REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE


**NIVERSITE MOHAMED BOUDIAF - M'SILA**
**FACULTE DE TECHNOLOGIE**

**DEPARTEMENT D'ELECTRONIQUE**


Mémoire de fin d'études

**Pour l'obtention du diplôme de Master-02 en Electronique**

***Option: Microelectronique***

*Thème*

# VIRTUALISATION DES RESEAUX SAN FIL (LTE)


Presenter par:

- Salem Mohamed.


Présenté le .. /.. / …. Devant le jury composé de MM:

- Mr : Ben hassan Madani — Encadreur
- Mr …….. — Président
- Mr……..... — Examinateur

**Année universitaire : 2019 /2020**

# Résumé

Le développement rapide de la technologie de communication sans fil a entraîné une augmentation explosive au réseau de communication et du nombre d'utilisateurs mobiles.

Un réseau de communications électroniques est constitué de plusieurs éléments ayant chacun une fonction bien particulière, parmi lesquels se trouvent des équipements en charge du contrôle d'accès au réseau, des pare-feux, des routeurs, des passerelles qui permettent l'interfaçage entre domaines distincts, des plateformes de service, des bases de données etc1.

Pour la plupart et historiquement, ces fonctions sont intrinsèquement indissociables de l'équipement (hardware) sur lequel elles sont exécutées; ce couplage fait que l'équipement et sa fonction sont vendus comme un unique produit intégré par les équipementiers. La virtualisation des fonctions des réseaux (« Network Function Virtualisation » ou NFV) brise ce couplage, en s'appuyant sur des solutions matures originaires du monde de l'informatique (technologies de l'IT) depuis la démocratisation du Cloud Computing. L'analogie peut être faite avec les smartphones qui regroupent les fonctions de plusieurs équipements : un seul objet peut remplir différentes fonctions telles que celles d'un téléphone, d'un appareil photo, d'une console de jeux, ou d'un podomètre, car elles sont réalisées par des logiciels2. En découplant les fonctions (logicielles) de l'équipement de son support matériel (hardware), la virtualisation permet d'acquérir indépendamment ces logiciels et de les installer sur des serveurs informatiques banalisés. Ces serveurs de grande capacité sont répartis sur quelques points de présence de l'opérateur (« data-centers »).

Le concept du NFV a été introduit en 2012 dans un Livre Blanc co-signé par 13 opérateurs, synthétisant les avantages de la virtualisation et invitant l'industrie à le développer pour le cadre des opérateurs télécoms. L'ETSI (European Telecommunications Standards Institute), à travers un groupe de travail dédié, a publié les spécifications d'un cadre commun permettant la mise en oeuvre de la virtualisation des réseaux (cf. annexe 1 pour plus de détails sur la structure du NFV) dans un environnement multi-vendeurs.

**Mots clés:** Virtualisation sans fil (WAV), Dépenses CAPEX / OPEX, Qualité de service QoS, Technologie de virtualisation "green Field", Déploiement progressif optimisé

**Keywords:** Wireless network virtualization,Cost CAPEX / OPEX, Quality of service QoS, Green field technology, low-cost optimization tool for progressive deployment

# Remerciement

Avant tout je tiens mes remerciements à mon dieu tout puissant De mon avoir donner la force et le courage.

Je saisons cette occasion pour addresser mes remerciements les plus profonds à Mes parents, mon encadreur Mr Ben hassan Madani qui a fourni des efforts énormes, par ses informations ses conseils et ses encouragements.

A tout les professeurs de département electronique.

A tout ce qui furent à un moment ou à tout instant partie prenante de ce travail et surtout monsieur Khanouf Saleh et monsieur laadjal Mohamed.

Mes remerciements également aux membres de jury pour l'intérêt qu'ils ont manifesté pour évaluer mon travail de maîtrise.

Mes plus chaleureux remerciements pour tous ceux qui de prés et de loin ont contribué à réalisation de ce projet.

# Table des matières

# Table des figures

# Liste des Abréviations et Acronymes

.

**1G :** 1ère Génération.
**2G :** 2ème Génération.

**3G :** 3ème Génération.
**4G :** 4ème Génération.
**5G :** 5ème Génération.
**3GPP :** 3rd Generation Partnership Project.

**ADSL :** Asymmetric Digital Subscriber Line.
**API :** Application Programming Interface .
**ARCEP :** Autorité de Régulation des Communications Électroniques et Postes.
**ASIC** : Application Specific Integrated Circuit.
**AUC :** AUthentification Center.

**BG:** Border Gateway.
**BSC:** Base Station Controller.
**BSS:** Base Station Sub-system.
**BTS:** Base Transceiver Station.

**CCU:** Channel Codec Unit.
**CDMA2000:** Code Division Multiple Access.
**CLI :** Command Line Interface.
**CGF:** Charging Gateway Function.

**CQI:** Channel Quality Indicator.

**DFT: T**ransformation de **F**ourier **D**iscrète.
**DiffServ:** Differenciated Service.
**DSCP:** DiffServ Control Protocol.

**EDGE:** Enhanced Data Rates for GSM Evolution.
**EIR:** Equipment Identity Register.
**eNodeB:** evolved NodeB**.**
**EPS:** Evolved Packet System.
**EPC:** Evolved Packet Core.
**E-UTRA:** Evolved-Universal Terrestrial Radio Access.
**eUTRAN:** evolved UTRAN.

**FDD:** Frequency Division Duplexing.

**GGSN:** Gateway GPRS Support Node.
**GMSC:** Gateway MSC.
**GPRS:** General Packet Radio Service.
**GSM:** Global System for Mobile Communication.

**HLR:** Home Location Register.
**HP :** Hewlett Packard .
**HSDPA:** High Speed Downlink Packet Access.
**HSOPA:** High Speed OFDM Packet Access.
**HSPA:** High Speed Packet Access.
**HSS:** Home Suscriber Service.
**HTTP :** HyperText Transfer Protocol.

**IEEE:** Institute of Electrical and Electronics Engineers.
**IETF:** Internet Engineering Task Force.
**IMEI:** International Mobile Equipment Identity.
**IMS IP:** Multimedia Sub-system.
**IMSI:** International Mobile Subscriber Identity.
**IP:** Internet Protocol.
**IPv6:** IP version 6.

**JDK :** Java Development Kit.

**KT:** Korea Telecommunication.
**KVM :** Kernel-based Virtual Machine.

**Iaas :** Infrastructure as a Service .
**LAN:** Local Area Network.
**LMT:** Latvijas Mobilais Telefons.
**LTE:** Long Term Evolution.

**M2M:** Machine To Machine.
**MAN:** Metropolitan Area Network.
**MIMO:** Multi Input Multi Output.
**MME:** Mobility Management Entity.
**MMS:** Multimedia Messaging Service.
**MS:** Mobile Station.
**MSC:** Mobile Switching Centre.
**MSISDN:** Mobile Station.
**MTS:** Mobile TeleSystem.

**NAT :** Network Address Translation.
**NIST :** National Institute of Standards and Technology.
**NMC:** Network and Management Centre.
**NS-2:** Network Simulator 2.
**NS-3:** Network Simulator 3.

**NSS:** Network Sub System.
**NTT:** Nippon Telegraph &Telephone.

**OFDM:** Orthogonal Frequency Division Multiplexing.
**OFDMA:** Orthogonal Frequency Division Multiple Access.
**OMC:** Operations and Maintenance Center.
**ONF :** Open Networking Foundation .
**OSI :** Open Systems Interconnection .
**OSS:** Operation Sub-System.

**PaaS:** Platform as a Service.
**PAPR:** Peak-to-Average Power Ratio.
**PCRF:** Policy and Charging Rules Function.
**PCU:** Packet Control Unit.
**PDN GW:** Packet Data Network Gate-Way.
**PDP:** Packet Data Protocol.
**PGW:** Packet Switch-GetWay**.**
**PLMN:** Public Land Mobile Network.

**QAM:** Quadrature Amplitude Modulation
**QdS:** Qualité de Service
**QoS:** Quality of Service
**QPSK:** Quadrature Phase Shift Keying.

**RDP :** Remote Desktop Protocol.

**RSVP:** Resource ReSerVation Protocol.

**RNC:** Radio Network Controller.

**RNIS:** Réseau Numérique à Intégration de Services.
**RTC:** Réseau Téléphonique Commuté.
**RTCP:** Real-time Transport Control Protocol.

**SaaS :** Software as a Service.
**SAE:** System Architecture Evolution.

**S-GW:** Serving-Get Way.

**SC-FDMA:** Single Carrier-Frenquency Division Multiplexing Access.

**SGSN:** Serving GPRS Support Node.
**SIM:** Subscriber Identity Module.
**SINR:** Signal Interference Noise Ratio.
**SMS:** Short Message Service.

**SMSC:** Short Message Service Center.
**SS7:** Signal Semaphore 7.

**TB:** Transport Block.
**TCP:** Transmission Control Protocol.
**TDD:** Time-Division Duplex.
**TLS :** Transport Layer Security.

**TMN:** Telecommunications Management Network.

**UDP:** User Datagram Protocol.
**UE:** Terminal Mobile.
**UIT:** Union Internationale des Télécommunications.
**UMTS:** Universal Mobile Telecommunications System.

**VLR:** Visitor Location Register.
**VoIP:** Voice over IP.

**VSF-OFCDM:** Variable Spreading Factor Orthogonal Frequency and Code Division Multiplexing.

**WAP:** Wireless Application Protocol.
**WCDMA:** Wide Coding Division Multiple Access.
**WiFi:** Wireless Fidelity.
**WIMAX:** Worldwide Interoperability for Microwave Access.

# Introduction générale

Le monde des réseaux et des télécommunications mobiles connait une évolution très rapide. Cette évolution s'accompagne toujours par une nouvelle technologie et plus de complexité pour subvenir à une forte recrudescence des demandes de clients en termes de débit et de disponibilité des services proposés. Pour répondre à cette

grande sollicitation des réseaux, les opérateurs télécom cherchent des solutions pour augmenter la capacité et la couverture tout en maintenant le KPI (Key Performance Indicator) souhaité.

Les opérateurs souhaitent déployer leurs réseaux et introduire de nouveaux et divers services plus rapidement et à moindre cout. Pour cela, ils doivent avoir des environnements de travail flexibles. C'est ici qu'intervient la virtualisation des fonctions réseaux (qu'on va étudier en détail par la suite) en permettant aux fournisseurs de services la possibilité d'évoluer plus rapidement et plus aisément au sein des nouvelles infrastructures telles que le LTE (Long Term Evolution). Elle permet aussi d'automatiser et de donner une certaine intelligence au réseau et d'utiliser plus efficacement les ressources pour accroitre ou diminuer les services. La virtualisation va apporter un plus en ce qui concerne la réduction des couts et les délais de commercialisation des services réseaux.

Les architectures des réseaux de télécommunications mobiles sont constituées de trois domaines essentiels, à savoir le domaine qui comprend les équipements propre à l'utilisateur, à savoir les terminaux, le domaine du réseau d'accès qui permet à l'abonné d'accéder aux ressources radio, et contribue à la gestion de sa mobilité, et enfin le domaine du réseau coeur qui regroupe l'ensemble des équipements assurant des fonctions telles que l'enregistrement de l'abonné au réseau et la mise à jour de sa localisation etc. Dans ce projet de fin d'études nous allons nous focaliser sur le domaine réseaux coeur, plus précisément le réseau coeur paquet EPC (Evolved Packet Core) de la quatrième génération de réseau mobile, et faire le lien entre la virtualisation et l'EPC pour la conception d'un vEPC (réseau coeur paquet virtualisé), considéré comme un bloc fonctionnel essentiel des réseaux de 5ème génération.

Ce projet se veut pertinent en ce qui concerne les nouvelles approches et technologies des réseaux de mobile, ceci détaillé en cinque chapitres. Dans le premier chapitre, nous allons revoir brièvement les quatre générations de réseaux mobiles et mettre l'accent sur les évolutions opérées et puis présenter les perspectives pour la cinquième génération.

Dans le second chapitre nous allons présenter l'IoT (Internet of Things) et la relation entre celle-ci et la téléphonie mobile. Puis on va étudier en détails le réseau coeur EPC du réseau mobile 4G et les différents composants. Lors du troisième chapitre nous allons introduire une nouvelle technique utilisée de plus en plus dans les réseaux, qui est la virtualisation. Le dernier chapitre (chapitre 4) est dédié à la présentation de notre application qui est un outil de dimensionnement d'un réseau coeur virtualisé vEPC.

# CHAPITRE (01)

## L’évolution des réseaux de télécommunication mobiles

### 1.1 : Introduction :

Au fil des années la téléphonie mobile a connu un réel essor. En effet la communication entre utilisateurs mobiles se développe et représente un marché immense en ce début du XXIe siècle.

Quatre générations se sont succédées, se distinguant chacune d'entre elles par la nature de la communication transitant dans le réseau : avec tout d'abord une communication analogique, ensuite une communication numérique sous forme circuit, puis troisième avec, en plus de la voix numérique, des applications sous forme paquet, Enfin la quatrième que nous vivons actuellement et qui propose beaucoup plus de débits et d'applications que la précédente.

A ces quatre générations s'ajoute une cinquième en cours de recherches et de développements et prévu pour 2020 offrant encore plus de débits ( de l'ordre de plusieurs Gbps par utilisateurs) et concrétisant la notion de société connectée. Dans ce chapitre on va étudier brièvement et se familiariser avec ces différentes technologies en prenant comme exemple les systèmes et standards NMT, GSM, GPRS, UMTS, LTE et en fin la 5G.

## 1.2 : Le NMT (Nordic Mobile Telephone):

C'est un système analogique de communications mobiles. Le système est développé en Suède, au Danemark, en Norvège et en Finlande. La première version, le NMT-450 était introduit en 1981, Le système a été utilisé par plusieurs pays, la plupart en Europe. Le NMT fonctionne à 450 MHZ sur la bande (450-470 MHZ). Ensuite la seconde version, NMT-900, a été introduite en 1986 Cette version fonctionne sur une fréquence de 900 MHZ, qui est utilisée en ce moment pour le GSM.
Cette première génération ne propose aucun service autre que la voix, elle repose sur une communication analogique. Elle n'a pas connu de réel succès en raison des coûts des équipements, les terminaux eux étaient lourds car ils n'ont pas connu de miniaturisation. Toutefois la flexibilité et la portabilité que propose cette technologie la rendent rapidement populaire. En effet les réseaux cellulaires de première génération ont été les premiers à rendre possible l'utilisation du téléphone mobile de façon continue.
Les problèmes rencontrés dans cette première technologie, qui présente certains avantages comme la flexibilité et la portabilité, ont poussé les chercheurs à améliorer la qualité de service. La numérisation et la miniaturisation des équipements ont rendu la communication mobile accessible et beaucoup plus flexible, ce qui a permis l'introduction de la deuxième génération de téléphonie mobile, notamment le GSM

## 1.3 : Le GSM (Global System for Mobile communication) :

Le GSM est la première norme de téléphonie mobile numérique. Le GSM a connu un grand succès, grâce notamment à équipements terminaux plus petits, plus maniable et plus facile à transporter, avec plus d'autonomie et à un coût moindre.

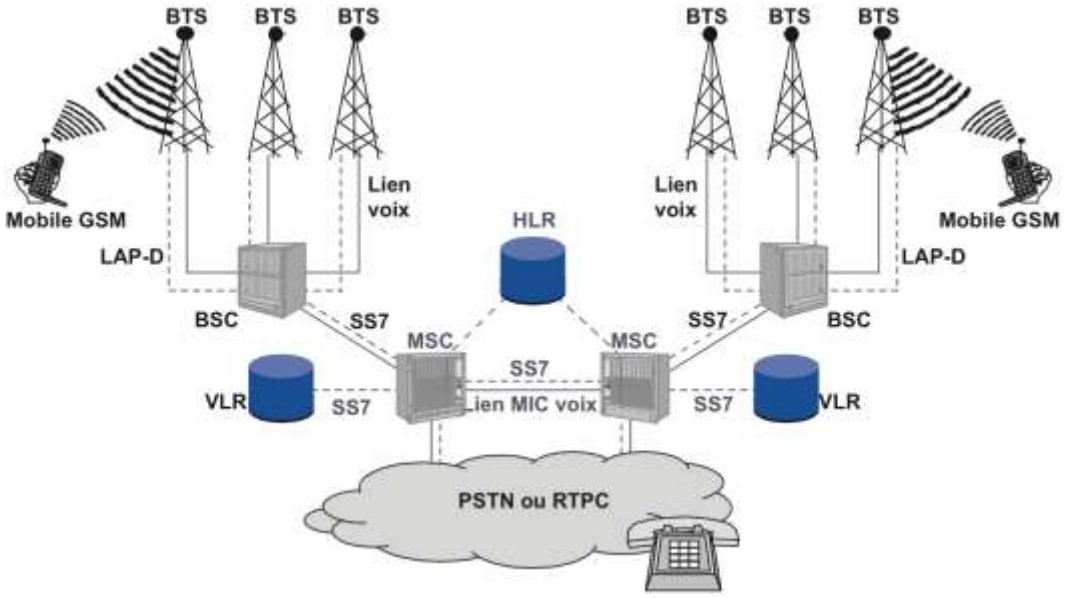


**Figure 1.1 Architecture d'un réseau GSM**

L'architecture d'un système GSM se décompose en trois sous-systèmes:

- Le sous-système radio (BSS, Base Station Subsystem):

Le BSS gère la partie radio des communications et est constitué de plusieurs BTS (Base Tranceiver Station), les BTS sont des émetteurs-récepteurs contrôlés par des BSC (Base Station Controller). Les BTS couvrent une zone géographique où éventuellement peut se trouver un utilisateur avec son terminal mobile MS (Mobile Station).

- Le sous-système réseaux (NSS, Network SubSystem): comprenant les commutateurs de coeur de réseau MSC (Mobile services Switching Center) associés aux bases de données VLR (Visited Location Register) et HLR (Home Location Register).

- Le sous-système d'exploitation: appelé l'OMC ou le centre et l'exploitation de la maintenance qui regroupe trois activités principales qui sont la gestion administrative, commerciale et technique. Il permet de gérer les fautes, les alarmes et les performances des équipements, contrôle les droits d'accès des gestionnaires au réseau, et assure l'interface homme-machine d'exploitation.

## 1.4 : Le GPRS (General Packet Radio Service) :

C'est un réseau mobile IP utilisant l'accès du GSM. Il est considéré comme la génération 2,5G ou la 2G+. C'est une mise à jour logicielle et matérielle des éléments de base du réseau GSM. Aussi on a introduit de nouveaux équipements

permettant l'accès au réseau de données (ex. Internet). Ces équipements sont le PCU, GGSN, le SGSN.

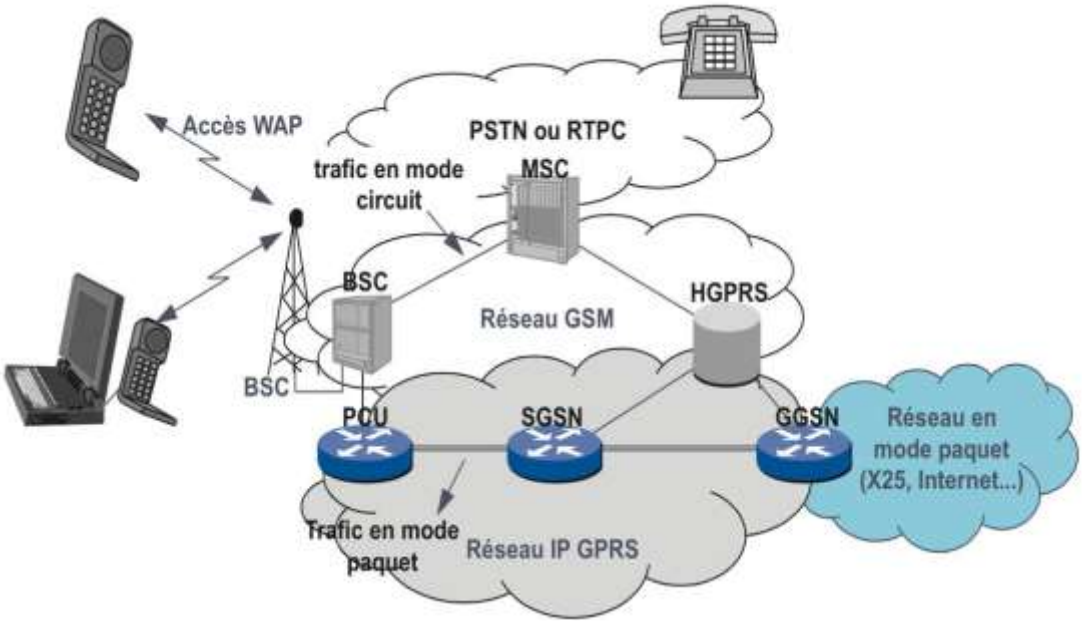


**Figure 1.2 Structure d'un réseau GPRS**

**PCU** (Packet Controler Unit): Le PCU assure l'adaptation des données issues du terminal mobile au format paquet et inversement.

**SGSN** (Serving GPRS Support Node ) : c'est un noeud qui gère la signalisation dans le réseau coeur GPRS, afin de permettre la gestion de la gestion de la mobilité, l'attachement de l'abonné et l'établissement des sessions IP.

**GGSN** (Gateway GPRS Support Node) : c'est la passerelle dans le réseau coeur du GPRS qui permet aux utilisateurs d'accéder à Internet. On retrouve également ces deux noeuds (SGSN, GGSN) dans le réseau coeur de l'UMTS qu'on verra juste après.
L'ensemble des équipements SGSN et GGSN forme ce que l'on appelle «le réseau fédérateur» ou Backbone.

## 1.5 : L'UMTS (Universal Mobile Telecommunications System) :

L'arrivée de l'année 2000 a vu apparaitre une nouvelle génération de réseau mobile, la 3G. Cette troisième génération a connu un fort déploiement dès l'année 2005. La nouveauté par rapport à la deuxième génération concerne l'introduction du mode paquet à l'exception de la parole téléphonique, qui reste très semblable à celle du GSM. Toutes autres informations, en dehors de la parole, sont mises dans des paquets et transportées dans un réseau à transfert de paquets.
On constate une nette amélioration des débits proposés par rapport au GSM, qui plafonne 9.6 Kbit/s, puisque le débit 3G atteint 384 kbit/s, lors de la première génération de l'UMTS. La 3G permet ainsi à l'utilisateur d'accéder aux premiers services multimédias.

Plusieurs types de schémas de modulations ont été étudiés pour l'émission numérique du signal. Il s'agit d'extensions des modulations classiques en fréquence, en amplitude et en phase. L'accès au canal radio utilise les techniques FDMA, TDMA et CDMA. Mais la méthode principale retenue pour la troisième génération est le CDMA. Les mobiles de la même cellule se partagent un canal radio par des techniques d'étalement de spectre, le système alloue un code unique à chaque client. [2]

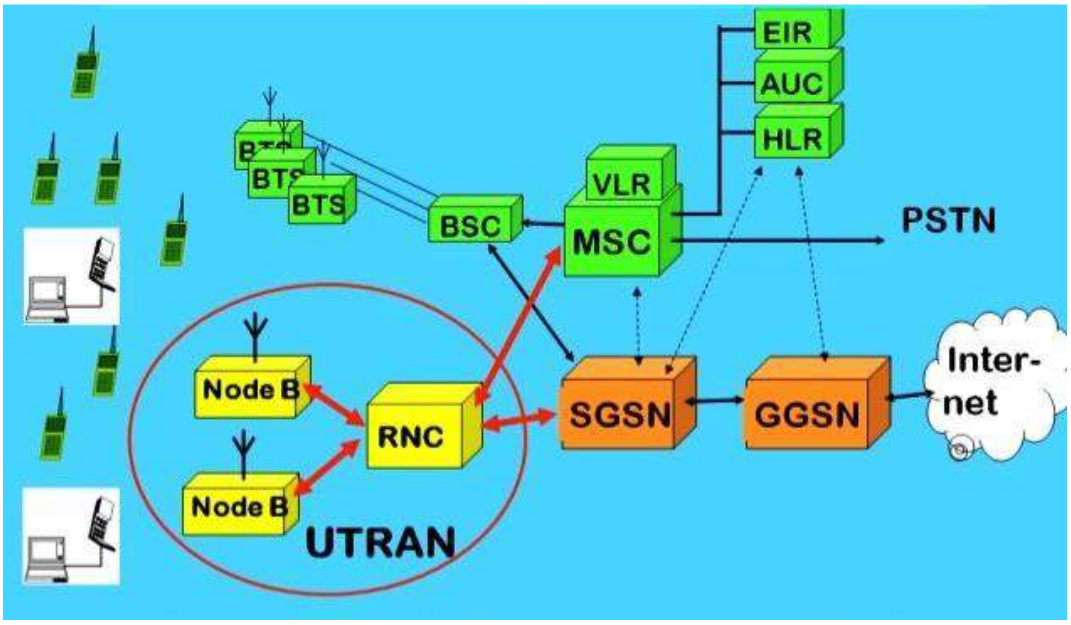


**Figure 1.3 Architecture de l'UMTS**

L'architecture du réseau UMTS est modulaire. Ses éléments constituants sont indépendants, de façon à autoriser en théorie des mises à jour de telle ou telle partie du système sans avoir à en redéfinir la totalité. Toutefois les règles de compatibilité et d'interopérabilité sont à observer.

L'UMTS définit trois domaines qui sont les suivants:

- le domaine utilisateur ;
- le domaine d'accès radio, ou UTRAN (Universal Terrestrial Radio Access Network) ;
- le domaine du réseau coeur (Core Network).

L'interface Iu permet de connecter la Radio au coeur paquet (Iu entre RNC et SGSN et Iu-U entre RNC et SGSN en cas de 3GDT).

Le domaine utilisateur est similaire à celui du GSM. Il se compose d'un terminal capable de gérer l'interface radio et d'une carte à puce, qui contient les caractéristiques de l'utilisateur et de son abonnement. En revanche, l'accès radio de l'UMTS (UTRAN) est complètement différent. L'UTRAN regroupe les stations de base NodeB et les contrôleurs de station de base, ou RNC (Radio Network Controller).

Le réseau coeur est composés de deux parties, le réseau coeur de type circuit contenants les commutateurs circuits, les MSC (Mobile Service Switching Center) et

le réseau coeur de type paquet composé de commutateurs paquet, les SGSN et GGSN (Serving and Gateway GPRS Support Node) qui relient le réseau de l'opérateur au monde extérieure.
Pour gérer les données relatives aux utilisateurs, telles que leur position dans le réseau, leur abonnement, etc., les bases de données introduites dans le GSM sont toujours présente dans l'UMTS. Il s'agit, entre autres, des HLR (Home Location Register), VLR (Visitor Location Register) et EIR (Equipement Identity Register).
La figure 1.3 décrit l'architecture générale de l'UMTS.

## 1.6 : La quatrième génération ou 4G :

C'est la génération de téléphonie mobile (et parfois fixe) succédant à la 2G et 3G. Elle permet le très haut débit mobile (débit théorique 150 Mbit/s, par cellule, voir plus) et permet également l'accès à plusieurs réseaux simultanément. L'une des caractéristiques de la 4G est d'avoir un réseau coeur basé que sur l'IP, c'est le Evolved Packet Core, en voulant simplifier l'architecture et de ne pas avoir, comme dans les générations précédentes, des réseaux coeur traitant les deux domaines circuit et paquet. La voix peut être transportée en VoIP (ou VoLTE, Voice over LTE). Il est possible aussi de permettre à l'utilisateur de passer des appels Circuit sur le réseau 2G/3G (CS Fallback).
Le contrôleur de station de base (RNC) a été supprimé pour permettre une architecture plus plate. L'architecture de la 4G sera étudiée en détail un peu plus dans ce chapitre.

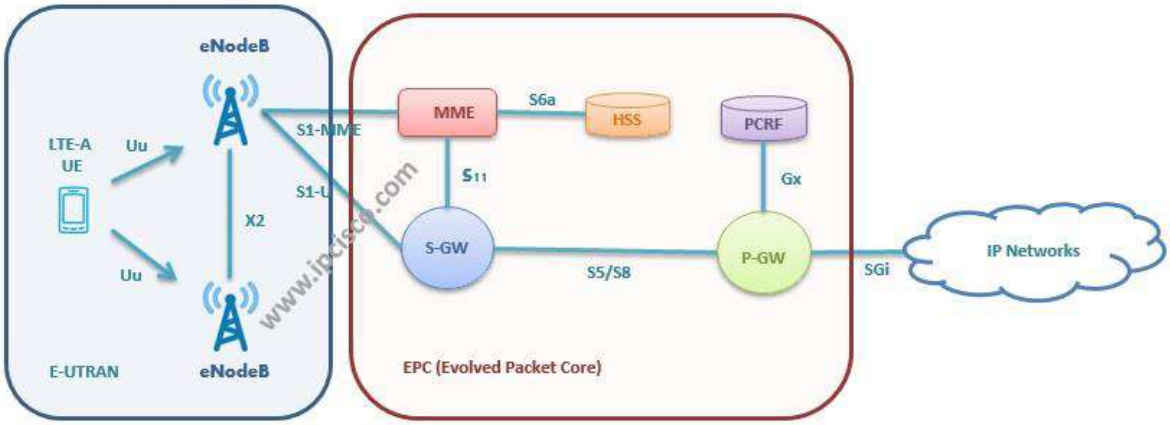


**Figure 1.4 Architecture 4G**

Les différents noeuds (MME, S-GW, P-GW etc.) seront traités en détail. Les stations de bases peuvent être Co-localisés avec les stations de bases 2G/3G avec un rajout de module matériel et logiciel, grâce aux fonctionnalités Multistandards offerts par la plupart des équipementiers.

Le tableau ci-dessous résume les caractéristiques de ces quatre générations de la téléphonie mobiles. La cinquième génération (5G) n'est pas encore normalisée, elle

le sera vers 2020. Mais on verra quand même dans les lignes suivantes quelques nouveautés apportées par cette dernière technologie.

| Génération | 1G | 2G | 2.5G | 3G | 4G |
|---|---|---|---|---|---|
| Standards | NMT, AMPS, TACS | GSM, IS95 A | GPRS, IS95 B | UMTS, CDMA2000 | LTE |
| Techniques d'accès | FDMA | TDMA | FDMA/TDMA | CDMA | OFDMA |
| Fréquences | 900 MHz | 900 et 1800MHz | 900 et 1800MHz | 1900-2024MHz 2110-2200Mhz | 800 MHz et 2600MHz |
| Débits réels | - | 9.6 Kbps | 48kbps | 384kbps HSPA 14,4Mbps HSPA+ 42Mbps | 150 Mbps |

**Figure 1.5 Tableau récapitulatif des quatre générations de réseaux mobiles**

## 1.7 : La cinquième génération ou 5G :

La cinquième génération de téléphonie mobile faisant suite à la 4G, permet des débits plus importants, le débit maximum devrait se situer entre 1 et 10 Gbit/s soit 100 à 1000 fois plus rapide que celui de la 4G. L'une des caractéristiques principales concerne l'internet des objets (IoT) qu'on va voir juste après, les applications IoT couvriront plus le domaine médical, le domicile (application domotique) et d'autres domaines.

Voici une la figure illustrant comment serait éventuellement l'architecture de la cinquième génération des réseaux mobiles.

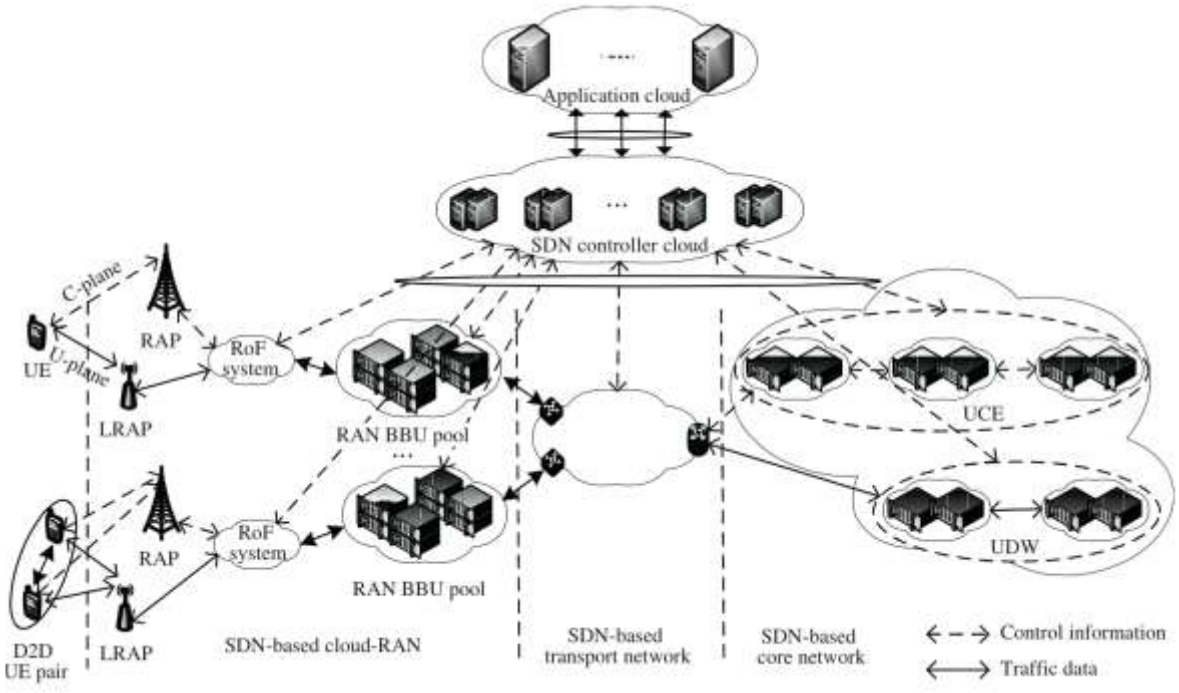


**Figure 1.6 Architecture 5G**

Cette technologie n'est pas encore normalisée, il se pourrait qu'il y ait des modifications. Cependant on peut citer des éléments et technologies clés caractérisant cette cinquième génération, ces termes vont aider le lecteur à mieux s'informer sur cette dernière génération de réseaux cellulaires :

- UWB (Ultra Wide Band), une très large bande passante

- Smart Antenna, c'est des antennes intelligentes capables de récupérer n'importe quel type de signal de différentes technologies radio (Wi-Fi, 2G, 3G etc.), utilisant des différentes techniques de codage et de modulation. Elles peuvent travailler pratiquement sur une fréquence quelconque.

- La virtualisation, que l'on va étudier en détail au chapitre 2, est à la base de la révolution du monde des réseaux, il suffit d'écrire un code qui fait exactement ce que fait le matériel. Ainsi toutes les machines matérielles peuvent ce transformées en machines logicielles (virtuelles) à l'instar d'un certain nombre d'élément comme une antenne ou un capteur.

- Le C-RAN (Cloud- Radio Access Network)

- Le RAP (Radio Access Point) et le LRAP (Light Radio Access Point)

- Le SDN (Software-Defined Networking),qu'on va voir au chapitre 3

- Multi-homing, c'est le multi-accès

- Machine to Machine (M2M) ou bien Device to Device (D2D).

L’architecture réseau est basée sur le Cloud RAN (C-RAN) et le SDN (Softwar-Defined Networking) (voir chapitre 3). En plus du réseau de transport et du réseau coeur l’architecture 5G est constituée d’un Cloud d’application et un Cloud de contrôleur SDN.

## 1.8 : Réseau sans fil LTE :

Cette réseaux LTE devra répondre aux besoins qui seront les nôtres à l’horizon 2020 et probablement pour la décennie qui suivra. L’enjeu est donc de ne pas se tromper au moment des choix menant à sa conception, d’identifier les priorités et de garantir son évolutivité. Il est difficile d’imaginer ce que sera exactement ce nouveau système aujourd’hui, néanmoins nous pouvons caractériser la 5ème génération mobile au travers:

- Des exigences définies par les différents acteurs de l’écosystème des télécommunications
- Des nouveaux enjeux économiques et sociétaux qui apparaissent dans nos sociétés
- Des limitations et opportunités technologiques contemporaines.

Aujourd'hui, les normes les plus courantes, sont la LTE et la LTE Advanced, qui correspondent aux technologies mobiles de quatrième génération. elles sont commercialisées sous l'appellation 4G et 4G+ par les opérateurs mobile, et permettent des débits théoriques pouvant aller jusqu'à 1 Gb/s. (Testez votre connexion 4G ici avec 4G Monitor).

La LTE (Long Term Evolution), est la technologie par excellence des réseaux sans-fil à l’échelle mondiale. Elle est nettement supérieure aux technologies précédentes, comme les réseaux 3G, 4G, et HSPA+. C’est l’évolution de toutes les normes de téléphonie mobile. La LTE offre une expérience Internet plus rapide, plus riche, et plus fluide que jamais. De plus, elle possède la capacité de permettre à un plus grand nombre d’utilisateurs d’accéder au réseau sans-fil à grande vitesse sans compromettre la performance. Cela signifie que vous pouvez télécharger du contenu plus rapidement et bénéficier du stockage en nuage, de la lecture en continu de contenu multimédia à haute définition et des jeux multijoueurs en ligne sans les frustrations associées au décalage et au temps d’attente. Autrement dit, vous libérez vraiment le pouvoir de votre appareil sans-fil. C’est comme si vous exploitiez au creux de votre main le plein potentiel de votre réseau résidentiel à large bande. En d’autres termes, c’est la convergence parfaite d’Internet, du mobile, des nouvelles technologies et de tout ce qu’il s’y rapporte.

Le réseau LTE offre, en théorie, des vitesses de téléchargement pouvant atteindre100 Mbps.

Les vitesses types des clients se situent vers 45 Mbps pour la plupart des appareils, voire jusqu’à 40 Mbps pour certains appareils sur le réseau LTE. Ce réseau est prêt pour l’avenir et l’avancement des appareils.

Le réseau LTE est basé sur SAE (System Architecture Evolution) qui est une évolution du réseau GPRS (General Pack t Radio Service), et comme on peut le voir dans la Figure 1.7, LTE est devenu un réseau complètement dépendant du protocole IP.

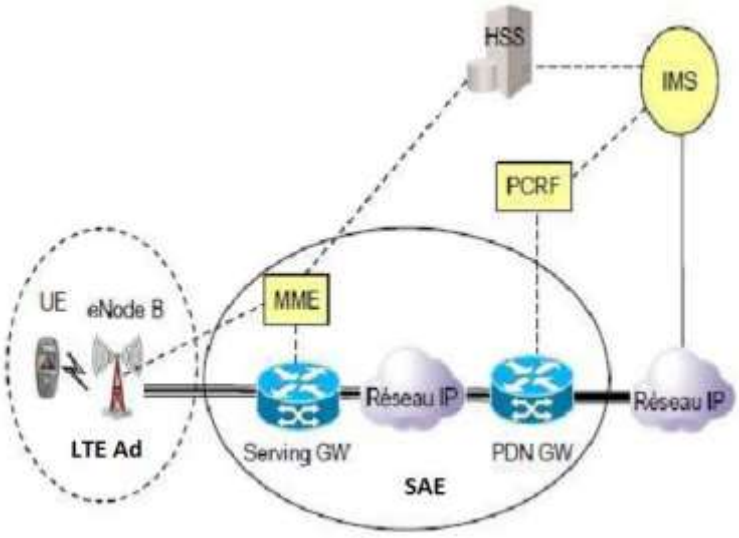


**Figure 1.7 L'architecture générale du LTE**

Les principales interfaces entre toutes les entités du système partagent des protocoles similaires pour faciliter la communication et la mise à niveau. C'est une architecture à commutation de paquets, largement inspirée par son prédécesseur (3G), et ses principales entités sont :

- **eNodeB :** est la station de base LTE qui communique directement avec l'équipement des utilisateurs (UE). Elle est équivalente à la station de transmission de base (BTS) dans les réseaux GSM.

- **UE (User Equipment) :** c'est l'équipement (ou terminal) utilisateur, qui est habituellement le téléphone mobile connecté au réseau pour transmettre de la voix et les données internet.

- **MME (Mobility Management Entity) :** c'est l'entité qui gère l'accès au réseau, elle s'occupe de toute communication avec les clients, comme l'enregistrement des abonnements, le marquage la retransmission. Le module est responsable de l'authentification des utilisateurs par exemple en choisissant le SGW pour l'utilisateur. Il 'occupe aussi du transfert de l'authentification et d'autres procédures qui maintiennent une communication continue.

- **SGW (Serving Gateway) :** fonctionne comme un routeur et s'occupe de tous les paquets échangé avec tou les utilisateurs en jouant le rôle de relais entre le anciennes technologies 3GPP et LTE. Le trafic global d'un UE en particulier passe par un seul SGW.

- **PDN-GW (Packet Data Network Gateway) :** c'est le point de sortie des paquets internes d'un UE vers les réseaux externes (passerelle PD), donc il fonctionne comme une interface entre les protocoles 3GPP et non 3GPP. Un UE peut avoir plusieurs PD -GW (appelé PGW au i) qui lui sont assignés.

- **HSS (Home Subscriber Server) :** il agit comme la base de données principale du réseau LTE et contient des informations comme les comptes utilisateurs, abonnements, sessions d'appel et autre données utiles.

- **ePDG (Evolved Packet Data Gateway) :** c'est une passerelle qui sécurise les paquets qui circulent entre les UE et la voie aux réseaux externes. À cette fin, l'ePDG établit un tunnel IPsec avec les UE (Davis, 2001).

- **AAA (Serveur 3GPP) :** offre des services comme l'authentification et l'autorisation d'accès aux utilisateurs non 3GPP. La couche physique LTE a été conçue pour le mode duplex intégral, de sorte que l'information est transmise dans le deux sens sans affecter le performances, même si le mode TDD est permis aussi, le mode FDD est le plus utilisé par 1es fournisseurs des services de communication.

L'objectif principal de la conception du réseau LTE est d'atteindre une meilleure performance comparée à ses prédécesseurs, ce qui fait que LTE a un débit plus élevé, une latence plus faible et a été optimisé pour les paquets IP. La couche physique conçue pour supporter des mode de transmission flexibles avec l'introduction des échanges par plusieurs antennes.
La couche physique LTE a été conçue pour fonctionner dans des bandes de fréquences allant de 1 ,4 à 20 MHz.

La composition et l'administration des structures de réseaux posent de grands défis à de nombreuses entreprises. Comme les réseaux conventionnels basés sur du materiel physique ne répondent que rarement aux besoins des entreprises modernes, le choix de solutions externes Infrastructure-as-a-Service-solutions (Iaas) est de plus en plus fréquent. Par rapport aux infrastructures internes traditionnelles, ces services Cloud, qui permettent aux clients d'accéder à des ressources informatiques virtualisées, se caractérisent par un degré élevé de flexibilité et une excellente maîtrise des coûts, contrairement à un cadre matériel fixe, les ressources souhaitées peuvent être mises à l'échelle à tout moment par simple pression sur un bouton.
Dans la plupart des cas, l'approvisionnement et la mise à l'échelle des ressources virtuelles (tant de la part du client que du fournisseur) s'effectuent à l'aide de logiciels, sans qu'il soit nécessaire d'accéder manuellement aux différents composants physiques du réseau. Le concept de réseau sous-jacent est également appelé réseau défini par le logiciel (SDN).

## 1.9 : Conclusion :

A l'issu de ce chapitre on a pu voir les différentes étapes qu'a connu la téléphonie mobile qui est à ce jour en plein développement. On a suivi globalement l'évolution des réseaux mobiles en passants du mode analogique au mode circuit numérique, ensuite au mode paquet.

# CHAPITRE (02)

## Contexte générale de La virtualisation

## 2.1 : Introduction :

Un réseau de communication est par définition un ensemble de ressources matérielles et logicielles mis en place pour offrir aux usagers un ensemble de services. Avec l'évolution des services de télécommunications et des trafics de données multimédia, les opérateurs ont déployé plusieurs technologies dans le but d'augmenter la capacité et les fonctionnalités des réseaux.

La gestion des ressources d'un réseau garantissant une qualité de service QoS adéquate est devenue un enjeu majeur lors de la planification du réseau.

Bien qu'Internet soit largement considéré comme le grand succès de ces dernières années, il est devenu une infrastructure critique en raison de son ossification. Cette ossification est principalement causée par l'absence de changement dans le réseau coeur et par la rigidité des équipements déployés. Elle rend la mise en place et le déploiement de nouveaux services réseaux difficiles et coûteux. D'autre part, avec l'émergence des services multimédia, la gestion de la qualité de service dans le réseau Internet actuel est devenue une tâche difficile. En effet ces nouveaux services comme la vidéo à la demande (VoD) ou la téléphonie sur IP (VoIP) nécessitent différents types de QoS que l'architecture actuelle d'Internet ne peut offrir. Cette QoS comporte diver's paramètres tels que la fiabilité, le débit ou encore le délai de bout en bout.

Le service "best effort" ne permet d'offrir une QoS que lorsqu'il n'y a aucune congestion dans aucun lien entre le serveur et l'utilisateur final. Puisqu'on ne peut pas prédire une congestion provoquant l'augmentation des délais ou des pertes de paquets, cette QoS ne peut pas être garantie tout le long de la session. Les services multimédia sont très sensibles à ces facteurs.

Les problèmes de résistance au facteur d'échelle de IntServ et la gestion de la QoS par classe de service dans DiffServ ne permettent pas de garantir cette QoS.

La virtualisation est une technique bien qu'ancienne mais très efficace pour consolider les ressources offrant une abstraction qui masque les détails d'un équipement.
La virtualisation des réseaux a été présentée comme un nouveau paradigme pour les nouvelles architectures réseaux et pour l'Internet du futur. Elle permet d'offrir une diversité de réseaux en masquant l'hétérogénéité de l'infrastructure physique et une flexibilité en contournant la rigidité des équipements réseaux.

La virtualisation des réseaux est définie comme une nouvelle technologie qui permet la segmentation logique des liens et des noeuds physiques d'un réseau en des noeuds et des liens virtuels.

Ceci permet de créer des réseaux logiques appelés réseaux virtuels en interconnectant les noeuds virtuels par des liens virtuels. La virtualisation des réseaux fait preuve de beaucoup de promesses garantissant la qualité de service

requise. Elle permet une gestion et un contrôle optimisé de l'infrastructure physique partagée par plusieurs services. En effet, un fournisseur de services peut déployer un mécanisme de gestion de la QoS spécifique pour chaque type de service déployé. Il peut aussi agir sur l'allocation de la bande passante de chaque lien physique utilisé par le réseau virtuel (VN) déployé. La flexibilité introduite par cette technologie permet aux FSs la gestion de la QoS de bout en bout.

Aujourd'hui les fournisseurs d'accès à Internet (FAIs) ont deux rôles à jouer. Le premier consiste à gérer leur infrastructure du réseau. Le second est dédié aux services sur Internet à offrir aux utilisateurs finaux.

La virtualisation des réseaux introduit un nouveau modèle d'affaires (business model) qui permet d'avoir deux acteurs distincts. Le premier acteur est le fournisseur de l'infrastructure physique (FIP) qui est le propriétaire de l'infrastructure réseau. Il est responsable du déploiement et du maintien des ressources physiques du réseau (routeurs, liens, etc).

Le second acteur est le fournisseur de services (FS) qui déploie des protocoles en louant les ressources d'un ou de plusieurs fournisseurs d'infrastructure physique pour créer un VN. Ce dernier ne possède pas d'infrastructure réseau. il a la responsabilité de délivrer des services de bout en bout aux différents utilisateurs. Le FIP permet aux différent FSs de partager son infrastructure grâce à des interfaces programmables. Il peut rejeter la demande d'un FS si un accord ne peut être conclu entre eux suite à une sur-utilisation des ressources. À titre d'exemple, imaginons qu'un fournisseur de service de vidéo à la demande (comme Netflix) déploie un équipement chez l'utilisateur pour lui permettre de visionner des vidéos à sa demande. Ce FS loue l'infrastructure physique de différents FIPs (comme SFR, Orange, etc) pour créer des réseaux virtuels et fournir ce service de bout en bout avec garantie de service.

La **Figure 2.1** présente un exemple d'environnement réseau virtualisé où deux fournisseurs de services louent des ressources de deux fournisseurs d'infrastructure afin de créer à chacun son propre réseau virtuel sur lequel il déploie un service de bout en bout.

La virtualisation des réseaux vise à offrir une meilleure flexibilité supportant plusieurs topologies et mécanismes de routage et de transfert de paquets. Elle permet à une configuration du VN d'être indépendante de celle des autres VNs. Son objectif est d'offrir une meilleure gestion du réseau tout en maximisant le nombre de VNs qui coexistent sur une même infrastructure physique. Elle permet également d'offrir une isolation des flux de chaque VN pouvant être créé sur plusieurs infrastructures réseaux hétérogènes.

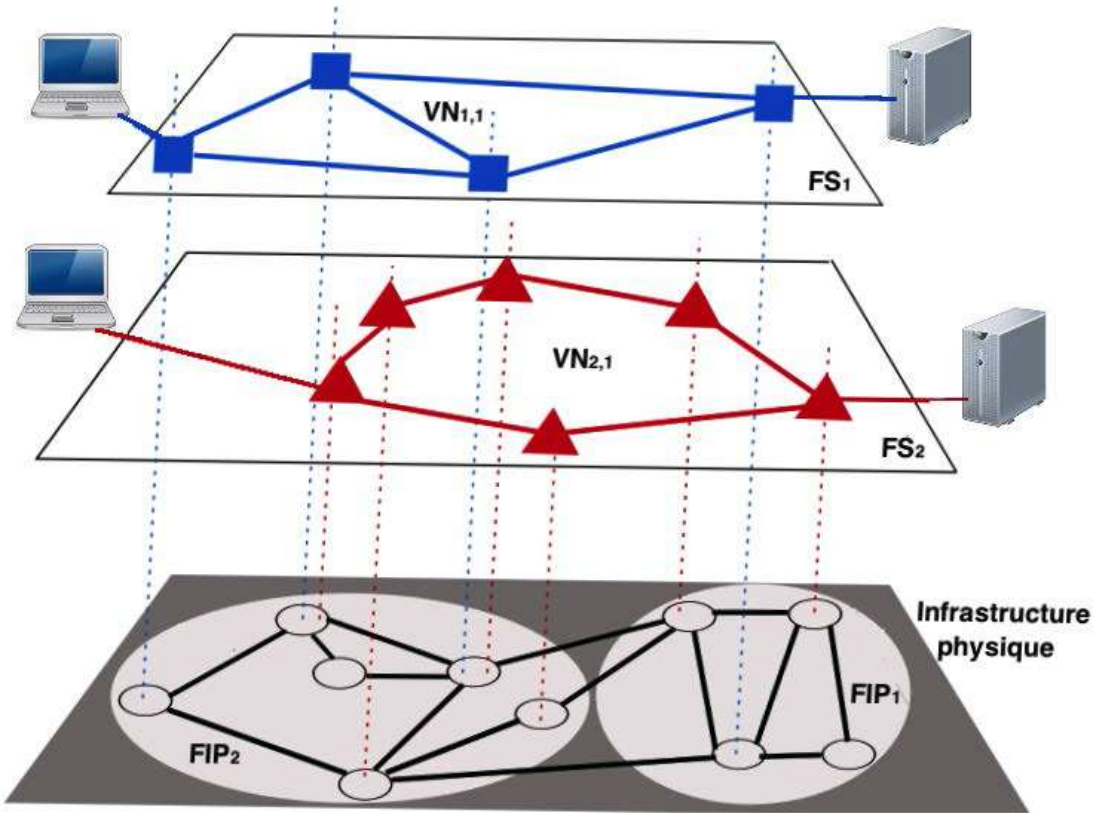


**Figure 2.1: Schéma d'un environnement réseau virtualisé**

Les réseaux définis par logiciels (Software-Defined Networking, (SDN)) sont une nouvelle technologie qui permet d'introduire plus d'intelligence dans les réseaux. Elle offre aux FSs la possibilité de mieux contrôler les différentes composantes de leur réseau.

Cette technologie permet d'augmenter l'efficacité du réseau physique pour que leur infrastructure logique soit plus fonctionnelle.
L'utilisation de SDN permet un contrôle centralisé sur la programmation des différentes composantes du réseau. Dans une architecture SDN, le plan de contrôle et le plan de données d'un noeud physique sont découplés. En conséquence, cette flexibilité offre aux fournisseurs de services la possibilité de programmer et d'automatiser le contrôle du réseau. Ils peuvent ainsi répondre rapidement aux besoins changeants en QoS des services déployés sur un VN.

Dans ce travail de thèse, nous nous intéressons aux réseaux de commutation de paquets et à la QoS dans ce type de réseaux. Nous défendons l'idée que la virtualisation des réseaux et le SDN constituent des solutions efficaces gérant aux mieux la QoS dans les réseaux. En effet, avec ces deux technologies, les FSs peuvent offrir de nouveaux services aux utilisateurs avec une QoS meilleure que le "best effort" offert par l'Internet d'aujourd'hui.

## 2.2 : Problématique et contributions de la virtualisation :

Le réseau IP actuel présente plusieurs limitations liées au fait que les routeurs déployés sont plus ou moins programmables. Ses fonctionnalités sont parfois limitées en termes de gestion du trafic. De plus, ces entités présentent un comportement relativement statique, étant donné qu'ils n'effectuent que seulement les tâches pour lesquelles ils sont conçus. Par ailleurs, ces tâches qui diffèrent d'un constructeur à un autre engendrent des problèmes d'interopérabilité. On peut déduire à ce niveau, qu'avec l'évolution des réseaux à haut débit et la prolifération des terminaux fixes et mobiles, l'architecture actuelle ne peut plus répondre de façon efficace aux besoins des utilisateurs

Deux problèmes majeurs, concernant le réseau Internet contemporain, sont mis en exergue. Le premier problème concerne la mise en place et le déploiement de nouveaux services réseaux.
Ce type de déploiement s'avère très difficile et coûteux, et ne peut pas garantir la qualité de service requise. Cependant, avec l'apparition de nouveaux services que l'on peut déployer sur Internet, comme les services multimédia, les besoins en termes de QoS deviennent prépondérants et décisifs. En plus, Internet est constitué de nombreux systèmes autonomes (ASs) gérés par différentes organisations qui collaborent ensemble pour garantir la QoS de bout-en-bout. Ceci n'est pas une tâche facile parce que les méthodes de collaboration et la QoS à travers plusieurs Ass sont actuellement mal définies. À titre d'exemple, un fournisseur de vidéos à la demande (VoD) comme Canal+ doit s'assurer que les ASs par lesquels son trafic transite arrivent à subvenir à ses besoins en QoS.

Dans certaines situations, une demande importante en débit peut ne pas être satisfaite par ces ASs. Ceci engendre une détérioration du service au niveau de l'utilisateur final. D'une part, le fournisseur de VoD n'a pas le contrôle sur l'infrastructure pour pallier à ces problèmes. D'autre part, les mécanismes actuels de gestion de la QoS utilisés dans les ASs, comme DiffServ, traitent la QoS par classe de service et non par flux individuel de chaque utilisateur final. Ces déficiences liées à la QoS restent de nos jours une problématique ouverte. Le besoin d'offrir une QoS de bout en bout permettant aux fournisseurs de services de garantir une meilleure qualité d'expérience (QoE) pour les utilisateurs finaux se fait ressentir. Le travail présenté dans le cadre de cette thèse tente de trouver des solutions à ces problèmes.
Le second problème est celui de l'équilibrage du coût et de la tarification entre les fournisseurs de services et les opérateurs réseaux. Cette situation entraine plusieurs conflits mutuels. Les différends qui opposent Free à Youtube en France et Verizon à Netflix aux Etats-Unis sont une illustration de ce conflit. Pour mieux expliquer ceci, revenons en détail sur l'exemple du conflit Free-Youtube. Le problème provient du fait que Free est appelé à fournir le service Youtube à ses utilisateurs finaux, cependant, Youtube exige de ces derniers d'utiliser son service de streaming vidéo. Pourtant, dans ce scénario Free paye toujours Youtube pour raccorder ses utilisateurs. Lorsque Free a décidé d'arrêter le payement du raccordement de ses utilisateurs en exigeant une rémunération de l'excès de trafic qu'il génère, le fournisseur de services vidéo a sanctionné les utilisateurs de Free en provoquant une latence significative lors de l'accès à la plateforme.

Le travail effectué dans le cadre de cette thèse concerne la virtualisation des réseaux comme solution à ce type de problèmes rencontrés souvent dans le réseau IP actuel. Nous nous proposons dans cette thèse d'amener quelques éléments de réponse aux déficiences qui apparaissent, de plus en plus, de façon hétérogène dans les réseaux.
La thématique autour de laquelle s'articule cette étude concerne la virtualisation des réseaux comme solution possible palliant le manque d'efficacité dans les réseaux actuels. Pour cela, on se propose d'introduire de nouveaux mécanismes tenant compte des spécificités des services dans le but d'atteindre le niveau de performance requis. Dans ce travail nous traitons les problems de gestion de la QoS dans les réseaux de paquets. SDN et la virtualisation des réseaux sont des technologies conçues pour répondre à cette problématique.

Comme nous l'avons expliqué précédemment, le fournisseur de services est responsable des protocols créant plusieurs réseaux virtuels (VNs) consolidant les infrastructures physiques de plusieurs FIPs. Les FS offrent des services de bout en bout aux différents utilisateurs. Chaque service s'exécute sur un réseau virtuel où le fournisseur de services définit ses besoins en termes de QoS.

La segmentation logique des noeuds et des liens physiques est une tâche complexe. Cette operation est couramment appelée approvisionnement des réseaux virtuels (Virtual Network Embedding VNE). Elle comprend généralement trois étapes. La première est la découverte des ressources du réseau physique. Chaque FIP surveille la charge de son infrastructure physique et informe les FSs sur l'usage et les performances de chaque lien et noeud. La seconde étape est le mappage du réseau virtuel. Cette étape est effectuée par le FS qui définit le chemin virtuel par lequel transitent tous les paquets d'un réseau virtuel. Il s'agit de l'étape qui choisit les noeuds et les liens physiques qui vont êtres utilisés par le VN. Ce choix depend de la disponibilité des ressources du réseau physique. Cette étape est considérée comme la plus difficile, car il est nécessaire de combiner les deux contraintes de noeud et du lien. La troisième et dernière étape est l'allocation des ressources physiques pour le réseau virtuel. Cette étape est effectuée par le FIP lors de la réception de toutes les demandes des FSs. De nombreux travaux de recherche ont considéré l'allocation statique où le FIP alloue une portion de chaque noeud et de chaque lien pour un VN déployé par un FS.

Le manque d'adaptation des ressources du réseau physique offertes pour chaque VN à la dynamique de son flux engendre une sous-utilisation de ces ressources. À titre d'exemple, pour un fournisseur de vidéos à la demande, le nombre d'utilisateurs finaux qui utilisent leur service augmente le soir et diminue pendant la journée. Donc il est nécessaire de libérer les ressources et les réutiliser lorsqu'il y a un besoin. Nous défendons la thèse de rendre cette allocation dynamique pour offrir une meilleure QoS pour les réseaux virtuels tout en maximisant l'utilisation des ressources. En effet, cette approche permet d'augmenter les bénéfices du FIP et d'offrir un meilleur service sur ses réseaux virtuels déployés au profit du FS.

Ce travail vise à mettre en place plusieurs solutions à la fois théoriques et techniques dans le but de : allouer dynamiquement des ressources pour les réseaux virtuels qui coexistent sur une même infrastructure physique, optimiser les performances de l'infrastructure physique garantissant un niveau de QoS adéquat

pour les réseaux virtuels. Nous avons découvert que la même idée apparaît aussi dans le Framework ACTN pour l'abstraction et le contrôle des réseaux de transport, présenté récemment dans le travail en cours de l'IETF.

Dans ce draft, les auteurs ont montré la faisabilité et la mise en oeuvre d'un cadre qui permet le partage dynamique d'une ou de plusieurs infrastructures réseaux entre les FSs à l'aide de SDN. Ils présentent plusieurs types de contrôleurs qui permettent une granularité fine lors de l'allocation des noeuds et des liens physiques entre plusieurs VNs. Le Framework proposé adopte le même business model que le nôtre, où différents fournisseurs de services louent une partie du réseau physique de plusieurs fournisseurs d'infrastructures, afin de délivrer un ou plusieurs services à des utilisateurs finaux.

Nous partons d'un exemple simple où plusieurs fournisseurs de services déploient leurs serveurs dans un centre de données (data center) directement connecté à plusieurs opérateurs réseaux comme Orange ou Free. Ces derniers sont à leurs tours connectés aux utilisateurs finaux. Ces utilisateurs sont soit des utilisateurs résidentiels fixes ou bien des utilisateurs mobiles qui payent les différents FSs pour un service donné (par exemple la VoD). Ces opérateurs réseaux se concentrent seulement sur la simple tâche de gestion de l'infrastructure physique du réseau. Quant aux FSs, ils déploient plusieurs protocoles afin de créer des réseaux virtuels dédiés à l'acheminement des paquets pour un service donné. La question critique est comment partager les ressources physiques limitées (noeuds et liens) d'un fournisseur d'infrastructure entre plusieurs fournisseurs de services qui ont des contraintes de qualité de service. Chacun cherche à consommer toutes les ressources disponibles pour délivrer au mieux un service aux utilisateurs finaux.

Ce partage doit tenir compte du comportement dynamique de chaque flux de données tout en étant équitable.

Nos contributions dans le cadre de la thématique de la virtualisation des réseaux peuvent être résumées comme suit :

- une solution implémentée qui démontre la faisabilité technique de SDN et de la virtualization des réseaux dans la gestion de la QoS dans un réseau domestique connecté à Internet haut débit. L'idée est de déléguer les fonctionnalités de la classification des flux et la limitation du trafic à un contrôleur SDN. Ce contrôleur peut être déployé par le FAI au niveau du réseau d'accès (au niveau de la boucle locale) ou bien connecté directement à la passerelle du réseau domestique comme la boxe ADSL.

- une modélisation en deux étapes de l'interaction entre le fournisseur de services et le fournisseur de l'infrastructure physique pour le partage des ressources (noeuds et liens). La première est la négociation des ressources. La seconde étape est l'approvisionnement de la bande passante au niveau du lien et des cycles du processeur et des mémoires au niveau du noeud. À travers cette modélisation, nous pouvons partager équitablement et efficacement l'infrastructure physique entre plusieurs VNs. Cette solution peut être implémentée comme un système d'aide à la décision pour un FS et un système de facturation pour le FIP lors du partage des ressources entre plusieurs FSs.

- une approche prédictive qui permet de trouver le besoin en termes de bande passante pour chaque VN lors du partage du lien physique. Cette dernière dépend de l'estimation des performances de chaque VN et des allocations actuelles et passées de la bande passante.

L'objectif est d'offrir un contrôle adaptatif de l'allocation de bande passante tout en maintenant la QoS offerte à chaque VN. Cette solution peut être considérée comme un système de monitoring et de contrôle des performances de chaque VN.

## 2.3 : La virtualisation des réseaux :

### 2.3-1/a : Les réseaux logiciels :

La réalisation de réseaux logiciels (virtuels) se fait à l'aide des machines virtuelles. Pour cela il faut les interconnecter comme on l'aurait fait pour des machines physiques. il faut ainsi partager les circuits de communications entre les multiples réseaux logiciels. La **Figure 2.2** représente un ensemble de réseaux logiciels bâtit sur un seul et unique réseau physique.

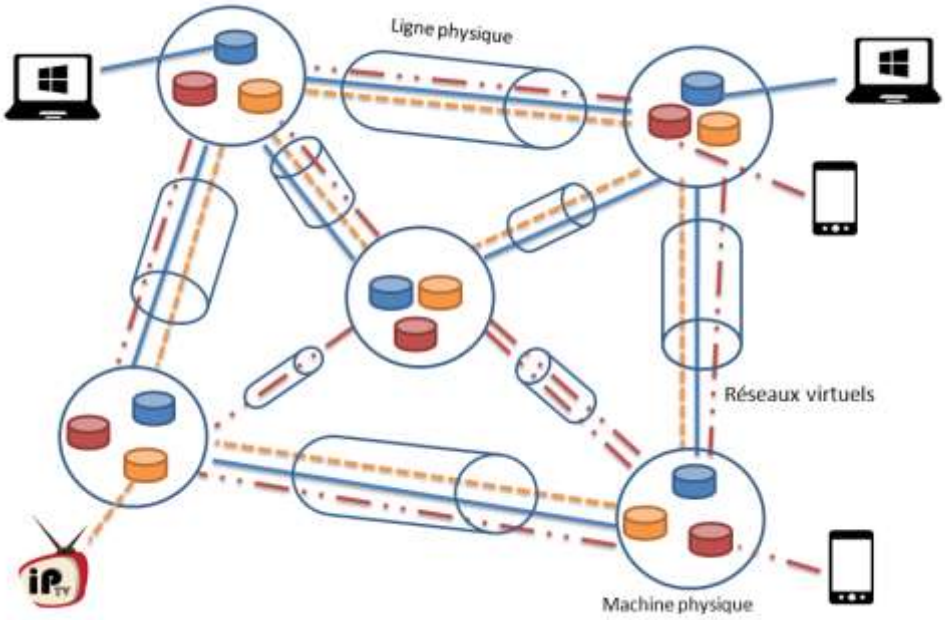


**Figure 2.2: Réseaux logiciels sur un seul réseau physique**

Chaque réseau virtuel peut avoir ses propres caractéristiques et sa propre architecture. Un réseau logiciel peut être destiné à un service ou une application particulière, par exemple un réseau logiciel dévolu à un service de VoIP (Voice over IP) réseau en rouge sur la figure ci-dessus, un autre à de la IP-TV, réseau en orange sur la figure, et un troisième réseau pour le transport de données, réseau en bleu. De manière générale on peut faire un réseau logiciel pour chaque utilisateur. Le réseau logiciel est créé au moment ou l'utilisateur se connecte et il est détruit lorsqu'il se retire. Mais aujourd'hui on se limite à un nombre de réseaux logiciels compatibles avec la capacité du matérielle de l'infrastructure physique sous-jacente. Chaque

réseau virtuel se voit allouer des ressources en fonctions des demandes et des droits des utilisateurs.

Les noeuds virtuels qui sont à la base de la constitution des réseaux logiciels, se trouvent dans des datacenters qui peuvent être plus ou moins importants : cela va des gros datacenters centraux aux petits datacenters (femto-datacenters) en passant par d'autres tailles intermédiaires.

Un des avantages de ces réseaux est de pouvoir faire migrer des machines virtuelles d'un équipement physique à un autre et d'un datacenter à un autre, car on est désormais sur du logiciel. La migration peut être automatisée si par exemple un noeud de transfert est surchargé, montre des signes de faiblesse ou bien tombe en panne. En réalité lorsqu'on réalise la migration d'un noeud, on ne procède pas au transport de tout le code de la machine, ce qui serait un peu lourd dans certains cas. En général, le logiciel à transporter se trouve déjà présent dans le noeud distant mais il est dans un état de veille. Il suffit de mettre

en route le logiciel et de lui envoyer la configuration du noeud à déplacer. Cela demande que peu de données à transférer et une latence faible avant l'ouverture de la machine migrée.

L'isolation est une propriété importante que doivent posséder ces réseaux, il ne faut pas qu'un problème sur un réseau logiciel puisse entacher les autres réseaux. L'isolation est complexe, car il faut à la fois partager les ressources communes et être sûr que chaque réseau ait à tout moment l'accès à ses propres ressources.

### 2.3-1/b : Equipements virtuels :

Pratiquement tous les équipements peuvent être virtualisés à l'exception de ceux qui doivent prendre en charge la réception de signaux (électromagnétiques, pression atmosphérique, température …). Par exemple une antenne ou un thermomètre ne peuvent pas être remplacés par un logiciel. Mais toute la partie qui concerne le traitement de signal peut être gérée dans une machine virtuelle.
Tous les équipements des réseaux sont virtualisés ou en cours de virtualisation, que ce soit la partie traitement des Node-B des réseaux mobile 3, 4 et dans pas longtemps 5G, les bases de données HLR, VLR, HSS, les routeurs, commutateurs, firewalls, load balancer etc. De plus, ces machines peuvent être partitionnées pour s'exécuter sur plusieurs machines en parallèle.
C'est ici qu'on voit toute l'importance des datacenters qui forment le Cloud puisque ce sont des environnements où la puissance de traitement est disponible avec un vaste espace de stockage pour les machines virtuelles et tout un ensemble d'informations liées aux réseaux, aux clients et aux algorithmes de traitement.

### 2.3-1/c : Les réseaux Overlays :

La conception des réseaux overlay est fondée sur la création d'une topologie virtuelle sur une topologie physique existante. Son objectif est de fournir des services personnalisés à l'utilisateur. il s'agit d'un réseau informatique construit au-dessus d'un second réseau dont le but est d'optimiser la distribution des données. Les

optimisations réalisées concernent le délai, la bande passante et l'accroissement de la disponibilité des services sur ce réseau. L'internet a commencé comme un réseau overlay superposé sur un réseau de télécommunications.

Cette implémentation se fait, généralement, au niveau de la couche application du modèle OSI bien qu'actuellement elle s'effectue au niveau des couches inférieures .

les réseaux overlays ne nécessitent pas une modification dans l'infrastructure physique d'un réseau. Ils sont par conséquent très utilisés comme moyens simples et peu coûteux pour déployer de nouveaux services réseaux. il est important de noter que ces réseaux nécessitent beaucoup d'entretien pour permettre le déploiement et la gestion des réseaux sous-jacents. La plupart des réseaux overlays ont été implémentés au-dessus de la couche IP, cette solution ne peut pas aller au-delà des limites d'internet. Bien que les notions des réseaux Overlays et de la virtualisation des réseaux soient liées, ils ne s'agissent pas des mêmes réseaux. Les réseaux overlays sont proposés comme solution aux problèmes liés à Internet tels que la gestion de la QoS et le multicast mais la virtualisation des réseaux se présente comme une nouvelle architecture d'internet palliant tous les problèmes existants.

## 2.3-2 : Les techniques de la virtualisation des réseaux :

La virtualisation est une technique utilisée pour modifier les propriétés d'un service de réseau sans introduire de modifications au niveau des clients et des serveurs. Elle permet la coexistence de multiples réseaux hétérogènes dans une seule infrastructure. Pour cela, cette technologie doit assurer un niveau adéquat d'isolation afin de permettre l'utilisation des ressources physiques du réseau en temps réel et à grande échelle.

Au cours de ces dernières années, plusieurs techniques ont été utilisées pour créer des réseaux virtuels comme les VLANs (réseaux locaux virtuels) et les VPNs (réseaux privés virtuels). Récemment, les approches de virtualisation des serveurs ont été utilisées pour créer des routeurs et des liens virtuels sur des équipements physiques et des canaux de communication. Dans ce qui suit nous présentons une brève discussion des approches de virtualisation des réseaux.

## 2.3-2/a : Techniques basées sur la virtualisation des protocoles :

Les approches basées sur le protocole mettent en oeuvre un protocole permettant l'identification et l'isolation des réseaux virtuels [Chowdhury and Boutaba, 2009]. Ce type d'approche exige que l'équipement physique soit capable de supporter le protocole choisi.

Un exemple de la virtualisation de réseau à base de protocole est les réseaux locaux virtuels (VLAN). L'idée est de diviser logiquement un réseau local en plusieurs réseaux virtuels. Les hôtes dans un même VLAN communiquent entre eux si elles sont sur le même réseau local, quel que soit l'emplacement physique.

La création de réseaux privés virtuels est une autre approche couramment utilisée. Les VPN sont généralement utilisés pour fournir un canal de communication sécurisé

entre plusieurs sites géographiquement distants. Des protocoles pour assurer la confidentialité des données et l'authentification de l'utilisateur sont utilisés dans ce genre de réseau virtuel.

Dans ce qui suit nous allons présenter en détail les techniques utilisées pour la virtualisation basées sur les protocoles.

## 2.3-2/a-1 : Les réseaux locaux virtuels (VLANs)

Un VLAN permet le regroupement de plusieurs hôtes, de façon logique et non physique indépendamment de leur connectivité physique. il permet de créer des domaines de diffusion gérés logiquement sans se soucier de l'emplacement de ses hôtes.

Plusieurs VLANs peuvent coexister sur un même commutateur réseau et ils peuvent être locaux à un commutateur ou s'étendre sur un ensemble de commutateurs reliés entre eux. L'objectif est de contourner les limitations de l'architecture physique.

Ceci conduit à l'amélioration de la gestion du réseau et de l'optimisation de la bande passante tout en séparant les flux de chaque ensemble d'hôtes. Chaque trame possède un identifiant du VLAN dans l'en-tête de contrôle d'accès au support (Media Access Control, (MAC)). Les réseaux locaux virtuels fonctionnent au niveau des couches liaison de données et réseau du modèle OSI. Ils sont définis par le standard IEEE 802.1Q.

La **Figure 2.3** illustre le regroupement de plusieurs PCs en VLANs indépendamment de leur connectivité physique.

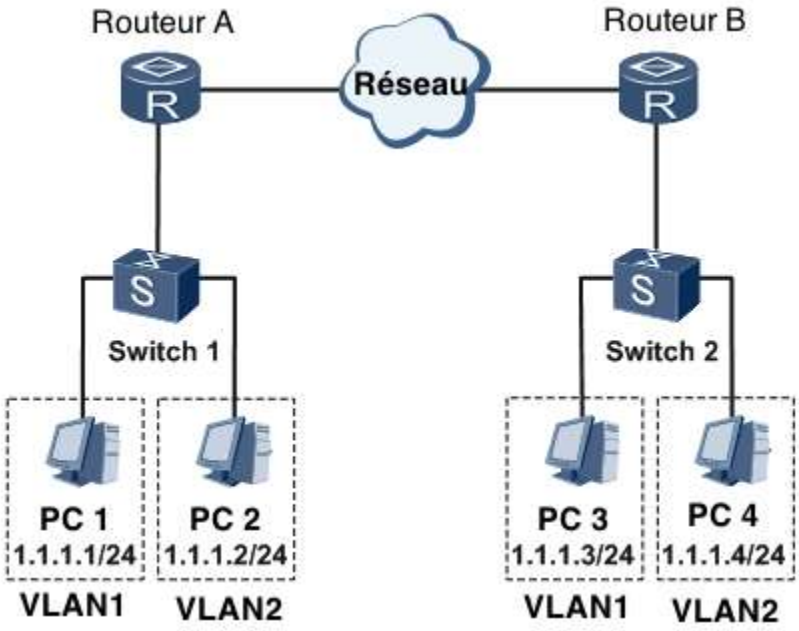


**Figure 2.3: Regroupement des PCs en VLANs**

On distingue trois types de VLAN:

- **VLAN de niveau 1 ou VLAN par port** (*Port-Based VLAN*)

Il définit un réseau virtuel en fonction des ports de raccordement au commutateur. On associe un port physique de ce commutateur à un numéro de VLAN.

- **VLAN de niveau 2 ou VLAN MAC** (*MAC Address-Based VLAN*)

Il définit un réseau virtuel en fonction des adresses MAC des hôtes. Le déploiement de ce type de VLAN est plus souple que le VLAN de niveau 1 puisque la machine, peu importe le port sur lequel elle sera connectée, elle fera toujours partie du VLAN dans lequel son adresse MAC est configurée.

- **VLAN de niveau 3**

On distingue deux types de VLAN de niveau 3:

- **VLAN par sous-réseau** (*Network Address-Based VLAN*)

Il a le même principe que pour les VLAN de niveau 2 mais en indiquant les adresses IP source des datagrammes et en associant des sous-réseaux à différents VLANs. Ceci permet de modifier automatiquement la configuration des commutateurs lors d'un changement ou d'un déplacement de la hôte.

- **VLAN par protocole** (*Protocol-Based VLAN*)

Il définit un réseau virtuel en fonction des protocoles. Par exemple, il définit un VLAN pour TCP/IP. Dans ce cas, tous les hôtes, qui utilisent ce protocole dans un même VLAN, sont regroupées.

## 2.3-2/a-2 : Les réseaux privés virtuels (VPN) :

Un réseau privé virtuel (*Virtual Private Network* (VPN)), est une technologie utilisée pour connecter en toute sécurité deux réseaux physiques géographiquement distants. Elle permet de créer une liaison virtuelle, sur un réseau public comme Internet, entre des ordinateurs distants.
Cette technologie repose sur un protocole de tunneling qui permet de faire circuler les informations de façon cryptée d'un bout à l'autre du tunnel. Son objectif est de masquer la distance entre deux ou plusieurs sites. Les VPNs ne sont pas conçus pour créer plusieurs réseaux virtuels sur une même infrastructure physique.

La **Figure 2.4** montre un schéma du déploiement d'un VPN entre plusieurs sites distants d'une organisation.

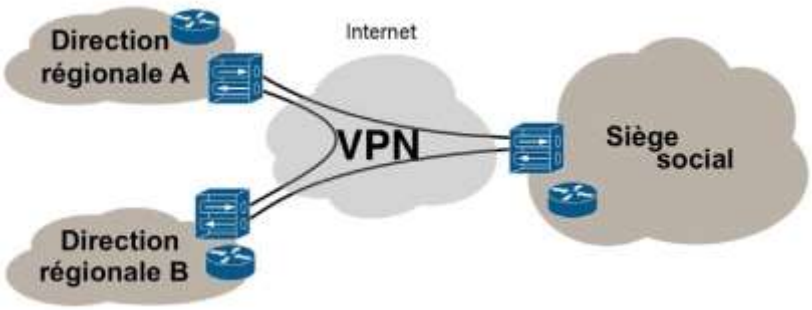


**Figure 2.4: VPN entre plusieurs sites distants d'une entreprise donnée**

On distingue trois types de VPN :

- **VPN de couche 1 (L1VPN)**

Il permet d'offrir un service par la couche physique du modèle OSI. Il fournit la connectivité entre deux ou plusieurs sites clients. Il offre aux clients un certain contrôle sur la création et le type de la connectivité du réseau virtuel.

- **VPN de couche 2 (L2VPN)**

Il permet le transport des trames de la liaison de données des différents sites participants. Il s'agit d'une solution simple pour des connexions point à point via des tunnels pour permettre le transport de paquets de n'importe quel protocole de la couche réseau.

- **VPN de couche 3 (L3VPN)**

Il permet de connecter plusieurs sites géographiquement séparés à travers le réseau commute par paquets. Il permet de garantir à chaque trafic utilisateur d'être séparé des autres trafics au niveau du réseau coeur. Ceci est réalisé en déployant des équipements client de bord (*Customer Edge-CE*) qui sont à leur tour connectés aux équipements fournisseur de bord (*Provider Edge-PE*) dans le réseau coeur. L'objectif est de faire transiter le trafic entre les PE en utilisant ses routeurs internes. Le L3VPN garantit, pour chaque utilisateur, un trafic privé et séparé des autres utilisateurs connectés au réseau coeur.

## 2.3-2/b : Techniques basées sur la virtualisation des machines :

Les approches basées sur la virtualisation des machines consistent à isoler les ressources informatiques de façon à pouvoir executer plusieurs instances de réseaux virtuels par un moyen de groupes de machines virtuelles interconnectées (VMs).

Ces machines sont utilisées pour créer des routeurs virtuels et des liens virtuels pour servir de liaison. Cette technique est relativement souple, car elle permet l'utilisation d'un logiciel personnalisé, pas cher, pour la création d'un commutateur ou d'un routeur virtuel.

Dans ce qui suit nous allons présenter en détail les techniques basées sur la virtualisation de machines qui sont le cloisonnement, la virtualisation complète et la para-virtualisation.

## 2.3-2/b-1 : Le cloisonnement :

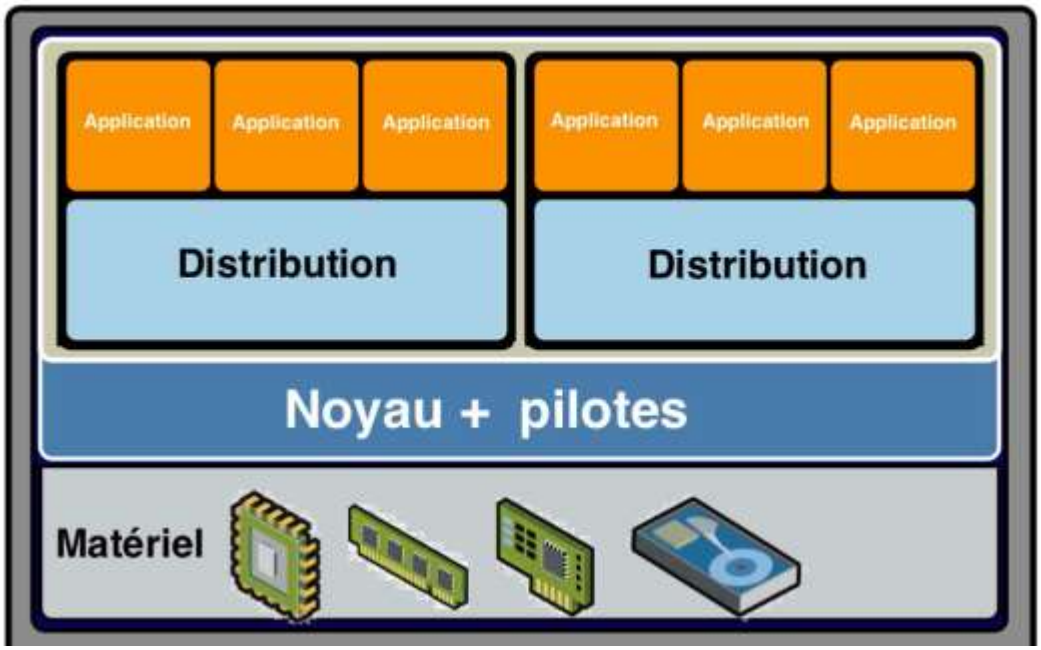


**Figure 2.5: La technique du cloisonnement**

Le cloisonnement est la technique de virtualisation la plus légère qu'on peut mettre en oeuvre au sein d'un même système d'exploitation. Pour cela on le divise en plusieurs espaces ou environnements.

Chaque environnement est géré par le système d'exploitation hôte comme un processus isolé dans un conteneur partageant le même noyau. Le conteneur apporte une virtualisation de l'environnement d'exécution. il permet aux programmes de chaque context de communiquer seulement avec les processus et les ressources qui leur sont associés.

L'espace noyau n'est pas différencié, il est unique et partagé entre les différents contextes. Il fournit la virtualisation, l'isolement et la gestion des ressources. Ce partage du noyau limite cette technique aux environnements de mêmes types.

## 2.3-2/b-2: La virtualisation complète :

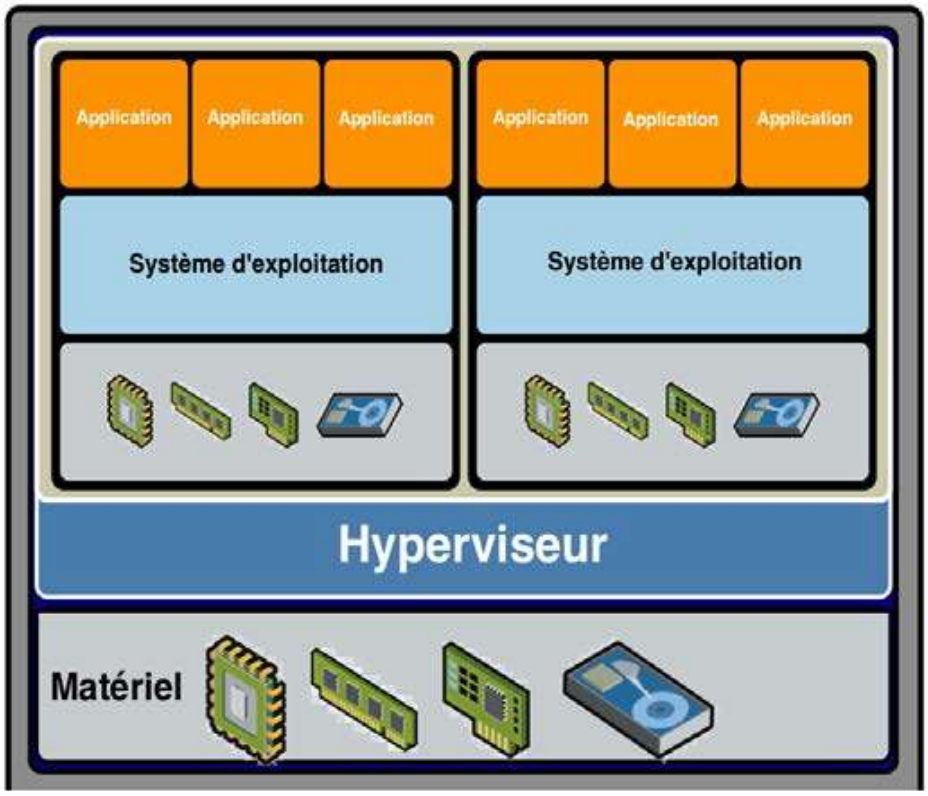


**Figure 2.6: Schématisation de la virtualisation complète**

La virtualisation complète (full virtualization) est une technique de virtualisation qui offer une réplique virtuelle du matériel du système de sorte que les systèmes d'exploitation et logiciels peuvent fonctionner sur le matériel virtuel exactement comme sur le matériel d'origine.

L'hyperviseur constitue la couche d'abstraction entre les systèmes d'exploitation invités et le matériel. Son rôle est de créer un environnement virtuel complet simulant globalement un nouvel ordinateur. Au moment de l'exécution, les instructions du système d'exploitation invité ne donnent accès qu'au matériel virtuel présenté par l'hyperviseur. La virtualisation complète offer une meilleure isolation et plus de sécurité pour les machines virtuelles en simplifiant la migration et la portabilité.

La **Figure 2.6** illustre la création de deux machines virtuelles sur une seule machine physique tout en isolant les ressources. Les programmes qui s'exécutent sur l'espace utilisateur d'un système d'exploitation invité n'ont pas un accès direct au matériel, mais uniquement à la couche d'abstraction.
La machine virtuelle émule le matériel pour faciliter l'accès du système d'exploitation aux ressources physiques.

## 2.3-2/c : La para-virtualisation :

La para-virtualisation est très proche du concept de la virtualisation complète. Les deux types de virtualisation reposent sur un hyperviseur qui gère l'interfaçage avec les ressources matérielles. Excepté le fait que la para-virtualisation a des

fonctionnalités différentes dans chaque technique de virtualisation, elle permet une coopération entre l'hyperviseur et le système d'exploitation invité. En effet, lors de l'exécution du système d'exploitation invité, l'hyperviseur capture les appels système de l'invité et les transmet au matériel. L'hyperviseur gère l'interface qui va permettre à plusieurs systèmes d'exploitation invités d'accéder de manière concurrente aux ressources.

Le système d'exploitation invité est conscient de l'exécution sur une machine virtuelle (VM). Cette opération nécessite certaines modifications logicielles non seulement au niveau du système d'exploitation hôte mais également au niveau du système d'exploitation invité. Ce dernier doit être muni des pilotes permettant d'adresser des commandes au matériel.

La **Figure 2.7** schématise la technique de para-virtualisation.

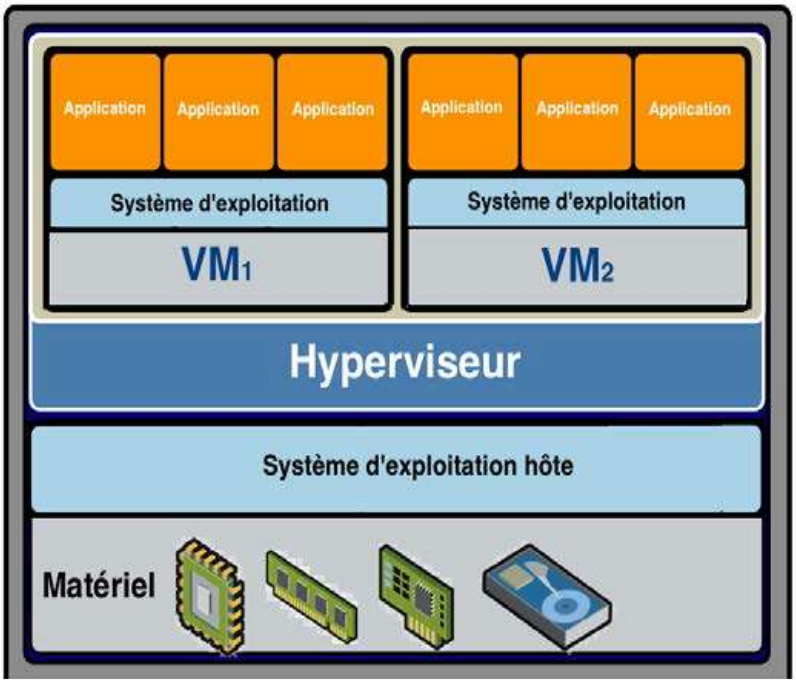


**Figure 2.7: La technique de la para-virtualisation**

La virtualisation dépasse ces limites en permettant d'exécuter simultanément plusieurs systèmes d'exploitation et plusieurs applications sur le même ordinateur, ce qui accroît l'utilisation et la flexibilité du matériel.

Ce concept couvre différents aspects, on peut ainsi virtualiser les serveurs, le stockage, les applications (simplification de l'administration) ou encore le poste client.

A ce stade, la virtualisation et l'informatique en nuage, ces deux sujets vont de pair, les avantages de la virtualisation prennent tout leur sens à l'échelle de l'informatique en nuage.

## 2. 4: Le cloud computing (L'informatique en nuage):

### 2.4-1: Déffinition :

L'informatique en nuage ou Cloud Computing est un concept où les ressources informatiques sont virtualisées et dynamiquement élastiques (provisionnement et déprovisionnement automatique).
Ces ressources sont fournies comme un service à travers Internet, de manière transparente pour les utilisateurs.

### 2.4-2: Modèles de services :

Les services cloud sont disponibles sous diverses options, selon les besoins des clients. Dans la publication spéciale 800-145, le NIST (National Institute of Standards and Technology) définit les trois principaux services cloud suivants [Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)].

Ces trois modèles de service doivent être déployés sur des infrastructures qui possèdent les cinq caractéristiques essentielles citées plus bas pour être considérées comme du Cloud Computing.

### 2.4-2/a: Software as a Service (SaaS) / Logiciel en tant que Service :

Les applications sont mises à la disposition des utilisateurs via le web. Ce modèle de service est caractérisé par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud.

L'utilisateur accède à l'application par le réseau au travers de divers types de terminaux (souvent via un navigateur web). L'administrateur de l'application ne gère pas et ne contrôle pas l'infrastructure sous-jacente (réseaux, serveurs, applications, stockage). Il ne contrôle pas les fonctions de l'application à l'exception d'un paramétrage de quelques fonctions utilisateurs limitées.

### 2.4-2/b: Platform as a Service (PaaS) / Plate-forme en tant que Service :

Représente les outils et les services utilisés pour fournir les applications.
L'utilisateur a la possibilité de créer et de déployer sur une infrastructure Cloud PaaS ses propres applications en utilisant les langages et les outils du fournisseur. L'utilisateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous-jacente (réseaux, serveurs, stockage) mais l'utilisateur contrôle l'application déployée et sa configuration.

### 2.4-2/c: Infrastructure as a Service (IaaS) / Infrastructure en tant que Service :

Représente le matériel et les logiciels utilisés pour les serveurs, le stockage, les réseaux et les systèmes d'exploitation.

L’utilisateur loue des moyens de calcul et de stockage, des capacités réseau et d’autres ressources indispensables (partage de charge, pare-feu, cache).

L’utilisateur a la possibilité de déployer n’importe quel type de logiciel incluant les systèmes d’exploitation.
L’utilisateur ne gère pas ou ne contrôle pas l’infrastructure Cloud sous-jacente mais il a le contrôle sur les systèmes d’exploitation, le stockage et les applications. Il peut aussi choisir les caractéristiques principales des équipements réseau comme le partage de charge, les pare-feu, etc.

## 2.4-3 : Caractéristiques du cloud :

Le cloud présente les caractéristiques suivantes:

- **Self-service à la demande (Accès aux services par l’utilisateur à la demande):**

un utilisateur peut allouer des ressources sans interaction humaine avec le fournisseur.

- **Accès réseau élargi (Accès réseau large bande):**

les ressources sont accessibles via le réseau par des systèmes hétérogènes (clients légers, clients lourds, etc.).

- **Partage de ressources (Réservoir de ressources (non localisées)):**

les ressources sont dynamiquement affectées, libérées puis réaffectées à différents utilisateurs. L'utilisateur n'a pas besoin de connaître la localisation (pays, région, centre de donnée) des ressources.

- **Élasticité (Redimensionnement rapide):**

La mise en ligne d’une nouvelle instance d’un serveur est réalisée en quelques minutes, l’arrêt et le redémarrage en quelques secondes. Toutes ces opérations peuvent s’effectuer automatiquement par des scripts. Ces mécanismes de gestion permettent de bénéficier pleinement de la facturation à l’usage en adaptant la puissance de calcul au trafic instantané.

- **Facturation à l’usage:**

Il n’y a généralement pas de coût de mise en service (c’est l’utilisateur qui réalise les opérations).

La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Une unité de traitement stoppée n’est pas facturée.

# CHAPITRE (03)

## Les réseaux programmables

### 3.1 : Les réseaux définis par logiciels (SDN)

Les réseaux définis par logiciels (SDN) et la virtualisation des fonctions réseau Network Function Virtualization (NFV) sont de nouvelles façons de concevoir, construire et exploiter les réseaux. Dans ce travail nous n'allons pas nous intéresser pas à la technologie de virtualization des fonctions réseau qui sont exécutées sur des serveurs standards. Nous considérons que toutes les fonctions du réseau telles que le routage ou le traitement des paquets sont effectuées par des routeurs programmables virtualisés.

Les routeurs sont des équipements du réseau qui permettent de transférer les paquets d'un réseau à un autre. Sur la base de leur fonctionnement, on peut diviser un routeur en deux parties ou plans : le plan de contrôle et le plan de données.

Le plan de données est responsible du transport des données de la source à la destination. Il récupère les paquets de données au niveau de l'interface d'entrée afin d'exécuter des fonctions de commutation.

Les tables de routage sont alors consultées pour déterminer l'interface de sortie des paquets. Tous les paquets qui ont la même destination suivent le même chemin. Le plan de contrôle est en charge de la construction et du maintien des tables de routage. Cette configuration est effectuée soit manuellement par les administrateurs du réseau soit à l'aide des informations distribuées collectées par des protocoles de routage comme BGP ou OSPF. Actuellement, dans un routeur ou un commutateur classique les tables de routage sont programmées localement.

Les noeuds du réseau choisissent librement la meilleure façon de traiter un flux. Le plan de données et le plan de contrôle sont co-localisés sur le même équipement. Les réseaux définis par logiciels (Software- Defined Network (SDN)) ont introduit une séparation entre le plan de données et le plan de contrôle pour rendre le réseau programmable. Le plan de contrôle est place dans un contrôleur centralisé qui a une vision sur la topologie du réseau. Le plan de données réside encore sur le commutateur ou le routeur.

L'objectif de SDN est d'offrir une flexibilité et une programmabilité des réseaux afin de render sa gestion simple. La **Figure 3.1** illustre les différentes couches des réseaux SDN.

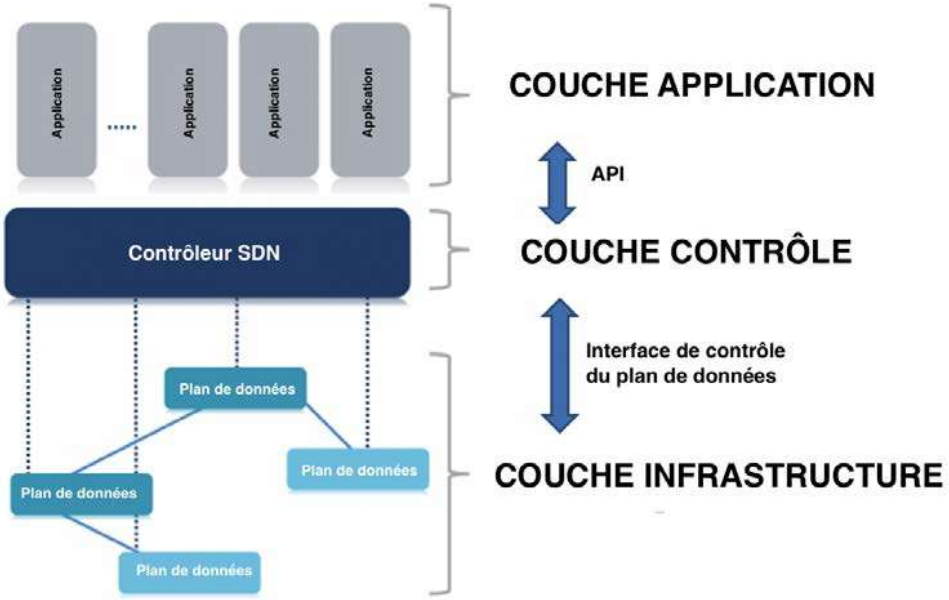


**Figure 3.1: Les différentes couches des réseaux définis par logiciels (SDN)**

Les réseaux SDN sont le résultat de plusieurs travaux de recherche antérieurs visant à render programmable les réseaux afin qu'ils soient plus flexibles et innovants. Deux groupes de recherche ont proposé de séparer le plan de contrôle logiciel du matériel sous-jacent et d'offrir des interfaces pour la gestion. Le but est de pouvoir déployer rapidement des services personnalisés et offrir une configuration dynamique des réseaux au moment de l'exécution.

Ces groupes sont celui du travail OpenSig et celui de l'initiative Active Networking. OpenSig est apparu en 1995 en proposant une application du concept de programmation dans les réseaux ATM. L'idée principale était la séparation du plan de contrôle des réseaux de données, avec un protocole de signalisation entre les deux plans. L'objectif de cette contribution est la programmation et le contrôle à distance des commutateurs ATM.

L'initiative Active Networking est apparue au milieu des années 90, dont l'idée est de gérer les ressources des noeuds du réseau par une API (Application Programming Interface).

Ceci permet aux opérateurs des réseaux de contrôler aisément et activement les noeuds en exécutant un code souhaité.

## 3.2 : Architecture:

La **Figure 3.4** présente une vue logique de l'architecture typique d'un réseau SDN.
- **La couche « Plan de données »** est composée principalement des équipements d'acheminement (commutateurs, routeurs...).

- **La couche « Plan de contrôle »** est constituée principalement d'un contrôleur SDN qui permet d'héberger la logique de contrôle du réseau. Ce contrôleur met en oeuvre cette logique en accédant au plan des données à travers une interface unifiée appelée **'south-bound'.**

- D'autre part**, la couche « Application »** représente les programmes qui définissent la logique de contrôle du réseau. Ces programmes sont construits moyennant une interface de programmation appelée **'north-bound'** et offerte par le contrôleur. Une application de contrôle pourrait, par exemple, définir la politique de routage ou la façon de gérer la qualité de service (QOS) dans un réseau SDN.

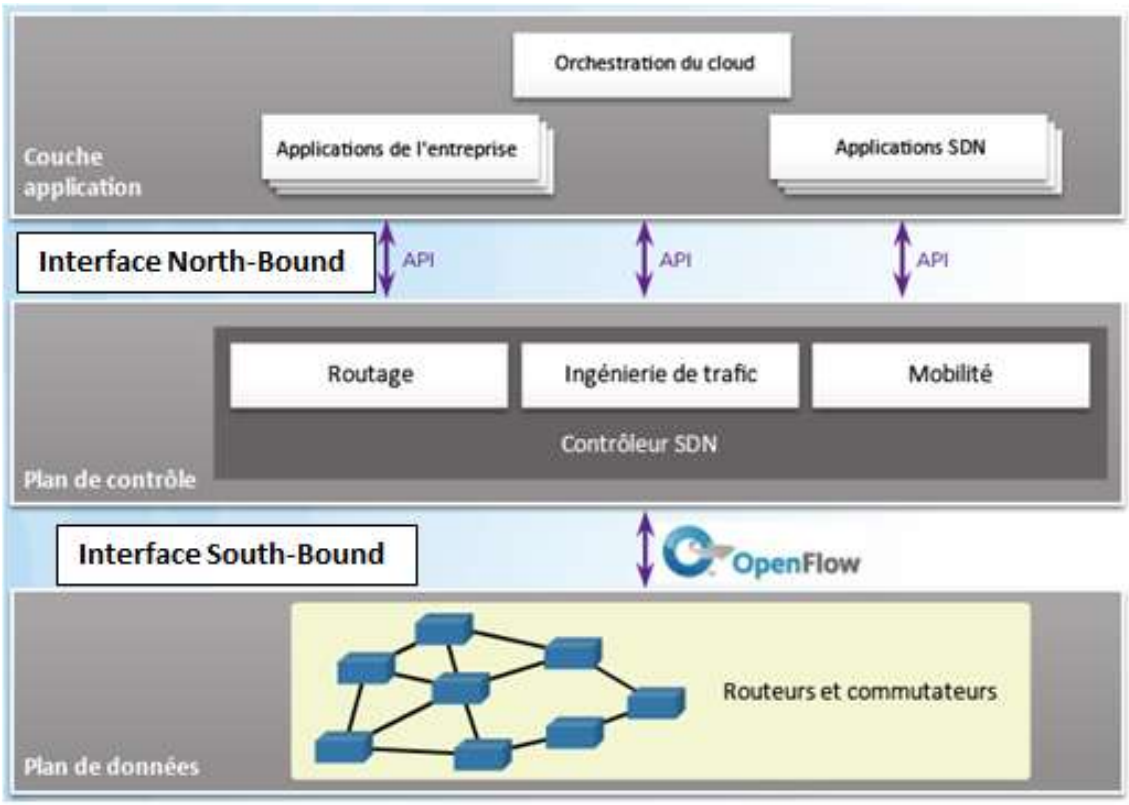


**Figure 3.2: Architecture SDN**

Ci-dessous une figure comparative entre le réseau traditionnel et le SDN:

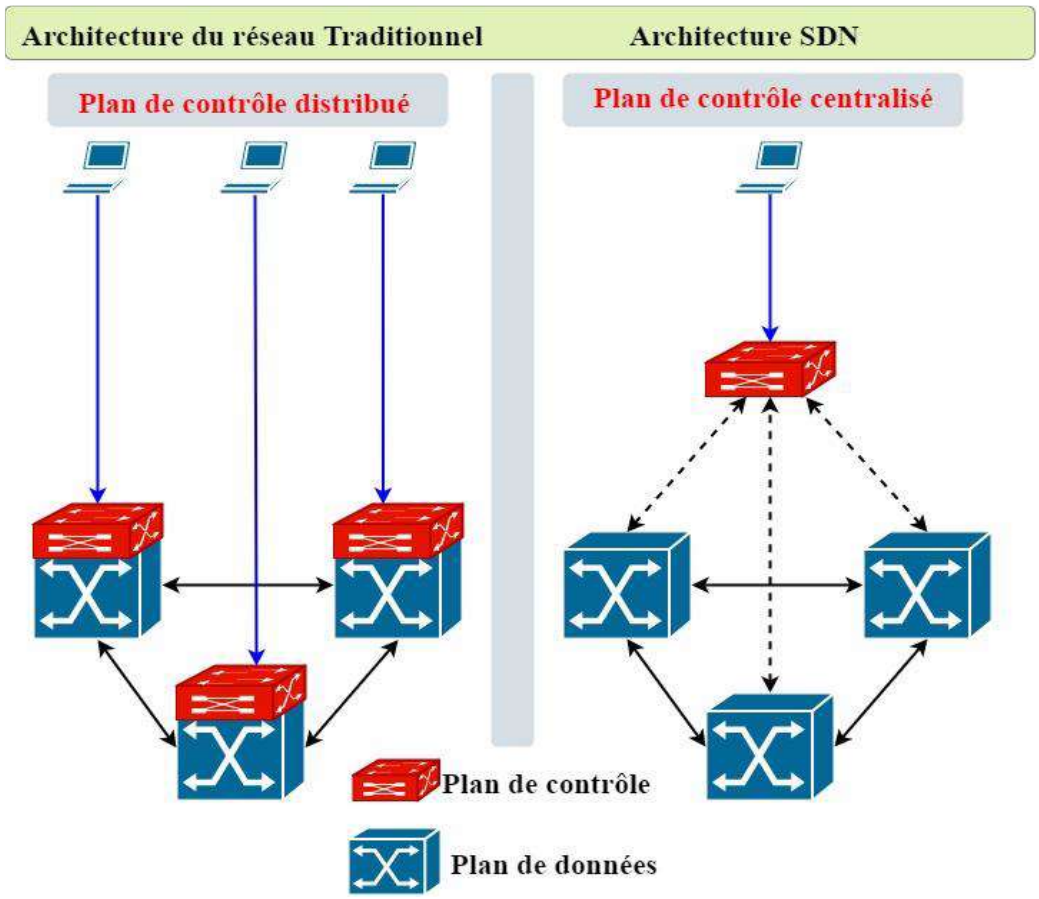


**Figure 3.3: Réseau traditionnel et SDN**

## 3.3 : Le Protocole OpenFlow dans l'architecture SDN:

Il s' agit d'un protocole standard utilisé par le contrôleur pour transmettre au commutateur des instructions qui permettent de programmer leur plan de données et d'obtenir des informations de ces commutateurs afin que le contrôleur puisse disposer d'une vue globale logique (abstraction) du réseau physique.

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

Cette vue est utilisée pour toutes les décisions que doit prendre le plan de contrôle (routage, filtrage de trafic, partage de charge, traduction d'adresse, etc).

## 3.3/a : La genèse d'OpenFlow:

L'histoire d 'OpenFlow est intéressante et permet de mieux comprendre son rôle fondamental dans la conception de l'architecture SDN (Software Defined Network) et la virtualisation des fonctions réseau.

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.

OpenFlow 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, c'est ça qui a était le résultat de l'équipe de recherche à Stanford.

## 3.3/b : Structure d'un commutateur OpenFlow:

Les commutateurs Ethernet et les routeurs les plus modernes contiennent des tables de flux qui sont utilisées pour effectuer des fonctions de transfert indiquées dans les entêtes de paquets.

Les principales fonctions des commutateurs sont les suivantes:

- **Fonction de support du contrôle** (Control support function) : Interagit avec la couche contrôle SDN afin de supporter la programmabilité via les interfaces ressource-contrôle.

Le commutateur communique avec le contrôleur et le contrôleur gère le commutateur avec le protocole OpenFlow. OpenFlow peut être utilisé aussi bien pour du contrôle que pour de la gestion.

- **Fonction d'acheminement des données** (Data forwarding function) : Accepte les flux de données entrants provenant d'autres équipements de réseau et des systèmes de terminaison et les relaie sur un chemin de commutation qui a été calculé et établi à partir des règles définies par les applications SDN, passées au contrôleur et redescendues au commutateur.

À l'aide de la table de flux, une des actions suivantes est exécutée:

α): Relayer le paquet sur un port de sortie,
β): Supprimer le paquet,
λ): Passer le paquet au contrôleur. Le paquet est encapsulé dans un message OpenFlow **PACKET_IN.**

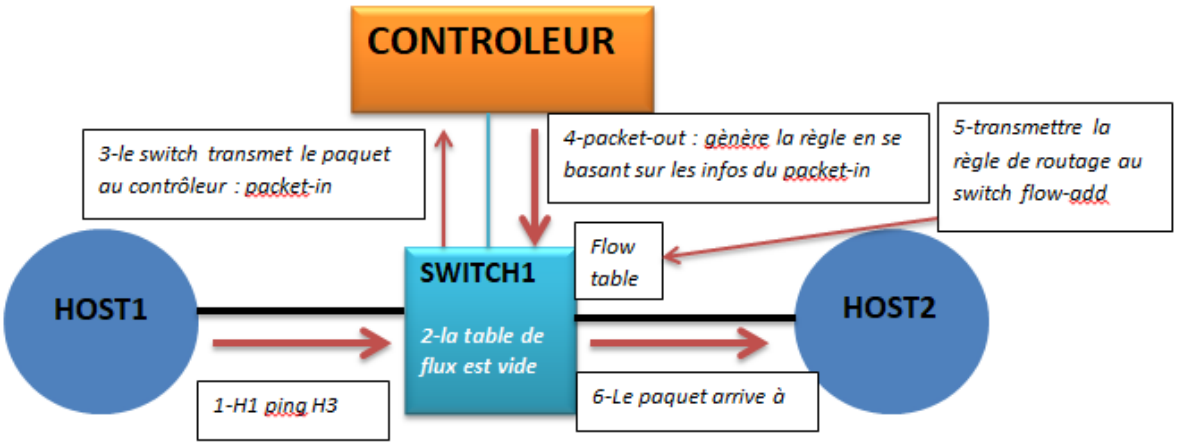


**Figure 3.4: Diagramme de flux des messages OpenFlow**

## 3.3/c : Table de flux:

Chaque table de flux du commutateur contient un ensemble d'entrées de flux qui présentent les règles d'acheminement des paquets. Chacune est structurée comme suit:

| Champ de correspondance | Instructions | Compteurs |
|---|---|---|

**Figure 3.5: Table de flux des messages OpenFlow**

• **Champ de correspondance** (Match fields) : qui définissent le modèle du flux de paquets à travers l'instanciation des champs d'en-tête allant de la couche Ethernet à la couche Transport.

• **Compteurs (Counters) :** des compteurs sur les paquets servent essentiellement à garder des statistiques sur les flux pour ensuite décider si une entrée de flux est active ou non.

• **Instructions** (Actions) : Actions à appliquer aux paquets qui correspondent à l'entrée de flux.

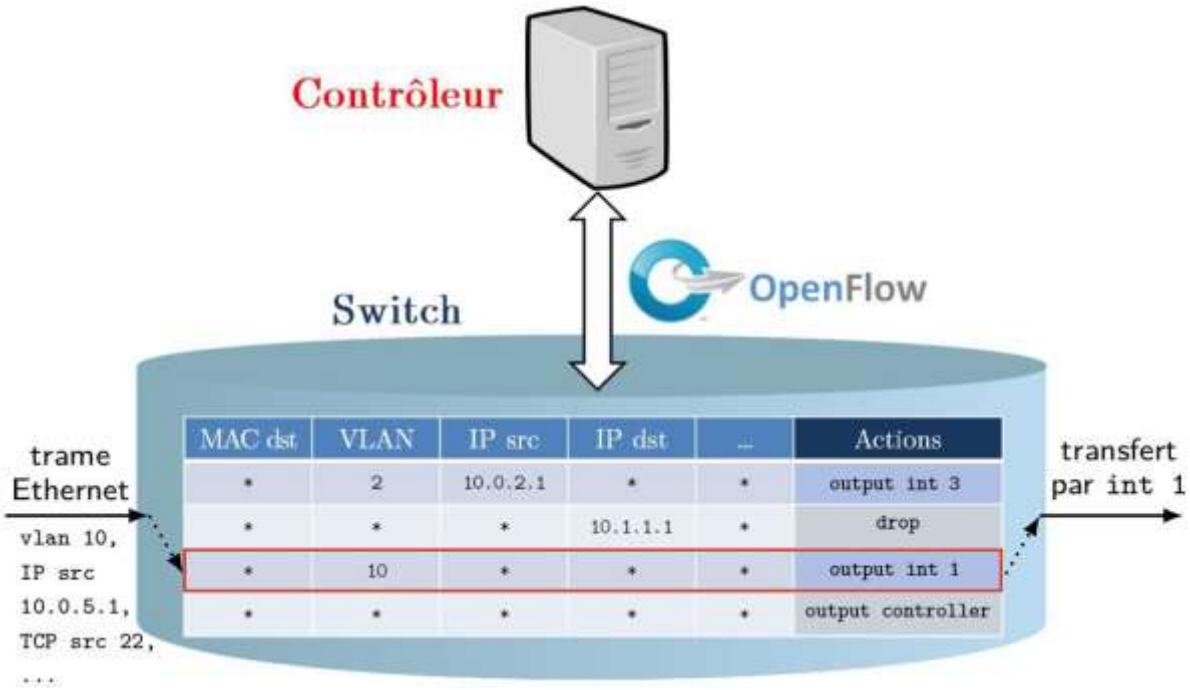


**Figure 3.6: Schématisation d'un modèle de flux OpenFlow**

## 3.3/d : Messages OpenFlow

Les messages OpenFlow entre le contrôleur et le commutateur sont transmis via un canal sécurisé, implanté via une connexion TLS sur TCP. Le commutateur initie la connexion TLS lorsqu'il connaît l'adresse IP du contrôleur. Chaque message entre le commutateur et le contrôleur commence avec l'en-tête OpenFlow.

Cet en-tête spécifie le numéro de version OpenFlow, le type de message, la longueur de message, et l'identificateur de transaction du message.

Il existe trois catégories de message : **symétrique, contrôleur-commutateur et asynchrone.**

## 3.3/d-1 : Messages symétriques :

Les messages symétriques peuvent être émis indifféremment par le contrôleur ou le commutateur sans avoir été sollicité par l'autre entité. Par exemple, on trouve les messages **HELLO** qui sont échangés une fois que le canal sécurisé a été établi.

Les messages **ECHO** sont utilisés par n'importe quelle entité (contrôleur, commutateur) pendant le fonctionnement du canal pour s'assurer que la connexion est toujours en vie et afin de mesurer la latence et le débit courants de la connexion. Chaque message **ECHO_REQUEST** doit être acquitté par un message **ECHO_REPLY.**

## 3.3/d-2 : Messages asynchrones:

Les messages asynchrones sont émis par le commutateur au contrôleur sans que le commutateur ait été sollicité par le contrôleur.

Par exemple, on peut citer le message **PACKET_IN** qui est utilisé par le commutateur pour passer les paquets de données au contrôleur pour leur prise en charge (lorsqu'aucune entrée de flux ne correspond au paquet entrant ou lorsque l'action de l'entrée correspondante spécifie que le paquet doit être relayé au contrôleur).

Si le commutateur dispose d'une mémoire suffisante pour mémoriser les paquets qui sont envoyés au contrôleur, les messages **PACKET-IN** contiennent une partie de l'en-tête (par défaut 128 octets), les commutateurs qui ne supportent pas de mémorisation interne (ou ne disposant plus de mémoire) émettent le paquet entier au contrôleur dans le message **PACKET-IN.**

Le commutateur peut informer le contrôleur qu'une entrée de flux a été supprimée de la table de flux via le message **FLOW_REMOVED**. Cela survient lorsqu'aucun paquet entrant n'a de correspondance avec cette entrée pendant un temporisateur spécifié par le contrôleur lors de la création de cette entrée au niveau de la table de flux du commutateur.

Le message **PORT_STATUS** est utilisé afin de communiquer un changement d'état du port (le lien est hors service). Finalement le commutateur utilise le message **ERROR** pour notifier des erreurs au contrôleur.

## 3.3/d-3 Messages contrôleur-commutateur:

Les messages contrôleur-commutateur représentent la catégorie la plus importante de messages OpenFlow. Ils peuvent être représentés en cinq sous-catégories : **switch configuration, command from controller, statistics, queue configuration, et barrier.**

α): Les messages **switch configuration** consistent en un message unidirectionnel et deux paires de messages requête-réponse.

- Le contrôleur émet le message unidirectionnel **SET_CONFIG** afin de positionner les paramètres de configuration du commutateur.

- Le contrôleur utilise la paire de message **FEATURES_REQUEST** et FEATURES_REPLY afin d'interroger le commutateur au sujet des fonctionnalités (notamment optionnelles) qu'il supporte.

- La paire de message **GET_CONFIG_REQUEST** et **GET_CONFIG_REPLY** est utilisée afin d'obtenir la configuration du commutateur.

β): Les messages **command from controller** sont au nombre de 3. **PACKET-OUT** est analogue à **PACKET_IN** mentionné précédemment.

- Le contrôleur utilise **PACKET_OUT** afin d'émettre des paquets de données au commutateur pour leur acheminement via le plan usager (Plan de données).

- Le contrôleur modifie les entrées de flux existantes dans le commutateur via le message **FLOW_MOD**.

- PORT_MOD est utilisé pour modifier l'état d'un port OpenFlow.

λ): Des statistiques sont obtenues du commutateur par le contrôleur via la paire de message **STATS_REQUEST et STATS_REPLY.**

γ): La configuration de files d'attente associées à un port n'est pas spécifiée par OpenFlow. Un autre protocole de configuration doit être utilisé pour ce faire. Toutefois le contrôleur peut interroger le commutateur via **QUEUE_GET_CONFIG_REQUEST** acquitté par **QUEUE_GET_CONFIG_REPLY** pour apprendre quelle est la configuration des files d'attente associées à un port afin de pourvoir acheminer des paquets sur des file d'attente spécifiques et ainsi fournir à ces paquets un niveau de QoS désiré.

δ): Le message **BARRIER_REQUEST** est utilisé par le contrôleur pour s'assurer que tous les messages OpenFlow émis par le contrôleur et qui ont précédé cette requête ont été reçus et traités par le commutateur.

## 3.4 :Les routeurs programmables :

Habituellement, les routeurs hautes performances ont été conçus en utilisant des ASICs (*Application-Specific Integrated Circuit*) permettant de traiter les paquets à grande vitesse. Généralement, ces ASICs regroupent un grand nombre de fonctionnalités mais n'offrent aucune flexibilité. Ils ne peuvent présenter que les fonctionnalités préalablement prédéfinies par les concepteurs.

Récemment, une autre approche utilisant les processeurs réseau (Network Processor) pour la conception de routeurs hautes performances et programmables a gagnée en popularité.

L'objectif est d'offrir une solution intégrant des fonctions applicatives aux routeurs. Les processeurs réseau sont programmables et offrent des fonctions génériques qui peuvent être utilisées dans différents types de réseaux. L'objectif est d'offrir une flexibilité dans l'aiguillage des paquets arrivant au port d'entrée. Les processeurs réseau sont entièrement programmables et offrent la possibilité de modification des instructions de traitement des paquets afin d'atteindre la vitesse des ASICs.. Les routeurs programmables sont des routeurs qui peuvent exécuter un code personnalisé tout en continuant à utiliser le format standard de paquets. Par conséquent, ils facilitent le développement et le déploiement rapide de nouveaux services dans les réseaux. En outre, Ils créent une excellente plate-forme pour tester de nouvelles architectures de service réseau.

Afin d'expliquer le fonctionnement et le rôle de chaque composante d'un routeur programmable, nous présentons dans la **Figure 3.3** un schéma d'une architecture classique d'un routeur programmable.

Un routeur programmable contient plusieurs cartes réseaux (line cards). Ces dernières représentent le point d'entrée et de sortie des paquets. Lorsqu'un paquet arrive au niveau de la carte d'entrée (ingress line card), le processeur réseau traite le paquet et effectue la recherché dans la table de routage afin de trouver la carte de sortie (egress line card) .

C'est l'opération de recherche de route (IP lookup) qui vise à trouver la meilleure route pour un paquet à partir de son adresse de destination. Cette opération nécessite des mémoires telles que la TCAM (*Ternary content-addressable memory*), la SRAM (*static random-access memory*), la DRAM (*Dynamic random-access memory*). Le paquet transite par la suite par le *Switch Fabric*. Les composants essentiels dans une carte ligne sont l'émetteur-récepteur (Transceiver), le dispositive de tramage (*Framer*), le CPU, le gestionnaire de trafic (*Traffic Manager*) et plusieurs autres types de mémoire.

L'émetteur-récepteur permet la conversion des signaux comme la conversion d'un signal optique vers un signal électrique. Le dispositif de tramage est responsable de la délimitation des trames selon les exigences de la signalisation physique. il garantit la bonne réception du paquet. Le gestionnaire de trafic est responsable de toutes les fonctionnalités du contrôle de flux. Il assure aussi la gestion des mémoires tampons. Le CPU est responsable des fonctionnalités nécessaires au traitement des paquets comme la mise à jour des tables de routage. La TCAM est une mémoire spéciale à très haute vitesse qui permet de rechercher l'ensemble de contenus dans un seul cycle d'horloge. Elle est très utilisée dans les routeurs hautes performances pour augmenter la vitesse du processus de recherche de route et de la transmission de paquets.

Chaque adresse dans la TCAM est composée de l'adresse du réseau et celle de la machine. La SRAM permet d'enregistrer les informations sur le transfert des paquets à titre d'exemple le saut suivant (next hop) ou le port de sortie. Elle utilise le résultat de la recherche dans la TCAM pour s'acquérir des informations de renvoi du paquet à partir de son en-tête. La DRAM est utilisée comme une mémoire tampon lorsque les paquets sont traités par le processeur réseau.

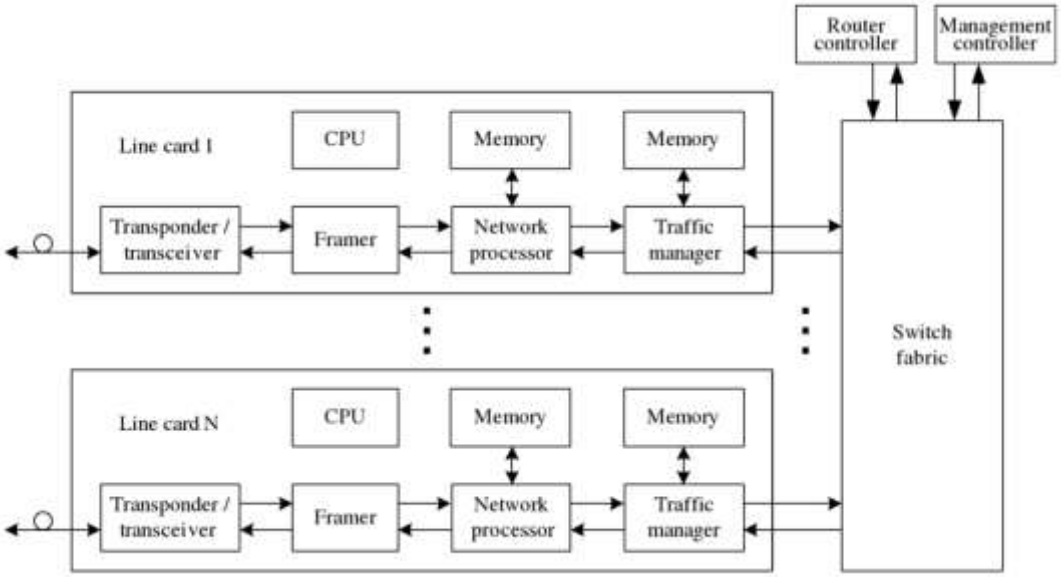


**Figure 3.7: Architecture d'un routeur programmable**

# CHAPITRE (04)

## Mise en place d'une solution SDN

### 4.1 : Introduction :

Dans un système distribué, il existe des enjeux à résoudre tels que la gestion des ressources et l''ordonnancement des applications car, ces taches sont compliquées et il n'existe pas une solution optimale pour répondre à ces issues.

D'autre part, dans l'environnement d'un système distribué comme Cloud, il est difficile d'effectuer les different scénarios avec différent nombre de ressources et utilisateurs afin d'évaluer la performance des algorithemes du partage de charge, Broker, gestion des ressources … etc.

Lorsque en veut évaluer les scénarios de manière répétable et controlable, cela est parfois impossible car l'issue du cout et la gestion. Afin de résoudre cette issue, les chercheurs utilisent les simulateurs pour effectuer leurs scénarios avant de les effectuer au sein d'un système distribué réel.
Dans ce qui suit, on va presenter l'un des outils de simulation le plus connu et utilize dans le domaine de simulation du Cloud.

Il y a plusieurs outils de simulation de systèmes distribué. Les plus connus sont Packet Ttacer, MiniNet, CloudSim, GNS3…. . Dans ce qui suit, on va présenter le framework MiniNet, qui permet de la simulation de l'environnement du Cloud computing.

Ce chapitre représente la première contribution de notre travail, nous commençons d'abord par une présentation de Mininet et open Vswitch, puis nous parlerons du contrôleur POX ainsi que ses fonctionnalités.

## 4.2: Emulateur du réseau MiniNet :

Mininet est un émulateur de réseau, il permet de créer une topologie réseau qui se compose d'un ensemble de hosts, de switchs, de contrôleurs et liens virtuels. Il fournit la capacité de créer des hôtes, les commutateurs et contrôleurs via:

- Ligne de commande,
- Interface interactive,
- Script Python.

### 4.2-1 : Configuration utilisée :

Afin de bien visualiser le fonctionnement du SDN, nous avons choisi d'utiliser une machine virtuelle téléchargée depuis:

https://github.com/mininet/mininet/releases/download/2.2.2/mininet-2.2.2-170321-ubuntu- 14.04.4-server-amd64.zip et accessible via VirtualBox, elle utilise le système d'exploitation Linux (Ubuntu-14.04.4).

### 4.2-2 : Fonctionnement de base du Mininet :

Mininet permet avec un simple jeu de commande de réaliser des réseaux virtuels en utilisant la commande mn.

### 4.2-2/a : Création d'une topologie avec la ligne de commande :

Mininet fournit, en plus de la topologie « minimal », la topologie « single », « linear » et « tree » comme montre dans la **Figure (3-1) .**

Pour charger l'une de ces topologies, on utilise l'option «**--topo**», par exemple :

**$ sudo mn –topo single**.

« single » tout court donne la même topologie que minimal, c'est-à-dire créer un hôte relie avec un switch, mais on peut ajouter également comme argument un chiffre, qui indique le nombre d'hôtes à créer.

**$ sudo mn –topo single, 4**

Si nous souhaitons personnaliser une topologie déjà créée, nous appliquons une des commandes citées ci-dessous:

- Ajouter un switch : **self.addSwitch('s1')**
- Ajouter un hôte : **self.addHost('h1')**
- Ajouter un lien : **self.addLink(h1,s1)**

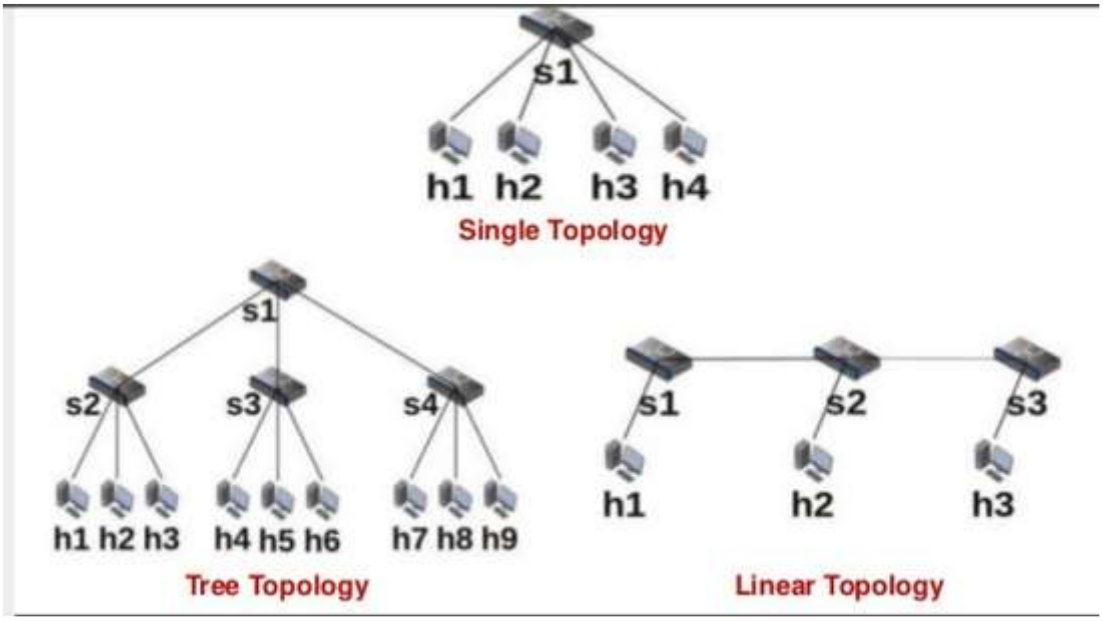


**Figure 4.1: Different types de topologie reseau**

Apres la configuration, nous devons tester si les paquets sont routes correctement avec la commande ping qui est un bon moyen pour vérifier la connectivité:

**Mininet> h1 ping -c h2 ou bien Mininet> pingall**

et pour analyser les règles insérées dans chaque commutateur, nous utilisons la commande dpctl: **Mininet> dpctl dump-flows**, et d'une façon générale, pour accéder a toutes les commandes utilisées sous Mininet, il suffit de taper :

**Mininet> help**

## 4.2-2/b : Topologies personnalisées

Si nous souhaitons travailler avec une topologie autre que tree, linear ou simple, et même sans la ligne de commande, il faut créer une topologie personnalisée à l'aide des scripts Python. Par exemple, si on veut créer 2 hôtes h1 et h2, les deux sont connectes a un switch s1.

Voici le script Python à écrire:

```
Class Test_Topo(Topo):
    def __init__(self):
        "Create P2P topology"
        # Initialize topology
        Topo.__init__(self)
        # Add hosts and switches
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        s1 = self.addSwitch('s1')
        Add links #
        self.addLink(h1,s1)
        self.addLink(h2,s1)
        topos = { 'toptest': (lambda: Test_Topo())}
```

Apres, nous sauvegardons le code dans un fichier toptest.py, et nous executons la commande suivante:

**$ sudo mn --custom /chemin/vers/toptest.py --topo toptest**

## 4.3 : Open vSwitch:

Open vSwitch est bien adapte pour fonctionner comme un commutateur virtuel dans les environnements Virtuels. En plus, il supporte plusieurs technologies de virtualisation basées sur Linux, y compris Xen, KVM et VirtualBox.

Open vSwitch est donc une implémentation logicielle d'un switch Ethernet. Concrètement il est constitué d'un service (ovs-vswitchd) et d'un module kernel (openvswitch_mod).

Le service permet de commuter effectivement les paquets vers les bons ports virtuels, alors que le module kernel permet de capturer le trafic provenant des interfaces réseau.

Pour fonctionner comme n'importe quel switch, Open vSwitch utilise la notion de ports. Chaque port est constitué d'une ou plusieurs interfaces, qui correspondent à des interfaces du système hôte (logiques ou physiques).

## (Interagir avec Open vSwitch) :

- Pour connaitre l'état global d'un switch: **ovs-vsctl show**
- La création d'un switch se fait par la commande:

**ovs-vsctl add-br <nom du virtuel switch>**

- Pour ajouter un flux directement dans la table de flux sans avoir utiliser un controleur, nous utilisons la commande: **dpctl**

**mininet> dpctl add-flow in_port=1,actions=output:2**
**mininet> dpctl add-flow in_port=2,actions=output:1**

Cette commande permet de transférer les paquets arrivant à port1 vers port 2 et vice versa.

- Pour vérifier le contenu de la table de flux: **$ dpctl dump-flows**

## 4.4 : Contrôleur SDN

Les deux principaux piliers de l'approche SDN sont la programmabilite des equipements reseaux et la mise en oeuvre d'un point de controle centralise.

Dans les SDN, le plan de controle est place dans un controleur centralise qui a une visibilite sur l'ensemble du reseau, y compris les hotes qui s'y connectent, les applications qui disent aux controleurs comment programmer le reseau utilisent des interfaces de programmation ≪ **NorthBound** ≫ tandis que les API et protocoles utilises par le controleur pour communiquer avec les equipements du reseau sont dits ≪ **SouthBound** ≫.

### Floodlight et Opendaylight**:**

Nous avons choisi d'etudier et de comparer deux controleurs : Floodlight et Opendaylight afin d'evaluer leurs parametres de performance tel que le debit et la latence.

### 4.4-1: Fonctionnalité de base des 2 contrôleurs :

- Floodlight est un controleur de reseau defini par un logiciel, soutenu par la societe Big Switch, il offre un point de gestion central pour les reseaux OpenFlow et il peut gerer des peripheriques tel que open vswitch de maniere transparente. Il prend egalement en charge une large gamme de commutateurs physiques OpenFlow, de sorte qu'il simplifie grandement la gestion du reseau.

Il est sous licence Apache et ecrit en java et il a pour mission :

- Installer / Supprimer les regles de transfert sur les commutateurs.
- Decouverte de la topologie : besoin de savoir a quoi ressemble le reseau (Link Layer Discovery Protocol ( LLDP )).
- Statistiques : besoin de savoir ce qui se passe dans le reseau.

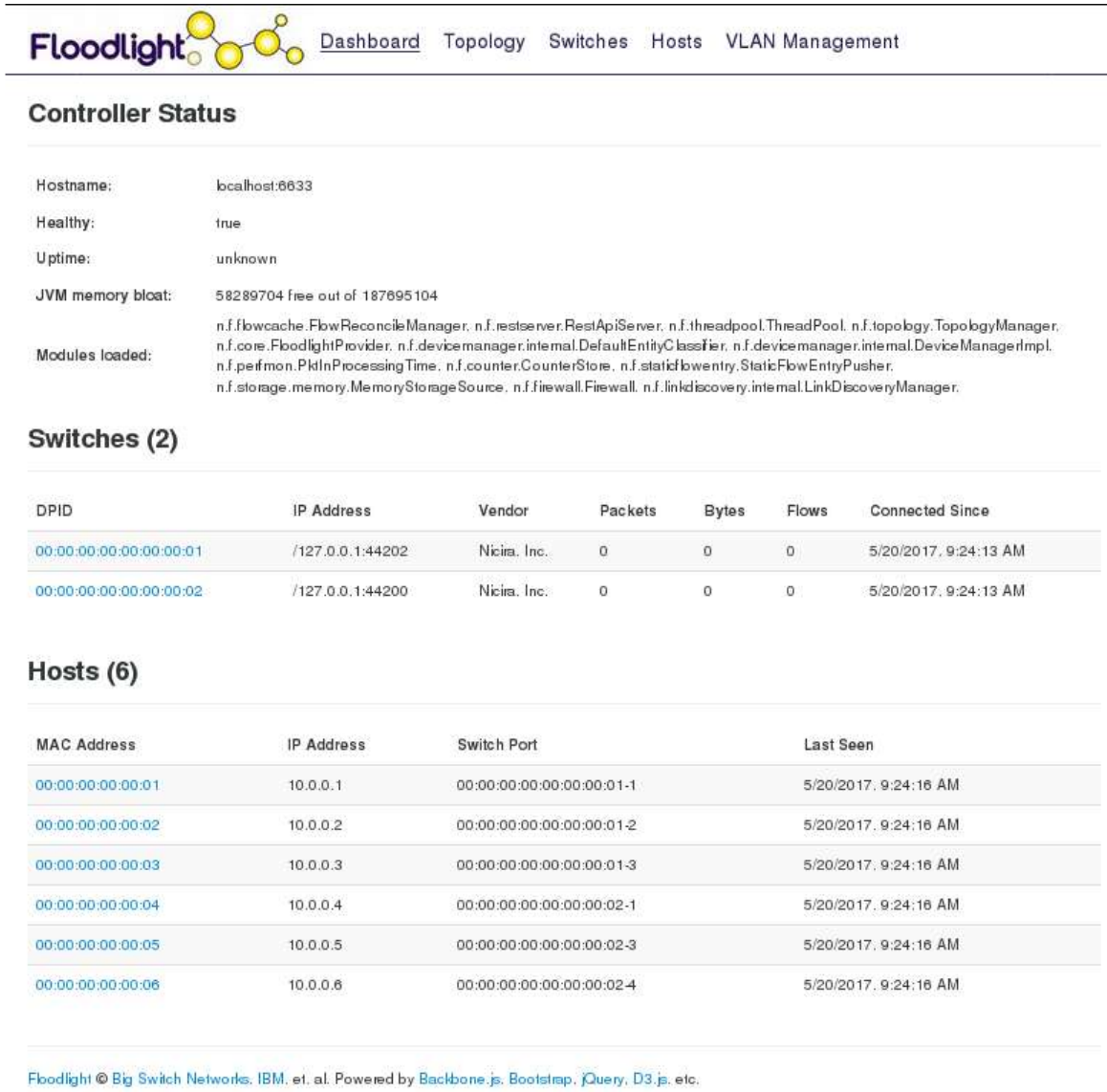


**Figure 4.2: Interface web du Floodlight.**

- OpenDaylight est un projet open source pris en charge par IBM, Cisco, Juniper, VMWare et plusieurs autres grands fournisseurs de reseau. OpenDaylight est une plate-forme de controleur SDN implementee en Java, il peut etre deploye sur n'importe quelle plate-forme materielle et systeme d'exploitation prenant en charge Java.

Le projet OpenDaylight repose sur des principes de developpements ouverts et transparents, il vise a reunir les acteurs du reseau pour travailler sur des solutions communes.
Big Switch Networks, Cisco, Citrix, Ericsson, IBM, Juniper Networks, Microsoft, Red Hat ou encore VMware sont ainsi les membres fondateurs d'OpenDaylight.

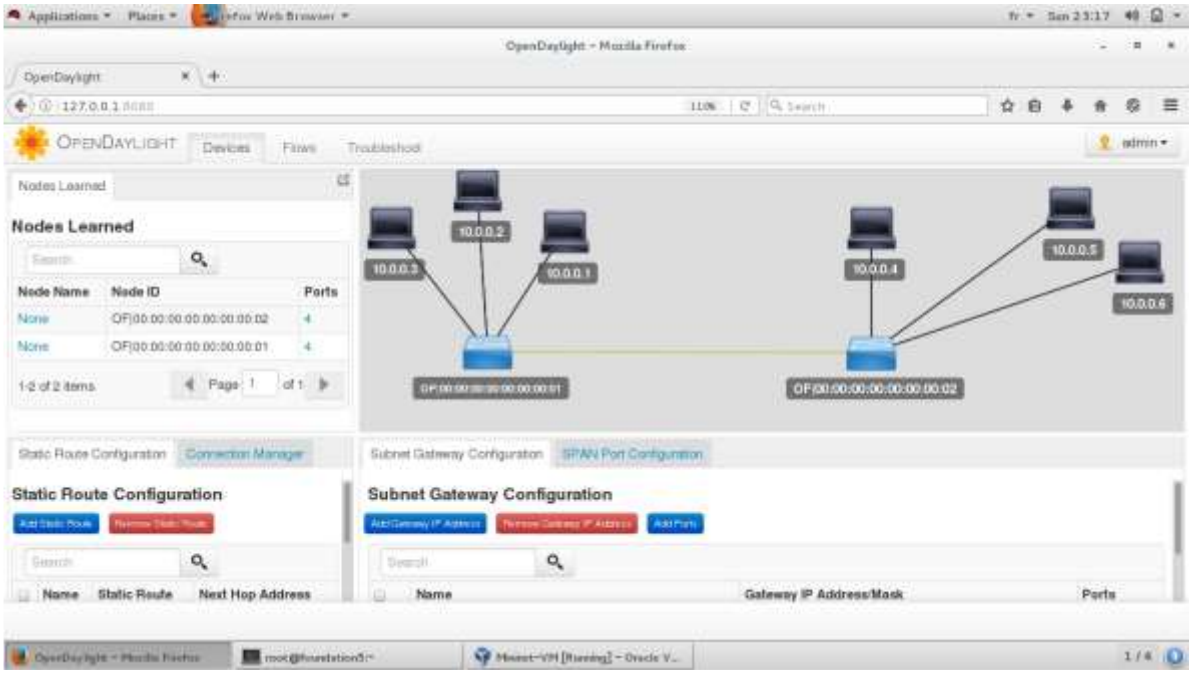


**Figure 4.3: Interface web du Opendaylight**

## 4.4-b): Comparaison entre Floodlight et Opendaylight :

| **Caractéristiques** | **Floodlight** | **OpenDaylight** |
|---|---|---|
| Développé par | Big Switch Networks | Linux Foundation |
| Soutenu par | Big Switch Networks | Cisco, HP, IBM , Juniper, VMWare, etc. |
| Ecrit en Langage | java | java |
| Langages supportés | Java, Python et n'importe quell langage qui supporte Rest API | java |
| Open source | oui | oui |
| Interface utilisateur | web | web |
| Virtualisation | Mininet, OpenVswitch | Mininet, OpenVswitch |
| Interface | southbound (OpenFlow), northbound (Java, REST) | southbound (OpenFlow et autres protocoles), northbound (Java RPC) |
| Plateforme | Linux, Mac, Windows | Linux, Windows |
| Documentation | Site officiel | Moyen |
| Activité de liste de mailing | tres eleve | Moyen |

**Tableau (3-1) :** comparaison entre Floodlight et Opendaylight.

# CHAPITRE (05)

## Réalisation d'une application SDN « Gestion des VLANS »

Par rapport aux réseaux traditionnels, l'introduction de SDN et OpenFlow peut simplifier la gestion des VLANs tout en offrant une grande flexibilité:

• Améliorer l'efficacité du réseau grâce à une gestion et un contrôle centralisé, ainsi qu'un niveau élevé d'automatisation. Le contrôleur peut définir des règles et fournir un contrôle d'accès.

• Améliorer la disponibilité du service.

• Fournir une analyse avancée de toutes les ressources afin qu'on puisse facilement surveiller et contrôler ces ressources et prendre des décisions stratégiques.

• Réduction des dépenses d'investissement , ainsi que maximiser l'utilisation des ressources.

Alors, bien que SDN et OpenFlow puissent remplacer toutes les infrastructures de gestion de réseau actuelles, l'utilisation du VLAN est encore nécessaire, et même les administrateurs de réseau continuent à les utilises afin d'avoir des réseaux sécurisés.

Donc, SDN et OpenFlow seraient une bonne solution pour faciliter la gestion des VLANS dans les réseaux d'entreprise et de campus.

### 5.1:Implémentation d'une application de gestion des VLANS avec SDN

L'objectif de notre projet est d'avoir une application centralisée, qui peut aider l'administrateur réseau à configurer et à gérer le VLAN d'une manière souple et efficace.

Les principales démarches de notre travail sont les suivantes:

• Mettre en place d'une topologie à l'aide d'un outil d'émulation basé sur Mininet, et d'un contrôleur Floodlight.

• Conception et développement d'une application de gestion de VLAN via une interface web (GUI).

Nous avons basé dans les différentes parties de nos tests sur 2 scenarios:

## 5.1/a: Scenario 1 : Utilisation d'un seul switch

Dans ce scenario, nous avons créé une topologie réseau, montrée dans la **Figure 5.1**, qui se compose d'un seul switch connecté avec un contrôleur Floodlight, et avec plusieurs hôtes (de 1 à 5).

Puis nous avons configuré 2 VLANs: **vlan10** et **vlan20**, chacun se compose de 2 hosts, et évidemment le test du fonctionnement de ces VLANs a été fait via la commande **ping**.

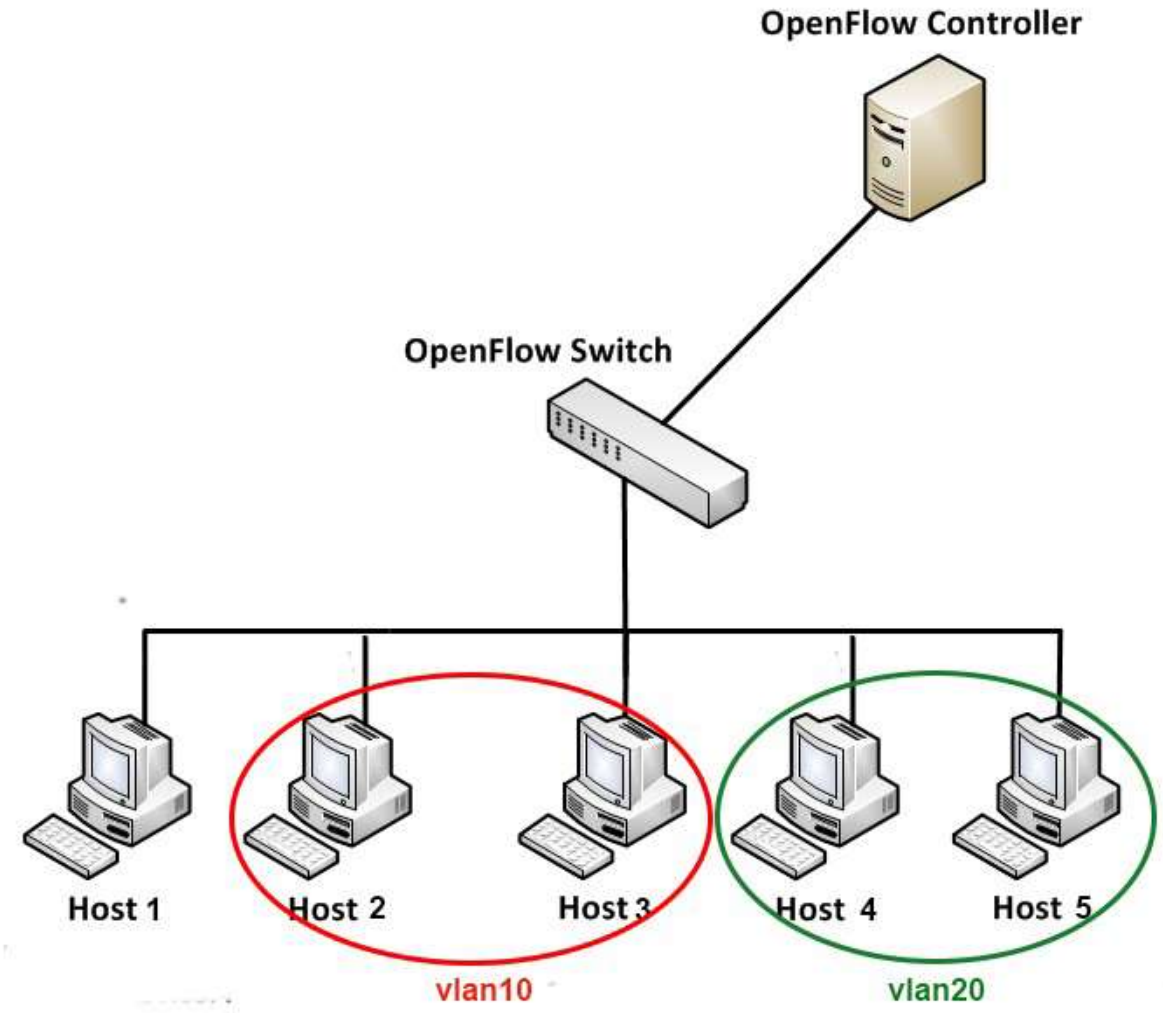


**Figure 5.1: Topologie du scenario 1**

## 5.1/b: Scenario 2 (Utilisation de 2 switchs):

Ici, nous avons créé une autre topologie réseau, montrée dans la **Figure 5.2**, qui se compose de 2 switchs, chacun est connecté avec plusieurs hôtes (de 1 à 6), les 2 switches sont reliés entre eux et relié aussi avec un contrôleur Floodlight, puis nous avons configuré 2 VLANs: **vlan10** et **vlan20.**

**Vlan10** se compose d'un hôte connecté au 1er switch et d'un autre hôte connecté au 2ème switch, et c'est pareil pour le **vlan20**.

Nous avons testé par la suite la connectivité des hôtes d'un VLAN avec la commande **ping**.

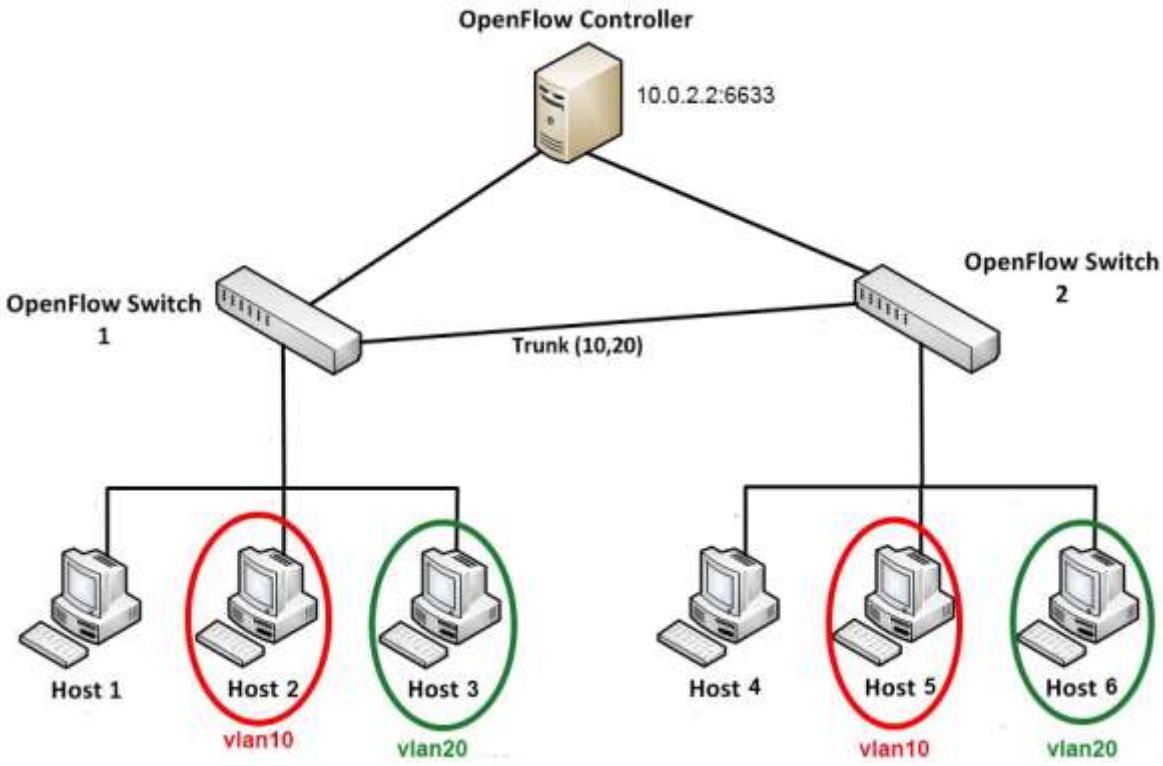


**Figure 5.2: Topologie du Scenario 2**

## 5.2: Mettre en place une topologie à l'aide du Mininet:

### 5.2/a : Création de topologie avec Mininet :

#### 5.2/a-1 : Scenario 1 :

pour créer la topologie du scénario 1, nous avons exécuté la ligne de commande suivante :

**$ sudo mn --topo=single,5 --mac --switch=ovs --controller=remote,ip=10.0.2.2,port=6633**

```
mininet@mininet-vm:~$
mininet@mininet-vm:~$
mininet@mininet-vm:~$ sudo mn --topo single,5  --mac  --switch ovs --controller=
remote,ip=10.0.2.2,port=6633
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
mininet>
```

**Figure 5.3: Création du topologie de scenario 1**

Aprés avoir créer la topologie, nous pouvons la tester en exécutant des différentes commandes, nous pouvons utiliser la commande **help** pour avoir cette liste de commandes, comme montre la **Figure 5.4.**

```
mininet>
mininet>
mininet>
mininet> help

Documented commands (type help <topic>):
========================================
EOF    gterm  iperfudp  nodes        pingpair      py      switch
dpctl  help   link      noecho       pingpairfull  quit    time
dump   intfs  links     pingall      ports         sh      x
exit   iperf  net       pingallfull  px            source  xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet> _
```

**Figure 5.4: Exécution de la commande help du Mininet.**

Par exemple, la **Figure 5.5** nous affiche la sortie de la commande **dump** (la configuration de la topologie) ainsi que la sortie de la commande **net** (qui nous montre les liens entre hôtes et switch).

```
mininet>
mininet>
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1490>
<Host h2: h2-eth0:10.0.0.2 pid=1493>
<Host h3: h3-eth0:10.0.0.3 pid=1495>
<Host h4: h4-eth0:10.0.0.4 pid=1497>
<Host h5: h5-eth0:10.0.0.5 pid=1499>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None,
s1-eth5:None pid=1504>
<RemoteController{'ip': '10.0.2.2', 'port': 6633} c0: 10.0.2.2:6633 pid=1484>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
h4 h4-eth0:s1-eth4
h5 h5-eth0:s1-eth5
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0 s1-eth4:h4-eth0 s1-eth5:
h5-eth0
c0
mininet>
mininet>
mininet>
mininet>
```

**Figure 5.5: sortie des commandes net et dump**

## 5.2/a-2: Scenario 2 :

pour créer la topologie du scénario 2 , nous avons utilisé un fichier python : **multiswitch-vlan.py**, puis nous avons appelé ce fichier dans la ligne de commande ci-dessous :

**$ sudo mn --mac --custom multiswitch-vlan.py --topo=mytopo --switch=ovs --controller=remote,ip=10.0.2.2,port=6633**

```
from mininet.topo import Topo
class MyTopo( Topo ):
    "Simple topology example."
    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        Host1 = self.addHost( 'h1' )
        Host2 = self.addHost( 'h2' )
        Host3 = self.addHost( 'h3' )
        Host4 = self.addHost( 'h4' )
        Host5 = self.addHost( 'h5' )
        Host6 = self.addHost( 'h6' )
        Switch1 = self.addSwitch('s1')
        Switch2 = self.addSwitch('s2')
        # Add links
        self.addLink( Host1, Switch1 )
        self.addLink( Host2, Switch1 )
        self.addLink( Host3, Switch1 )
        self.addLink( Host4, Switch2 )
        self.addLink( Switch1, Switch2 )
        self.addLink( Host5, Switch2 )
        self.addLink( Host6, Switch2 )
topos = { 'mytopo': ( lambda: MyTopo() ) }
~
~
"multiswitch-vlan.py" 27L, 878C                              1,1           All
```

**Figure 5.6: Fichier multiswitch-vlan.py**

```
mininet@mininet-vm:~/mininet/perso-test$
mininet@mininet-vm:~/mininet/perso-test$ sudo mn --mac  --custom multiswitch-vla
n.py  --topo=mytopo --controller=remote,ip=10.0.2.2,port=6633
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet>
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet>
```

**Figure 5.7: Topologie du scenario 2**

## 5.2/b: Contrôleur Floodlight

### 5.2/b-1: Pré-requis :

- Pour étudier et tester un réseau SDN, nous avons utilisé une machine Linux (Redhat Enterprise linux 7), et installé les outils de développement JDK et eclipse.
- Puis, nous avons téléchargé et installé Floodlight. (Voir la partie Annexe.)

### 5.2/b-2: Exécution du Floodlight :

Pour démarrer Floodlight, nous exécutons le fichier floodlight.jar avec la commande suivante: # java -jar floodlight.jar et via un navigateur web, nous accèdons à l'URL suivant:

**http://localhost:8080/ui/index.html**

Les figures ci-dessous, représentent l'interface web du contrôleur Floodlight, nous avons choisi d'étudier dans cette partie une topologie du scenario 2 qui constitue de 2 switch, chacun est relié avec 3 hôtes.

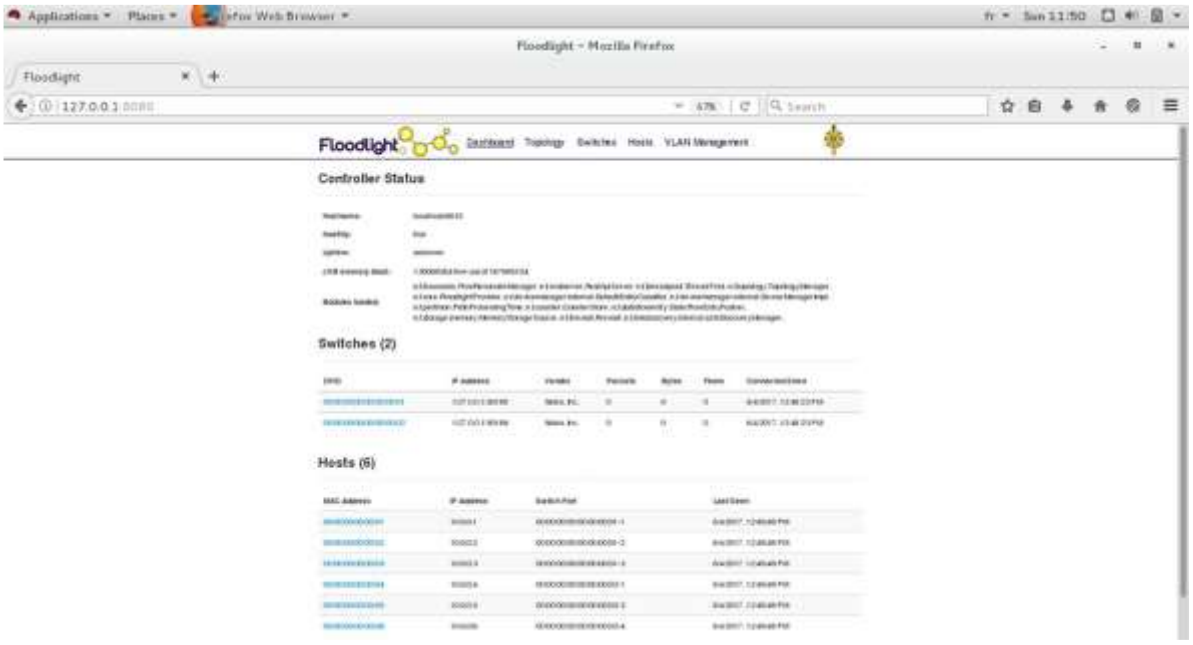


**Figure 5.8: Interface web du Floodlight**

Une fois la topologie est créée avec mininet, tout le détail sera affiché dans la page d'accueil de Floodlight comme montre la **Figure 5.8.**

Si on clique sur le lien switches, on aura l'adress IP et l'adress MAC du switch ainsi que le nombre de ports et des informations sur ces derniers tel que le flux circulant, **Figure 5.9.**

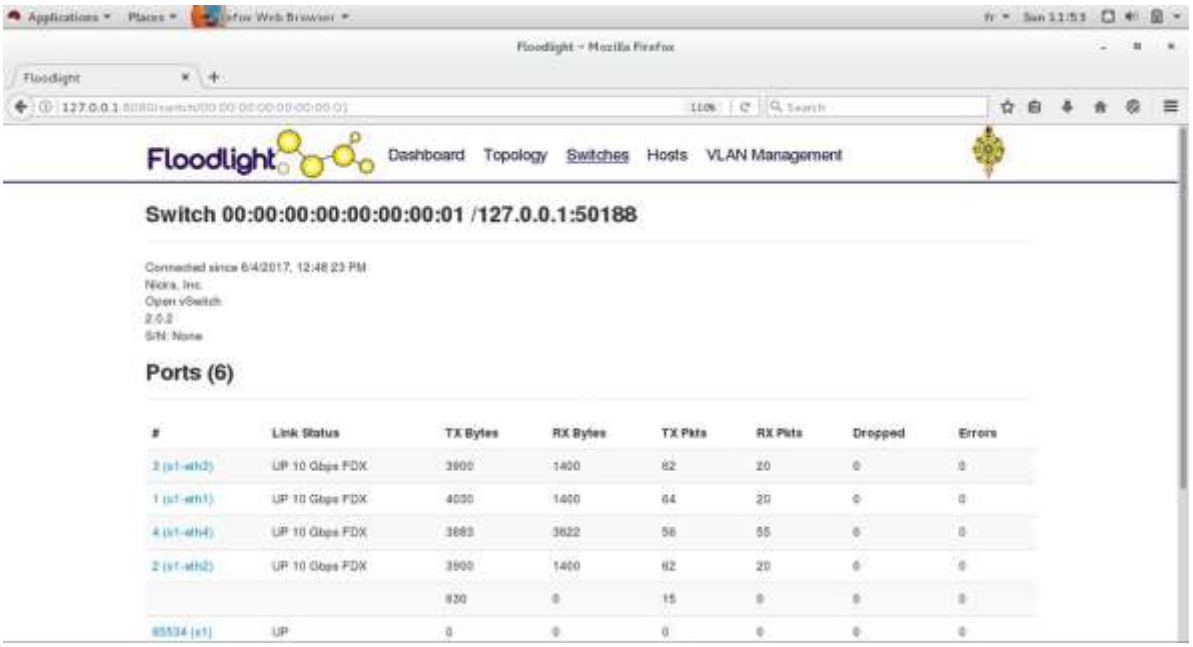


**Figure 5.9: Onglet switches de l'Interface web du Floodlight**

Et si on clique sur le lien topology, on aura :

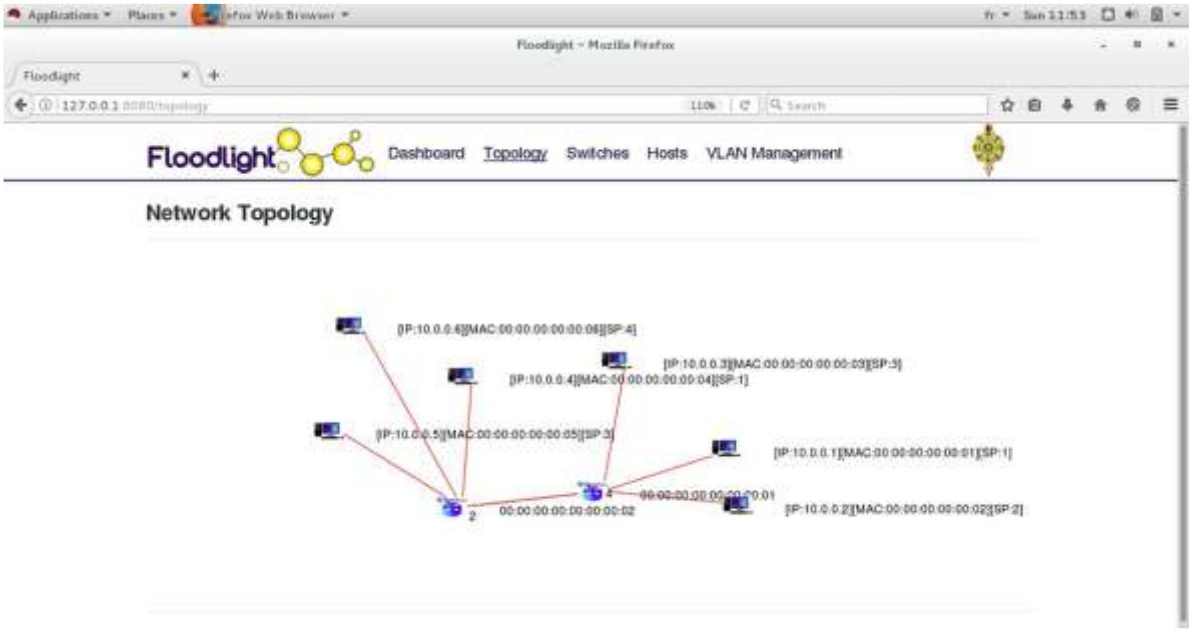


**Figure 5.10: Onglet Topology de l'Interface web du Floodlight**

## 5.3 : Conception et développement d'une application de gestion de VLAN

### 5.3/a: Présentation de l'application

Notre application «Vlan Management » permet de réaliser toutes les tâches de gestion de VLAN, y compris l'ajout d'un nouveau réseau VLAN, la suppression d'un VLAN ou de tous les VLAN, l'édition d'un VLAN existant et l'affichage de la topologie des réseaux VLAN.

Ce module est accessible via l'interface web du Floodlight, dans la partie header figure le lien **« VLAN Management »**, nous avons utilisé le langage javascript, qui peut être intégré directement au sein des pages web, pour y être exécuté sur le poste client.

Généralement, JavaScript sert à contrôler les données saisies dans des formulaires HTML, nous avons intégré le JavaScript dans la méthode **AJAX**（ ***A****synchronous* ***J****avascript* ***A****nd* ***X****ML*）qui sert à modifier le contenu des pages web, des scripts en langage JavaScript vont envoyer des demandes au serveur Web en utilisant l'objet **XMLHTTPRequest.**

Le transfert de données est géré par le JavaScript, en utilisant la technologie de formatage de données, **JSON**（*JavaScript Object Notation*）qui est un format de données textuel, permet de représenter des informations structurées.

### 5.3/a-1: Ajout d'un VLAN:

Nous utilisons cette fonction afin de créer un nouveau réseau VLAN. Une fois ce dernier est créé, un ensemble de règles de flux sont installés dans la table de flux du switch.

Les informations requises pour la création sont : le nom et l'ID du VLAN, les ports auxquels les hôtes VLAN sont connectés. Nous avons également défini deux types de ports:

**Tagged** et **untagged ports**.

Les ports tagged (étiqueté) sont des ports où un identifiant VLAN sera ajouté dans l'en-tête du paquet. Les ports untagged sont des ports où l'ID VLAN sera supprimé de l'en-tête du paquet. Les ports de transfert sont les ports du commutateur qui n'ont pas d'hôte VLAN, mais qui sont impliqués dans le chemin d'accès au VLAN.

Chaque communication entre 2 hôtes consiste à deux entrées de flux, une pour envoyer le paquet dans un sens et l'autre utilisée dans l'autre sens. Pour cela, deux actions clés ont été utilisées pour la configuration des flux, **set-vlan-id** qui correspond à un tagged port et **strip-vlan** qui correspond à un port untagged.

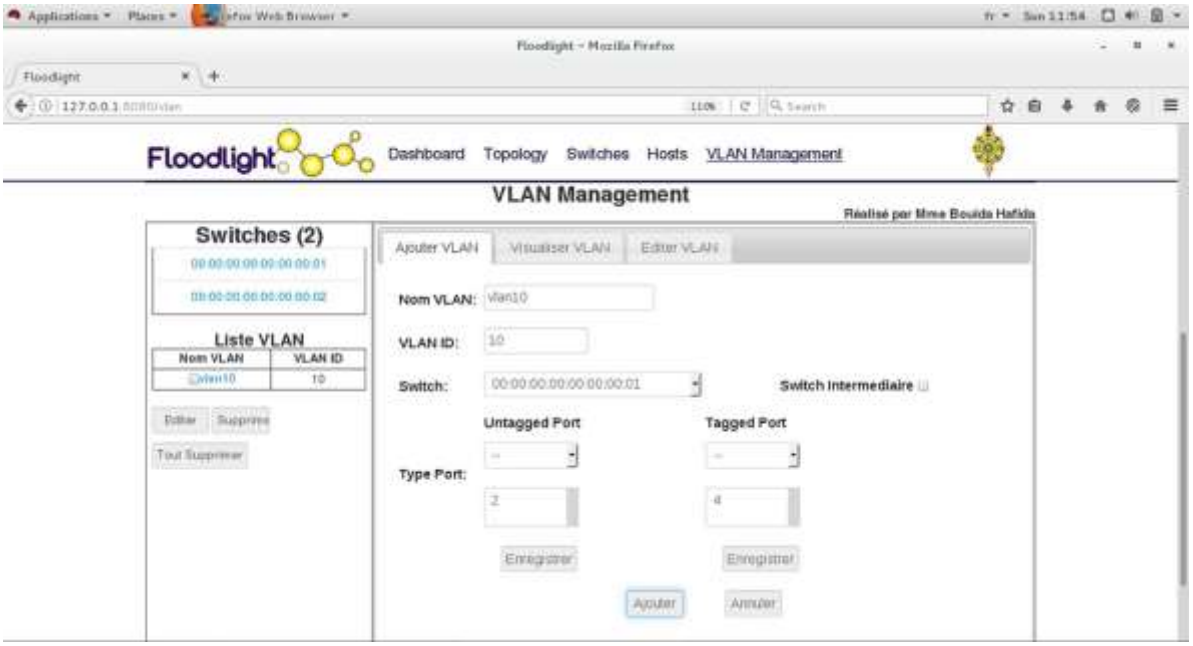


**Figure 5.11: Ajout d'un VLAN**

**H1 at port 1 Openflow Switch 01 (DPID=00:0a:6c:3b:e5:f9:7c:80)**

"switch": "00:0a:6c:3b:e5:f9:7c:80","name":"VLAN 10-1","ingress-port":"1","actions":"set-vlan-id=10,output=2"

"switch": "00:0a:6c:3b:e5:f9:7c:80","name":"VLAN 10-2","ingress-port":"2", "vlan-id":"10","actions":"strip-vlan,output=1"

**H2 at port 2 of Openflow Switch 03 (DPID = 00:0a:6c:3b:e5:f9:99:74)**

"switch": "00:0a:6c:3b:e5:f9:99:74","name":"VLAN 10-5","ingress-port":"2","actions":"set-vlan-id=10,output=3"

"switch": "00:0a:6c:3b:e5:f9:99:74","name":"VLAN 10-6","ingress-port":"3", "vlan-id":"10","actions":"strip-vlan,output=2"

**Figure 5.12: Exemple d'une règle de flux VLAN ajoutée à la table des flux d'un switch**

L'exemple illustré dans la **Figure 5.11** montre que le port numéro 1 du Switch 01 et le numéro de port 2 du switch 03 sont des ports tagged ( étiqueté ), tandis que le port numéro 2 du switch 01 et le numéro de port 3 du switch 03 sont des ports untagged ( non étiqueté ).

## 5.3/a-2 : Suppression d'un VLAN:

Cette fonction permet à un administrateur réseau de supprimer facilement le réseau VLAN en un seul clic sur l'interface graphique Web, au lieu d'avoir à le supprimer manuellement dans les commutateurs individuels. Cela faisant que toutes les entrées de flux dans tous les commutateurs appartenant au VLAN seront supprimées en même temps.

## 5.3/a-3 : Modification d'un VLAN:

Cette fonction permet aux administrateurs réseau de modifier un VLAN existant, tel que ajout ou suppression d'un nouvel hôte d'un VLAN existant.

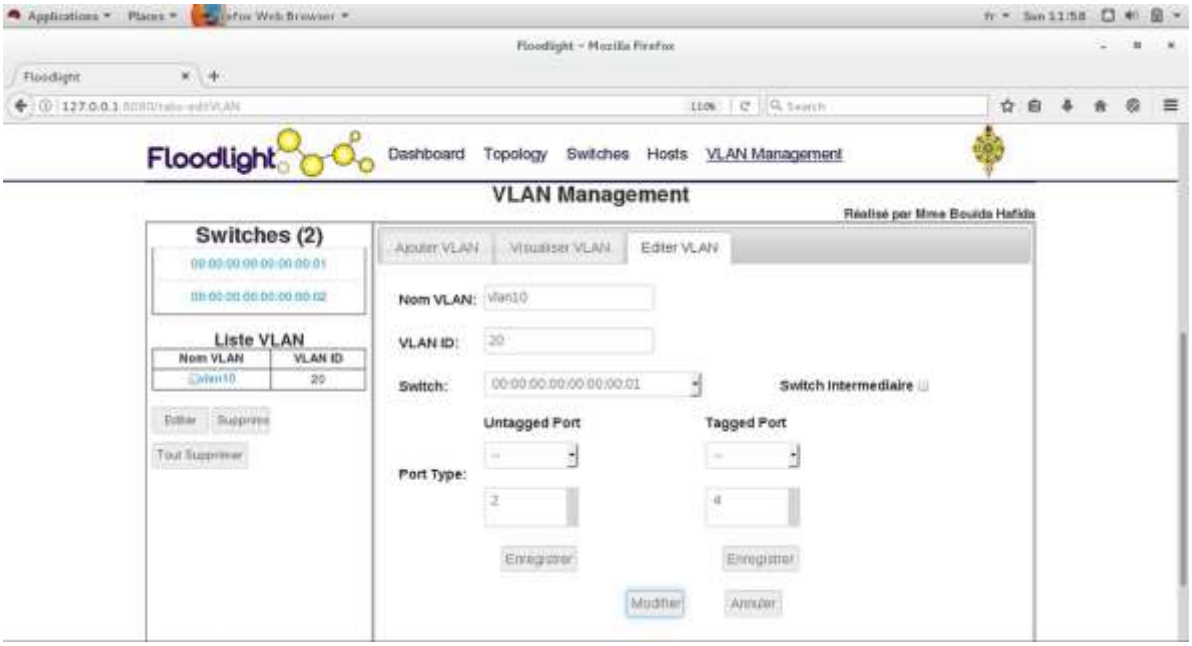


**Figure 5.13: Modification d'un VLAN**

## 5.3/a-4 : Visualisation d'un VLAN:

Cette fonction permet d'afficher tous les réseaux VLAN ainsi que leurs informations. elle fournit également une vue séparée pour afficher plusieurs topologies de réseau VLAN simultanément.

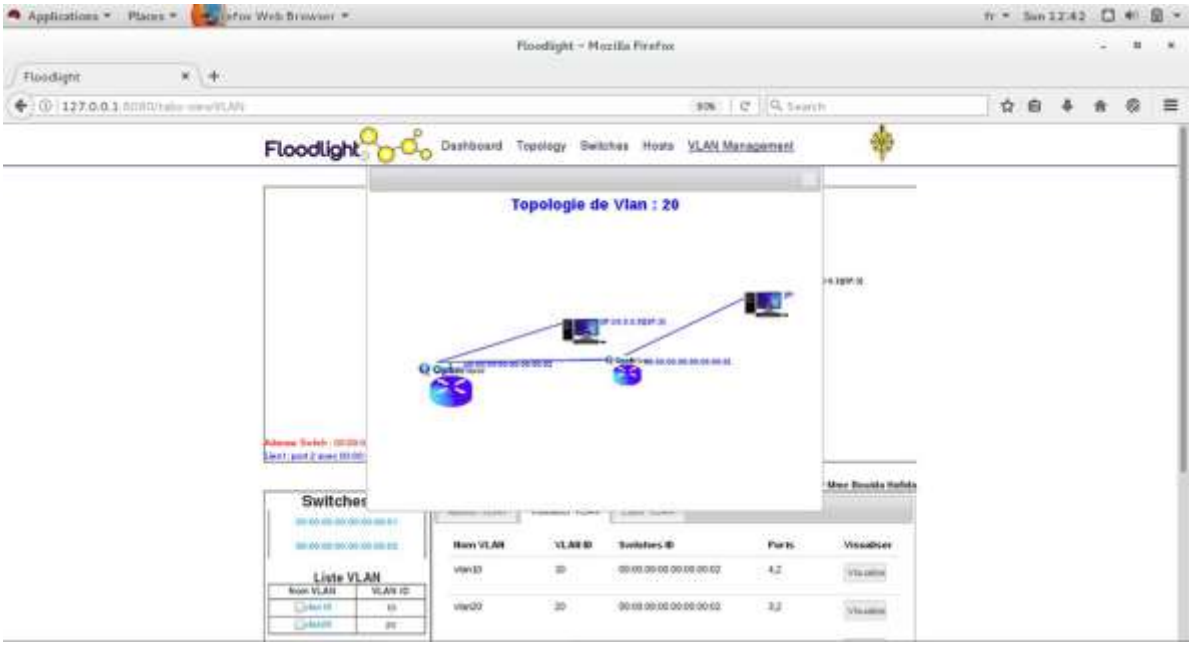


**Figure 5.14: Visualiser un VLAN**

## 5.3/b : Fonctionnement du vlan:

Dans cette section, nous configurons deux VLANs vlan10 et vlan20 de 2 manières différentes :

## 5.3/b-1 : Scenario 1 (Utilisation d'un seul switch):

Une fois la topologie est créé, et le VLAN aussi, nous testons si les hosts appartenant à un seul VLAN (vlan10 ou vlan20) communiquent entre eux, vlan10 se compose du hôte 2 et hôte 3, et vlan20 se compose du hôte 4 et hôte 5.

Alors, nous observons dans la **Figure 5.15**, que h2 peut pinger que h3 (et vice versa) et aussi h4 peut pinger que h5 (et vice versa), nous concluons donc que les 2 VLANs sont fonctionnels.

```
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X
h2 -> X h3 X X
h3 -> X h2 X X
h4 -> X X X h5
h5 -> X X X h4
*** Results: 80% dropped (4/20 received)
mininet>
```

**Figure 5.15: ping entre les hôtes (h2, h3: vlan10 et h4, h5:vlan20)**

## 5.3/b-2 : Scenario 2 (Utilisation de 2 switchs):

Comme le scénario précédent, la **Figure 5.16** montre que les hôtes appartenant à un VLAN donné (h2 de Switch1 et h5 de Switch2: vlan10, h3 de switch1 et h6 de switch2:vlan20) communiquent entre eux, d'où nous concluons que les 2 VLANs sont aussi fonctionnels.

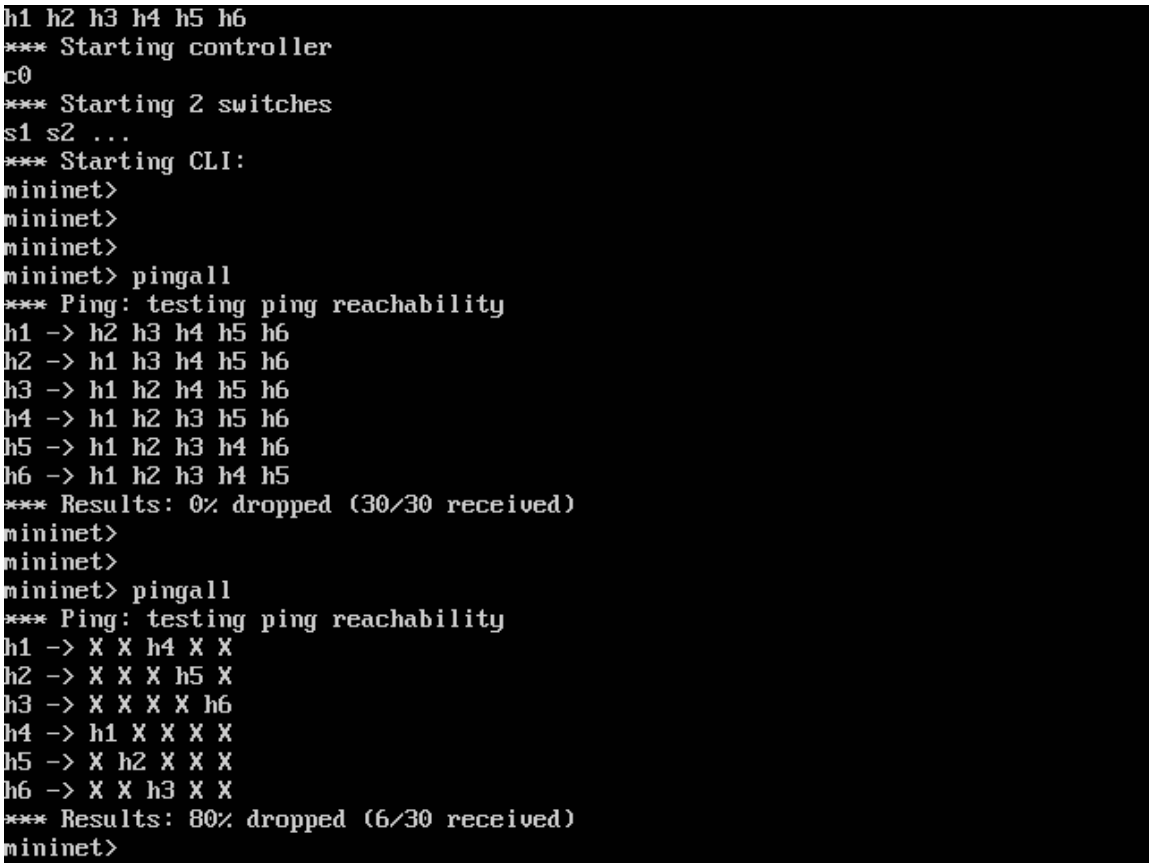


**Figure 5.16: Scenario2, ping entre les hosts (h2, h5: vlan10 et h3, h6:vlan20)**

En résumé, nous pouvons confirmer, à partir des différents tests vu précédemment, que le SDN permet de faciliter la gestion des réseaux.

## 5.3: Conclusion

Dans ce chapitre, nous avons exploité mininet, Floodlight. Notre contribution principale est le développement d'une application de gestion des VLANs dans un environnement SDN. Notre application fournit une interface web interactive qui nous aide à gérer, créer, modifier et aussi visualiser un réseau VLAN, elle offre une flexibilité d'utilisation à un administrateur réseau.

En fournissant une vue topologique séparée pour chaque réseau VLAN, l'administrateur réseau peut surveiller facilement plusieurs réseaux VLAN en même temps. De nos jours, VLAN n'est pas seulement présent sur le campus et les réseaux d'entreprise, mais est également largement utilisé pour gérer un réseau dans le système de cloud computing. Nous croyons que notre système peut être facilement adapté pour être utilise dans cet environnement.