Number	Cost per Device (Nu.)	No. of devices	Total Cost (Nu.)
Cisco Catalyst 3560	6161.04	3	18483.12
Cisco Nexus 3064	94690.37	3	284071.11
Cisco SG300-53	33043.72	8	264349.76
DGS-1210-28	10615.94	8	84927.52
Total cost of SDN non-supported devices			651831.51

Figure 4: Cost structure for SDN supported and unsupported devices

As per the ICT, the total expenditure (approximately):

1. If all networking devices were replaced SDN supported devices (Allied Telesis) will cost Nu. 55,00,000 with a warranty period of 5 years.

2. If all the networking devices were replaced by SDN non-supported devices (Cisco/D-Link) will cost Nu. 35,00,000 with a warranty period of 1 year.

Therefore, Phuentsholing being a lightning prone area, and considering the fact that the Cisco/D-Link devices were highly vulnerable to damages with a warranty of just 1 year:
Total cost of SDN non-supportive devices in 5 years = 5 * 35,00,000 = Nu. 175,00,000 (approximately)
Total cost of SDN supportive devices in 5 years = Nu 55,000,000 (approximately)

Hence, since 55,00,000 < 175,00,000, the cost of SDN supportive devices which will last for 5 years was lesser compared to the cost of SDN non-supportive devices in 5 years. Approximately, an amount of Nu.120,00,000 could be saved in a period of 5 years.

VII. College Network Simulation

DrukREN provides connectivity to all the colleges under Royal University of Bhutan, Technical Training Institutes, hospitals and schools. All Bhutanese network operators peer with DrukREN at 10 Gbps, allowing its members to access domestic content and its speed corresponding to its the membership subscription [5].
CST is one of the colleges that has access to the DrukREN network with a speed of 1 Gbps/74 Mbps. It is the primary source of internet connection for the college. In CST network as shown in fig.5, the network traffic from the DrukREN is received by the border router which resides at the edge of the network and ensures its network connectivity with external network or internet. The college network is divided into three layers; a core switch which resides within the backbone of the network that serves as the gateway to the internet and to deliver frames or packets as quickly as possible in the center of the network. There are various types of servers set up which are being accessed by the students and staff. A distribution switch collects data from all the access switches which are then forwarded to the core layer switches. At the bottom layer, there are access switches which directly interact with the end user devices.

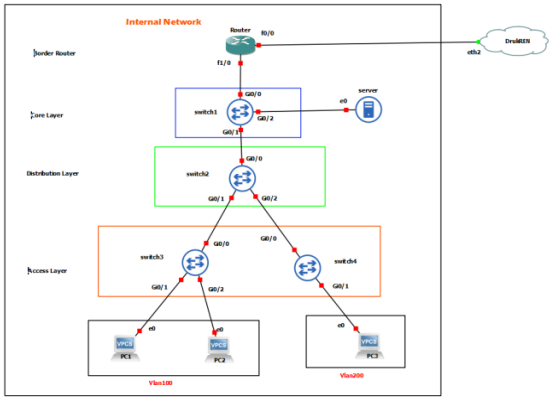

Figure 5:College network simulation in GNS3

The local control plane, which is effectively the device's brain, and the data plane, which forwards traffic, are both present in legacy networking devices. Each device's control plane runs locally. SDN advocates said that this was a difficult approach to figure out the network's topology. There was no single device that could be seen as the complete network; each device must be figured out on what the network looks like independently before synchronizing. Since the college has several network devices, using the terminal from a management standpoint necessitates connecting to each device and manually configuring the networking equipment.
Not only was the manual configuration time consuming but also prone to human error. The configurations of the devices should be done on each device and there was no centralized control of the underlying network. Since networking devices are proprietary, no one can write an application for them, and they have no access to the vendor's operating system code to make changes.

VIII. Simulation using ODL controller

This simulation was carried out by using two virtual machines installed on VMware Workstation pro. Ubuntu which was used to install ODL controllers and then for the infrastructure layer or the switching layer, Mininet was used.
A controller coupled to a collection of switches forming a tree topology is used to replicate the design in an SDN architecture. The controller is the network orchestrator and supervises all switches connected to it, hence, the first level is dedicated to it. The controller was regarded as the network's nerve center. The second level consists which serves as the network's distribution layer, ensuring the network's redundancy and availability. The last level includes three switches that connect two host machines. The access level is the name of the level as shown in fig.6 .

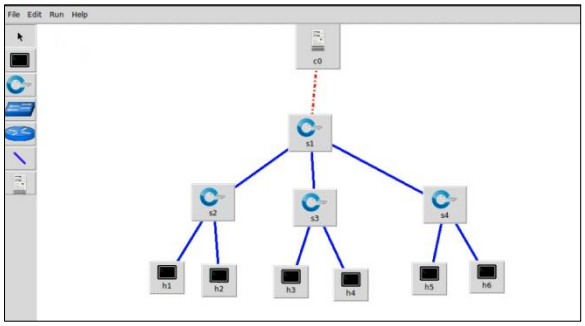

Figure 6: SDN simulation with mininet GUI emulator

The ODL controller orchestrates the SDN controller. It connects directly to all of the network's switches. The Open vSwitch (OVS) specification is used to implement the switches. This program is intended to provide complete support for the OpenFlow protocol, which will be used by all SDNs for routing.
An application developer can update flow tables and flow entries on networking devices using a RESTful API-based interface based on the NBI without having to talk to them directly. The application developer does not need to comprehend the complexity and requirements of real Application Specific Integrated Circuits (ASICs) on switches and networking devices because he or she is isolated from the hardware.

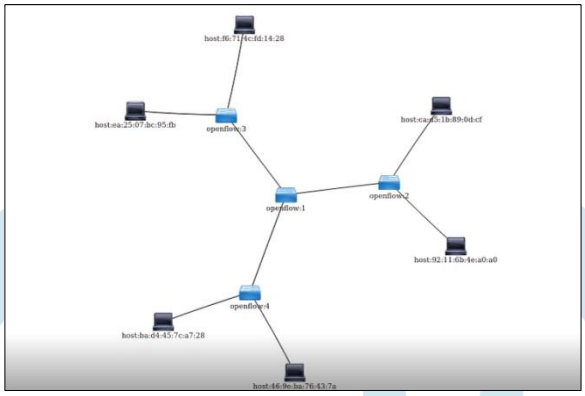

Figure 7: Network visualization in OpenDaylight

As a result, rather than having their own brain, networking devices are now governed by a centralized brain. This provides a centralized control, regulatory, and management device. The flow entry table to the OVS is refreshed by the controller. The controller is only a conduit or abstraction layer through which an application developer can affect network flows. With applications, the potential of SDN and open flow is revealed.
Be it in any network, networking devices like routers and switches are used to forward the packets based on IP address and Mac address respectively, and firewalls to block certain traffic flows. For legacy network, the devices are required to be configured for forwarding the packets based on certain protocols whereas in the case of software defined networking, there is a centralized controller which configures the flow table containing rules dynamically or statically of the OVS. The rules that are written in the flow table can determine the functionality of the devices. If the packet is forwarded based on the IP address, the OVS is working as a router. In addition to a centralized control, the scalability of the network will also be ensured. Thus, compared to traditional architecture, SDN architecture has lots of advantages.

IX. RECOMMENDATION

As a recommendation to the existing college network, a two-tier architecture or so called the collapsed core architecture where the core and distribution layers are merged as one to the existing three tier architecture is proposed as shown in fig.8 . With the two-tier architecture, not only the cost of the networking devices is reduced but also the benefit provided by the three tiers can be reaped within the two tiers. CST, being a small campus area network and considering its less tendency towards growing in size as compared to large enterprises, makes it more suitable for CST to use two tier architecture in terms of cost as well as benefit. Additionally, this recommended topology provides a redundancy for the core-distribution layer which serves as a backup whereby if one point fails, the other one will still function.

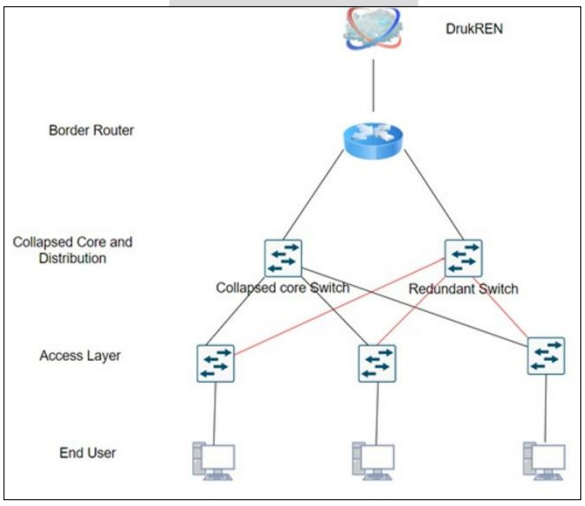

Figure 8: Recommended topology as an alternative to the existing network topology

X. Conclusion

SDN is one of the trending technologies in the field of networking. The rise of SDN has dominated the traditional form of networking ever since its inception in 2011. The same case is happening in the field of networking across the organizations and institutes in Bhutan as well. CST is currently halfway through its migration from legacy networking towards the SDN with 50% of the networking devices being SDN supported. However, the majority of the organizations in Bhutan are found to be reluctant to depart from the legacy networking system mainly due to the cost that would be incurred while migrating towards SDN.

For the reasons stated above, this research aimed to analyze and study the implementation of SDN in CST and check the feasibility of it so as to recommend the implementation of SDN throughout the nation. This analysis was performed through two phases of simulations of the existing network topology using the GNS3 network simulator and then by using ODL which is one of the SDN controllers.

The simulations of the existing network topology of the college helped gain a better insight towards learning the bottleneck of the legacy networking system which was mainly to do with the network congestion, automation and programmability. From the analysis, the ultimate conclusion drawn was that it is absolutely feasible for CST or any other organizations to migrate the network towards SDN in terms of cost efficiency as well as in terms of functionality as a whole.

XI. Acknowledgment

We are grateful to the College of Science and Technology for giving us the opportunity to work on such a fascinating issue. This research provided us with a wealth of information and abilities.

For always being there to provide overall guidance and support throughout this project, we would like to express our utmost respect and gratitude towards the Project Guide Mr. Yeshi Wangchuk without whom we would not have been able to drive this project to completion.

Along with the guide, we would like to take the opportunity to express our gratitude to Sr. ICT Officer, Mr. Jiwan Gurung for his time and for supplying us with all the necessary resources required for this project. We also would like to express our special thanks to the project coordinator, Mr. Karma Wangchuk and department head, Mr. Tandin Wangchuk, for taking interest in our project and helping us find the related courses for the project.

References

[1] Contributors, M. P. (2022, January 1). “Mininet: An instant virtual network on your laptop (or other PC)-Mininet”. Mininet. Retrieved on May 13,2022, from: https://mininet.org/

[2] Christodoulopoulos, J., Paraskevas, M., & Triantafyllou, V. (2016, November). “Software Defined Networks: A case for QoS implementation at the Greek School Network”. In proceedings of the 20th Pan-Hellenic Conference on Informatics (pp. 1-6)

[3] Gupta, S. G., Ghonge, M. M., Thakare, P. D., & Jawandhiya, P. M. (2013, April 4). “Open-Source Network Simulation tools: An overview”. Retrieved 03 2022, from: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-2-ISSUE-4-1629-1635.pdf

[4] JavaTPoint. (2022, January 1). “Programming Language | What is programming language”. Retrieved on May 10, 2022, from: https://www.javatpoint.com/programming-language

[5] Lodey, D. P., & Jamyang, K. (2021, April 19). “Druk Research & Education Network”. Retrieved on March 1, 2022, from: https://drukren.bt/

[6] Mahmood, A., Saleem, M. F., & Latif, A. (2013, 09 09). “Key features and optimum performance of network simulators: A brief study”. (Academia, Compiler) Retrieved 03 2022, from: https://www.academia.edu/4861597/Key_Features_and_Optimum_Performance_of_Network_Simulators_A_Brief_Study

[7] PriceSpy. (2021, 1 january). PriceSpy Ltd. PriceSpy UK. Retrieved on 9 february 2022, from: https://pricespy.co.uk/

[8] Reddy, N. D. P., Sivakumar. B. (2018, April 24). “Implementing Software - Defined Networking in a campus environment”. International Journal of Engineering Research & Technology. ISSN: 2278-0181. Vol 3 (18).

[9] Sahu.H & Hungyo.M. (2018). “Introduction to SDN and NFV. Innovations in Software-Defined Networking and Network Functions Virtualization”. 1-25. 10.4018/978-1-5225-3640-6.ch001.

[10] Shangmugam, M. J., & Ramya, S. T. (2018). “Software Defined Networking: A paradigm shift in networking for future. Emerging trends and applications”. International Journal of Applied Engineering Research. 13 (18), 134575-13481

[11] Stoltzfus, J. (2021, May 28). “Graphical User Interface (GUI)”. Retrieved on May 10, 2022, from: https://www.techopedia.com/definition/5435/graphical-user-interface-gui

[12] VmWare. (2022, May 5). “What is software defined networking”. Retrieved on May 10 from: https://www.vmware.com/topics/glossary/content/software-defined-networking.html

[13] Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). “A survey on software defined newtorking”. IEEE Commuincations Surveys & Tutorials. 17 (1), 27-51. https://doi.org/10.1109/comst.2014.2330903

[14] Zhu, L., Karim, M., Sharif, K., Xu, C., Li, F., Du, X., & Guizani, M. (2021). “SDN Controllers”. ACM Computing Surveys, 53 (6), 1-40. https://doi.org/10.1145/3421764