REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES

Faculté de **T**echnologie

Département **I**ngénierie des **S**ystèmes **E**lectriques

Mémoire de Master

Présenté par

ARABI Nadjoua

YAMOUTENE Naima

Filière : Télécommunications

Spécialité : Réseaux et **T**élécommunications

Etude comparative des méthodes de routage traditionnel et SDN

Soutenu le 15 /12/2020 devant le jury composé de :

Messaoudi	Noureddine	MCA	Université	Président
Meraihi	Yassine	MCA	Université	Examinateur
BAICHE	Karim	MCA	UMBB	Rapporteur

Année universitaire : 2019/2020

Résumé

Le SDN – Software Defined Networking – est certainement le sujet chaud qui agittelle monde du réseau depuis ces dernières années. La technologie SDN a déclenché unchangement radical à long terme dans la conception des réseaux, le marché s'estrapidement approprié le SDN comme ensemble de solutions/architectures permettant desupprimer les frontières existantes entre les mondes des applications et du réseau. Alorsque le déploiement d'applications est toujours plus aisé et dynamique, notamment grâce àla virtualisation et au Cloud.
Dans ce mémoire, nous avons évalué l'étude comparative de deux architectures réseaux àsavoir, réseautage traditionnelle et le SDN.

Mots clés :

Réseau traditionnelle, routage, réseau SDN, mininet, miniedit

ملخص

SDN - الشبكات المعرفة بالبرمجيات - هي بالتأكيد الموضوع الساخن الذي يثير الاهتمام في عالم الشبكات في السنوات الأخيرة. أحدثت تقنية SDN تغييرًا جذريًا طويل المدى في تصميم الشبكة، سرعان ما تبنى السوق الـ"SDN" كمجموعة من الحلول/البنى لإزالة الحدود الحالية بين عوالم التطبيق والشبكة . في حين أن نشر التطبيقات يكون دائمًا أسهل وأكثر ديناميكية، لا سيما بفضل المحاكات والسحابة.
في هذه المذكرة، قمنا بتقييم دراسة مقارنة لمخططي شبكتين، وهما الشبكات التقليدية وشبكات الـSDN.

الكلمات المفتاحية

الشبكات التقليدية .SDN شبكات . تطبيق البرمجة .mininetتوجيه الشبكة

Abstract

The SDN - Software Defined Networking - is certainly the hot topic that has beenstirring the network world for the past few years. SDN technology has triggered a radical change over the long term in network design, the market quickly appropriated the SDN as a set of solutions/architectures to remove the existing boundaries between the worlds of applications and the network. While deploying applications is easier and more dynamic, especially with virtualization and the Cloud.
In the context of our project, we evaluate the comparative study of two network architecture: traditional Networking and SDN.

Keywords :

Traditionnel network, sdn network, mininet,miniedit,routing

Dédicaces

A mes chers parents

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma

Considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance. Que ce travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices. Puisse Dieu, le Très Haut, vous accorde santé, bonheur et longue vie.

A mes chères sœurs et frère

Je ne pourrais d'aucune manière exprimer ma profonde affection et mon immense gratitude pour tous les sacrifices consentis, votre aide et votre générosité extrêmes ont été pour moi une source de courage, de confiance et de patience. Qu'ALLAH vous protège et vous accorde santé, longue vie et bonheur.

A toutes mes amies

En souvenir des moments agréables passés ensemble, veuillez trouver dans ce travail l'expression de ma tendre affection et mes sentiments les plus respectueux avec mes vœux de succès, de bonheur et de bonne santé.

ARABI Nadjoua

Dédicaces

A ma maman même si aucune dédicace ne saurait exprimer ma profonde gratitude et ma vive reconnaissance envers tous les sacrifices qu'elle a faits pour moi.

A mon cher mari, et A mon Superbe trésor, l'adorable ange : Zahra
Vous êtes ma plus grande richesse, Je vous aime !

A mes frères et mes sœurs, mon neveu et mes nièces,
Ainsi qu'à toute ma famille : petits et grands

YAMOUTENENaima

Remerciement

Ma reconnaissance ainsi que ma dévotion se dirigent tout d'abord versmon Créateur, le Tout Puissant, qui m'a offert la vie et le bonheur quiva avec. Dieu, qui m'a toujours guidé lors de ma route, Dieu qui m'adopté d'une force, d'un courage et d'une patience afin d'affronter mesobstacles et de mener à bien.

Je tiens à exprimer ma gratitude et mes respects les plus sincères à mon encadreur, monsieur BAICHE Karim, pour avoir accepté, si généreusement, de m'encadrer et pour l'aide qu'il a bien voulu m'accorder tout au long de mon travail, moyennant ses critiques constructives et ses suggestions pertinentes.

Mes remerciements vont aussi aux membres du jury pour l'honneur D'avoir voulu examiner et évaluer ce travail.

Je remercie également mon mari pour ses précieux conseils et son soutien tout au long de mes études, ainsi que ma mère, pour son amour et son engagement inconditionnel. J'espère qu'elle sera toujours fière de moi.

Si ce document a pu voir le jour, c'est grâce aux actions conjuguées de Plusieurs personnes à qui je dis sincèrement merci.

NAIMA

Liste des figures

Liste des tableaux

Table de matières

Introduction Générale

Les technologies de l'information et de la communication ont été la révolution la plus importante et innovante qui a marqué ces dernières décennies, pour permettre aux individus de communiquer toujours plus librement à tout instant et à distance. Cette opération est faite grâce aux routages qui est une technique qui assure le bon acheminement des datagrammes entre les équipements d'interconnexions à l'aide d'un ensemble de protocoles de routage basés sur des algorithmes qui prennent en charge des facteurs trèsimportants comme la durée moyenne de transmission, la charge du réseau, la langueur totale du message, l'état de la liaison et le débit pour améliorer ainsi l'efficacité du routage. Le principal périphérique de routage est le routeur.

Les techniques de communication à distance sont devenues indispensable dans notre vie quotidienne pour faciliter les opérations et les activités, mais avec l'augmentation de nombre d'utilisateurs et la vitesse de développement des télécommunications, ces méthodes traditionnelles ne sont plus suffisantes, ce qui a poussé de les rendre plus professionnelles pour l'adaptation aux besoins actuels.

Un paradigme alternatif, cependant, attire actuellement l'attention d'un nombre croissant d'acteurs du marché qui est le fameux SDN (les réseaux définis par logiciel).

Au terme de notre mémoire de Master, le stage de fin d'étude est une étape permettant de mettre en application l'ensemble des connaissances acquises dans le cadre d'un projet industriel. C'est avec grand enthousiasme que nous avons choisis d'intégrer Algérie télécom afin de réaliser notre projet de fin d'étude, mais malheureusement avec lapandémie (COVID-19), nous n'avons pas pu le faire. Le projet à réaliser s'intitule "Etude comparative des méthodes de routage traditionnel et SDN" qui agit comme une étude comparative entre le routage traditionnel dont chacun de ses équipementsregroupe le plan de données, le plan de contrôle et avoir étudié aussi ces faiblesses. Une amélioration est faite par l'introduction de SDN, qui se caractérise par laséparation physique du plan de contrôle et du plan de données, ainsi quela centralisation des fonctions de contrôle. Administration mise à part, lefonctionnement d'un routeur traditionnel est reproduit grâce à deux typesd'outil distincts.

Notre mémoire est composé de 3 chapitres. Dans le chapitre 1, nous aborderons des généralités sur les techniques de routage traditionnel. Le deuxième chapitre, traite les réseaux définit par logiciel SDN.Nousdonnons une vuegénérale surles technologies des réseaux qui sont les motivations de software -Defined –network (SDN), la technologie SDN et ses avantages avec les composants importants qui constituent les réseaux définit par logiciel.

Finalement nous terminonsavec une analyse comparative entre le réseau traditionnel et réseau SDN.

Chapitre 1

Chapitre 1 : Les Techniques de Routage Traditionnel

1.1. Introduction

Le routage est une méthode d'acheminement où les paquets entre les différents réseaux seront livrés d'un nœud à un autre. Dans ce chapitre nous abordons les caractéristiques principales de routage et les algorithmes utiliser pour l'organiser à l'aide des mécanismesqui sont appelés des protocoles de routage pourétablir des tables de routage afin que le trafic atteindre sa destination finale. Ensuite, nous présentons les avantages et les inconvénients d'un réseau traditionnel après l'étude de son architecture. Les limitations de ce dernier nous ont poussées de trouver des autres solutions qui répond à nos besoins d'où la création des réseaux programmables.

1.2. Définition duroutage

Le routage est la fonction qui s'occupe de diriger les données réseaux à travers différents segments. Il va les diriger jusqu'au prochain point de route. Cette fonction emploie des algorithmes de routages et des tables de routage (carte routière en quelque sorte). Principal périphérique de routage est le routeur. Il utilise les adresses IP pour diriger correctement les paquets d'un réseau ou segment à un autre. Il doit maintenir sa table de routage à jour et connaître les changements effectués sur les autres appareils par lequel il pourrait faire transiter le paquet. Pour remplir et mettre à jour la table de routage, il y a deux manières de faire, on peut le faire soit manuellement soit de manière dynamique, en employant des processus tournant sur le réseau.

1.3. Architecture des réseauxtraditionnels

À l'intérieur de chaque périphérique réseau et de sécurité, à savoir chaque commutateur, routeur et pare-feu, vous pouvez séparer le logiciel en quatre couches ou plans.

- ✓ **Transfert :** A pour rôle d'acheminer les paquets réseau. Il est optimisé afin dedéplacer les données aussi rapidement que possible.
- ✓ **Contrôle :** Ce plan décide de la direction des flux réseau, il décode les protocoles pourassurer la fluidité du trafic. Celui-ci se trouve dans un logiciel dédié intimement reliéau hardware.
- ✓ **Services :** Cette couche n'existe pas dans les switches. Un exemple de son utilisationest le pare-feu, elle permet d'exploiter et de déployer des services plus complexes surl'appareil en question.
- ✓ **Gestion :** Le plan Gestion fournit les instructions de base indiquant comment lepériphérique réseau doit interagir avec le reste du réseau. On y accède par l'interfacede ligne de commande (CLI) pour insérer la configuration du périphérique. [1]

La figure ci-dessous (figure 1.1) montre l'architecture des équipements réseau traditionnels et leur fonctionnement. Ces réseaux traditionnels sont désavantagés par les points suivants :

- Complexité : l'ajout ou la modification d'équipements et l'implémentation des politiques réseaux sont complexes, longues et peuvent être source d'interruptions de service. Ce qui décourage les modifications et l'évolution du réseau.
- Passage à l'échelle : l'impossibilité d'avoir un réseau qui s'adapte au trafic a obligé les opérateurs a sur-provisionner leurs réseaux ce qui ajoute une complexité de gestion sur le plan de contrôle
- Dépendance aux constructeurs : les constructeurs réalisent des produits avec des durées de vie limitées et un manque de standard, d'interfaces, ouvertes. Ce qui restreint les opérateurs réseaux d'adapter le réseau à leurs propres besoins [2].

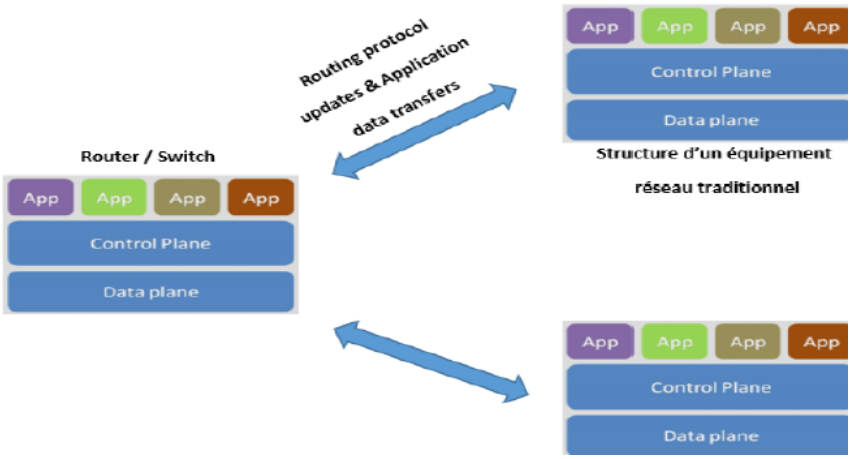


Figure 1.1 : Fonctionnement des structures des équipements réseau traditionnels

1.4. Méthodes de routage

1.4.1. Routage Direct

Le routage direct il s'agit de délivrer un datagramme à une machine destinatrice raccordée au même réseau (même LAN) que la machine émettrice. L'émetteur trouve l'adresse physique du correspondant à l'aide le protocole (ARP), encapsule le datagramme dans une trame et l'envoie.

Dans ce cas, un datagramme IP peut être émis directement, sans passer par un routeur c'est ce qu'on appelle la remise directe. [3]

1.4.2. Routage indirect

Si la machine de destination se trouve sur le même réseau physique mais sur un sousréseau différent, c'est le routage indirect qui sera illustré. Ce qui implique qu'un routeur estnécessaire pour acheminer le trafic entre les deux sous- réseaux.

Le routage indirect Permet de transmettre les datagrammes d'une machine émettrice à une autre machine qui ne se trouve pas dans mêmeréseau, passant par un routeur, ainsi il est nécessaire de déterminer vers quel routeur envoyer un datagramme IP en fonction de sa destination finale.

Dans le cas du routage indirect le choix du routeur vers lequel va être envoyé le datagramme IP se fait à l'aide des tables de routages ou par routage par défaut [4].

1.4.2.1. Table de routage

Est une structure de donnée utilisée par un routeur ou un ordinateuren réseau et qui associe des préfixes à des moyens d'acheminer les datagrammes vers leursdestinations. Ces tables contiennent les informations relatives aux différentes destinations possibles et à la façon de les atteindre. Machines et routeurs possèdent tous des tables de routage.

D'un point de vue fonctionnel une table de routage contient des paires d'adresses du type (D, R) où D est l'adresse IP d'un réseau destination et R l'adresse IP du routeur suivant sur le chemin menant à cette destination. Tous les routeurs mentionnés dans une table de routage doivent bien sûr être directement accessibles à partir du routeur considéré. Cette technique, dans laquelle un routeur ne connaît pas le chemin complet menant à une destination, mais simplement la première étape de ce chemin, est appelée routage par sauts successifs (next-hoprouting). [5]

Une table de routage ressemble à l'exemple ci-dessous :

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet1/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
S    192.168.3.0/24 is directly connected, FastEthernet0/1
S    192.168.4.0/24 is directly connected, FastEthernet0/1
C    192.168.5.0/24 is directly connected, FastEthernet0/0
S    192.168.6.0/24 is directly connected, FastEthernet0/1
C    192.168.7.0/24 is directly connected, FastEthernet1/1
```

Figure 1.2 : Table de routage

La table de routage fonctionne en mémoire vive (RAM) et comprend des informations telles que :

- Les réseaux directement connectés - pour tout réseau directement connecté à une interface.
- Les réseaux distants joignables - pour tout réseau qui n'est pas directement connecté au routeur.

Des informations détaillées à propos de ces destinations incluent l'adresse du réseau, son masque et l'adresse du prochain saut (routeur) vers la destination.

La commande show ip route affichent la table de routage. [6]

1.4.3. Routage par défaut

Si la table de routage IP ne contient aucune entrée faisant référence à la destination du datagramme, celui-ci est alors envoyé vers une passerelle dite passerelle par défaut (DefaultGateway), dont l'adresse est généralement stockée dans la table de routage. [7]

1.5. Les types de routage

Il existe trois types de routage statique,dynamique et hybride :

1.5.1. Routage statique

Le routage statique il le fait par l'administrateur, il faut inscrire manuellement dans la table de routage. Il doit gérer toutes les routes créées de chaque unité de routage du réseau. Ces routes statiques sont utilisées pour des raisons de sécurité en cas de dysfonctionnement. Les opérations de routage statique s'articulent comme suit :

- L'administrateur réseau configure la route.
- Le routeur insère la route dans la table de routage.
- Les paquets sont acheminés à l'aide de la route statique. [5]

1.5.1.1. Avantages du routage statique

- Le routeur n'a pas à consacrer une partie de ses ressources à l'entretien d'un protocole de routage (CPU, mémoire)
- Économie de la bande passante
- Connaissance du chemin à l'avance
- La sécurisation.

Le routage statique est utilisé dans les petits réseaux, les réseaux privés connecté à l'internet via un seul fournisseur d'accès.

1.5.1.2. Inconvénients du routage statique

Toute modification de topologie requiert l'intervention de l'administrateur ce qui peut rapidement devenir pesant la panne d'un équipement ou d'une interface est une modification de topologie accidentelle, non planifiée. Le temps d'indisponibilité est fonction du délai de prise en compte du défaut par l'administrateur.

1.5.2. Routage dynamique

Ce type de routage se fait à l'aide d'un ensemble des protocoles de routage, un routeur partage les informations concernant les réseaux connait par d'autre routeur qui utiliseles mêmes protocoles.

A chaque route unecorrespondance, l'adresse réseaux et l'adresse de prochaine saut et motionne le mode d'apprentissage de la route (S pour statique, C pour directement connecter, R pour le RIP).

Les correspondances sont maintenues à jour au fur et à mesure de la vie du réseau. [7]

1.5.2.1. Avantages du routage dynamique

- Maintenance réduite par l'automatisation des échanges et des décisions de routage
- Modularité et une flexibilité accrue, il est plus facile de faire évoluer le réseau avec un réseau qui se met à jour automatiquement.
- Sa performance et sa mise en place ne dépendent pas de la taille du réseau.

1.5.2.2. Inconvénients du routage dynamique

- Il consomme de la bande passante de par les messages que les routeurs s'envoient périodiquement sur le réseau.
- La diffusion automatique des messages sur le réseau peut constituer un problème de sécurité car un attaquant peut obtenir des informations sur la topologie du réseau simplement en écoutant et en lisant ces messages d'information du protocole de routage et même en créer afin de se faire passer pour un membre du réseau.
- Le traitement des messages réseau et le calcul des meilleures routes à emprunter représentent une consommation de CPU et de RAM supplémentaire qui peut encombrer certains éléments du réseau. [5]

1.5.3. Routage hybride

Ce type de routage est un mélange entre les deux types précédents, ilramasse les avantages de ces derniers alors il est plus amélioré.

1.5.3.1. Routage à système autonome

Un système autonome est constitué par un ensemble de réseaux interconnectés par desrouteurs. Les systèmes autonomes sont reliés entre eux par des routeurs externes. Un système autonome correspond à un groupe de routeurs dépendant d'une même responsabilitéadministrative du point de vue du routage et appliquant une politique de routage unique. On distingue donc les protocoles de routage interne au sein d'un système autonome et des protocoles externes qui concernent le trafic entre systèmes autonomes. Les protocoles de routage externe privilégient les informations d'accessibilité par rapport aux informations de topologie du réseau. Un réseau IP appartient à un seul système autonome. Le principal protocole de routage externe est bien le BGP. L'internet est composé donc d'un ensemble de systèmes autonomes reliés entre eux par des routeurs externes. Un système autonome utilise un protocole de routage interne tel que RIP, OSPF, ISIS ou IGRP. Les numéros de systèmes autonomes sont attribués par le NIC (Network Information Center). Ce numéro sur deux octets est appelé "Autonomes System Number". Les numéros du système autonome de 65412 à 65535 sont privés.

1.6. Les protocoles de routage :

Pour que 2 routeurs se partagent ensemble leur table de routage, il faut bien entendu que ceux-ci soient configurés sur le même protocole de routage.

Un protocole de routage sert à :

- Gagner du temps en évitant de devoir configurer manuellement toutes les routes sur chaque routeur.
- Améliorer la stabilité du réseau en choisissant chaque fois la meilleure route.

La figure suivant présent les déférant protocoles de routage

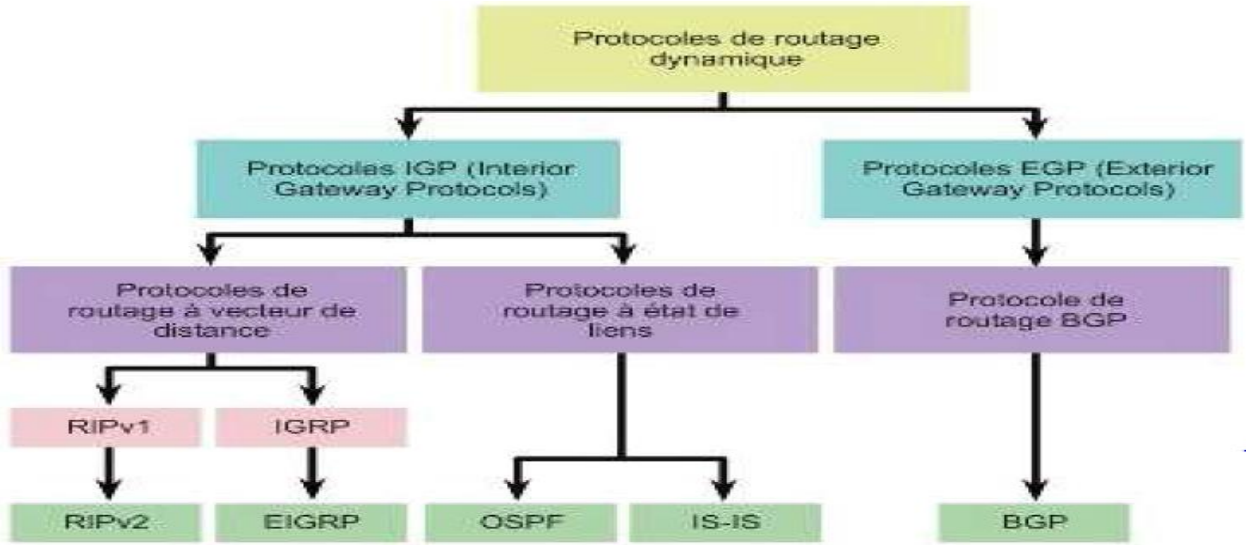


Figure 1.3 : Classification des protocoles de routage

1.7. Protocole de routage externe (EGP) :

EGP a été le premier protocole de routage inter domaine. Personne n'en est satisfait, mais il répondait à l'époque à un besoin. Et malgré les problèmes d'EGP, l'internet l'a trainé pendant de nombreuses années et s'est débrouillé pour survivre avec.

Aujourd'hui, ce protocole est obsolète et remplacer par BGP. BGP est un protocole de routage par vecteur distance, mais au lieu de transmettre le cout vers une destination, un routeur BGP transmet la suite des numéros de systèmes autonomes rencontrés sur le chemin vers l'adresse IP de destination. Chaque routeur BGP est configuré avec des algorithmes de classement de route. Le routeur calcul sa route préférer vers une adresse IP particulière et annonce cette route dans son message de routage à ses voisins BGP.

Dans BGP, une politique peut être mise en œuvre à l'aide d'une information configuré manuellement qui peut permettre à un routeur BGP de classer les routes par ordre de préférence. Dans la mesure où BGP n'annonce pas de cout vers une destination, il doit exister un autre moyen pour déterminer quelle route il va choisir parmi plusieurs. BGP autorise des politiques très complexes. Il suppose qu'il y a une méthode locale de gestion d'un routeur BGP pour construire une fonction qui prendra comme entré toute l'information annoncé dans un message de mise à jour BGP au sujet d'une destination particulière et qui aura comme sortie un nombre. Une fois les déférentes routes possibles associées aux nombres, on peut les comparer. La route préférer est celle qui correspondant au nombre le plus petit. [8]

1.8. Les protocoles de routage interne

Les protocoles de routage internes permettent une configuration automatique des tables de routage des routeurs à l'intérieur d'un même système autonome. Les routeurs déterminent le plus court chemin pour atteindre un réseau distant. Les protocoles de routage internes nécessitent une configuration minimale du routeur notamment en ce qui concerne les

annonces de routes initiées par ce routeur (ex. réseaux directement accessibles par une interface du routeur, annonces statiques ...).

Deux types de protocole de routage interne existent : les protocoles à vecteur de distance et les protocoles à état de liens.

Les premiers attribuent un coût à chaque lien en fonction de divers paramètres (type du lien...), tandis que les seconds calculent le chemin le plus court en comptant le nombre de sauts pour atteindre le préfixe de destination. [5]

1.8.1. Les protocoles à vecteur de distance

Les protocoles de routage à vecteur de distance transmettent régulièrement des copies d'une table de routage d'un routeur à l'autre.

Chaque routeur reçoit une table de routage de son voisin immédiat. Le même processus est répété étape par étape dans toutes les directions entre les routeurs immédiatement voisins.

Ainsi, le protocole cumule les distances réseau afin de à jour la base de données contenant les informations sur la topologie du réseau.

Les protocoles à vecteur de distance ne permettent pas à un routeur de connaître la topologie exacte d'un réseau [9]

Un protocole à vecteur de distances est de nature distribuée, itérative et asynchrone. De nature distribuée parce que les calculs se font au niveau de chaque nœud, à partir des informations fournies par les voisins immédiats, et les résultats sont partagés de la même manière. De nature itérative, car ce processus se répète jusqu'à ce qu'il n'y ait plus d'informations à échanger entre nœuds voisins. De fait, cet algorithme s'arrête de lui-même. Enfin, de nature asynchrone dans la mesure où il n'impose pas à tous les routeurs de travailler ensemble [10]

Les protocoles les plus connu sont :

1.8.1.1. Le protocole RIP

RTP (routing information protocole) c'est un protocole de routage très simple basé sur le vecteur distance s'appuyant sur l'algorithme de détermination des routes décentralisé **Bellman-Ford.**La métrique est calculée en fonction de nombre de saut pour atteindre la

destination. Un saut représente le nombre de routeur que doit traverser la donnée avant d'arriver à destination. Il existe deux versions de RIP :

- ✓ RIPv1 c'est le protocole le plus vieux, il supporte un maximum de 15 nœuds traversés. Il utilise les masques sous réseaux par défaut (classfull), les mises à jour de routage sont diffusées sur l'adresse (255.255.255.255)toutes les 30 secondes et ne prennent pas en charge le VLSM et l'authentification.
- ✓ RIPv2 c'est une version améliorée, a apporté les améliorations suivantes :
  Le support de VLSM et de l'authentification, La diffusion des mises à jourseffectue par multicast (224.0.0.9), permet d'utiliser des masques modifies (classes).

• Mode de fonctionnement

RIP a été conçu pour fonctionner dans un réseau de taille moyenne. Le protocole est basé sur un algorithme de distance vectorielle.

Chaque routeur qui intervient dans un protocole va établir une base de données. Cette base de données contiendra pour chaque réseau accessible, l'adresse du premier routeur intervenant dans l'acheminement de l'information, ainsi qu'un cout associé à cette route et un temps qui indique l'âge de cette information. Le gestionnaire du réseau doit associer à chaque segment un cout d'utilisation. Le protocole RIP limite à 15 les couts utilisables dans les bases de données.

Lorsque tous les segments sont équivalents un cout de 1 est associé à chaque segment. Si les segments du réseau ne sont pas tous équivalents, le gestionnaire du réseau donnera aux segments les plus lentes des métriques plus élevées pour privilégier les segments rapides. Dans tous les cas, un cout de 16 représente une longue infinie.

Tant que la topologie ne change pas, les métriques convergentes en un temps fini. Mais la topologie peut être changée pour plusieurs raisons, par exemple en cas de panne d'un routeur. Pour prendre en compte ces cas, les routeurs doivent en permanence entretenir leur base de données. Le protocole prévoit l'envoi de messages toutes les 30s vers tous les voisins du routeur.

En réception, le routeur met à jour les informations de sa base. Si un routeur ne reçoit pas d'information d'un de ces voisins pendant 3 intervalles de temps, les entrées concernant ce routeur sont invalidées. Ensuite, il en informe ses voisins. Il utilise pour cela un message indiquant que le cout est devenu infini.

Si un routeur est informé qu'un réseau est inaccessible, alors, durant un certain temps, toute information concernant l'accessibilité de ce réseau ne sera plus prise en compte. Une information d'inaccessibilité a donc une durée de vie qu'une autre mise à jour ne peut modifiée immédiatement. Cette durée appelée hold down, n'est pas toujours la même dans tous les routeurs, ce qui peut entrainer des incohérences dans les bases de données, certains routeurs acceptant la mise à jour, d'autre non. [11]

1.8.1.2. Le protocole IGRP et EIGRP

Protocole de routage de la passerelle intérieure. IGRP est un protocole de routage intérieur à vecteur de distance utilisé par les routeurs pour échanger des données de routage au sein d'un système autonome.

Il n'y a pas de limites de taille de réseau avec IGRP. Par contre il ne supporte pas différents masques de sous réseau, il est actualisé toutes les 90 secondes. Il a plus de critères que le protocole RIP, il peut aussi prendre en compte, la bande passante, le délai, la charge réseau. On peut même donner manuellement une priorité à chacune de ses conditions.

Le protocole EIGRP (Enhanced Interior Gateway Routing Protocol) est un protocole de routage à vecteur de distance amélioré et développé par Cisco.

Il constitue une version perfectionnée du protocole IGRP (Interior Gateway Routing Protocol) qui mit au point par Cisco et compatible avec ses produits.

Il a été spécifiquement entendu pour pallier les problèmes associés au routage dans de grands réseaux qui dépassaient la portée des protocoles tels que RIP. Il utilise l'algorithme DUAL (Diffusing Update Algorithme) qui a été développé à SRI International ce qui permet une meilleure convergence du réseau. [12]

- **Mode de fonctionnement :**

Le fonctionnement du protocole EIGRP est le suivant : le routeur qui est configuré pour utiliser le protocole EIGRP va garder en mémoire toutes les tables de routage de ses voisins dans une table nommée table de voisinage, ce qui permet en cas de défaillance du réseau de trouver très rapidement un chemin alternatif. Le protocole EIGRP ne fait pas de mise à jour périodique de ces tables, ce qui peut s'avérer être un inconvénient. En effet si une route change pour une quelconque raison, le routeur utilisant EIGRP en tant que protocole ne sera pas averti aussi rapidement que les routeurs utilisant d'autres protocoles, mais si le protocole EIGRP ne permet pas aux routeurs de faire des mises à jour périodiques, il leur permet

néanmoins d'envoyer de partielles mises à jour lorsque la distance pour une route change. Ces informations sur la route qui a changé sont alors uniquement envoyées vers les routeurs qui ont besoin de ces informations

1.8.2. L'état de lien

Un protocole de routage à état de liens utilise la connaissance de la topologie du réseau et du coût des différents liens pour ses calculs. Chaque nœud du réseau diffuse à l'ensemble de ses homologues l'identité des nœuds et la valeur des liaisons auxquelles il est rattaché. Cette diffusion d'état de liens peut s'effectuer sans que les différents nœuds aient initialement connaissance de l'existence de tous les autres nœuds du réseau. Un nœud doit simplement connaître l'identité de ses voisins immédiats ainsi que la valeur des liaisons qui le lient à eux. Il peut découvrir le restant de la topologie du réseau grâce aux informations d'état de liens qui lui sont communiquées par ses voisins. Après un certain temps, tous les nœuds disposent des mêmes informations sur le réseau, chacun pouvant utiliser l'algorithme individuellement et obtenir ainsi les mêmes résultats que ses homologues. [13]

1.8.2.1. Le protocole OSPF

Le protocole OSPF (Open Shortest Path First) est un protocole de routage à état de liens, il a été développé par l'IETF7 (Internet Engineering Task Force) pour succéder à RIP. OSPF fait partie de la deuxième génération de protocoles de routage. Beaucoup plus complexe que RIP, mais au prix de performances supérieures, il utilise une base de données distribuée, qui garde en mémoire l'état des liens. Ces informations forment une description de la topologie du réseau et de l'état des nœuds, qui permet de définir l'algorithme de routage par un calcul des chemins les plus courts. [14]

- **Mode de fonctionnement :**

L'envoi de la table ne se fait pas de manière régulière, donc une meilleure utilisation de la bande passante. En plus de se baser sur l'état des liens, il se base aussi sur le cout de tel ou tel chemin. Il est calculé en fonction de la bande passante, plus la bande passante, plus le coût n'est faible. Si deux chemins ont le même cout, il se basera sur le nombre de sauts. Le fonctionnement d'OSPF au sein d'une seule zone et de la manière dont la table topologie ou la Link-state database est construite. La table de routage est constituée à partir de cette base de données. Ce résultat est obtenu grâce à l'application de l'algorithme de routage SPF. En voici les différentes étapes : [14]

- ✓ D'abord, un routeur doit trouver ses voisins. Pour se faire, il utilise des paquets Hello⎦ dès son initialisation (activation), qui sont envoyés sur chaque interface dont le routage dynamique est

activé, chaque routeur recevant ce paquet, intègre l'adresse IP de l'émetteur dans une base de données appelée base d'adjacence, et répond au routeur émetteur par un paquet IP unicast, le routeur émetteur intègre ainsi son adresse IP. Un routeur va générer un paquet LSA8 (Link-State Advertisement). Cette annonce va représenter la collection de tous les états de liens de voisinage du routeur ;

- ✓ Tous les routeurs vont s'échanger ces états de liens par inondation (flooding). Chaque routeur qui reçoit des mises à jour d'état de lien (Link-state update) en gardera une copie dans sa Link-state database et propagera la mise à jour auprès des autres routeurs ;
- ✓ Après que la base de données de chaque routeur soit complétée, le routeur va calculer l'arbre du chemin le plus court (Shortest Path Tree) vers toutes les destinations avec l'algorithme Dijkstra. Il construira alors la table de routage (routing table), en choisissant les meilleures routes;
- ✓ S'il n'y a pas de modification topologique, OSPF sera très discret. Par contre, cv en cas de changement, il y aura échange d'informations par des paquets d'état de lien et l'algorithme Dijkstra recalculera les chemins les plus courts.

1.8.2.2. Le protocole IS – IS

IS-IS a été conçu par ISO (International Organization for Standardization - Organisme international de normalisation) et est décrit dans la norme ISO 10589. La première application pratique de ce protocole de routage fut développée chez DEC (Digital Equipment Corporation) et est également connue sous le nom de DECnet Phase V. Radia Perlman a été le principal concepteur du protocole de routage IS-IS.

Il était à l'origine conçu pour l'ensemble de protocoles OSI et non pour TCP/IP. Par la suite Integrated IS-IS ou Dual IS-IS a intégré la prise en charge des réseaux IP. Bien qu'IS-IS soit connu comme le protocole de routage de la plupart des FAI et opérateurs télécom, un nombre croissant de sociétés commencent à l'utiliser.

Les protocoles OSPF et IS-IS ont beaucoup de points communs, mais présentent également de nombreuses différences. OSPF et IS-IS ont chacun leurs partisans, qui discutent et débattent des avantages respectifs d'un protocole de routage sur l'autre. Les deux fournissent les fonctions de routage nécessaires. [15]

1.9. L'état de lien ou vecteur distance

Nous avons vu qu'il y avait deux familles d'algorithmes (à vecteurs de distance et à état des liens). La première calcule le meilleur chemin selon sa longueur. La seconde calcule le meilleur chemin selon une fonction de coût.

Étudions quelques éléments d'analyse et de comparaison de ces deux familles : rapidité de convergence de l'algorithme, possibilités de métriques différentes, choix d'un chemin parmi plusieurs équivalents, utilisation de routes externes. On considère généralement qu'un

protocole à état des liens offre plusieurs avantages par rapport à un protocole à vecteurs de distance.

• ***Convergence rapide et sans boucle de l'algorithme.*** Dans un algorithme à vecteurs de distance, le nombre d'itérations est proportionnel au nombre de routeurs. Dans le pire cas, il est égal au nombre de routeurs moins 1. Dans un algorithme à état des liens, la convergence s'établit en deux phases : transmission rapide des nouvelles informations puis calcul local du chemin. De plus, cette méthode évite les boucles, puisque tous les chemins calculés sont sains.

• ***Métriques multiples***. Alors qu'il est difficile d'utiliser des métriques trop fines dans les algorithmes à vecteurs de distance, on peut supporter plusieurs métriques en parallèle, sans ralentir la convergence, dans les protocoles à état des liens. Cela provient du fait que la topologie est complètement connue pendant le calcul des chemins.

On peut donc choisir la meilleure route en fonction de critères différents, en appliquant des métriques différentes. Les algorithmes à état des liens sont les premiers à offrir un routage en fonction de la qualité de service requise par l'utilisateur.

• ***Chemins multiples***. Dans un protocole à vecteurs de distance, le choix d'un chemin parmi plusieurs se fait au hasard de la chronologie des échanges de messages. De plus, il n'est prévu qu'un seul routeur suivant dans la table de routage. Moyennant une légère modification de l'algorithme, les protocoles à état des liens peuvent tolérer des chemins multiples. On peut ainsi répartir le trafic entre plusieurs chemins équivalents en termes de coûts. L'équilibrage du trafic dans le réseau est une valeur ajoutée considérable, car elle contribue à la fluidité de la circulation des données et permet un réel contrôle de congestion.

• ***Routes externes.*** Les problèmes de routage que nous avons évoqués ne concernent que l'acheminement des données dans un réseau (considéré comme un ensemble homogène de stations et de routeurs). Une route externe est une route qui passe par d'autres zones ou d'autres réseaux que celui dans lequel on se trouve. Dans les grands réseaux et plus encore dans Internet, la connectivité se réalise à travers plusieurs points d'accès à différents réseaux de transit. Les éléments du choix des routes deviendraient trop complexes dans un protocole à vecteurs de distance : il faudrait prendre en compte plusieurs points d'accès, plusieurs prestataires de services, utiliser une route par défaut… Avec la possibilité d'utiliser des métriques multiples, les calculs de chemins intégrant des routes externes se font plus naturellement dans les protocoles à état des liens. [16]

1.10. Les algorithmes de routage

Il existe plusieurs types d'algorithmes de routage. Ceux-ci peuvent correspondre à des politiques déterministes ou adaptatives selon qu'elles s'adaptent ou non aux variations du trafic et de topologie du réseau. D'autre part, les algorithmes de routage peuvent être centralisés si les chemins sont définis par un nœudparticulier.

Dans le cas contraire, l'algorithme de routage est réparti entre tous les nœuds, ce qui est favorable du point de vue de la fiabilité, mais complique l'algorithme et rend plus difficile l'optimisation de l'acheminement des paquets. Le troisième paramètre à prendre en compte est le travail de routage effectué au niveau de chaque nœud, cela peut être le choix du prochain nœud ou l'indication de la route complète. Il existe enfin le routage par inondation ou aléatoire dans lequel le choix ne dépend pas de la topologie du réseau. Et parmi les algorithmes les plus classiques on peut citer celui l'algorithme de Bellman-Ford et l'algorithme de Dijkstra. [5]

1.11. Le besoin d'un réseau programmable

L'architecture fondamentale des réseaux traditionnels n'a connue aucun changementmajeur depuis plus de 20 ans. Ce statut qui est en partie dû à l'architecture des appareils en eux même, le fait est que la coexistence du plan de contrôle et du plan de données dans chaque équipement physique ce qui rend très difficile tout déploiement de nouvelle fonctionnalité ou protocole car ceux-ci doivent être incorporées dans l'équipement physique, 5 à 10 ans pour concevoir et déployer un protocole de routage. Cet environnement n'encourage pas le développement, car seuls les fabricants ont accès au hardware. Mais ce système a aussi créé une dépendance des particuliers envers leurs fabricants, faute de problèmes d'interopérabilité entre les différentes marques, les utilisateursse trouvent comme liés à leurs fabricants et ne peuvent en aucun cas changer de fournisseur, une atmosphère sans compétitivité et qui encourage la flambée des prix des appareils réseaux. Ces systèmes renfermés ont instauré une inertie dans le réseau en comparaison avec l'informatique et les technologies de stockage, et si on devait s'inspirer de celles-ci et d'un des pionniers du développement de l'informatique qu'est Linux, L'open source est la meilleure initiative.

Mais le point qui force le Networking à sortir de sa bulle est le contraste de sa nature rigide et statique avec les nouvelles techniques dans les data center que ce soit la virtualisation des machines ou le Cloud computing. Ces dernières apportent une flexibilité sans précédent au déploiement des services et l'exploitation des ressources, Or l'infrastructure réseau opère comme un frein dans ce modèle qui requiert une agilité particulière, car la ou il suffit de quelques minutes pour créer une nouvelle instance d'une VM, l'ajout d'une instance

quelconque dans le réseau pourrait durer des jours. Ceci crée des problèmes d'extensibilité dans les data center qui sont censés croitre au besoin. C'est la virtualisation qui fait toute la différence. La tendance du Cloud est le « everything as a service » (Tout en service : Ressources RAM, CPU, Stockage, Applications …), et la dernière pièce manquante pour le tout virtuel est le « Network as a service (Naas) » (Le réseau en service). [1]

1.12. Conclusion

Ce chapitre consiste à étudier profondément l'architecture des réseaux traditionnels et ses techniques de routage, plus les algorithmes principaux utiliséspuis nous avons arrêté sur ses faiblesses et limitations.

Avec le développement actuel dans tous les domaines de télécommunicationqui se basent tous sur le routage, nous sommes obligées d'améliorer ses techniques traditionnelles avec desréseaux programmable que nous allons leprésenter dans le chapitre suivant.

Chapitre 2

Chapitre 2 : : Le Concept Software Defined Networking-SDN-

2.1. Introduction

Durant ces dernières années, les réseaux informatiques classiques ont connus de grands défis qui résultent principalement des applications modernes déployés sur ces réseaux. Parmi ces applications le SDN (Software Defined Networking) qu'est une nouvelle approche du monde actuel des réseaux.

Dans ce chapitre, nous allons présenter de générale du SDN et ses différents concepts clés.

2.2. La route vers le SDN

Avant l'apparition des réseaux SDN, plusieurs idées et travaux ont été proposée auparavant, notamment la programmation des réseaux et la séparation des plans de contrôle et de donné. Par exemple active network et programming network, et le projet de recherche dénommé DCAN (devloped control of ATM networks).

Le début des réseaux SDN a commencé avec le projet ethane, lancé en 2006 à l'université destanford. En effet, le projet ethane défini une nouvelle architecture pour les réseaux d'entreprise.

L'objective d'éthane étais d'avoir un contrôleur centralisé pour gérer les règles et la sécurité dans les réseaux. Ethane utilise deux composent : un contrôleur pour décider si un paquet doit être transmet, et un Switch ethane composé d'une table et d'une chaine de Communication entre les deux. [17]

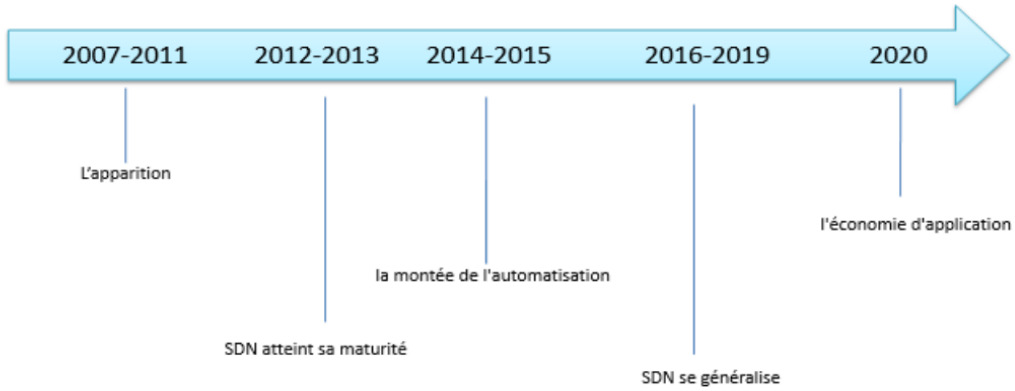


Figure 2.1 : Le développement de réseaux SDN à travers le temps

2.3. Définition

SDN signifie littéralement Software Defined Networking, c'est-à-dire le réseau défini par logiciel. On comprend donc immédiatement que le sujet est vaste et qu'il va être difficile d'avoir une définition unique.

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

Dans ce modèle, les équipements réseau se contentent d'implémenter des règles, injectées par les applications, de traitement des flux de données. Plus besoin d'avoir sur ces équipements de protocoles de routage : une entité intelligente, appelée « contrôleur » voit le réseau dans sa globalité et injecte directement les règles de traitement des données sur chaque équipement.

Le SDN est donc reconnue aujourd'hui comme une architecture permettant d'ouvrir le réseau aux applications. Cela intègre les deux volets suivants :

- Permettre aux applications de programmer le réseau afin d'en accélérer le déploiement.
- Permettre au réseau de mieux identifier les applications transportées pour mieux les gérer.

2.4. L'architecture de réseauSDN

Le réseau SDN est composé de 3 couches communiquant entre elles par le biais d'interfaces APIs. De la plus basse à la plus haute nous avons de contrôle

2.4.1. La couche d'application

Contient les applications ou fonctions réseaux typiques utilisées par les organisations. L'exemple le plus répandu étant notamment les systèmes de détection d'intrusion, l'équilibrage de charge ou les pare-feux. Lorsqu'un réseau traditionnel utilise une Appliance spécialisée, telle qu'un pare-feu ou un équilibreur de charge, un réseau défini par logiciel remplace l'Appliance par une application qui utilise le contrôleur pour gérer le comportement du **data-plane** (plan de données). [17]

2.4.2. La couche de contrôle

Représente le logiciel contrôleur centralisé qui agit comme cerveau du réseau défini par logiciel. Ce contrôleur réside sur un serveur et gère les règles et le flux de trafic sur le réseau. [17]

2.4.3. La couche d'infrastructure (Ou le plan de donnée)

Est composer d'un ensemble d'équipement réseaux (commutateurs, routeurs…)

Ces trois couches communiquent à l'aide d'interfaces de programmation d'applications (API) respectives en direction du nord et du sud. Par exemple, les applications communiquent avec le contrôleur via son interface en direction du nord, tandis que le contrôleur et les commutateurs communiquent via des interfaces en direction sud, tel que Open Flow bien que d'autres protocoles existent. [17]

La figure suivanteprésentel'architecture de SDN :

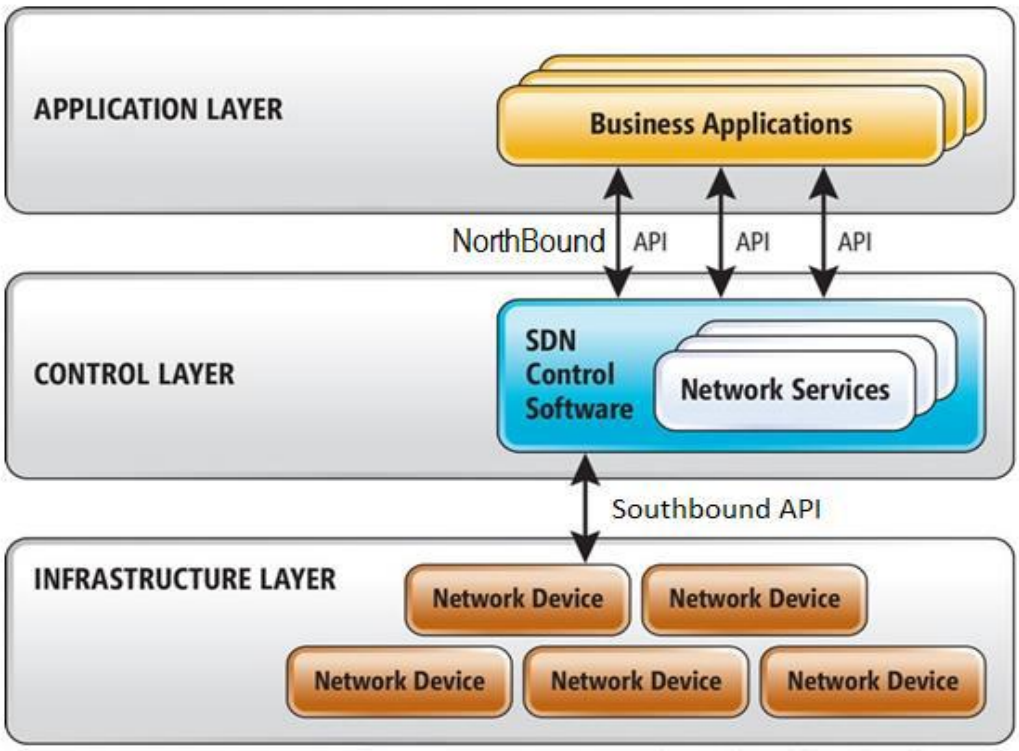


Figure 2.2 : Architecture SDN

2.5. Les principaux acteurs de l'architecture SDN :

2.5.1. Le Switch SDN

Tous les équipements de transmission SDN sont appelés Switch, ce terme est attribué à tout équipement de transmission qui compare l'entête des paquets aux tables de flux, que la comparaison se fasse à base d'adresses MAC (Couche 2), d'adresses IP (Couche 3) ou une combinaison de plusieurs champs. La terminologie Switch a été adoptée en référence à l'unique tache de ces équipements qui est la transmission

On distingue deux types de Switch Open Flow

- ✓ Switch Open Flow-only : Supporte uniquement les actions requises pour le fonctionnement du protocole Open Flow.
- ✓ Switch Open Flow-enabled : En plus des actions requises pour le protocole Open Flow, ceSwitch supporte les actions d'un Switch ordinaire.

Un Switch SDN est composé d'une API pour communiquer avec le contrôleur qui représenté par l'application Open flow, une couche d'abstraction qui consiste en un pipeline de tables de flux et une fonction de traitement des paquets. [1]

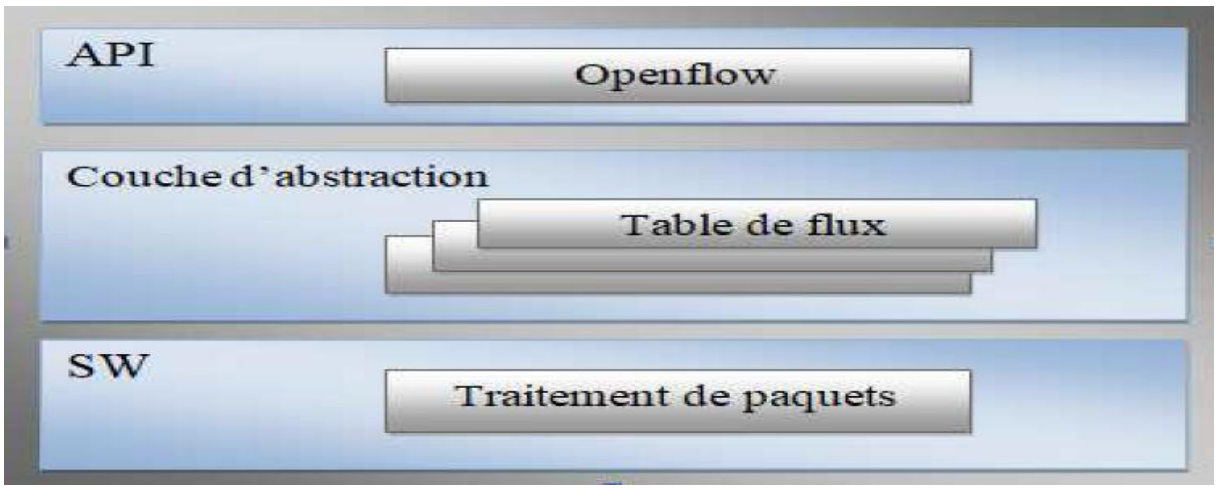


Figure 2.3 : Les composants d'un Switch open flow

2.5.2. Les types de Switch SDN

On distingue deux types de Switch SDN :

2.5.2.1. Switch SDN logiciel :

C'est le moyen le plus simple pour créer un équipement SDN. Les tables de flux, les entrées et les champs de correspondance sont facilement implémentés dans les structures de données d'un software. Ce modèle est plus lent et moins efficace car il ne bénéficie pas d'une accélération hardware, mais présente l'avantage d'être plus flexible, plus riche dans les actions disponibles, et supporte un nombre d'entrées beaucoup plus important. Les Switch logiciels sont très présents dans les environnements virtualités et sont généralement open source. Les deux principales alternatives sont Open Switch (OVS) de Nicira et Indigo de Big Switch. [1]

2.5.2.2. Switch SDN matériel

Ces implémentations sont plus rapides, et sont la seule possibilité pour un environnement très haut débit (100Gbs) étant sensible aux performances. Pour transcrire les différentes entrées de flux et leurs composantes dans les Switch on a opté pour l'utilisation de hardware spécialisé, pour le traitement de couche 2 (MAC) on utilise les Cams (Content-AddressableMemories) et pour la couche 3 les TCAMs (Ternary Content-adressable Memories). Ces Switch sont adaptés aux data center et au cœur du réseau, les politique de flux n'y sont pas centrés sur les utilisateurs ce qui fait que le nombre d'entrée y est moins important que dans les Switch qui sont plus près de l'accès. [1]

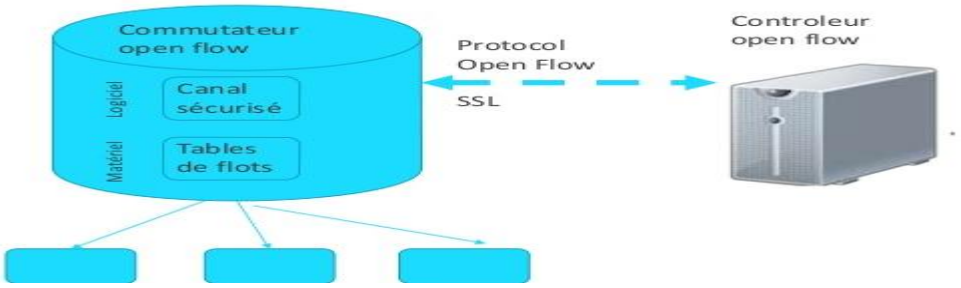


Figure 2.4 : Switch SDN matériel

2.6. Le contrôleur

Le rôle du plan de contrôle est de contrôler et de gérer les équipements de l'infrastructure, et de les relier avec les applications. Il est composé d'un ou de plusieurs contrôleurs et il est considéré comme système d'exploitation du réseau. Les premières

versions du SDN présentent un plan de contrôle composé d'un seul contrôleur centralisé. Par la suite des architectures distribuées utilisant plusieurs contrôleurs ont été proposées pour améliorer les performances et la sociabilité du réseau. Les performances des contrôleurs SDN sont caractérisées par le débit, qui est la quantité de flux traités par seconde et la latence, qui est le temps d'installation d'une nouvelle règle Afin de pouvoir interagir avec le réseau, le contrôleur a besoin d'une vue précise de ce Dernier. C'est ainsi que le concept de NIB (Network Information Base) a vu le jour. Cette NIB est construite au niveau du contrôleur et permet à ce dernier de savoir comment implémenter chaque ordre abstrait, trouver les équipements qui doivent être reconfigurés, s'assurer de la capacité de ces derniers à implémenter une directive et les API supportées par l'équipement. [1]

2.6.1. Le mode opérationnel de contrôleur

Le contrôleur se base sur deux modes opérationnels : réactif et proactif.

2.6.1.1. L'approche réactive :

Fait tout transiter par le contrôleur. Lorsqu'il y a un paquet entrant sur le Switch, celui-ci est directement redirigé vers le contrôleur pour que ce dernier décide du comportement à adopter vis-à-vis du paquet. Ce model peut causer des temps de latences considérable, en fonction des ressources à disposition du contrôleur et la distance Switch-contrôleur.

2.6.1.2. L'approcheproactive

Le contrôleur introduit préalablement des règles dans les switches pour qu'ils puissent traiter les paquets localement. Ici, le nombre de paquets envoyés au contrôleur diminue considérablement ce qui fait gagner en efficacité et en débit à l'architecture.
Plusieurs contrôleurs SDN sont développés, commerciales ou Open source. Ils différent par leur architecture (Centralisé ou Distribuée), leurs langages de programmation, les API qu'ils supportent, les techniques utilisés et leurs performances comme le débit et la latence.

2.6.2. Les types de contrôleurs

Il existe plusieurs contrôleurs SDN, tel que :

- **NOX :** Initialement développé chez Nicira, NOX est le premier contrôleur OpenFlow. C'est Open-source et écrit en C ++. Il est actuellement à la baisse : il n'y a pas eu de changements majeurs depuis mi 2012 [17].
- **POX :** 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

- **Beacon :** Beacon est un contrôleur OpenFlow rapide, multiplateforme, modulaire basé sur 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
- **Flood light :** Floodlight est un contrôleur open-source OpenFlow basé sur Java, pris en charge par BigSwitch Networks. Il est sous licence Apache .Il est facile à configurer et à montrer aussi de grandes performances. Avec toutes ses fonctionnalités, Floodlight est plus une solution complète.
- **OpenDaylight :** 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.

Le tableau ci-dessous résume ces types des contrôleurs :

Contrôleur	Architecture	Openflow	Langage	Tolérance pannes	API Nord
Beacon	Centralisé multi-thread	V 1.0	Java	NON	Ad-hoc API
Floodlight	Centralisé multi-thread	V 1.1 1.2 1.3 1.4	Java	NON	RESTful API
ONOS	Distribué	V 1.0	Java	OUI	RESTful API
OpenDaylight	Distribué	V 1.0 1.3	Java	NON	REST/RESTCONF
Ryu	Centralisé	V 1.1 1.2 1.3 1.4 1.5	Python	NON	Ad-hoc API
NOX	Centralisé	V 1.0 1.1 1.2 1.3	C++	NON	Ad-hoc API

Tableau 2-1 : Les types de contrôleurs

2.7. Les interfaces de communication

Au niveau de l'architecture SDN nous trouvons 3 types d'interfaces de communication : l'interface Sud, l'interface Nord et l'interfaces Est/Ouest, qui jouent le rôle d'intermédiaires entre les différentes couches de cette architecture pour assurer le bon fonctionnement des réseaux programme parlogiciel. [2]

2.7.1. L'interface Sud

Ce sont les interfaces (South-bound) qui permettent le processus de communication entrele contrôleur et les switches/routeurs et autres éléments de la couche infrastructure réseau.
C'est par le biais de cette interface et notamment le protocole OpenFlow dans le cas du standard Open SDN (Open Networking Foundation, 2012), que le contrôleur injecte les différentes politiques aux équipements, et récupère les informations permettant aux applications de construire une vue globale du réseau.

:

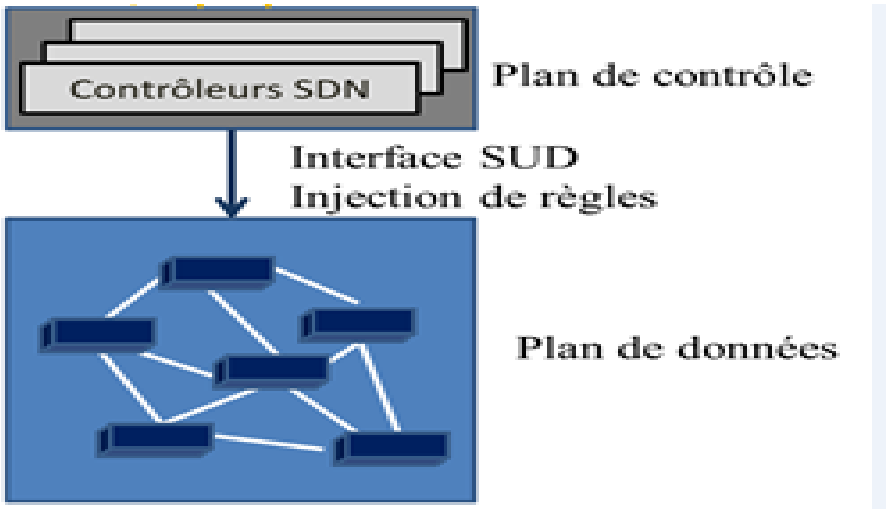


Figure 2.5 : L'Interfaces de Communication Sud

2.7.2. Interface Nord

Les interfaces Nord (North-bound) servent à programmer les éléments de la transmission en exploitant l'abstraction du réseau fourni par le plan de contrôle. En d'autres termes elles permettent la communication entre le contrôleur la couche applicative. Elles sont considérées davantage comme des API que comme protocole de programmation et de gestion de réseau.
Il n'existe aucun standard intervenant entre la couche de contrôle et celle d'application. Selon l'ONF (Open Networking Foundation, 2012), plusieurs niveaux d'abstraction et différents cas d'utilisation peuvent être caractérisés, ce qui signifie qu'il peut y avoir plusieurs interfaces Nord pour servir tous les cas d'utilisation. Parmi les propositions des industriels, nous trouvons une API basée sur REST API (REpresentational State Transfer) pour fournir une interface programmable utilisable
par les applications. [1]

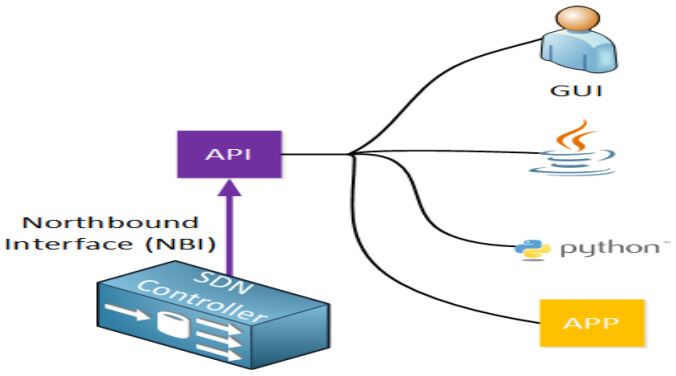
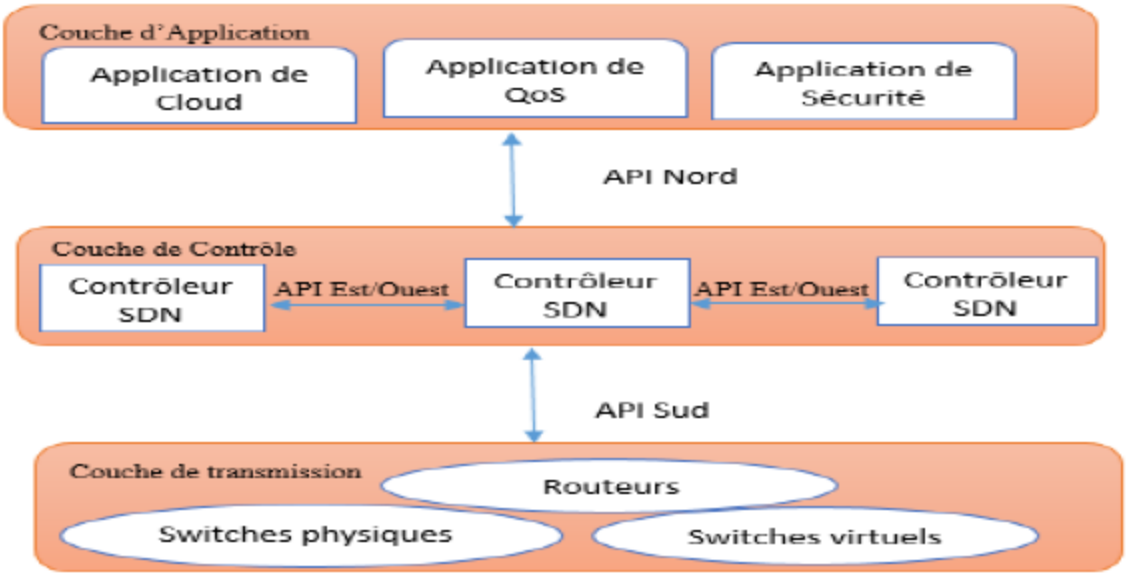


Figure 2.6 : L'Interfaces de Communication Nord

2.7.3. Interface Est/Ouest

Ce sont des interfaces inter-contrôleurs, on les trouve dans les architectures distribuées(Multi-contrôleurs). Ils permettent la communication entre contrôleurs pour synchroniser les états du



réseau. Aucun standard n'est encore disponible pour ce type d'interfaces

Figure2.8: L'Interfaces de Communication Est/Ouest

2.8. Le protocoleOpenFlow:

2.8.1. Historique de protocole :

L'histoire d'OpenFlow est intéressante et permet de mieux comprendre son rôle fondamental dans la conception de l'architecture SDN.

OpenFlow a été initié comme un projet à l'université de Stanford lorsqu'un groupe de chercheurs exploraient la manière de tester de nouveaux protocoles dans le monde IP (créer un réseau expérimental confondu avec le réseau de production) mais sans arrêter le trafic du réseau de production lors des tests.

C'est dans cet environnement que les chercheurs à Stanford ont trouvé un moyen de séparer le trafic de recherche du trafic du réseau de production qui utilise le même réseau IP. En décembre 2009, la version 1.0 de la spécification du commutateur OpenFlow a été publiée. Depuis sa création, OpenFlow est géré par l'Open Networking Foundation (ONF) , une organisation dirigée par des utilisateurs dédiée aux normes ouvertes et à l'adoption du SDN.

Depuis sa sortie, plusieurs entreprises et projets open source comme le projet OpenDaylight prennent en charge OpenFlow, et fournissent même des contrôleurs OpenDaylight. D'autres sociétés comme Cisco et Brocade proposent également des contrôleurs compatibles OF, avec Cisco XNC et Brocade Vyatta Controller. [19]

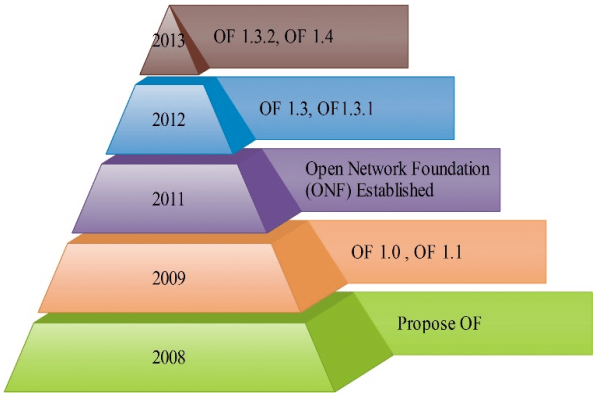


Figure 2.9:OpenFlow à travers le temps

2.8.2. Définition :

Open Flow 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** établie via le port **6633 ou 6653** du serveur contrôleur.

Open flow est donc une composante du SDN.

Ce protocole est ce qui identifie le plus de manière unique la technologie OpenFlow. Dans son essence, le protocole se compose d'un ensemble de messages qui sont envoyés du contrôleur au commutateur et d'un ensemble correspondant de messages qui sont envoyés dans la direction opposée. Les messages, collectivement, permettent au contrôleur de programmer le commutateur de manière à permettre un contrôle fin sur la commutation du trafic utilisateur.

Open Flow est donc un protocole ouvert (open Protocol) qui permet aux Administrateurs de réseau de programmer les tables de flux (flow tables) dans leurs différents commutateurs, chacun avec son ensemble de fonctionnalités et caractéristiques de flux. [18]

2.8.3. L'architecture OpenFlow :

L'architecture open flow est l'implémentation réelle des réseaux SDN, Cette architecture est basée principalement sur trois composantes : le plan de données, qui est composée des switchesOpenFlow, le plan de contrôle : constitué par des contrôleurs OpenFlow et une chaîne sécurisée qui permettent aux commutateurs de se connecter au plan de contrôle.

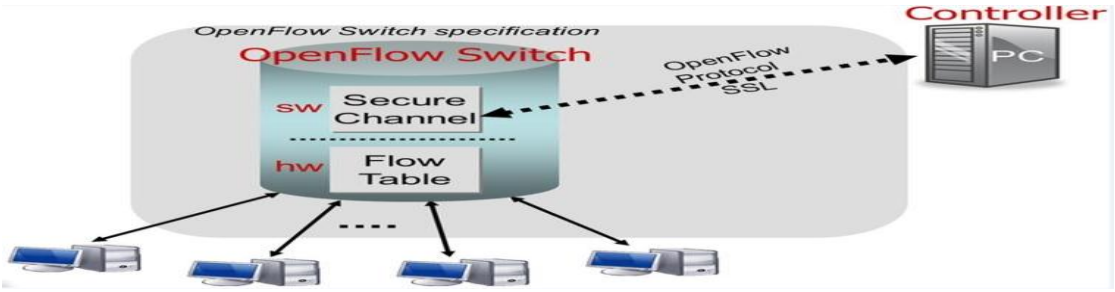


Figure 2.10: L'ArchitectureOpenFlow

2.8.4. CommutateurOpenFlow:

Le commutateur OpenFlow est standardisé selon la spécification d'ONF, un commutateur OpenFlow doit contenir un ou plusieurs tables de flux, ces tables de flux contiennent plusieurs d'entrées qui correspondent à des règles, où chacune est constituée principalement des trois champs suivants :

L'En-tête de paquet : il définit le flux de données, il contient les informations nécessaires pour déterminer le paquet auquel cette règle sera appliquée. L'en-tête de paquet peut identifier différents protocoles tel qu'Ethernet, IPv4, IPv6 ou MPLS, cela dépend de la spécification d'OpenFlow déployée.

L'Action : spécifie comment les paquets d'un flux seront traités. Une action peut être l'une des suivantes transférer le paquet vers un ou plusieurs ports, supprimer le paquet, transférer le paquet vers le contrôleur, ou modifier le champ d'entête de paquet.

Les Compteurs : sont réservés à la collecte des statistiques de flux. Ils enregistrent le nombre de paquets et d'octets reçus de chaque flux, et le temps écoulé depuis le dernier transfert de flux.

2.8.5. Canalsécurisé

Le canal OpenFlow est l'interface qui connecte chaque commutateur OpenFlow à un contrôleur. Cette interface permet aux contrôleurs de recevoir le message du commutateur et de pouvoir de gérer à travers le réseau.

Le canal doit être sécurisé afin d'assurer le bon déroulement des communications entre le commutateur et contrôleur. Pour cela l'échange de message se fait au cours d'une session TCP (transmission control Protocol) établie via le port 6633/6653 du serveur contrôleur ou va travers une connexion SSL/TLS (Secure socket layer /transport layer Security).

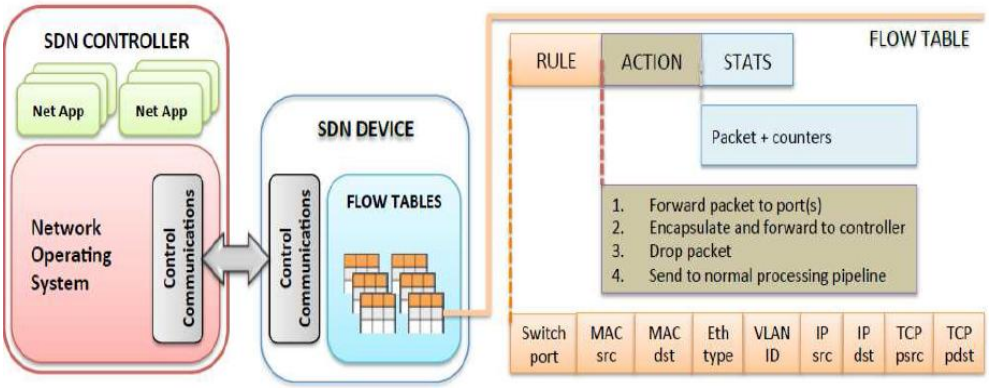


Figure 2.11: Table de flux.

2.9. Fonctionnement OF

Lors de démarrage du commutateur OpenFlow. Ce dernier envoie un paquet « OFPT_HELLO » avec le numéro de version d'OpenFlow supportée. Le contrôleur vérifie la version d'OpenFlow supportée par le commutateur et lui répond par un message « OFPT_HELLO » en indiquant la version d'OpenFlow avec laquelle ils communiqueront. La connexion est établie. [5]

La figure présente les étapes de connexion :

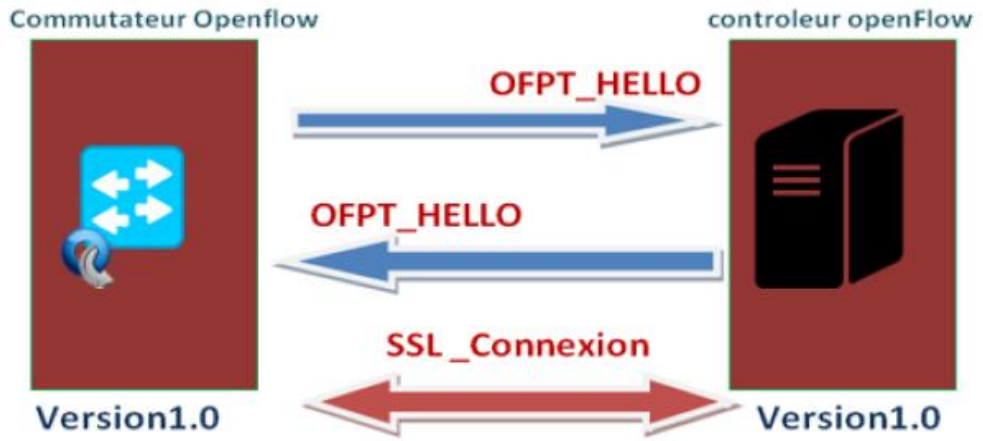


Figure2.12: La connexion au contrôleur

Si le contrôleur ne supporté pas la version OpenFlow de commutateur, il lui répond par un paquet « OFPT_ERROR » en indiquant que c'est un problème de compatibilité. Commeillustré dans la figure suivante :

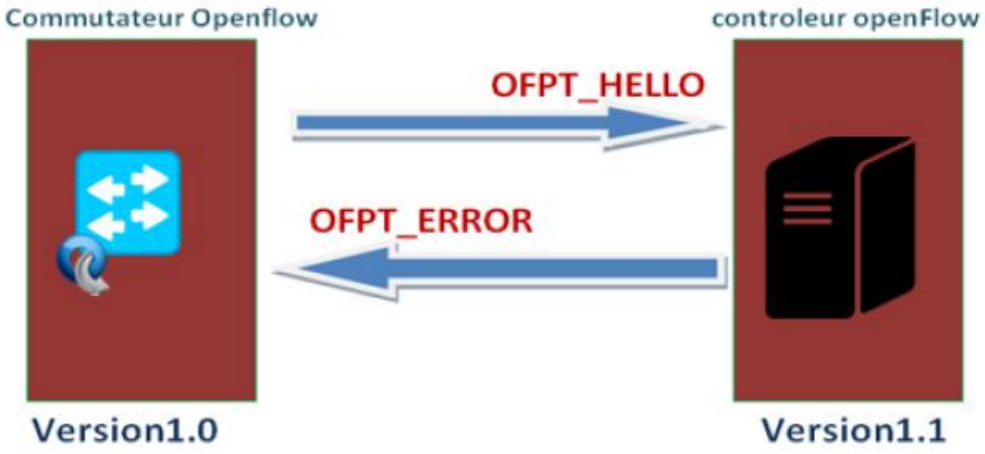


Figure 2.13: Echoc de la connexion au contrôleurOpenFlow

Si lecontrôleur ne répond pas, donc il se met en mode urgence « EMERGENCY MODE » le commutateur utilise sa table de flux par défaut.

Si le paquet ne correspond à aucun enregistrement dans la table il le supprime. Comme la montre la figure ci-dessous :

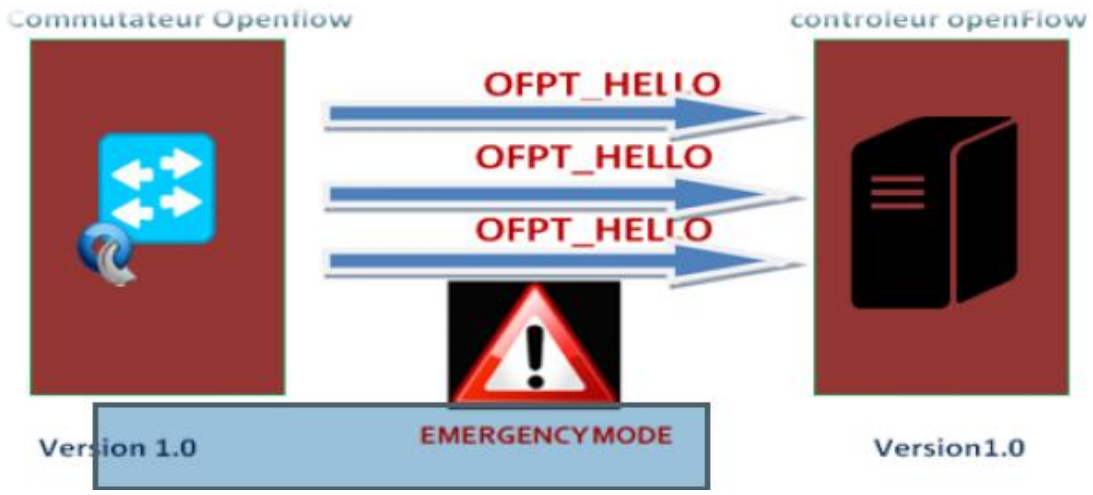


Figure2.14: Mode d'urgence

Lorsqu'un paquet arrive à un commutateur, le commutateur vérifie s'il y a une entrée dans la table de flux qui correspond à l'en-tête de paquet. Si c'est le cas, le commutateur exécute l'action correspondante dans la table de flux. Dans le cas contraire, c'est-à-dire il n'y a pas une entrée correspondante (1), le commutateur génère un message asynchrone vers le contrôleur (2) sous la forme d'un 'Packet_in', puis le contrôleur décide selon sa configuration une action pour ce paquet, et envoie une nouvelle règle de transmission sous la forme d'un 'Packet_out' et 'Flow-mod' au commutateur (3), et enfin, la table de flux du commutateur est actualisée, pour prendre en compte la nouvelle règle installée par le contrôleur (4).

La Figure décrit le processus de transmission d'un paquet avec open flow.

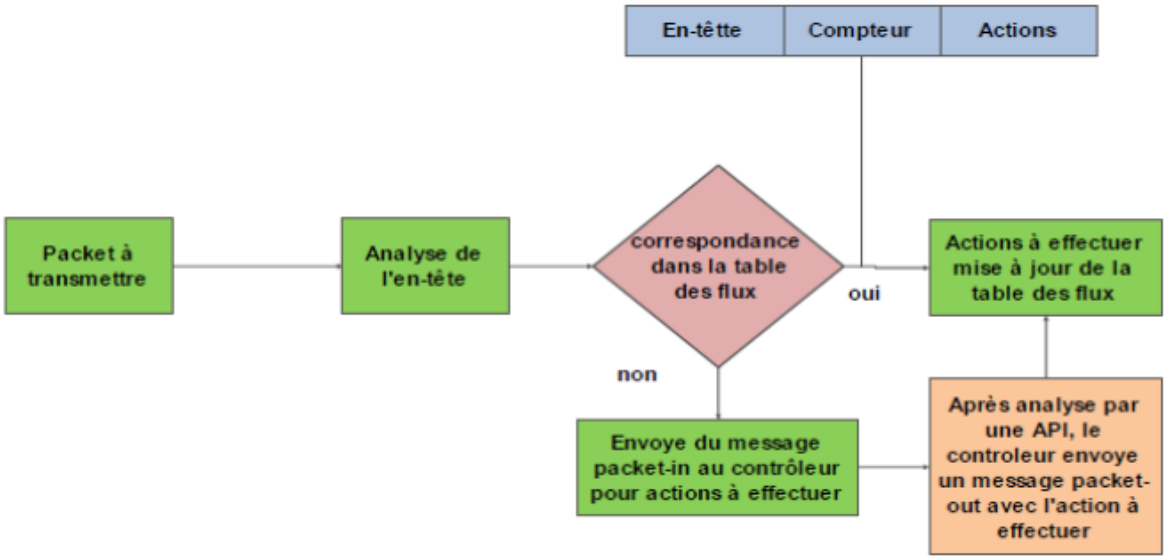


Figure2.15: Processus de transmission des paquets au sien de commutateur OpenFlow.

2.9.1. LaTable de flux :

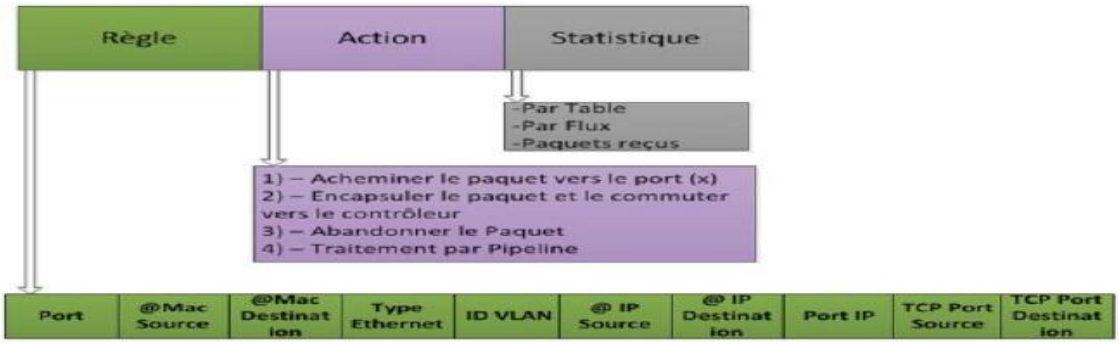


Figure 2.16: Les composants d'une table de flux

Comme illustré la figure, Chaque table de flux de commutateur contient un ensemble d'entre de flux qui se composé de : [5]

- Champs de correspondance (Match Fields ou règle) : Utilisés lors de la recherche de l'entrée correspondante au paquet. Ils sont constitués essentiellement des entêtes des paquets et des ports d'entrées afin d'y appliquer une action X
- Compteurs (counters ou statistique) : 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, chaque file d'attente et chaque groupe de tables, des compteurs de statistiques sont maintenus.
- Instruction (action) : Représentent l'ensemble des instructions OpenFlow qui servent à modifier le traitement (pipeline) que va subir le paquet. Les instructions supportées sont :

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.

2.9.2. Les messages open flow

Open Flow prend en charge trois types de messages. Ces messages sont :

2.9.3. Les messages Controller to Switch

Sont les massages utilisés pour gérer et configurer les commutateurs. Ils ont géré les tableaux de flux. L'échange de capacité de commutation est également effectué avec ces messages. Les messages Controller to Switch sont lancés à partir du contrôleur.

2.9.4. Les messages asynchrones

Sont envoyés pour informer le contrôle de tout événement et changement d'état. Les messages asynchrones sont envoyés par les Commutateur

2.9.5. Les messages symétriques

Sont utilisés pour diagnostiquer tout problème entre le contrôleur et les commutateurs. Les messages symétriques peuvent être envoyés à la fois à partir du contrôleur et des commutateurs. [21]

Message Category	Message	Message Type	Direction	Process
Conf.	Hello	Symmetric	Controller->Switch	"Here is my Version Number!"
	Hello	Symmetric	Switch->Controller	"Here is Verision Number, that I support!"
	Features Request	Control/Switch	Controller->Switch	"Which ports are available?"
	Set Config	Control/Switch	Controller->Switch	"Could you send Flow Expirations?"
	Features Reply	Control/Switch	Switch->Controller	"Here are the available ports / supported actions!"
	Port Status	Asynchronous	Switch->Controller	Informing Controller about some features.
Flow	Packet-In	Asynchronous	Switch->Controller	"There is no match in Flow Table for this Flow!"
	Packet-Out	Control/Switch	Controller->Switch	"Send packet out to these ports!"
	Flow-Mod	Control/Switch	Controller->Switch	"Add this Flow to the Flow Table!"
	Flow-Expired	Control/Switch	Switch->Controller	Flow timed out after being inactive for a period.

Figure 2.17 : Les défirent type des messages et son direction et procès

2.10. Les avantages de SDN

2.10.1. Réseaux programmables

Avec SDN, il est plus simple de modifier les stratégies réseau car il suffit de changerune politique de haut niveau et non de multiples règles dans divers équipements de réseau. De plus, la centralisation de la logique dans un tel contrôleur entièrement personnalisé avec des connaissances globales et une puissance de calcul élevée, simplifient ledéveloppement des

fonctions plus sophistiquées. Cette aptitude à programmer le réseau estl'élément clé du SDN. [20]

2.10.2. Flexibilité

SDN apporte également une grande flexibilité dans la gestion du réseau. Il devient facile de rediriger le trafic, d'inspecter des flux particuliers, de tester de nouvelles stratégies oude découvrir des flux inattendus.

2.10.3. Politique unifiée

Avec son contrôleur, SDN garantit également un politique réseau unifiée et à jour, puisque le contrôleur est responsable de l'ajout de règles dans les commutateurs, il n'y a aucunRisque qu'un administrateur de réseau ait oublié un commutateur ou installé des règles incohérentes entre les dispositifs. En effet, l'administrateur va simplement spécifier une nouvelle règle et le contrôleuradaptera la configuration pour envoyer des règles cohérentes dans chaque dispositif pertinent.

2.10.4. Routage

SDN peut également être utilisée pour gérer les informations de routage de manière centralisée en déléguant le routage et en utilisant une interface pour le contrôleur.

2.10.5. Gestion du Cloud

SDN permet également une gestion simple d'une plateforme Cloud. En effet, la dynamique apportée par SDN traite des problèmes spécifiques aux clouds tels que l’évolutivité, l'adaptation, ou des mouvements de machines virtuelles.

2.10.6. Simplification matérielle

SDN a tendance d'utiliser des technologies standard et de base pour contrôler les équipements du réseau, tandis que la puissance de calcul n'est requise qu'au niveau ducontrôleur. Ainsi, les équipements de réseau deviendront des produits à bas prix offrant des interfaces standard. Avec ce type de matériel, il serait également simple d'ajouter de nouveauxpériphériques, puisqu'ils ne sont pas spécialisés, de les connecter au réseau et de laisser lecontrôleur les gérer conformément à la politique définie. Ainsi, le réseau devient facilementévolutif dès que le contrôleur est évolutif.

2.11. Conclusion

Au cours de la deuxième partie, nous avons fourni une base théorique sur les réseauxDéfinis par logiciel (SDN), en présentant la définition et l'architecture et les avantages de ce dernier, puis nous avons traité les composants essentiels de cette solution afin d’appliquer ses concepts à notre contexte.

Chapitre 3

Chapitre 3 : : comparaison entre le réseau traditionnel et le réseau SDN

3.1. Introduction :

Le SDN a pour but pratique de rendre les réseaux programmables par le biais d'un contrôleur centralisé. Un contrôleur central omniscient permet aux ingénieurs de réseaux de mettre en œuvre des politiques de transfert unique et flexible dont les seules limitations sont liées à la capacité du logiciel faisant fonctionner le contrôleur.

Dans ce chapitre, nous commençons par une présentationet une comparaison entre le réseau traditionnel et le réseauSDN, puis nous présentons l'émulateurmininet et l'interface graphique Miniedit.Et à la fin on a introduit une analyse comparative des performances.

3.2. Comparaison entre le réseau traditionnel et SDN :

Réseau traditionnel	RéseauSDN
▪ Le contrôle du réseau est complexe ▪ Une configuration manuelle et la possibilité de faire des erreurs qui vont entrainer un comportement erroné du réseau ▪ Le problème de configuration statique ▪ Difficulté d'implémentation de logiciel et des mises à jour dans le réseau ▪ Environnement de test limité	▪ Découpler le plan de contrôle de celui du plan de donné ▪ Offre un meilleur contrôle du réseau et la possibilité de la programmer ▪ Configuration automatique à travers une centralisation de contrôle du réseau ▪ Optimisation de la configuration ▪ Contrôle global de l'information ▪ L'implémentation facile de logiciels et des mises à jour dans le réseau ▪ Environnement de test suffisant

Tableau 3-1 : Tableau comparative entre réseau traditionnel et réseau SDN

3.3. Créé une topologie de réseaux traditionnelle à l'aide de Miniedit

3.3.1. Démarrer Miniedit :

Miniedit est une interface graphique qui permet de créer des réseaux virtuels sans passer par les commandes. C'est un outil qui est en phase d'expérimentation, avec très peu de fonctionnalités. D'ailleurs, jusqu'à maintenant, il ne permet que de créer une topologie du réseau avec quelques configurations basiques. Pour ouvrir l'interface graphique, exécuter la commande suivante :

```
$ Sudo~/mininet/examples/Miniedit.py
```

Pour réaliser un réseau virtuel, il suffit de glisser le dessin situé à gauche et le déposer sur la fenêtre comme le montre la figure 2.20

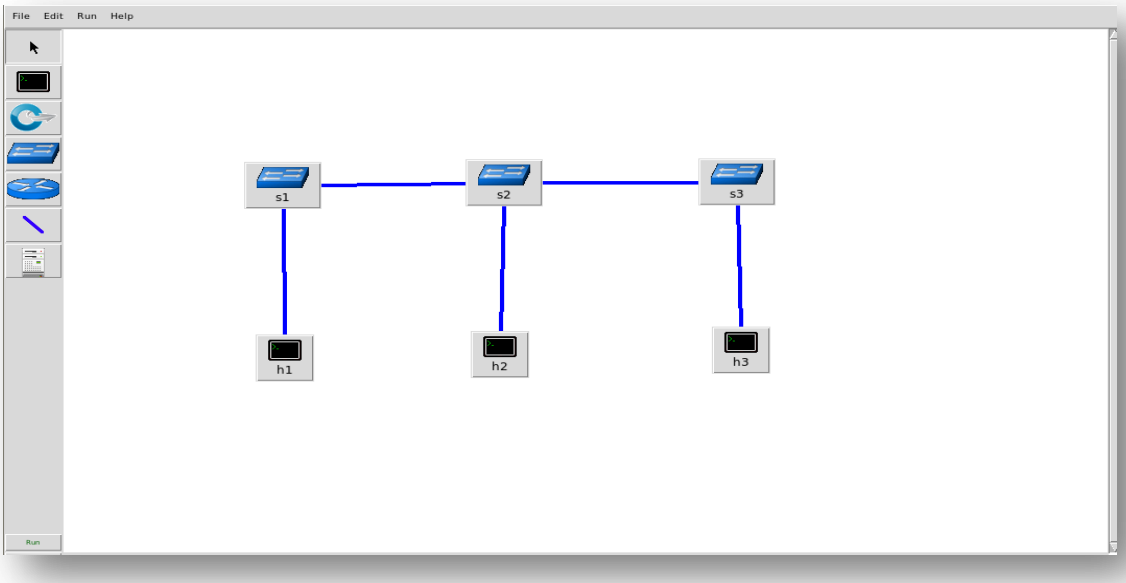


Figure3.1:L'interface graphique Miniedit du mininet

3.3.2. Configurer les hosts

Dans la topologie de base, nous avons trois switches et trois hosts. Cliquez avec le bouton droit sur chaque host et sélectionnez *Propriétés* dans le menu qui apparaît.et Modifiez l'adresse IP de l'host.

3.3.3. Définir les performances de *Miniedit*

Pour définir les préférences de *Miniedit*, utilisez la commande de menu *Miniedit*, *Edition* → *Préférences*. Dans la boîte de dialogue qui apparaît, apportez les modifications dont vous avez besoin.

Par défaut, la fenêtre de la console *Miniedit* ne donne pas à l'utilisateur l'accès à l'interface de ligne de commande *Mininet*. Si vous souhaitez pouvoir utiliser la CLI *Mininet* lorsqu'une simulation est en cours d'exécution, cochez la case *Démarrer CLI*.

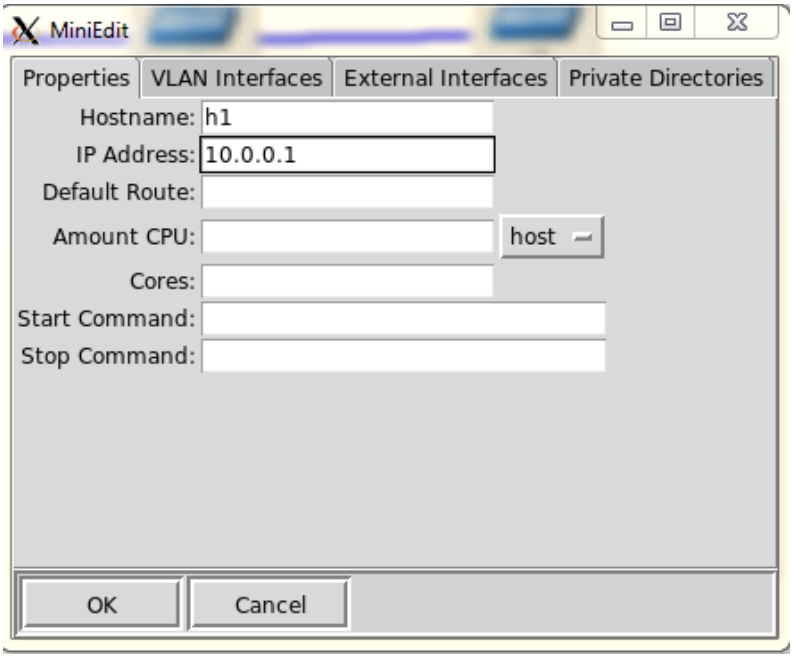


Figure 3.1 : Configurationdes hosts

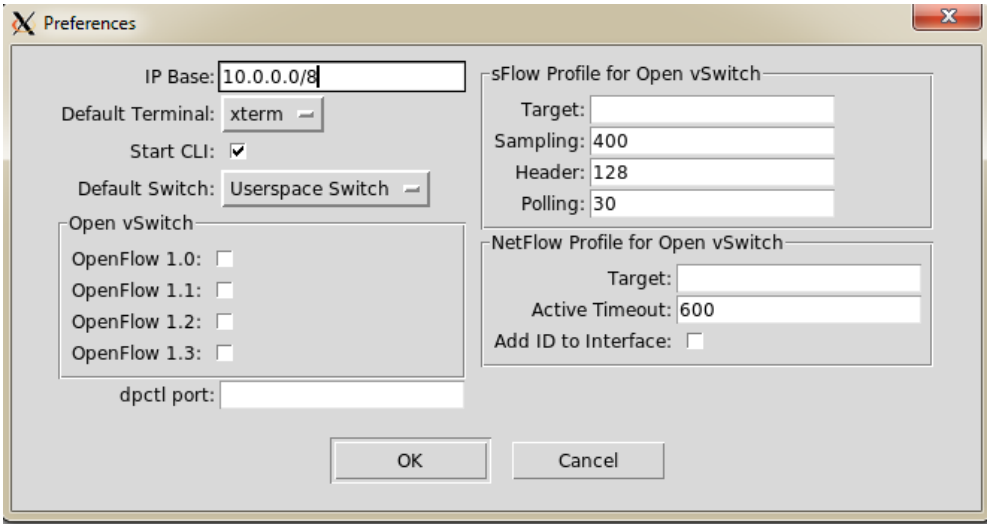


Figure 3.2 : Configuration des Propriétés de *Miniedit*

3.3.4. Exécuter le scénario de réseau *Miniedit*

Pour démarrer le scénario de simulation, cliquez sur le bouton *Exécuter* de l'interface graphique *Miniedit*. Dans la fenêtre du terminal à partir de laquelle vous avez démarré *Miniedit*, vous verrez des messages indiquant la progression du démarrage de la simulation, puis l'invite de la CLI *Miniedit* (car nous avons coché la case *Démarrer la CLI* dans la fenêtre des préférences de *Miniedit*.

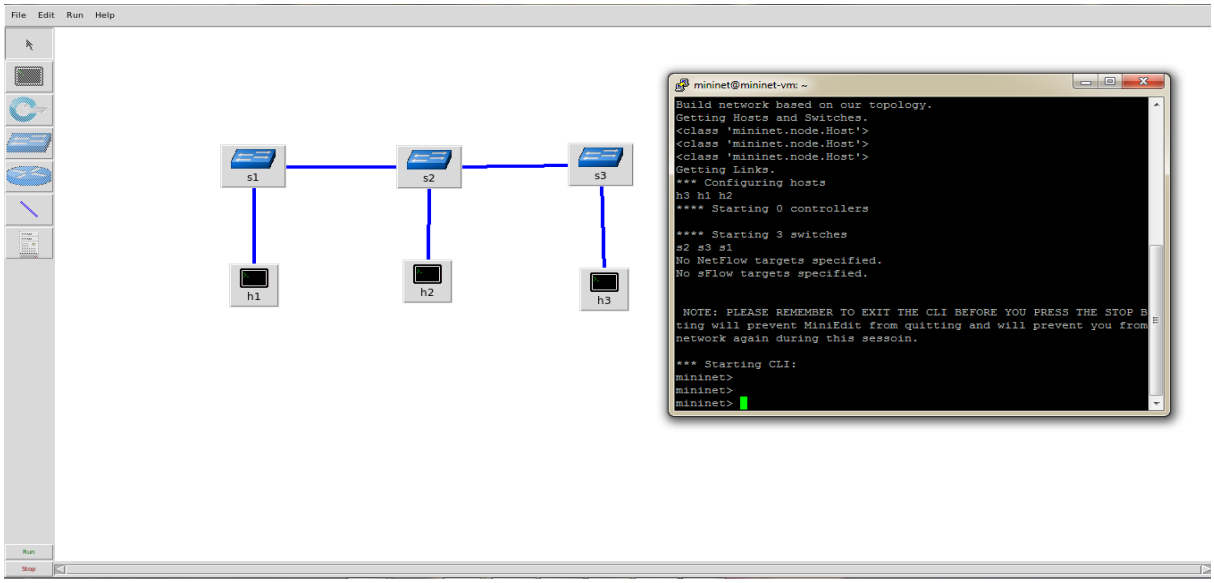


Figure 3.3 : Exécution du scénario de réseau *Miniedit*

3.4. Tests de performance

3.4.1. Test 1 : tester le temps de réponse

Pour voir que notre réseau fonctionne correctement et que les paquets peuvent être acheminés entre tous les équipements à travers toutes les routes de la topologie, nous avons Tracé des Pings vers plusieurs nœuds de réseau et nous avons constaté que le ping passe correctement.

Nous avons également lancé un ping continue entre la source « h1 » et la destination «h2 » pour savoir la manière avec laquelle le trafic traverse le réseau pour atteindre la destination.

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.270 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.073 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.073/0.141/0.270/0.091 ms
mininet>
```

Figure 3.4 : Ping réussi entre « h1 » et « h2 ».

Le temps de réponse de 3 paquets en ICMP entre les deux hosts est de 1998 ms

3.4.2. Test 2 : tester la bandwidth :

L'outil *iperf* est un outil de test de réseau couramment utilisé pour mesurer la bande passante et la qualité d'un lien réseau. L'outil peut créer des flux de données TCP et UDP et mesurer le débit d'un réseau qui transporte ces flux.

Figure 3.5 : Test de la largeur de bande

On tourne bien autour de 7Gbps/s de débit maximum.

3.5. Créé une topologie de réseaux SDNà l'aide de *Mininet*

3.5.1. Présentation de *Mininet*

Mininet est un *émulateur de réseau* qui crée un réseau d'hôtes virtuels, de commutateurs, de contrôleurs et de liens. Les hôtes *Mininet* exécutent un logiciel de réseau Linux standard et ses commutateurs prennent en charge Open Flow pour un routage personnalisé très flexible et une mise en réseau définie par logiciel.

Mininet prend en charge la recherche, le développement, l'apprentissage, le prototypage, les tests, le débogage et toute autre tâche qui pourrait bénéficier d'un réseau expérimental complet sur un ordinateur portable ou un autre PC. [22]

- Fournit un **banc** de **test réseau** simple et peu coûteux pour le développement d'applications Open Flow.
- Permet à **plusieurs développeurs simultanés** de travailler indépendamment sur la même topologie.
- Prend en charge **les tests de régression au niveau du système**, qui sont répétables et faciles à emballer.
- Permet **des tests de topologie complexes**, sans avoir à câbler un réseau physique.
- Inclut une **CLI** prenant en charge la topologie et Open Flow, pour le débogage ou l'exécution de tests à l'échelle du réseau.
- Prend en charge **les topologies personnalisées arbitraires** et inclut un ensemble de base de **topologies paramétrées.**

- Il est **utilisable** sans programmation, maisfournit également une **API Python** simple et extensible pour la création et l'expérimentation de réseau.

Mininet offre un moyen simple d'obtenir le *comportement* correct du système (et, dans la mesure prise en charge par votre matériel, les performances) et d'expérimenter des topologies.

Les réseaux *Mininet* exécutent du *code réel*, y compris les applications réseau standard Unix / Linux ainsi que le noyau Linux réel et la pile réseau (y compris toutes les

Extensions de noyau dont vous disposez, à condition qu'elles soient compatibles avec les espaces de noms réseau.)

Pour cette raison, le code que vous développez et testez sur *Mininet*, pour un contrôleur Open Flow, un commutateur modifié ou un hôte, *peut passer à un système réel avec des modifications minimes*, pour des tests, une évaluation des performances et un déploiement dans le monde réel. Surtout, cela signifie qu'une conception qui fonctionne dans *Mininet* peut généralement passer directement aux commutateurs matériels pour le transfert de paquets à débit linéaire

3.5.2. Création de topologie avec *Mininet*

Afin de créer notre topologie on doit ouvrir un terminal avec putty et on accèdevia le SSH à la machine virtuelle, puis on démarre l'émulateur *mininet*, celui-ci qui va créer la topologie propre, avec la commande suivante :

```
mininet@mininet-vm:~$ sudo mn --topo=linear,3 --controller=remote,ip=192.168.56
102
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.56.102:6653
Unable to contact the remote controller at 192.168.56.102:6633
Setting remote controller to 192.168.56.102:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

Figure 3.6 : Création d'une topologie avec *Mininet*

Une fois la topologie est créée avec *mininet*, tout le détail sera affiché dans la page d'accueil de hpe comme montre la figure 3.6.

La topologie est affichée par le contrôleur hpe, Cette topologie contient :

- Commutateur.
- Hôtes.
- Liens qui relient les commutateurs et les hôtes entre eux.

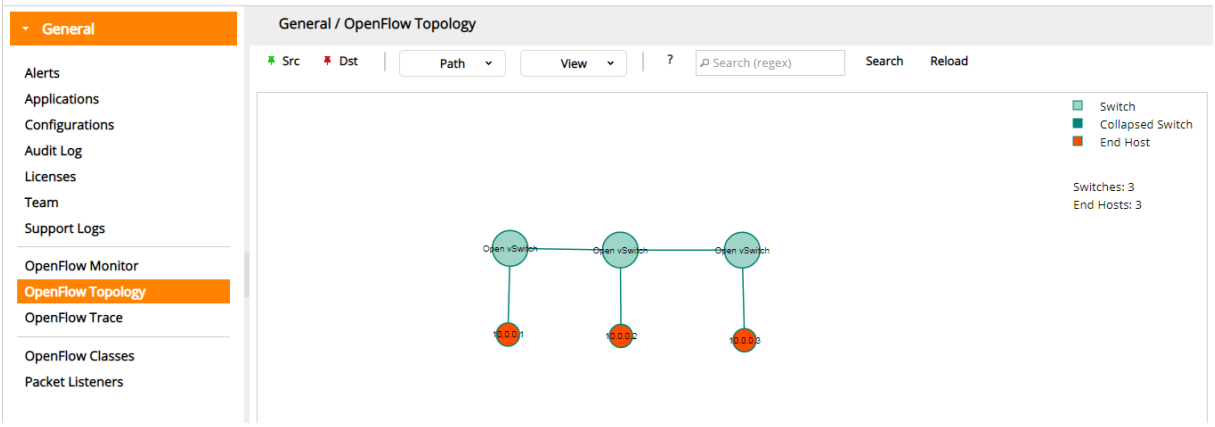


Figure 3.7 : Onglet Topologie de l'Interface web du hpe

3.6. Tests de performances

3.6.1. Test 1 : tester le temps de réponse

Ping mesure le temps d'aller-retour pour les messages envoyés de l'hôte d'origine à un hôte de destination qui sont renvoyés à la source, il fonctionne en envoyant des paquets de demande d'écho ICMP à l'hôte cible et en attendant une réponse d'écho ICMP.

La ligne de commande utilisée pour mesurer le temps de réponse est la suivante :

```
mininet> h1 ping -c 3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.537 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.083 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.082 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.082/0.234/0.537/0.214 ms
mininet>
```

Figure 3.8 : Mesure du temps de réponse

Le temps de réponse de 3 paquets en ICMP entre les deux hosts est de 2003 ms.

3.6.2. Test 2 : tester la bandwidth

Nous avons testé la performance du débit avec la commande suivante :

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.81 Gbits/sec', '9.83 Gbits/sec']
mininet>
```

Figure 3.9 : Test de performance du débit

Le débit mesurer entre les deux hôtes est de 10 Gbits/s au max.

3.7. Analyse des résultats etdiscussions

L'infrastructure est la principale différence entre les SDN et les réseaux traditionnels : le SDN repose sur un logiciel, tandis que les réseaux traditionnels sont basés sur du matériel. Le plan de contrôle étant basé sur un logiciel, le SDN est beaucoup plus flexible qu'un réseau traditionnel. Il permet aux administrateurs de contrôler le réseau, modifier les paramètres de configuration, provisionner les ressources et augmenter la capacité du réseau, et ce à partir d'une interface utilisateur centralisée, sans ajouter de matériel supplémentaire.

Une analyse comparative des performances entre le réseau traditionnel et le réseau compatible OpenFlow pour les topologies de réseau définies ci-dessus , Une simple connectivité réseau est testée en exécutant ' ping 'commande qui envoie un message de demande d'écho ICMP et attend sa réponse pour vérifier la connectivité IP entre les nœuds définis et pour trouver le temps de réponse aller-retour entre les nœuds, Le test est effectué entre l'hôte h1 et l'hôte h2, Il est clairement visible que le temps de réponse entre les hôtes du réseau OpenFlow est supérieur à celui du réseau traditionnel.

Ensuite, une comparaison entre le même réseau traditionnel et le réseau compatible OpenFlow en analysant le débit obtenu d'un réseau. Un débit de réseau est défini comme la quantité de données transmises du nœud source au nœud de destination dans une période de temps donnée. Le débit est généralement mesuré en bits par seconde (bps). Analytiquement, il peut être défini comme le rapport entre la bande passante maximale du récepteur et le temps d'aller-retour entre les nœuds,

Le débit entre les hôtes de données est plus élevé dans un réseau compatible OpenFlow que dans un réseau traditionnel. Cela signifie qu'une plus grande quantité de transfert de données aura lieu entre les nœuds dans une période de temps spécifique pour les réseaux OpenFlow que les réseaux traditionnels.

Sur la base des résultats obtenus, il est conclu que le réseau compatible OpenFlow est plus rapide en raison de son architecture de contrôle central par rapport au réseau traditionnel.

	Réponse de temps	bandwidth
Réseau traditionnel	1998 ms	7Gbits/s
Réseau SDN	2003 ms	10Gbits/s

Tableau 3-2 : Comparaison entre le temps de réponse etle débit dans les réseaux traditionnels et SDN

3.8. Conclusion

Dans ce chapitre, nous avons exploité *mininet*, *Miniedit*. Nous avons atteint le but de projet qui est l'étude comparative entre réseau traditionnel et réseau SDN. On a testé les performancesde réseau traditionnel et réseau SDN, ainsi qu'une comparaison entre ces performances.

L'architecture du réseau traditionnel n'est désormais plus commode. C'est ainsi que le fameux SDN entre dans l'histoire du réseau pour lui offrir beaucoup plus d'agilité, de performance et le rendre beaucoup plus facile à administrer. On peut facilement implémenter et étudier SDN grâce à son émulateur *mininet*.

Chapitre 4 : Conclusion générale

Un réseau défini par logiciel basé sur la technologie OpenFlow est la technologie prometteuse pour le futur d'internet et de NGN. Les réseaux traditionnels sont de conception complexe et sont en quelque sorte difficile à gérer en raison de l'intégration verticale du plan de contrôle et de données sur les périphériques de base du réseau. Ainsi, contrairement à l'architecture de réseau traditionnelle, l'architecture SDN est contrôlée de manière centralisée en séparant le plan de données et le plan de contrôle des dispositifs centraux du réseau, ce qui rompt à son tour le problème d'intégration verticale des réseaux traditionnels.

Ce projet a été pour nous une chance et une formidable opportunité de découvrir un réseau reposant sur les technologies SDN,Car c'est bien là que réside le point fort du réseau qui désormais n'évolue plus avec la lourdeur du matériel mais avec la célérité du logiciel. Le SDN va changer donc le quotidien des administrateurs réseau, moins pris par des opérations répétitives et fastidieuses, ils pourront se Consacrer à des tâches présentant plus de valeur ajoutée pour leur entreprise.

Bibliographie

Chapitre 5 : Bibliographie

[1] Mémoire de Fin d'Etudes De MASTER ACADEMIQUE Filière : Télécommunications Spécialité : Réseaux et Télécommunications *Présenté par :* Mr AICHAOUI Anis Mr AITBELKACEM Yanis.

[2] Mémoire de fin d'étude de masteracadémique présenté par MERIDJI Rania.

[3] Routagedirecthttps://www.ibm.com/knowledgecenter/fr/routage%20scope=SSEQTPconsulté le 06 aout 2020.

[4] Routage indirecthttp://www.iro.umontreal.ca/~kropf/ift6052/exercices/applets/applet5/introduc.htmconsulté le 06 aout 2020.

[5] Mémoire de fin d'étude de master présenter par NOUAL Ikaria.

[6] Table de routage : https://cisco.goffinet.org/consulté le 10 aout 2020.

[7] André vaucamps, « Le réseau avec Cisco » : page 424.

[8] Radia Perlman, « Interconnexions Ponts et routeurs », page 389-397.

[9] Modèle OSI et routage : http : //benabdellah-informatique.wifeo.com/Modle-OSI-etroutage.pdf consulté le 03 septembre 2020.

[10] Mémoire fin d'étude de master, Les réseaux tolérants aux délais : le routage et lesproblèmes d'acheminement.

[11] GuyPujolle et Marlène Schwartz, « Réseaux Locaux informatiques », page 228.

[12] EIGRP, http:// www.reseamaroc.com/files/EIGRP P%2023.PDF.consulté le 05 septembre 2020.

[13] GUERROUT EL-Hachemi. Optimisation des Protocoles de Routage avec la méthode de Colonie de Fourmis. Mémoire d'ingénieur d'état en informatique, Institut National de formation en Informatique (I.N.I) Oued-Smar, Alger, 2007.

[14] Mémoire de fin de cycle : proposition et misse en ouvert d'une solution de segmentation et de routage de réseau LAN étendu de la RTC Bejaïa (région transport centre) Sonatrach réalisé par oussalahe bilal, radouane salim.

[15] Mémoire de fin d'étude de master : protocole de routage a l'état de liaisons présenté par Mr.ibeghouchene Amar.

[16] Daniéle dromard, « Architecture des réseaux », page 201 à 208

[17] ArchitectureSDN:https://www.bicomm.fr/definition-sdn-software-defined-network/consulté le 20 septembre 2020.

[18] Définition open flow:https://www.sciencedirect.com/topics/computer-science/openflow-protocol consulté le 30 septembre 2020.

[19] Histoire OpenFlow:https://www.sdxcentral.com/networking/sdn/definitions/what-is-openflow/consulté le 01 octobre 2020.

[20] Mémoire de fin d'étude : Etude et mise en œuvre d'une solution SDN : Application de Gestion de VLANs présentée par - Mme BOUIDA Hafida née SAIDI.

[21] Les messages OpenFlow : https://ipcisco.com/lesson/open-flow-messages/consulté le04octobre 2020.

[22] www.mininet.org, consulté 20 octobre 2020.

Chapitre 6 : Annexe 1 : Installation et configuration de *Mininet*

- Télécharger la machine virtuelle à partir du lien :

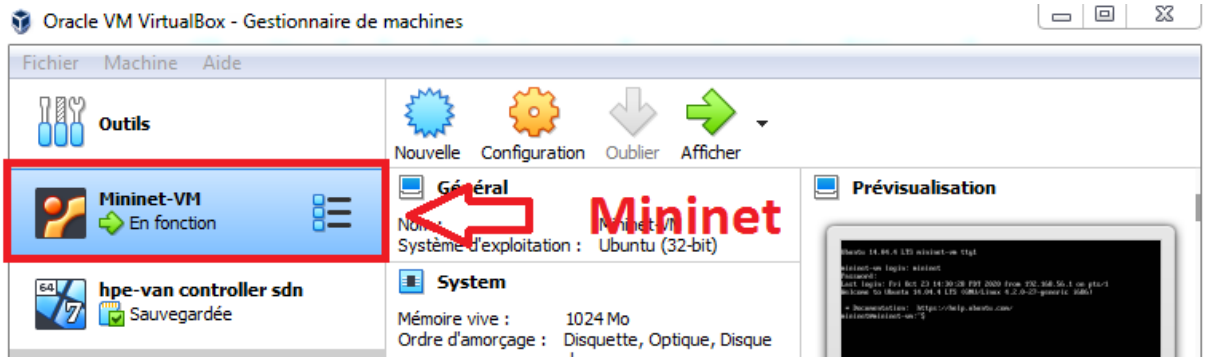
  https://github.com/mininet/mininet/wiki/Mininet-VM-Images

- Télécharger et installer un système de virtualisation. Le système le plusRecommandé et que nous l'avons utilisé est Virtualbox, il est gratuit et il peutFonctionner sous Windows et sous linux, a partir le lien suivant :

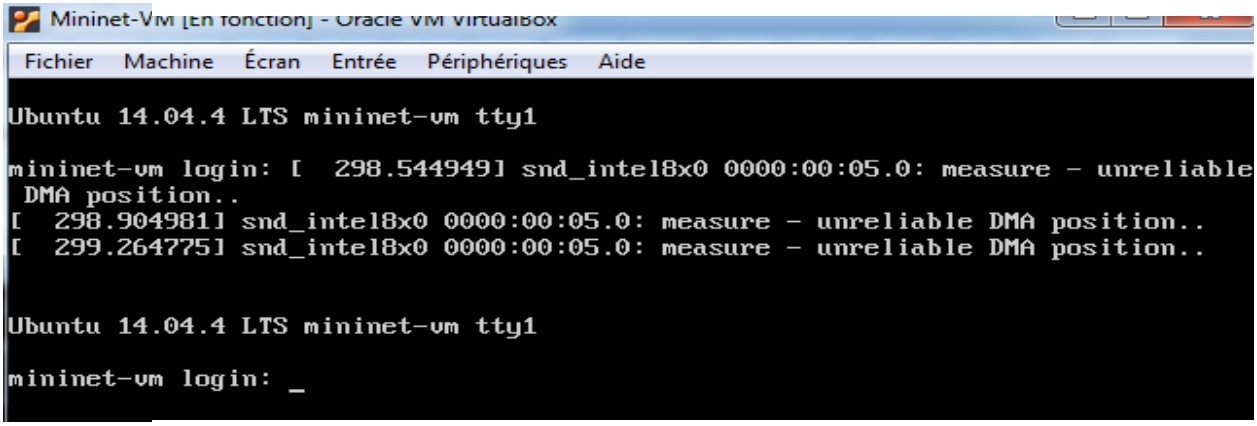
  https://www.virtualbox.org/wiki/Downloads

Apres le téléchargement de **l'image *Mininet*** et le programme de virtualisation **virtualbox** :

1) Connecter à virtualbox et ajouter le fichier de l'image Mininet



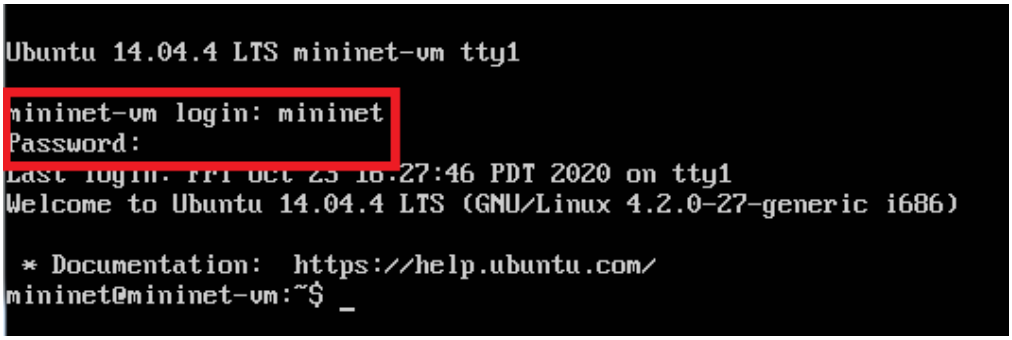
2) Lancer la VM*Mininet* une fois qu'elle est activée



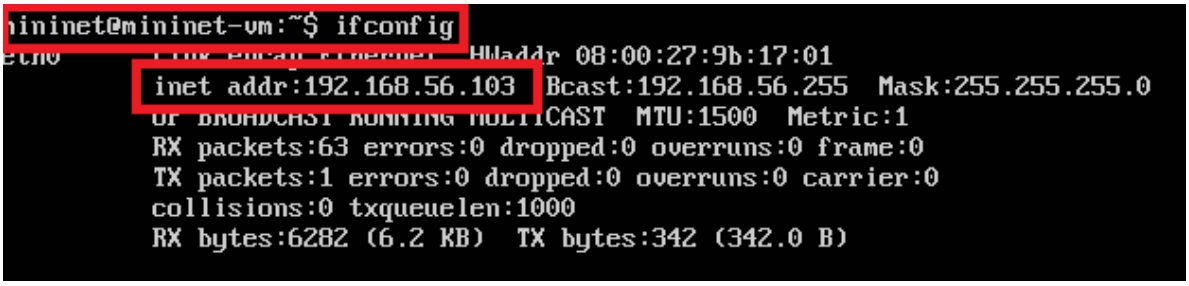
3) Se connecter à *Mininet* à l'aide d'un nom d'utilisateur et un mot de passe :

Login :**Mininet**

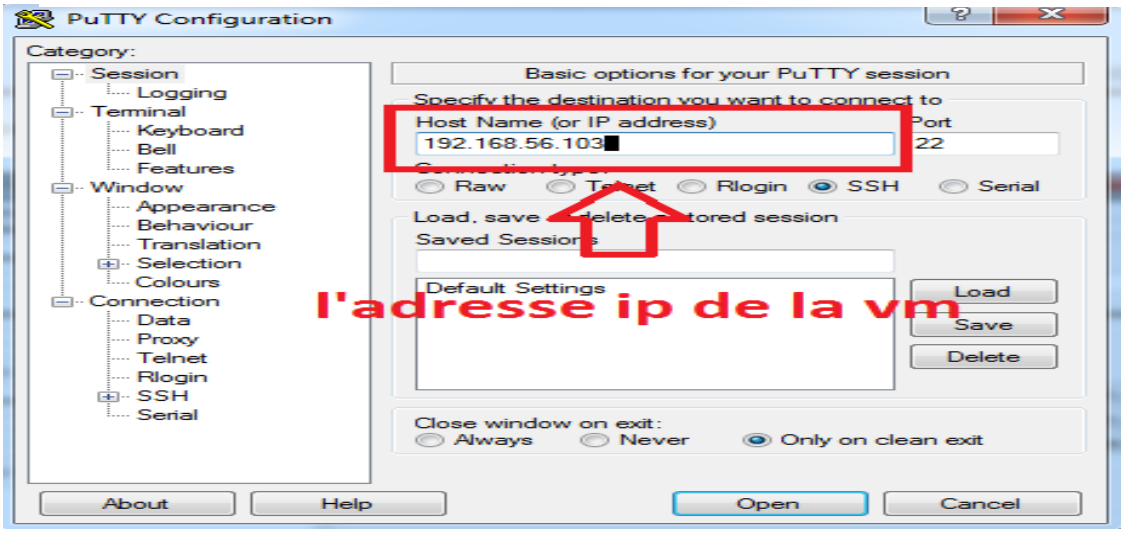
Password: **Mininet**



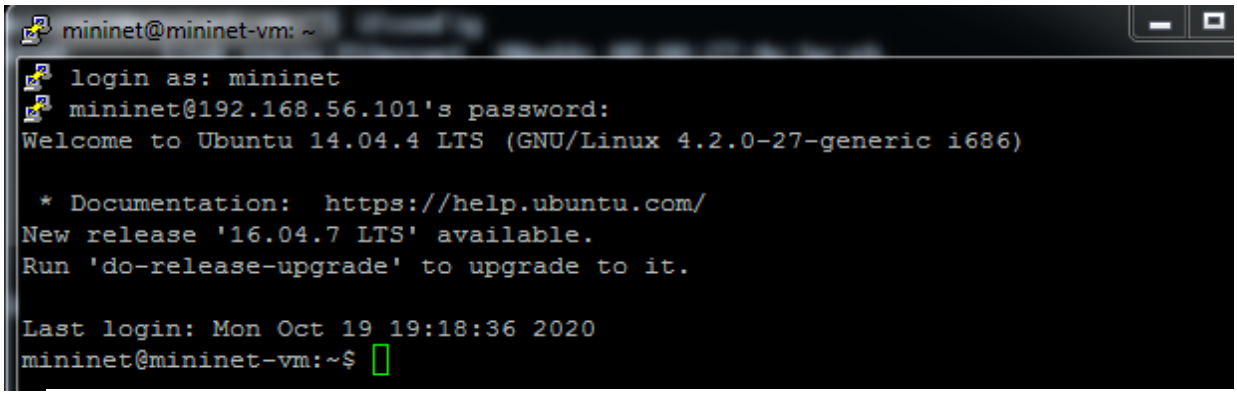
4) Exécuter la commande **ifconfig** pour trouver l'adresse IP de la VM*Mininet*



5) Ouvrez le terminal putty pour se connecter via ssh



6) Se connecter via une connexion ssh :

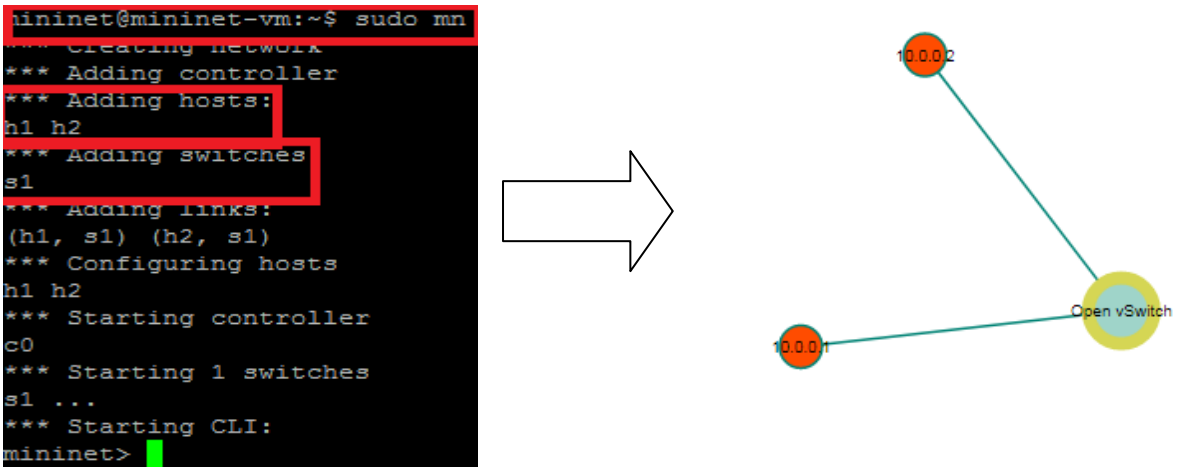


Différents types de topologies réseau sur *Mininet*

Plusieurs types de topologies sont disponibles par défaut avec *Mininet*.

La Topologieminimal

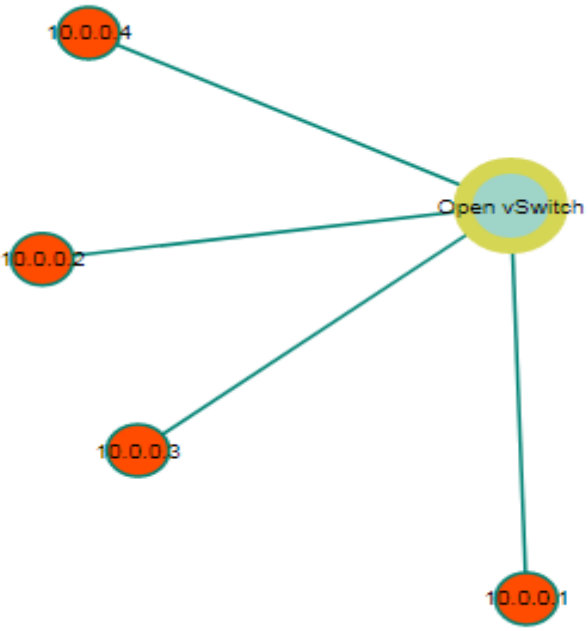
Nous utiliserons CLI pour gérer notre réseau virtuel, La topologie par défaut est la **minimal** topologie, qui comprend un commutateur (s1) de noyau OpenFlow connecté à deux hôtes (h1 et h2), plus le contrôleur (c0) de référence OpenFlow.



La topologie single

Qui se compose d'un nombre spécifié d'hôtes (h1, h2…hn) connecté à un seul commutateur (s1).La topologie suivante crée une topologie single avec 4 hôtes :

```
mininet@mininet-vm:~$ sudo mn --topo=single,4 --controller=remote,ip=192.168.56
102
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.56.102:6653
Unable to contact the remote controller at 192.168.56.102:6633
Setting remote controller to 192.168.56.102:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```



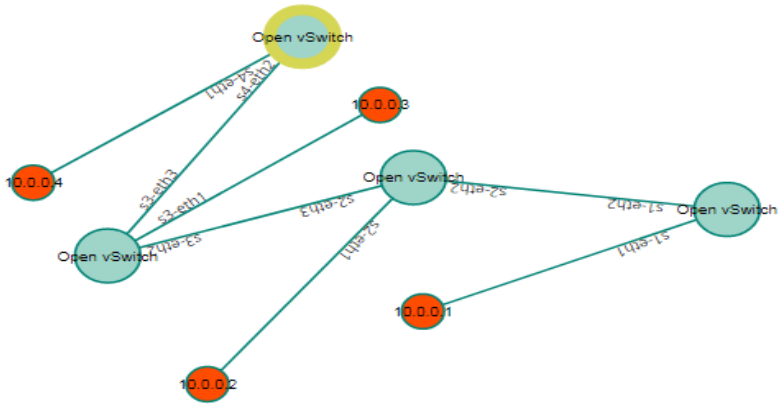
d

La topologie linéaire

Une topologie linéaire se compose de commutateurs connectés dos à dos, chacun ayant un seul hôte connecté à chaquecommutateur.

La topologie suivante crée une topologie linéaire avec 4 commutateurs et 4 hosts :

```
mininet@mininet-vm:~$ sudo mn --topo=linear,4 --controller=remote,ip=192.168.56.
102
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.56.102:6653
Unable to contact the remote controller at 192.168.56.102:6633
Setting remote controller to 192.168.56.102:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
```

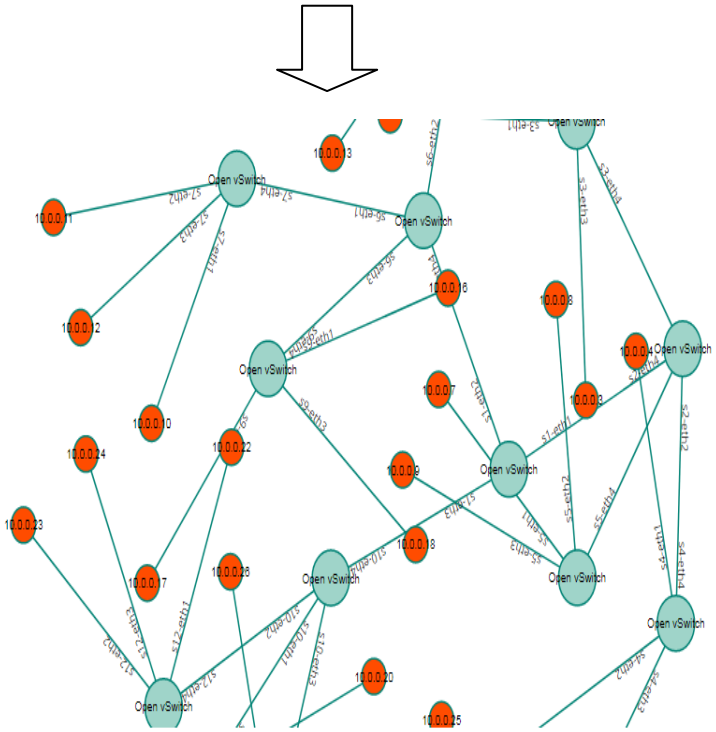


La topologie arborescente

Une topologie arborescente a un seul commutateur avec d'autres connectés en fonction d'un nombre de fanout. De sortie de 3 signifie que 3 commutateurs sont connectés au commutateur principal et que chacun de ces commutateurs a potentiellement 3 commutateurs connectés. Cela continue en fonction de la profondeur spécifiée. La fanout détermine également le nombre d'hôtes connectés à chaque commutateur feuille / bord.

La topologie suivante crée une topologie arborescente avec une profondeur 3 et une répartition de 3.

```
mininet@mininet-vm:~$ sudo mn --topo=tree,depth=3,fanout=3 --controller=remote,i
p=192.168.56.102
*** Creating network
*** Adding controller
Unable to contact the remote controller at 192.168.56.102:6653
Unable to contact the remote controller at 192.168.56.102:6633
Setting remote controller to 192.168.56.102:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13
*** Adding links:
(s1, s2) (s1, s6) (s1, s10) (s2, s3) (s2, s4) (s2, s5) (s3, h1) (s3, h2) (s3, h3
) (s4, h4) (s4, h5) (s4, h6) (s5, h7) (s5, h8) (s5, h9) (s6, s7) (s6, s8) (s6, s
9) (s7, h10) (s7, h11) (s7, h12) (s8, h13) (s8, h14) (s8, h15) (s9, h16) (s9, h1
7) (s9, h18) (s10, s11) (s10, s12) (s10, s13) (s11, h19) (s11, h20) (s11, h21) (
s12, h22) (s12, h23) (s12, h24) (s13, h25) (s13, h26) (s13, h27)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18 h19 h20 h21 h22 h
23 h24 h25 h26 h27
*** Starting controller
c0
*** Starting 13 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 ...
*** Starting CLI:
mininet>
```



Les commandes utiles

Pour exécute*Mininet* et Démarrez une topologie minimale il faut utiliser la commande suivante :

```
mininet@mininet-vm:~$ sudo mn
```

Mininet fournit une interface en ligne de commande qui peut être utilisée pour voir l'état du réseau et faire des tests dessus.

- *mininet>nodes* - Affiche les nœuds dans le réseau. h correspond à un host, s à un Switch et c a un contrôleur.
- *mininet> net* - Affiche la topologie du réseau. Quelle machine est connectée à quel host et le nom de l'interface réseau de connexion.
- *mininet> dump* - Affiche les adresses IP de chaque machine ainsi que le nom de la carte réseau.
- *mininet> h1 Ping h2* - Demande à l'hôte h1 d'effectuer un Ping sur l'hôte h2.
- *mininet> h1 ifconfig -a* - Effectue et affiche les résultats de la commande *ifconfig* sur la machine souhaitée.
- *mininet>pingall* - Permet de tester la connectivité du réseau. Toutes les machines vont se ping entre elles.
- *mininet>link s1 h1 down* - Enlève le lien réseau entre s1 et h1.
- *mininet>link s1 h1 up* - Remet le lien réseau entre s1 et h1.
- *mininet>xterm h1* - Ouvre une fenêtre en ligne de commande sur l'hôte h1.

Chapitre 7 : Annexe 2 : Installation du contrôleur hpe van SDN

Le contrôleurhpe van SDN

Le contrôleur HPE VAN SDN fournit un point de contrôle unifié dans un réseau compatible OpenFlow, simplifiant la gestion, le provisionnement et l'orchestration et permettant la fourniture d'une nouvelle génération de services réseau basés sur les applications.

Présentation du contrôleur hpe van SDN

Dans l'architecture SDN (Software Defined Networking) de Hewlett Packard Enterprise, les plans de contrôle et de données du réseau sont découplés les uns des autres, centralisant l'intelligence du réseau et soustrayant l'infrastructure réseau sous-jacente des applications. Le logiciel du contrôleur provisionne directement les commutateurs physiques et virtuels sous son contrôle via le protocole OpenFlow standard de l'industrie. Les ports réseau, les liaisons et les topologies sont tous directement visibles, ce qui permet une administration centralisée des stratégies et une sélection de chemin plus efficace basée sur une vue dynamique et globale du réseau. Cela simplifie considérablement l'orchestration des environnements multi-locataires et l'application de la stratégie réseau pour les clients mobiles et les serveurs.

Le contrôleur HPE VAN SDN est conçu pour fonctionner dans une variété d'environnements informatiques, y compris le campus, le centre de données, le fournisseur de services, le Cloud privé et le Cloud public. Le contrôleur HPE VAN SDN comprend :

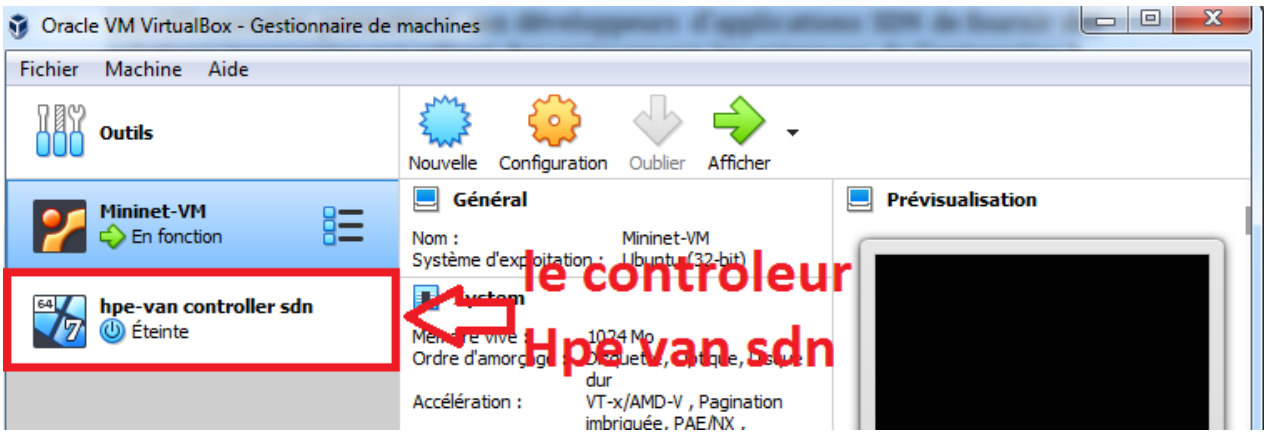
- Une plate-forme de classe entreprise pour la livraison d'une large gamme d'innovations réseau
- Une architecture de contrôleur extensible, évolutive et résiliente
- Conformité aux protocoles OpenFlow 1.0 et 1.3
- Prise en charge des commutateurs Hewlett Packard Enterprise et H3C OpenFlow
- Authentification sécurisée à l'aide d'un serveur Keystone local ou distant
- Association de contrôleurs pour la haute disponibilité (HA) et l'évolutivité des plates-formes distribuées
- Applications intégrées qui fournissent des services réseau communs

- Les API ouvertes permettent aux développeurs d'applications SDN de fournir des solutions innovantes qui relient dynamiquement les exigences de l'entreprise à l'infrastructure réseau à l'aide de programmes Java personnalisés ou d'interfaces de contrôle RESTful à usage général, y compris des fonctions permettant d'étendre l'API et l'interface utilisateur REST du contrôleur.
- Intégration avec HPE Intelligent Management Center (IMC). HPE IMC offre une gestion et une surveillance complètes du cycle de vie des applications de contrôleur, des rapports améliorés et une visualisation du réseau SDN.

Installation et configuration hpe van SDN

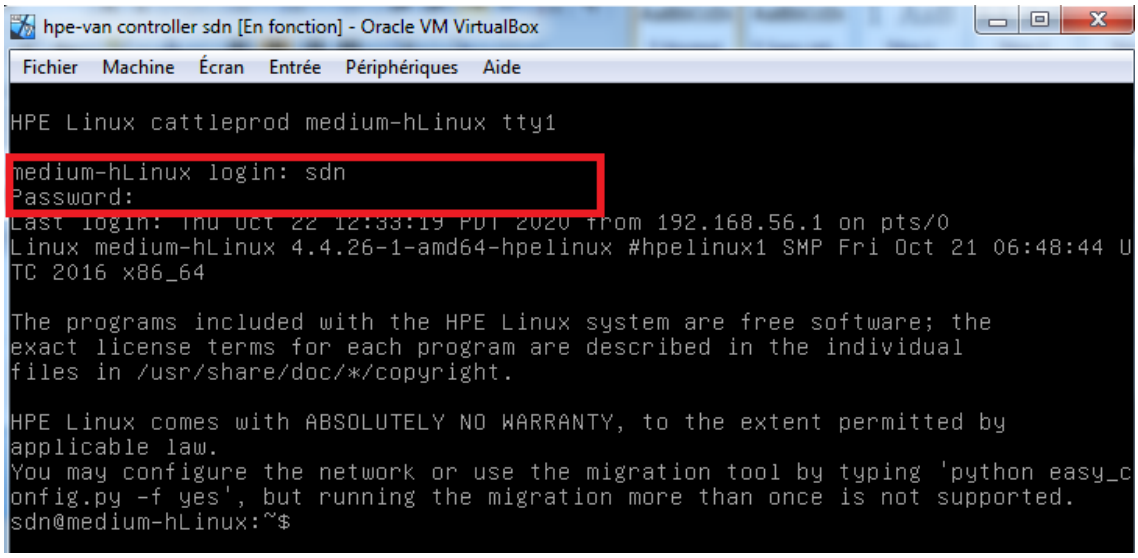
- Télécharger la source à partir du site github

1) importer le fichier de contrôleur hpe van sur virtualbox, après l'importation on démarre le contrôleur.

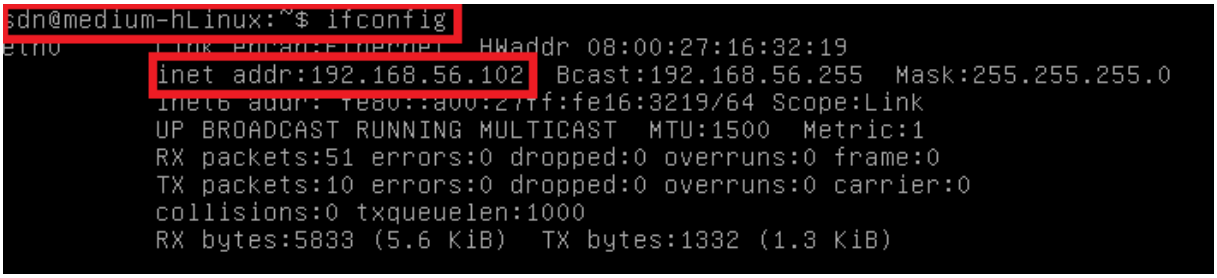


1) Entrer le nom d'utilisateur et le mot de passe pour connecter au contrôleur :

- **Le nom d'utilisateur : sdn**
- **Le mot de passe : skyline**

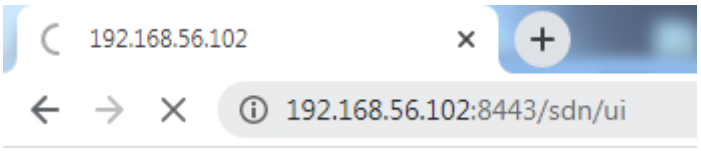


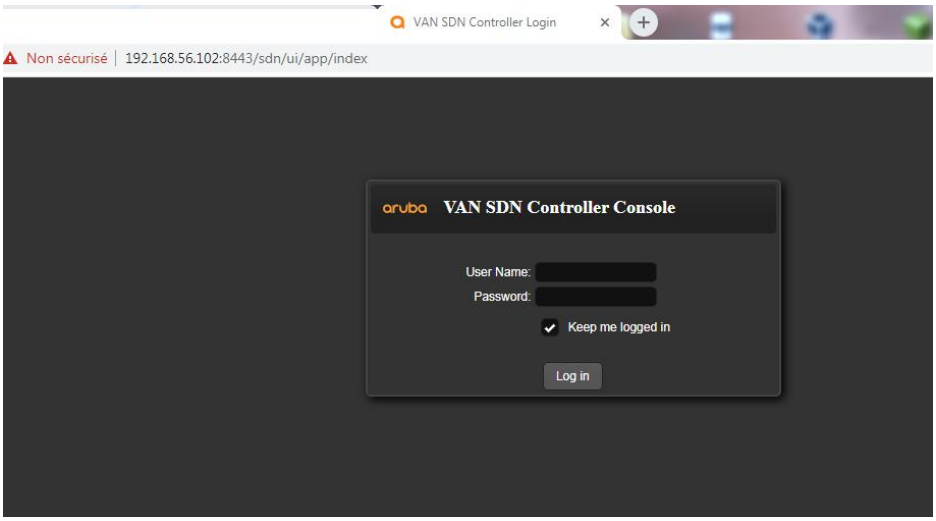
2) On cherche l'adresse IP de contrôleur sdn avec la commande ifconfig



Accéder au contrôleur Hpe van sdn :

Pour accéder au contrôleur, il faut connecter au navigateur web (chrome, …) avec l'adresse ip**19.168.56.102**




VAN SDN Controller Login
Non sécurisé | 192.168.56.102:8443/sdn/ui/app/index
aruba VAN SDN Controller Console
User Name:
Password:
Keep me logged in
Log in

k