A Comprehensive Introduction to SDN Architectural Foundations and Applications

Dr. Yogita Hande[1], Dr. Rupali Vairagade[2]

[1]Department Computer Engineering and Technology, Dr Vishwanath Karad MIT World Peace University, Pune, India
Email: *yhande18[at]gmail.com*

[2]Department of Information Technology, G H Raisoni College of Engineering, Nagpur, India
Email: *makde.rupali[at]gmail.com*

Abstract: ***It is Software defined networking (SDN) is a new network approach which allows separation of the forwarding plane from the control plane in network architecture, that aids in the efficient control of data flow and gives network administrators a software-based method for managing and configuring the networks. In contrast to SDN, which provides a centralized controller, traditional network topologies require the individual configuration of each network device. Due to a centralized control functionality, SDN is increasingly used by businesses, telecom service providers, cloud providers, and many other users, as it encourages flexibility and scalability in the network while also assisting in streamlining operations. There is significant demand of SDN for effective and adaptable networks. SDN adoption has also been rising in the campuses and corporate sectors. This paper provides a concise overview of the Software Defined Networks (SDN) topic, which is essential for comprehending the fundamental principles of SDN in order to conduct research and implement SDN in many sectors. The initial focus is on the discussion of traditional networks, including their limitations, as well as the many issues encountered by industries. The topic of Software Defined Networks is explored, encompassing their necessity, historical background, architectural structure, interfaces, and models. Next, the many applications of Software-Defined Networking (SDN) are examined. Additionally, the various SDN case studies, vendors, setup issues with SDN, and current research fields were elucidated. The discussion focuses on the role of Software-Defined Networking (SDN) in Artificial Intelligence (AI), specifically addressing the issues associated with implementing SDN in AI.***

Keywords: Traditional network, Network Devices, SDN, SDN Architecture, SDN Interfaces, ONF

1. Introduction

Over the past decades, computer systems and the Internet have raised enormous security issues because of the wide use of networks. Moreover, the internet lacks the ability to put up the recent applications demand nature as well as consistently changing needs. The Software Defined Network (SDN) has been proposed as framework that allowed network services implementation and unparalleled as well as scalability compliance in the configuration. SDN's origin is outlined to research collaboration between University of California and Stanford University at Berkley that ultimately yielded the OpenFlow protocol in 2008. OpenFlow is the first key component of SDN. Open Network Foundation (ONF) defined this SDN in 2017 to overcome the challenges in traditional networking system. The main aim of SDN is to make the network programmable. It can be attained by the software applications that run on network operating system (NOS) top.

SDN mainly focused on open interfaces between the devices in data plane and those in control plane (controllers), separation of control plane from the data plane, programmability of network by external applications and view as well as centralized controller of the network. It divides the data and control plane and supports the network control's logical centralization. It has the capability to generate and propose new abstractions in networking, to enable network evolution and to make simple the network management. SDN divides the data and control plane to break the vertical integration. The control plane or the controller decides to handle the traffic in network and the data plane performs the traffic forwarding operation based on the decision taken by control plane. SDN found their commercial application in network virtualization techniques and cloud computing. The SDN concept is proposed with assisting the configuration and management in the network requirement. This chapter presented a detailed study of traditional networks and software defined networks.

2. Traditional Network

Over The traditional network is a digital telecommunication system, also known as a conventional or computer network, which allows networking devices to communicate with one another to exchange or share data. The data can be transmitted through cables or wireless signals. The networking devices is nothing but nodes which generate the data, route the data, and terminate the data. The node is the destination for the data and includes hosts such as servers, phones, and personal computers as well as hardware used for networking support like routers. The traditional network devices are dedicated devices which are made up of three planes such as Data Plane, Management Plane and Control Plane as illustrated in Figure 1.

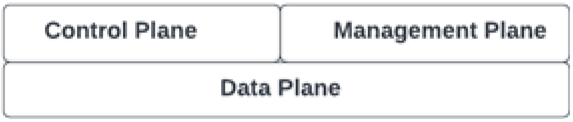

Figure 1: Traditional Network Device

A traditional network manages routing paths, and data flow using special algorithms which are implemented on dedicated hardware devices such as routers and switches. This is done through Application Specific Integrated

Circuits (ASIC). ASICs are specifically invented for certain types of network operations such as packet forwarding, where data packets can be processed by either an inexpensive routing device or more expensive devices based on different packet types or contents. The existing traditional network devices also have limitations when there is a large amount of network traffic, which reduces the performance of the network. Increasing demands for reliability, security, network speed, and scalability, can severely impede the performance of today's traditional networking devices. As a result of the underlying hardwired implementation of routing rules and current network devices are not flexible enough to handle different packet types and contents.

As an alternative to hardware embedded data handling rules, network administrators can implement them as software modules. This has the potential to significantly improve performance when it comes to resource efficient use of bandwidth and speed. Such an approach is defined in software defined networking (SDN). Unlike in traditional data networks, SDN isolates data handling from hardware, and its control is implemented in software. SDN separates data handling from hardware implementation and also implements it in a software module the controller. Consequently, network performance is improved in terms of data handling, resource management, and control. As a result, SDN is gaining popularity with cloud computing applications as a solution for conventional networks. It can also be used to implement workload optimized systems.

2.1 Limitations of Traditional Network

Traditional networks remain the most prevalent network types in today's world. Due to the rapid growth of huge data traffic, these networks have some limitations day-to-day. A network of the future must be able to scale to handle increased workloads with extra agility, while also retaining minimal expenses. The limitations of conventional networking architectures must be overcome for modern IT needs to be met. However, the conventional approach has substantial limitations.

Network Complexity - The complexity of the network has increased because of the many protocols, all with their own features that are only for specific use cases. Old technologies were often recycled as quick fixes for new necessities, or features weren't standardized across network devices and instead operated in proprietary commands.

Inconsistent policies - Since hundreds of network devices must be manually configured, changing network security and QoS policies can be challenging. This process is complicated and prone to error. Whenever an application is removed, all the associated policies remain on the devices and become more complex.

Inability to scale- Over the course of a day, application workloads change and there is constantly high demand for more bandwidth. Information Technology (IT) departments either need to grow with the organization or be satisfied with a static network that is under-provisioned. As a result, most conventional networks are statically configured, so adding more endpoints, services, or channel bandwidth requires more redesign and planning.

2.2 Traditional Network Challenges

The traditional networks were built with hardware-based platforms that were only optimized for precise features. These boxes consist of routers, ethernet switches, wireless controllers, server load balancers and community safety appliances, such as firewalls and intrusion-detection structures. Network hardware typically runs complicated, distributed control software.

Network Provisioning- Conventional network provisioning procedures can't keep up with the complexities of distributed applications. Network automation makes it possible to rapidly deploy the right resources to where they're needed, all while shifting resources as demand changes.

Configuration and Network Change management - To keep up with changes in network applications, computer, storage and device locations, network professionals spend a lot of time and resources adapting the physical and virtual networks.

3. SDN (Software Defined Networks)

SDN is an emerging area that promises to convert the way of designing, building, and operating the network. Based on simple, open as well as programmable network structure, the users is shifting from the conventional network to the SDN. SDN has the ability to reduce complexity, moreover it is cost-effective, manageable, adaptable and dynamic network architecture. SDN eliminates the distributed and static nature of the traditional network [1]. Figure 2 depicts the structure of SDN devices.

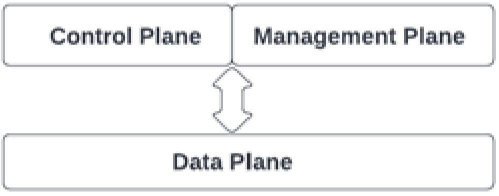

Figure 2: SDN Network Device

The SDN Networks decoupled the data plane (forwarding plane) and control plane to manage different types of data traffic more effectively and innovatively. Due to this separation SDN provides centralized and programmable network which allows network administrators to control the data traffic flows also modify network routing devices characteristics from the central place using software modules called controller, rather than dealing with each device on an individual basis. In SDN Network, administrators may change routing path information in routing tables as per the needs in network routing devices. Additionally, they have the ability to assign different priorities to specific data packets and perform different actions such as allow or block data traffic flows.

In solutions such as multitenant architecture, cloud computing, etc., SDN can be applied because network administrators can better manage network traffic and utilize network resources. By comparison to conventional network devices, SDN is more cost-effective and provides greater control of network traffic flow. The SDN implementation is based on several standards. The OpenFlow standard is a widely accepted and widely used standard for implementing SDN. In SDN the routing devices is managed by controlling the network routing information through OpenFlow standard.

3.1 SDN History

The development of programmable computer networks is the result of innovation in network management. A SDN's history can be divided into three stages. A major innovation of active networks was the introduction of programmable functions in the mid-1990s and early 2000s. Developed open interfaces between the control and data planes from 2001 to 2007. From 2007 to 2010, OpenFlow APIs and network operating systems enabled a scalable and practical separation of control and data. Virtualization has played a significant role in SDN's history.

Active Networks - Back in the 1990s, there were many issues with internet traffic. Preferred treatment and routing led to increases in route caching, as well as traffic engineering issues. Network gear at the time just didn't offer enough capability to manage this. To help control data passing through passive networks, researchers came up with network level API that allowed a level of programmability. This work most notably included Tenerhouse and Wetherhall's introduction of an "Active Network." In active networks, custom programs are injected into network nodes to manipulate data flow. This idea was radical at the time but has gone on to prove its validity in research such as GENI, FIND, and FIREControl and Data Planes Separation - During the early 2000s, network operators introduced traffic engineering (the control of traffic forwarding paths) to manage all the additional traffic. Although primitive, this approach exacerbated frustration, increasing the need for an alternative. Several proposals were made, such as having an open interface between data planes and control planes, but only a few had an impact on the industry in any significant way. Routing Control Platform, Forwarding and Control Element Separation, and SoftRouter are examples of these. Enterprise networks are protected against threats by Ethane's flow-based policy enforcement mechanism. By using flow-based networking and a central domain controller that secures bindings and enforces flow, access control, and policy, the research project aims to enforce enterprise policy.

OpenFlow and Network OS - In the mid-2000s, Broadcom released an open API for programmers to control certain functions in the network. Network operators, equipment vendors and networking researchers were constantly pushing for network programmability and disaggregation. The genesis of OpenFlow can be attributed to these laborious works. Then, in 2008 a group at Stanford University began working on the initial concept of OpenFlow. In December 2009 they released version 1.0 of the specification. Since its inception, OpenFlow has been managed by four entities: ONF (Open Networking Foundation), Plugfest, Metrolinx and AARNet. These companies have continued developing the software in various projects since it first began being developed at Stanford in 2008.

3.2 Traditional Network Vs SDN

Data transmission in a network relies on network planes such as data plane and Control plane. Once the device data is available for routing, the control plane determines which path is optimal for transmitting the data packets, and then the data plane forwards data packets based on the path specified by the control plane. Devices in a traditional network include a control plane and a data plane to provide network control that is decentralized. SDNs are characterized by separate control planes and data planes, with the forwarding elements such as switches at data plane being controlled by a centralized controller of control plane. SDN enables network configuration and management from a centralized controller through programming and provides a global view of centralized network. A comparison of two networks is illustrated in Figure 3.

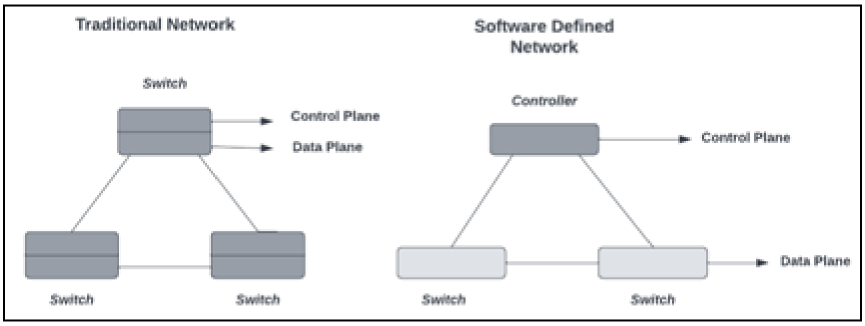

Figure 3: Traditional Network and SDN Network

Table 1: Traditional Network Vs SDN

Key Points	Traditional Networking	Software Defined Networking
Network Plane (Control and Data)	Coupled	Decoupled
Network Control	Distributed	Centralized
Global View of Network	Difficult	Controller Provides global View
Network configuration and update time	Takes time	Quickly
Network Maintenance Cost	More	Less
Resources Utilization	Less	More

5. SDN Architecture

SDN framework is commercial, adaptable, dynamic, and controllable. It decouples the data plane that performs forwarding functions and the control plane that is the network intelligence. There are three layers in SDN such as the infrastructure layer (Data plane), the Control layer (control plane), and the Application layer (application plane) [2]. SDN architecture Figure 4.

Infrastructure Layer/Data Plane - SDN has a bottom layer called infrastructure. It is composed of physical forwarding elements / network elements (NEs) like virtual switches, routers, wireless access point (AP), optical switches and Ethernet switches and it forms a data plane. An individual network is formed by interconnecting all these physical NEs. Through various transmission media like wireless radio, optical fiber and copper wires, the switching devices are interconnected. The data plane forwarding elements are the dummy devices and their role is to forward the data as per the forwarding rules set by controller.

Control Layer/Control Plane - Control Layer is the middle layer which consists of programmable and centralized controllers that are accountable to deploy the paths and flows in the network. This centralized controller provides a global view of the network. Two types of entities are handled by this SDN controller, namely one that manages network controlling and one that network monitoring. The network controlling composed of packet forwarding rules for infrastructure layer and the policies imposed by application layer.

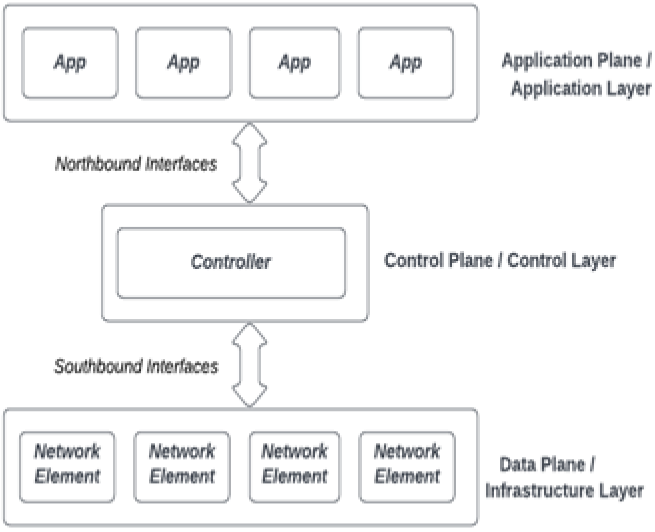

Figure 4: SDN Architecture

Infrastructure Layer/Data Plane - SDN has a bottom layer called infrastructure. It is composed of physical forwarding elements / network elements (NEs) like virtual switches, routers, wireless access point (AP), optical switches and Ethernet switches and it forms a data plane. An individual network is formed by interconnecting all these physical NEs. Through various transmission media like wireless radio, optical fiber and copper wires, the switching devices are interconnected. The data plane forwarding elements are the dummy devices and their role is to forward the data as per the forwarding rules set by controller.

Control Layer/Control Plane - Control Layer is the middle layer which consists of programmable and centralized controllers that are accountable to deploy the paths and flows in the network. This centralized controller provides a global view of network. Two types of entities are handled by this SDN controller, namely one that manages network controlling and one that network monitoring. The network controlling composed of packet forwarding rules for infrastructure layer and the policies imposed by application layer.

Application Layer/Application Plane - In SDN, the application layer is located over the control layer. Security, access control, energy efficient networking, load management and bandwidth management are some of the applications presents in the application layer and their network needs are transferred to control plane via Northbound Interface (NBI) API.

Using a northbound and southbound API, this SDN application abstracts global network status information. The control layer is connected to the infrastructure layer via an Application Programming Interface (API) called southbound API. Some of the southbound API protocols such as OpenFlow, NetFlow and sFlow are used for analyzing and monitoring the traffic in the interwork. Northbound API is the controller application interface. This API is utilized for the applications development for load balancing, network management, etc. For northbound communication, there is no commonly accepted standard protocol. For a southbound API, OpenFlow is the standard protocol that is commonly accepted. OpenFlow permits the implementation of SDN in both software as well as hardware. A significant part of SDN is the controller, which is otherwise known as a network system. Recently, POX, FloodLight, Open Daylight, etc. are some of the most popular controllers utilized for SDN.

5.1 SDN Characteristics

- Agility - Network administrators can modify the configuration of the network as business and application requirements change.
- Centralized management - For operating a network, SDN centralizes network intelligence. Also, the SDN provides an all-inclusive network view.
- Programmable Network - Through automated SDN services, it is possible to program network features and configure network resources directly from central place.
- Open connectivity - Open standards are used to implement SDN. As a result of its vendor neutral architecture, SDN simplifies network design and ensures consistency.

5.2 Interfaces for SDN

SDN structure includes the following interfaces.

- Southbound Interface - The Southbound Interface (SBI) is an application programming interface (API) exposed to lower layers. The controller plane communicates with the data plane through a southbound interface. In most SDN implementations, OpenFlow and the southbound APIs are used to configure the network. The defined protocols

Paper ID: SR24129155630 DOI: https://dx.doi.org/10.21275/SR24129155630

enable the controller to push policies to the forwarding plane.

- East-West Interface - The east-west interface allows communication between the multiple controllers. It uses distributed routing protocols such as BGP and OSPF.
- Northbound Interface - Network operators can easily customize the network controls using the northbound interface (NBI) through APIs. The applications running on the application layer and controller communicate with the help of NBI. The well-known northbound API is REST.

5.3 SDN Models

- Open SDN - OpenFlow is a protocol that network administrators can use to control physical and virtual switches at the data plane level. Data packets are routed using the OpenFlow protocol in both virtual and physical devices.
- SDN by APIs - The flow of data from each device is controlled through APIs, or application programming interfaces. Data movement through a network is determined by application programming interfaces.
- SDN Overlay Model - Software-defined networking overlays an existing hardware infrastructure with a virtual network. Assigning devices to each channel of a virtual network, which is created atop existing hardware, allows bandwidth to be allocated across multiple channels.
- Hybrid SDN - Using hybrid SDN, you can support different network functions by combining software defined networking with traditional networking protocols.

5.4 ONF (Open Networking Foundation)

OpenFlow is an open-source software infrastructure for data center networks originally proposed in 2008 by its inventors, Nicira, now part of VMware. A group of telecommunications service providers formed the Open Networking Foundation (ONF) in 2011 to standardize, commercialize, and promote OpenFlow. As part of these efforts, the ONF hosts an annual conference called the Open Networking Summit.

One of the most interesting aspects of ONF is that its board is made up entirely of major network operators. This ensures that ONF only provides services that are determined to be widely used and mutually beneficial, not just what serves the interests of one networking vendor over another. Along with this, ONF's board has an emphasis on real-world business experience and not just theory, which helps it guide ONF in a way that really works.

5.5 Need of SDN

Increased Development Cost - Currently, autonomous networking devices have complex control plane software to store, manage, and run. These demands of networking devices increase the cost per unit resulting from the processing power and storage space needed to execute the sophisticated software.

Encouragement for Vendor Lock-in than closed environment - Most relevant protocols that are used by switches and routers are constantly being developed as standards in the networking space. Despite enhancements made by one vendor over another, vendors attempt to implement these standards so that networks of devices from different vendors can coexist without difficulties. By adhering to these standards, misconceptions about interoperability between different vender's products are eliminated and customers can switch vendors without worries of future difficulties. This results in competition cost, which benefits the customer with continually dropping costs and high-quality products.

Need to change Complexity and Resistance- It's usually a good idea to leave networking alone once it is established, unless the system breaks and needs restarting. And even then, there is discussion about whether swapping to an open, efficient, and less costly networking solution would improve things. If networking were simpler and more progressive, this would be ideal.

Increased Cost of Network Operations - SDN has the potential to lower operational costs for companies that provide their own equipment for use in a network environment. The operation expenses of such companies may be higher than the respective ones of those that are buying from external vendors. This SDN can provide acceleration to automate actions on the networks and provisioning new services or switching between them could become more agile and versatile under this potential change.

5.6 Challenges in SDN

SDN networks offer many benefits to IT enterprises and cloud providers, though they deal with a variety of challenges [3]. Some of the SDN challenges are discussed below:

Network Reliability - The SDN controller intelligently configures and validates network topologies to maximize network availability and prevent manual errors. SDN controllers experience single failures due to brain split issues, which lead to intelligence being reserved. If one or more devices fail, flow continuity can be observed by routing traffic via nearby nodes. The whole network is controlled by only one controller, so if the controller fails, the whole network collapses.

Scalability of controller - If the APIs linked, both the planes in SDN are “evolved independently” and it rushes various alteration in the control plane. But the scalability constraints raisedis the main drawback of this decoupling. Hence, the constraints on network scalability occurred due to the Flow-setup process of the controller overload.

Latency constrained performance - The performance of SDN is estimated according to two metrics namely the number of flows per second and the flow-setup time, because it is a flow-based approach. The reactive as well as proactive modes are the two types of flow setup. In the proactive mode, the flow setup occurred earlier than the packet arrival at the switch. Thus, the switch already knows to deal with the packets when it arrives. In the reactive mode, the negligible delay is there and the constraints on number of flows per second are eliminated and this mode is managed by the controller.

Data Path Controlling between CPU and ASIC - The data path controlling among CPU and ASICs not a conventional task. In each flow-table entry, the flow duration, number of packet bytes and the number of matches is the three counters, and these matches are specified by OpenFlow. The function of counters modification is a complex task since it is implemented in ASIC hardware. ASIC re-designing or developing new switch hardware is needed to develop the SDN protocol. Due to the local counter transfer from the ASIC to controller, the SDN performance is limited.

Low-Level Interface Utilization between the Network Device and the Controller - The control applications are developed with simple interfaces for the determination of high-level network policies to simplify the network management using SDN. Moreover, SDN architecture is essential to convert these policies into low-level switch configurations. A programming interface is provided by the controllers that assist the event driven, imperative and low-level model. The switch-by-switch, rule by-rule and low-level packet-processing rules are installed and uninstalled individually by the interface for the network events like link status updates as well as packet arrivals. In such cases, it is necessary to constantly by the programmers that whether there is any critical for other controller monitored future events by the un-installing switch policies. Further, even the switches, the simple tasks must be performed by managing multiple asynchronous events, this will increase the complexity.

Controller Placement Issues - From the flow-setup latencies, every aspect of decoupled control plane is influenced by the controller placement issue for the fault tolerance of network reliability and performance metrics. For instance, the availability and the convergence time are restricted by the long propagation delay wide area networks (WANs). For software design, it impacts whether events must be moved earlier or whether controllers can react to events in real time. This issue consists of the requirement of number of controllers and the deployment of controller for the accessible network topology.

Security - The security risks in SDN arise from the inability to push around each packet and the lack of assimilation with conventional security systems. Moreover, the controller vulnerability to attack surfaces as well as hackers is maximized by the controller software intelligence improvement. The network's each characteristic is damaged while the controller is accessed by the hackers. Further, the network administrator authorization and authentication classes are assisted by maximizing the security in SDN.

Management and Convergence - The initial intention for developing the SDN OpenFlow is to analyze the protocol of enterprise campuses networks and designed SDN for the private network or the small network. Though, there is some issues like routing between two networks or the interdomain routing, etc., in the extension of small network to the large network.

Flexibility of controller - The prominent component to the SDN network is the centralized controller. SDN provides access to program the network on the centralized controller by the network administrator using the software. However, the flexibility of the entire network gets damaged due to the malfunction of the controller. Thus, the way to control the failure in controller needs to be defined by the SDN network.

6. SDN Applications

In SDN, several applications [4] are utilized the benefits of SDN. Figure 5 represents the SDN applications, and these applications are seen detailed as follows:

Internet Research - Several challenges arise when updating the internet as it frequently being utilized. A new approach or idea testing is a complex to solve the existing networks problem. Without changing network, the SDN can provide ideas for the future internet. It is simple to divide the hardware from the software in SDN, even the separation of data and control traffic with OpenFlow switch in SDN. A new Internet framework strategy is tested by allowing this separation for experimenting with a novel addressing schemes. OpenFlow permits several firms' switches, access point and routers for using the data and control plane separation.

Rural connections - The complex enterprise networks and data centers are simplified by the SDN. A rural Wi-Fi networks simplification also this SDN is utilized. In rural environments, small profit margins, resource constraints, sparse population, etc. are considered as some of the primary issues. The separation of construction and configuration of the network is essential in rural environments. In this context, SDN is important and places the management and control functions in the central controller. The Internet Service Provider (ISP) business as well as the rural infrastructure deployment business is enabled by this separation to operate these two businesses in different entities. Thus, the rural network management made by SDN is more flexible than other conventional network frameworks (in this framework, the customized control is required for local network devices). Therefore, in rural areas the control of rural devices is performed.

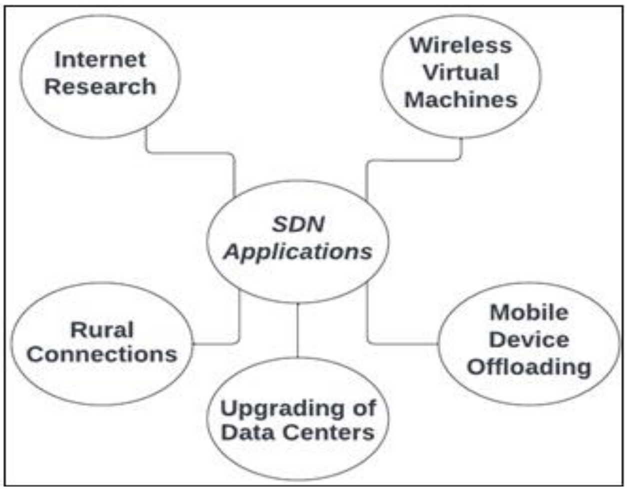

Figure 5: SDN Applications

Upgrading of Data Centers - Several companies rely heavily on data centers for their operations. As an example, consider Google, which has more data centers. Therefore, the data provided for the request is performed efficiently as well as quickly. In a similar way, various other firms also used the data center for providing the data quickly to the user. But the main drawback is that these data centers are more expensive in terms of maintenance.

To address this issue, OpenFlow is utilized, which permits the organizations in setting up as well as network configuration to save money. This OpenFlow allowed the switches for controlling from the central location. Using SDN, the data centers are connected through OpenFlow based network infrastructure service. The issue of workload with small latency is solved by these interconnected data centers by transferring the workload to an underutilized network. The workload is completed quickly at various time zones in SDN, if the network is busy at a particular period. Therefore, this method is considered as energy efficient [5].

The performance, bandwidth and throughput of data centers are tested by creating a huge number of nodes in data centers, which consists of two core switches and four regular switches with OpenFlow controller [4]. In this architecture, the firewall is provided between the routers, OpenFlow controller and the core switches. Mininet is an application utilized for testing the performance of networks and to prototype their network. SDN using OpenFlow protocols are customized by the researchers using the Mininet application. CBench, PingPair, Iperf, PingAll and Ping are some of the tools used to analyse their setup of network. These tools are useful to verify the feasible connectivity, speed, and bandwidth by the user. The network traffic can also be viewed using Wireshark tool.

Mobile device offloading - For business applications, privacy is a significant one, because the data work performed by the user needs to be kept safe from the intruders. Few information is transferred among some users, whereas the other information does not need the similar level of security. In [5], an Enterprise Centric Offloading System (ECOS) is utilized by the authors for addressing these issues. The data are offloaded using the ECOS to idle computers, when the applications are ensured with extra security needs are offloaded only on the approved machines. For various applications and users, the performance is also considered. SDN is used for choosing the resources as well as for managing the network. The controller regulated the flows by utilizing the OpenFlow switches. ECOS has the capability for offloading, when the security needs are considered without any overly difficult approach.

Wireless virtual machines - In business, the applications running on wireless virtual machines are widely spread as common. Due to the flexibility and the less operational cost, these virtual machines (VMs) are utilized by several companies. These VMs make portable by the companies to obtain all potential from these machines. The maintenance of IP address of VM is the primary issue for the firms. A dynamic DNS or the mobile IP is utilized for solving these issues [6]. But the network settings manual reconfiguration is the complex task in this process after the removal of VMs. This problem restricts the data centers as well as business from simply transferring their VMs to the new location.These VM's mobility problems are solved by CrossRoads application. Both offline and live VMs are allowed by this application. Three primary purposes are there in this application. They are:

Purpose 1: The ability to solve traffic flow from external users and data centers.

Purpose 2: To utilize OpenFlow, we assume that each data center will use the OpenFlow controller.

Purpose 3: By using pseudo addresses for MAC and IP addresses, the addresses will remain constant while porting when IP addresses are allowed to change.

Therefore, SDN is a valuable resource in several applications that permits the user to test advanced protocols and to adapt network to a new situation. For its versatility, OpenFlow is utilized in many applications. In SDN, the hardware changes are simple, and several benefits are provided because of control and data plane separation.

7. SDN Case Studies

SDN is a networking technology that separates the control plane from the data plane, making network management more centralized and programmable. Here are several case studies demonstrating how SDN has been implemented in a variety of situations:

Google implemented SDN in its B4 network [6] to manage the infrastructure of the global backbone. This enabled Google to manage network resources dynamically and reroute traffic in response to network congestion or failures. It increased network utilization and decreased delays.

The objective of the ONF's Atrium initiative was to develop an SDN-based solution for a multisite, MultiTech ology network [7]. This case study demonstrated the viability of deploying SDN in a vast, heterogeneous network environment.

The Clean Slate project at Stanford intended to redesign the internet from scratch using SDN principles. OpenFlow is one of the earliest SDN protocols that they developed. This case study demonstrates the potential for reconsidering and enhancing networking architectures utilizing SDN [8].

Microsoft and Facebook [9] are among the companies that have implemented SDN in their data centre networks to increase resource utilization, simplify management, and enable more flexible and efficient network infrastructure scaling.

Telecom operators such as AT&T have utilized SDN to virtualize their network functions and offer services such as Network Function Virtualization (NFV), which entails operating network functions as software as opposed to dedicated hardware appliances [10].

In their campus networks, educational institutions such as Stanford University and the University of Michigan have adopted SDN. SDN enables simple administration of network policies, traffic optimization, and user group isolation [11].

SDN has been used to employ sophisticated techniques for traffic engineering and load balancing. SDN controllers can, for example, modify routing paths dynamically to optimize traffic flow and prevent network congestion [12].

SDN has been incorporated in security applications to detect and mitigate network threats more effectively. By dynamically segmenting the network, traffic anomalies can be detected and isolated, preventing the spread of threats [13].

8. SDN configurations Problems

Due to the paradigm transition from conventional networking approaches, configuring a Software-Defined Network (SDN) can be challenging [14]. Here are some typical issues that organizations may encounter when configuring SDNs, along with possible solutions [15]:

1) SDN introduces new concepts such as the separation of control and data planes, programmable interfaces, and centralized management, which contribute to the complexity of SDN concepts. The network administrator is acclimated to the traditional network structure for them it’s very hard and difficult to grasp the SDN concepts and techniques. To understand the SDN techniques need to invest in training programs.
2) Incorporating SDN functionality in traditional networking infrastructure might present a challenge. To design and implement the API to provide the SDN functionalities is quite difficult with the legacy network system.
3) Software-defined networking controllers should be selected according to the application. There are many types of SDN controllers, each with its own features and attributes. The process of selecting the most appropriate controller for a set of requirements can prove challenging. Several controllers will be evaluated as part of the proposed solution, taking into account specific needs, scalability, and compatibility.
4) SDNs expose potential security vulnerabilities due to their single point of failure. Unauthorized controller access could result in a network breach. SDN controllers and their communication channels should incorporate robust access controls, encryption, and authentication procedures to enhance security.
5) Software-Defined Networking (SDN) environments are susceptible to misconfigurations that can disrupt networks. It is possible for errors to occur during the coding or modification of SDN settings because they are software-based.
6) As the size of the network increases, it becomes imperative for Software-Defined Networks (SDNs) to exhibit scalability while maintaining optimal performance levels. Inadequately built software-defined networking (SDN) architectures can result in the occurrence of bottlenecks and diminished operational efficiency. To handle scalability needs, it is recommended to design the software-defined networking (SDN) system in a manner that can accommodate the anticipated growth. Additionally, it is important to consistently monitor the performance of the SDN system to promptly detect and resolve any potential issues that may arise.
7) The environment of Software-Defined Networking (SDN) encompasses a range of standards and protocols, hence giving rise to potential issues in achieving interoperability when integrating components from diverse suppliers.
8) The selection of a proprietary solution may result in the establishment of vendor lock-in, hence presenting difficulties in transitioning to alternative vendors or technologies in subsequent periods. One potential way to mitigate the risk of vendor lock-in is to explore open-source alternatives or solutions that comply with open standards. By adopting open-source software or systems that adhere to open standards, organizations can avoid being dependent on a particular vendor's proprietary ecosystem.
9) The implementation of a Software-Defined Networking (SDN) infrastructure typically necessitates modifications to existing operational processes and workflows. The implementation of this change may encounter opposition from IT personnel and might disrupt established procedures. The proposed solution involves effectively conveying the advantages of Software-Defined Networking (SDN) to the team and actively engaging them in the stages of planning and implementation, so ensuring a more seamless transition.
10) Conventional network monitoring solutions may not offer equivalent levels of visibility within a software-defined networking (SDN) environment. The process of troubleshooting and determining the underlying cause of problems can often include a higher level of complexity. One potential solution is to allocate resources towards acquiring specialized tools that are specifically tailored for the purpose of monitoring and debugging Software-Defined Networks (SDNs). Additionally, it is crucial to invest in training programs to equip your staff with the necessary skills and knowledge to properly utilize these.

The process of designing a Software-Defined Network (SDN) might present complexities; however, by meticulous preparation, comprehensive education, and a strategic methodology, a significant portion of these issues can be effectively addressed. An organization must carefully assess the needs of the organization, critically evaluate the range of potential solutions, and formulate a meticulously developed implementation strategy.

9. SDN Vendors

Numerous companies presented their unique solutions and technologies for Software-Defined Networking (SDN). Since then, the landscape may have changed significantly. To obtain the most accurate and current information, individuals should conduct their own research. A number of vendors [16] in the field of Software-Defined Networking (SDN) emerged during that time.

Vendor	SDN Solution/Product
Cisco Systems	- Data center networking using Cisco Application Centric Infrastructure (ACI) - Cisco Software Defined Access (SD-Access) for campus networks
VMware	- VMware's NSX platform, widely recognized SDN solution providing network virtualization and security features for data centers and cloud environments
Juniper Networks	- Contrail Networking, an SDN solution focused on automating and orchestrating virtual and physical network infrastructure
Arista Networks	- CloudVision platform, offering SDN capabilities for cloud networking and data center environments, emphasizing automation and programmability
Huawei	- CloudEngine switches and CloudFabric solutions encompassing SDN and intent-driven networking for various deployment scenarios
Extreme Networks	- ExtremeFabric and Extreme Management Center, offering SDN capabilities with a focus on simplifying network management and improving visibility
Big Switch Networks	- Solutions like Big Cloud Fabric and Big Monitoring Fabric, aiming to simplify network operations and enhance visibility and security
Cumulus Networks	- The open network operating system Cumulus Linux enables network automation, programmability, and adaptability.
ONF (Open Networking Foundation)	- Promotes open-source SDN platforms through initiatives such as ONOS (Open Network Operating System) and CORD (Central Office Re-architected as a Datacenter).
Pica8	- Open networking operating system, PICOS, allowing organizations to run their choice of switching hardware and customize their SDN deployments
Pluribus Networks	- SDN-based fabric architecture called Netvisor, emphasizing visibility, security, and network analytics
Nokia (Nuage Networks)	- Nuage Networks offers SD-WAN and data center SDN solutions, focusing on creating scalable and secure virtualized network services

A limited number of software-defined networking (SDN) enterprises are illustrated. The SDN market has a dynamic nature and is characterized by ongoing evolution. Therefore, it is advisable to undertake an exploration of the latest products, conduct feature comparisons, carefully analyze the unique requirements of one's organization, and actively engage with suppliers in order to obtain the most up-to-date information.

10. Current SDN Research Areas

Here are some possible areas of current SDN research.

Areas	**Description**
5G Integration	Researchers explore SDN integration into 5G networks for dynamic network slicing and resource allocation.
Edge and Fog Computing	SDN applied to edge and fog computing for managing networking in distributed computing environments.
Intent-Based Networking	High-level policies used to drive network configuration and management, with SDN as a key enabler.
AI and Machine Learning	Integration of AI/ML with SDN for traffic prediction, anomaly detection, and network optimization.
Security-Driven SDN	SDN leveraged for network security, including real-time threat detection and adaptive security policies.
Multi-Cloud Networking	SDN used to manage and secure network connectivity across multiple cloud environments.
Network Slicing	SDN applied to create industry-specific network slices tailored to various sectors.
Traffic Engineering	Advanced algorithms and protocols for efficient traffic engineering in SDN networks.
Interdomain SDN	Challenges and solutions for connecting SDN domains across different administrative domains.
Container Networking	SDN's role in facilitating networking for containerized applications and microservices.
Resource Utilization	SDN's potential to optimize resource allocation and energy efficiency in data centers and networks.
Standardization & Interop.	Efforts for SDN standards and interoperability to facilitate integration and deployment.

11. Artificial Intelligence in SDN

SDN (Software-Defined Networks) capabilities and effectiveness are significantly improved by AI (Artificial Intelligence). SDN is a networking architecture that separates the control plane and data plane, allowing for the centralized control and programmability of network devices. Various aspects of SDNs can be optimized using AI, including network administration, performance, and security. Here are some examples of AI integration in SDNs:

1) Optimization of Networks and Traffic Engineering - AI can analyze network traffic patterns, predicting traffic surges, and optimizing routing paths to guarantee efficient data flow. Adapting in real time to changing network conditions, machine learning algorithms can make decisions that result in minimized latency, reduced congestion, and optimal bandwidth allocation.
2) Detection of Anomalies and Security - AI can be used to identify and respond to network anomalies and security concerns. By understanding normal network behavior, AI systems are able to identify deviations that may indicate a security compromise or abnormal activity. This expedites threat mitigation and improves network security.
3) Predictive maintenance - Using artificial intelligence, predictive maintenance is able to monitor network infrastructure and predict when components are likely to fail. This enables proactive maintenance scheduling, reduces downtime, and prevents unplanned disruptions.
4) Resource Allocation and Load Balancing: AI algorithms are capable of dynamically allocating resources and distributing workloads across various network nodes. This guarantees that no single component becomes a bottleneck and that all available resources are utilized effectively.
5) Optimizing Quality of Service (QoS) - AI can analyze application requirements and network conditions to prioritize specific categories of traffic, ensuring

mission-critical applications receive the required bandwidth and latency.

6) Network Configuration and Provisioning: Artificial intelligence can help automate the process of configuring and provisioning network devices. This decreases the likelihood of human error and accelerates the rollout of new network services.
7) Capacity Planning: Artificial intelligence is able toanalyze historical data and predict future network capacity requirements, thereby assisting organizations in planning for development and allocating resources appropriately.
8) Policy Implementation - AI is capable of enforcing network policies and compliance requirements by perpetually monitoring network activity and adjusting to ensure policy adherence.
9) Variable Adaptation - AI enables networks to adapt to altering conditions, such as fluctuating workloads, hardware failures, and network topology alterations. This adaptability is crucial for maintaining optimal performance in environments that are complex and dynamic.
10) Intent-Based Networking (IBN): AI-powered IBN enables network administrators to define high-level business intent, which is then translated into network configurations and actions by the AI system. This simplifies network administration and simplifies manual configuration.

AI challenges in Software Defined NetworkingWhile AI offers numerous benefits in Software-Defined Networking (SDN), its successful integration requires addressing several challenges and complexities. Some of the primary obstacles include:

Challenge	Description
Data Quality and Availability	Obtaining comprehensive and clean network data in SDNs is difficult due to data inconsistencies.
Data Security and Privacy	Integrating AI without adequate security measures may expose sensitive data to potential breaches.
Absence of Standardization	The absence of standard data formats and APIs can complicate the integration of AI across diverse SDN deployments.
Network Dynamics	Networks undergo rapid shifts in traffic patterns, topology, and device behavior, challenging AI adaptability.
Scalability	As networks grow larger, AI algorithms may struggle to scale efficiently, leading to latency and performance issues.
Real-time Requirements	Ensuring precise predictions and decisions by AI models within strict time constraints is challenging.
Interpretability and Explainability	Models of artificial intelligence should shed light on their decision-making processes, but a lack of interpretability can impede implementation.
Integration with Existing Systems	Legacy systems and varying vendor-specific implementations can hinder AI integration into established SDN environments.
Training Data Representativeness	AI models trained on historical data may not capture the full range of network behaviors, leading to biased predictions.
Expertise Gap	Bridging the knowledge gap between AI experts and network engineers can be challenging due to differences in terminology and expertise.
Resource Limitations	SDN environments may lack the computing resources needed for resource-intensive AI algorithms.
Regulatory and Ethical Considerations	AI-driven actions in SDNs may have legal and ethical consequences, necessitating compliance with regulations and ethical standards.

To address these challenges, ongoing research and development focuses on developing AI algorithms tailored to the specific requirements of SDNs, enhancing data quality and availability, bolstering security measures, and developing standard interfaces for AI-SDN integration. As AI and SDN technologies continue to develop, it is likely that these obstacles will be resolved, allowing for more efficient and intelligent network management.

12. Conclusion

The new network approach SDN promises to reduce network configuration and management complexity and make network operation possible by separating the network device control plane and data plane. A key feature of the SDN is the programmability and automation over the network that helps industries which are using cloud services to manage resources as needed. Rapid growth in IoT, 5G, and Industry 4.0 is a key driver of the SDN market. In this chapter, an introduction to SDN is described in detail, including traditional network's structure. Also explored the limitations and challenges of traditional networks to understand SDN's importance in large growing industries. Moreover, SDN needs, challenges and applications are discussed. In order to stimulate interest in SDN research, we deliberated on the existing areas of study in SDN. Furthermore, given the current surge in the field of Artificial Intelligence, it is worth exploring the potential use of Software-Defined Networking (SDN) in this domain. This discussion will focus on the issues associated with implementing SDN in the context of Artificial Intelligence.

References

[1] Jimson E.R, Nisar K, and bin Ahmad Hijazi M.H, Bandwidth management using software defined network and comparison of the throughput performance with traditional network. International Conference on Computer and Drone Applications (IConDA); 2017 November 2017; Malaysia; IEEE; p. 71-76.

[2] Yogita H, Akkalakshmi M. A survey on Intrusion Detection System for SDN. International Journal of Intelligent Information and Database Systems. 2022; p. 157-165.

[3] Yogita Shivaji Hande, M. Akkalakshmi, A Study on Software Defined Networking, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 11, November 2015

[4] Jianfeng Jiang, SDN Technology Analysis and Application Research, 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), IEEE, November 2021, DOI 10.1109/ICNISC54316.2021.00014

[5] Santos R, Souza D, Santo W, Ribeiro A, and Moreno E. Machine Learning Algorithms to Detect DDoS Attacks in SDN. Concurrency and Computation: Practice and Experience. 2020; p. 1-14.

[6] Sushant Jain, Alok Kumar, ubhasree Mandal etal, ,B4: experience with a globally-deployed software defined wan, SIGCOMM '13: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, August 2013, Pages 3–14

[7] SDN Architecture – https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf

[8] Clean Slate Design for the Internet - https://www.technologyreview.com/2007/03/19/226188/a-fresh-start-for-the-internet/

[9] Why Facebook Does SDN - https://www.enterprisenetworkingplanet.com/data-center/why-facebook-does-sdn/

[10] Accelerating Software Defined Network Deployments https://about.att.com/innovationblog/2020/02/accelerating_sdn.html

[11] SDN in the Campus Environment - https://opennetworking.org/wp-content/uploads/2013/03/sb-enterprise-campus.pdf

[12] Ian F. Akyildiz a, Ahyoung Lee a, Pu Wang b, Min Luo c, Wu Chou c, Show moreA roadmap for traffic engineering in SDN-OpenFlow networks, Computer Networks, Volume 71, 4 October 2014, Pages 1-30

[13] Yassine Maleh, Youssef Qasmaoui, Khalid El Gholami, Yassine Sadqi&Soufyane Mounir, A comprehensive survey on SDN security: threats, mitigations, and future directions, Journal of Reliable Intelligent Environments , Volume 9, pages 201–239, (2023)

[14] Abigail O. Jefia, Segun I. Popoola and Aderemi A. Atayero , Software-Defined Networking: Current Trends, Challenges, and Future Directions, Proceedings of the International Conference on Industrial Engineering and Operations Management Washington DC, USA, September 27-29, 2018

[15] Deepak Singh Rana, Shiv ashishDhondiyal, Sushil Kumar Chamoli, Software Defined Networking (SDN) Challenges, issues and Solution, Journal of Computer Sciences and Engineering. Vol.-7, Issue-1, Jan 2019 , DOI - DOI: 10.26438/ijcse/v7i1.884889

[16] SDN controller (software-defined networking controller) - https://www.techtarget.com/searchnetworking/definition/SDN-controller-software-defined-networking-controller

[17] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, Andrew Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Computer Networks, Volume 167, 2020, 106984, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2019.106984.

[18] An Wang, Zili Zha , Yang Guo , Songqing Chen, Software Defined Networking (SDN) Enhanced Edge Computing: A Network Centric Survey,IEEE, 2019

[19] Applications of Artificial Intelligence, Machine Learning and related techniques for Computer Networking Systems - https://doi.org/10.48550/arXiv.2105.15103

[20] Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, Juan Felipe Botero Vega, Security in SDN: A comprehensive survey, Journal of Network and Computer Applications, Volume 159, 2020, 102595, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2020.102595.

[21] Paul Goransson, Chuck Black, Software Defined Networks A Comprehensive Approach, 2014 Elsevier