**Impact of SDN on traditional Network**

Abstract – With the advancement of technology and new innovation bombarding the mankind, the dependence of the world on the internet is increasing at an unprecedented manner. More and more devices are using the internet and network architecture, be it our smartphones or smart televisions. The progress made in the new domains like Internet of Technology and cloud computing, has added weight to it. The current networking technology which is so called “Traditional Networks” is turning to be insufficient to cope the demand of the modern world. Hence, an alternative was needed and SDN or Modern Networks comes to rescue. The current network architecture is hardware oriented and thus it has its own set of cons. The SDN is software oriented which provides much more flexibility and advancement in maintaining the dynamic demands of the world. The main goal of SDN is to separate software logic from the hardware and to shift the entire software logic to a centralized system which then controls the entire network. This approach provides not only ease of maintenance and flexibility but also rapid scalability which is fundamental to tackle the needs. This paper embraces the importance and impact of SDN on traditional networking approach and also puts light on the basic tools and technologies used in modern networks along with the potential it has to revolutionize not just one but entire domain.

Introduction

In this modern era a significant advancement of internet communication and its service has been observed with the deployment of various new technologies which has been made simpler by just deploying it on the basis of networking standards. This has resulted in a large digital global network that can be accessed at any time and from any location throughout the globe. The internet has swiftly evolved from the first phase of just connectivity and digital access to information to serving a wide range of application and services including online service, database access, multimedia broadcasting, entertainment, computer to computer communication real-time audio and visual communication, and so on with no limit of its application at this moment. The elements utilised for the unique sophisticated design are controlled by their own firmware loaded in their storage space in this traditional network architecture. Traditional IP-based networks, however are nonetheless complex and difficult to administer, despite this well developed approach.

Of course the traditional network no wonder did marvelous job at the time of its introduction to the real world but as the hardware devices increased , network density increased , user increased the rise in complexity made made it difficult for the traditional architechture to accommodate all the stuffs as it made the architecture of the network too complex to handle and control everything. Here a network administrator must configure each and every network device separately in order to apply any desired network policies in a limited network architecture due of its scattered nature. This will take time and may result in increased complexity or the introduction of defects owing to individual hardware reconfiguration or load imbalance. The difficult task for network architecture is that the network must be dynamic and controllable without the intervention of a human operator. In a standard IP-based network, this type of automated hardware reconfiguration and administration task is lacking.

Traditional networks vs SDN

|  |  |
| --- | --- |
| Traditional Network | SDN |
| It has distributed control | It has centralized control |
| The network is not programmable | The network is programmable |
| It is tightly coupled to hardware | It is loosely coupled to hardware |
| It is a closed interface | It is an open interface |
| Data plane and network plane are mounted on the same plane | Data plane and network plane are separated by software |
| No priority support | Priority support availbale |
| Scalability on large scale becomes costly | Scalability is affordable |
| As network size increases, structural complexity also increases | Structural complexity remains significantly less compared to the counterpart approach |
| The adaptibility is less | The adaptibility is more |
| Difficult to debug | Easy to debug |
| Sustentetion is more | Sustentetion is less |
| Over the air update cannot change the configuration | Over the air update can change the entire configuration |
| Operating costs are high | Operating costs are less |
| More time is required to make changes and to provide new services | Speedy changes and provision of new services is possible |

Traditional Network Charachterstics

The traditional internetworking approach is based on the TCP/IP protocol architecture. Three noteworthy characteristics of this approach are as follows:

Two-level end system addressing

Routing based on destination

Distributed, autonomous control

The protocol architecture built around the TCP and IP protocols, consisting of five layers: physical, data link, network/Internet (usually IP), transport (usually TCP or UDP), and application.

Network interface IDs are commonly used in traditional architectures. Hardware-based identifiers, such as Ethernet MAC addresses, are used at the physical layer of the TCP / IP architecture to identify devices connected to the network. The architecture is a network of internetworking layers that spans both the Internet and the private Internet. Each associated device has a physical layer ID that is recognized by the direct network and an IP address that is a logical network identifier that allows visibility around the world.

This addressing system is used in TCP / Design IP to facilitate networking of autonomous networks under distributed control. This design is highly resilient and easily extensible when it comes to adding new networks. Routes can be identified and used throughout the Internet using IP and distributed routing protocols. To handle congestion, transport-level protocols such as TCP can be used to implement distributed and distributed algorithms.

Routing is used to be based on the destination address of each packet. Because this DataGramTechnique tries to determine the shortest path for each individual package, consecutive packets are used between the source and targets of the entire Internet and the target. Packets are often considered as packet streams to meet QoS requirements. The package associated with the flow comes with a specific QoS attribute that affects flow routing.

A method to transmit message was that Long messages are broken into short packets and sent across a communications network using this manner. Each packet is routed through intermediate nodes from source to destination. Each node receives the complete message, stores it briefly, and then forwards it to the next node.This dispersed, autonomous method, on the other hand, emerged at a time when networks were mostly static and end systems were mostly fixed. It has been discussed about the whole transmission here in [2].

Based on these characteristics, the Open Networking Foundation (ONF) cites four general limitations of traditional network architectures [ONF12]:

**Static, complex architecture:** Networking technology has become more complex and difficult to administer as a result of demands such as varying levels of QoS, large and variable traffic volumes, and security needs. As a result, there are a variety of different protocols, each of which tackles a different aspect of networking. When devices are added or moved, for example, this can be challenging. To make changes to configuration parameters in many switches, routers, firewalls, web authentication portals, and other devices, network administration employees must use device-level management tools. Changes to access control lists (ACLs), virtual LAN settings, QoS settings in a variety of devices, and other protocol-related tweaks are among the upgrades.Adjusting QoS parameters to match changing user requirements and traffic patterns is another example. On a per-application and even per-session basis, manual techniques must be employed to configure each vendor's equipment.

**Inconsistent Policies:** Staff may have to make configuration modifications to hundreds of devices and processes to apply a network-wide security policy. When a new virtual machine is enabled in a big network, reconfiguring ACLs throughout the entire network can take hours or even days.

**Inability to Scale:** The volume and variety of network demands are continually increasing. Because of the network's complicated, static architecture, adding extra switches and transmission capacity, including various vendor equipment, is difficult. Oversubscription of network links based on projected traffic patterns is one approach that businesses have utilised. However, as virtualization and the range of multimedia apps expand, traffic patterns are becoming more unpredictable.

**Vendor dependence:** Because of the nature of today's network traffic demands, organisations and carriers must quickly deploy new capabilities and services to meet changing business objectives and user demands. Due to a lack of open interfaces for network services, businesses are constrained by vendor equipment's very long product cycles.

As encountering the difficulties in traditional network a and technologies lying under it a new domain of research challenges the evolution of Next Generation Networks.[3]. For the efficient deployment of the Future Internet, a variety of options were considered in [4].

As described in [5,] a variety of research subjects and projects have already been undertaken by both government and private groups. One of the newest approaches for evolving networking paradigm is Software Defined Networking (SDN) , which is one of the many contributions linked with improving stiff existing infrastructure networks. It has helped to overcome the traditional IP based networks limitations hence completing all the major and minor requirements to be present. The Open Data Center Alliance (ODCA) provides a useful, concise list of requirements, which include the following :

**Adaptability:** Networks must be able to modify and respond in real time to changing application requirements, corporate policies, and network conditions.

**Automation:** Policy changes must be communicated automatically to reduce manual labour and errors.

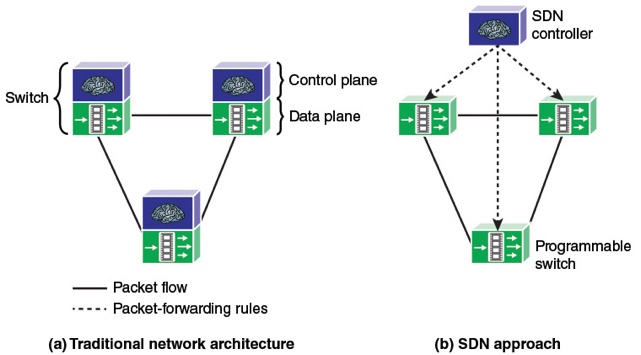
**Maintainability:** New features and capabilities (software updates, patches) must be effortlessly implemented with little downtime.

**Model management:** Rather than making conceptual changes by reconfiguring individual network pieces, network management software should allow management of the network at the model level.

**Mobility:** Control functions, including mobile user devices and virtual servers, must allow mobility.

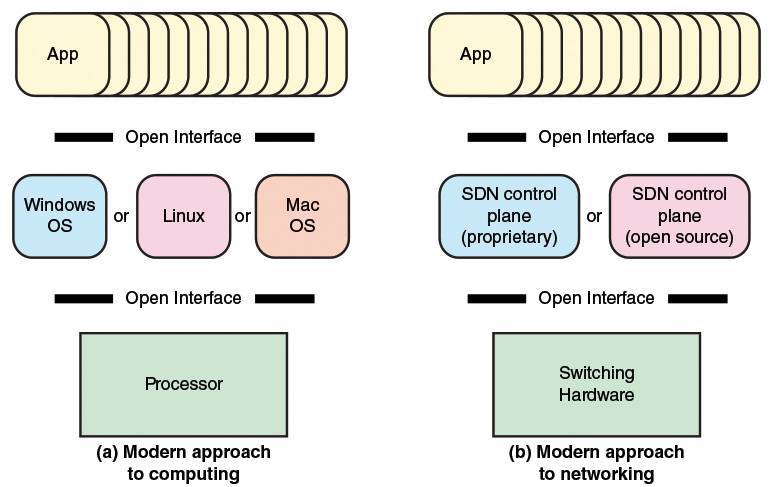
**Integrated security:** Instead of being an add-on solution, network applications must integrate seamless security as a core function.

**Scalability on demand:** Implementations must be able to scale up or down the network and its services on demand.

SDN is a type of programmable network that has the potential to surpass the limits of conventional network infrastructure in a variety of ways. To begin with, it decouples the control plane from the network infrastructure core devices that convey data traffic (also known as the data plane), which solves the vertical integration complexity problem. Second, because the data plane and control plane are separated, all controlling tasks are handled by a centralised software-based controller, whereas core devices are solely responsible for data traffic forwarding tasks as directed by the logically centralised controller. The controller in SDN is a type of controlling element that uses a well-developed programmable coding scheme to govern the underlying networking devices. As described in [8], the controller as well as the software operating inside the controller is referred to as Network Operating System (NOS) due to this design. Figure 1 depict a standard network design and a streamlined SDN network architecture that separates the control and data planes.

SDN ARCHITECTURE

The SDN controller is at the heart of a feedback loop because of its primary functions of giving virtualized views of underlying resources to clients and applications and orchestrating resources to continually satisfy all client/application requirements (Figure 2). The controller acts as a mediater between network resources and applications. The controller applies management-control functions to the resources within its scope, allowing them to modify their status in accordance with the administrator's policies. These include meeting contract agreements to all customers, but with prioritised regulations in place for exception instances.



When resource states or service requests change, the controller evaluates the situation by comparing the current real state to the requested state. When a difference is detected, the controller responds to it by imposing the optimization policies. When it is not possible to fulfil a service/resource request, it is rejected. The controller throws exceptions when it encounters a condition it can't manage on its own, or when a state arises that fits an application's subscription. If SDN control is implemented with several managercontroller components, attention must be made with respect to their rights and responsibilities to act on resources, according to the centralised control principles. Although different controller components may share resources, they must act in lockstep to avoid conflicts and inconsistencies. It is unlikely that the resources will be able to resolve disagreements on their own.

Throughout the lifecycle of a service, the required resources must be identified, assembled, and managed. The services may compete for existing resources because they are offered via the same infrastructure. Furthermore, service requests from applications are dynamic, necessitating a continual orchestration process to update and adapt.

NORTH BOUND INTERFACE

The SDN controller is accessed via the northbound interface. This gives a network administrator access to the SDN, allowing them to configure it or retrieve data from it. This can be done via a GUI, but it also includes an API that allows other apps to contact the SDN controller.

Some instances are as follows:

List the information from all of your network's devices.

Shows the status of all network physical interfaces.

On all of your switches, create a new VLAN.

Display the full network's topology.

When a new virtual machine is established, configure IP addresses, routing, and access-lists automatically.

SOUTHBOUND INTERFACE

In order to programme the data plane, the SDN controller must interface with our network devices. The southbound interface is used for this. This is a software interface, frequently an API, rather than a physical interface (Application Programming Interface).

An API is a software interface that enables a programme to provide access to other programmes via pre-defined functions and data structures. In a moment, I'll go through this in further detail.

The following are some of the most prevalent southbound interfaces:

OpenFlow is the most widely used SBI at the moment; it is an open source protocol developed by the Open Networking Foundation. OpenFlow is supported by a large number of network devices and SDN controllers.

Cisco OpFlex is the company's response to OpenFlow. It's also an open source protocol that has been proposed for standardisation to the Internet Engineering Task Force (IETF).

CLI: For the current generation of routers and switches, Cisco offers APIC-EM, which is an SDN solution. It makes use of protocols like telnet, SSH, and SNMP, which are all available on current-generation systems.

Why companies are shifting to SDN?

Vendors are turning to SDN as data centres continue to evolve and traditional networking fails to adapt to the changing environment. Here are a few of them.

To begin with, the widespread use of cloud services necessitates users having unrestricted access to infrastructure, applications, and IT resources.Second, IT is becoming a consumer commodity, and the BYOD (bring-your-own-device) trend necessitates networks to be flexible and secure enough to protect data and assets while still meeting compliance requirements and standards.

Traditional networking, on the other hand, is unable to fulfil rising demands due to the need to adhere to product cycles and proprietary interfaces common in vendor-specific contexts. When it comes to customising network content, network operators are frequently stymied. Traditional networking is complicated and time-consuming when it comes to adding and transferring devices or increasing capacity. Individual devices and consoles must be accessed manually. SDN is becoming a viable alternative because it allows administrators to instantly configure resources and bandwidth, bringing flexibility, efficiency, and robustness to the data centre. It also reduces the requirement for further physical infrastructure investments.

## How does SDN support edge computing, IoT and remote access?

## The core idea of SDN has been influenced by a number of networking trends. Using a correctly configured SDN environment, distributing computing resources to remote sites, shifting data centre tasks to the edge, adopting cloud computing, and supporting Internet of Things environments can all be made easier and more cost effective.

## Customers can often see all of their devices and TCP flows in an SDN environment, which means they may slice up the network from the data or management plane to enable a number of applications and configurations, according to Capuano. Users can, for example, more easily segment an IoT application from the production world.

Some SDN controllers have the intelligence to detect network congestion and, in response, increase bandwidth or processing to ensure that remote and edge components do not experience latency.

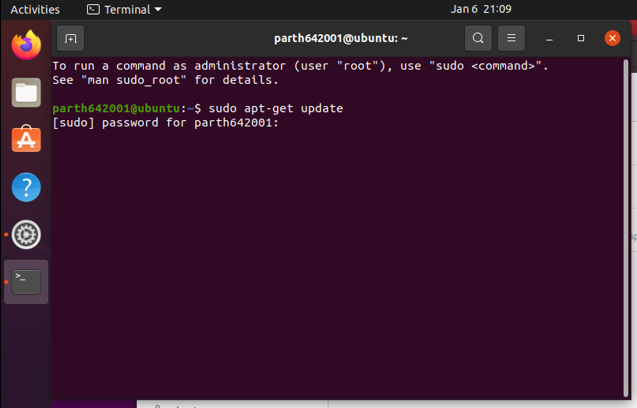
## According to Michael Bushong, vice president of enterprise and cloud marketing at Juniper Networks, SDN technologies can also help in distributed locations with few IT people on site, such as an enterprise branch office or a service provider central office.

INTRODUCTION TO MININET

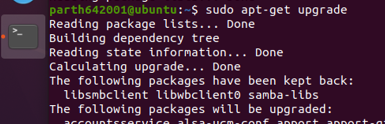
Mininet is a network tool and protocol testing virtual testbed.We'll learn how to utilise mininet via the CLI (Command Line Interface) as well as the GUI version, Miniedit.Mininet allows you to establish a realistic virtual network on any PC.Mininet has the following capabilities:Prototyping new networking protocols in a short amount of time. Testing for complex topologies is simplified without the requirement for expensive hardware.It executes real code on the Unix and Linux kernels, giving it a realistic feel. A huge community contributes rich documentation to an open source environment. Mininet was created as a testbed for OpenFlow2 and Software-Defined Networking (SDN).

**INSTALLATION STEPS:**

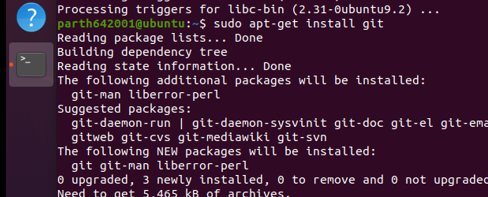
To install the mininet, follow the below steps



Then follow the second step



Then install GIT



Then enter the following command



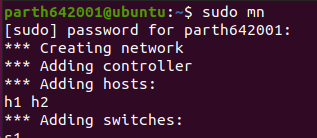
Then traverse to the mininet folder and enter the below command



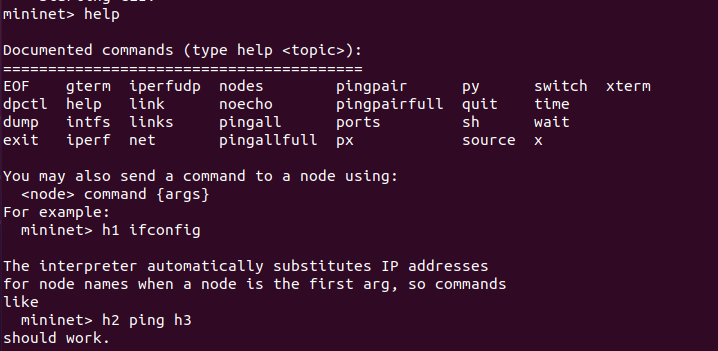
Then select and the version of the mininet and enter the below commands



Enter the following command to start the mininet in the terminal



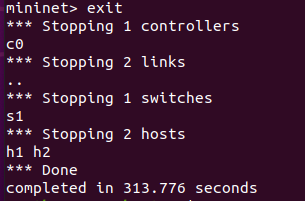
To display the list of Mininet CLI commands and examples on their usage, type the following command



To display the available nodes, type the following command



Stop the emulation by typing the following command



Creating a network topology with mininet and miniedit

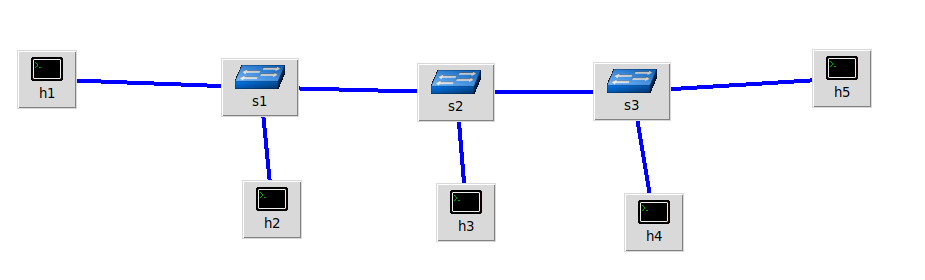
To start the miniedit, Enter the following command



The following window will be prompted

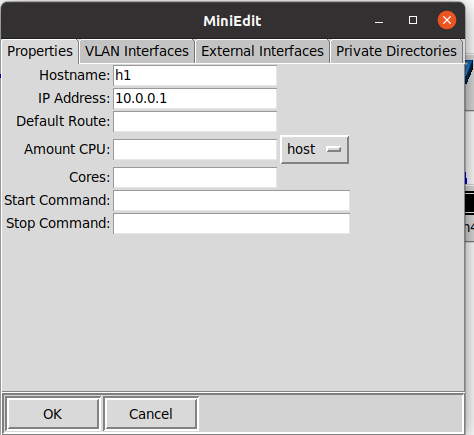


Create the following topology with the help of miniedit components



Now, right click on h1, then click on properties option

The following option will open

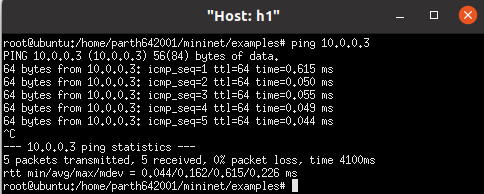


Follow the same procedure for all the hosts.

Click on Run button

Right click on h1 and select Terminal option.

Then perform the ping operation.



Thus, the connection is established.