International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

A Unified and Scalable Programming language for SDN

Elakya R, M.Vishal, Allen Christopher, Devender Singh, Karthikeyan G

Abstract: ***In networking, the introduction of Software-Defined Networking (SDN) has brought many opportunities to both networking and computing community. Decoupling of the network management logic from data forwarding plane could be a key innovation in SDN that contributes to the introduction of network programmability. With the global view of the entire network, the centralized controller can manage the entire network through the customized control logic programmed for an individual use case. In this SDN, the network bandwidth is divides into multiple layers to produce various quality of network fuctions.. Network Function Virtualization became more feasible with the introduction of SDN, where the virtualized network function can move around dynamically around the network with SDN's dynamic network reconfiguration ability.***
In data center, cloud providers use network virtualization in order to store and compute the data to the users through which elastic service is provided to the users. The virtualized resources are shared and leased to cloud tenants, which are provisioned from the physical resources in the data center.

Index Terms- Software Defined Network (SDN), Quality of Service (QoS), Network Function Virtualization,Network Programability.

.

I. INTRODUCTION

SOFTWARE-DEFINED Networking (SDN) is an rising architecture that is used improve the performance of the network. It is simply said as an approach to cloud computing. In SDN, the network is defined by the software module that is used in network protocols. It aims towards the improvement of agility and flexibility of the network. It is widely used in business applications where it enables the enterprises and service providers. SDN programming languages are evolving and these make use of open protocols. It provides network programmability where modifications in the network can be done through the software.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
Ms Elakya R*, Assistant Professor In SRM Institute of Science and Technology, Chennai, India.
Mr M. Vishal , pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India.
Mr Allen Christopher, pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India.
Mr Devender Singh, pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India.
Mr Karthikeyan G, pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India.

Most of the network engineers, cloud computing engineers, Network administrators and other technical departments of network operation work on this in order to respond to the applications quickly
The SDN architecture is divided as 3 main layers: Data Plane, Control Plane, and Application Plane. The Data plane which are called as network devices are controlled or managed by a remote. The Control Plane studies and manages the whole network from a logically centralized point, contribution functions such as access control, orchestration, and network virtualization.
The interface between Application plane and Control Plane can also be implemented as high-level SDN programming languages. The network functions (e.g. routing, load balancer, firewall) are performed by the network applications in the Application Plane. OpenFlow is a standard interface b/w control plane and data plane but there is no standardization on the interface between application plane and control plane. As this interface is used for creating the applications that will modify the network behavior, the lack of standardization makes it hard for researchers and developers to know which features and abstractions are contributed by SDN can be useful to them; however, they also do not present any kind of standardization, focusing mainly on network policies and contributing abstractions focusing network operators.
There are many SDN programming languages where those are focused on the issues and those issues are solved based on some functionality sets. The works that are carried out in the beginning with their contributions are detailed review of our work. The discussed models are to represent the abstractions to solve the issues carried over. At last, all information is summarized and the main challenges to be faced and other research efforts are discussed.

II. EXISTING SYSTEM

Currently based on packet forwarding. Our existing system acts as network control platform. A high level programmatic interface for the development of network control applications is provided here. And its abstractions flip networking in a code downside. Here, Networks are controlled through low-level configuration of unique elements. However, configurations of this type usually depend on the primary network; as an example: Only by knowing the user's IP address, blocking of user's access with ACL entry is possible. More sophisticated tasks need a lot of intensive network knowledge; forcing guest users' port 80 traffic to traverse an Hyper TextTransferProtocol proxy needs the network topology and guest's location. In this manner, an enterprise network represents a pc without an OS, dependent network part configuration enjoying the machine language programming in hardware part.

A. DISADVANTAGES OF EXISTING SYSTEM

Currently focused on packet forwarding. The existing system ignore functionality such as bandwidth and packet-processing functions.

Declarative : Specify what, but not how
Low-level : Manage flow rule patterns, priorities, timeouts
Restrictive : Must use special programming language.

III. PROPOSED SYSTEM

The proposed system represents a high-level programming for
1. Packet classification
2. Forward path control
3. Packet-Processing functions (specifying)
4. Provisioning bandwidth

Programming with different controllers is comparable, although with different features. SDN controllers expose the switch capability, make programming SDN easier. The proposed application has a goal to provide a platform in which researchers and developers can develop innovative applications within enterprise network. Applications typically determine how each flow is routed or not routed in the network. Before the use of SDN the data plane and the control plane are used in the same medium as a hardware source and after the arrival of SDN into the networking field the data plane and control plane are separated such that data plane is fixed to the hardware device such as Data Drivers etc; and the control plane is set into the server.

The proposed system makes the following contributions:
1. High-level network control is presented and abstractions are realized in an expressive policy language.
2. It describes a completely unique compilation algorithmic program that selects forwarding methods and bandwidth is allocated using the formulation called mixed integer program and constraint convergent thinker.

A. ADVANTAGES OF PROPOSED SYSTEM:

- Simple and flexible conceptual model
- Reduces computation demand
- Significantly reduce the load on the controller
- Better abstraction at the language level

IV. PROPOSED SYSTEM ARCHITECTURE

The System Architecture of the our proposed system is as follows such that there exists three layers known as Application layer, Control Layer, Infrastructure Layer. For which Application layer takes care of all the business applications such as the console application, Window Application, Web Application. And the other layer is the control layer where the control of the system is done through this layer such that it has connection manager, Event Dispatcher, IO Sockets. And then Infrastructure layer exists such that it acts as controller which is used to control the network devices and connect to other network devices. We can say that SDN has many benefits from the software side over the hardware such as it brings innovation to the network industry.

Fig [1]: System Architecture (Block Diagram)

➢ *Working of Modules :*

A. INFRASTRUCTURE LAYER :

Infrastructure Layer is a type of layer in the system that has been proposed which acts as a hardware controller and also which is used to connect to the network devices. For this layer CLI (Command Line Interface) is used, where it is used to interact with the computer program where commands to the program are issued by the user in the form of successive lines of text. In simple words, it accepts text input to execute the OS (Operating System) Functions. CLI is associate older methodology for interacting with applications and software system. And These controllers are used to connect the other real networks from which it connects to various new network devices in the bounded path of the network zone. Here, We make use of CLI in order to make use of less memory when compared to other user interfaces.

B. CONTROL LAYER :

Control layer also acts as a controller which a bit different from the Infrastructure layer such as the infrastructure layer acts as the controller that which connects the other networks and network devices but whereas here, this controller acts as the controller that which manages the flow of network and control it to improved the network management and performance of the applications. Here we can use Java as a programmming language to make it simple and easy to build the controller for control layer which is used to watch the flow of protocols over the network. An SDN controller provides services that can realize a distributed control plane, as well as the concepts of breif state management and centralization.

C. APPLICATION LAYER :

In this application layer, we represent the business applications such as the Console Application, Windows Applications, Web Application where the data that has to be presented will be displayed in this layer such that the clients can access the data which is presented and that is the data which is passed through the other layers of the proposed sytem. In Applications, We usually have 2 types of applications called as SDN Appplications and Non-SDN Applications. SDN Applications are virtual applications which are used to control the Applications. In this SDN Applications, we just have to know what kind of SDN controller we are going to use. And Non-SDN Applications, we can use any type of controller like we can use any programming language.

V. HELPFUL HINTS

Fig [2] : Module Process

A. *CREATING CONTROLLER*

A Controller connects to real networks and also it connects to various network devices which are in the network path. Here in this SDN controller platform, it usually runs on a server and makes use of protocols to inform switches that where it has to send packets.

SDN Controllers direct traffic consistent with forwarding policies that a network operator puts in place, thereby minimizing manual configurations for individual networks.

B. TOPOLOGY :

Here we create topologies to be controlled by controllers. Overall network resources are described by the network topology. And collecting important network properties such as network management, Routing, quality of service(QoS) among the others as specified from their topologies.

C. IMPLEMENT TOPOLOGY :

The created topology is implemented using Merlin Language. As specified in the previous notes of the research work we can make use of any programming language that suits our system. Here we have chosen python as the medium of topology implementation such that this programming language is very efficient which is easy to make use and develop the required topology. Most of the work is completed once the topology is implemented successfully.

VI. RESULT ANALYSIS

Getting into the results, the main objective of the project is to provide a platform to developers and researchers the ability to innovate enterprise networks in the form of developing novel applications. In general, it simplifies network administration by providing high level abstractions with specifying network policies that provision network resources. That's the main reason we use SDN as the base in our project.

CONCLUSION

The conclusion of this proposed system is that network administration is simplified using Merlin and high level abstractions for specifying network policies that provision network resources is provided. Our application aims in providing a platform to the researchers and developers, in order to develop innovative applications. In general terms, the objective of the proposed system is to simplify the network topologies using software as a medium. Here, the work done by us is based on merlin which is a new type of framework which is used to manage resources in software defined networks (SDN).

A. Future Enhancement

The main problem that has been observed is the combined module of control plane and data plane in a hardware source such that the modification of anything should be done through the hardware source where it is difficult to arrange and provide the data through the hardware source. So, We made a conclusion through the research work which enhances the future development of Network topology through the software, which is determined through software defined network (SDN).

ACKNOWLEDGMENT

The authors would like to thank Mrs.R.Elakya for guiding us throughout the project. The concepts present here are borne through the research work done for this project.

REFERENCES

1. A Survey of Software-Defined Networking: Past, Present, and Future of ProgrammableNetworks: Bruno Astuto A. Nunes ; Marc Mendonca ; Xuan-Nam Nguyen ; Katia Obraczka ; Thierry Turletti : IEEE Communications Surveys & Tutorials.
2. Development of architecture of visual program system for distributed data processing in SDN:DmitryPerepelkin ;Maria Ivanchikova ; Ilya Tsyganov 2018, 28th International Conference Radioelektronika (RADIOELEKTRONIKA)
3. ULLA-X: a unified programmatic middleware for on-demand network reconfiguration: Avishek Patra ; Andreas Achtzehn ; Petri Mähönen, 2015 IEEE 40th Conference on Local Computer Networks (LCN)
4. On the Impact of Network State Collection on the Performance of SDN Applications: Mohamed Aslan ; Ashraf Matrawy; IEEE Communications Letters
5. Pairing SDN with network virtualization: The network hypervisor placement problem: Andreas Blenk ; Arsany Basta ; Johannes Zerwas ; Wolfgang Kellerer: 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)
6. A Software Engineering Perspective on SDNProgrammability:
7. Felipe A. Lopes ; Marcelo Santos ; Robson Fidalgo ; Stenio Fernandes:IEEE Communications Surveys & Tutorials

AUTHORS PROFILE

Ms Elakya R is working as assistant Professor in SRM Institute of Science and Technology,Chennai,India. Her area of subject interests include computer networks and IOT.

Mr M. Vishal is currently pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India. He has experience in the field of web development and administration, who is passionate towards the Cloud Computing domain. His skills include marketing and management, who is currently working in technical team of an esteemed Society.

Mr Allen Christopher is currently pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India. He has professional experience in Application Testing and security. His research interest includes Security Management.

Mr Devender Singh is currently pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India. He has professional experience in programming with high logical standards.

Mr Karthikeyan G is currently pursuing his Bachelor of Technology in Computer Science and Engineering at SRM University, Chennai, India. He has professional experience in programming with high logical standards.