IJCSN - International Journal of Computer Science and Network, Volume 9, Issue 4, August 2020
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Software Defined Networking Framework for Campus Network Management

[1] **Edward Udo;** [2] **Etebong Isong;** [3] **Emmanuel Nyoho**

[1] Department of Computer Science, University of Uyo, Uyo, Nigeria

[2] Department of Computer Science, AkwaIbom State University, IkotAkpaden, Nigeria

[3] Department of Computer Science, University of Uyo, Uyo, Nigeria

Abstract - Nowadays, university campus networks are architecturally designed following the traditional approaches where network devices are interconnected and are fully responsible for critical management decisions within the network. The devices in this traditional setting are configured by IT managers and network administrators. The number of devices connected to campus network is growing rapidly which makes manual configuration tedious and complex, especially considering the heterogeneous nature of these devices/services and Bring Your Own Device (BYOD) concept. This makes management of traditional network and configuration of these devices according to predefined policies inefficient and difficult respectively. This work therefore proposes the use of Software Defined Networking (SDN) framework as a tool for the management of campus network where network-wide rules are incorporated into the network. SDN is a new network paradigm for network management as it provides a programmable network platform through a programmable device (controller). Our framework employs more controllers to handle the heterogeneous nature of the present day campus environment for proper management of users and services and allocation of different network functions across the various infrastructures based on predefined policies. Implementing this proposed framework will undoubtedly yield flexible campus network management, efficiency of data transmission within the network, better network performance and security and also guarantees network evolution.

Keywords - *Software Defined Networking, Campus Network, Network Management, Network Systems, Floodlight Controller*

1. Introduction

Campus network is an internal network of a university - a dynamic environment with many events occurring. It is a collection of Local Area Networks (LANs) in a concentrated geographical area consisting of a large number of network devices such as switches and routers where the networking equipment and communications links belong to the university and are managed by the university (Goransson et al., 2017). In this present age, most administrative, academic and research processes are done via a network and as such every campus needs a network that is secured, fast, high performing and available with remote access and centralized management capabilities. Designing a network that will allow for these functions would result in increased complexity making such a network highly heterogeneous especially when equipment, applications and services are provided by different manufacturers and providers (Xia et al., 2015).

The present day university campuses networks are architecturally designed following the traditional approaches where routers, switches and other physical hardware are interconnected in such a way that data flow and communication is possible within the network. IT system managers, administrators and operators are fully responsible for manually configuring each of these devices for traffic and security events and for ensuring that each device is updated with the latest configuration settings (Rana et al., 2019). As critical management decisions are left in the hands of these physical devices, complex administration of the network becomes a challange as the network evolves. Manual configuration of these devices using low-level device specific syntax, which these devices support, is tedious, complex, time consuming and error prone; and network operators are required to be present all the time to configure these devices (Sahay et al., 2019). The design of a traditional campus network environment is shown in Figure 1.

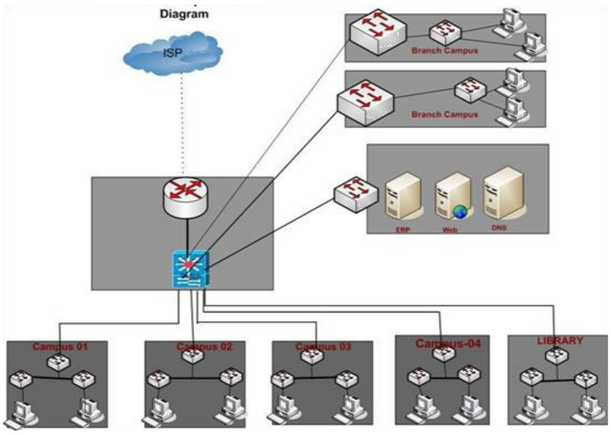


Fig. 1 – Traditional Campus Network Design (Source: Ali et al., 2015)

There has been a massive increase in the number of devices connected to network as demand and usage for technology is growing and this goes beyond what data communication networks may be able to handle in the nearest future (Jefia et al., 2018). Campus network continue to widen and become more complex as students and staff now carry multiple devices connected to the network which results in the expansion of the network infrastructure and demand for huge network resources. It is therefore no longer a matter of managing a fixed number of hosts; it is now a matter of managing several hosts/devices from different vendors running diverse applications. These make traditional approach to network management quite inefficient and difficult to configure according to predefined policies, prompt response to fault and load changes and probably reconfiguration when need arises. The static nature of the conventional network makes it difficult to accomplish the dynamic computing and storage desires of big data centres, enterprises and campuses (Singh and Kaur, 2017). The huge increase in the number of devices, social media applications, mobile based services and management of network components have put a serious pressure on network operators (Kutay and Ercan, 2016) and achieving centralized and coordinated management of user and device identity has become a key challenge in enterprise networking. With the increase in the number and types of devices connecting to campus networks, managing different application traffic, mobility issues and security becomes challenging and requires more manual intervention (through programming) for the management of the network.

The traditional network is also difficult to manage because the data and the control plane of the network devices are fused together which may reduce flexibility and dynamicity of the network. But, these days' computer networks are dynamic and complex, therefore their configuration and management continue to be challenging (Kim and Feamster, 2013). Moreover, lack of programmability and centralized control in the traditional network makes it difficult to deploy and integrate new services with the existing ones without halting the ongoing services (Chen et al., 2009).

These problems, which have long been noticed by networking research community, practitioners and the industry, have demanded novel approaches to support the deployment of network devices and applications in future computer networks for enhanced management (Sahay et al., 2019; Xia et al., 2015). Few ideas have been introduced in designing and deploying a good network solution, Zhang et al., (2010); Campbell et al., (1999); Popa et al., (2010) (cited in Xia et al., 2015). Despite many previous proposals to make networks easier to manage, these proposed solutions were only temporary solution because of the difficulty in adjusting or changing the underlying infrastructure of the traditional network. The rigidity of these infrastructures hinder the possibilities for innovation or improvement, since network devices have generally been closed, proprietary, and vertically integrated (Kim and Feamster, 2013).

Software Defined Networking (SDN) has emerged as a new networking paradigm for managing different kinds of networks ranging from enterprise to home network through software enabled control (Sahay et al., 2019). It has become the ideal technology and the most promising solution needed to manage campus network in an efficient, automated and systematic way. SDN is strongly supported by renounced internet players such as Google, CISCO, NEC and other standardization organizations such as ONF and IETF (Wickboldt et al., 2015).

SDN is a computer network which is implemented and managed by software to facilitate the network requirement of any organization (Rana et al., 2019). It is about abstracting the logic of traditional networks setting from a fixed hardware solution and takes it to the level defined by software. In this emerging network architecture, network control is decoupled from the data plane of the network devices (Open Networking Foundation, 2012, 2013) allowing them to behave simply as forwarding devices which contain the rules used in processing a packet as it moves from source to destination. The rules table contains information such as source IP, destination IP, protocol etc (Sahay et al., 2019). In SDN, network intelligence is logically centralized in an entity called the controller (which is software based) from where the network devices can be directly programmed using a standard interface, such as OpenFlow, which is the most important SDN implementation available (McKeown et al., 2008) and is being standardized by Open Networking Foundation (ONF) (Kim and Feamster, 2013). SDN controller gives the flexibility to configure, manage and troubleshoot a network and network devices using automated sets of programs meaning that network elements can be remotely managed from a centralized controller (Jefia et al., 2018).

SDN encourages innovation by providing a programmable network platform to implement, experiment and deploy new ideas and new applications (Xia et al., 2015) and forwarding decisions are made in a logical single point which has a global knowledge of the network state. It facilitates network operations such as routing and addition of more rules to forwarding device (Latah and Toker, 2016). SDN is rapidly becoming the go to solution for those who are having trouble overcoming the limitations of traditional networking (Rana et al., 2019) as it simplifies or solve critical management tasks that are faced in traditional network management. The problems faced by network administrators in implementing traditional network protocols have posed a great challenge to network management. As the world continue to transpire in digital technologies, great demand has been placed on the application of high-level policies that will take network management to level of automated configuration and modification which are absent in the traditional IP networks (Benson et al., 2009).

Majority of efforts are focused on providing services and solutions over SDN Networks while network management is not given much attention (Wickboldt et al., 2015). Management, as widely opined, should not be an afterthought phenomenon (Schonwalder et al., 2009). Campus network that could cope with the challenges of a modern network environment should be driven by policies and policy-driven network requires a lot of configurations. These policies include usage policies, access-defined policies, security policies, Quality of Service (QoS) policies etc.

Owing to complexity of campus network policies, using traditional approaches to implement a network-wide policy for managing such a network is prone to management error and is incredibly difficult (Kutay and Ercan, 2016). This makes campus network a very good area to deploy SDN.

This work therefore proposes a SDN framework, as a campus network management tool, where network-wide policies (authentication, security, access control, traffic monitoring) are incorporated in managing the network and are coordinated through a centralized management interface for enhanced performance and prompt/error-free management decision.

2. SDN Architecture

As depicted in Figure 2, the original design of SDN constitutes three majorlayers: infrastructure, controller and application layers; as well as the interfaces between successivelayers. The infrastructure layer comprises network devices, computers, firewall devices etc that perform packet forwarding and filtering, hence, the elements in this layer are also collectively called the data layer, the forwarding plane or the data plane (Alshnta, et al., 2018).

The networking devices in the infrastructure layer have flow tables that contain the necessary forwarding information to be used in processing the incoming packets. The entries of the tables have match fields (matching rules) which holds information like packet-header, entrance port and meta data; counters, which collects statistics of the flow such as flow duration, number of packets and bytes received; and actions which defines how to handle matching packets (Nunes et al., 2014).

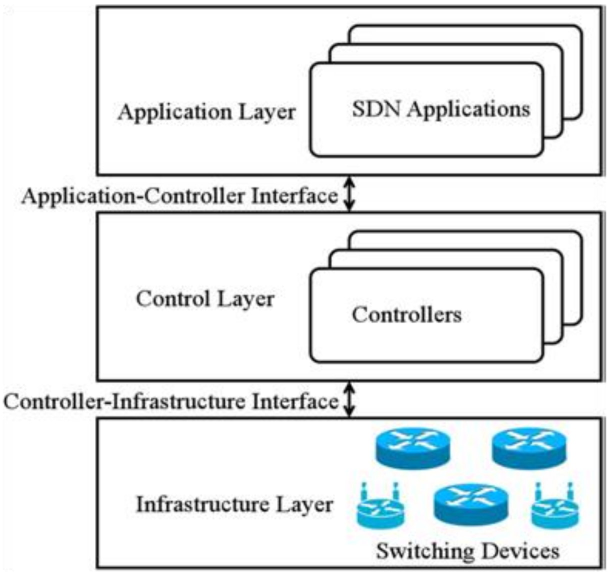


Fig. 2 - SDN Architecture (Source: Xia et al., 2015)

When a packet arrives a forwarding device, matching fields are compared with flow entries, if a match is found, the predefined action is performed and if there is no match, the device looks for a second miss-flow table that defines the rules (drop the packet or send to the controller) for no match case (Kutay and Ercan, 2016).

The control layer, otherwise called the control plane is responsible for programming and managing the data plane and controlling how the routing logic should work. The logic and algorithms that are used to program the infrastructure layer reside in the control layer. The control layer determines how the forwarding tables and logic in the data plane should be programmed or configured (Alshnta et al., 2018). Control plane is also responsible for making decisions on how traffic would be routed from one node to another within the network based on end user application requirements. The control layer also communicates the policies of the entire network to the data plane (Bakhshi, 2017).

The central component of a control plane is the controller, where all the management and control functions of the network are performed. It has the topology of the entire network and address information of all the network hosts. The controller and the switches in the data plane communicate over a secured channel using set of messages and can delete, add or update flow entries of a switch. When a switch sends a packet-in request to the controller in respect of a packet whose flow entries is not in the flow table, the controller will handle the forwarding of such packet or take any other actions (drop the packet, modify the header, change the route or update the flow table for subsequent flows). The controller therefore listen to traffic, take actions and define routes when there is a need (Kutay and Ercan, 2016).

The control layer interacts with the infrastructure layer through the Controller-Infrastructure Interface (Southbound Interface), which specifies the functions needed by the controller to access functions provided by the switching devices. That is, it specifies the communication protocol between data plane devices and the SDN controller. The most common southbound interface is the OpenFlow protocol (McKeown et al., 2008), which is a standard communication protocol defined by Open Networking Foundation (ONF). OpenFlow is the protocol designed to negotiate communication between the controller and the network devices. Before the negotiation, connected network devices must support the OpenFlow protocol. OpenFlow exploits common sets of functions that run in many switches and routers and provides an open protocol to program the flow table of different switches and routers (Reddy and Sivakumar, 2015).

Application plane comprises network specific (network virtualization, traffic engineering, routing monitoring and quality of service) and business applications (Bakhshi, 2017). The application layer contains SDN applications designed to fulfill user requirements. Through the programmable platform provided by the control layer, SDN applications are able to access and control switching devices at the infrastructure layer (Xia et al., 2015).

The application layer interacts with the control layer through the Application-Controller Interface (Northbound Interface), which determines how to express operational tasks and network policies and how to translate them into a form the controller can understand (Kim and Feamster, 2013). The northbound Application Programming Interface (API) allows software applications to be plugged into the controller thereby allowing that software to provide the algorithms and protocols that can run the network efficiently. These applications can quickly and dynamically make network changes as the need arises. The northbound API of the controller is intended to provide an abstraction of the network devices (which allows the application programmer to deal with the network as whole rather than individual nodes) and protocols (hiding the application developer from the details of southbound protocol – OpenFlow) (Alshnta et al., 2018).

It is obvious therefore that the SDN architecture allows a unified and global view of a complicated network, such as a campus network, and provides a robust control platform for the management of flow of traffic and interactions among the different layers.

3. Related Works

Previous studies in network technologies bring about programmable networks which laid the foundation for the development of SDN. These technologies include Routing Control Platform (RCP) – an approach where information routing was separated from Internet protocol (IP) routers and allow the routers to simply forward packets. RCP selected routes on behalf of IP router in autonomous system and exchange reachability information with other domains (Feamster et al., 2004); 4D – named after the architecture’s four planes: Decision, Dissemination, Discovery and Data, is an architecture which emphasized on separation of routing decision logic from protocols that govern the interaction among network elements. The decision plane had a global view of the network and called upon the services of dissemination and discovery planes to control the data plane for traffic forwarding. The discovery plane gave information about what resources were available to network controller. Dissemination plane showed how to detect network topology (Greenberg et al., 2005); Ethane – in this technology, all forwarding decisions in a network are done first in software and then the hardware mimics these decisions for subsequent packets to which that decision applies. The hardware does not need to understand the logic of packet forwarding (Casado et al., 2008).

Most literature present theories, reviews and surveys on SDN covering its meaning, architecture, protocols, interfaces, programming languages etc with few authors actually implementing SDN paradigm in practical scenarios or networking domains as SDN is still at the early stage of development. Some literatures are:

1. Sahay, et al. (2019) – provided an extensive survey on the application of SDN to enhance the security of computer networks by highlighting recent research studies on attack detection and mitigation, traffic monitoring and engineering, configuration and policy management. They also presented recent efforts on securing smart grid infrastructure via SDN.
2. Bakhshi (2108) – reviewed the state of the art in SDN by providing historical background on complementary technologies and their associated shortcomings which paved way for SDN. Several new and legacy protocols used in SDN were discussed together with SDN deployments in data centres, cloud computing and wireless communication, residential networks, campus and high speed networks.
3. Wickboldt, et al.(2018) – discussed on how to manage networks base on SDN by identifying some management requirements (configuration, availability, programmability, security,

monitoring, performance etc) of SDN and highlighted major challenges to be addressed to enhance wide adoption of SDN.

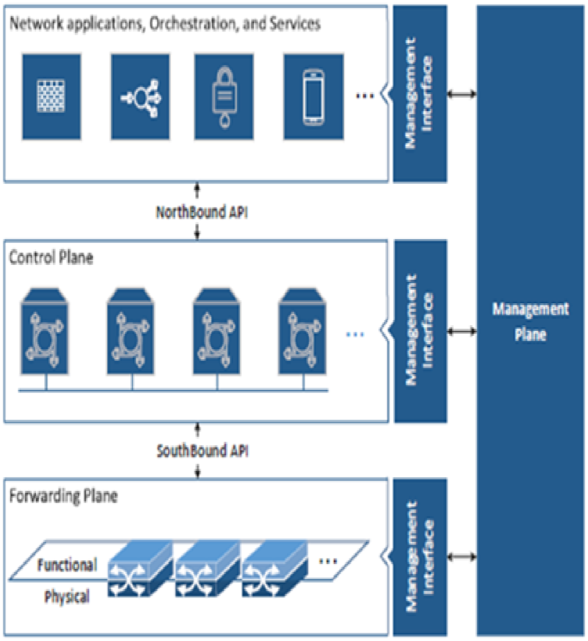
4. Singh and Kaur (2017) – presented the theories of SDN and highlighted how it can help in configuring and managing network needs.
5. Barros et al.(2015) – showed how SDN technology can be integrated to develop, organize and virtualize cloud networking. They also showed practical analysis of integration of OpenDaylight SDN controller and OpenStack operating system for building a consistent and fully functional cloud virtual networking environments.
6. Xia, et al.(2015) – surveyed latest development in SDN, architecture of SDN and substantiated existing researches associated with each layer of the SDN architecture and also looked at the implementation of SDN using OpenFlow protocol.
7. Kim and Feamster (2015) – developed a framework called Procera, which helps operators express event driven network policies (under four control domain: time, data usage, authentication status and traffic flow) that react to various types of events using high-level functional programming language. They focused on three problems in network management: enabling frequent changes to network conditions and state; providing support for network configuration in a high level language; providing better visibility and control over tasks in performing network diagnosis and troubleshooting. They described the deployment of procera in campus and home networks to demonstrate that SDN can improve network management tasks.
8. Reddy and Sivakumar (2015) – Implemented SDN in campus network environment in a centralized, separate protocol block that manages the entire network via software. The SDN is connected to different OpenFlow controllers which are connected to Open Flow switches or other networking devices. The switches monitor the network and send periodic reports to the network administrator. In event of network or node failure the OpenFlow controller will trigger an alert to the network administrator.

The problem with the work of Reddy and Sivakumar (2015) is that the approach may not function effectively as the network keeps expanding because larger networks requires a high-level decision policy as such more SDN controllers is required in the application layer (Alser, 2015). Also unavailability and resilience issues becomes more critical because of a single point of failure (Wickboldt, et al., 2015) associated with the centralized, separate block for the controller proposed in their work.

The framework proposed in this work employ more controllers to handle the heterogeneous nature of the present day campus environment for proper management and troubleshooting of any network problem and also eliminate a complete breakdown of the entire network due to a single point of failure as may be experienced in a centralized/separate block for a controller.

4. SDN Framework for Campus Network Management

The workings of the management plane and the management interfaces in conceptual architecture proposed by Wickboldt et al. (2015) as shown in Figure 3 is used in the framework proposed in this work. The inclusion of the management plane and the management interfaces proffer appropriate platform for the co-ordination of a university (campus) network. The management plane allow for the organization of all operations and management functions in the network while the management interfaces provide a channel for proper flow of information between the management plane and other planes (forwarding, control and application). All adjustments to the settings of the network devices in the forwarding plane are enforced at the management plane while all reports to the network administrator are channeled through the management interfaces. These interfaces are generic to accommodate new software introduced into the network



Fig, 3 - High-Level Conceptual Architecture of SDN

The management plane and the interfaces will be embedded within the administrator's platform and linked to a dedicated controller in the control layer to enable the network administrator effectively allocate management functions to appropriate controllers and planes using a software. Our proposed SDN framework for campus management is shown in Figure 4.

The core management functions of the management plane, domiciled in the administrator's end, include:

i. Configuration of network devices and hosts considering the heterogeneity of the network devices and Bring Your Own Device (BYOD) phenomenon.
ii. Coordinating device deployment so as to dictate the behaviour of the network.
iii. Grouping the different network users into appropriate groups based on status (teaching staff, non-teaching staff and students)
iv. Assigning different level of access (access control) and privileges to different groups of users of the network with appropriate log-in duration.
v. Carrying out proper authentication and authorization of users through web-portal registration and validation of users' devices MAC addresses and users' login details.
vi. Monitoring and reporting on unexpected conditions within the SDN campus network due to software changes, device reconfiguration, unauthorized access and threats.

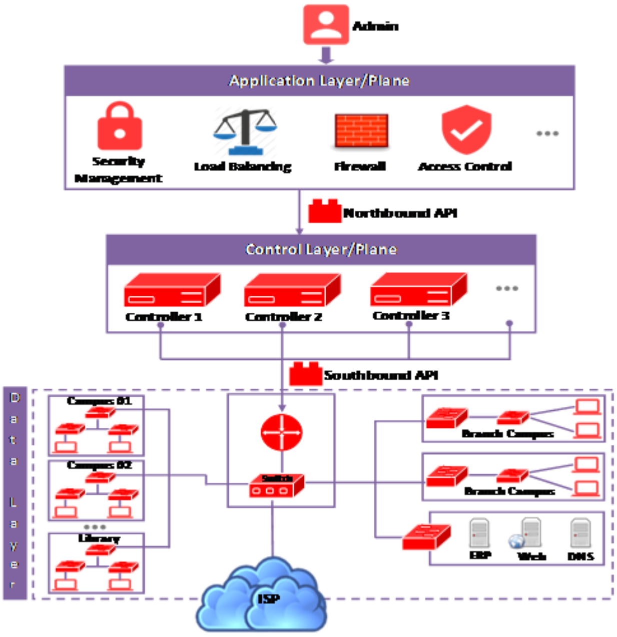


Figure 4 – Proposed SDN Framework for Campus Network Management

The switching devices in the data plane all support OpenFlow protocol and receive rules generated by the SDN controller in the control plane. These rules describe the policies which examine and regulate the network and also aid the switching devices in performing packet forwarding as well as gathering and reporting network status of all packets processed. The rules are installed in appropriate switching devices for operations and are updated as a result of configuration changes and dynamic control. The controller also use traffic statistics and network status which are collected and stored by the switching devices to build a global view of the entire network and provide the application layer with necessary information. The controllers exchange these traffic statistics and network status among themselves for a global view of the network for enhancement of quick network convergence as well as processing of policies from network administrator.

The controller (in the control plane) is responsible for translating application requirements into packet forwarding rules. This function relies on the programming language (protocol) between the application and the control planes. The programmability, through the programming language allows for the automation of the controller using policies of the application which are available in the northbound interface. The southbound interface is the OpenFlow protocol since the network devices are SDN complaint. The schematic diagram of a controller is depicted in Figure 5.

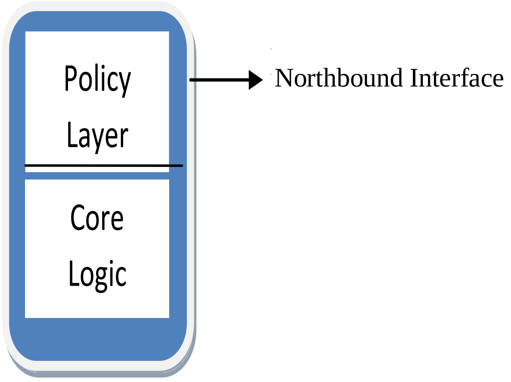


Fig. 5 – Schematic Diagram of a Controller

The northbound interface is a set of APIs that is controlled by a web-based GUI to dominate the OpenFlow switch parameters. The northbound interface in this work is floodlight, which is a Java-based OpenFlow controller supported by open source community of developers. It is a commercial SDN product produced by Big Switch company and has been actively tested and improved by the industry and developers community. Floodlight is recommended when administrators know java programming language, need production level performance and would like to have industry support.

5. Architecture of the Policy Layer

The components and the operations of the policy layer are shown in Figure 6

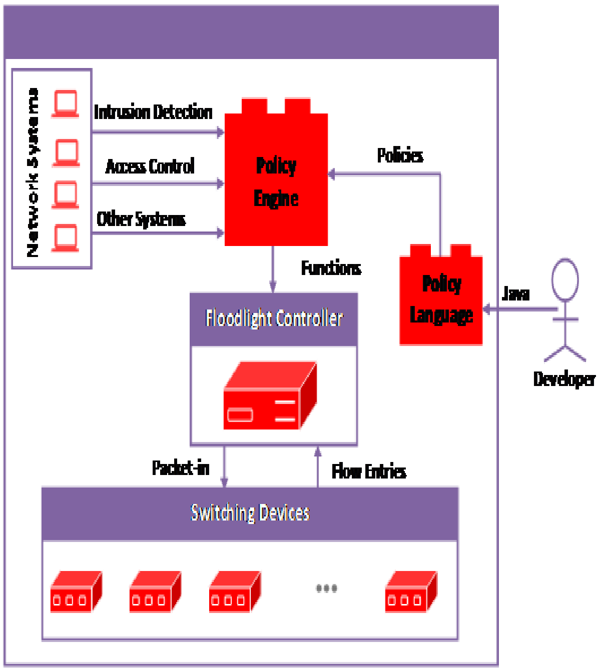


Fig. 6 - Architecture of Policy Layer

The network systems are different systems handling the overall coordination and functionality of the entire campus network. These systems include Authentication System, Intrusion Detection System, Access Control System, Network Management System etc. Updates and events from these systems are sent to the policy engine for onward transmission to the controller as policy functions after proper decoding of such updates and event. These policy functions are translated to the network devices (switches) as forwarding rules. The switches are OpenFlow compliant. The policy engine is responsible for parsing all the network policies formulated by the developer and are built using the policy language (Java). The policies in the policy engine are used to enforce the policy of the entire campus network. The Floodlight controller is a framework that interacts with policies in Java using a set of APIs. The above architecture is an example of northbound interface definition.

7. Conclusion

The proposed framework is hoped to be implemented in the campus of University of Uyo, Uyo, Nigeria. University of Uyo, Uyo has three campuses located within the metropolis, which makes it a good environment to implement the SDN framework. A database for all users and their corresponding devices will also be created and combined with users and access control policies to form the operational instructions for the network devices and infrastructure. The results will be flexible management and troubleshooting of network problems, efficiency of data transmission within the network, better network performance, reduction on administrators' overheads and cost, deployment of new technologies/ideas and better network security.

References

[1] Ali, M., Hossain, M. and Pavez, M. (2015). Design and Implementation of Secure Campus Network.International Journal of Emerging Technology and Advanced Engineering, 5(7), 370 – 374.

[2] Alsher, A. (2015). An Overview of Network Virtualization and Cloud Network as a Service.International journal of Network Mangement, Vol. 25, 1 – 30.

[3] Alshnta, A., Abddollah, M. and Al-Haiqi, A. (2018). SDN in the Home: A Survey of Home Network Solutions Using Software Defined Networking. Cogent Engineering, 5(1), 1 – 40.

[4] Bakhshi, T. (2017) – State of the Art and Recent Research Advances in Software Defined Networking, Wireless Communication and Mobile Computing, 1 – 35.

[5] Barros, B., Simplicio, M., Carvalho, T., Rojas, M., Redigolo, F., Andrade, E. and Magri, D. (2015). Applying Software-Defined Networks to cloud computing. In 33rd Brazilian Symposium on Computer Networks and Distributed Systems, Vitoria, ES, Brazil.

[6] Benson, T., Akelia, A. and Maltz, D. (2009). Unraveling the Complexity of Network Management. In proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, Berkeley, CA, USA, 335 – 348

[7] Casado, M., Koponen, T., Moon, D. and Shenker, S. (2008). Rethinking Packet Forwarding Hardware.In Proceedings of 7th ACM SIGCOMM HotNets Workshop, Calgary, Alberta, Canada, 1-6.

[8] Chen, X., Mao, Z. M. and Van Der M. J. (2009). Shadownet: A Platform for aRapid and Safe Network Evolution. In Proceedings of the USENIX Annual Technical Conference, Berkeley, CA, USA, p 3.

[9] Feamster, N., Balakrishnan, H., Rexford, J., Shaikh, A. and Merwe, J. (2004). The Case of Separating Routing from Routers. In Proceeding of ACM SIGCOMM Workshop on Future Direction in Network Architecture, Portland Oregon, USA, 5 - 12

[10] Goransson, P., Black, C. and Culver, T. (2017). SDN in Other Environments. Software Defined networks, A Comprehensive Approach (2nd Edition), Morgan Kaufmann Publishers, San Francisco, USA, 217 – 239.

[11] Greenberg, A., Hjalmtysson, G., Maltz, D., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J. and Zhang, H. (2005).A Clean State 4D Approach to Network Control and Management. ACM Computer Communication Revolution, 35(5), 41 – 54.

[12] Jefia, A., .Popoola, S. and Atejero, A. (2018). Software Defined Networking: CurrentTrends, Challenges and Future Directions. In Proceedings of International Conference on Industrial Engineering and Operations Management, Washington DC, USA, 1677 – 1685.

[13] Kim, H. and Feamster, N. (2013). Improving Network Management with Software Defined Networking, IEEE Communication Magazine, 51(2), 114 -119

[14] Kutay, M. and Ercan, T. (2016). An Overview of Software Defined Campus Networks. Selcuk University Digital Archive Systems, 4(2), 155 – 164

[15] Mckeown, N., Anderson, T., Balakrishuan, H., Palmkar, G., Peterson, L., Rexford, J., Shenker, S. and Tuner, J. (2008). OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer Communication Review, 38(2), 69 – 74.

[16] Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K. and Turletti, T. (2014). A Survey of Software Defined Networks: Past, Present and Future Programmable Networks, IEEE Communication Survey and Tutorials, 16(3), 1617 -1634.

[17] Open Networking Foundation (2012). SDN Security Considerations in the Data Center. Technical Report, ONF.

[18] Open Networking Foundation (2013). SDN Security Considerations in the Data Center. Technical Report, ONF.

[19] Rana, D., Dhondiyal, S. and Chamoli, S. (2019). Software Defined Networking (SDN) Challenges, Issues, Solution. International Journal of Computer Sciences and Engineering, 7(1), 884 – 889

[20] Reddy, L. and Sivakumar, B. (2015). Implementing Software-Defined Networking in Campus Environment, International Journal of Engineering Research and Technology, 3(18), 1 – 6.

[21] Sahay, R., Meng, W. and Jensen, C. (2019). The Application of Software Defined Networking on Securing Computer Networks: A Survey. Journal of Network and Computer Applications, Vol 131, 89 – 108

[22] Schonwalder, J., Fouquet, M., Rodosek, G. and Hochstatter, I. (2009). Future Internet: Content, Services and Management, IEEE Communication Magazine, 47(7), 27 – 33.

[23] Singh, J. and Kaur, Y. (2017). Network Management Using Software Defined Networking. International Journal of Advanced Research in Computer Science, 8(5), 261 – 265

[24] Wickboldt, J., de Jesus, W., Isolani, P., Both, C., Rochol, J. and Granville, L. (2015). Software Defined Networking: management Requirements and Challenges. IEEE Communication Magazine, 53(1), 1 – 8.

[25] Xia, W., Wen, Y., Foh, C., Niyato, D. and Xie, H. (2015). A Survey on Software-Defined Networking. IEEE Communications Survey and Tutorials, 7(1), 27-51.