الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Mention Électronique
Spécialité Système des télécommunications

présenté par

DJEZZAR MARIA

&

BELHARBI CHEMS IKHLASS

Programmation du réseau avec Python et les APIs REST du contrôleur SDN CISCO APIC-EM

Proposé par : Mr.BENZIANE Mohamed Amine et Mr.AIT SAADI Hocine

Année Universitaire 2019-2020

Remerciement

Merci à Dieu de sa grâce, source de notre force et courage tout au long de nos études universitaires.

Nous tenons à exprimer nos vifs remerciements et notre profonde gratitude à Mr. Benziane Mohamed Amine pour son aide, son encouragement et de nous avoir dirigés dans notre travail.

Nous remercions également Mr. Ait Saadi Hocine de nous avoir encadrés dans notre mémoire de fin d'études.

Nous remercions les membres du jury d'avoir accepté d'évaluer notre travail.

Un grand merci également à notre famille pour leur soutien aussi bien moral que financier et pour leurs sacrifices

Nous ne pourrons terminer sans remercier tous ceux qui ont participé d'une manière ou d'une autre dans l'élaboration de ce projet de fin d'études.

Dédicace :

C'est avec profonde gratitude et des mots sincères, que je dédie ce modeste travail de fin d'études à ma mère

Qui a sacrifié sa vie pour ma réussite et m'a éclairé le chemin par ses conseils judicieux.

J'espère qu'un jour, je pourrais lui rendre un peu de ce qu'elle a fait pour moi que Dieu lui prête bonheur et longue vie ainsi a tres cher père.

Je dédie aussi ce travail à ma moitié, ma très chère sœur « Sara » qui m'a soutenue toute ma vie.

À toute ma famille

À tous mes amis et mes collègues

Enfin, toute personne qui m'aime et que j'aime

MARIA

Dédicace :

À mes chers parents

Aucune dédicace ne serait exprimée mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance.

À mes très chers frères « Amine » et « Karim »

À toute ma famille

À tous mes amis et mes collègues

Et à tous ceux qui me sont chers

CHEMS

ملخص:

ظهرت اليوم اتجاهات جديدة في مجال تكنولوجيا المعلومات, ونتيجة ذلك انفجرت معدات شبكات الكمبيوتر والاتصالات، الأمر الذي تطلب اللجوء إلى الشبكات المعرفة عن طريق البرمجة. يعتبر ال التطور الذي يمكن من خلاله SDN برمجة الشبكة والذي يسمح بتطور الأجهزة والبرامج بطريقة مستقلة سريعة وسهلة ,التحكم .APIC المفتوحة (EM CISCO ، تقدم مقاربة مفتوحة وبرمجة لشبكة الشركة، و ال- WAN وحدة باستخدام واجهات برمجة التطبيقات . الذي)API للأمن والإدارة القائمة على السياسات. هذه المقاربة توفر أدوات فعالة لمراقبة الشبكة و جعل براعة لغة البرمجة بايثون و واجهات برمجة التطبيقات REST يكون عادة مرهق، التكوين الي بدل التكوين اليدوي يجعلها مثالية لمختلف التطبيقات، بما في ذلك استخراج البيانات، إنترنت الأشياء، المحاكاة سحابة.

كلمة مفتاحية : APIC API REST, SDN ، بايثون، EM-

Résumé : De nos jours, des nouvelles tendances arrivent dans le monde informatique : le résultat est une explosion des équipements réseau et des communications, ce qui nécessite de se tourner vers le SDN. Le Software Defined Network (SDN) est vue comme une évolution qui permet de contrôler et programmer le réseau, et d'évoluer le matériel et le logiciel d'une façon indépendante, rapide et facile. Le module APIC-EM de CISCO offre une approche ouverte et programmable du réseau d'entreprise, campus et WAN par le biais d'API ouvertes pour une sécurité et une gestion basées sur des politiques. Cette approche fournit des outils efficaces de surveillance du réseau et permet d'automatiser les tâches de configuration manuelles, généralement, fastidieuses. La polyvalence du langage de programmation Python et les API REST le rend idéale dans des applications variées y compris l'exploration de données, Internet of Things et la simulation de cloud.

Mot clé: SDN, APIC-EM, Python, API REST.

Abstract :

Nowadays, new trends are happening in computing: the result is an explosion of networking and communications equipment, which requires turning to the SDN.

The Software Defined Network (SDN) is seen as an evolution that allows to control and program the network, to evolve hardware and software in an independent, fast and easy way. The APIC-EM module of CISCO offers an open and programmable approach to the corporate network, campus and WAN through open APIs for policy-based security and management. This approach provides effective network monitoring tools and automates manual configuration tasks, typically tedious. The versatility of the python programming language and the REST APIs makes it ideal for a variety of applications, including data mining, Internet of Things, cloud simulation.

Key words: SDN, APIC-EM, Python, API REST.

Liste abréviations

À

- ACI: Application Centric Infrastructure
- API: Application Programming Interface
- APIC: Application Policy Infrastructure Controller
- APIC-EM: Application Policy Infrastructure Controller-Enterprise Module
- ARP: Address Resolution Protocol
- ASON: Automatically Switched Optical Network

B

- BGP: Border Gateway Protocol
- BGP-EVPN: BGP- Ethernet VPN
- BGP-LS: BGP-Link State

C

- CLI: Command Line Interface
- CUCM: Cisco Unified Communications

D

- DC: Data Center
- DHCP: Dynamic Host Configuration Protocol
- DWDM: Dense Wavelength Division Multiplexing

E

ECMP: Equal-Cost Multi-Path

G

- GMPLS: Generalized Multi-Protocol Label Switching
- GUI: Graphical User Interface

H

- HA: High Avaibility
- HTTP: Hypertext Transfer Protocol
- HTTPS: Hypertext Transfer Protocol Secure

I

- IETF: Internet Engineering Task Force
- IGP: Interior Gateway Protocol
- IP: Internet Protocol
- IPsec: Internet Protocol Security
- IS-IS: Intermediate System to Intermediate System
- ITU: International Telecommunication Union
- IWAN: Intelligent Wide Area Network

J

- JSON: JavaScript Object Notation

L

- LAN: Local Area Network
- LSP: Label Switched Paths
- LTE: Long Term Evolution

M

- MAC: Media Access Control
- MP-BGP EVPN: Multi-Protocol BGP Ethernet VPN
- MPLS: Multi-Protocol Label Switching

- MS Project : MicroSoft Project
- MIB : Management Information Base

N

- NB: NorthBound
- NETCONF: NETwork CONFiguration Protocol
- NFV: Network Functions Virtualization
- NGN : Next Génération Network

O

- ODL: OpenDay Light
- OIF: Optical Internetworking Forum
- ONF: Open Networking Foundation
- ONOS: Open Network Operating System
- OPEX: OPerational EXpenditure
- OSPF: Open Shortest Path First
- OSPF-TE: OSPF-Traffic Engineering
- OTN: Open Transport Network
- OVSDB: Open Virtual Switch DataBase

P

- PaaS: Platform As A Service
- PCC: Path Computation Client
- PCE: Path Computation Element
- PCEP: Path Computation Element Protocol
- PERT : Program Evaluation and Review Technique
- PfR: Performance Routing
- PKI: Public Key Infrastructure
- PnP: Plug and Play

Q

- QoS: Quality of Service

R

- REST: REpresentational State Transfer
- RESTConf: REST Configuration
- ROADM: Reconfigurable Optical AddDrop Multiplexer ROI : Return On Investment

S

- SB: SouthBound
- SDH: Synchronous Digital Hierarchy
- SDN: Software Defined Network
- SNMP: Simple Network Management Protocol
- SOAP: Simple Object Access Protocol
- SONET: Synchronous Optical NETwork
- SSH: Secure SHell

T

- TCP: Transmission Control Protocol
- TED: Traffic Engineering Database
- TLS: Transport Layer Security
- T-SDN: Transport-SDN
- TTL: Time To Live

U

- URI: Uniform Resource Identifier
- URL: Uniform Resource Locator

V

- VLAN: Virtual Local Area Network
- VoIP : Voice Over IP
- VTEP : VxLAN Tunneling EndPoint
- VTS: Virtual Topology System
- VXLAN: Virtual EXtensible LAN

W

- WAAS: Wide Area Application Services
- WAN: Wild Area Network
- WDM: Wavelength Division Multiplexing
- WSDL: Web Services Description Language
- WSS: Wavelength Selective Switch
- WWW: World Wide Web

X

- XML: eXtensible Markup Language

Y

- YANG: Yet Another Next Generation

Z

ZTD: Zero Touch Deployment

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

Historiquement, les réseaux d'entreprise ont été construits sur des périphériques dans lesquels le matériel et les logiciels étaient étroitement couplés. Aujourd'hui, l'une des principales approches informatiques, pour une évolution des réseaux, est le Software Defined Network (SDN), dans laquelle le logiciel et le contrôle sont extraits du matériel. Cela permet au réseau d'être plus ouvert et conscient des applications, avec une centralisation de contrôle.

L'automatisation du centre de données a débuté il y a plusieurs années. On peut faire apparaître une machine virtuelle en quelques minutes. Cisco a développé APIC, principalement, pour les centres de données et cloud. Par la suite, elle a innové dans les réseaux d'entreprises, campus et WAN en adoptant une approche similaire au centre de données basée sur les politiques appelées APIC-EM. Cette solution est venue pour améliorer la productivité des réseaux et introduire la notion de l'automatisation au réseau WAN (Wild Area Network), entreprise et campus.

APIC-EM permet le déploiement et la configuration des réseaux très complexes d'une façon facile, rapide et efficace, à travers une approche basée sur des politiques, afin de pouvoir passer à un niveau d'automatisation nouveau, qui soit basé sur des politiques. Généralement, lorsqu'on parle d'un dispositif individuellement, sa configuration se fait ligne par ligne, d'où la mise en place du réseau qui se fait de bas en haut.

APIC-EM dispose également de nombreuses applications et fonctionnalités pour la manipulation et l'automatisation de réseaux. Comme il fournit ses REST API interactifs en direction du nord ce qui donne plus de flexibilité.

Algérie Télécom, en tant que fournisseur d'accès historique et dans le but d'automatiser la construction et la manipulation de son réseau d'une part, et de suivre

l’évolution technologique d’autre part a choisi la solution de SDN CISCO APIC-EM pour le déploiement de son réseau. Malgré les applications disponibles pour la capture et le suivi des paquets, les administrateurs réseau préfèrent utiliser leurs propres logiciels développés par eux même, au lieu d’utiliser les applications tierces pour des raisons de sécurité, la confidentialité et l'intégrité.

Aussi, de donner plus de flexibilité et de richesse aux fonctions hautement sécurisées qui peuvent fournir un contrôle programmable facile à utiliser pour les éléments du réseau, interfaces et hôtes.

L’objectif de notre étude est d’interroger le contrôleur APIC-EM à travers les API REST et un script écrit en langage Python afin de pouvoir récupérer les différentes informations du réseau (informations sur les périphériques réseau, les hôtes, la topologie …) et avoir une visibilité totale. Ce qui facilite le suivi du réseau.

Pour ce faire, nous avons organisé notre mémoire en trois chapitres détaillés comme suit : dans le premier chapitre, nous allons présenter des généralités sur le SDN ainsi que les principaux protocoles utilisés dans cette technologie du côté sud (Southbound Protocol .

Le deuxième chapitre sera consacré à la présentation du contrôleur de CISCO APIC-EM ainsi que ses différentes applications et fonctionnalités. Tandis que, le troisième et dernier chapitre présentera en premier lieu les différentes méthodes d’interagir avec le contrôleur.

Par la suite, nous allons aborder les étapes avec une description détaillée du script utilisé. Nous présenterons aussi, les principaux résultats obtenus, nous terminons notre travail par une conclusion évoquant quelques perspectives.

Chapitre I : Introduction générale au Software Defined Network

I.1 Introduction

SDN Software Defined Network est une nouvelle approche du monde actuel des réseaux qui a bouleversé les approches traditionnelles. dans ce chapitre nous allons répondre aux différentes questions qui se posent : qu'est-ce que le SDN ? Quelle est la différence entre SDN et approches traditionnelles ? Quels sont son principe et ses origines ? Et sur tout quels sont ses protocoles ?

I.2 Définition SDN

SDN signifie littéralement Software Defined Networking, selon l'ONF (open networking foundation), le SDN n'a pas de définitions uniques.

Le SDN est une architecture émergente à la fois dynamique, facilement gérable, rentable et évolutive, idéalement adaptée à la nature dynamique et aux bandes passantes élevées associées aux applications modernes.

La définition académique consiste à voir le SDN comme une architecture qui découple les fonctions de contrôle et de transfert des données du réseau afin d'avoir une infrastructure physique complètement exempte de tout service réseau.

I.3 L'origine du SDN

Le terme SDN a été inventé à l'origine pour représenter les idées et le travail autour d'OpenFlow à l'université de Stanford, USA. Tel que défini à l'origine, le SDN se réfère à une architecture du réseau ou l'état de transmission dans le Data plane est géré à distance par un contrôle plane découplé du précédent. L'industrie réseau s'est à maintes reprises éloignée de cette vision originale du SDN en qualifiant de SDN tout ce qui concerne les logiciels [1].

I.4 Principe de SDN

Le principe du SDN est simple :

- Séparer le plan de contrôle du plan de transmission
- Centraliser et mutualiser le plan de contrôle entre tous les équipements
- Réduire les équipements à leur fonction la plus élémentaire : transmettre (ou ne pas transmettre) des données sur le réseau
- Les serveurs chargés de fournir le plan de contrôle de la solution SDN sont généralement appelés contrôleurs SDN [2].

I.5 Architecture SDN

Le SDN présente une architecture réseau ou le plan de contrôle est totalement découplé du plan de données, cela est illustré par la figure suivante [3] :

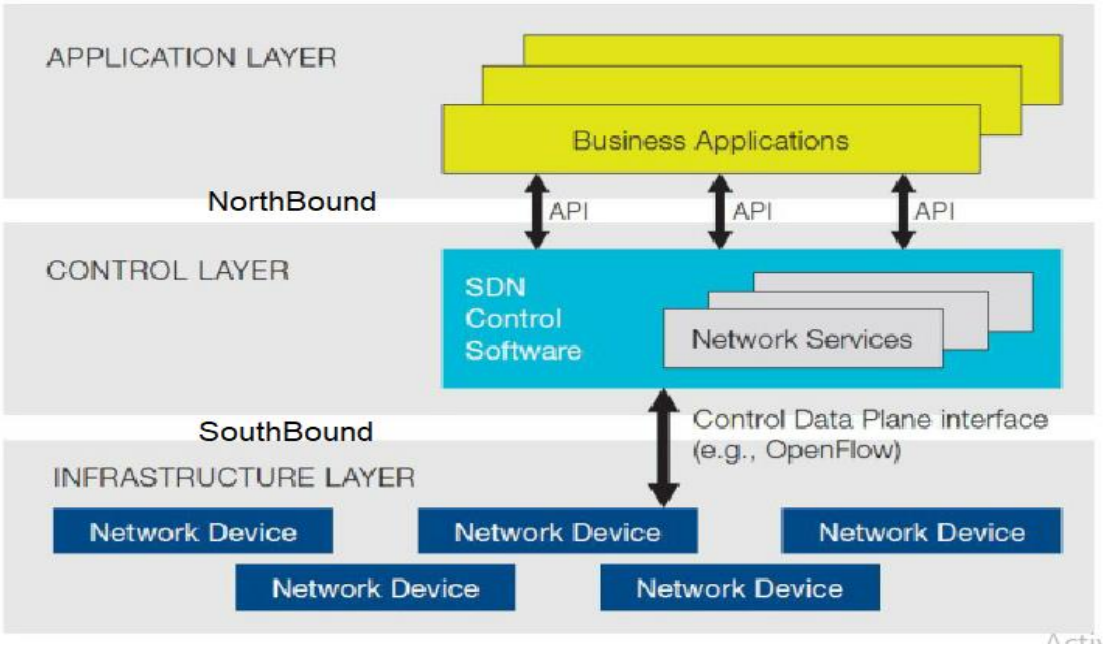


Figure 1.1 : Architecture SDN

Un réseau traditionnel est composé généralement des équipements d'interconnexions tels que des Switchs et des routeurs. Ces équipements incorporent à la fois la partie transmission et la partie de contrôle de réseau. Dans ce modèle

d’architecture, il est difficile de développer de nouveaux services, en raison du fort couplage qui existe entre le plan de contrôle et le plan de transmission.

I.6 Séparation du réseau

Afin d’ouvrir les équipements réseau aux innovations, l’architecture SDN, a vu le jour. Elle permet de découpler la partie de contrôle de la partie transmission des équipements d’interconnexions. Le SDN est composé principalement de trois couches et d’interfaces de communication [3].

I.6.1 La couche d’infrastructure (la couche transmission)

Cette couche est appelée aussi « Plan de données », c’est la couche la plus basse, elle contient les équipements de transmission (FEs : Forwarding Elements) tels que les switch physique et virtuels. Son rôle principal est l’acheminement du trafic et qui supporte le protocole OpenFlow qu’ils partagent avec le contrôleur [3].

I.6.2 La couche de contrôle

Cette couche est appelée aussi « Plan de contrôle », elle est constituée principalement d’un ou plusieurs contrôleurs SDN, son rôle est de contrôler et de gérer les équipements de l’infrastructure à travers une interface appelée « south-bound API » [3].

I.6.3 La couche application

Elle représente les applications qui permettent de déployer de nouvelles fonctionnalités réseau, comme l’ingénierie de trafic, QoS (Quality of service), la sécurité, configuration dynamique, etc. Ces applications sont construites moyennant une interface de programmation appelée « north-bound AP » [3].

I.7 Interfaces de communications

Il existe principalement deux types d’interfaces qui permettent aux contrôleurs de communiquer avec leur environnement : interface Sud, Nord [3].

I.7.1 Interface du sud (SouthBound API)

Elle est utilisée pour la communication entre le contrôleur SDN et les équipements réseau. Ces API sont de plusieurs types par exemple OpenFlow, BGP-LS (BGP-Link State), PCEP (Path Computation Element Protocole), NETCONF (NETwork CONFiguration), SNMP, CLI, etc [3].

I.7.2 Interface du nord (NorthBound)

Cette interface est dirigée vers les applications (d'où l'utilisation de l'orientation nord). Elle sert à rajouter des fonctionnalités, à déployer et configurer des services. Ses APIs sont basées sur REST (REpresentational State transfer), RESTCONF (REST CONFiguration)[3].

I.8 Les protocoles SDN

I.8.1 Openflow

OpenFlow est un protocole de lien entre le plan de contrôle et le plan de données. L'échange de messages se fait au cours d'une session TCP (Transmission Control Protocol) établie via le port 6633 du serveur contrôleur [4].

Openflow est donc une composante du SDN, son développement a commencé en 2007 dans le cadre d'une collaboration entre les mondes de l'université et des affaires. Établie à l'origine par l'université de Stanford et l'université de Californie à Berkeley [4].

Un commutateur compatible avec le protocole OpenFlow (Commutateur OpenFlow) sépare les deux fonctions classiques (control et data plane) de telle façon que la partie plan de données réside toujours dans le commutateur, tandis que la partie contrôle est déplacée vers un contrôleur distinct ; généralement, un serveur standard. Le commutateur OpenFlow et le contrôleur communiquent via le protocole OpenFlow [5].

I.8.2 Contrôleur

Le concept de contrôleur est né parallèlement à l'introduction de la mise en réseau définie par logiciel (SDN) elle-même.

SDN sépare le contrôle de la transmission des données du réseau ; tout ce contrôle est centralisé dans une logique délivrée via une application logicielle, c'est pourquoi il est souvent appelé le « cerveau » du réseau. En tant que point de contrôle central, le contrôleur peut simplifier et automatiser l'orchestration du réseau, pour améliorer l'intelligence, l'agilité, l'évolutivité et la rentabilité de leur infrastructure globale [6].

Le contrôleur utilise le protocole OpenFlow pour connecter et configurer les fonctions réseau (routage, QoS, etc.) dans les périphériques afin de déterminer le meilleur chemin pour le trafic applicatif [7].

(a) Les Contrôleurs communs d'OpenFlow

Le contrôleur SDN permet d'implémenter un changement sur le réseau en traduisant une demande globale en une suite d'opérations sur les équipements réseau (ajouts d'états Openflow, configuration en CLI…), les ordres sont donnés au contrôleur par une application via une API dite « Northbound » ou nord. Le contrôleur communique avec les équipements via une ou plusieurs API dites « Southbound » ou sud. Openflow se positionne comme une API sud agissant directement sur le plan de données, il existe plusieurs contrôleurs SDN, tel que [7] :

- **NOX** Initialement développé chez Nicira, est le premier contrôleur OpenFlow. C'est un Open-source et écrit en C ++. Il est actuellement à la baisse: il n'y a pas eu de changements majeurs depuis mi 2012 [8].
- **POX** est le plus jeune frère de NOX. C'est un contrôleur open-source écrit en Python, et comme NOX, fournit un cadre pour le développement et le test d'un contrôleur OpenFlow, mais les performances POX sont nettement inférieures à celles des autres contrôleurs et ne convient donc pas au déploiement d'entreprise [8].
- **Beacon** est un contrôleur Java connu par sa stabilité. Il a été créé en 2010 et est toujours maintenu, il a été utilisé dans plusieurs projets de recherche. En raison de ses performances, c'est une solution fiable pour l'utilisation dans des

conditions réelles. Ce contrôleur a également été utilisé dans d'autres projets tels que Floodlight ou OpenDaylight [8].

- **Floodlight** est un contrôleur open-source OpenFlow basé sur Java, pris en charge par BigSwitch Networks. Il est sous licence Apache [9]. Il est facile à configurer et à montrer aussi de grandes performances. Avec toutes ses fonctionnalités, Floodlight est plus une solution complète [8].
- **OpenDaylight** est un projet de la Fondation Linux pris en charge par l'industrie. C'est un framework open source pour faciliter l'accès au logiciel de définition de réseau (SDN). Comme Floodlight, il peut également être considéré comme une solution complète [8].

Le tableau ci-dessous présente les contrôleurs les plus connus en général

Contrôleur	Organisation	Langage	Fonctionnalités
NOX	Nicira	C++	le premier contrôleur openflow
POX	Nicira	Python	améliorer les performances de NOX
Beacon	Stanford	Java	basé sur le Multithreading
Floodlight	Big Switch	Java	testé avec des commutateurs OpenFlow physiques et virtuels.
Opendaylight	Linux Foundation	Java	supporte le Framework OSGi et le REST API

Tableau 1.1 : les contrôleurs les plus connus

(b) Commutateur (switch) OpenFlow

Un commutateur OpenFlow est un commutateur de données compatible OpenFlow qui communique via le canal OpenFlow à un contrôleur externe. Il effectue la recherche et le transfert de paquets selon une ou plusieurs tables de flux et une table

de groupe. Le commutateur OpenFlow communique avec le contrôleur et le contrôleur gère le commutateur via le protocole de commutateur OpenFlow [10].

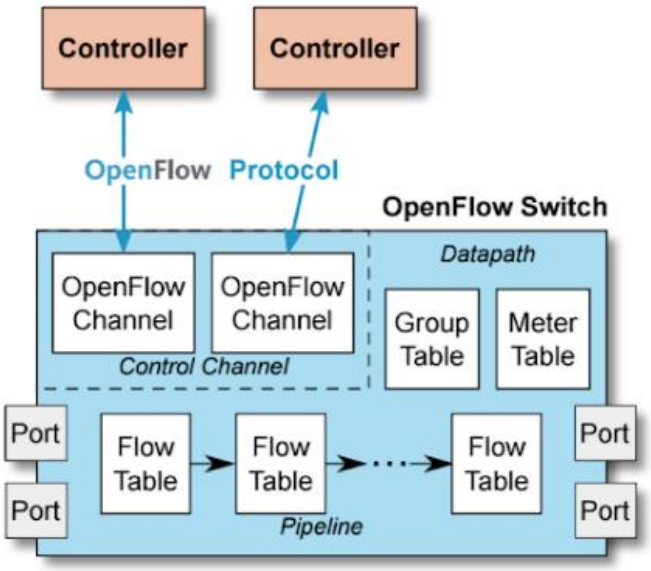


Figure 1.2: Commutateur Openflow (OpenFlow switch)

On distingue deux types de switch OpenFlow [11]:

- Switch OpenFlow-only : supporte uniquement les actions requises pour le fonctionnement du protocole OpenFlow[11].
- Switch OpenFlow-enabled : en plus des actions requises pour le protocole OpenFlow, ce switch supporte les actions d'un switch ordinaire [11].

I.8.3 Structure d'un commutateur OpenFlow

Les commutateurs OpenFlow contiennent des tables de flux qui sont utilisées pour effectuer des fonctions de transfert indiquées dans les en-têtes de paquets [12]. À l'aide de la table de flux, une des actions suivantes est exécutée :

1) Relayer le paquet sur un port de sortie,
2) Supprimer le paquet,
3) Passer le paquet au contrôleur. Le paquet est encapsulé dans un message OpenFlow PACKET_IN.

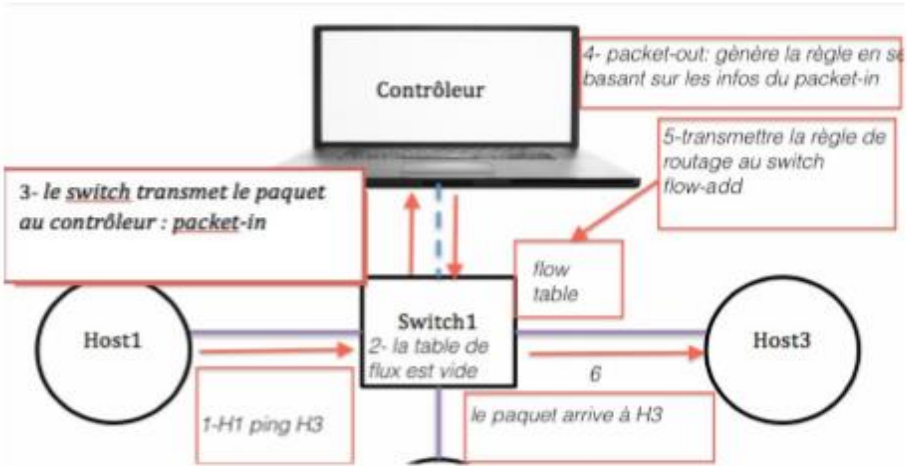


Figure 1.3: Diagramme de flux des messages OpenFlow.

I.8.4 Table de flux

Une table de flux "Flow Table" est composée de plusieurs entrées de flux, chacune est structurée comme suit [11] :

Champ de correspondance	Instructions	Compteurs

- **Champ de correspondance (Match fields)** : Utilisé lors de la recherche de l'entrée correspondante au paquet. Ils sont constitués essentiellement des entêtes des paquets et des ports d'entrées [11].
- **Compteurs (Counters)** : Servent essentiellement à garder des statistiques sur les flux pour ensuite décider si une entrée de flux est active ou non . Pour chaque table, chaque flux, chaque port, des compteurs de statistiques sont maintenus [11].
- **Instructions :** Représentent l'ensemble des instructions OpenFlow qui servent à modifier le traitement que va subir le paquet. Les instructions supportées sont [11] :

1) Apply-Actions : Pour appliquer les actions sur le paquet immédiatement.
2) Clear-Actions : Pour supprimer une liste des actions du paquet.
3) Write-Actions : Ajouter une liste d'actions au paquet.

4) Write-Metadata : Ajouter des données utiles pour le séquencement entre les tables OpenFlow.
5) Goto-Table : Indique que le paquet doit être acheminé vers une table d'indice supérieur.

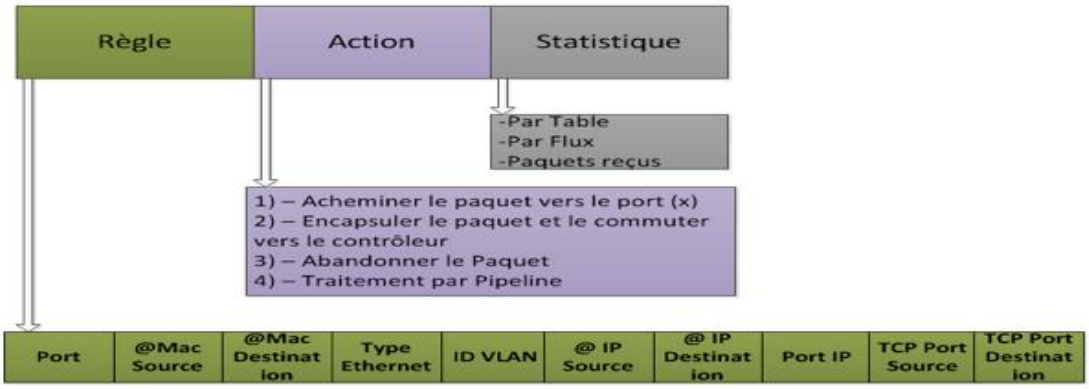


Figure 1.4 : La table de flux

Les instructions d'une entrée de flux peuvent explicitement diriger le paquet vers une autre table de flux "Flow Table" via l'instruction "Goto". Si le paquet ne correspond à aucune entrée de flux, le comportement du switch vis-à-vis ce paquet dépendra de la configuration de la table. Le comportement par défaut est d'envoyer le paquet au contrôleur avec un message PACKET_IN. La table peut aussi spécifier que dans ce cas, le paquet doit continuer son chemin vers la table suivante Le contrôleur répondra généralement par un PACKET_OUT donnant l'instruction à suivre [13].

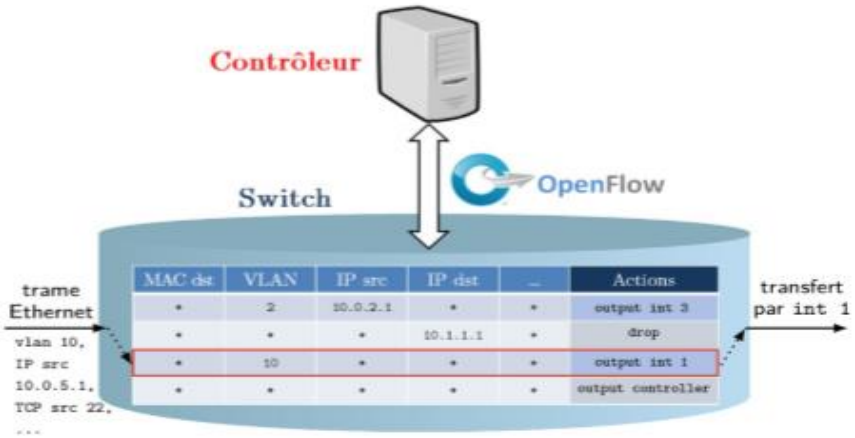


Figure 1.5 : Schématisation du modèle de flux OpenFlow

La figure 1.5 illustre le traitement d'un paquet parvenant à un switch OpenFlow contenant une seule table de flux : parmi les différents champs d'en-têtes de la trame, l'appartenance au VLAN 10 fait correspondre cette trame à un des flux du switch, dont l'action associée consiste en un transfert par l'interface [13].

I.8.5 Les Tables de groupe

La table de groupe contient des entrées, et chaque entrée une liste d'actions appelée Conteneur D'actions les actions d'un ou plusieurs conteneurs d'actions sont appliquées sur les paquets envoyés au groupe. Chaque entrée dans la table de groupe contient [14].

Identifiant de Groupe	Type de Groupe	Compteurs	Conteneurs d'actions

- L'identifiant de groupe : c'est un entier de 32 bits.
- Le Type de groupe : sert à déterminer le type du groupe 'All – Select – Indirect – Fast Failover .
- Les compteurs : mis à jour quand un paquet est traité par un groupe.
- Conteneur d'action : un ensemble d'actions et de paramètres associés, qui sont définis pour les groupes [14].

I.8.6 Protocole Openflow

(a) Messages OpenFlow

Le protocole OpenFLow supporte trois types de messages. Messages contrôleur vers Switch, messages asynchrones et messages symétriques, chaque type a une sous-catégorie [14].

- Les Messages depuis le Contrôleur vers Switch

Sont initiés par le contrôleur, ils servent à gérer ou vérifier l'état du switch, ces types de messages peuvent ou non demander une réponse de la part du switch [14].

- Features : lors du Handshake le contrôleur peut demander l'identité et les capacités d'un switch en envoyant une requête.
- Modify-State : Ce type de message est envoyé pour gérer l'état dans les Switch. Sa fonction primaire est d'ajouter, modifier ou effacer les entrées dans les tables OpenFlow Flow/Groupe.
- Read-state : Ces messages sont utilisés par le contrôleur pour collecter différentes informations du switch, comme sa configuration actuelle, des statistiques et des capacités.
- Packet-Out : sont utilisés pour transférer les paquets reçus par les messages. Packet-In. Soit ils contiennent le paquet en entier soit l'id du buffer faisant référence au paquet stocké dans le switch. Ils doivent contenir aussi une liste d'actions à appliquer, s'il n'y a pas d'action définie le paquet sera détruit.

➢ Les Messages asynchrones

Les messages asynchrones sont envoyés par le switch vers le contrôleur pour indiquer un changement d'état ou l'arrivé d'un paquet [14].

- Packet-In : avec ce type de message, le switch transfère le contrôle du paquet au contrôleur.
- Flow-removed : informe le contrôleur de la suppression d'une entrée dans la table de Flux
- Port-Status : Informe le contrôleur d'un changement sur un port du switch.

➢ Les messages symétriques

Les messages symétriques sont envoyés sans aucune sollicitation ni du switch ni du contrôleur [14].

- Echo : ont comme utilité la vérification de la connectivité entre switch et contrôleur.
- Hello : ces messages sont échangés entre les deux une fois la connexion établie.
- Error : Utilisé pour signaler de part et d'autre des problèmes de connexion.

(b) Canal OpenFlow (OpenFlow Channel)

Le canal OpenFlow est l'interface qui connecte chaque commutateur OpenFlow à un contrôleur. Cette interface permet au contrôleur de recevoir les messages du commutateur et de pouvoir le gérer à travers le réseau. Le canal doit être sécurisé afin d'assurer le bon déroulement des communications entre le commutateur et le contrôleur. Pour cela, l'échange de message se fait au cours d'une session TCP (Transmission Control Protocol) établie via le port 6653 du serveur contrôleur ou à travers une connexion SSL/ TLS (Secure Sockets Layer / Transport Layer Security) [15].

I.9 NETCONF ET YANG

I.9.1 NETCONF

Le protocole de configuration réseau NETCONF (Network Configuration Protocol) est un protocole de gestion de réseau développé et normalisé par l'IETF (Internet Engineering Task Force). Il a été développé dans le groupe de travail NETCONF et publié en décembre 2006 puis révisé en juin 2011 et publié sous le numéro RFC 6241 [16].

NETCONF est conçu pour couvrir les insuffisances de « Simple Network Management Protocol » (SNMP) et Command-Line Interface (CLI), dans les fonctions de configurations réseau. Le protocole prévoit des mécanismes d'installation, manipulation, et suppression de la configuration des périphériques réseau. Il utilise un langage de balisage extensible (XML) (eXtensible Markup Language) ; codant les données de configuration ainsi que les messages du protocole [17].

Le protocole NETCONF utilise un appel de procédure distante (RPC). Un client encode un RPC (Remote Procedure Call) en XML et l'envoie à un serveur utilisant une méthode de connexion sécurisée. Le serveur répond avec une réponse codée au format XML [17].

I.9.2 YANG

YANG (Yet Another Next Generation) est le langage de modélisation des données du protocole de configuration des réseaux NETCONF [17].

Ce langage de modélisation est actuellement développé par le groupe NETMOD de l'organisme IETF (Internet Engineering Task Force). Il peut être utilisé pour modéliser les données de configuration ainsi que les données d'état des éléments du réseau. En outre, YANG peut être utilisé pour définir le format des notifications d'événements émis par les éléments de réseau et permet de définir la signature d'appels de procédure distante qui peut invoquer des éléments de réseau via le protocole NETCONF [17].

YANG est un langage modulaire représentant des structures de données dans un format d'arbre XML. Le langage de modélisation des données est livré avec un certain nombre de types de données intégrés. D'autres types de données d'application spécifiques peuvent être dérivés des types de données intégrés. En plus des structures de données complexes réutilisables peuvent être représentées comme des groupements. Données YANG modèles peuvent utiliser des expressions XPath pour définir des contraintes sur les éléments de YANG un modèle de données [17].

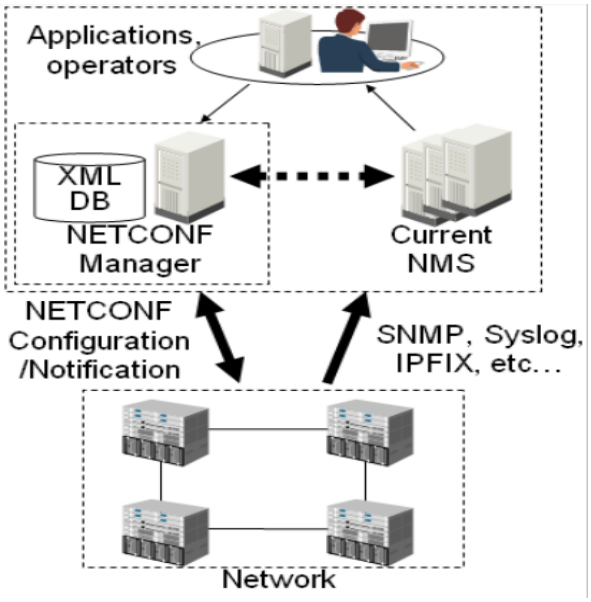


Figure 1.6 : NETCONF Configuration/ Notification

I.10 BGP-LS et PCEP

Le mouvement Software Defined Networking (SDN) offre de plus grands avantages pour rendre les réseaux plus personnalisés, efficaces, centrés sur les applications et programmables. Il existe de nombreuses approches différentes pour créer SDN. Les protocoles OpenFlow et VXLAN gagnent en popularité dans les environnements de datacenter, tandis que le routage de segments et PCEP / BGP-LS sont l'option technologique SDN des opérateurs pour des raisons évidentes. Contrairement au routage de segments OpenFlow / VXLAN et PCEP / BGP-LS, ces solutions basées sur des protocoles peuvent être mises en œuvre en effectuant une mise à niveau logicielle [18].

- Qu'est-ce que le BGP-LS ?

BGP Link-State (LS) est un identificateur de famille d'adresses (AFI) et un identificateur de sous-adresse de famille (SAFI) définis pour transporter la base de données d'état des liens IGP (Interior Gateway Protocol) via BGP. BGP-LS fournit des informations sur la topologie du réseau aux serveurs de topologie et aux serveurs ALTO (Application Layer Traffic Optimization). BGP-LS permet un contrôle basé sur des règles d'agrégation, de masquage d'informations et d'abstraction. [19].

- Qu'est-ce que le PCE?

Path Computation Éléments (PCE) est une entité qui calcule les chemins en fonction des contraintes fournies pour le compte des routeurs, d'un OSS ou d'un autre PCE du réseau. Lorsqu'un network node a besoin d'un chemin pour un LSP (étiqueté chemin de commutateur), il fait une demande au PCE en utilisant le protocole PCE (PCEP). Le PCE a accès aux informations de topologie pour l'ensemble du domaine réseau et utilise ces informations pour les calculs de correctifs. L'architecture PCE et le protocole PCE sont définis par l'IETF dans les RFC 4655 et 5440 respectivement [18].

I.11 Le protocole SNMP

SNMP est un protocole de gestion de réseau utilisé pour échanger des messages entre NMS (Network Management Systems) et des agents intégrés dans le système d'exploitation (système d'exploitation) des périphériques réseau gérables. SNMP permet à NMS de gérer les périphériques réseau à distance. SNMP permet aux systèmes de gestion de gérer les périphériques réseau à l'aide de MIB (Management Information Base).

Le concept de la MIB est simple lorsqu'une technologie, une application ou un système est en cours de développement, le développeur peut déterminer quelles informations sous forme de variable seraient utiles pour gérer l'élément spécifique. Le développeur créerait alors une MIB, qui contiendrait des OID (Object Identifiers). Ce sont ces OID qui référencent la variable de données de gestion souhaitée [20].

I.12 L'interface CLI

Une interface en ligne de commande ou ILC (en anglais command line interface, couramment abrégé CLI) est une interface homme-machine dans laquelle la communication entre l'utilisateur et l'ordinateur s'effectue en mode texte [21] :

- L'utilisateur tape une ligne de commande, c'est-à-dire du texte au clavier pour demander à l'ordinateur d'effectuer une opération ;
- L'ordinateur affiche du texte correspondant au résultat de l'exécution des commandes tapées ou à des questions qu'un logiciel pose à l'utilisateur.

Une interface en ligne de commandes peut servir aussi bien pour lancer l'exécution de divers logiciels au moyen d'un interpréteur de commandes, que pour les dialogues avec l'utilisateur de ces logiciels. C'est l'interaction fondamentale entre un homme et un ordinateur (ou tout autre équipement informatique) [21].

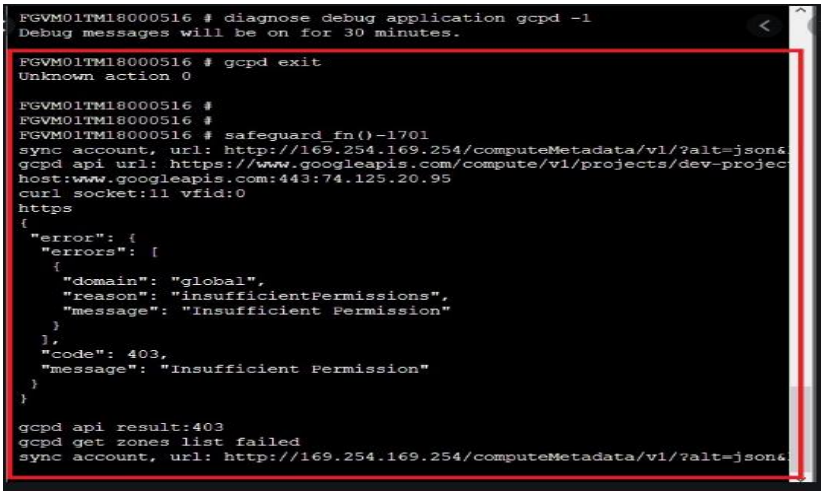


Figure 1.7 : commande CLI

I.13 API REST

Les API REST imitent la façon dont le web lui-même marche dans les échanges entre un client et un serveur.

-Une API REST est :

- Sans état
- Cacheable (avec cache = mémoire)
- Orienté client-serveur
- Avec une interface uniforme
- Avec un système de couche
- Un code à la demande (optionnel)

Le principe du client-serveur définit les deux entités qui interagissent dans une API REST : un client et un serveur, les mêmes entités qui communiquent sur le web. Un client envoie une requête, et le serveur renvoie une réponse. Ce dernier doit avoir le plus d'informations possible sur le client, car il est important qu'ils soient capables de travailler indépendamment l'un de l'autre [22].

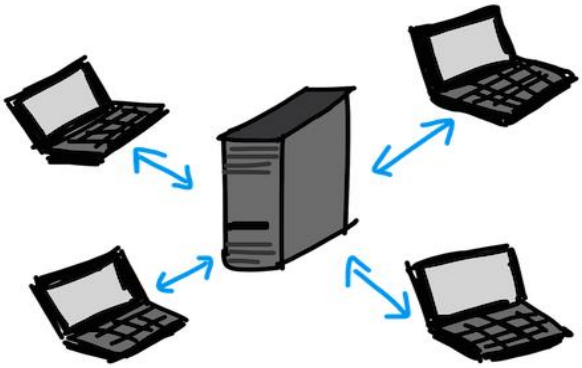


Figure 1.8: L’échange entre le serveur et le client

- API

Une API (Application Programming Interface) consiste en un code qui permet à deux programmes logiciels de communiquer.

Côté développeur, l'API définit la manière de rédiger un programme qui sollicite des services auprès d'un système d'exploitation ou d'une autre application. Les API sont mises en œuvre au moyen d'appels de fonction, constitués de verbes et de noms. La syntaxe requise est décrite dans la documentation de l'application appelée.

En général, les API sont diffusées à des fins de développement tiers dans le cadre d'un kit de développement logiciel (SDK, Software Development Kit) ou sous la forme d'une API ouverte publiée sur Internet. Si les applications sont rédigées dans des langages différents ou pour des plates-formes différentes, un logiciel intermédiaire, ou middleware, permettra la communication des deux applications en fournissant des services de messagerie [23].

- REST

REST (Representational State Transfer) est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web. Les services web conformes au style d'architecture REST, aussi appelés services web RESTful, établissent une interopérabilité entre les ordinateurs sur Internet. Les services web REST permettent aux systèmes effectuant des requêtes de manipuler des ressources web via leurs représentations textuelles à travers un ensemble d'opérations uniformes et prédéfinies sans état. D'autres types de services web tels que les services web SOAP exposent leurs propres ensembles d'opérations arbitraires [24].

REST est inspiré de l'architecture du web basée sur le protocole HTTP. Dans le paradigme REST, la pièce principale est la ressource, toutes les ressources doivent disposer d'un URI (Uniform Resource Identifier) et répondre aux opérations (verbs) HTTP (GET, PUT, POST, DELETE). Ces quatre méthodes permettent l'interaction et la manipulation des ressources comme le montre la figure:

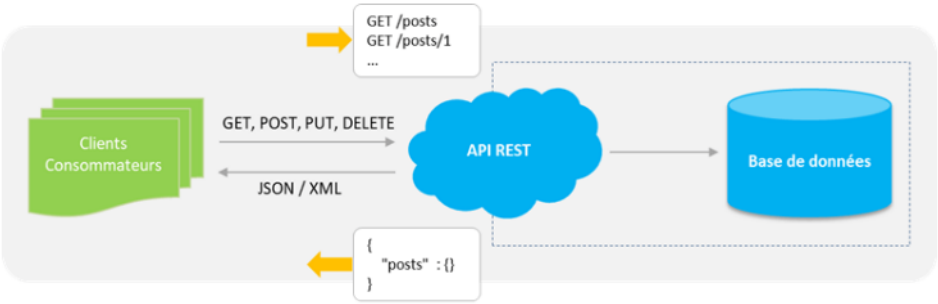


Figure 1.9: Fonctionnement REST

I.14 VxLAN

En raison du nombre limité de VLANs possible (4096) en 802.1Q, les constructeurs ont mis au point de nouvelles technologies de VLANs déployable dynamiquement au-dessus d'un réseau IP routé. VxLAN est l'une d'entre elles [25].

Nous sommes là au cœur des technologies réseau (SDN : Software Defined Networks) permettant de faire fonctionner des clouds répartis sur plusieurs DataCenters [25] .

I.14.1 C'est quoi le VxLAN ?

VxLAN est un format d'encapsulation porté par Cisco et VMware ayant des fonctionnalités semblables aux VLANs mais avec quelques améliorations. VxLAN est l'acronyme de (Virtual eXtensible LAN) qui est une standardisation a été proposée à l'IETF en 2011.Grâce à ce format, il est possible de faire transiter des trames de niveau 2, dans UDP (User Datagram Protocol) [25].

Cette propriété permet, par conséquent, d'utiliser une segmentation de type VLAN au-delà d'un domaine Ethernet [25].

De plus, grâce à son champ d'identification sur 24 bits, il est possible de créer sur un même domaine VxLAN plus de 16 millions de VLANs différents (224 plus précisément). Cependant, puisque VxLAN est un format d'encapsulation, il provoque une surcharge d'entête dans les trames Ethernet. Le schéma ci-dessous compare une trame Ethernet avec VLAN "standard" avec une trame Ethernet avec VxLAN [25].

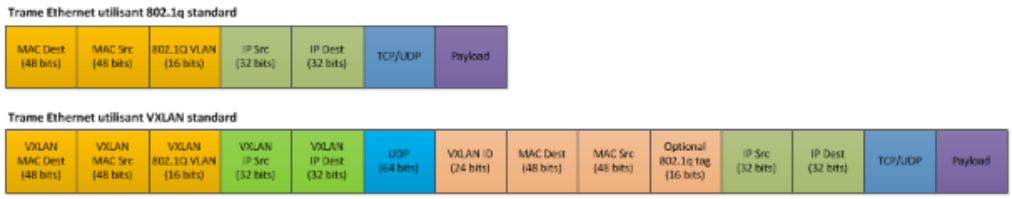


Figure 1.10 : comparaison entre trames Ethernet avec VLAN et avec une trame Ethernet avec VxLAN

I.15 Vue générale sur les différentes implémentations de SDN dans les réseaux

I.15.1 Réseaux WAN (SD-WAN)

SD-WAN (Software-Defined Wide-Area Networking, ou réseau étendu à définition logicielle), est une technique logicielle visant à rendre les réseaux étendus plus intelligents et plus flexibles. Elle commence généralement par la connexion de bureaux directement à Internet via des liens haut débit de commodité, plutôt que d'envoyer tout le trafic vers un bureau régional via des lignes privées (qui sont souvent basées sur une technologie plus ancienne et plus coûteuse, appelée MPLS).

Les configurations et les stratégies d'accès sont gérées de manière centralisée et sont facilement applicables à travers tous les sites, ce qui évite d'avoir à gérer manuellement chaque appareil WAN individuellement [26].

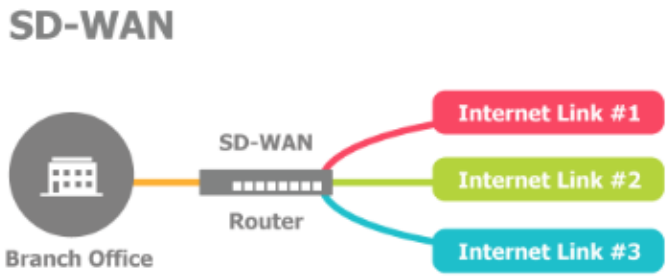


Figure 1.11: fonctionnement SD-WAN

I.15.2 Réseaux de Transport Optique (Transport SDN)

Le réseau de transport optique OTN, offre des débits de l'ordre du térabit pouvant acheminer des volumes toujours plus grands de trafic de données et de trafic vidéo. La technologie OTN (Optical Transport Network) est une solution qui combine les avantages de SDH et WDM dans les réseaux de transport optiques de nouvelle génération. Pour permettre la programmation du logiciel et réaliser le transport Cas d'utilisation SDN, les réseaux de transport optique doivent être plus flexibles alors ils ont toujours été [27].

I.16 Avantages de la commutation OTN pour le transport SDN

Historiquement, OTN a été le protocole de facto pour la gestion des réseaux DWDM, mais s'est limité à un protocole de trame pour les fonctionnalités FEC et OA&M. Il n'a été que récemment défini et utilisé comme couche réseau profondément canalisée et commutable. En tant que couche réseau, OTN joue quelques rôles clés [28]:

- Optimisation, commutation et ajustement de la bande passante.
- Convergence IP / paquets et multi-services.
- Protection.

Optimisation, commutation et ajustement de la bande passante OTN - bien que le SDN ne soit pas une condition préalable au déploiement de la commutation OTN, un réseau de transport optique conçu avec la commutation OTN ajoute plus de flexibilité au SDN de transport. Alors que le déploiement de 100G dans le réseau de transport résout le problème global de la bande passante, la majorité des clients alimentant le réseau de transport restent largement 10G ou moins12 [28].

I.17 Réseaux Data Center (SDN Datacenter)

Data center – l'architecture SDN facilite la virtualisation réseau, ce qui permet l'hyper évolutivité dans le data center, d'automatiser la migration des Machines virtuelles, une meilleure utilisation des serveurs, une consommation d'énergie plus faible, et l'optimisation de la bande passante [29].

Dans le cas d'un Data Center hébergeant plusieurs applications en se basant sur une architecture réseau traditionnelle, lorsqu'une application change de serveur on doit reconfigurer l'adressage, la configuration du Vlan , alors qu'avec SDN il suffit juste d'ajouter un nouveau flux dans le contrôleur pour prendre en compte le changement, ce qui est révolutionnaire c'est que l'application elle-même peut communiquer avec le contrôleur, par ex lors d'une maintenance, le contrôleur peut arrêter ainsi d'envoyer le trafic en destination du serveur hébergeant l'application[29].

I.18 Réseaux sans fil (Wireless network Transport SDN)

Le SDN, qui permet l'instanciation des capacités des réseaux de transport sans fil à la demande, peut jouer un rôle essentiel dans la résolution de ces problèmes et devenir l'un des besoins clés des opérateurs mobiles du monde entier.

Tout semble parfait, mais d'une manière ou d'une autre, le marché du SDN dans le domaine du transport sans fil ne décolle pas à un rythme qui correspond au grand nombre de communiqués de presse publiés récemment sur les innovations SDN et les projets communs fabricants-opérateurs. En effet, les fabricants de liaisons sans fil

investissent dans l'innovation alors que les opérateurs recherchent des solutions de reconfiguration dynamique [30].

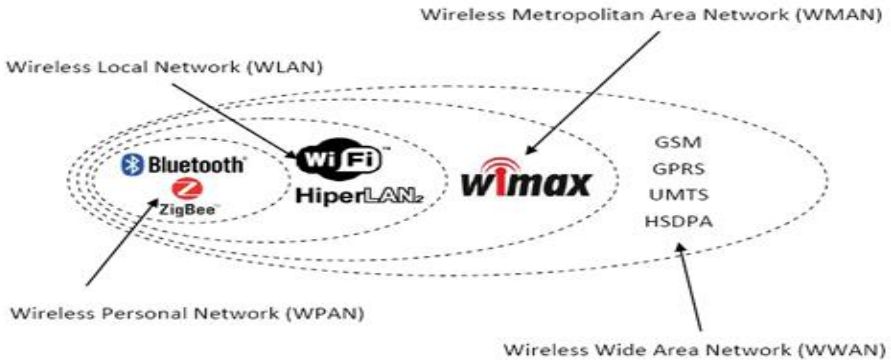


Figure 1.12 : Présentation des différents réseaux sans-fil et mobiles

I.19 Conclusion

Après avoir présenté les différents concepts théoriques lié au Software Defined Networking qui consiste à définir l'architecture de SDN les interfaces de programmation API, les différentes implémentations de SDN dans les réseaux (réseau SD-WAN, réseau transport optique, réseau data center et réseau sans fil).

Avec l'accroissement continu des technologies, la nécessité de développer de nouveaux outils est devenue primordiale, c'est pourquoi l'entreprise Cisco a engagé le développement de ses propres solutions SDN, une de ses solutions est appelée APIC-EM pour Application Policy Infrastructure Controller Enterprise Module, que nous allons détailler dans le prochain chapitre.

Chapitre II : La plate-forme Cisco APIC-EM

II.1 Introduction

Comme déjà mentionné dans le premier chapitre, dans une l'architecture SDN, le réseau est contrôlé par une entité centrale responsable de la gestion et de l'application des politiques « Le contrôleur ». Le contrôleur est l'élément clé de la solution SDN, il maintient une vue sur l'ensemble du réseau, met en œuvre les décisions stratégiques et contrôle tous les périphériques qui composent l'infrastructure réseau. Il existe un grand nombre de contrôleurs SDN qui varient selon les cas d'utilisation auxquels ils sont spécifiquement ciblés. Il existe une sélection de contrôleurs SDN open source, et d'autre qui sont propriétaires (commerciales). Dans notre projet, nous allons utiliser un des contrôleurs propriétaire CISCO APIC-EM que nous allons détailler au cours de ce chapitre.

II.2 Présentation du SD-WAN entreprise

Aujourd'hui, les approches SDN, SD-WAN bouleversent les systèmes traditionnels des réseaux. Elles initient un « véritable changement » dans les infrastructures réseau des entreprises avec des promesses en termes de réduction des coûts, d'agilité, de flexibilité, de rapidité de mise en œuvre et d'adaptation. Les tendances se succèdent et impactent fortement la conception même du réseau. SDN, SD-WAN sont devenus des briques de base pour préparer au mieux son réseau aux défis d'aujourd'hui et demain. Mais encore faut-il bien comprendre ces différents concepts [31].

II.2.1 Définition de SD-WAN ?

SD-WAN est un acronyme pour « Software-Defined Wide Area Network », soit réseau étendu à définition logicielle, et il est présenté dans les années 2020 comme la nouvelle évolution majeure des télécommunications. Un SD-WAN facilite la gestion

du réseau en séparant la partie matériel du réseau de ses mécanismes de contrôle et de gestion. Ce concept est similaire à la manière dont le réseau à définition logicielle met en œuvre la virtualisation pour améliorer la gestion et l'exploitation des Centres de données.

Une application majeure du SD-WAN consiste à permettre aux entreprises de construire des WANs de meilleure performance en utilisant un accès internet moins coûteux et disponible dans le commerce, permettant aux entreprises de remplacer, partiellement ou totalement, les technologies en connectivité WAN privées plus chères, à l'instar de la technologie MPLS [32].

II.2.2 Que dire du WAN HYBRIDE et de SD-WAN ?

Le WAN hybride regroupe des lignes privées pour la transmission d'informations en entreprise (MPLS) et des lignes provenant du réseau public (réseau internet). Des précautions sont prises pour transmettre les données stratégiques et permettre aux différents sièges de l'entreprise de communiquer plus facilement entre eux. Le premier avantage est l'économie réalisée à l'année. Les moyens mis en œuvre facilitent les transmissions de données entre l'entreprise et ses filiales ou le siège social et ses succursales. Le second avantage est l'évolutivité : la bande passante peut être augmentée au fur et à mesure des besoins. Très proche, le SD WAN (Software Defined Wan) assure la gestion des différents réseaux grâce à un logiciel dédié agissant selon la bande passante à disposition, le trafic et les connexions possibles. Plusieurs technologies permettent donc aux entreprises de développer leurs moyens de communication et IT en respectant les politiques réseaux [33].

Pour autant, les limites du WAN hybride et des solutions SD WAN « low cost » sont facilement perceptibles lorsqu'un grand nombre d'applications doivent être gérées simultanément, des applications qui nécessitent de plus en plus de débit, de connectivité et de sécurité [33].

II.3 Définition d’un contrôleur

Traditionnellement, les contrôleurs SDN sont utilisés dans les réseaux de centres de données. Cependant, au fur et à mesure de l'évolution de la technologie SDN, le WAN est devenu un cas d'utilisation convaincant, entraînant la croissance de la technologie WAN définie par logiciel (SD-WAN). Un contrôleur SD-WAN exécute plusieurs des mêmes tâches qu'un contrôleur SDN, en suivant les configurations de stratégie pour diriger le trafic WAN sur l'itinéraire le plus efficace. Le marché SD-WAN a moins d'options open source notables que SDN, car la plupart des contrôleurs SD-WAN sont généralement liés à la plate-forme SD-WAN propriétaire du fournisseur [34].

II.4 Solutions SDN Cisco

Dans le monde de technologie le SDN avait une évolution rapide et plusieurs entreprises tel CISCO APIC qui développe leur contrôleur pour répondre aux besoins désirés.

II.4.1 Cisco APIC (Cisco Application Policy Infrastructure Controller)

Le module de réseau de grande entreprise Cisco Application Policy Infrastructure Controller (APIC) est un contrôleur logiciel qui automatise et simplifie la configuration, le provisionnement et la gestion du réseau. Depuis un seul ordinateur x86 ou système configuré pour la virtualisation, vous pouvez rapidement déployer des services, politiques et applications pour les réseaux de succursales et de campus [35].

le module de réseau de grande entreprise Cisco APIC peut être utilisé dans les réseaux filaires, sans fil, physiques et virtuels. Il protège l’investissement en fonctionnant avec l’infrastructure existante. En outre, il permet de créer un réseau intelligent, ouvert et programmable qui aide à [35] :

● répondre rapidement aux besoins croissants des applications.

● vous concentrer davantage sur l’innovation pour créer des occasions commerciales.

● réduire la complexité de la mobilité/stratégie PAP (prenez votre appareil personnel), du nuage et des autres tendances.

-Caractéristiques et fonctionnalités

APIC offre plusieurs fonctionnalités. Parmi lesquelles, on peut citer [36] :

- Contrôle centralisé des applications et automatisation des services réseau.
- Structure de gestion et de politique commune pour l'infrastructure physique, virtuelle et cloud.
- Architecture ouverte qui supporte les API nord et sud.
- Contrôle, visibilité, mobilité et reconnaissance des applications.
- Mise en œuvre de la sécurité multi locataires (multi-tenant) et de la qualité de Service.

APIC a été développé, principalement, pour les DC (Data Center) et Cloud. Cisco innove également dans les entreprises et le campus, en adoptant une approche similaire au DC basée sur les politiques en introduisant une nouvelle solution « APIC-EM » [36].

II.4.2 Cisco APIC-EM

Cisco APIC-EM fournit le SDN à l'entreprise sur le campus et le WAN, pour les périphériques filaires et sans fil. Cisco APIC-EM fournit une automatisation basée sur des politiques de l'infrastructure réseau, simplifiant le déploiement et les opérations réseau. Toutes les fonctionnalités de Cisco APIC-EM sont exposées via une API REST. Les API offrent trois avantages [37]:

1- Automatisation des tâches courantes - automatisation de l'attribution d'étiquettes de localisation aux appareils en fonction du nom de l'appareil.
2- Intégration - définition dynamique d'une politique QoS.
3- Utiliser l'API d'une manière impossible dans l'interface utilisateur.

Cisco APIC-EM dispose d'une interface Southbound qui communique directement avec le réseau. L'interface Southbound n'est pas exposée directement, faisant abstraction de la complexité sous-jacente du réseau traditionnel et offrant une simplification. La figure suivante montre les différents composants de l'architecture d'APICEM :

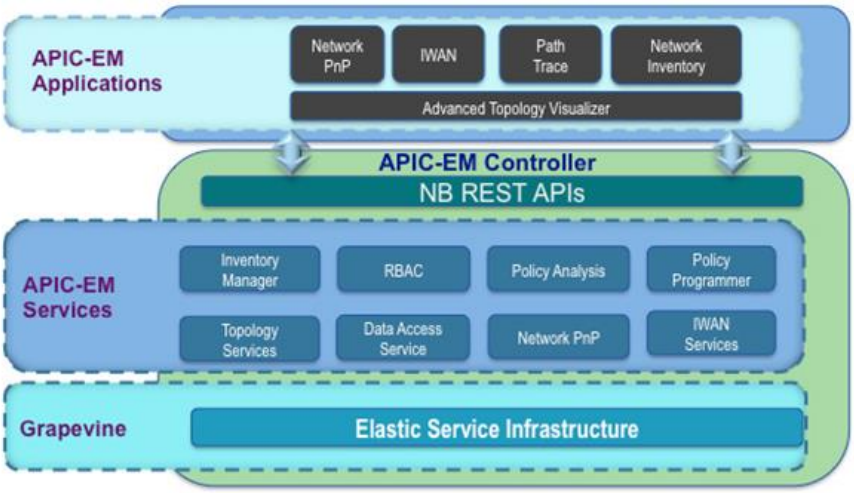


Figure 2.1 : Architecture APIC-EM

II.5 Applications d'APIC-EM

Cisco APIC-EM comprend les applications client clés suivantes:

-Intelligent WAN (IWAN) : Cisco IWAN sur APIC-EM prolonge SDN aux branches avec une approche d'application centrée. Cela fournit une gestion centralisée avec des applications distribuées sur l'ensemble du réseau [38]. Avec Cisco IWAN, l'organisation peut fournir plus de bande passante à leurs connexions de filiales en utilisant des options de transport WAN moins coûteuses sans affecter les performances, la sécurité ou la fiabilité. Il automatise le déploiement via une interface graphique et réduit le temps nécessaire pour la configuration des services réseau avancés.

-Plug and Play (PnP) : PnP est un mécanisme permettant d'automatiser le déploiement des périphériques. Il suffit de brancher un périphérique (routeur, commutateur, point d'accès) dans le réseau. Ce dernier découvre le contrôleur et l'automatisation commence. Quelques minutes plus tard, l'équipement est mis à niveau, configuré et devient opérationnel. La solution PnP comporte trois composants principaux [39] :

1. PnP Server : une application exécutée sur APIC-EM qui gère les sites, les périphériques, les images, etc.

2. PnP Agent : un switchs, routeurs ou points d'accès sans fils, qui recherche un «contrôleur» lorsque le périphérique est initialement en fonctionnement.

3. PnP Protocol : permet à l'agent et au contrôleur de communiquer.

-Easy QoS : La qualité de service désigne la capacité d'un réseau à fournir un service préférentiel ou différentiel au trafic réseau sélectionné. La fonction EasyQoS permet de configurer la qualité de service sur les périphériques du réseau qui ont été découverts par le Cisco APIC-EM.

-Path Trace : simplifie la résolution des problèmes de performances du réseau en traçant les chemins des applications sur le réseau et en fournissant des statistiques pour chaque saut le long du chemin [37].

II.6 Fonctionnalités d'APIC-EM

Cisco APIC-EM offre des fonctionnalités qui seront détaillées dans le tableau [40] :

Fonctionnalités	**Description**
Base de données des informations du réseau (NIDB) « Network Information DataBase »	Cisco APIC-EM scanne périodiquement le réseau pour créer une source unique d'informations. Cet inventaire comprend tous les périphériques réseau, ainsi qu'une abstraction pour l'ensemble du réseau d'entreprise.
Visualisation de la topologie du réseau	Cisco APIC-EM détecte et mappe automatiquement les périphériques réseau sur une topologie physique avec des données détaillées au niveau du périphérique d'une façon graphique.
Serveur d'infrastructure à clé publique (PKI) (Public Key Infrastructure)	Cisco APIC-EM prend en charge une fonctionnalité de gestion de certificats PKI utilisée pour authentifier les sessions (HTTPS). Ces sessions utilisent des agents de confiance reconnus appelés autorités de certification,

	qui gèrent et délivrent les demandes de certificats aux entités participantes et fournissent une gestion centralisée des clés. Cisco APIC-EM utilise le PKI pour importer, stocker et gérer un certificat X.509. Le certificat importé devient un certificat d'identité pour le contrôleur lui-même, et le contrôleur présente ce certificat à ses clients pour l'authentification.
Haute disponibilité (HA) (High Avaibility)	APIC-EM fournit une HA physique (cluster) en mode redondant N+1 où tous les nœuds fonctionnent en mode Active-Active pour optimiser les performances et le partage des charges.
Sauvegarde et restauration	Cisco APIC-EM prend en charge la sauvegarde complète et la restauration de la base de données entière à partir de l'interface graphique du contrôleur.
Vérification de l'intégrité	L'application de vérification d'intégrité tire parti des capacités fournies par APIC-EM pour collecter des mesures d'intégrité à partir d'appareils surveillés, évalue ces mesures pour l'exactitude et les changements inattendus, et fournit une visibilité sur les résultats, dans le but d'identifier rapidement un compromis afin de minimiser son impact.

Tableau 2.1 : Fonctionnalités du contrôleur APIC-EM

II.7 Avantages

La plate-forme Cisco APIC-EM prend en charge les réseaux d'entreprise filaires et sans fil sur les infrastructures Campus, Branch et WAN. Il offre les avantages suivants [41]:

- Crée un réseau intelligent, ouvert et programmable avec des API ouvertes.

- Gain de temps, de ressources et de coûts grâce à une automatisation avancée.
- Transforme les politiques d'intention commerciale en une configuration réseau dynamique.
- Fournit un point unique pour l'automatisation et le contrôle à l'échelle du réseau.

II.8 Description des APIs REST de l'APIC-EM

Les APIs REST représentent les interfaces de nord de l'APIC-EM qui permet le dialogue entre les applications et le contrôleur comme la montre la figure :

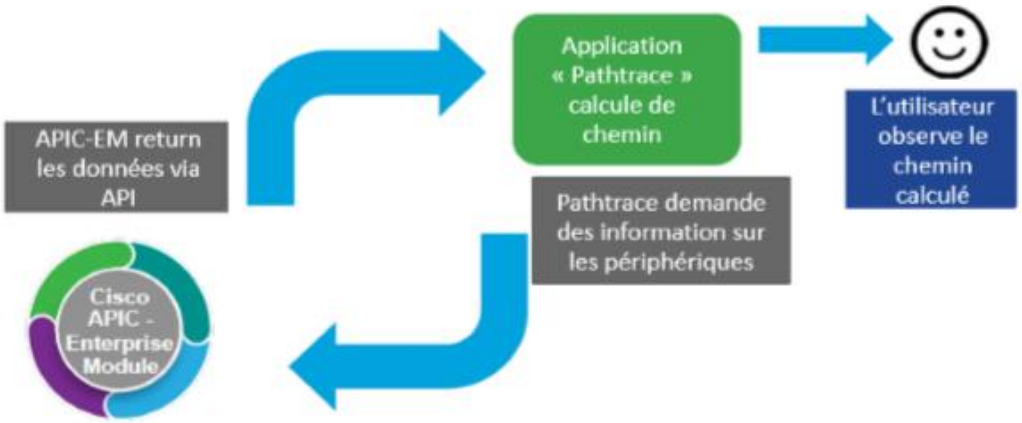


Figure 2.2 : Interaction avec APIC-EM à travers les API REST

Les API Northbound sont basées sur REST et peuvent permettre aux applications de découvrir et de contrôler les éléments de votre réseau à l'aide du protocole HTTPs avec des verbes HTTP (par exemple, GET, POST, PUT et DELETE) avec la syntaxe JavaScript Object Notation (JSON). Il est riche en fonctions, hautement sécurisé et peut fournir un contrôle par programme facile à utiliser des éléments de réseau, interfaces et hôtes.

L'interface en direction sud parle aux éléments du réseau à l'aide de l'interface de ligne de commande (CLI) et du protocole SNMP (Simple Network Management Protocol). L'utilisation de CLI et de SNMP peut garantir que APIC-EM fonctionne avec les produits Cisco existants. Les futures versions d'APIC-EM tireront pleinement parti

des autres technologies et API de périphériques vers le sud au fur et à mesure de leur mise en œuvre [42].

II.9 Conclusion

Au cours de ce chapitre, nous avons abordé des notions générales sur les contrôleurs et nous avons détaillé les différents aspects d'APIC-EM, celui que nous allons utiliser dans notre solution, tout en présentant son architecture, son fonctionnement et les applications qu'il offre à ses utilisateurs, avec une simplification et une automatisation avec plus de performance.

Les applications présentées dans ce chapitre ne sont pas le seul moyen d'interagir avec le contrôleur, APIC-EM donne aussi la possibilité d'utiliser les API REST pour plus de flexibilité et de contrôle. Cette méthode sera présentée dans le chapitre suivant.

Chapitre III : mise en œuvre de l'application

III.1 Introduction

APIC-EM fournit un point de contrôle unique qui simplifie le fonctionnement des réseaux d'entreprise en donnant aux utilisateurs la possibilité de contrôler leurs réseau d'une façon facile et rapide avec plus de performances. Le cœur du contrôleur est un moteur de politique qui traduit l'intention des entreprises en configuration réseau. Le contrôleur expose des API basée sur une architecture REST pour permettre l'interaction et l'exploitation de ses capacités. Dans ce chapitre, nous allons définir la notion des APIs REST, son fonctionnement et comment nous pouvons les utiliser pour interroger le contrôleur APIC-EM ?

III.2 Interaction avec APIC-EM

Il existe deux façons pour interagir avec le contrôleur APIC-EM, soit en utilisant l'interface graphique soit à travers les API REST.

III.3 Interface graphique

Le contrôleur que nous avons utilisé dans notre projet se trouve dans le cloud de Cisco. L'accès au contrôleur APIC-EM se fait à l'aide d'un navigateur web (Firefox v. 46.0 ou plus, ou bien Google Chrome v. 50.0 ou plus) à travers l'adresse https://sandboxapicem.cisco.com/ . Après la saisie de l'adresse, une page web d'authentification s'affiche comme le montre la figure 3.1:

Figure 3.1 : Authentification

Après l'authentification, la première page qui va apparaitre est le tableau de bord qui contient diverses informations d'une façon graphique (Inventaire des appareils, Découverte des dispositifs inaccessibles, Sites de succursale, Projet de réseau Plug and Play, EasyQoS, et l'application de trace de chemin Pathtrace) comme illustrée dans la figure 3.2 :

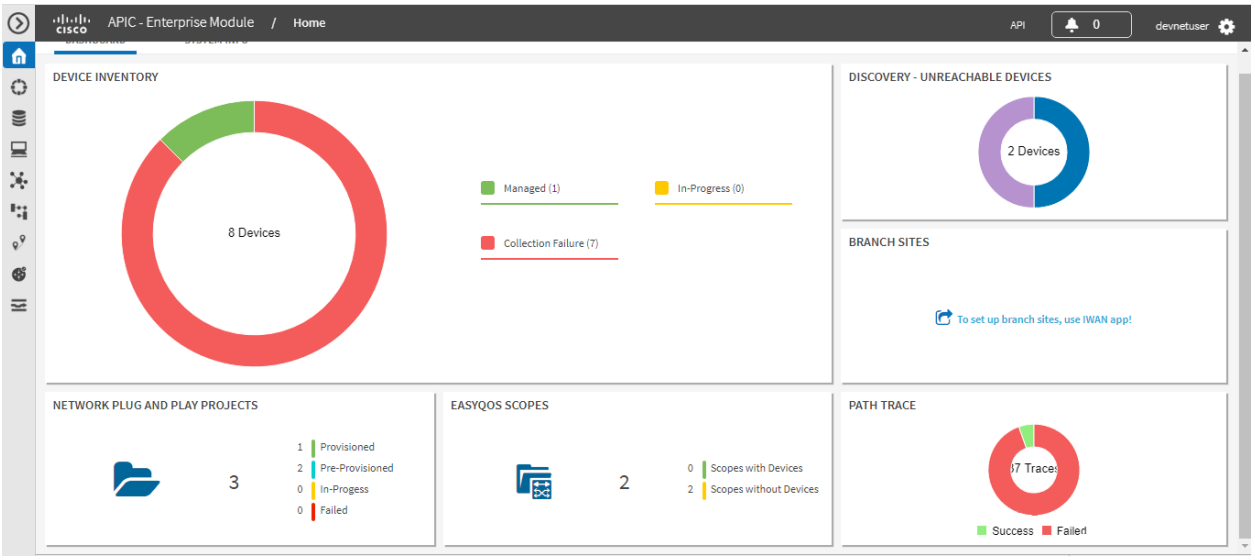


Figure 3.2 : Tableau de bord

Le volet de navigation offre des options pour accéder aux principales fonctionnalités et applications Cisco APIC-EM comme expliqué dans le tableau 3.1:

Icône	Nom	Description
	Masquer/afficher la navigation	Permet de masquer et afficher le volet de navigation.
	Accueil	Fournit des informations sur l'APIC-EM, telles que son état du réseau, l'état du système et ses informations.
	Découverte	Permet de configurer les options de découverte pour numériser les périphériques et les hôtes du réseau.
	Inventaire des appareils	Fournit l'accès à la base de données d'inventaire où l'on peut afficher, filtrer et trier des informations de tableaux sur les périphériques découverts sur le réseau.
	Inventaire hôte	Fournit l'accès à la base de données d'inventaire où l'on peut afficher, filtrer et trier des informations de tableaux sur les hôtes découverts dans le réseau.
	Topologie	Présente les périphériques et les liens que Cisco APIC-EM découvre comme une carte de topologie physique avec des données détaillées sur le niveau de l'appareil. La topologie des périphériques et les liens peuvent également être présentés sur une carte géographique.
	IWAN	Simplifie la fourniture de profils réseau IWAN avec des politiques commerciales simples. L'application IWAN définit les

		préférences de niveau métier par application ou groupes d'applications avec les chemins préférés pour les liens WAN hybrides. Cela améliore l'expérience de l'application sur toute connexion et économise les coûts de télécommunication en utilisant des liens WAN moins chers.
	EasyQoS	Permet de configurer la qualité du service sur les périphériques réseau.
	Path Trace	Permet au contrôleur de collecter les informations sur les différents équipements réseau et les utiliser pour calculer le chemin entre deux périphériques
	Network Plug and Play	Fournit une expérience de déploiement ZTD hautement sécurisée, évolutive, transparente et unifiée pour les clients sur les routeurs, les commutateurs et les points d'accès sans fil de Cisco.

Tableau 3.1 :Volet de navigation du contrôleur APIC-EM

III.4 Requête

REST est centrée sur le modèle de requête / réponse HTTP. Utiliser une API est tout aussi simple que de faire une requête HTTP qui se forme de la façon suivante :

- Méthode :
  - GET - Récupérer des données.
  - POST - Créer une ressource.
  - PUT - Mettre à jour les données.
  - DELETE - Supprimer les données.

- URL (Uniform Resource Locator) L'URL du point final que nous souhaitons appeler.
- Authentification : Avec Cisco APIC-EM, chaque ressource est mappée à une ou plusieurs actions et chaque action est mappée à une autorisation requise pour un utilisateur.

Toutes les API REST sont donc protégées par le processus d'authentification dont il existe trois types :

- HTTP de base.
- OAuth: c'est un standard ouvert pour l'authentification HTTP et la gestion des sessions.
- Token: comme avec OAuth, un jeton est créé et passé avec chaque appel API, mais il n'y a pas de gestion de session ni de suivi de clients.

- Organisme de demande : JSON ou XML contenant des données nécessaires à la demande complète peut être envoyé dans le corps de la demande.

III.5 Réponse

Les informations renvoyées (par le contrôleur) sont définies dans la partie Réponse. Ils comprennent le format des données, les attributs et les codes d'état HTTP qui sont utilisés pour renvoyer des réussites, des erreurs ou d'autres états. Nous pouvons citer comme exemples des codes:

- 2xx : Succès.
- 3xx : Redirection.
- 4xx : Erreur utilisateur.
- 5xx : Erreur du serveur.

- Le mouvement vers REST, comme l'un des moyens de manipuler les réseaux SDN repose sur un certain nombre d'avantages perçus:

- Simplicité : Le mécanisme REST basé sur le Web utilise les commandes simples HTTP : GET, PUT, POST et DELETE.

- Flexibilité : L’accès aux ressources se fait à travers l’utilisation des URL.
- Extensibilité : La prise en charge de nouvelles ressources ne nécessite pas de recompilation de schémas ou de MIB (Management Information Base), mais exige seulement que l'application appelante accède à l'URL appropriée.
- Sécurité : La sécurisation des communications REST se fait simplement en l’exécutant via HTTPs.

-les API permettent le déploiement et l'exécution de la politique d'application dans l’infrastructure de réseau. Nous pouvons donc utiliser ces API pour récupérer des informations sur les différents équipements réseau, y compris la liste d'hôtes, la liste des périphériques, la liste des utilisateurs et plus. Ils sont disponible sur GUI sous forme d'une liste dans le swagger comme le montre la figure 3.3 ou nous pouvons y’accéder à travers ce lien: https://sandboxapicem.cisco.com/swagger

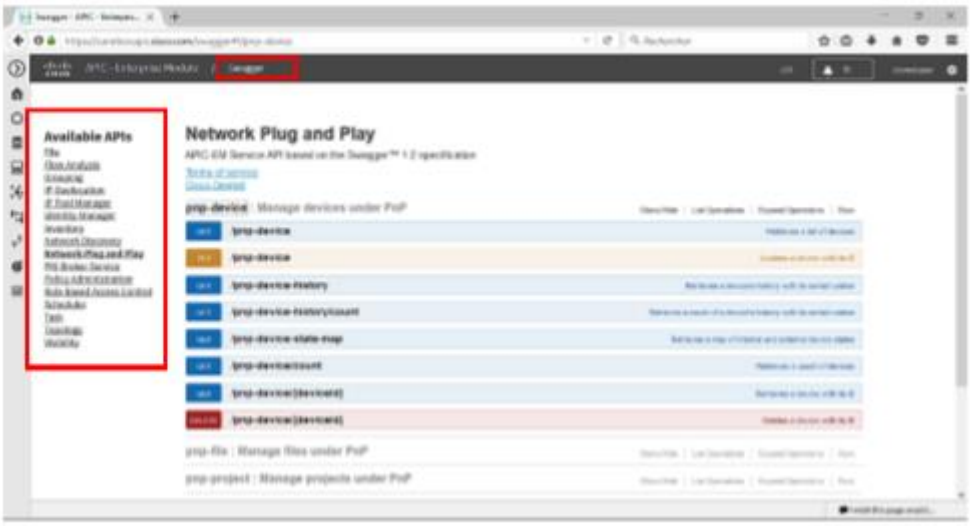


Figure 3.3 :API disponibles sur la GUI d’APIC-EM

III.6 Postman

La manipulation des API se fait soit par l’outil Postman qui peut être installé séparément ou comme une extension dans le navigateur Google Chrome soit à l’aide d’un langage de programmation.

Postman est un client HTTP pour tester les services Web. Il permet de construire rapidement des demandes et analyser les réponses envoyées par l'API en choisissant leur format (XML / JSON).

Postman facilite le test et la manipulation des APIs d'APIC-EM en permettant aux utilisateurs de mettre en place des requêtes HTTP d'une manière simple et rapide.

Après l'installation de l'outil Postman, nous avons importé les requêtes APIC-EM Sand box comme afficher dans la partie gauche de la figure 3.4 :

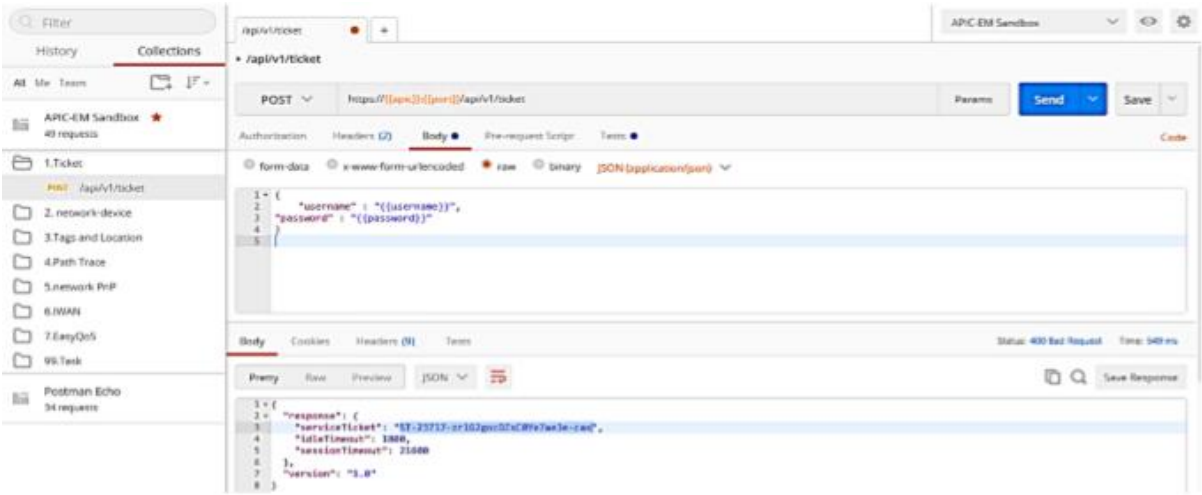


Figure 3.4 : postman

III.7 Langage de programmation

Malgré le nombre d'outils disponibles pour la capture et le suivi des paquets, les programmeurs préfèrent utiliser leur propre logiciel développé par le codage et les scripts. Le code auto-développé et programmé offre beaucoup de flexibilité dans la personnalisation de l'outil. De nombreuses organisations concernées par la sécurité, la confidentialité et l'intégrité, choisissent de ne pas utiliser de logiciel tiers. Au contraire, ils développent leurs propres outils en utilisant des langages de programmation efficaces, notamment Python, Java, PERL, PHP et bien d'autres.

Lors de génération des requêtes, nous pouvons utiliser n'importe quel langage de programmation tant qu'il supporte les requêtes REST. Dans notre cas nous avons choisi le langage de programmation Python.

III.8 Python

Python est un langage facile à apprendre, son code est plus lisible et dispose de nombreuses librairies pour interagir avec le système. Il est jusqu'à 5 fois plus concises que le langage Java par exemple. Ce qui augmente la productivité du développeur et réduit considérablement le nombre de bugs.

III.8.1 Installation de Python

L'installation se fait directement après le téléchargement de l'exécutable à partir du site de Python. Nous pouvons, par la suite, tester son succès en écrivant Python dans la ligne de commande, comme la montre la figure 3.5 :

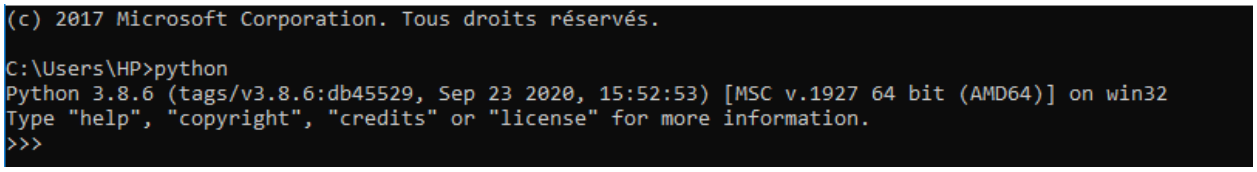


Figure 3.5 : Succès d'installation du Python

Une des forces de python est la multitude des bibliothèques disponibles. Installer une bibliothèque peut prendre beaucoup de temps : trouver le bon site, la bonne version de la bibliothèque, l'installer… Pour faciliter cette tache nous avons utilisé le PIP.

III.8.2 Installation de PIP

PIP est un environnement qui permet d'installer des paquets pour une utilisation par une application particulière, plutôt que d'être installé dans l'ensemble du système. L'installation de PIP se fait par l'exécution de la commande : « py –m pip insatll» et comme on peut touver dans les nouvelles version python l'installation de pip se fait automatiquement, pour vérifier l'installation de pip en écrivant « py –m pip » seulement comme illustré dans la figure 3.6 :

```
  C:\Python\Python38\python.exe -m pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
  cache                       Inspect and manage pip's wheel cache.
  wheel                       Build wheels from your requirements.
  hash                        Compute hashes of package archives.
  completion                  A helper command used for command completion.
  debug                       Show information useful for debugging.
  help                        Show help for commands.

General Options:
  -h, --help                  Show help.
  --isolated                  Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose               Give more output. Option is additive, and can be used up to 3 times.
  -V, --version               Show version and exit.
  -q, --quiet                 Give less output. Option is additive, and can be used up to 3 times (corresponding to
                              WARNING, ERROR, and CRITICAL logging levels).
  --log <path>                Path to a verbose appending log.
  --no-input                  Disable prompting for input.
  --proxy <proxy>             Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>         Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>             Set the socket timeout (default 15 seconds).
  --exists-action <action>    Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
                              (a)bort.
  --trusted-host <hostname>   Mark this host or host:port pair as trusted, even though it does not have valid or any
                              HTTPS.
  --cert <path>               Path to alternate CA bundle.
  --client-cert <path>        Path to SSL client certificate, a single file containing the private key and the
                              certificate in PEM format.
  --cache-dir <dir>           Store the cache data in <dir>.
  --no-cache-dir              Disable the cache.
  --disable-pip-version-check
                              Don't periodically check PyPI to determine whether a new version of pip is available for
```

Figure 3.6 : Installation PIP

-La figure 3.6 montre l'installation des package.

III.8.3 Installation de requests

Pour que Python puisse générer les requêtes HTTP, l'installation de requests est indispensable. Elle se fait comme illustrée dans la figure 3.7 par l'exécution de la commande : «py –m pip install requests » dans la ligne des commande (CMD).

```
C:\Users\HP>py -m pip install requests
Collecting requests
  Downloading requests-2.24.0-py2.py3-none-any.whl (61 kB)
     |████████████████████████████████| 61 kB 36 kB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2020.6.20-py2.py3-none-any.whl (156 kB)
     |████████████████████████████████| 156 kB 273 kB/s
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
  Downloading urllib3-1.25.10-py2.py3-none-any.whl (127 kB)
     |████████████████████████████████| 127 kB 65 kB/s
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 54 kB/s
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 28 kB/s
Installing collected packages: certifi, urllib3, idna, chardet, requests
Successfully installed certifi-2020.6.20 chardet-3.0.4 idna-2.10 requests-2.24.0 urllib3-1.25.10

C:\Users\HP>
```

Figure 3.7 : Installation de requests

Après l'installation, Python est capable d'importer la bibliothèque des requêtes pour les utiliser après dans les appels REST.

III.9 Code source

III.9.1 Description du code

Notre code est structuré en trois parties essentielles : l'authentification, la récupération des informations réseaux et l'affichage.

• Code: des scripts en Python qui exécute une requête particulière de l'API REST.

• Sortie: résultat complet de l'exécution de code qui sera affiché comme spécifier dans le script.

III.9.2 Authentification

La première étape est de créer une fonction qui permet de s'authentifier auprès du contrôleur, la figure 3.8 montre le script qui va être utilisé pour avoir un jeton d'authentification (service ticket) en envoyant le nom d'utilisateur ainsi que le mot passe. Ce jeton va être utilisé, par la suite, dans les communications entre eux (le contrôleur et le client).

```
import requests
from tabulate import tabulate
from flask import Flask
import json
from flask import render_template, jsonify
requests.packages.urllib3.disable_warnings()


def geticket():
    url = "https://sandboxapic.cisco.com/api/v1/ticket"
    payload = {"username":"XXXXXX ","password":"XXXXXX"}
    header = {"content-type": "application/json"}
    response= requests.post(url,data=json.dumps(payload), headers=header, verify=False)
    print(response.text)
    r_json=response.json()
    ticket = r_json["response"]["serviceTicket"]
    return ticket

geticket()
```

Figure 3.8 : Code de la fonction d'authentification

Dans ce script, nous avons commencé par la spécification des bibliothèques dont on a besoin :

- Requests: Pour les requêtes.
- JSON: Le format des messages.
- Tabulate: Une bibliothèque qui permet d'afficher un tableau en plusieurs formats.
- Flask : Framework qui fournit des outils, des bibliothèques et des technologies qui permettent de créer une application Web (page web, site, blog).

Par la suite, nous avons défini le nom de la fonction, spécifié l'URL du contrôleur qui peut être soit l'adresse IP, soit le nom DNS, défini le type du contenu qui doit être inclus dans l'en-tête (Application/JSON pour le contenue Web). Nous avons effectué un POST sur l'URL spécifiée et finalement, affiché le résultat et spécifier que cette fonction retourne une variable Ticket return (ticket). La figure 3.9 montre le résultat de l'exécution de ce code dans le CMD .

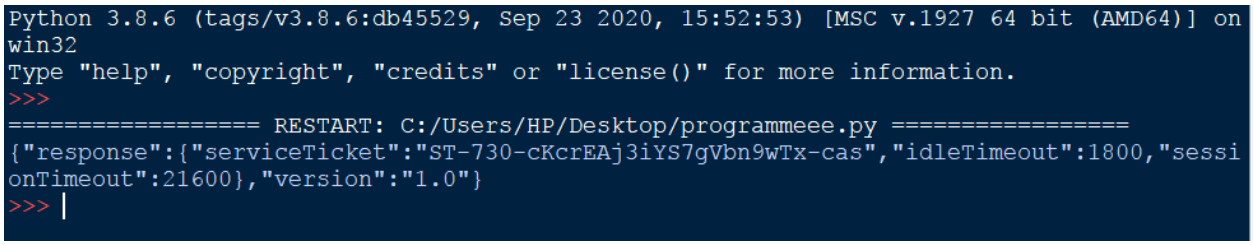


Figure 3.9 : Réponse de GetTicket

III.9.3 Corps du code

Dans cette partie, chaque fonction déclarée fait appel au résultat de la fonction GetTicket comme une variable. Le résultat obtenu par les API REST suit un modèle déjà défini dans l'API elle-même, les variables de chaque API sont définies en JSON ce qui les rend lisibles et faciles à utiliser. Dans notre cas nous allons faire appel aux trois APIs suivantes :

1. /Network-Device : Cette API donne des informations relatives aux équipements réseau (nom, Adresse, famille d'équipement, etc.).

2. /host : Donne comme résultat tous les périphériques finaux ainsi que leurs informations telles que le nom d'hôte, Adresse Mac, son type, et autre.

3. /interface : Retourne les informations liées aux interfaces comme ses statuts administratifs, ID et l'état (Up/Down).

Pour faciliter la manipulation du code, nous avons ajouté une ligne, comme illustrée dans la figure 3.10 qui sert à récupérer une variable (un nombre) par l'utilisateur où chaque nombre correspond à une valeur (1- Device, 2-hosts, 3-interfaces). Nous avons par la suite utilisé des conditions pour afficher le résultat choisi, par exemple, si la valeur entière est égale à «1» il va exécuter la partie du code qui sert à récupérer et afficher les informations relatives aux équipements réseau ou Network Device.

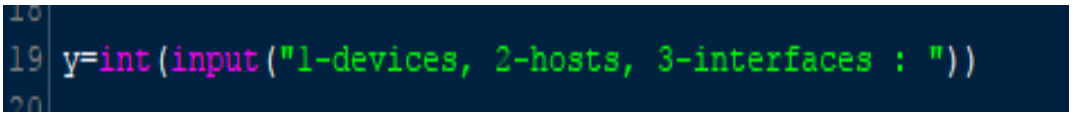


Figure 3.10 : Récupération de la variable pour le choix du résultat

La partie suivante du code sert à récupérer les informations et afficher le résultat obtenu. Tout d'abord, nous avons utilisé le ticket obtenu par la fonction getTicket pour s'authentifier, spécifier l'URL de la ressource. Un tableau Device est déclaré par la suite. La condition Try / except signifie que s'il y'a un problème, alors il va afficher le message d'erreur et s'il n'y a pas, il va l'exécuter et retourner la réponse dans le tableau Device. Un autre tableau, déclaré device_list, vas récupérer les informations du Device (là où on peut spécifier les informations qu'on voudra avoir). Et à la fin, nous allons afficher le résultat final en spécifiant le nom des entêtes du tableau dans header et la façon d'affichage dans tablefmt. La figure 3.11 montre le script :

```python
#### script quand on choisi la command 1-devices ####

if y == 1:
    print("process reloading please wait_______")

    ticket=getTicket()
    url="https://sandboxapic.cisco.com/api/v1/network-device"
    headers = {"X-auth-Token": ticket}

    device = []
    try:
        resp= requests.get(url,headers=headers,verify = False)
        response_json = resp.json()
        device = response_json["response"]
    except:
        print("something wrong, cannot get network device information")

    device_list= []
    for item in device:
        device_list.append([item["hostname"],item["managementIpAddress"],item["type"],item["id"]])
    print(tabulate(device_list,headers=['Hote','Adresse IP','type','Identificateur'],tablefmt="grid"))
    print("_____ end")
```

Figure 3.11 : Récupération des informations d'équipements réseau

Le résultat de l'exécution du script précédent (choix «1» qui correspond à network Device) est affiché dans la figure 3.12 :

```
================== RESTART: C:\Users\HP\Desktop\programmeee.py ==================
1-devices, 2-hosts, 3-interfaces : 1
process reloading please wait_______
{"response":{"serviceTicket":"ST-1622-jsZ9bg6e2Jft2a6e6NQq-cas","idleTimeout":1800,"sessionTimeout":21600},"version":"1.0"}
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| Hote             | Adresse IP   | type                                          | Identificateur                       |
+==================+==============+===============================================+======================================+
| AP7081.059f.19ca | 10.1.14.3    | Cisco 3500I Unified Access Point              | cd6d9b24-839b-4d58-adfe-3fdf781e1782 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| Branch-Access1   | 10.2.1.17    | Cisco Catalyst 29xx Stack-able Ethernet Switch | 26450a30-57d8-4b56-b8f1-6fc535d67645 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| CAMPUS-Access1   | 10.1.12.1    | Cisco Catalyst 3850-48U-E Switch              | 5b5ea8da-8c23-486a-b95e-7429684d25fc |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| CAMPUS-Core1     | 10.1.7.1     | Cisco Catalyst 6503 Switch                    | 30d39b18-9ada-4148-ad6c-2ee20975b845 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| CAMPUS-Dist2     | 10.1.11.1    | Cisco Catalyst 4507R plus E Switch            | 4af8bf34-295f-46f4-97b7-0a2d2ea4cf22 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| CAMPUS-Router1   | 10.1.2.1     | Cisco 4451 Series Integrated Services Router  | 9712ab62-6140-43fd-b1ee-1b07d1fb67d7 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| CAMPUS-Router2   | 10.1.4.2     | Cisco 4451 Series Integrated Services Router  | 55450140-de19-47b5-ae80-bfd741b23fd9 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
| Campus-WLC-5508  | 10.1.14.2    | Cisco 5508 Wireless LAN Controller            | ae19cd21-1b26-4f58-8ccd-d265deabb6c3 |
+------------------+--------------+-----------------------------------------------+--------------------------------------+
_____ end
```

Figure 3.12 : Résultat du choix Network Device

La figure 3.12 montre les informations de Network Device que l'utilisateur veut récupérer, le tableau les Hote et les adresse IP et son type cisco et les identifications

```
#### script quand on choisi la command 2-hosts ####

elif y == 2:
    print("process reloading please wait_______")

    ticket=getTicket()
    url="https://sandboxapicem.cisco.com/api/v1/host"
    headers = {"X-auth-Token": ticket}

    device = []
    try:
        resp= requests.get(url,headers=headers,verify = False)
        response_json = resp.json()
        device = response_json["response"]
    except:
        print("something wrong, cannot get hosts information")

    device_list= []
    for item in device:
        device_list.append([item["hostIp"],item["hostType"],item["connectedNetworkDeviceIpAddress"]])
    print(tabulate(device_list,headers=['host IP','type','connected to network device'],tablefmt="grid"))
    print("_____ end")
```

Figure 3.13 : programme du choix host

```
#### script quand on choisi la command 3-interfaces ####

elif y == 3:
    print("process reloading please wait_______")

    ticket=getTicket()
    url="https://sandboxapicem.cisco.com/api/v1/interface"
    headers = {"X-auth-Token": ticket}

    device = []
    try:
        resp= requests.get(url,headers=headers,verify = False)
        response_json = resp.json()
        device = response_json["response"]
    except:
        print("something wrong, cannot get interface information")

    device_list= []
    for item in device:
        device_list.append([item["deviceId"],item["interfaceType"],item["status"]])
    print(tabulate(device_list,headers=['Id','type','status'],tablefmt="grid"))
    print("_____ end")
```

Figure 3.14 : programme du choix interface

-Par la suite nous avons écrit le script pour les deux autres API hot et interface , c'est presque le même script, la différence c'est l'adresse URL et les information que nous voulons récupérer dans les list device item.

nous avons le résultat des autres variables, il suffit juste de choisir le numéro qui correspond au résultat voulu, les figures (3.15 et 3.16) montrent l'affichage selon les choix 2 et 3 :

```
================== RESTART: C:\Users\HP\Desktop\programmeee.py ==================
1-devices, 2-hosts, 3-interfaces : 2
process reloading please wait_______
{"response":{"serviceTicket":"ST-1623-ZghXNLUlKogDv3QCl2cC-cas","idleTimeout":1800,"sessionTimeout":21600},"version":"1.0"}
+-------------+----------+------------------------------+
| host IP     | type     | connected to network device  |
+=============+==========+==============================+
| 10.1.15.117 | wireless | 10.1.14.3                    |
+-------------+----------+------------------------------+
| 10.2.1.22   | wired    | 10.2.1.17                    |
+-------------+----------+------------------------------+
| 10.1.12.20  | wired    | 10.1.12.1                    |
+-------------+----------+------------------------------+
_____________________________ end
```

Figure 3.15 : Résultat de choix host

Dans la figure 3.15 de choix 2 ''host'' nous avons récupérer les informations sur le hoste nous avons eu : le host IP, son type et a quoi il est connecter.

```
================== RESTART: C:\Users\HP\Desktop\programmeee.py ==================
1-devices, 2-hosts, 3-interfaces : 3
process reloading please wait_______
{"response":{"serviceTicket":"ST-1624-yNA4qIbP4t0eTE2O9hnW-cas","idleTimeout":1800,"sessionTimeout":21600},"version":"1.0"}
+--------------------------------------+----------+----------+
| Id                                   | type     | status   |
+======================================+==========+==========+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Virtual  | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Virtual  | up       |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
| 26450a30-57d8-4b56-b8f1-6fc535d67645 | Physical | down     |
+--------------------------------------+----------+----------+
```

Figure 3.16 : Résultat de choix interface

La figure 3.16 montre le résultat de choix 3 ‘’interface ‘’ nous avons recuper les informations sur les ID, son type physique ou virtuel et la situation des informations down/up.

III.9.4 Présentation de l’organisme d’accueil « ALGÉRIE TÉLÉCOM »

C’est une société publique opérant sur le marché des Télécommunications, Réseaux et Services de Communications électroniques. Elle offre une gamme complète de services de voix et de données aux clients résidentiels et professionnels. L’activité majeure d'Algérie Télécom est de :

- Fournir des services de Télécommunication permettant le transport et l'échange de la voix, de messages écrits, de données numériques, d'informations audiovisuelles.
- Développer, exploiter et gérer les réseaux publics et privés de Télécommunications.
- Établir, exploiter et gérer les interconnexions avec tous les opérateurs des réseaux.

Avec la mondialisation et l’explosion des nouvelles technologies, et afin de garder le rythme du développement technologique, ALGÉRIE TÉLÉCOM s’est engagée dans le monde des nouvelles TIC afin d’accroître l'offre et la qualité de services et facilite l'accès aux services de télécommunication à leurs clients. Notre projet est un exemple de l’adaptation des nouvelles technologies et de changement dans la manière d’offrir les services d’une part, la gestion et la manipulation d’autre part. Un projet est, généralement, considéré comme réussi s’il atteint les objectifs prédéfinis dans les délais et avec les ressources convenues.

Figure 3.17 : ALGÉRIE TÉLÉCOM

III.10 Conclusion

Au cours de ce dernier chapitre, nous avons présenté les différentes étapes de réalisation de notre application qui nous ont permis de communiquer avec le contrôleur CISCO APIC-EM à travers une approche programmable du réseau par le biais des API REST dans le but d'avoir une visibilité complète en collectant les différentes informations du réseau et ses différentes ressources.

Conclusion générale

Dans le cadre de notre projet de fin d'études, l'idée principale du SDN, est de rendre le réseau programmable. L'objectif de notre projet de fin d'études était de matérialiser cette idée ou nous avons abouti à la programmation du réseau avec python et les API REST du contrôleur APIC-EM.

Le concept d'APIC EM est qu'une entreprise peut définir et mettre en œuvre des politiques de réseau sans avoir à se soucier des configurations.

L'automatisation du réseau est activée via un module de contrôleur SDN centralisé et programmable. En raison de la nature programmable du contrôleur, les applications peuvent être écrites pour atteindre les objectifs commerciaux en interconnectant avec le contrôleur et, par la suite, avec le réseau, via des API dans la direction nord « API REST ».

Notre travail consiste à utiliser ces API et le langage de programmation python afin de communiquer avec le contrôleur APIC-EM. En interrogeant ce dernier, nous pouvons avoir la visibilité complète du réseau par la collection et l'analyse des informations retenues sur le réseau et l'ensemble de ses ressources.

Cette visibilité nous permet de mieux surveiller le réseau et avoir une vue qui permet de prendre des décisions éclairées sur la manière de le configurer pour gérer et améliorer ses performances et même intervenir si un problème est rencontré.

Notre connexion avec le contrôleur s'est établie correctement et nous avons bien été authentifiées, ainsi, notre interrogation a été menée avec succès et nous avons récupéré toutes les informations que nous voulions avoir.

APIC-EM offre plusieurs avantages et contient plusieurs fonctionnalités que, malheureusement, nous n'avons pas pu en bénéficier en raison des droits d'accès limités. Pour cela, nous proposons comme perspectives d'améliorer et d'élargir la notion de la communication avec APIC-EM

Pour conclure, nous pouvons dire que notre PFE, réalisé au sein d'ALGÉRIE TÉLÉCOM était d'une grande opportunité pour nous. Ce qui nous a permis de vivre une expérience très riche durant laquelle nous avons pu développer et approfondir le savoir et le savoir-faire que nous avons acquis durant notre cursus.

Nous espérons que ce travail participera au développement des connaissances et l'accroissement du savoir et servira d'appui pour les futurs étudiants qui désirent effectuer des recherches dans ce domaine très vaste.

Bibliographie

[1] TR10: Software-Defined Networking, disponible à l'adresse suivante :

http://www2.technologyreview.com/news/412194/tr10-software-defined-networking/

[2] Benoit Petit , «SDN : principes et fonctionnement» , 01 Mars 2018, disponible à l'adresse suivante:

https://blog.wescale.fr/2018/03/01/sdn-principes-et-fonctionnement/#:~:text=Le%20principe%20du%20SDN%20est,des%20donn%C3%A9es%20sur%20le%20r%C3%A9seau

[3] Ihssane Choukri, Mohammed Ouzzif, Khalid Bouragba. Software Defined Networking (SDN): Etat de L'art. Colloque sur les Objets et systèmes Connectés, Ecole Supérieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologie d'Aix-Marseille (France), Juin 2019, CASABLANCA, Maroc. hal-02298874

[4] Traore Issa, Kouassi Brou Médard, et Atta Ferdinand, « Etude du nomadisme dans un Cloud éducatif administré par la technologie SDN/OpenFlow », Institut de recherches mathématique Université Félix Houphouët-Boigny, conférence WACREN 2016.

[5] ONF, « OpenFlow Switch Specification», Décembre 2014, disponible à l'adresse suivante :

https://fr.slideshare.net/sianhnguyen9/openflow-switchv150noipr-53136228

[6] Market Report , « SDxCentral SDN Controllers Report», 2015 Edition, disponible à l'adresse suivante:

https://www.sdxcentral.com/members/join/

[7] Jérôme Durand, « Le SDN pour les nuls », Cisco Systems,Montpellier, JRES 2015.

[8] Thomas Paradis, « Software-Defined Networkin », mémoire de Master, School of Information and Communication Technology KTH Royal Institute of Technology Stockholm, Sweden, le 20 Janvier 2014.

[9] Ouafae IFAKREN, « Software Defined Network », Rapport du semestre, hepia Genève 2015/2016.

[10] Aria Zhu,« Network Switches and tagged 100GbE Switch, 10GbE switch, data switch, openflow protocol, openflow switch », 27 juillet 2018 .disponible à l'adresse suivante :

http://www.cables-solutions.com/whats-openflow-switch-how-it-works.html

[11] Karim IDOUDI, « implémentation d'un plan de contrôle unifié pour les réseaux multicouches IP /DWDM », rapport de projet présenté comme exigence partielle de la maîtrise en génie électrique, université du Québec à Montréal, Mai 2014

[12] Enric Caceres, « Le Protocole OpenFlow dans l'Architecture SDN », EFORT 2016.
[13] Maxence Tury, « Les risques d'OpenFlow et du SDN », ANSSI 2014.
[14] Anouar KOUACH,« Protocole OpenFLow », 28octobre 2015. Disponible à l'adresse suivante :

https://www.supinfo.com/articles/single/1010-protocole-openflow

[15] Open Network Foundation, «OpenFlow Switch Specification», version 1.4.0, Octobre 2013

[16] presentation de NETCONF disponibles à l'adresse suivante :

https://en.wikipedia.org/wiki/NETCONF

[17] Laurent ZUCCARLLIL, Mathieu LE LAOUENAN, Projet SER TÉLÉCOM LILLE 1, LILLE, 2010. Disponible à l'adresse suivante :

http://wapiti.enic.fr/commun/ens/peda/options/ST/RIO/pub/exposes/exposesser2010-ttnfa2011/lelaouenan-zuccarelli/intro.html

[18] K. Bhamre, «Ixia Blog Team»,28 Octobre 2014. Disponible à l'adresse suivante

https://www.ixiacom.com/company/blog/pcepbgp-ls-based-sdn-approach-ideal-choice-service-providers

[19] Chapter: BGP Link-State. disponible à l'adresse suivante :

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/irg-xe-16-book_chapter_01010101.html

[20] Douglas Mauro et Kevin Schmidt ,*Essential SNMP* (1st ed.). Sebastopol, CA: O'Reilly et Associates, juillet 2001.

[21] l'interface CLI definition . disponible à l'adresse suivante :

https://fr.wikipedia.org/wiki/Interface_en_ligne_de_commande

[22] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software architectures, thèse de doctorat, Université de Californie IRVINE, en 2000. Disponible à l'adresse suivante :

https://openclassrooms.com/fr/courses/3449001-utilisez-des-api-rest-dans-vos-projets-web/3501901-pourquoi-rest

[23] Appplication programming interface. Disponible à l'adresse suivante : https://www.lemagit.fr/definition/API#:~:text=Une%20API%20(Application%20Programming%20Interface,ou%20d'une%20autre%20application.

[24] Respresentation state transfer. Disponible à l'adresse suivante :

https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest

[25] Nicolas PANHALEUX ,« VXLAN : des VLANs dynamiques et routables pour les clouds », février 16, 2013.disponible à l'adresse suivante :

https://www.randco.fr/blog/2013/vxlan/?fbclid=IwAR0L9Qjf4WleL1yxutqn4lzV7kyHpwYBlQ5MRPs3YfA7NyKZj1fwHb2TCQU

[26] Réseau SD WAN. Disponible à l'adresse suivante :

https://www.forcepoint.com/fr/cyber-edu/sd-wan

[27] Abdelkader CHAARI , «Réseau de Transport Optique (OTN)», 09 mai 2017.disponible à l'adresse suivante :

https://www.supinfo.com/articles/single/4364-reseau-transport-optique-otn

[28] les avantages de la commutation OTN pour le transport SDN. Disponible à l'adresse suivante :
http://ww1.microchip.com/downloads/en/DeviceDoc/2150738_otn_in_sdn_273951.pdf

[29] Anouar KOUACH, «SDN: Software-Defined Networking - Concept», 27 Octobre 2015. Disponible à l'adresse suivante :

https://www.supinfo.com/articles/single/938-sdn-software-defined-networking-concept

[30] Yana Persky, « SDN Challenges in 5G Wireless Transport Networks - Part I»,October 24, 2019.

[31] Présentation du SD-WAN entreprise. Disponible à l'adresse suivante :

https://www.silicon.fr/hub/colt-hub/comprendre-la-virtualisation-des-reseaux-sdn-nfv-sd-wan

32] Définition de SD-WAN. Disponible à l'adresse suivante :

https://fr.wikipedia.org/wiki/SD-WAN

[33] WAN hybride . disponible à l'adresse suivante :

https://www.pyxya.fr/le-wan-hybride-et-le-cloud-compatibilite-limites-et-solutions/

[34] Définition de controleur SD WAN. Disponible à l'adresse suivante :

https://searchnetworking.techtarget.com/definition/SDN-controller-software-defined-networking-controller#:~:text=An%20SDN%20controller%20is%20an,switches%20where%20to%20send%20packets.

[35] Cisco APIC (Cisco Application Policy Infrastructure Controller). Disponible à l'adresse suivante:

https://www.cisco.com/c/dam/global/fr_ca/assets/pdf/c78-730594-00_cisco_application_policy_infrastructure_controller_ds_v3a_fr-ca.pdf

[36] Caractéristique et fonctionnalités. Disponible à l'adresse suivante:

https://www.cisco.com/c/fr_fr/products/cloud-systems-management/application-policyinfrastructure-controller-apic/index.html

[37] Cisco APIC-EN entreprise module .disponible à l'adresse suivante :

https://developer.cisco.com/docs/apic-em/#!overview/cisco-apic-em-northbound-interface

[38] Application d'APIC-EM. Disponible à l'adresse suivante :

https://docwiki.cisco.com/docwiki-eol.html

[39] Adam RADFORD, Network Automation with Plug and play (PnP), Cisco Community, Juin 2016

[40] Fonctionnalités d'APIC-EM. Disponible à l'adresse suivante :

https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/datasheet-c78-730594.html#:~:text=The%20Cisco%20%C2%AE%20Application%20Policy,simplifies%20and%20abstracts%20the%20network.

[41] les avantages de cisco APIC-EM. Disponible à l'adresse suivante :

https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/1-0-x/config-guide/b_apic-em_config_guide_v_1_0/b_apic-em_config_guide_v_1_0_chapter_01.html#:~:text=The%20Cisco%20APIC%2DEM%20platform,and%20costs%20through%20advanced%20automation

[42] Description des APIs REST de l'APIC-EM. Disponible à l'adresse suivante :

https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/application-policy-infrastructure-controller-enterprise-module/datasheet-c78-730594.html#:~:text=The%20Cisco%20%C2%AE%20Application%20Policy,simplifies%20and%20abstracts%20the%20network.

Annexes I :

Annexe I : Protocoles dans l'interface de sud (Southbound)

I.1 : SNMP (Simple Network Management Protocol)

C'est un protocole de gestion de réseau de base développé par l'IETF (Internet Engineering Task Force) à la fin des années 1980. Il permet aux administrateurs de gérer les équipements du réseau et de diagnostiquer leurs problèmes.

I.2: CLI (Command Line Interface)

Tire son nom du fait qu'il s'agit d'une interface utilisateur du système d'exploitation d'un ordinateur ou d'une application qui contient des lignes de commande. Une ligne de commande est un espace sur l'écran d'affichage dans lequel des commandes (instructions indiquant à un ordinateur de faire quelque chose) sont tapées par l'utilisateur. Les insuffisances de SNMP et CLI qui ont poussé les développeurs à introduire Netconf :

- SNMP n'est pas utilisé pour configuré les équipements de réseau, mais il a étais utilisé seulement pour la surveillance du réseau

- Les opérateurs utilise principalement des CLI propriétaire a fin de configuré leurs équipements.

I.3 : OpenFlow

I.3.1 : Les champs de table de flux

- Match fields : Utilisés lors de la recherche de l'entrée correspondante au paquet. Ils sont constitués essentiellement des entêtes des paquets et des ports d'entrées.

- Priority : C'est un champ de 16 bits qui contient la priorité relative à chaque entrées de table.

- Counters : Les compteurs sont des éléments principaux des statistiques OpenFlow. Les compteurs comptent le nombre des paquets correspondant à la table du flux qui passe par l'élément OpenFlow.

- Instructions : Représentent l'ensemble des instructions OpenFlow qui servent à modifier le traitement que va subir un paquet si une correspondance se produit. Les instructions supportées sur un paquet sont :

- Output : Envoyer un paquet par un port spécifié.

- Set-field : Modifier la valeur d'un champ d'en-tête spécifique, comme le TTL (Time To Live), l'adresse MAC (Media Access Control), ou l'ID du VLAN (Virtual Local Area Network), etc…

- Drop : Supprimer le paquet.

- Group : Traiter le paquet selon un groupe spécifique (entrée spécifique de la table de groupe).

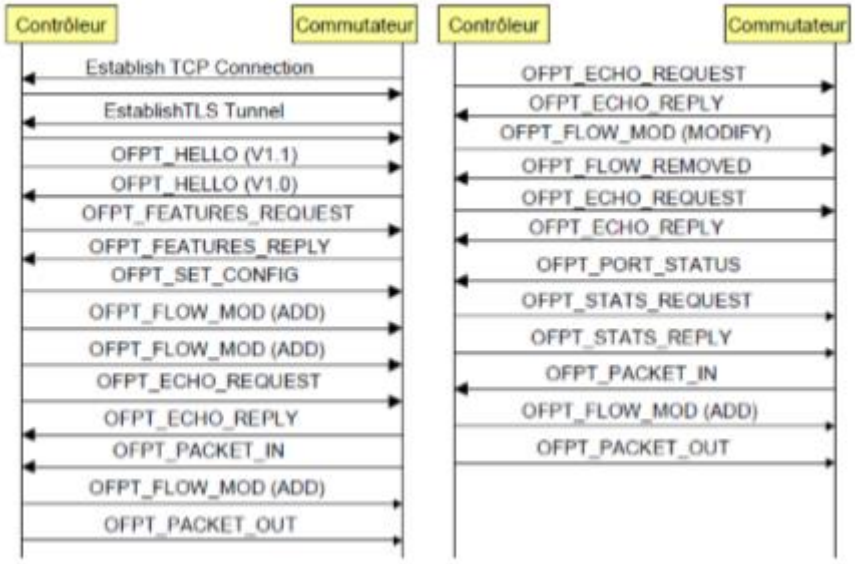
- Timeouts : Le temps maximum ou le temps d'inactivité avant que le flux ne soit expiré par le commutateur.

- Cookie : Une valeur de données de 64 bits choisie par le contrôleur peut être utilisée pour filtrer les statistiques de flux, la modification du flux et la suppression du flux, non utilisées lors du traitement des paquets.

- Flags : flags modifie la façon dont les entrées de flux sont gérées.

I.3.2 : Messages de protocoles OpenFlow

La figure suivante montre des échanges de message OpenFlow entre contrôleur et commutateurs après l'établissement de la connexion TCP et du tunnel TLS, initiés par le commutateur.



Annexe II :

La segmentation de réseau traditionnelle a été fournie par des VLAN (Virtual LAN) normalisés sous le groupe IEEE 802.1Q. Les VLAN fournissent une segmentation logique du réseau avec une évolutivité maximal de 4096 VLAN. Ce nombre est devenu un facteur limitant pour les départements informatiques et les fournisseurs de Cloud le faite qu'ils construisent de grands centres de données.

Cisco, en partenariat avec d'autres fournisseurs, a proposé la norme VXLAN (Virtual Extensible LAN) à l'IETF en tant que solution aux défis réseau du centre de données posés par la technologie VLAN traditionnelle. La norme VXLAN assure le placement élastique de la charge de travail et une plus grande évolutivité de la segmentation couche 2 requise par les demandes d'application actuelles (16 millions segments).

Avant, les réseaux VXLAN fonctionnent avec le modèle « Flood and Learn ». Dans ce modèle, la collecte des informations sur l'hôte final et la découverte de VTEP (VxLAN Tunneling EndPoint) se fait à travers le multicast. MP-BGP EVPN change ce modèle. Il s'introduit comme un plan de contrôle pour les réseaux VxLAN.

MP-BGP a été étendue de diverses façons. Dans le contexte de la virtualisation de réseau. MP-BGP fournit un répertoire de données de toutes les adresses MAC et IP des périphériques (logiques et physique) derrière un VTEP dans un segment de réseau VXLAN. En conséquence, MP-BGP permet une structure claire de la numérotation et de la nomenclature normalisées des destinations VXLAN, ce qui élimine la nécessité d'inonder afin d'obtenir des informations d'adressage.

Annexe III : Mise en œuvre de l’application

III.1 : JSON (JavaScript Object Notation)

JSON est le format le plus utilisé et le plus pratique. Comme l'indique son nom il s'agit d'une représentation des données sous forme d'objet JavaScript, Il est principalement utilisé pour transmettre des données entre un serveur et des applications Web. Les services Web et les API utilisent le format JSON pour fournir des données publiques, et il est souvent utilisé comme le principal format de sérialisation de données lors de la conception des systèmes qui respectent les principes REST.