International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Software Defined Networks: The New Norm for Networks

Gunjan Tank[1], Anmol Dixit[2], Alekhya Vellanki[3], Dr. Annapurna D[4]

[1, 2, 3]Department of Computer Science and Engineering, PESIT-BSC, Bangalore, Karnataka 560-100, India

[4]Department of Information Science and Engineering, PESIT-BSC, Bangalore, Karnataka 560-100, India

Abstract: ***Our paper deals with the Software Defined Networking which is in extensive use in present times due to its programmability that helps in initializing, controlling and managing the network dynamics. It allows the network administrators to work on centralized network configuration and improve data center network efficiency. SDN is basically becoming popular for replacing the static architecture of traditional networks and limited computing and storage of the modern computing environments like data centers. Operations are performed by the controllers with the static switches. Due to imbalance caused due to dynamic traffic controllers are underutilized. On the other hand controllers which are overloaded may cause switches to suffer time delays. Wireless networks involve no cabling, therefore it is cost-effective, efficient, easy-installable, manageable and adaptable. We present how SDN makes it easy to achieve end point security by checking the device's status. Local agents collect device information and send to cloud service to check for vulnerabilities. The results of those checks are sent to the SDN Controller through published Application Program Interfaces (APIs). The SDN Controller instructs Open Flow switches to direct vulnerable devices to a Quarantine Network, thus detecting suspicious traffic. The implementation is done using the data network mathematical model.***

Keywords: Software Defined Networking (SDN), Open Flow Switches, Cloud service, Quarantine Network

1. Introduction

SDN is an emerging network architecture where network control and forwarding are separate from each other. In addition to that, it is directly programmable.

According to the Open Networking Foundation (ONF), software-defined networking (SDN) is a network architecture that decouples the control and data planes, moving the control plane (network intelligence and policy making) to an application called a controller [1].

There is a lot of overhead involved in configuring the telnet boxes where there might be cases where configurations are done wrongly. SDN is basically the separation of the control plane and the data plane where the networking devices are controlled or updated using the Open Flow protocol.

SDN appears to be a single logical switch where network intelligence is centralized maintaining a global view of the network.

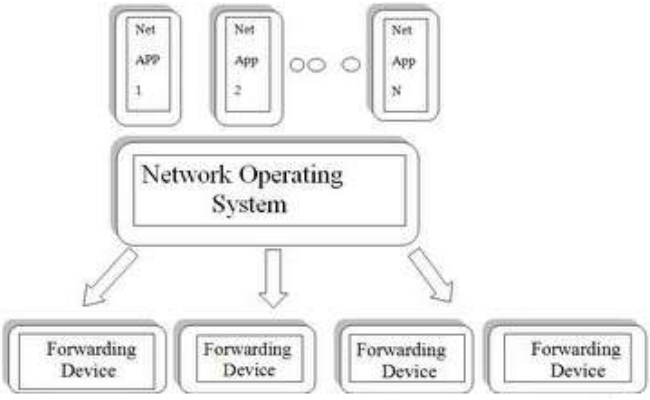

Figure 1: Basic Architecture

SDN meets all the needs of the current network technology which include complex policies, inability to scale and vector dependence.

In the traditional approach, network operators and administrators would have to hand-code tens of thousands of lines of configuration across a large number of devices. On the contrary, this simplified network abstraction can be configured programmatically thus saving a lot of time and effort.

Moreover, they can write these programs themselves and not wait for features to be embedded in vendors' proprietary and closed software environments in the middle of the network. In addition, network behaviour can be altered in real-time and new applications and network services can be deployed in a matter of hours or days, rather than the weeks or months needed today by leveraging SDN controller's centralized intelligence. By centralizing network state in the control layer, SDN gives network managers the flexibility to configure, manage, secure, and optimize network resources via dynamic, automated SDN programs.

In addition to abstracting the network, SDN architectures support a set of APIs that make it possible to implement common network services, including routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, processor and storage optimization, energy usage, and all forms of policy management, custom tailored to meet business objectives making it possible to manage the entire network through intelligent orchestration and provisioning systems.

Thus, with open APIs between the SDN control and applications layers, business applications can operate on an abstraction of the network, leveraging network services and capabilities without being tied to the details of their

implementation. SDN gives networks the advantage of not having to be very “application-aware" and applications the advantage of not having to be very "network-aware". As a result, computing, storage, and network resources can be optimized.

2. Understanding OpenFlow

Open Flow is the first standardised communication interface between the control and forwarding layers of SDN architecture. **OpenFlow allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual [1]**. It being implemented on both the sides of the interface, allows it to demonstrate how much traffic should flow based on parameters like usage, applications and cloud resources.

Working and forwarding of packets in OpenFlow:
In a classical router or switch, the fast packet forwarding (data path) and the high level routing decisions (control path) occur on the same device. An OpenFlow Switch separates these two functions. The data path portion still resides on the switch, while high-level routing decisions are moved to a separate controller, typically a standard server. The OpenFlow Switch and Controller communicate via the OpenFlow protocol, which defines messages, such as packet-received, send-packet-out, modify-forwarding-table, and get-stats.

Table abstraction is represented clearly by the data path of an OpenFlow; each flow table entry contains a set of packet fields to match, and an action. When an OpenFlow Switch receives a new packet, for which it has no flow entries, it sends this packet to the controller. The controller then handles this packet by taking appropriate decision. It can either drop the packet, or it can add a flow entry directing the switch on how to forward similar packets in the future.

OpenFlow technique allows user to easily deploy innovative routing and switching protocols in your network. Its various applications are virtual machine mobility, high-security networks and next generation IP based mobile networks.

SDN and Open Flow:
An SDN Controller in SDN is the “brains” of the SDN network, relaying information to switches/routers ‘below’ and the applications and business logic ‘above’. Recently, as organizations deploy more SDN networks, SDN Controllers have been tasked with forming a single centralized unit between SDN Controller domains, using common application interfaces, like OpenFlow and open virtual switch database (OVSDB).

3. Benefits of OpenFlow

a) Programmable Environment
- Enable innovation/differentiation
- Accelerate new features and services introduction

b) Centralized Control and Intelligence
- Simplify provisioning
- Optimize performance
- Granular policy management

c) Abstraction
- Decoupling of Hardware & Software, Control plane & forwarding, and Physical & logical configuration.

4. Cloud Sim

Cloud Sim is a discrete event based cloud simulator. It enables the simulation of data centers with a number of hosts and features to model cloud, virtual Machine and cloud Market. As mentioned in the paper, the utilization of simulation tools is a reasonable solution for conducting large-scale, repetitive, dynamical evolving and expensive experiment[2]. Other simulators for grid modeling and simulating platforms have been used in the past but none of them could support fully cloud computing. It guarantees both QoS and energy consumption.

Network Cloud Sim [3] simulates application with communication tasks in Cloud Sim. To enable flow of packets between Virtual Machines, a switch class is developed which performs SDN switch functions which is managed by controller. Forwarding packets depends on the network traffic, thus is dynamic in nature. Packets generated by VMs are forwarded to the destination (host) using forwarding routes. Channel remains common for the packets which are being generated from the same VMs. However, if any new channel is created, link updates the bandwidth among all the channels of that link. This is how the controller manages the overall network behavior of the simulation.

5. Securing End Point Security with OpenFlow and SDN

- **Checking Status**

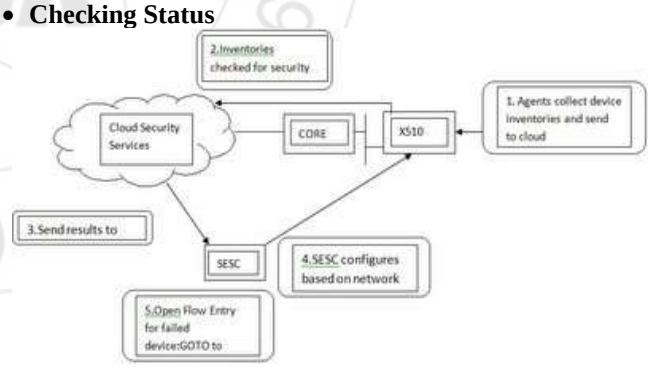

Figure 2: Process of checking status

- **Detecting Suspicious Traffic**

The OpenFlow Switch-Controller setup present in every SDN can be used to detect suspicious traffic. Every switch-controller pair communicates with each other by means of the OpenFlow Protocol. Traffic flowing in from the internet that may contain malicious content reaches the switch. It is then forwarded to the controller. The controller mirrors this traffic so that further partitioning can be done by the Security App. The clean traffic so obtained is forwarded to the corresponding enterprise network.

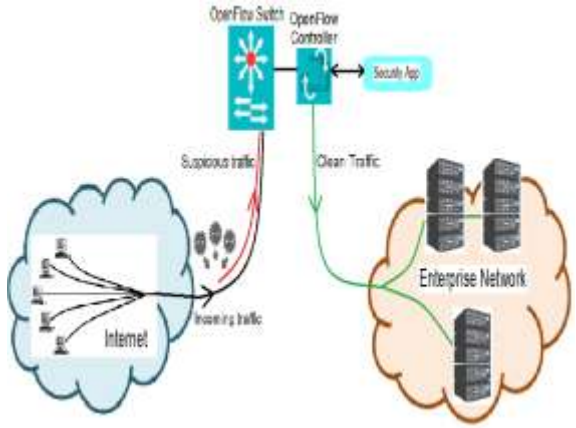

Figure 3: Detecting Suspicious Traffic

- **Log Analysis which helps in easy traffic monitoring**

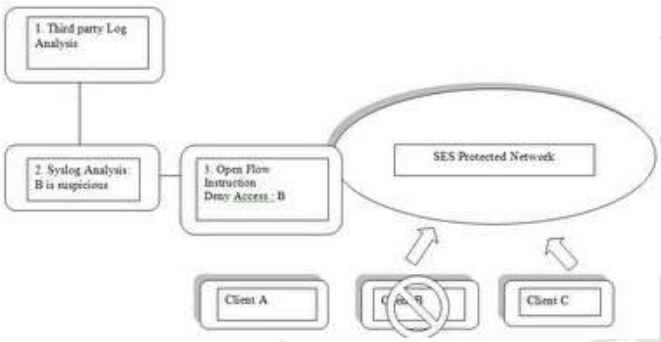

Figure 4: Log Analysis

6. Conclusion

In our paper, we summarize the need for Open Flow based Software Defined Network Architecture.SDN promises to convert the static network to the flexible, programmable platform dynamically with some intelligence. Enormous data centers and virtualization and secure cloud environments can get support through SDN. With number of advantages SDN is becoming new form of networks.

Increasing demand of cloud computing related services is scaled up by SDN and through simulations. SDN controller is programmable in the simulator. VM management policies and scheduling algorithms can also get tested.

7. Acknowledgement

This research paper couldn't have been possible without the help of few people. Firstly, we would like to thank our parents for their constant support and motivation. We would also like to thank our teachers for guiding us without losing their patience. We would be failing in our duty if we forget to thank PESIT-BSC for giving us the opportunity to display our work and talent

References

[1] Allied Telesis, "Practical Application Of SDN in Enterprise Networks
[2] Jungmin Son,Cloud Computing and Distributed Systems, "Cloud Sim SDN: Modulation and Simulation of Software Defined Cloud Data Centers"
[3] Brendan Ziolo "Cloud and Virtualization Require SDN"