Mansoura University

Faculty of Computers and Information Sciences (Information Technology Dept.)

Software Defined Network

SDN

By

Amr Elsayed Mohamed Ahmed
Yassien abdalla yassien abdalla
Yusuf Mohamed yusuf mosa
Mahmoud Gamal Mohamed El Ghareeb
Mohamed Tarek Abd El Aziz Abd El kader
Androw Amir Abdelmeseh Nesem
Mohamed Khalid aboelsayed
Alaa Abd Elbasset Elmetwally Awad
Mohamed Zinhom Elsayed Shillua
Mohamed abdelrazek Mohamed elshazly

Supervised by

Dr. Weal GabAllah

Abstract

Software Defined Networking (SDN):-

SDN enables the programming of network behavior in a centrally controlled manner through software applications using open APIs. By opening up traditionally closed network platforms and implementing a common SDN control layer, operators can manage the entire network and its devices consistently, regardless of the complexity of the underlying network technology.

Acknowledgement

We would like to express our special thanks to Dr. Weal GabAllah for her great efforts in making this project, and we also are very thankful to all lecturers of Faculty of Computer & Information Science, Mansoura University, we also would like to thank our great parents who have spared no effort to help and support us all the time. We promise we will keep doing ourbest to make them proud of their sons. We had a wonderful time spent in the Collegeof Computer and Information Sciences, which taught us many things, and now is the time to use what we had learned.

Chapter1 Introduction

1.1 Introduction:

Network Virtualization techniques are not something new in the network world. virtual local area networks (VLANs), tunneling, and virtual private network (VPNs) have been around for quite long. So, Software defined networking (SDN), it intends to centralize all information (control planes) in the network on a software layer allowing centralized control and abstraction of the underlying complex infrastructure. Theoretically, all network nodes would only need the muscle (forwarding or data plane) to push packets out.The major difference between SDN and traditional networking is infrastructure: SDN is software-based, while traditional networking is hardware-based. Because the control plane is software-based, SDN is much more flexible than traditional networking. It allow administrators to control the network, change configuration settings, provision resources, and increase network capacity from a centralized user interface, without the need for more hardware.

SDN has many benefits. One significant advantage of using an SDN is its ability to enhance a servers' processing ability, which could help to reduce network latency. An additional benefit of using an SDN is its ability to compress storage.

The most significant SDN benefits are derived from the opportunity to increase flexibility in the network, making it easier to manage. Using SDN enables networking functions to be automated easily which improves efficiency, and allows networked resources to be easily managed from anywhere.

1.2 Problem Definition :

SDN differs from traditional networking in that it is software-based, whereas traditional networking is typically hardware-based. SDN is more flexible as it is software-based, giving users more freedom and suitability to manag resources virtually throughout the control plane. On the other hand, Traditional networks, make connections and run the network using switches, routers, and other physical infrastructure.

A northbound interface on SDN controllers communicates with APIs. Because of this communication, instead of employing the protocols required by traditional networking, application developers can directly program the network.

SDN allows IT administrators to direct network channels and proactively arrange network services by allowing users to supply new devices using software rather than physical infrastructure. SDN, unlike traditional switches, may also connect more effectively with networked devices.

The primary difference between SDN and traditional networking is represented by virtualization. When you virtualize your whole network with SDN, you get an abstract duplicate of your physical network that you can manage from a single location.

In a traditional network, on the other hand, the physical status of the control plane makes it difficult for an IT administrator to control traffic flow.

The control plane becomes software-based with SDN, making it accessible via a connected device. This access allows IT managers to better manage traffic flow from a centralized user interface (UI). Users have more control over how their networks run and are setup thanks to this one place. For network segmentation,

the ability to swiftly process multiple network configurations from a centralized UI is extremely useful.

Because it allows IT administrators to save resources and bandwidths as needed without having to invest in extra physical infrastructure, SDN has become a popular alternative to traditional networking. To expand network capacity, traditional networking requires new hardware. The difference between SDN and traditional networking can be summarized as follows: one requires more equipment for expansion, while the other just requires keystrokes.

1.3 Project Objective:

As we know that the expansion of networks is continues, so we need SDN to achieve the following:-

- Helps to expand the network.
- Save time and effort.
- Add different vendors to the network .
- User friendly and make tasks easily.
- Optimizes resource utilization and operating efficiency.
- High performance computing techniques.
- Provides the power quality for the range of need.
- Accommodation of all generation and storage options.
- Prevents interference between network administrators.
- Provides more security for the whole network.

Document Organization:

This project consists of six chapters .These chapters are organized to provide the scientific steps toward our main objectives.

A brief description about the contents of each chapter is given in the following schedule:

Chapter 1	Introduces our project.
Chapter 2	Provides the reader with an overview of the literature review (SDN Architecture) and contains background information project.
Chapter 3	Open flow.
Chapter 4	Provides an overview of system analysis for DNA CENTER.

chapter 2

Literature Review

chapter 2: literature Review

2.1 Traditional Networking vs. SDN:

For the control plane, traditional networking implements a distributed paradigm. For each network device protocols as,(ARP,STP,OSPF,BGP,EIGRP) and others operate independently. These network devices connect, but no centralized machine manages the whole network. The most critical difference between conventional networking and SDN is that traditional networking is hardware-based, whereas SDN is usually software-based. SDN is more diverse as it is software-based, helping users better control and ease handling resources remotely in the control plane.

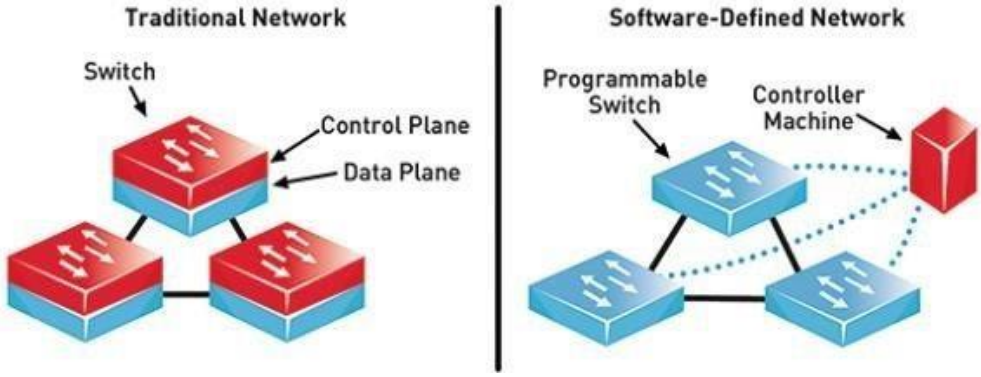


Fig. 1. The architecture of SDN Vs Traditional network

Traditional networks utilize switches, routers, and other physical hardware to produce connections and operate the network. A northbound interface that communicates with Application Programming Interfaces (APIs) is used in SDN controllers. Because of this connectivity, device developers, as opposed to using the protocols required for conventional networking, can explicitly program the network.

Conventional networks are used to mount all data planes and control planes in one physical unit and then to share their capacity, increase the traffic load and the burden on the CPU and memory in two processes. Detachments of control planes and data planes in SDN can be easily monitored and managed by the controller and network to take the right ride decisions and thus enable the network to better configure with a less traffic load, by separating these processes and having a dedicated server.

SDN is considered a popular alternative to traditional networking because it allows IT managers to provide extra physical infrastructure services and bandwidths without requiring an investment. In order to expand the network power, traditional networking requires new hardware

2.2 Need for SDN:

SDN is defined as a modern paradigm that is rapidly becoming the alternative for networks that are unable to solve the shortages of traditional networking via isolating software from the hardware. In SDN, management/control is provided for the hardware from a centralized software program. This software program is isolated from the hardware itself. The prime focused need of SDN is an open source framework standard and layered architecture. Because software can be produced via different vendors easily, it is more effective, more flexible programmability, and more facilitating creativity in computer networking. In SDN, several issues need to be addressed, such as scalability problems, virtualization, continuity of connectivity, and location of controllers. Reliability is one of the serious SDN difficulties. Reliability is an especially important issue for large-scale networks. As the SDN controller tends to be a single point of failure, it is a technically unified control feature in the SDN. Accordingly, steps need to be taken to ensure that the reliability of modern technological solutions is at least as high as or better than before. SDN is one of the most important innovations for developing the new economy's network infrastructure. However, unreliable networks cannot be the basis of the digital economy.

2.3Architecture of SDN:

SDN Architecture explains how SDN operates at its different stages and ensures the stability and reliability of software. For software-defined networking, there are primarily three layers: Application plane, Data plane, and Control plane. SDN consists of 2 interfaces, one between the southbound APIs (e.g., OpenFlow) and the other between the API's application layer and the Northbound API's control layer.

2.3.1 Control plane:

It can be defined as a control layer. It includes a series of software-based SDN controllers that provide a centralized control mechanism by a well-defined API to oversee network forwarding actions through an open interface. Generally, The control plane consists of three primary layers, (the device layer, the network operating system layer, and the network abstraction layer).

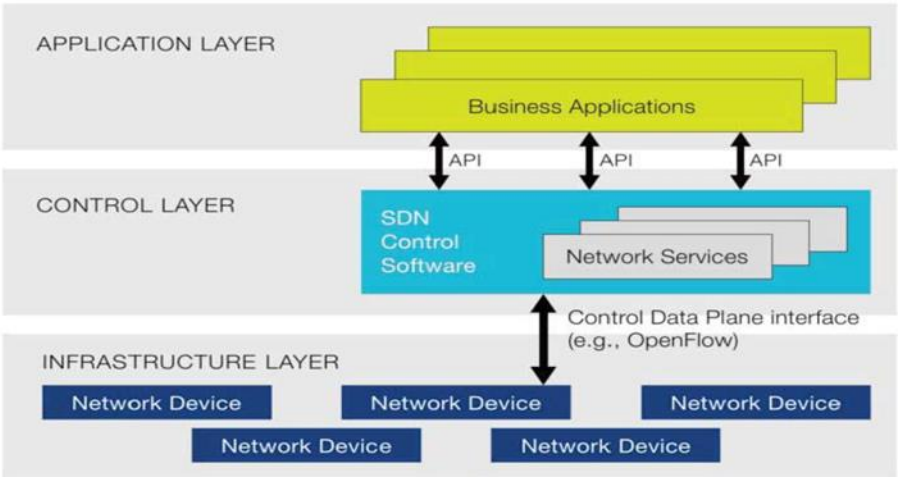


Fig. 2. SDN architecture

2.3.2 Southbound APIs:

To connect with the SDN controller and network switches and routers, SDN southbound APIs are used. In this interface, the most common protocol is the OpenFlow protocol.

2.3.3 Application plane:

The application layer consists of one or more programs, each of has exclusive power over one or more SDN controllers exposed to a collection of resources part of the SDN architecture, which consists of software implementing network services delivered to users/devices. In order achieve an abstract global view of the network they are using and to express the network activity they require at the moment, applications connect with the SDN controller by APIs (northbound interface).

2.3.4 Northbound APIs:

The relation between the applications and the SDN controller is the northbound APIs. The applications should inform the network what they need, and those services can be given by the network or convey what it has.

2.3.5 Infrastructure plane:

The infrastructure plane is also known as the data layer or data plane. Like the OSI model's physical layer, it comprises network components that interact with data traffic, such as physical and virtual machines. It is an SDN forwarding plane and responsible for forwarding packet frames physically via the protocols used by the control plane from its entrance .

2.4 Benefits of SDN:

One of SDN's key benefits is that it provides a platform for promoting more data-intensive software, for instance, virtualization and big data.

2.4.1 Centralized networking management:

SDN can control the whole network from a centralized unit called a central node to automate network administration and security, and ensure that security knowledge is reliably communicated across the organization.

2.4.2 Reduced hardware costs:

SDN uses the software principle to create a network with the minimal hardware available, removing the need for manual assistance and the expense of setup by the effectiveness of the organization performance and improving network usage by utilizing the virtualization concept.

2.4.3 Security approach:

It gets easier to track and control the security features when there is a single management console for networking. It may not have to deal with several applications around the system or dependent on them. It operates from one central point easily and provides a better security strategy. When there is a security-related alarm, the same console may also be used to divide information. In order to keep up with network management, virtualization made it more complex for IT administrators. Applying filtering rules and firewalls can be challenging for many virtual devices connected to the physical networks. With SDN, it is possible to monitor and spread all information and safety measures consistently within the organization.

2.4.4 Automation:

Today's network does not have to deal with internet access, unlike before. With SDN, it is also possible to adjust the cloud's automatic responses. In environments like enterprise-wide SD-WAN networks, the process works well.

2.5 Challenges of SDN:

Even though SDN is identified as the basic solution to the problems that the infrastructure of the expanding network is facing major, it is still in its infancy phase. In addition to many others, advantages such as better functionality, lower cost, and higher efficiency have been laid out, but different challenges also demand attention.

2.5.1 Scalability:

The main problems faced by SDN are scalability. From this single problem ,we can have two problems:

a)scalability of the controller.

b)scalability of the network node.

A single controller can handle up to 6 million flows per second. Therefore, this demonstrates that for a large number of data forwarding nodes, only one controller or several controllers can manage control plane services needed. To enhance scalability, rather than functioning on a peer-to-peer basis, the logically centralized controller should be physically distributed. However, the problems faced by the controller when interaction happens will be shared between network nodes, whether it be a distributed or peer-to-peer controller network. Hyper Flow and Onix are known as efficient means of achieving scalability. Through allocating and partitioning network status to separate physically dispersed controllers, Onix runs. HyperFlow is an application that allows for the interconnection of OpenFlow networks that are individually controlled. Specifically, the events that allow changes to the network condition will be distributed by HyperFlow program, then all the distributed events will be replayed by the other controllers to reproduce the situation. on this way, with the same homogeneous network topology.

2.5.2 Flexibility and performance:

How to deal with high-level packet processing flows proficiently is a fundamental problem of SDN. There are two main factors to be in consider (flexibility and performance). Flexibility refers to the ability of networks to respond to modern and unprecedented functionality, such as software and facilities for the network. The performance deals with the speed at which information is transmitted from the control plane via network nodes in the data plane.

2.5.3 Security challenges:

In software-defined networking, security is a very critical feature. In order to provide usability, integrity, and protection to all elements and info, SDN protection needs to be integrated into the architecture. You will have to secure and defend the device, depend on the SDN of each component, make sure the controller does what you want, and when a crash occurs, the architecture should be able to detect, fix and display the problem. The division of the data and control plane allows for security breaches and SDN safety issues. The optimal location of SDN controllers, switches, and other devices is an open challenge in SDN, which affects overall network security and performance. Its integration is another security problem because of the design of SDN as it is flat, Where monitoring systems and defense solutions need to be compatible to improve overall performance, energy savings, and network security.

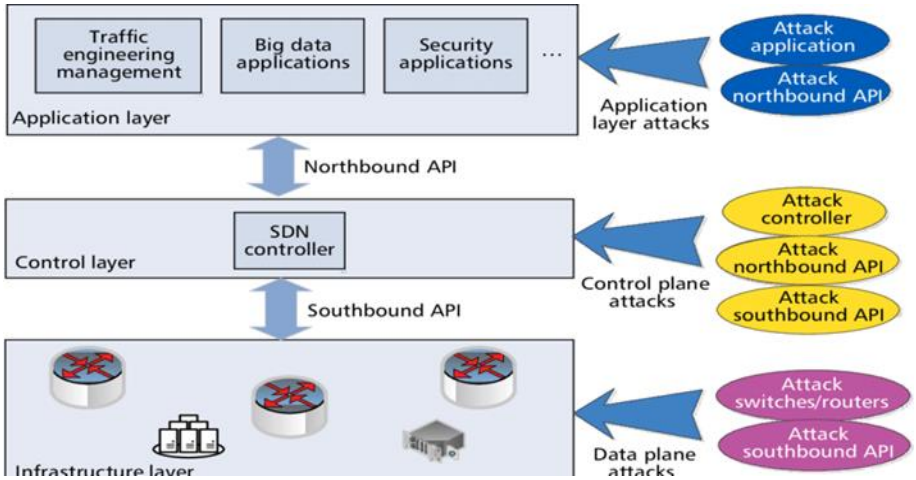


Fig. 3. Probable Attacks on SDN Architecture

A. Data plane layer security challenge:

The flood tables in the data plane lack space and flow tables storage flow entries generate overhead on flow tables, leading to high cost and low performance. Using intelligent flow table control techniques to store many low-cost and high-performance rules will overcome this problem. Switches or access points can interrupt network activity, which results from malicious users initiating a Denial of Service (DoS) attack resulting in the interruption or network loss.

B. Control plane layer security challenge:

Controllers are fundamental to SDN, but because of their centralized decision-making that can trigger networking in a security breach, it becomes a single point of failure. The control layer is an attractive function for security attacks due to its transparent environment. Another problem is how many switches to the controller are attached, and requests are sent to the controller, waiting for a response. If you add many switches to your controller's response time, your controller can crash due to the load on the controller.

C. Application plane layer security challenge:

The hacker can flood malicious data into the application layer to monitor a network node that can infect other connected network nodes. By inserting malicious code to monitor network packets' flow and steal valuable information, the attacker may obtain unauthorized access to the network node.

Chapter 3

OPENFLOW

3.1 Introduction

The OpenFlow protocol structures communication between the control and data planes of supported network devices ;

OpenFlow has been designed to provide an external application with access to the forwarding plane of a network switch (or router). Access to this part of the router can be gained over the network, which allows the controlling program not to have to be collocated with the network switch.

Traditional networking protocols have tended to be defined in isolation, with each solving a specific problem and without the benefit of any fundamental abstractions. The result of this isolation has been the creation of one of the primary limitations of today's networks: complexity.

As an example of this complexity, to move a device from one location on the network to another location on the network, networking professionals must touch multiple switches, routers, firewalls, web authentication portals, and so on and update access control lists (ACLs), virtual local-area networks (VLANs), quality of services (QoS), and other protocol-based mechanisms (ONF, 2012) using network management tools that operate at the device and link levels.

In addition, when these types of changes are being made, the network topology, vendor switch model, and software version all have to be taken into account. The end result of this network complexity is that once a network is built, it often stays as it is so that nothing becomes broken.

The OpenFlow protocol has been created to solve the problems that legacy networking protocols have created. In the software-defined networking (SDN) architecture, OpenFlow is the first standard communications interface defined between the control and forwarding layers.

OpenFlow allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor based). Currently, no other standard protocol does what OpenFlow does, and it has been determined that a protocol like OpenFlow is needed to move network control out of the networking switches to logically centralized control software.

When the OpenFlow protocol is implemented, it is implemented on both sides of the interface between the network infrastructure devices and the SDN control software. To identify network traffic, OpenFlow uses the concept of flows based on predefined match rules that can be statically or dynamically programmed by the SDN control software. OpenFlow allows network professionals to define how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources. OpenFlow allows the network to be programmed on a per flow basis.

This means that an OpenFlow-based SDN architecture can provide extremely granular control, enabling SDN to respond to real-time changes at the application, user, and session levels. In today's legacy networks, routing based on the Internet Protocol (IP) does not provide this level of control because all flows between two end points must follow the same path through the network, regardless of their different requirements.

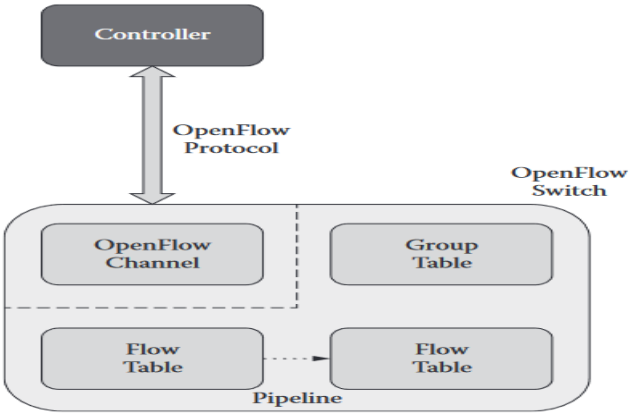
The OpenFlow protocol has been created to enable software-defined networks and is now the only standardized SDN protocol that permits direct manipulation of the forwarding plane of network devices. OpenFlow was initially applied to Ethernet-based networks; however, OpenFlow switching can extend to a much broader set of use cases. OpenFlow-based SDNs can be deployed on existing networks, both physical and virtual.

Network devices can simultaneously support OpenFlow-based forwarding as well as traditional forwarding. This means that it is easy for enterprises and carriers to progressively introduce OpenFlow-based SDN technologies, even in multivendor network environments.

3.2 Overview of the OpenFlow switch specification

An OpenFlow switch consists of three main components: an OpenFlow channel, a group table, and one or more flow tables. The flow tables and the group table are responsible

for performing packet lookups and packet forwarding. The OpenFlow channel is used to communicate with an external controller. The external controller uses the OpenFlow protocol to manage one or more OpenFlow switches.



Fig(1)

In an OpenFlow switch, the abstraction at a high level is that Ethernet frames ("packets") arrive, and their headers are compared to data that has been stored in a flow table within the switch. Every OpenFlow switch must contain one flow table and can contain multiple flow tables. If a match between the contents of the packet's header fields and the flow table entry is made, then a set of instructions is then executed. Each one of the switch's flow table entries contains three pieces of information: the data that is used to match the fields in the received packets ("match fields"), counters that keep track of the number of matches that have been made, and the instructions that are to be executed if a match is successfully made.

The matching of the fields in a packet starts with the first table (numbered "0") and continues until a match is made or a table-miss event is declared to have happened. Exactly what happens when a table-miss event occurs is dependent on how the switch is configured, but options include dropping the packet, forwarding it to the controller for further processing over the OpenFlow channel, or continuing on to the next flow table to continue the search for a match to the packet.

As the network changes, the OpenFlow switch's flow tables have to be updated. Updating these tables is the responsibility of the external controller to which the switch is connected.

If a packet reaches the end of a flow table and it still has not been matched, then if the table-miss instructions modify the packet-processing pipeline, the packet may be allowed to be sent to the next flow table in the pipeline. Any time the instructions that have been retrieved from a flow table entry because of a packet match or a table-miss event do not specify a next flow table, then processing of the packet will come to a halt. The packet will generally then be modified and forwarded to the next switch.

The OpenFlow switch specification is just that a specification.

This means that the designers of switches have the ability to implement the OpenFlow functionality in any way as long as it conforms to the standard. The functionality of the switch can be split between hardware and software in any fashion that the designer chooses.

3.3 OpenFlow ports

Ports are a critical part of the OpenFlow protocol because they specify where a packet comes from and ultimately where it will be going. When two OpenFlow switches are connected, they are connected via ports.

An OpenFlow switch contains three different types of ports: physical,logical, and reserved. Each of these port types behaves differently. Any one of these port types can be used as both an input and an output port for a packet.

Physical ports are exactly what they sound like ports that correspond to a physical interface port on the OpenFlow switch hardware.

Logical ports are not related to a physical port on the switch. However,logical ports can be made to map to a physical port on the switch. When packets are being processed by the OpenFlow switch, both physical and logical ports are treated exactly the same way.

Reserved ports are special ports that are used to cause a specificaction to occur. This action is triggered by sending a packet to a reserved port. An OpenFlow switch is required to support five types of reserved ports. There are three additional types of optional reserved ports that can be supported by the switch.

3.4 OpenFlow packet-processing

he OpenFlow switch to send this packet to another table (that table must have a larger table number than the table that is currently processing the packet) to continue to attempt to make more matches between flow entries and the packet's header bits.

The packet-processing pipeline will stop when the flow entry that is matched to the packet does not have instructions that request that the packet be sent to another flow table for processing.

Once this happens, the rest of the instruction set for this flow table entry will be processed against the packet, and then the packet will be forwarded by the switch.

Packets will not always match to the current contents of a flow table.

When this occurs, it is called a table-miss, and it is handled as specified in the implementation of the OpenFlow protocol on a switch.

Many different actions can be taken, including dropping the packet, passing the packet on to another table, or sending it to the controller via the control channel using packet-in messages .

3.4.1 Flow tables

it the center of an OpenFlow switch's packet-processing pipeline are its flow tables. These tables are used to determine what, if any, action should be taken based on receiving a given packet. The flow tables are an important part of the OpenFlow switch's packet-processing pipeline.

Each flow table consists of a number of flow entries, and each flow entry consists of six information items. Two of these items are the match fields and the priority. The match fields are the values that are compared against specific fields in a received packet to determine if there is a match, It is possible that multiple flow table entries may match the same packet at the same time. If this occurs, then the flow entry's priority value is used to determine which match will be used to provide the instructions that will be executed against the packet.

3.4.2 OpenFlow channel

An OpenFlow switch is connected to an external controller via an OpenFlow channel. This is the interface that the controller uses to configure and manage the switch, receive events from the switch, and send packets out of the switch. The OpenFlow protocol supports the following three types of messages for exchanging information between the controller and the OpenFlow switch:

1.Controller to switch

2. Asynchronous

3. Symmetric

Controller-to-switch messages are used to directly manage or inspect the state of the switch and are initiated by the controller.

Asynchronous messages are used to update the controller with network events and changes to the switch state and are initiated by the switch.

Symmetric messages are sent without solicitation and are initiated by either the switch or the controller.

There are seven controller-to-switch messages that are initiated by the controller and may or may not require a response from the switch.

The controller does not have to request that asynchronous messages be sent from an OpenFlow switch. Asynchronous messages are sent by the switch to controllers to denote a packet arrival or switch state change, There are three main types of asynchronous messages. Symmetric messages can be sent without solicitation in either direction. Four symmetric messages have been defined as a part of the OpenFlow protocol.

3.4.3 OpenFlow channel connections

An OpenFlow switch and an OpenFlow controller exchange OpenFlow messages using an OpenFlow channel. In general, a single OpenFlow controller will communicate with multiple OpenFlow switches using multiple OpenFlow channels. A single OpenFlow switch will typically have either a single OpenFlow channel connection to a single OpenFlow controller or multiple OpenFlow connections to multiple OpenFlow controllers for backup and reliability.

An OpenFlow controller is generally located remotely and uses one or more networks to connect to a given OpenFlow switch. The only requirement of the controller/switch network is that it supports the Transmission Control Protocol/Internet Protocol (TCP/IP). The network that is used to support controller-to-switch communications can be a dedicated network, a shared network, or an in-band network (the network that is being managed by the OpenFlow switch).

The OpenFlow channel between the OpenFlow switch and the OpenFlow controller is generally a single network connection that uses the Transport-Layer Security (TLS) or plain TCP. It is possible to create an OpenFlow connection that is composed of multiple network connections to exploit parallelism.

The OpenFlow switch is responsible for establishing a connection with the OpenFlow controller. In some cases, the OpenFlow switch may permit the OpenFlow controller to establish a connection with it. However, in this case, the switch usually should restrict itself to using only secured connections (TLS) to prevent unauthorized access to the switch.

3.5 Open Flow Versions

This section contains release notes highlighting the main changes between the main versions of the OpenFlow protocol.

1-OpenFlow version 0.2.0 (Release date : March 28,2008 Protocol version : 1)

2-OpenFlow version 0.8.0 (Release date : May 5, 2008Protocol version : 0x83) :

• Reorganized OpenFlow message types

• Add OFPP_TABLE virtual port to send packet-out packet to the tables

• Add global flag OFPC_SEND_FLOW_EXP to configure flow expired messages generation

• Add flow priority

• Remove flow Group-ID (experimental QoS support)

• Add Error messages

• Make stat request and stat reply more generic, with a generic header and stat specific body

• Change fragmentation strategy for stats reply, use explicit flag OFPSF_REPLY_MORE instead of empty packet

• Add table stats and port stats messages

3- OpenFlow version 0.8.1 (Release date : May 20, 2008Protocol version : 0x83)

Chapter 4

SYSTEM ANALYSIS FOR DNA CENTER

4.1 Introduction

In this chapter we will discuss the system analysis for Cisco DNA Controller and it's consider the best and we will know why he is the best and after we talking about SDN and the history of the network evolution now we should know what is the controller and the features of the system controller what is it composed of.

So, what is Cisco DNA controller stand for Cisco Digital Network Architecture and Cisco DNA Center is based on something called Intent-Based Networking.

It is a new approach to networking in that the network admin can now define and input what the needs of the network are into the IBN software controller. This ensures that the network works in conjunction with the needs of the business.

We choose the Cisco DNA because it simplifies network operations, configurations and troubleshooting. With DNA Center, your network feels Network Automation convenience. Your network is managed easily, network operations are done efficiently and your network down times decrease and Cisco DNA Center is a powerful network controller and management dashboard for secure access to networks and applications. It lets you take charge of your network, optimize your Cisco investment, and lower your IT spending.

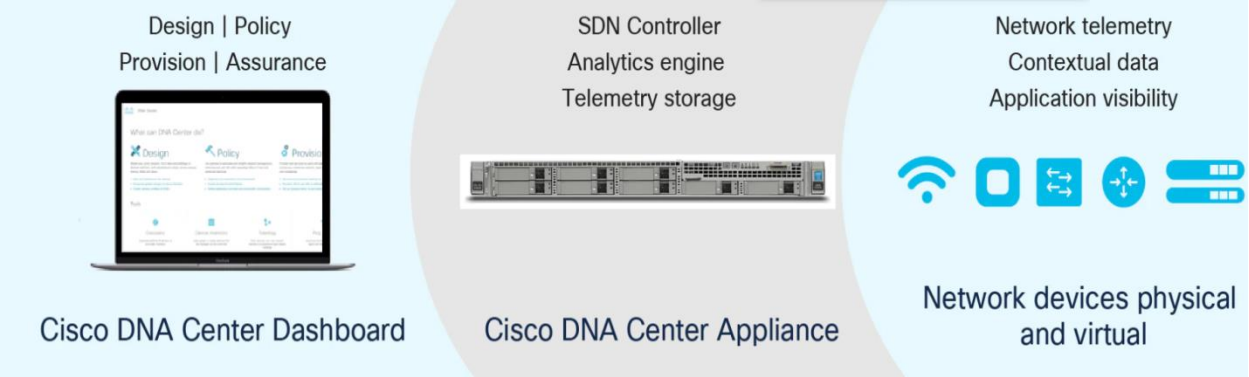
This chapter is important for simplify how the system work and how dealing with the controller , we are in 2022 and Network Automation is consider as mutation in IT filed and the programming language is important for the Network Engineer to improve himself he must study and know the programming because keep himself and improve himself and the Enterprise companies going now to using the controller to control they our network so you must improve yourself with new technology.

4.2 CISCO DNA Center Overview

our network is more strategic to your business than ever before. You need a network management system that can automate the deployment, connectivity, and lifecycle of your infrastructure and proactively maintain the quality and security of your applications so that your IT staff can focus on networking projects that enhance your core business. You need an intent-based networking controller.

As the foundational controller and analytics platform at the heart of Cisco's intent-based network, Cisco DNA Center is a set of software solutions that manages your network, automates your virtual devices and services, and, with its assurance capabilities, supports the best network experience for all your users. With Cisco DNA Center, the days of time-consuming network provisioning and tedious troubleshooting tasks are over. Plug-and-play (PnP) deployment and Software Image Management (SWIM) features reduce device installation and upgrade times from hours to minutes, and new remote offices using off-the-shelf Cisco devices can be brought online with ease. Through its assurance feature, Cisco DNA Center enables every point on the network to become a sensor, sending continuous, streaming telemetry on application performance and user connectivity in real time. This capability, coupled with automatic path trace visibility and guided remediation, means network issues are resolved in minutes before they become problems. Integration with Cisco security solutions such as Cisco Secure Network Analytics and Cisco Umbrella™ provides DNS protection, detection, and mitigation of threats, even when they are hidden in encrypted traffic.

Cisco DNA Center also provides an open, extensible platform with broad support for external applications and systems to exchange data and intelligence, building upon its native functions. And it is the only centralized network controller to bring all of this functionality into a single pane of glass.



Fig(1)

4.3 System Requirement

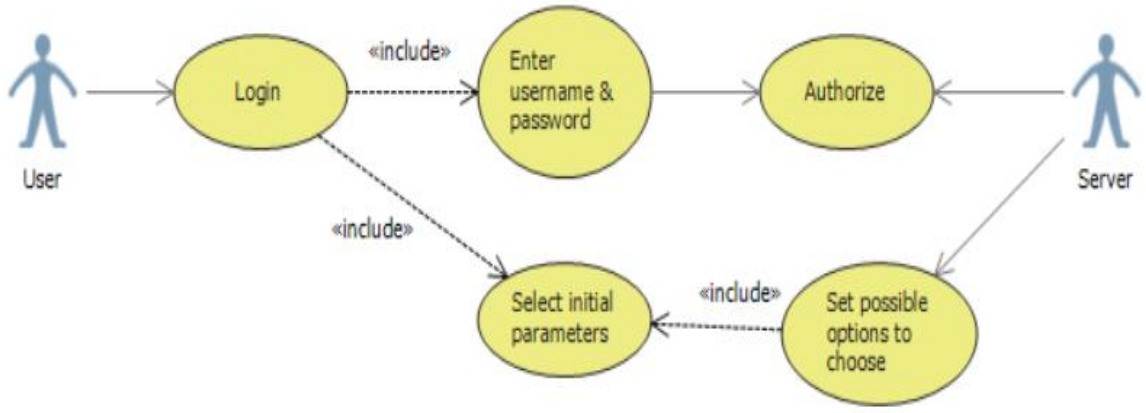
In this chapter we will define the functional requirement and nonfunctional requirement about the user and the Next Generation Network Engineer and the system in brief and why we choose the cisco DNA.

4.3.1 Functional Requirements

For user and network engineer it will be under conditions and important for us to make the network stable so we will start by the User Functional Requirements and second the Network Engineer and the end we will explain the functional requirements for the Cisco DNA controller.

4.3.1.1 User Requirements

The User in the network will be care about speed of the network and reachability to service that network introduce for the user like DNS ,DHCP,INTERNET or Database of clients or whatever the service the network will provide to they our users so the user don't care about the Network Engineer can do to provides the service for him , we will say in brief the user care only about the services to be accessible and the speed for reachability for the Network system .



Fig(2)

In Fig(2), Every user have (Password and username) to access the Network we called it User Authentication and the Network Engineer will put a privilege to the users in the Network .In use case diagram in Fig(2) explain the user what care about.

4.3.1.2 NETWORK ENGINNER Requirements

The Responsibility is big here because the network engineer job is so hard they work for 24/7 h to keep the users connected and services up and NE who dealing with the DNA Controller to give the configuration of network devices and monitoring the performance and how many users in network and protect the network from attacks, most of this tasks we handling it by the controller so the DNA controller makes it easy for us.

4.3.2 Non functional Requirement

The user and network admin between then many intersection point because the care about the performance and speed and reachability .
We classified it into 2 categories for the (user , network admin) , so we will start with :
USER , he will don't care about the controller or because not his job
Network Admin :he is the boss of the system and care about anything inside the network include the user requirement because the network admin here to make the user accessible to his service or internet or whatever .

4.4 CISCO DNA CONTROLLER

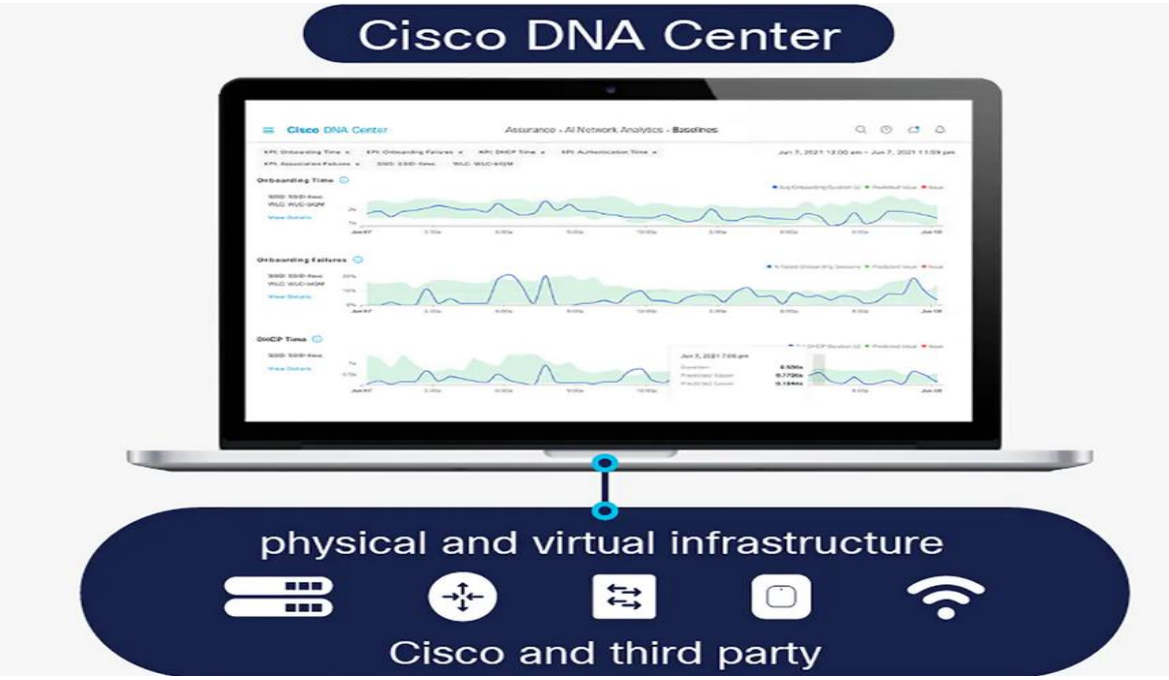
Cisco DNA Center is a powerful network controller and management dashboard for secure access to networks and applications. It lets you take charge of your network, optimize your Cisco investment, and lower your IT spending.

4.4.1 why we choose the cisco DNA :

Cisco DNA Center is a complete management and control platform that simplifies and streamlines network operations. This single, extensible software platform includes integrated tools for AIOps, NetOps, SecOps, DevOps, and Internet of Things (IoT) connectivity with AI/ML technology integrated throughout. Functionality this complete could be achieved before now only through the purchase and operation of multiple third-party software tools.

4.4.2 Benefits

- **Simplify management.** Operate your local and branch networks over a centralized dashboard.
- **Increase security.** Translate business intent into zero-trust policies and dynamic segmentation of endpoints based on usage behavior.
- **Lower costs.** Policy-driven provisioning and guided remediation increase network uptime and reduce time spent managing network operations.
- **Transform your network.** Deploy cloud services and applications that benefit from the intelligent network optimization delivered by Cisco DNA Center.
- **Ensure network and application performance:** AI/ML network insights reduce time spent managing network operations and improve user experience.
- **Facilitate offsite IT teams:** Optimized for remote access, a clean, organized dashboard with single-button workflows makes remote management easy.

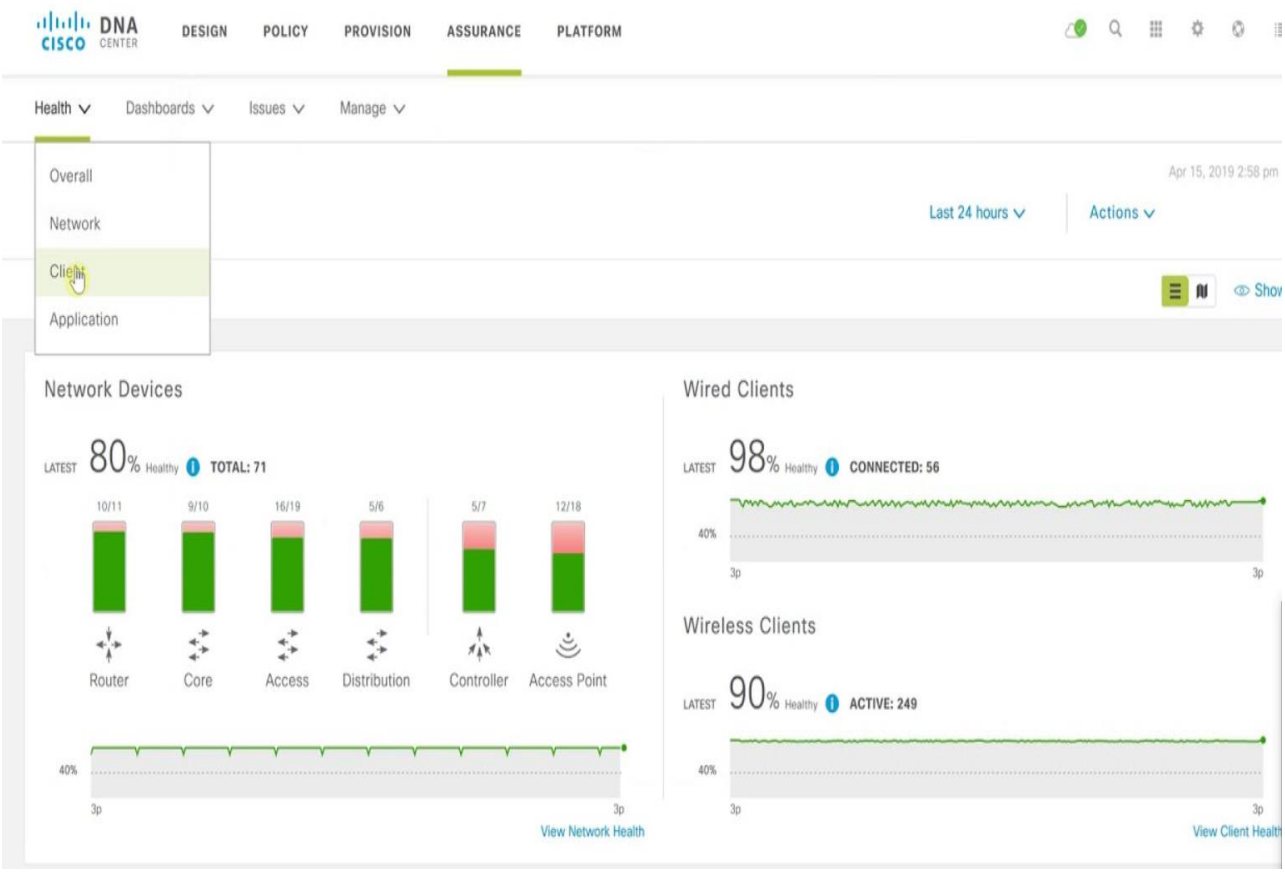

Cisco DNA Center
physical and virtual infrastructure
Cisco and third party

4.5 Properties of CISCO DNA

Cisco DNA Center offers a single dashboard for every core function in your network. With this platform, IT can become more nimble and respond to changes and challenges faster and more intelligently.

Cisco DNA Center is the network management system, foundational controller, and analytics platform at the heart of Cisco's intent-based network. Beyond device management and configuration, Cisco DNA Center is a set of software solutions that provide:

- A management platform for all of your network
- An intent-based networking controller for automation of your policies, segmentation, and services configurations
- An assurance engine to guarantee the best network experience for all your users

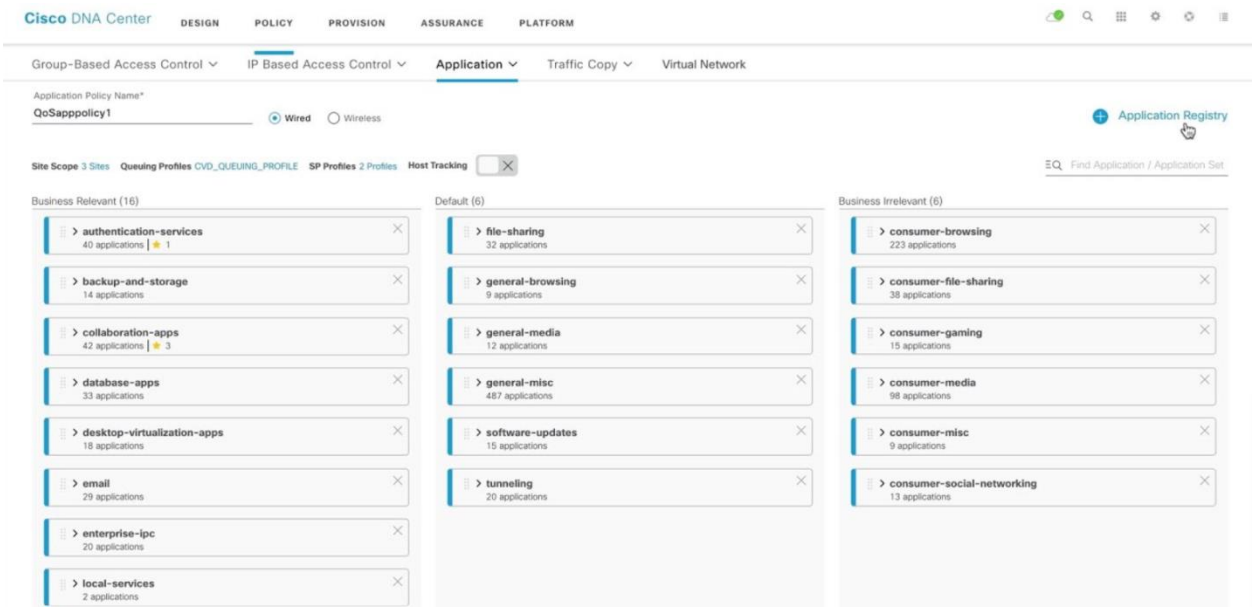


CISCO DNA DASHBORD

There is Features of DNA center by the Dashboard we can configure the devises and monitoring the performance and through the dashboard we can do and the main Dashboard consist of:-
(DESIGN,POLICY,PROVISION,ASSURANCE,PLATFORM)

Design: Design your network for consistent configurations by device and by site. Physical maps and logical topologies help provide quick visual reference. The direct import feature brings in existing maps, images, and topologies directly from Cisco Prime Infrastructure, making upgrades easy and quick. Device configurations by site can be consolidated in a "golden image" that can be used to automatically provision new network devices. These new devices can either be prestaged by associating the device details and mapping to a site, or they can be claimed upon connection and mapped to the site.

Policy: Translate business intent into network policies and apply those policies, such as access control, traffic routing, and quality of service, consistently over the entire wired and wireless infrastructure. Policy-based access control and network segmentation is a critical function of the Cisco Software-Defined Access (SD-Access) solution built from Cisco DNA Center and Cisco Identity Services Engine (ISE). Cisco AI Network Analytics and Cisco Group-Based Policy Analytics running in the Cisco DNA Center identify endpoints, group similar endpoints, and determine group communication behavior. Cisco DNA Center then facilitates creating policies that determine the form of communication allowed between and within members of each group. ISE then activates the underlying infrastructure and segments the network, creating a virtual overlay to follow these policies consistently. Such segmenting implements zero-trust security in the workplace, reduces risk, contains threats, and helps verify regulatory compliance by giving endpoints just the right level of access they need.



Provision: Once you have created policies in Cisco DNA Center, provisioning is a simple drag-and-drop task. The profiles (called scalable group tags or "SGTs") in the Cisco DNA Center inventory list are assigned a policy, and this policy will always follow the identity. The process is completely automated and zero-touch. New devices added to the network are assigned to an SGT based on identity—greatly facilitating remote office setups.

Assurance: Cisco DNA Center assurance capabilities, use AI/ML enabling every point on the network to become a sensor, sending continuous streaming telemetry on application performance and user connectivity in real time. The clean and simple dashboard shows detailed network health and flags issues. Then, guided remediation automates resolution to keep your network performing at its optimal level, reducing mundane troubleshooting work. The outcome is a consistent experience and proactive optimization of your network, with less time spent on troubleshooting tasks.

Automation: Speeds and simplifies the deployment of new software images and patches. Pre-and post-checks help prevent adverse effects from an upgrade, Enables deployment of new devices in minutes and without onsite support visits. Eliminates repetitive tasks and staging, Saves time in setting up network virtual services. Supports existing branch migration without hardware upgrades.

Why Cisco?

You need a network that is constantly learning, constantly adapting, and constantly protecting. This is the future of networking. With our deep understanding of technology and relationships with IT, Cisco can help bring the boardroom and your IT together to work effectively toward better outcomes for IT and the business. With Cisco DNA Center, we can help you create revenue opportunities, lower costs, reduce risks, and ensure regulatory compliance. And we can help you simplify your network operations and accelerate their response to changing IT and business needs. Together with our partners, we help you innovate, manage market transitions, and turn technology into business advantage.

The Summary

We were talking about DNA and the big benefit of the controller inside your network that make the tasks easy and easy monitoring and we was take overview about the amazing Dashboard.

We hope to explain everything in a right way and there is many version of DNA center depends on the size of your network on another example DNA version 2.2.3.x and 2.1.2.x and 2.2.2.x the best way to know more about the version go to cisco website or go to our team website.