Software Defined Networking(SDN) and Cloud Computing

Danish Raza · Follow
12 min read · Aug 5, 2021

Software Defined Networking (SDN) can be described as an architecture that allows networks to be more flexible and programmable by segregating and decoupling the control plane and the data/infrastructure plane from one another, this allows us to program the network according to our needs [1]. SDN plays a major role in Cloud Computing in that it allows the users to address different changes quickly as they come. One of its major advantages is that it streamlines the network configuration procedure and boosts network monitoring and performance. Cloud Computing along with the cloud storage services have provided a ton of benefits when used in conjunction with virtualization and automation but the only restriction that they face is the network limitations hence SDN allows the networks to be flexible and adapt to the changing network demands [2].

Software Defined Network (SDN) Architecture

There are three main layers in the SDN Architecture:

1. **Application Layer:** The application layer consists of the SDN Applications, this consists of all the network applications that are used by the organization, which includes the firewalls, load balancers and intrusion detection system. For a traditional network a physical device might be used to manage all of this but the SDN allows us to make use of an application to oversee and manage their working.

2. **Control Layer:** The control layer consists of a SDN controller application. It manages all the policies in the network along with the flow of network traffic.

3. **Infrastructure Layer:** The infrastructure layer as the name suggests consists of all the physical equipment required by the network such as the switches[2].

Components of Software Defined Network (SDN)

There are four major components of Software Defined Network:

1. **SDN Application:** This acts as a bridge between the network resources, devices and the SDN controller through a Northbound API.

2. **SDN Controller:** The SDN controller forwards the requirements of the SDN Application to the SDN Datapaths. This also contains all the network policies that might be required by the SDN Applications. It also provides them with a view of the network and the network traffic. The controller communicates with the SDN Datapaths/ Dataplane via the Southbound API [3].

3. **SDN Datapaths:** It provides the data packets a path to move on the network by implementing switches.

4. **SDN API:** SDN Application Programming Interface (API) provides a path for open communication between the SDN Controller and the network routers [1].

Implementation of Software Defined Network

Following are the steps that have to be followed in order to implement the Software Defined Network:

1. **Find a problem:** First off we need to find a problem which might need the implementation of SDN in order to be solved effectively. If we do this it will allow us to measure the outcomes and implement them when we are fully switching over to SDN.

2. **Create a team with diverse skill set:** SDN has many moving parts and people with only one area of expertise will not be able to implement it successfully so we need a team with a diverse skillset.

3. **Test on a subset:** Before fully switching over to SDN, first we should test it on a low risk area of the network so that if a problem does arrive we do not incur high loss.

4. **Evaluate:** After the implementation of the SDN we must observe the data and see whether the implementation meets the desired outcomes, if it does only then should we implement for the whole the network [1]

SDN Infrastructure w.r.t Cloud Computing

There are three main SDN Infrastructures:

1. **SDN without Cloud Computing**

Although it is extremely rare to find implementation of SDN without Cloud Computing nowadays, we might find it in fairly large organizations which have many networking devices and may only need a centralized place most likely the IT department to manage the changes occurring to the network in real time. This type of architecture might be beneficial for the organization's network management but there might be a server bottleneck as the client has to request resources from the IT department.

2. **Cloud Computing without SDN**

In recent time most of the Cloud Computing models being deployed follow this architecture, this architecture allows the users to easily manage increased load without the need to worry about the scalability as the server manages the new servers and VM's on its own. But if we opt to install SDN, we need to wait for a bunch of networking configurations to be satisfied which might not be feasible for small organizations that are looking for a quick solution. This can also be implemented in places where the security is not that big of a deal such as private clouds so SDN is not required along with cloud computing for this scenario as it would just be an overkill.

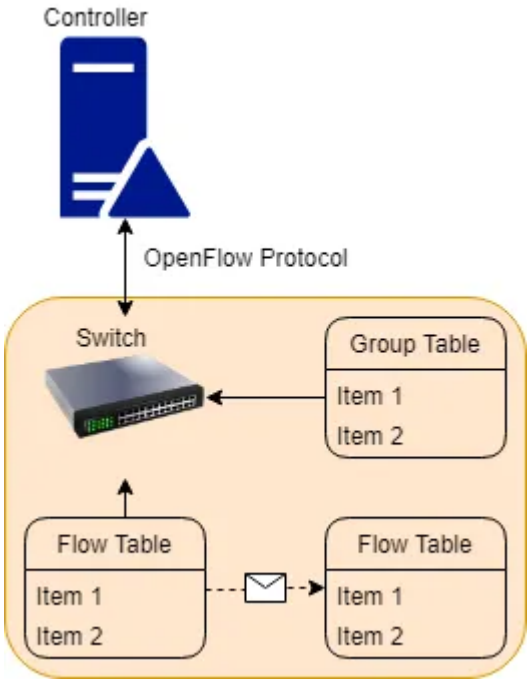
3. **SDN along with Cloud Computing**

In this scenario the organization might choose to deploy SDN along with Cloud Computing as they might need to orchestrate the cloud resources from a centralized point. These types of services are not as common nowadays but major companies are adopting them due to the benefits such as cost reduction, performance enhancement, increased security and other benefits which will be discussed below [5].

SDN Based Cloud Network

In an SDN Based Cloud Environment the communication between the controller and the Dataplane takes place via the OpenFlow Protocol as it allows network administrators to remotely manage routing tables, it also makes packet-switching decisions centralized which decouples the network from the individual switches and allows switches and data centers to be programmed independently.

A virtualization aspect has also been introduced via this protocol as the controller can be an entity operating on the cloud or it can be a physical component. The OpenFlow protocol is implemented in an OpenFlow switch which might be a hardware or a software component. Devices on the network can be connected to the OpenFlow switches to gain access to the SDN functionality. Traditional switches manage the Control path and Data path in the same device but an OpenFlow switch manages them separately. The Datapath layer exists within the switch while the controller layer is hosted on a different server, the OpenFlow switch then communicates with the controller via the OpenFlow protocol which retains information such as sent packets, received packets, updates done to forwarding table etc. [9].



Communication between switch and controller

In the figure above we can see the communication happening between the controller and the switch, the Flow Tables communicate with one another and share information such as set of instructions, matching fields and matching packets etc.

Functionality of Software Defined Network w.r.t. Cloud Computing

As we have discussed above that the control plane decides how the traffic will flow within the network i.e. it provides policies that govern the flow and the data plane will decide how the packets are to be moved i.e. which route has to be taken through the switches. So when a packet arrives at a switch with in the network then the control plane layer's functionality (centralized controller) comes into play and the policies and rules that it had provided to the switch's firmware decide where the packet is to be forwarded to.

Since the switch cannot decide on its own, it relies on the controller for the policies to be sent to it, it also sends data about the traffic that it has handled to the controller. All the packets bound for one destination are handled in the same manner which means same protocols are applied to them.

The above working is similar to when the SDN is combined with a Cloud Network and a virtualization aspect is introduced, this means that a virtual cloud network is running on top of the physical network in a layered manner. This might be done in order to separate the physical network and the network traffic in the virtualized environment. Such type of network segmentation is extremely helpful for Cloud Vendors which are running multi-tenant cloud environments as it allows them to define a different set of policies and rules for each of their different tenants [2].

Implementing SDN on the Cloud Implementing

SDN on the Cloud cannot be done without taking into account all the minor details, we need to plan out the whole implementation thoroughly as it is not a static system and updates will have to be done while the system is running without any downtime. Following details have to be taken into account while planning the implementation of SDN on the Cloud:

1. **Find cloud providers with robust enough framework to support SDN on their Cloud**

In order to implement SDN on the cloud we will have to find the Cloud Vendors whose framework supports SDNs, after finding such Cloud Vendors we will need to migrate the SDNs in a third party Cloud service which can be any public cloud service such as Google Cloud Platform(GCP) or Amazon Web Services(AWS). This type of service will allow us to make load balancers, firewalls and other services available on the Cloud which the tenants can scale according to their needs.

2. **Understand the applications requirements to make the network dynamic**

The major reason why people prefer moving SDNs to the Cloud is the ability for the network to understand the changing demands and adapt to accommodate the applications demands. It will be extremely difficult to make the network dynamic without thoroughly understanding the applications needs. We need to outline all the programmable aspects which change the networks scalability, throughput, response time and security etc. The network will be the most efficient when we make it the most dynamic.

3. **Identify deployment approach**

There are three ways in which we can deploy our SDN which will be discussed below:

3.1. Private SDN: This means that the SDN will reside on the user's infrastructure, this will not be able to make use of the advantages provided by the cloud such as the cost reduction and scalability.

3.2. Public SDN: This type of SDN will be hosted entirely on the cloud and we will be able to make use of all the advantages associated with the cloud.

3.3. Hybrid SDN: In hybrid SDN partial system will be hosted on the cloud and the remaining part will be on the company premises. This approach is best suitable for use cases where we need to push and pull data constantly form the database.

4. **Determine how you will migrate the current network and test it**

This is one of the most important part of the procedure of switching an SDN to the cloud as we will need to define how the applications will make use of the network services which have been migrated over to the cloud. We will also need to test these services, which can be done by assessing the scalability, security and performance etc. We will also need to define a future plan of how the SDN will be maintained and what roles each actor will play [8].

Benefits of Software Defined Network for Cloud Computing

Following are the benefits of Software Defined Network for Cloud Computing:

1. SDN allows the network administrator to manage the protocols, rules for the switches in the network, it allows them to make very minute changes such as blocking certain packets from reaching a switch. This type of control is

extremely helpful when the network is running a cloud multi-tenant architecture since traffic loads can be managed very efficiently.

2. The centralized system allows the network administrator to manage all the switches in the network from one point. This also provides a security advantage as configuring each devices individually is very inefficient and a lapse of concentration might result in wrong policies being dispatched so this allows security policies to be dispatched pretty easily by observing the network's traffic.

3. SDN also allows the virtualization of the hardware equipment this allows organization to skip the management of physical hardware and not bear the hardware cost.

4. Due to SDN another technology has emerged which is Software Defined Wide Area Network (SD-WAN), this is similar to SDN but it allows the organization to manage its WAN without the need to worry about all the connection details between the different switches of different departments and the controller manages where the traffic has to be sent to [4].

5. SDN helps in reducing the downtime of the network as it virtualizes most of the physical networking devices. This allows us to quickly recover from failures, perform upgrades to subsystems easily without closing down the whole network.

6. The SDN controller allows us to manage and oversee the traffic in the network down to the packet level, this granular observation of data from a centralized position allows us to detect any inconsistencies easily and dispatch the required security policies rapidly [6].

Challenges with Software Defined Network on the Cloud

Since SDN is a relatively new technology hence switching over to it might come with its own challenges which are described below:

1. **Switching over too early:** If you switch over to SDN without assessing all the fine details there is a high chance of failure as there are a lot of moving parts which have to be accommodated correctly [8].

2. **Single point of failure:** Since everything is managed by a centralized controller, this provides a single point of failure and if an attacker can gain access to the

SDNs cloud it will be extremely damaging.

3. **Cost:** Since its inception it has not seen that high of an adoption rate as which was expected, this is due to the fact that there might be extra resources required in order to implement SDN and integrate it with the cloud efficiently whose cost mid to low sized firms would not want bear.

4. **Uncertainty about the best model:** Since there is no clear option for SDN deployment and vendors offers different choices such as private (hardware oriented), public (fully cloud based) and hybrid model (combination of both). This creates confusion about which model will be the best as implementing one model and then trying to switch over to a different model is very costly [4].

Implementation of SDN in different fields

SDN can be implemented in the following fields may it be with or without the cloud:

1. **University Networks:** In the current day and age provision of free Wi-Fi on campus premises has become a necessity most universities are trying to combine their legacy Ethernet systems with the new Wi-Fi systems. SDN allows them to manage everything from a centralized point.

2. **Service Provider Networks:** Implementation of SDN will help in the simplification of the service provision framework by providing a single point of management.

3. **DevOps:** DevOps can greatly benefit from SDNs by allowing the automation of application maintenance, updates and deployments.

4. **Increasing Data Center Security:** Most of the traditional data centers use a single firewall to guard against any breach but SDN allows us to implement separate virtual firewalls for every virtual machine in the network, this helps in preventing any breach from spreading to other VM's. The centralized controller also helps us in keeping an eye on the network activity and modify different policies to reduce the risk of a breach [4].

Conclusion

As we have discussed in detail above about the implementation of SDN as a standalone architecture or its implementation in conjunction with the cloud we can see that there are immense benefits to the usage of such an architecture. The only drawback that one might associate with the implementation is the need for

thorough planning about the usage of resources and the maintenance of the network as the architecture is a bit complex and might not be applicable for the use case at hand.

Extra Details

Following are the details about the new terminologies that we have come across above:

1. **Northbound Interface/API:** These are the Interfaces/API's that are used in order to communicate with components at a higher level from the current level. When the flow moves upwards it can be considered as northbound flow.
2. **Southbound Interface/API:** In SDN Southbound Interface/API is the OpenFlow protocol layer which facilitates SDN controller to communicate with the physical components i.e. the switches.
3. **OpenFlow Protocol:** It is the protocol that is used by the OpenFlow switch to communicate with the SDN controller. It helps in adding, updating and deleting flow entries from the flow table which allows the switch to keep track of sent packets, received packets, updates done to forwarding table etc.
4. **Group Table:** A group table resides inside the OpenFlow switch and contains a number of flow entries.
5. **Flow Tables:** A flow table consists of a set of different packet related information such as the packet header which tells it where the packet is coming from, action to be performed such as the forwarding of the packet from a particular port and statistics which is the number of packets and bytes that have flowed through the switch

References

Websites

[1] "What Is Software Defined Networking? Definition & FAQs." Avi Networks, 19 Mar. 2021, Available: avinetworks.com/glossary/software-defined-networking/.

[2] Rimavi, Mindaugas. "Software Defined Network." Medium, Syntropy, 1 Feb. 2019, Available: medium.com/syntropynet/software-defined-network-572b1deeabb.

[4] Rosencrance, Linda, et al. "What Is Software-Defined Networking (SDN)?" SearchNetworking, TechTarget, 17 Dec. 2020, Available: