ISSN No:-2456-2165

Software-Defined Networking and its Induction in Cloud Computing

A Newer Approach to Networking and Modernizing the Cloud

Sushovan Banerjee (*Author*)
B. Tech Computer Science
Kalinga Institute of Industrial Technology
Bhubaneshwar, India

Aniket Pathak (*Author)*
B. Tech Computer Science
Kalinga Institute of Industrial Technology
Bhubaneshwar, India

Abstract:- SDN principles may be traced back to the separation of the control and data planes, which was initially utilized in the public switched telephone network to facilitate provisioning and management long before it was adopted by data networks. The Ethane project at Stanford's computer sciences department gave birth to the usage of open-source software in split control/data plane systems. The creation of OpenFlow [1] was inspired by Ethane's simple switch design. Network controllers can use OpenFlow to determine the path network packets go through a network of switches. The OpenFlow standard is managed by the Open Networking Foundation (ONF), a user-led organization dedicated to the development and acceptance of software-defined networking (SDN). OpenFlow is specified by the ONF as the first standard communications interface defined between an SDN architecture's control and forwarding layers. Networks in businesses must be dependable. For many years, this has been assumed. Flexibility was not a consideration. Software-defined networking (SDN), on the other hand, is transforming the way IT and administrators think about network architecture.

The original use case for an SDN was to visualise the network by separating the system's control plane from the data plane where traffic flows. The data center's network traffic is handled by a smart controller running specific software, as well as a series of routers and switches that forward packets of traffic. Network virtualization has several benefits: networks may be dynamically scaled up and down, fine-tuned for specific application use cases, and security policies can be implemented on each individual server. Software-defined networks (SDNs) eliminate network hardware limits, allowing you to create more usable and responsive network infrastructures. While the benefits of SDNs for on-premises systems are widely known, installing them in the cloud can provide a significant advantage. A wise IT manager can use a hybrid SDN and cloud architecture strategy to gain the cost-effective agility needed to respond to the organization's infrastructure needs while also being able to proactively address important security risks.

The subsequent sections cover the majority of the topics needed to comprehend this material. Readers should feel free to look up further information in the reference section.
Keywords:- *SDN, OpenFlow [1], Control and Data Plane [2], ONF [4], SDN in Cloud, hybrid SDN [19]*

I. INTRODUCTION

Traditional networks are complex in general, and maintaining them is difficult due to the fact that the control and data planes [2] are merged in network parts (nodes). The control plane determines the paths and sends them to the data plane. The only way to update the flow management (forwarding policy) created using this approach is to change the network node's configuration. As a result, its operators must configure each individual network equipment (i.e. switches and routers) separately to express the proper high-level network regulations. Several low-level and vendor-specific commands are generally used to configure the system.

In addition to network element configuration challenges, network infrastructures must adapt to traffic variations and withstand fault dynamics. Traditional IP networks, on the other hand, have ineffective automatic reconfiguration and response techniques for enforcing the policies that are required.

Unlike traditional IP networks, which have the control and data planes closely coupled and included in the same networking elements, SDN separates the control and data planes. In SDN, control is transferred from network parts to a separate, centralized controller. In SDN, the control and data planes are separated, which offers various advantages: It can (1) break vertical integration and (2) make policy enforcement, network (re)configuration, and evolution easier. Because network elements (such as switches and routers) have been reduced to mere forwarding devices and a logically centralized controller exists, this is the case.

SDN can be investigated from a variety of angles. **Figure 1** depicts a three-dimensional view of SDN's layers, planes, and system design [3].

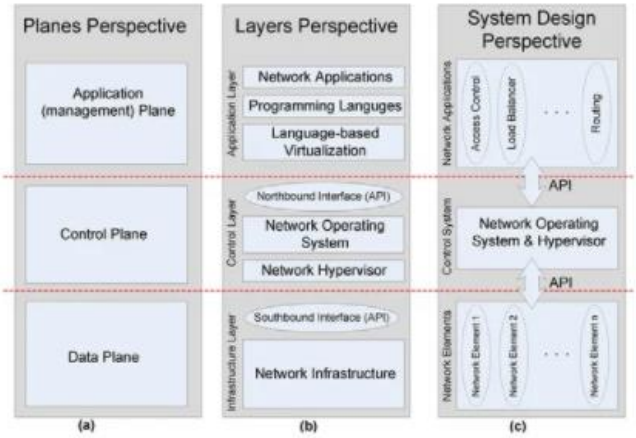

Figure 1: Three-dimensional view of SDN's layers, planes, and system design

II. NEED OF SOFTWARE-DEFINED NETWORKING

Traditionally, networks have been defined and managed by their tangible qualities, such as the hardware and wiring that connects points physically. Software-Defined Networking (SDN), a rapidly evolving network design, is reversing this trend. SDN decouples network control from network hardware. It employs software tools to intelligently program your network via centralized control. This means that the underlying hardware and technology are still present, but they are now controlled from a central location. As a result, you'll be able to manage your entire network consistently and holistically, with maximum flexibility and speed.

A. Why are operators and businesses adopting SDN?

Businesses are under unprecedented pressure to reduce expenses in order to compensate for stagnant earnings. However, multimedia, cloud apps, and mobile usage continue to develop at a breakneck pace. SDN technology is being used by enterprises, carriers, and service providers like you to deal with market demands and transform network architecture and operations.

Open APIs are at the heart of SDN, allowing software applications to program network behaviour from a central location. With those formerly restricted networks now open, networks may be managed from a single location, including all devices, endpoints, and infrastructure. SDN is meant to operate with complicated networks and make them readily managed and centrally run, regardless of how sophisticated the network technology is.

SDN offers a lot of benefits to businesses, operators, and providers, including:

- **Programmable networks:** Historically, the gear that controlled the network was only as good as the network itself. All of that changes with SDN, which allows for quick adjustments down to the individual client level. With hardware and software decoupled, innovative, differentiated new services can be provided quickly—something previously unimaginable with the limits of closed and proprietary platforms.
- **Centralized intelligence and control:** Bandwidth management, restoration, security, and policy have long been a source of frustration for network operators. Now that those services are centrally handled by a highly intelligent and optimized SDN controller, the network has a holistic perspective. That is a business asset, not a legacy liability. The network resources can be controlled and managed in a coordinated manner to deliver services end to end with network control centralized. Furthermore, devices are now aware of the state of the network as a whole.
- **API-based network interaction:** SDN eliminates the need for static physical hardware and network connections. Network hardware and connectivity are no longer a constraint for services and applications. Instead, applications leverage APIs to integrate OSS/BSS, orchestration, and assurance systems through the network infrastructure in a flexible manner.
- **Vendor-neutral architectures:** SDN allows an open, vendor-neutral approach that supports a wide range of applications. SDN supports cloud orchestration, SaaS, and business-critical networked apps, to name a few applications. Intelligent network services and applications run in a shared IT environment with SDN, which may govern hardware and associated technologies from a wide range of vendors.

III. ARCHITECTURE OF SDN

A network can be divided into two parts: a data plane and a control plane. The data plane is in responsible of forwarding data according to flow rules, whereas the control plane is in charge of defining the flow rules and control choices needed to get user input to the right place. In conventional networking, all of this is contained in a single box (e.g. Routers). SDN decouples network management from inter-networking devices and places it in the hands of a logically centralized controller, allowing these devices to serve as general-purpose data forwarding devices. The Open Networking Foundation (ONF) [4] definition of SDN is used in this article for clarity: "In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications." SDN focuses on four main characteristics: (1) Separation of logical intelligence from machines. (3) Data logic and control logic (i.e. controller and devices) APIs) (2) A central location for all intelligence and control (4) Programmability as a source of innovation (5) Improved security and dependability thanks to total network visibility and control.

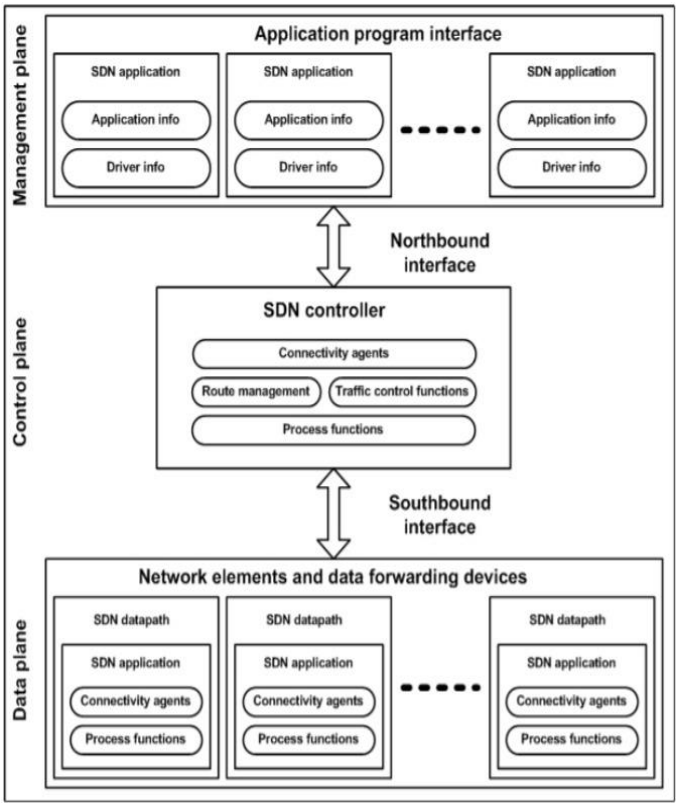

Figure 2: SDN Architecture model

A. The Role of OpenFlow and Open Source in SDN Architecture

Traditional networking architectures bundled both the SDN control plane and data plane pieces in one or more proprietary vendors distribute proprietary, integrated code. The OpenFlow open-source standard, which defined how the control and data plane elements would be separated and communicate with each other via the OpenFlow protocol, was recognised as the first SDN architecture in 2008. The Open Network Foundation (ONF) is in charge of maintaining the OpenFlow standards. OpenFlow is not the only protocol that makes up SDN; there are additional standards and open-source organizations offering SDN resources.

B. The SDN Stacks [5]

The separation of control and data forwarding operations in the SDN architecture is referred to as "disaggregation" since these parts may be supplied separately rather than as a single integrated system. Unlike traditional networks, which are merely application-aware, this design provides additional information about the state of the entire network from the controller to the applications.

SDN architectures typically consist of three components or functional groups:

- **SDN Applications:** SDN apps are programs that use application programming interfaces to communicate behaviors and resources with the SDN controller (APIs). Furthermore, the applications can construct an abstracted image of the network by gathering information from the controller for decision-making purposes. These programs include network management, analytics, and business applications that are utilized to run large data centres. An analytics software might be built to detect suspicious network activity for security concerns.
- **SDN Controller:** The SDN controller is a logical entity that receives and distributes SDN application layer instructions or needs to networking components. The controller also gathers network data from hardware devices and sends an abstract representation of the network to SDN applications, which includes statistics and events about what's going on.
- **SDN Networking Devices:** The SDN networking devices are in charge of the network's forwarding and data processing capabilities. This includes the data path's forwarding and processing.

Northbound and southbound interfaces define communication between applications, controllers, and networking systems, and are commonly referred to as SDN architectural APIs. A northbound interface connects the controller to applications, whereas a southbound interface connects the controller to actual networking devices. These pieces do not have to be physically positioned in the same area because SDN is a virtual network overlay.

IV. EVOLUTION OF SDN

In 2017, the physical network was the most profitable section of the global data-center SDN industry, accounting for approximately $2.2 billion in revenue, or around 42% of total revenue. The physical network, on the other hand, is predicted to generate $3.65 billion in revenue in 2022, somewhat less than the $3.68 billion attributed to network virtualization overlays/SDN controller software but more than the $3.18 billion attributed to SDN applications.

We've reached a point where SDN is better understood, where most datacenter network purchasers are familiar with its use cases and value propositions, and where an increasing number of businesses are discovering that SDN solutions provide practical benefits. The network is reclaiming lost ground and moving into closer alignment with a wave of new application workloads that are producing real business outcomes, thanks to SDN expansion and the transition toward software-based network automation.

A. Edge Computing, IoT and Remote Access [6]

A lot of networking trends have affected the basic concept of SDN. Distributing computing resources to remote sites, moving data center services to the edge, adopting cloud computing, and supporting Internet of Things environments may all be made easier and more cost-effective with a properly designed SDN system.

In an SDN environment, customers may frequently view all of their devices and TCP flows, allowing them to slice the network from the data or management plane to enable a variety of applications and configurations. Users can readily distinguish an IoT application from the industrial world, for example.

Certain SDN controllers have the intelligence to detect network congestion and enhance bandwidth or processing to avoid delay for distant and edge components.

SDN technologies are also beneficial in scattered locations with little IT employees, such as a branch office or a central office for a service provider.

Naturally, these locations necessitate centralised and remote connectivity, visibility, and security. SDN solutions increase operational reliability, speed, and experience by centralising and abstracting control and automating operations across numerous locations in the network and their devices.

B. Intent Based Networking with SDN support

Intent-based networking (IBN) [7] is a concept that allows network managers to describe what the network should accomplish and then have an automated network management platform produce the appropriate state and enforce regulations to guarantee that the company's goals are met. If abstract control of an infrastructure fleet is a key component of SDN, the deployment paradigm and dynamic control to maintain the infrastructure's health must be elevated. The recommendations shift away from device-specific information, obligatory and reactive commands, and toward declarative goals. Intent-based networking is a progression of software-defined networking (SDN) that aims for even more closed-loop functionality, operational simplicity, and automated intelligence. As a result, IBN is a significant step forward in the development of autonomous infrastructure, which will comprise a self-driving network that, like a self-driving automobile, would provide desirable results dependent on the goals of network operators and their organizations. While a self-driving car is designed to safely transport passengers to their destination with minimal human intervention, a self-driving network, as part of autonomous data-center infrastructure, will eventually achieve similar results in areas such as network provisioning, management, and troubleshooting — delivering applications and data, dynamically creating and altering network paths, and troubleshooting.

C. Network Security with SDN

SDN network security [9] must be present throughout a software-defined network in SDN deployments (SDN). To safeguard the availability, integrity, and privacy of all connected resources and information, SDN security must be incorporated into the architecture and supplied as a service.

Here are the Advantages, Advancements and Enhancements in the field of Network Security -

1. Advantages

SDN provides a number of security advantages. A client can divide a network connection between an end user and a data centre, with different security settings for different types of network traffic. On a network that does not process critical data, a public access network with minimal security might exist. Another component might have far more precise remote access control, such as a software-based firewall and encryption policies that enable sensitive data to get through. For example, if a customer has an IoT pool that they believe isn't particularly safe, they may utilize the SDN controller to isolate that pool from high-quality and critical business traffic. Security rules may be implemented across the network, from the data center to the periphery, using SDN. As a result, white box solutions might be 30 to 60% less expensive than traditional devices.

The capacity to watch a set of workloads and verify that they comply with a certain security policy is a fundamental aspect of SDN, which is especially essential when data is broadcast.

2. Advancements

It's no surprise that as the security stack at the network's edge grows more complex, network architects and security experts are seeking for new ways to secure their networks. When the number of data breaches began to increase in 2013, SDN appeared to be the answer.

Although software-defined networking did not completely transform network security in 2013, the potential to increase network defence remains, and technology is catching up. SDN could have a favorable impact on network security in the future in the following areas, to name a few:

- **Data Routing Through a Centralized System:** One of the most appealing characteristics of SDN is its ability to route all traffic through a single central controller. In terms of network security, SDN may be used to route data packets through a single firewall and increase the effectiveness of IDS and IPS data gathering.
- **Simplification of VLAN Configuration:** Organizations who utilize VLANs for security reasons understand how difficult it is to manage setups, especially when there are thousands of them. SDN makes it simple to automate settings while also enhancing their traceability.
- **Ease Pressure Off of the Perimeter:** Today, network perimeters are commonly employed to defend internal networks against a number of threats. With flow-based security processing, SDN enables a more dynamic way to route traffic through security appliances and apps, alleviating some of the pressure on perimeter defence.
- **More Effective Policy Management:** SDN allows for central management of security rules rather than physically deploying security systems, making network operator tasks more efficient.

There are numerous other potential applications for SDN in network security, but the basic line is that correctly implementing a software-defined network allows you to go beyond simply preventing specific assaults to proactively responding to new threats.

3. Enhancements

SDN provides a centralized entity controller with a complete picture of the network across the system, programmability through open application programming interfaces, and policy control, offering a choice of alternatives to improve security and threat mitigation [8]. SDN provides a

new platform for the development of innovative security approaches. SDN-enabled networks provide a central site for network device data gathering, as well as unique security measures that necessitate a centralized data model that was previously unavailable in traditional networks. This is a drastic change that has favorable implications for many network monitoring and firewall methods.

- **Network Monitoring:** Network monitoring is a critical component of network security. In fact, aberrant traffic patterns can be found by collecting real-time network data and analyzing it for security flaws using a variety of anomaly detection techniques. Before beginning an attack, an attacker can use scanning tools to learn about network behavior. In this circumstance, network monitoring becomes considerably more important. Network monitoring in SDN based on open flow is the gathering of flow-based data at the controller side, which is a natural open flow process in SDN. There are two ways to accomplish this. When a switch informs the controller that a flow has expired, it is done using the push operation (FlowRemoved message). Another method is the pull operation, in which the controller sends FlowStatisticsRequest and FlowStatisticsReply messages to the forwarding devices, inquiring about the state of flows. Push operation is demonstrated by FlowSense.
- **Network Verification and Automation:** In SDN, inconsistencies and policy violations can emerge if several controllers, different applications, and multiple users are all executing at the same time in the same domain. This can cause network issues like loops, black holes, and access control issues. In large networks with many switches, controllers must install thousands of flows and manage a large number of flow tables; controllers can install around 50,000 new streams per second; and security consistency, non-critical failover, and rapid failover must all be accomplished quickly and efficiently. Property-based validation tools that detect various network misconfigurations are the good job here, FlowChecker. FlowChecker [10] uses binary decision diagrams and the encoding switch flow table to generate a state machine that displays the flow statistics of the OF forwarding devices on the network. NICE is another troubleshooting tool in SDN settings. Furthermore, unlike these solutions, which are employed prior to the network launch or application installation, VeriFlow is an on-the-fly arrangement that verifies network accuracy in real-time as the network progresses.The FORTNOX error checking system embedded into the NOX controller detects conflicting flow rules in real time.

- **Improvised Threat Detection:** The SDN driver gives you a complete picture of your devices, which is great for threat detection. Open flow switches, unlike L2 learn switches, do not have a standard communication policy; instead, they follow the controller's commands, and the controller can reprogram data plane devices on the network to undertake network analysis for suspicious data and hostile devices. Traditional security systems are unable to detect harmful payloads at the application level and only provide Layer 3 and Layer 4 security. All packets must be transmitted to the controllers to provide application-level security in SDN, putting overhead on the controller and its bindings.

To address this issue, an algorithm based on the amount of lost connection attempts owing to bad requests was created. Only questionable packets are delivered to controllers based on the algorithms supplied. Microsoft's data centre also deployed SDN technology to detect fraudulent traffic. Traditional packet inspection solutions like port mirroring and switch port analyzer (SPAN), which necessitate a large number of physical ports and billing systems, are not feasible with Microsoft's massive infrastructure. Using virtual ports and the controller, this may be readily setup in SDN. Radware has used the SDN platform to create innovative security solutions such as DefenseFlow, which detects malicious network attacks such as DoS. There's also an open source version for research and development.

- **Dynamic Response to Threats:** SDN's system-wide network overall image, programmability via open application programming interfaces, and policy control via a centralized entity controller help security vendors and researchers. This opens up new ways to respond to threats in a more dynamic way. Because older networks lack centralized supervision, the only way to avoid malicious traffic is to rule it out. However, using SDN, we may dynamically reprogram switches through the controller to divert traffic for forensics. FRESCO and FORTNOX are two instances of SDN-assisted dynamic reactions to threats.

V. SDN IN CLOUD COMPUTING

SDN is a new network architecture in which "network control functionality" is separated from "forwarding functionality" and is programmable directly. The underlying infrastructure can be "abstracted" for applications and network services thanks to the relocation of control, which was previously firmly integrated in each networking equipment, to accessible computing devices (logically centralized). Enterprises who use OpenFlow-enabled SDN as the connectivity foundation for private and/or hybrid cloud connectivity will see a number of benefits. A logically centralized SDN control plane will provide a holistic picture of cloud resources and access network availability (abstract view). This will ensure that cloud-federation traffic is routed to properly resourced data centres via connections with suitable bandwidth and service levels. The following is a high-level list of critical building pieces for an SDN-based cloud federation: 1) OpenFlow enabled cloud backbone edge nodes that connect to the enterprise and cloud provider data centres, 2) OpenFlow enabled core nodes that efficiently route traffic between these edge nodes 3) an OpenFlow and/or SDN-based controller for configuring flow forwarding tables in cloud backbone nodes, as well as a WAN network virtualization application finally 4) Hybrid cloud operation and orchestration software to handle enterprise and provider data center federation, inter-cloud workflow, compute/storage resource management, and inter-data center network management. SDN-based federation will enable multi-vendor networks between enterprise and service provider data centers, allowing

enterprise clients to select best-in-class suppliers while avoiding vendor lock-in; choosing the appropriate access technology from a larger range (e.g., DWDM, PON, etc.); and more, access dynamic bandwidth for ad-hoc, real-time workload movement and processing between data centers; and free yourself from the weight of unused, expensive high-capacity fixed private leased lines. Cloud service orchestration logic and customer requirements drive SDN-enabled bandwidth-on-demand services, which allow automated and intelligent service provisioning.

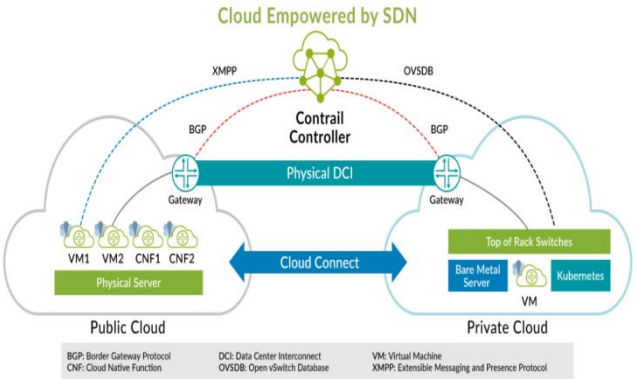

Figure 3: SDN orchestration architecture and its integration with cloud

A. Emerging Cloud Technologies

With the advent of (centralized) cloud computing, computing technology has entered a new era. The three types of services offered are Easy Software Service (SaaS), Easy Platform Service (PaaS), and Easy Infrastructure Service (IaaS) (IaaS). SaaS gives cloud customers access to the entire software package (for example, cloud-based email services, social media services, and scheduling services). Application developers can use the platform services provided by the PaaS cloud. Finally, IaaS cloud clients provide virtual machine servers and associated infrastructure. The use of IaaS can be used to implement SaaS and PaaS. DCs in centralized clouds, on the other hand, are geographically centralized and placed far away from end devices and users. As a result, they frequently fail to meet the needs of a variety of new real-time applications.

To solve these issues, various innovative concepts and architectures have been proposed in addition to centralized cloud computing. In the sections below, we categorize different cloud-related designs and concepts.

- ***Cloud-Related Architectures***

Several innovative cloud-related designs have been presented in recent years to satisfy the needs of various expanding applications, as previously indicated. These designs may be described using DC characteristics such as DC location, number, and size.

One or more DCs, as well as a number of end users, make up a cloud-related architecture. They are all connected via telecommunication networks. In this scenario, based on DC characteristics, we can classify cloud-related architectures into four basic groups: fully centralized (FC), partly distributed (PD), fully distributed with centralized controller (FD-CC), and fully distributed with centralized controller (FD-CC) (FD).

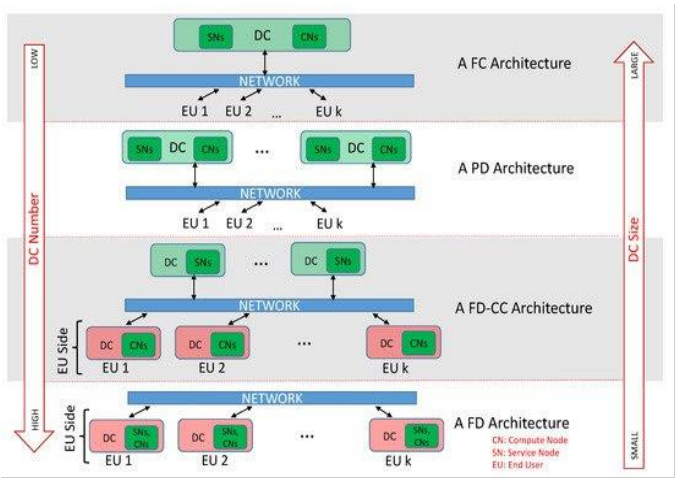

Figure 4: The cloud architecture taxonomy; ranging from fully centralized to fully distributed.

The FC design includes a large-scale DC, which all end users connect to and get services from. The PD design consists of numerous geographically dispersed DCs connected by a telecommunications network. It also connects DCs to end users. In comparison to the FC design, DCs in the PD architecture are typically closer to end users. However, DCs are no longer located on the premises of end customers. Several DCs are situated on end users' premises in the FD-CC architecture (i.e., one DC for one or a few end users). Controllers, on the other hand, stay in one or more DCs in the network's core. DCs deployed on end customers' premises are generally the size of a physical machine (PM). The remaining DCs are just for network administration and do not provide computation or storage. Finally, unlike the FD-CC design, the FD architecture spreads the management system to the locations of end users. It indicates that there is no DC in charge of resource management in the network's core.

- ***Hybrid Cloud-Fog Computing [11]***

The physical distance between a cloud service provider's CDs and end users (devices) is regarded a restriction for centralised cloud computing, as previously stated (i.e., FC architecture). Many future applications that need low latency, such as Public Protection and Disaster Relief (PPDR), real-time telesurgery, and autonomous driving, may not be compatible with the processing needs of the centralized cloud due to the vast distance.

Fog Computing [12] can work in line with cloud computing and provide lower latency by extending the architecture to the edge of the network and dedicating some calculations to the edge. It's crucial to keep in mind that fog is intrinsically linked to the existence of a cloud, and it can't exist without one. While fog resources help provide the best quality of service (QoS) for services, fog devices with limited capacity cannot replace the cloud's enormous resources. Fog Processing, in reality, may narrow the gap between the cloud and end devices (such as IoT nodes) by bringing some resources (such as computing power, storage, and network)

and data management closer to IoT nodes (for example, network nodes). Decisions on resource allocation and data management may be made not only in the cloud, but also throughout the data flow from the IoT to the cloud in this instance (preferably close to the IoT devices). It's also worth mentioning that putting resources closer to the edge can extend the life of edge devices while lowering network traffic at the core.

Several research on fog computing have proposed a three-layer generic architecture: The IoT devices (or edge) layer, the fog layer, and the cloud layer are the three layers.

The edge layer, which has extremely limited computation, energy, and bandwidth, is made up of terminal nodes, embedded systems, sensors, and actuators. The fog layer is made up of intermediary networking devices or nodes including routers, gateways, switches, and access points that use multiple protocols. These devices have both computational and storage capability. The last layer consists of cloud DCs with substantial virtual storage and processing capabilities. IoT devices (end-users) can connect with fog (i.e. fog nodes) over the local area network (LAN). The connection between IoT devices (end-users) and the Cloud, on the other hand, takes place through a wide area network (WAN), whether over fog or otherwise.

- ***Edge Computing***

Edge computing [13], as defined by OpenEdge computing, is computation performed at the network edge using small DCs close to users. It is a concept that moves computing resources from central data centers or public clouds closer to devices, which are embedded at the network's edge. The goal is to process data more quickly, which is required for many emerging applications, while also lowering network costs. Fog computing is a term used wrongly to describe edge computing. Fog computing, according to the OpenFog Consortium, is hierarchical and provides computation, storage, networking, control, and acceleration anywhere from the cloud to devices. Edge computing, on the other hand, is usually limited to computing at the edge.

B. SDN – the solution for cloud service providers

If you're a cloud service provider [14] wanting to ensure that clients can move their virtualized workloads without having to prepare, SDN might be the answer. SDN not only makes today's networks easier to manage, but it also allows cloud service providers to host millions of virtual networks without the need for traditional network separation methods like VLANs. SDN also allows network managers to administer network services using a central administration tool by virtualizing physical network connectivity into logical network connectivity.

Here are some of the specific advantages SDN can give, particularly for cloud service providers:

- ***Cost reduction:*** SDN does not require a substantial financial investment. A few SDN items are also accessible for free. While some SDN solutions, such as VMware's NSX, need a license fee, others, like Microsoft's Hyper-V Network Virtualization, are included as part of the operating system. SDN also eliminates the requirement for costly networking hardware by supporting Layer 1 through Layer 3 networking topologies.

- ***Intelligent global connections*:** SDN can assist load-balance cloud and datacenter infrastructures while also creating very intelligent and globally connected environments. Global traffic is already managed by SDN, which routes it to the most appropriate data centers based on network logic. In the future, SDN will allow architects to create even more fluid automation for data centre traffic flow. This endeavour will aid in reducing downtime, improving data resiliency, and improving disaster recovery planning.
- ***Granular security:*** Network management has become more complicated as a result of virtualization, and it's becoming increasingly difficult to enforce firewall and content filtering settings consistently. When you factor in difficulties like safeguarding BYOD devices, the security issue becomes even worse. The SDN Controller acts as a centralised point of control for distributing security and policy information across the company in a consistent manner.
- ***Reduced downtime:*** As SDN helps to virtualize most physical networking equipment, upgrading just one device rather than several is easier. You may also take a snapshot of the configuration with SDN, allowing you to quickly recover from any upgrade-related issues.

Network abstraction and the agility it enables for network administration and automation are two of the benefits of software-defined networks, which vary depending on the network. Software Defined Networks (SDN) may not be the greatest answer for all networks, but when the benefits are obvious, SDN may be the best option for keeping your organization running smoothly.

C. SDN in cloud services

- ***Microsoft Azure and Software Defined Networking***

Microsoft Azure [15] is the company's cloud platform, which consists of an ever-expanding set of integrated services such as computation, storage, data, networking, and apps that enable you to move quicker, do more, and save money.

Designing, developing, and operating global-scale datacenter networks for services like Microsoft Azure are all part of Microsoft's Software Defined Networking (SDN) strategy. Every day, tens of thousands of network modifications are made in Microsoft Azure global datacenters, which is only possible owing to SDN.

Microsoft Azure is built on the same Windows Server and Hyper-V platforms as Windows Server. Windows Server and System Center contain enhancements and best practises based on Microsoft's expertise running large-scale datacenter networks like Microsoft Azure, allowing you to use the same technologies for flexibility, automation, and control when leveraging SDN technologies.

- ***Google Cloud Platform and Software Defined Networking***

Google purchased Orbitera and announced plans to purchase Apigee. The acquisitions are clearly intended to strengthen the GCP (Google Cloud Platform) [16], but how Google's Andromeda project justifies the purchases is less clear.

Google's SDN-based network virtualization technology, Andromeda, enhances the speed of GCP, setting the framework for the Orbitera and Apigee purchases to be important. Apigee is a cloud services and API administration platform, while Orbitera is a cloud commerce platform.

Customers may assign and control their own slices of compute and storage resources in the cloud thanks to Google's use of SDN. Clients can accomplish this on their own virtual networks, and GCP can install network services like load balancing and security according to the needs of their customers at the time. GCP employs software-defined networking (SDN) to optimize cloud interconnections, as part of a trend away from vendor lock-in and toward unparalleled cloud options.

VI. THE EVOLUTION OF SDN ARCHITECTURE: CHALLENGES AND OPPORTUNITIES [17]

Both centralized and distributed controller models are supported by SDN. Every model has its own infrastructure.

Consider the following elements and requirements. Each SDN model is described in this section, along with a review of its benefits and downsides. Finally, hybrid SDN models are introduced, which incorporate the advantages of both techniques.

A. The Benefits and Drawbacks of Centralized SDN Model

A single centralized controller oversees and supervises the whole network in the Centralized SDN paradigm [18]. The Open Networking Foundation endorses this concept (ONF). Inside a single decision point, network intelligence and states are logically concentrated. OpenFlow is the official standard for making global management and control operations by the centralized controller. Because the whole network is controlled by a single centralised controller, the load on each switch along the routing path must be seen globally. It also has to keep track of which flows inside each router create bottlenecks on certain SDN node-to-node links. The controller also communicates with OpenFlow switches to gather data, failures, and faults from each network device, which it then sends to the management plane.

The centralized control plane provides a single point of management and better control over network state consistency, but it has a number of disadvantages. The controller, unlike traditional routers, must update OpenFlow switches more often. As a result, because all ports must be examined linearly, topology discovery generates additional overload, increasing response time and potentially increasing overload.

Second, the simplicity of the centralized paradigm may come at the price of control plane scalability. To put it another way, integrating all of the features on a single node demands more compute power, data storage, and throughput to handle the traffic, which results in a longer response time.

Third, in the centralized architecture, every new flow must have its first packet transmitted to a centralized SDN controller for examination.The controller hop-by-hop predicts the flow's future path and programming flow entries into each switch along the path, including both aggregation and core switches. As a result, when a new flow needs to be programmed, the controller must contact all of the switches in the path, which is a scalability issue for large networks and could lead to an explosion in the number of forwarding states in the flow tables if fine-grained flow matching is required.

Finally, SDN networks are becoming increasingly sophisticated and varied since they are meant to support different communication services and provide diverse features such as security enforcement, firewall, network virtualization, and load balancing. These services must coordinate their activities in the control plane to fulfil complicated control objectives and maintain a global picture of the whole network. It is, however, challenging to closely coordinate control operations and preserve network state consistency across distributed activities.

B. The Benefits and Drawbacks of Distributed SDN Model

The Distributed SDN [20] architecture attempts to remove single points of failure while also allowing for scale-up by spreading the load over numerous controllers. In data centers, where controller instances share a vast quantity of data to ensure fine-grained, network-wide consistency, distributed SDN control planes were built to be more responsive to local network events. The distributed SDN architecture, in particular, can easily adapt to the needs of users and applications for multi-domain SDNs using a wide range of network technologies, from high-capacity optical fibre to bandwidth-limited wireless links. Furthermore, a distributed controller is more responsive, resilient, and capable of responding to global events faster and more efficiently.

Policies may be controlled and sophisticated services can be added locally on each computer node, as there is already a local footprint available, which is one of the benefits of the Distributed SDN Model. Significantly improved scalability now that the control plane is spread entirely. Significantly improved latency when handling PACKET_IN in a reactive manner. By design, it is highly available and has no single point of failure. Smart NIC features are easier to incorporate at the local host level.

One downside of the Distributed SDN Model is that synchronizing the virtual network topology becomes increasingly challenging as the number of compute nodes grows. We'll need a centralized controller to connect heterogeneous forwarding elements (for example, older switches) to the distributed control plane (which can make management more difficult).

C. The Benefits of Transitioning to a Hybrid SDN Control Architecture

Hybrid SDN [19] architectures are being considered to solve the limitations of each of the techniques discussed above. However, determining the amount of network abstraction modules that can be centralized and efficiently structured to support conceptually centralized control activities while providing physically scattered protocols is a significant difficulty. As a result, a hybrid control plane is necessary to provide such coordination and reap the benefits of both centralized and distributed architectures. The benefits of the centralized model's straightforward control of controlling specific data are combined with the scalability and robustness of the distributed model in the hybrid SDN model.

To organize communication across SDN controllers, several critical components are required. To operate and interact with the control planes in distributed systems, these orchestrators will need common interfaces, methods, and policies, as well as high-availability and fault-tolerance capabilities.

The hybrid SDN paradigm could help determine which states belong in distributed protocols, which states need to stay local in switches, and which states should be centralized. Because each component of the network may be fine-tuned and automated at the application level, it could improve network performance by enabling optimal resource usage. A hybrid SDN approach could also include management policies to address state synchronization, security concerns, and network optimization in the event of control plane overload. Hybrid SDN deployment approaches may also enable migration that is genuinely non-disruptive. It permits existing infrastructure to be upgraded without having to overhaul the overall system.

VII. CONCLUSION

It's only natural that SDN finds a role in these new, complicated contexts as networks become more diversified and new workloads migrate to the edge. Even in the consumer world, SDN elements can be found in items. The processing of increasingly complex datasets required to identify topological connection and routing has imposed quite different requirements on the data plane forwarding function than those imposed by the growth of data rates and the resulting decrease of forwarding decision time. At the same time, all network service providers are faced with the necessity for visibility and management of significantly larger and more complicated communications topologies. SDN has demonstrated how to take advantage of these advances in network and computation capacities to provide a new approach to manage modern networks.

We started off the discussion with the complexities posed by the traditional networks and discussed how SDN was designed to address the limitations of existing network architectures.

SDN and OpenFlow are now treated as a "feature" added to a lengthy list of other existing features in LAN Ethernet switches. We explored a variety of networking technologies to demonstrate the approach's range of potential applications.

In terms of network monitoring, verification, automation, and threat detection, we explored the issues/challenges as well as the solutions for SDN. Because SDN is a new networking method, it has been used to examine various solutions to classic network problems, and many problems remain problematic. We attempted to simplify and explain each SDN issue in this document, as well as present an overall view. It's crucial to remember that the landscape of SDN-related challenges shifts as the technology evolves.

SDN's impact on Hybrid Cloud Fog Computing and Edge Computing was discovered when we went deeper into the Cloud Computing domain. In addition to making today's networks easier to manage, SDN enables cloud service providers to host millions of virtual networks without the need for traditional isolation methods like VLANs.

A number of businesses are working on components and standards to support the adoption of Software Defined Cloud Computing, or SDCC. Large-scale service providers (such as Google, Microsoft, and Amazon) stand to gain a lot from this.

We haven't attempted to provide a complete list of possibilities. We believe that there are more challenges and research opportunities in a variety of areas, ranging from formal modelling and model verification using SDN technologies in convergent packet and circuit switching networks to improve the reliability of SDN-based systems.

Finally, to conclude the SDN's decoupling approach should facilitate new applications on top of the control plane that promise to deliver richer functionality, more visibility, and better automation for network management. It remains to be seen whether SDN, in its current form, will lead to the deployment of more efficient and reliable communications infrastructures, although early development looks to be quite rapid and promising.

REFERENCES

[1]. OpenFlow , Retrieved from: https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/ https://www.sdxcentral.com/networking/sdn/definitions/inside-sdn-architecture/ https://www.ijrte.org/wpcontent/uploads/papers/v8i4/D6814118419.pdf https://www.researchgate.net/publication/267339360_SoftwareDefined_Networking_Challenges_and_research_opportunities_for_Future_Internet
[2]. Control and Data Planes , Retrieved from: https://www.mdpi.com/2673-8732/1/1/4/htm https://www.ijrte.org/wpcontent/uploads/papers/v8i4/D6814118419.pdf
[3]. System Design , Retrieved from: https://www.ijrte.org/wpcontent/uploads/papers/v8i4/D6814118419.pdf

[4]. Open Networking Foundation (ONF) , Retrieved from: https://opennetworking.org/sdn-definition/
[5]. The SDN Stacks , Retrieved from: https://www.sdxcentral.com/networking/sdn/definitions/inside-sdn-architecture/ https://docs.microsoft.com/enus/azurestack/hci/concepts/software-defined-networking
[6]. Edge Computing, IoT and Remote Access , Retrieved from: https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html
[7]. Intent Based Networking , Retrieved from: https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html
[8]. Threat Mitigation , Retrieved from: https://www.ijrte.org/wpcontent/uploads/papers/v8i4/D6814118419.pdf
[9]. Network Security , Retrieved from: https://www.garlandtechnology.com/blog/how-will-sdn impact-the-future-of-network-security https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html https://www.ijrte.org/wpcontent/uploads/papers/v8i4/D6814118419.pdf
[10]. FlowChecker , Retrieved from: https://www.researchgate.net/publication/247928775_FlowChecker_Configuration_analysis_and_verification_of_federated_OpenFlow_infrastructures
[11]. Hybrid Cloud Fog Computing , Retrieved from: https://www.mdpi.com/2673-8732/1/1/4/htm
[12]. Fog Computing , Retrieved from: https://encyclopedia.pub/10230 https://www.mdpi.com/2673-8732/1/1/4/htm
[13]. Edge Computing , Retrieved from: https://www.cisco.com/c/en/us/solutions/serviceprovider/edge-computing.html#~build-your-edge
[14]. Cloud Service Provider , Retrieved from: https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-providers https://rickscloud.com/the-benefits-of-software-defined-network-for-cloud-computing/
[15]. Microsoft Azure , Retrieved from https://docs.microsoft.com/en-us/windows server/networking/sdn/azure_and_sdn
[16]. GCP (Google Cloud Platform) Retrieved from: https://www.networkworld.com/article/3152415/sdn-is-the-unsung-hero-of-the-cloud-services evolution.html
[17]. The Evolution of SDN Architecture: Challenges and Opportunities, Retrieved from: https://www.researchgate.net/publication/267339360_SoftwareDefined_Networking_Challenges_and_research_opportunities_for_Future_Internet
[18]. Centralized SDN paradigm, Retrieved from: http://blog.gampel.net/2015/08/centralized-vs-distributed-sdn-control.html
[19]. Hybrid SDN , Retrieved from: https://www.researchgate.net/publication/267339360_SoftwareDefined_Networking_Challenges_and_research_opportunities_for_Future_Internet
[20]. Distributed SDN , Retrieved from: https://www.researchgate.net/publication/321764250_Distributed_SDN_Control_Survey_Taxonomy_and_Challenges