الجـــــــــــــــــمهـوريـــــة الجـــــــــــزائـريـة الديـــــقـراطيـة الـشـــــــعـبيـة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة النـــــــــــعـليـم العـــــالي و البـــــــحث العـــلـمـــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـــــــــــامعة أ بـكـــر بـلـقـايـد – تـلمســـــــــــان –
Université Aboubakr Belkaïd– Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Télécommunications

Spécialité : Réseaux et Télécommunications

Par :

MADOURI Chaima
KECHKECHE Amal

Sujet

Optimisation de placement des VNFs (Virtual Network Functions) dans le Cloud

Soutenu publiquement, le 15 / 09 /2021, devant le jury composé de :

Mr. MERZOUGUI. R	Professeur	Univ. Tlemcen	Président
Mr. ZERROUKI. H	Maitre de Conférences	Univ. Tlemcen	Directeur de mémoire
Mr. HADJILA. M	Maitre de Conférences	Univ. Tlemcen	Examinateur

Année Universitaire **2020/2021**

"Chaque difficulté rencontrée doit être
l'occasion d'un nouveau progrès"

Pierre de Coubertin

DEDICACES

Du plus profond de nos cœurs, nous dédions ce travail à tous ceux qui nous sont chers…

À NOS CHERS PARENTS, Aucune dédicace ne saurait exprimer notre respect, notre amour éternel et notre considération pour les sacrifices que vous avez consenti pour notre instruction et notre bien-être. Nous vous remercions pour tout le soutien inconditionnel, à la fois moral et économique et l'amour que vous portez depuis notre enfance et nous espérons que votre bénédiction nous accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que nous ne vous en acquittions jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais nous ne vous décevions.

À NOS CHERS SŒURS ET FRERES, pour leurs encouragements permanents, etleur soutien moral.

À NOS GRANDS PERES ET GRANDES MERES Qui nous ont accompagnées par leurs prières, leur douceur, puisse Dieu leur prêter longue vie et beaucoup de santé et de bonheur dans les deux vies.

À NOS CHERS oncles, tantes, leurs époux et épouses, à nos chers cousins et cousines ; Veuillez trouver dans ce travail l'expression de notre respect le plus profond et notre affection la plus sincère.

À TOUTES NOS TENDRES FAMILLES pour leurs soutiens tout au long de notre parcours universitaire, que ce travail soit l'accomplissementde vos vœux tant allégués, et le fruit de votre soutien infaillible. Merci d'être toujours là pour nous.

À NOS AMIS qui nous ont apporté leur soutien inestimable tout au long de notre démarche, et qui par leurs encouragements, on a pu surmonter tous les obstacles.

KECHKECHE.A & MADOURI.C

Remerciements

On remercie Allah le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Nous tenons d'abord à remercier très chaleureusement Dr. ZERROUKI Hadj, Maître de conférences à l'université Abou-Bekr Belkaid-Tlemcen, qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigué, la patience, pour son temps, ses bonnes orientations, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre projet de fin d'étude. Nos plus grands respects pour sa personne. Nous sommes conscientes de l'honneur que nous a fait.

Nous adressons aussi nos remerciements les plus distingués à notre chère co-encadreur Mme. FEDAOUCHE Amal, qui sans lui notre travail n'aura jamais pu être possible. Ainsi que Mr. ZELLAT Salah Merci à vous deux pour votre patience, encouragement, disponibilité et surtout votre extrême amabilité malgré vos grandes charges de travail. Nous vous remercions de nous avoir encadrés, orientés, aidés et conseillés. Nous avons eu beaucoup de plaisir de travailler à vos côtés.

Nos remerciements vont aussi au Pr. MERZOUGUI Rachid, Professeur à l'université Abou-Bekr Belkaid-Tlemcen, en étant président du jury et Dr. HADJILA Mourad, Maître de conférences à l'université Abou-Bekr Belkaid-Tlemcen, d'avoir accepté d'examiner ce travail. Nos vifs remerciements pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'juger notre travail et de l'enrichir par leurs propositions.

Nos remerciements s'étendent également à tous nos enseignant(e)s durant les années des études.

Nous souhaitons adresser aussi nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et leur soutien et qui ont contribué de loin ou de près à l'élaboration de ce projet parmi eux Mr.HOUARI, Mr.BENHIBA et Mr.ADELMO.

Enfin, nous tenons à remercier infiniment, nos chers parents, pour leurs contributions, leurs soutiens et leurs patiences au long de nos études.

Résumé

Le concept de virtualisation des fonctions de réseau (NFV) a été récemment introduit comme un nouveau paradigme qui permet d'offrir un certain nombre d'avantages tels qu'une maintenabilité accrue et une réduction des frais généraux de déploiement des équipements réseau.

NFV est souvent complété par le paradigme du Cloud qui est la mise en disposition des ressources d'une infrastructure informatique virtualisée permettant de présenter un ensemble de services aux clients, dont le fournisseur de services Cloud pense principalement à l'augmentation des gains que ses Data Centers pourraient lui engendrer. Il devra donc trouver un moyen efficace pour que ses profits ne soient jamais diminués.

Ainsi, trouver le bon placement des VNF en chaîne de services dans un environnement infonuagique est une étape importante et une problématique majeure pour les fournisseurs de services, tout en minimisant le coût de déploiement.

Mots clés : Réseaux informatiques, informatique en nuage, fonctions de réseau virtuel, Cloud.

Abstract

The concept of Network Functions Virtualization (NFV) has recently been introduced as anew paradigm that provides a number of benefits such as increased maintainability and reduced network equipment deployment overhead.

NFV is often complemented by the cloud paradigm, which is the provisioning of virtualized IT infrastructure resources to present a set of services to customers, with the cloud service provider thinking primarily about the increased revenue that its data centers could generate. Therefore, he will have to find an effective way to ensure that his profits are never diminished.

Thus, finding the right placement of VNFs in a service chain in a cloud environment is an important step and a major issue for service providers, while minimizing the cost of deployment.

Keywords: Computer networks, Cloud computing, virtual network functions, Cloud.

Table des matières

Liste des figures

Liste des tableaux

.

AG	Algorithme Génétique
API	Application Programming Interface
AS	Autonomous système
ASP	Active Server Pages
BGP	Border Gateway Protocol
BGP-LS	Border Gateway Protocol link state
CD	Compact Disque
CGI	Common Gateway Interface
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSP	Fournisseurs de services cloud
CSR1000v	Cloud Services Router
DHCP	Dynamic Host Configuration Protocol
DPI	Deep Packet Inspection
DR	Designated Router
EC2	Elastic Compute Cloud
ESXi	ESX integrated
ETSI	European Telecommunications Standards Institute
FAI	Fournisseur d'Accès Internet
FSA	Fournisseur de service d'applications
GAN	Global Area Network
GA	Genetic Algorithm
GB	Gigabit
GNS3	Graphical Network Simulator
HTTP	HyperText Transfert Protocol
HTTPs	HyperText Transfer Protocol Secure
IaaS	Infrastructure as a service
IDS	Intrusion Detection Systems
ID	Investigation Discovery
IGP	Interior Gateway Protocol
ILP	Integer Linear Programming
IOS	Internetwork Operating System
ISIS	Intermediate system to intermediate system
ISP	Internet Service Provider
IPX/SPX	Internetwork Packet Exchange/Sequenced Packet Exchange
IP	Internet Protocol
JSON	JavaScript Object Notation
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
MAC	Media Access Control
MAN	Metropolitan Area Network
MANO	Management Automation and Network Orchestration
MD-SAL	Model-Driven Service Adaptation Layer
MLS	Multilayer Switch
NACHOS	Network-Aware Chains Orchestration
NAT	Network Address Translation
NF	Network Function
NFV	Network functions virtualization
NFVi	L'infrastructure de virtualisation des fonctions réseau
NFV-O	Network functions virtualization Orchestration

NIST	National Institute of Standards and Technology
NV	Network Virtualization
MIP	Programme Linéaire Mixte
ODL	OpenDayLight
ONAP	Open Network Automation Platform
ONF	Open Network Foundation
ONOS	Open Network Operating System
OPNFV	Open Platform For Network functions virtualization
OSGi	Open Services Gateway initiative
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PaaS	Platform as a Service
PAN	Personal Area Network
PC	Personal Computer
PCC	Path Computation Client
PCE	Path Computation Element
PHP	Hypertext Preprocessor
POP	Point Of Presence
QoS	Quality Of Service
RAM	Random Access Memory
RJ-45	Registered Jack-45
RPC	Remote Procedure Call
SaaS	Software as a Service
SDN	Software Defined Network
SFC	Service Function Chaining
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSC	Single Service Chain
SSD	Solid State Drive
SWAN	Scottish Wide Area Network
SQL	Structured Query Language
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMware	Virtual Machine Software
VNF	Virtual Network Function
VNFI	Virtual Network Function Instances
VNFFG	Virtual Network Function Forwarding Graph
VNF-OP	Virtual Network Function Orchestration Problem
VNF-P	Virtual Network Function Placement
VPN	Virtual Privat Network
WAN	Wide Area Network
WEB	World English Bible (public domain translation)
WWW	World Wide Web
XML	Extensible Markup Language

Introduction générale

Depuis son introduction en tant que concept par l'ETSI pour découpler le logiciel du matériel en tirant parti de la technologie de virtualisation [1], la virtualisation des fonctions de réseau (NFV) a été largement et rapidement adoptée comme un remplacement plus rentable et facile à gérer des boîtes intermédiaires traditionnelles basées sur le matériel. NFV offre de nombreux bénéfices aux fournisseurs de services cloud (CSP), notamment la réduction des coûts des équipements de réseau, la minimisation de la consommation d'énergie, l'évolutivité, l'élasticité, la réutilisation du matériel, la multi-location facile et la configuration rapide de nouveaux services [1].

Ces solutions matérielles monolithiques ne sont pas seulement coûteuses à obtenir et à maintenir. Elles rendent également difficile la réaffectation des fonctions réseau telles que les firewalls, les équilibreurs de charge, les systèmes de détection des intrusions (IDS) et la translation des adresses réseau (NAT). En revanche, NFV permet d'exécuter les fonctions réseau sur des machines virtuelles (VM) hébergées par des serveurs de commodité en tant que fonctions réseau virtuelles (VNF).

Pourtant, elle a introduit de nouveaux défis pour les opérateurs de réseaux tels que trouver le placement le plus approprié de ces VNF dans l'infrastructure du Cloud, enchaîner plusieurs VNFde sorte que l'ordre de la chaîne de services ne soit pas violé, et évaluer l'efficacité en termes de performance.

Le problème du placement des VNF, beaucoup d'efforts ont été faits vers l'évaluation de la capacitédes VNF à traiter de grandes quantités de trafic, notamment avec la croissance du nombre d'utilisateurs finaux adoptant les services Cloud. Un seul VNF pourrait être suffisant pour traiter une certaine quantité de demandes avant d'être surchargé. Cela se produit notamment lorsqu'il y a un afflux de demandes pour un certain service. Prenons l'exemple de Netflix. La majorité des utilisateurs ne regardent peut être pas autant de vidéos sur Netflix pendant les jours ouvrables que pendant les week-ends. Par conséquent, il convient de mentionner que lorsqu'il y a un pic soudain de demandes, une seule VNF de la fonction d'optimisation vidéo ne peut pas suffire à traiter toutes les demandes reçues. Par conséquent, le nombre d'instances VNF doit être estimé au préalable afin d'allouer des ressources aux instances VNF qui peuvent traiter les demandes de manière plus efficace.

Ce mémoire est divisé en quatre chapitres:

Le premier chapitre étant généralités sur les réseaux informatiques, nous présenterons les différents concepts de base, les éléments de la problématique ainsi qu'une généralité sur les réseaux informatiques.

Le deuxième chapitre présente le contexte technique. Ce chapitre est divisé en deux parties. La première partie illustre les concepts de Cloud Computing, de virtualisation, et de NFV, tandis que la deuxième partie de ce chapitre décrit le problème du placement de VNF.

Le troisième chapitre se concentre sur les travaux connexes. Tout au long de ce chapitre, nous discuterons des solutions proposées pour le placement des VNF, puis nous décidons de formuler une synthèse critique des forces et des faiblesses des solutions discutées.

Dans le quatrième chapitre, nous présentons les résultats de l'évaluation des performances de VNF. Enfin, terminer par une conclusion.

Chapitre I : Généralités sur les réseaux informatiques

I.1 Introduction

A l'origine, un réseau était un rassemblement de personnes ou d'objets. De nos jours on entend par réseau, les réseaux d'entreprises, qui connectent différentes machines afin de pouvoir les faire communiquer entre elles. Que ce soit pour le partage de fichiers ou l'envoi des messages, la plupart des entreprises sont aujourd'hui dotées d'un réseau afin d'être plus efficientes (il est quand même plus simple de transférer un fichier par Internet que de l'envoyer sur CD par la poste). La connaissance préalable d'uneinfrastructure réseau et des applications est nécessaire pour acquérir la maitrise globale d'un environnement réseau. Au cours de ce premier chapitre nous allons définir quelques notions sur les réseaux informatiques.

I.2 Que signifie réseau?

Le terme réseau [2] en fonction de son contexte peut désigner plusieurs choses. Il peut désigner l'ensemble des machines, ou l'infrastructure informatique d'une organisation avec les protocoles qui sont utilisés, ce qui est le cas lorsque l'on parle d'Internet.

Le terme réseau peut également être utilisé pour décrire la façon dont les machines d'un site sont interconnectées. C'est le cas lorsque l'on dit que les machines d'un site (sur un réseau local) sont sur un réseau Ethernet, Token Ring, réseau en étoile, réseau en bus.

Le terme réseau peut également être utilisé pour spécifier le protocole qui est utilisé pour que les machines communiquent. On peut parler de réseau TCP/IP, NetBeui (protocole Microsoft) DecNet (protocole DEC), IPX/SPX,...

I.3 Objectifs et utilisation des réseaux

Les réseaux ont été et sont toujours développés pour un certain nombre de raisons. Il y en a en fait 3 objectifs principales [2]:

I.3.1 Partage des ressources

Les réseaux permettent de rendre accessible un certain nombre de ressources (logiciels, bases de données, imprimantes...) indépendamment de la localisation géographique des utilisateurs.

Le partage des données commerciales d'une entreprise en est une illustration : chaque employé d'une multinationale peut accéder aux derniers comptes de résultat de l'entreprise.

I.3.2 Augmentation de la fiabilité et des performances

Les réseaux permettent par exemple de dupliquer en plusieurs endroits les fichiers vitaux d'un projet, d'une entreprise ; en cas de problème, la copie de sauvegarde est immédiatement disponible. L'augmentation des performances vient également du fait qu'il est relativement facile d'augmenter les performances d'un système en réseau en ajoutant tout simplement un ou deux autres ordinateurs supplémentaires.

Ce dernier point associé à un constat économique (voir objectif suivant) rend presque obsolètes les grosses installations.

I.3.3 Accès à l'information et au courrier

Avec les réseaux et en particulier Internet, il est très facile de s'informer sur toute sorte de sujet très rapidement. Ce dernier objectif joue en fait un rôle capital dans l'utilisation que les gens ont des réseaux.C'est peut-être même l'utilisation principale aujourd'hui.

I.4 Les différents types des réseaux

On peut faire une classification des réseaux selon leur taille (en terme de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue comme on peut le voir dans la figure I.1.

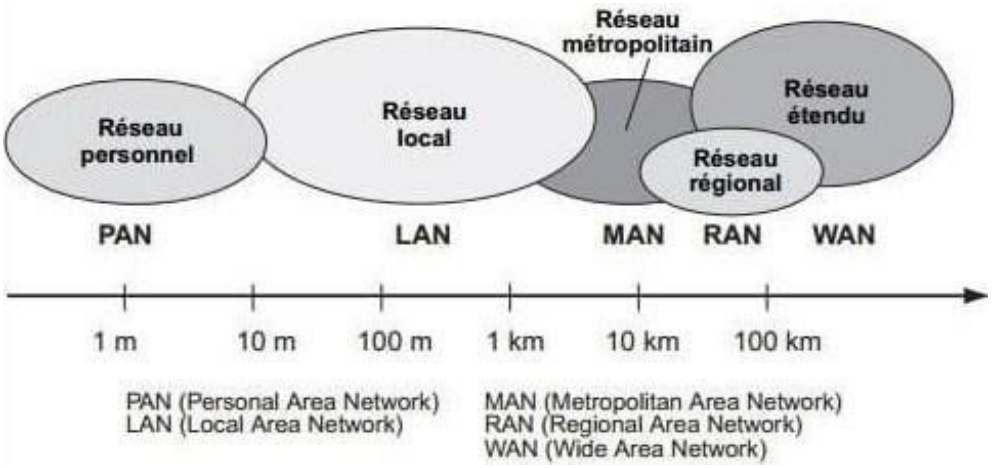


Figure I.1 : *Classification des réseaux filaires.*

I.4.1 Les réseaux PAN (*Personal Area Network*)

Un réseau personnel [3] est une connexion souvent sans fil des équipements personnel comme un ordinateur portable, un agenda électronique etc. (Figure I.2).

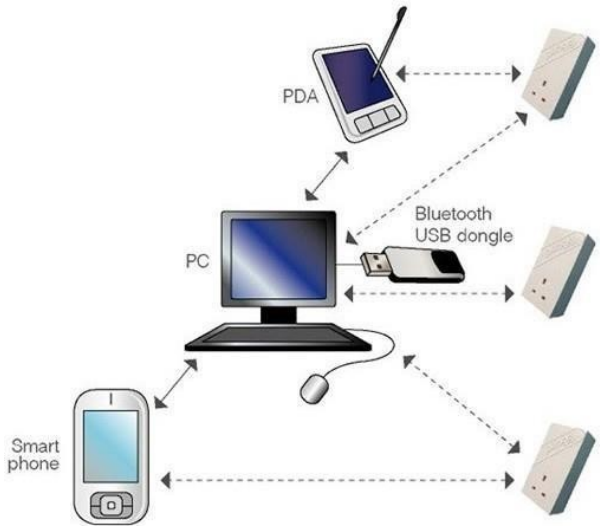


Figure I.2 : *Exemple d'un réseau PAN.*

I.4.2 Les réseaux LAN (*Local Area Network*)

Le LAN (*Local Area Network*) est le réseau local d'une organisation (Figure II.3). Il est dans un espace restreint. Chez les particuliers, les appareils interconnectés avec le box forment aussi un LAN.

Figure I.3 : *Exemple d'un réseau LAN.*

I.4.3 Les réseaux MAN (*Metropolitan Area Network*)

C'est un LAN étendu de quelques kilomètres souvent à l'échelle d'une ville. Un campus universitaire souvent larges de plusieurs kilomètres peut aussi être considéré comme un MAN (FigureI.4). Par exemple, une mairie peut voir des bâtiments éparpillés dans la ville. Elle peut alors tirer une ligne pour relier ses réseaux au LAN principal se trouvant dans l'hôtel de ville.

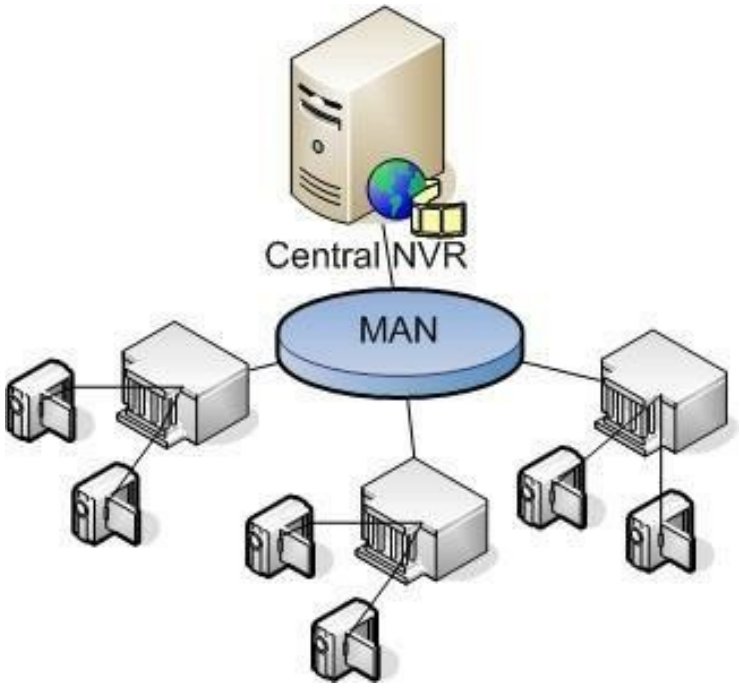


Figure I.4 : *Exemple d'un réseau MAN.*

Un MAN interconnecte généralement un certain nombre de réseaux locaux (LAN) à l'aide d'une technologie de dorsale haute capacité, telle que des liaisons à fibre optique, et fournit des services de liaison montante aux réseaux étendus (ou WAN) et à Internet.

I.4.4 Les réseaux WAN (*Wide Area Network*)

Les WAN pour réseau étendu sont des réseaux à l'échelle d'une région, pays ou continent. Ils couvrent donc de très vastes étendues comme l'interconnexion des régions, d'un pays ou continent.Une entreprise nationale qui possède des bureaux dans des différents endroits utilise un WAN. Un fournisseur d'accès national exploit un WAN puisqu'il est présent dans tout les pays.

Dans un réseau étendu WAN (Figure I.5), l'utilisateur ou l'organisation ne possède pas les liaisons ou l'infrastructure de communication qui connectent le système informatique distant. Au lieu de cela, un tel service est fourni par un fournisseur de services des télécommunications ou Internet.

Les WAN transmettent les données à des vitesses plus lentes que les LAN. Ils sont similaires dans leur structure à un MAN, mais sa gamme de services couvre des distances supérieures à cinquante kilomètres.

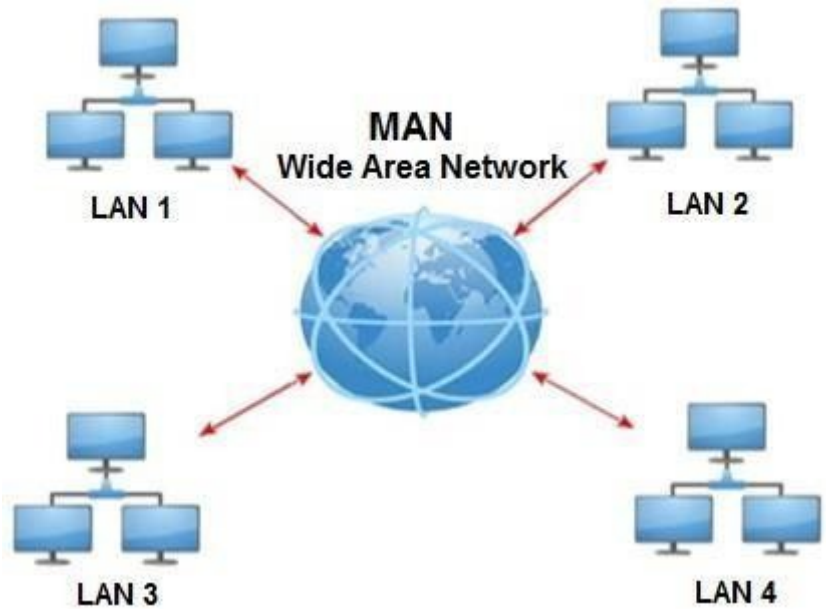


Figure I.5 : *Exemple d'un réseau WAN.*

I.4.5 Les réseaux GAN (*Global Area Network*)

Un réseau mondial (GAN) est un type de réseau composé de différents réseaux interconnectés qui couvrent une zone géographique illimitée. Le terme est vaguement synonyme d'Internet, qui est considéré comme un réseau mondial.

Il peut se composer de l'interconnexion de WAN qui peut se faire via des connexions par un lien satellitaire ou des backbones (câbles sous-marins). La figure suivante schématise un exemple d'un réseau global GAN (Figure I.6).

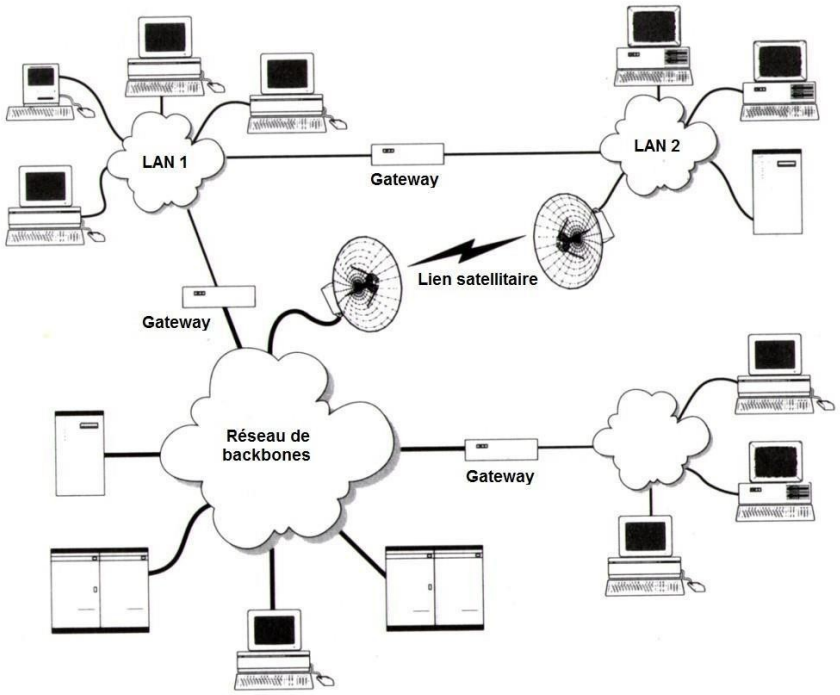


Figure I.6 : *Exemple d'un réseau GAN.*

I.4.6 Les réseaux VPN (*Virtual Privat Network*)

Un VPN (*Virtual Privat Network*) ou réseau privé virtuel est un réseau de communication virtuel qui utilise l'infrastructure d'un réseau physique pour relier logiquement les systèmes informatiques (Figure I.7). Il peut s'agir de n'importe quel type de réseau détaillé plus haut, cependant Internet est le moyen de transmission le plus couramment utilisé. Cela relie presque tous les ordinateurs dans le monde entieret reste disponible gratuitement par opposition à l'exploitation privé d'un MAN ou WAN. Les données sont transférées au sein d'un tunnel virtuel qui est construit entre un client VPN et un serveur VPN.

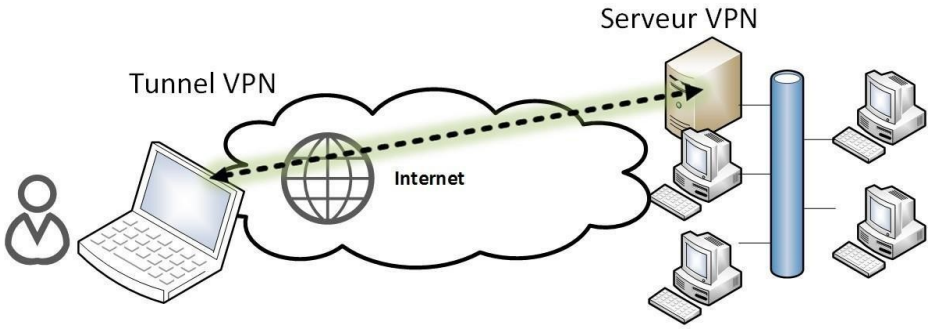


Figure I.7 : *Exemple d'un réseau VPN.*

Le réseau public est utilisé comme moyen de transport, les réseaux privés virtuels sont généralement cryptés pour s'assurer de la confidentialité des données. Les VPN sont utilisés pour connecter les réseaux locaux sur Internet ou pour permettre l'accès à distance à un réseau ou à un seul ordinateur via la connexion publique.

I.4.7 Les réseaux VLAN (Virtuel Local Network)

Le VLAN divise un réseau local en plusieurs réseaux locaux logiques, chacun étant un domaine de diffusion. Les hôtes d'un même VLAN peuvent communiquer entre eux comme dans un réseau local. Cependant, les hôtes de différents VLAN ne peuvent pas communiquer directement. De cette Ainsi, les paquets de diffusion sont limités à un seul VLAN, comme le montre (FigureI.8).

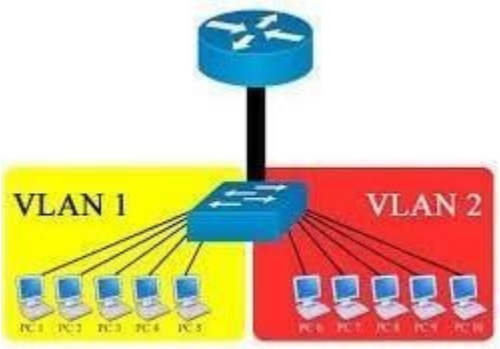


Figure I.8 : *Exemple d'un réseau VLAN.*

La technologie VLAN [4] présente les avantages suivants :

- Le trafic de diffusion est confiné à chaque VLAN, ce qui réduit l'utilisation de la bande passante et améliore les performances du réseau.
- La sécurité du réseau local est améliorée. Les paquets de différents VLAN ne peuvent pas communiquer directement entre eux. Autrement dit, les utilisateurs d'un VLAN ne peuvent pas interagir directement avec ceux des d'autres VLAN, à moins d'utiliser des routeurs ou des commutateurs de couche 3.
- Un moyen plus souple d'établir des groupes de travail virtuels. Avec la technologie VLAN, les clients peuvent être attribués à différents groupes de travail, et les utilisateurs du même groupe n'ont pas besoin de se trouver dans la même zone physique, ce qui facilite la construction et la maintenance du réseau beaucoup plus faciles et plus flexibles.

I.5 Les différentes catégories des réseaux

On travaille sur des ordinateurs pendant longtemps, vous avez peut-être entendu les termes Client-Serveur et Peer-to-Peer. Ces deux modèles de réseau sont communs que nous utilisons dans notre vie de tous les jours. L'architecture Client-Serveur se concentre sur le partage d'informations, tandis que l'architecture Peer-to-Peer se concentre sur la connectivité aux ordinateurs distants.

I.5.1 Réseau client-serveur

Le modèle de réseau client-serveur est un modèle qui est largement utilisé. Ici, Serveur est un système puissant qui stocke les données ou les informations qu'il contient. Tandis que, le client est la machine qui permet aux utilisateurs d'accéder aux données sur le serveur distant.

Comme tous les services sont fournis par un serveur centralisé, il peut y avoir des risques de plantage sur le serveur, ce qui ralentit l'efficacité du système. La figure suivante montre un exemple simple d'un réseau client-serveur (FigureI.9).

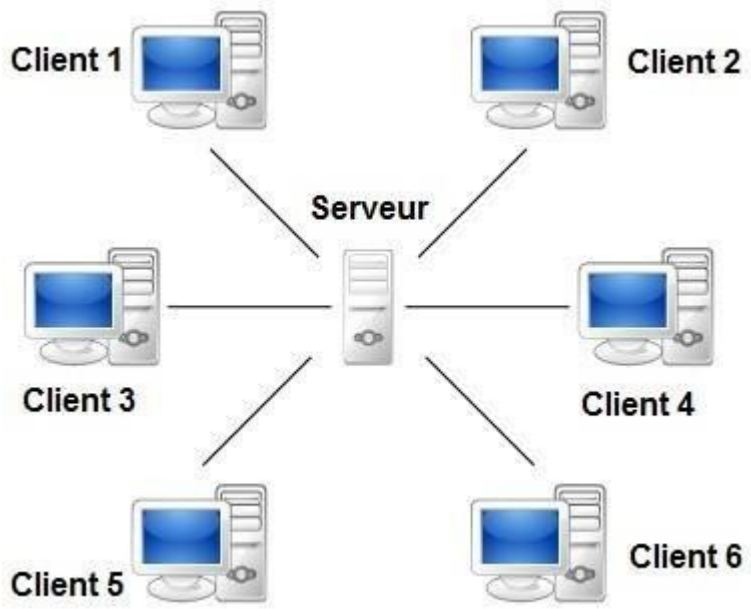


Figure I.9 : *Exemple d'un réseau Client-Serveur.*

I.5.2 Réseau Peer-to-Peer

Contrairement à l'architecture Client-Serveur, le modèle Peer-to-Peer (Figure I.10) ne se distingue pas entre le modèle client-serveur, mais chaque nœud peut être un client ou un serveur selon le nœud est-ce qu'il demande ou fournit le service. Puisque chaque nœud est considéré comme un pair.

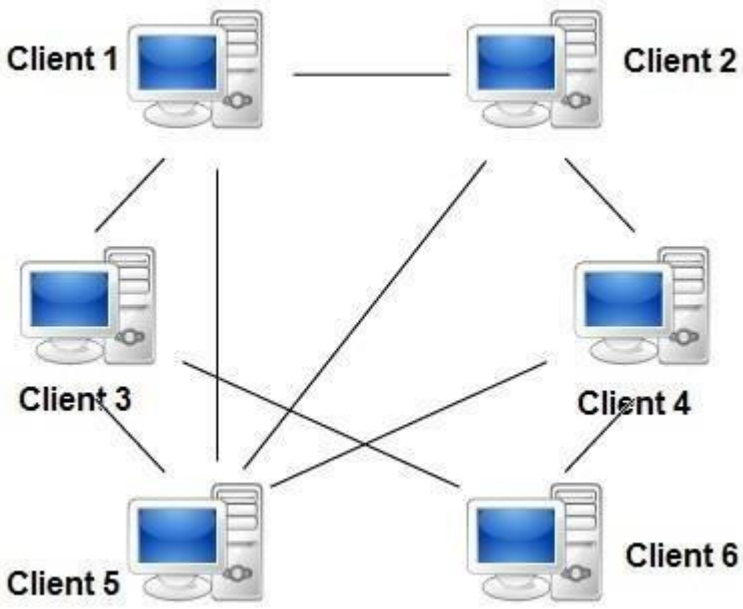


Figure I.10 : *Exemple d'un réseau Peer-to-Peer.*

Le réseau Peer-to-Peer a l'avantage par rapport au réseau client-serveur est que le serveur n'est pas encombré car les services sont fournis par plusieurs nœuds distribués dans un système peer-to-peer.

I.6 Les topologies des réseaux

La topologie du réseau [5] et les emplacements relatifs de source et de destination des flux de trafic sur le réseau déterminent le chemin optimal pour chaque flux et la mesure dans laquelle des options redondantes de routage existent en cas de défaillance. Il existe deux manières pour définir un réseau : la topologie physique et la topologie logique (ou signal).

La topologie physique consiste à définir des périphériques réseau et du câblage. La topologie logique définit le mode de transfert des données. Selon les topologies on obtient des performances différentes. A savoir, les débits, le nombre d'utilisateur maximum, le temps d'accès, la tolérance aux pannes, la longueur de câblage et le types d'applications seront différents.

Types de topologies des réseaux:

I.6.1 Topologie réseau en étoile

Une topologie en étoile (Figure I.11) [5] est une topologie dans laquelle tous les nœuds sont connectés à un périphérique central, formant ainsi une étoile. Deux types de périphériques fournissant un point de connexion central commun aux nœuds du réseau sont un Hub et un Switch.

Pour ce type de réseau, les câbles pairs torsadés sont le plus souvent utilisés.

- **Avantages**

  - Gestion centralisée du réseau, via l'utilisation du périphérique central (hub/commutateur).
  - Facile d'ajouter un autre ordinateur au réseau.
  - Si un ordinateur du réseau tombe en panne, le reste du réseau continue de fonctionner normalement.

Figure I.11 : *Exemple d'une topologie en étoile.*

- **Inconvénients**

  - L'implémentation peut avoir un coût plus élevé, notamment si vous utilisez un commutateur ou un routeur comme périphérique central.

- Le périphérique réseau central détermine les performances et le nombre de nœuds que le réseau peut gérer.
- Si l'ordinateur central, Switch ou Hub tombe en panne, tout le réseau tombe en panne et tous les ordinateurs sont déconnectés du réseau.

I.6.2 Topologie réseau en bus

Une topologie en bus (Figure I.12) [5] est une configuration réseau dans laquelle chaque ordinateur et chaque périphérique réseau sont connectés à un seul câble ou à un réseau fédérateur. Selon le type de carte réseau utilisé dans chaque ordinateur de la topologie en bus, un câble coaxial ou un câble réseau RJ-45 est utilisé pour les connecter ensemble.

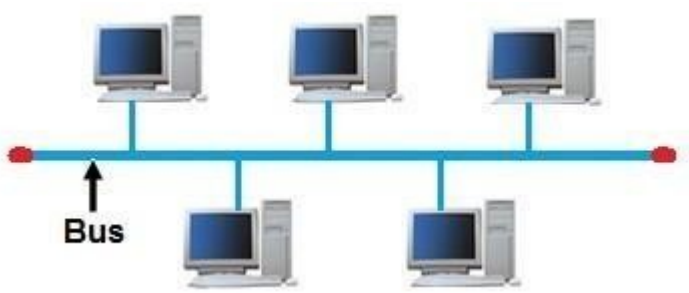


Figure I.12 : *Exemple d'une topologie en bus.*

➢ **Avantages**

- Cela fonctionne bien lorsque vous avez un petit réseau.
- C'est la topologie de réseau la plus simple pour connecter des ordinateurs ou des périphériques de manière linéaire.
- Il nécessite moins de câble par rapport à la topologie en étoile.

➢ **Inconvénients**

- Il peut être difficile d'identifier les problèmes si tout le réseau tombe en panne.
- Il peut être difficile de résoudre les problèmes de périphériques individuels.
- La topologie en bus n'est pas idéale pour les grands réseaux.
- Des terminaisons sont requises pour les deux extrémités du câble principal.
- Des périphériques supplémentaires ralentissent le réseau.
- Si un câble principal est endommagé, le réseau tombe en panne.

I.6.3 Topologie réseau en anneau

Sur un réseau en anneau [5], un câble forme une boucle fermée (anneau) avec tous les ordinateurs disposés tout au long de l'anneau, comme montre la figure I.13. Les données transmises sur un réseau en anneau circulent d'un périphérique à l'autre autour de la totalité de l'anneau, dans une seule direction.

Lorsqu'un ordinateur ou un périphérique envoie des données, les données sont transmises à chaque ordinateur de l'anneau jusqu'à ce qu'il atteigne sa destination. Si un ordinateur ou un périphérique sur un réseau en anneau tombe en panne, l'ensemble du réseau peut potentiellement cesser de fonctionner. Un réseau en anneau peut couvrir une plus grande distance qu'un réseau en bus, mais il est plus difficile à installer. La topologie en anneau est principalement utilisée pour les réseaux locaux, mais elle est également utilisée dans les réseaux étendus.

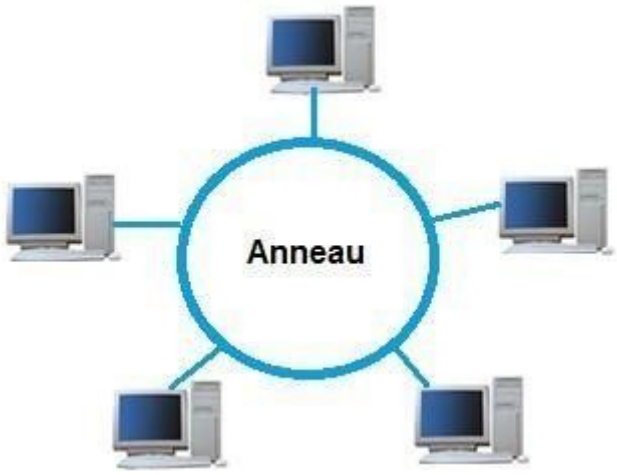


Figure I.13 : *Exemple d'une topologie en anneau.*

Il existe deux types de topologie en anneau basés sur le flux de données ; Unidirectionnel et Bidirectionnel [5].

Une topologie en anneau unidirectionnelle gère le trafic de données dans le sens des aiguilles d'une montre ou dans le sens contraire. Ce réseau de données peut également être appelé un réseau semi-duplex. Une topologie en anneau unidirectionnelle est donc facile à maintenir par rapport à la topologie en anneau bidirectionnelle.

- **Avantages**

- Risques réduits de collision des données car chaque nœud libère un paquet de données après la réception du jeton.
- Le passage de jetons améliore la performance de la topologie en anneau par rapport à la topologie en bus lorsque le trafic est dense.
- Pas besoin de serveur pour contrôler la connectivité entre les nœuds.
- Egalité d'accès aux ressources.

- **Inconvénients**

  - Dans la topologie en anneau unidirectionnelle, un paquet de données doit passer par tous les nœuds. Exp: Disons que A, B, C, D et E sont des nœuds qui font partie du réseau en anneau. Le flux de données va de A vers B et désormais. Dans cette condition, si E veut envoyer un paquet à D, le paquet doit traverser tout le réseau pour atteindre D.

- Seul point de défaillance, cela signifie que si un nœud tombe en panne, tout le réseau tombe en panne.

I.6.4 Topologie de réseau maillée

Une topologie maillée (Figure I.14) [5] est une topologie de réseau dans laquelle tous les nœuds de réseau sont connectés les uns avec les autres. Il n'existe pas de concept de commutateur (Switch) central, de hub ou d'ordinateur qui serve de point de communication central pour la transmission des messages.

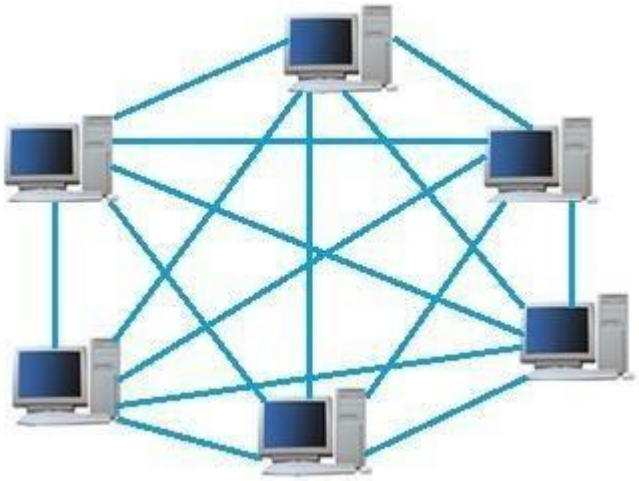


Figure I.14 : *Exemple d'une topologie maillée.*

Contrairement aux autres topologies de réseau, elles peuvent être divisées en deux types ; Topologie maillée entièrement connectée et Topologie maillée partiellement connectée.

Dans la topologie maillée entièrement connectée, tous les nœuds sont connectés les uns aux autres. Si vous connaissez la théorie des graphes, alors il s'agit d'un graphe entièrement connecté dans lequel tous les nœuds sont connectés à tous les autres nœuds.

Tandis qu'une topologie maillée partiellement connectée n'a pas tous les nœuds connectés les uns aux autres.

- **Avantages**

- Chaque connexion peut porter sa propre charge de données.
- Il est robust.
- Une faute est diagnostiquée facilement.
- Assure la sécurité et la confidentialité.

➢ **Inconvénients**

- L'installation et la configuration sont difficiles si la connectivité devient plus importante.
- Le coût de câblage est de plus en plus élevé dans le cas d'une topologie maillée entièrement connectée.
- Le câblage en masse est requis.

I.7 Les équipements d'interconnexions de réseau

I.7.1 Le répéteur (*Repeater*)

Un répéteur (Figure II.15) [5] est un périphérique réseau utilisé pour régénérer ou répliquer un signal. Les répéteurs sont utilisés dans les systèmes de transmission pour régénérer les signaux analogiques ou numériques déformés par une perte de transmission. Les répéteurs analogiques peuvent uniquement amplifier le signal, tandis que les répéteurs numériques peuvent reconstituer un signal à sa qualité d'origine.

Dans un réseau de données, un répéteur peut relayer des messages entre sous-réseaux. Les concentrateurs (Hub) peuvent fonctionner comme des répéteurs en transmettant des messages à tous les ordinateurs connectés. Un répéteur n'est pas intelligent comme les ponts (Bridge) et les routeurs.

Figure I.15 : *Exemple d'un répéteur (Repeater).*

I.7.2 Le pont (*Bridge*)

Un pont [5] est un type de périphérique réseau qui assure l'interconnexion avec d'autres réseaux utilisant le même protocole. Le pont fonctionnent au niveau de la couche liaison de données du modèle OSI (*Open System Interconnect*), connectant deux réseaux différents ensemble et assurant la communication entre eux.

Les ponts (Figure I.16) [5] sont similaires aux répéteurs et aux concentrateurs (Hub) puisqu'ils transmettent des données à chaque nœud. Pourtant, les ponts conservent la table d'adresses *MAC (Media Access Control*) dès qu'ils découvrent de nouveaux segments. Les transmissions ultérieures ne sont alors envoyées qu'au destinataire souhaité.

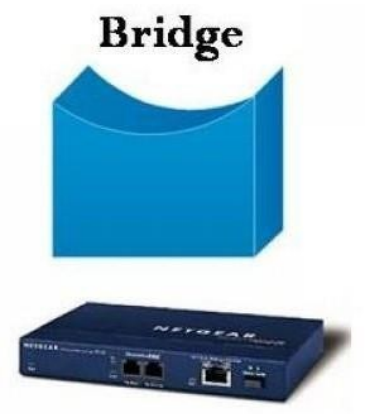


Figure I.16 : *Exemple d'un pont (Bridge).*

I.7.3 Le concentrateur (*Hub*)

Les concentrateurs de réseau (Figure I.17) [5] conviennent parfaitement aux environnements de réseau local (LAN) de petite taille et simples. Ils ne peuvent pas fournir de capacités de routage ni d'autres services réseau avancés. Puisqu'ils fonctionnent en transmettant des paquets sur tous les ports sans distinction.

Figure I.17 : *Exemple d'un concentrateur (Hub).*

Un concentrateur envoie des paquets de donnée (trames) à tous les périphériques d'un réseau, quelles que soient les adresses MAC contenues dans le paquet de données. La plupart des concentrateurs peuvent détecter des erreurs réseau de base telles que des collisions, mais le fait de diffuser toutes les informations sur plusieurs ports peut être un risque pour la sécurité et causer des goulots d'étranglement.

I.7.4 Le commutateur (*Switch*)

Un commutateur (Figure I.18) [5], dans le contexte réseau, est un périphérique haut vitesse qui reçoit les paquets de données entrants et les redirige vers leur destination sur un réseau local (LAN). Un commutateur LAN fonctionne au niveau de la couche liaison de données (couche 2) ou la couche réseau du modèle OSI et peut prendre en charge tous les types de protocoles de paquets.

Les commutateurs sont similaires aux hubs, mais plus intelligents. Un commutateur, crée un tunnel entre les ports source et destination pendant une fraction de seconde qu'aucun autre trafic ne peut entrer,

cela se traduit par une communication sans collision. Les commutateurs sont généralement les agents de la circulation d'un simple réseau local.

Figure I.18 : *Exemple d'un commutateur (Switch).*

Les commutateurs sont plus avancés que les concentrateurs (Hub) et moins capables que les routeurs. Il existe plusieurs types de commutateurs dans la mise en réseau:

- Les **commutateurs virtuels** sont des commutateurs logiciels uniquement instanciés dans des environnements d'hébergement de machines virtuelles.
- Un **commutateur de routage** connecte les réseaux locaux; en plus de la commutation de couche 2 basée sur MAC, il peut également exécuter des fonctions de routage au niveau de la couche 3de l'OSI (la couche réseau) en dirigeant le trafic en fonction de l'adresse IP (Internet Protocol) de chaque paquet.

I.7.5 Le routeur (*Router*)

Un routeur [5] (FigureI.19) est un périphérique matériel conçu pour recevoir, analyser et déplacer les paquets entrants vers un autre réseau. Il peut également être utilisé pour convertir les paquets en une autre interfaceréseau, les supprimer et effectuer d'autres actions relatives à un réseau.

Figure I.19 : *Exemple d'un routeur (Router).*

Les routeurs peuvent analyser les données envoyées sur un réseau, modifier la manière dont elles sont empaquetées et les envoyer vers un autre réseau ou via un autre réseau. Par exemple, les routeurs sont couramment utilisés dans les réseaux domestiques pour partager une seule connexion Internet entre plusieurs ordinateurs.

I.7.6 Le pare-feu (*Firewall*)

Les pare-feu (Figure I.20) vous protègent de l'Internet, appliquez certaines restrictions à votre réseau local.

Figure I.20 : *Exemple d'un pare-feu (Firewall).*

I.8 Découpage fonctionnel des réseaux

Fait allusion aux différentes fonctionnalités qu'ils offrent ou la technologie utilisée pour leur fonctionnement. Ainsi, selon les technologies utilisées, nous avons:

- Intranet
- Extranet
- Internet

La FigureI.21 montre la relation entre Intranet, Extranet et Internet.

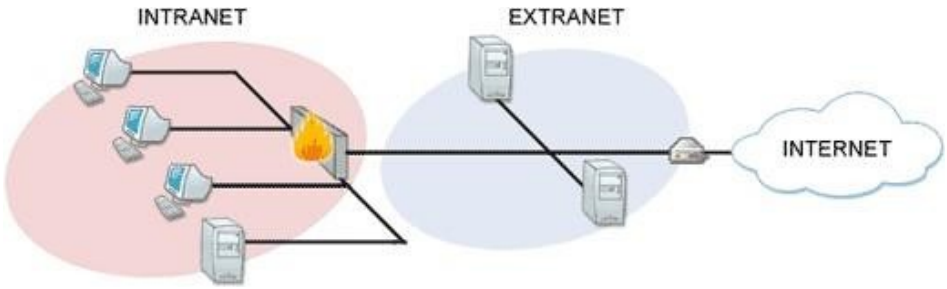


Figure I.21 : *Relation entre Intranet, Extranet et Internet.*

I.8.1 Intranet

Un intranet [5] est un ensemble de services internet (par exemple un serveur web) interne à un réseau local, c'est-à-dire accessible uniquement à partir des postes d'un réseau local, ou bien d'un ensemble

de réseaux bien définis, et invisible de l'extérieur. Il consiste à utiliser les standards client- serveur de l'internet (en utilisant les protocoles TCP/IP), comme par exemple l'utilisation de navigateurs internet (client basé sur le protocole HTTP) et des serveurs web (protocole HTTP), pour réaliser un système d'information interne à une organisation ou une entreprise.

Un intranet repose généralement sur une architecture à trois niveaux (Figure I.22), composée :

- De clients (navigateur internet généralement).
- D'un ou plusieurs serveurs d'application (middleware): un serveur web permettant d'interpréter des scripts CGI, PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données
- D'un serveur de bases de données

De cette façon les machines clientes gèrent l'interface graphique, tandis que le serveur manipule les données. Le réseau permet de véhiculer les requêtes et les réponses

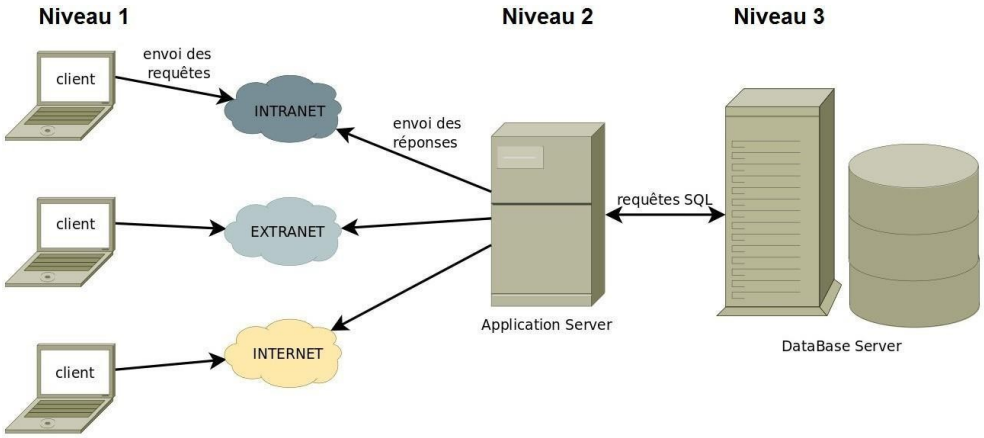


Figure I.22 : *Architecture Intranet à trois niveaux.*

Un intranet [5] possède naturellement plusieurs clients (les ordinateurs du réseau local) et peut aussi être composé de plusieurs serveurs. Une grande entreprise peut par exemple posséder un serveur web pour chaque service afin de fournir un intranet composé d'un serveur web fédérateur liant les différents serveurs gérés par chaque service.

Un intranet permet de constituer un système d'information à faible coût (concrètement le coût d'un intranet peut très bien se réduire au coût du matériel, de son entretien et de sa mise à jour, avec des postes clients fonctionnant avec des navigateurs gratuits, un serveur fonctionnant sous Linux avec le serveur web Apache et le serveur de bases de données MySQL).

D'autre part, étant donné la nature "Universelle" des moyens mis en jeu, n'importe quel type de machine peut être connecté au réseau local, donc à l'intranet.

I.8.2 Extranet

Un extranet est une extension du système d'information de l'entreprise à des partenaires situés au-delà du réseau.

L'accès à l'extranet doit être sécurisé dans la mesure où cela offre un accès au système d'information à des personnes situées en dehors de l'entreprise. Il peut s'agir soit d'une authentification simple (authentification par nom d'utilisateur et mot de passe) ou d'une authentification forte (authentification à l'aide d'un certificat). Il est conseillé d'utiliser HTTPS pour toutes les pages web consultées depuis l'extérieur.

De cette façon, un extranet n'est ni un intranet, ni un site internet, il s'agit d'un système supplémentaire offrant par exemple aux clients d'une entreprise, à ses partenaires ou à des filiales, un accès privilégié à certaines ressources informatiques de l'entreprise par l'intermédiaire d'une interface Web.

I.8.3 Internet

Internet [6] est le réseau informatique mondial qui rend accessible au public des services comme le courrier électronique et le World Wide Web. Techniquement, Internet se définit comme le réseau public mondial utilisant le protocole de communication IP (Internet Protocole). Le Web, le courrier électronique, la messagerie instantanée et les systèmes de partage de fichiers posteà poste sont des applications d'Internet.

L'internet est composé d'une multitude de réseaux répartis dans le monde entier. Chaque réseau est rattaché à une entité propre (université, fournisseur d'accès à Internet, armée) et se voit attribué un identifiant unique appelé *Autonomous système* (AS).

Afin de pouvoir communiquer entre eux, les réseaux s'échangent des données, soit en établissant une liaison directe, soit en se rattachant à un nœud d'échange (point de peering).

Chaque réseau est donc connecté à plusieurs autres réseaux. Lorsqu'une communication doit s'établir entre deux ordinateurs appartenant à des systèmes autonomes différents, il faut alors déterminerle chemin à effectuer parmi les réseaux. Aucun élément ne connaît le réseau dans l'ensemble, les données sont simplement redirigées vers un autre nœud selon des règles de routage. On conclue:

- **Intranet** : c'est le réseau interne de l'entreprise. Souvent, on trouve un site internet qui donne des informations sur la vie de l'entreprise, un annuaire, etc.
- **Extranet** : c'est la partie du réseau externe à l'entreprise qui peut communiquer avec d'autres réseaux. On trouve souvent des services WEB avec des API ou des parties WEB privées pour consulter des données internes mises à disposition pour les partenaires.
- **Internet** : toute la partie externe du réseau de l'entreprise, interconnectée dans internet.

I.9 L'avenir des réseaux

Les réseaux et toutes les technologies environnantes sont en pleines expansions. Augmentation de la bande passante, de plus en plus d'utilisateurs, autant d'éléments motivant les entreprises dans la réalisation de solutions techniques innovantes. Le but n'est plus de proposer un moyen de connecter les gens, il est de fournir la meilleure connexion et le meilleur service possibles au moindre coût [2].

La virtualisation est devenue une solution d'entreprise qui permet de réduire le nombre des serveurs physiques. Mais, par contre, elle permet d'augmenter conséquemment le nombre des serveurs virtuels sur chaque serveur physique, en vue d'optimiser son utilisation, de réduire les dépenses sur le matériel serveur, de diminuer la consommation électrique ainsi que de libérer beaucoup d'espace dans la salle serveur en facilitant l'administration du système informatique.

I.10 Conclusion

Dans ce chapitre, nous avons présenté quelques notions sur les réseaux informatiques, telle que les différents types de réseaux, les différents catégories des réseaux et les topologies des réseaux, nous avons également énuméré les avantages et les inconvénients de chaque topologie et finalement introduit les équipements d'interconnexions de réseaux et le découpage fonctionnel des réseaux. Dans le chapitre suivant nous allons décrire les concepts de Cloud Computing, de virtualisation, et de NFV et expliquer la nécessité de virtualiser les serveurs.

Chapitre II : Définitions et Concepts de bases de cloud computing

II.1 Introduction

Tout au long de ce chapitre, nous mettrons en évidence certains concepts clés concernant la virtualisation, les VMs (*Virtual Machine*), la virtualisation des réseaux et les NFV (*Network Functions Virtualization*). Nous réaliserons également certains des avantages et inconvénients de chacune de ces technologies. En outre, nous discuterons du problème de VNF et introduirons sa correspondance avec le concept de SDN (*Software Defined Networking*). Avant tout ça, nous allons définiet fournir quelques concepts clés sur le *Cloud Computing*.

II.2 Le Cloud Computing

II.2.1 Définition

Plusieurs approches de définitions du *Cloud Computing* ont été proposées par différents auteurs. Mais la définition la plus couramment adoptée est celle du *National Institute of Standard and Technology* (NIST) [7] formulée en ces termes: "*Cloud Computing is a model for enabling ubiquitous, convinient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or provider interaction*". Autrement dit, Le Cloud Computing est un modèle permettant d'accéder à travers le réseau, à la demande, à un ensemble de ressources informatiques partagées et configurables (les réseaux, les serveurs, des capacités de stockage, des applications et les services), qui sont rapidement mobilisables et libérables avec le moindre effort d'administration ou d'intervention du fournisseur.

A partir de cette définition, nous constatons que le Cloud Computing est basé sur la technologie de virtualisation, qui consiste à partager des ressources informatiques configurables d'un *Data Center*. A titre d'exemple, les réseaux, les serveurs, stockage de base de données, puissance de calcul, d'applications, etc. Le Cloud fournit aussi à ses clients des services à la demande, instantanés et élastiques avec facturation de ce qui a consommé.

II.2.2 Les caractéristiques du Cloud Computing

Selon l'organisme NIST, le Cloud Computing est caractérisé par les propriétés suivantes:

- Service à la demande (*On-demand self-service*) : Il permet de fournir des ressources Cloud à la demande, à chaque fois que le besoin se fait sentir. L'utilisateur accède à ces services à travers un panneau de commande en ligne, en utilisant par exemple un portail Web et une interface de gestion.
- Accessibilité à l'ensemble du réseau (Broad network Access) : Les ressources sont disponibles à travers le réseau et accessibles par des plateformes hétérogènes de clients lourds ou légers.
- Mutualisation des ressources (*Resource pooling*) : Les ressources de plusieurs fournisseurs sont rassemblées et mis à la disposition de plusieurs consommateurs, avec différentes ressources physiques et virtuelles attribuées dynamiquement, en fonction de la demande des utilisateurs.

Exemples des ressources: le stockage, le traitement, la mémoire, le réseau, la bande passante et les machines virtuelles.

- Elasticité rapide (*Rapid Elasticity*) : C'est la capacité du Cloud à s'approvisionner de manière élastique et rapide. Selon la charge à un instant t, la capacité de mise à l'échelle rapidement et instantanément. Pour le consommateur, les ressources disponibles semblent illimitées et peuvent être augmentées ou diminuées, à tout moment.

- Service mesurable (*Mesaured Service*): Les systèmes Clouds peuvent contrôler et optimiser automatiquement l'utilisation des ressources en tirant parti d'un dosage de la capacité à un niveau d'abstraction, selon le type de service (stockage, traitement, bande passante). L'utilisation des ressources est gérée, contrôlée et déclarée de manière transparente au fournisseur et au consommateur du service utilisé.

Le Cloud Computing se caractérise aussi par:

- La mise à l'échelle (*Scalability*) : C'est la capacité du système à allouer dynamiquement des capacités (puissance de calcul, mémoire, stockage, réseau) lors d'une montée en charge ou un besoin ponctuel assimilable à une contrainte. Ces ressources sont libérées dynamiquement lorsque la contrainte disparaît.

- Facturation à l'usage (*Pay-as-you-use model*) : L'utilisation des ressources et des services associés est contrôlée, mesurée et facturée selon l'usage qu'en fait l'utilisateur. Les fournisseurs de Cloud public comme Amazon permettent aux entreprises d'éviter des investissements importants. Ce modèle est particulièrement favorable pour les petites entreprises et les start-ups, qui ne peuvent souvent pas se permettre de dépenser de grosses sommes d'argent en début de voyages d'affaires.

- Fiabilité (*Reliability*) : Le Cloud Computing est beaucoup plus fiable et plus cohérent que l'infrastructure informatique interne. La plupart des fournisseurs offrent un contrat de niveau de service qui garantit 24/7/365 et 99,99% de disponibilité. L'organisation peut bénéficier d'une piscine massive de ressources informatiques redondantes, ainsi que du mécanisme de basculement rapide. Si un serveur tombe en panne, les applications et les services hébergés peuvent facilement être transmis à l'un des serveurs disponibles.

II.2.3 Modèle en couches de l'informatique en nuage

Le modèle architectural du Cloud Computing est divisé en quatre couches principales, comme le montre la figure II.1. Chacune de ces couches accomplit certaines tâches, qui sont les suivantes :

- ***La couche d'application :*** Cette couche se situe au sommet de la hiérarchie des couches et contient toutes les applications du Cloud. Ces applications sont différentes des applications traditionnelles car elles peuvent être automatiquement mises à l'échelle pour améliorer les performances et la disponibilité, et elles réduisent également les coûts opérationnels.

- ***La couche plate-forme :*** Cette couche est créée au sommet de la couche d'infrastructure et se

compose de systèmes d'exploitation et de cadres d'application. L'objectif principal de cette couche est de réduire le déploiement des applications directement dans les conteneurs.

- ***La couche d'infrastructure :*** Cette couche est également appelée couche de virtualisation. Dans cette couche, l'infrastructure crée des ressources de stockage et de calcul en utilisant des hyperviseurs tels que KVM, Vmware ESXi ou XEN pour déployer et gérer les ressources virtuelles du Cloud.

- ***La couche matérielle :*** Elle est responsable de la gestion des ressources physiques (c'est-à-dire CPU, mémoire, stockage), notamment les serveurs, routeurs, commutateurs. Typiquement, l'équipement matériel est implémenté dans les centres de données ou tout point de présence (POP) [8].

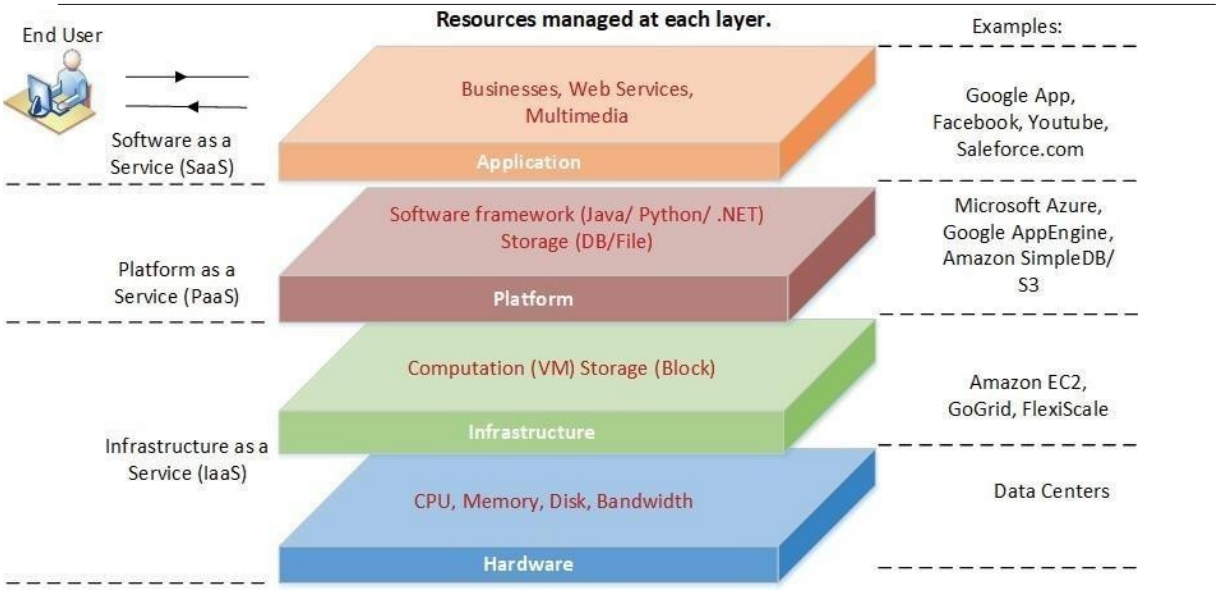


Figure II.1: *Architecture de l'informatique en nuage* [8].

II.2.4 Modèles de services

Le groupe de travail NIST a classé les services Cloud en trois modèles (couches), en fonction de la nature du service, à savoir logiciel, plateforme où infrastructure.

II.2.4.1 SaaS (Software as a Service)

C'est un modèle de déploiement d'applications, dans lequel un fournisseur (Opérateurs SaaS) loue une application logicielle en main de ses clients, en tant que service, à la demande. Beaucoup de logiciels sont proposés en tant que service, comme les messageries en ligne, les gestionnaires de la relation client "*Customer Relationship Management*" (CRM), les logiciels de stockage comme Flickr, la gestion des ressources humaines, comptabilité, etc.

La mise en place des infrastructures, des serveurs d'hébergements et des systèmes d'exploitations est entièrement assurée par les fournisseurs de services Cloud et elle est totalement abstraite pour l'utilisateur. Ce dernier peut néanmoins configurer où personnaliser son application via une API

(*Application Programming Interface*) que le fournisseur met en place pour ces clients afin qu'ils puissent configurer le SaaS selon leurs besoins. Egalement l'utilisateur est libre de n'effectuer aucune mise à jour, c'est le rôle des fournisseurs qui doivent contrôler l'accès et la mise à jour des SaaS qu'ils fournissent.

II.2.4.2 PaaS (Platform as a Service)

Ce modèle offre à l'utilisateur un ensemble de langage de programmation et des outils afin de lui permettre de développer des applications logicielles (à titre d'exemple, des applications web, des applications mobiles où des applications bureautiques), sans recourir au besoin d'installer un outil dans son poste de travail local, et de déployer ces applications développées sur une infrastructure Cloud du fournisseur. Ce dernier exerce un contrôle et une gestion complète de l'infrastructure sous-jacente à l'application, néanmoins l'utilisateur/le développeur exerce uniquement un contrôle sur les applications déployées.

Les services proposés par une plateforme en tant que service sont généralement accessible via des interfaces Web publiques. Les plateformes de développement les plus connues sont proposées par de grands éditeurs de logiciels en tant que services tels que Google App Engine et Windows Azure.

II.2.4.3 IaaS (Infrastructure as a Service)

Ce modèle fournit une infrastructure sous la forme d'un service pour le déploiement et l'exécution des applications, par exemple des serveurs, des capacités de calcul, des réseaux, un espace de stockage, de bande passante etc. Ces services sont offerts via l'internet et sous une forme de paiement à l'utilisation. Les clients de l'infrastructure ne paient que les ressources qu'ils consomment. L'utilisateur contrôle les systèmes d'exploitation, la capacité de stockage, et les applications déployées. Amazon EC2 (*Amazon Elastic Compute Cloud*) est un exemple d'IaaS.

II.2.5 Les types du Cloud Computing

Dans cette section, nous allons présenter les principaux types de Cloud Computing, qui comprennent les suivants :

- **Cloud privé** : Dans ce type, l'infrastructure du nuage est exclusivement utilisée par une entreprise comprenant plusieurs utilisateurs finaux. Elle peut être exploitée soit par l'organisation elle- même, soit par un tiers. Elle peut également être disponible sur site ou hors site.
- **Cloud communautaire** : L'infrastructure de ce type de Cloud peut être fournie à une communauté spécifique d'utilisateurs finaux qui ont des objectifs communs. L'infrastructure peut être détenue ou exploitée par l'un des membres de cette communauté d'utilisateurs finaux, par un tiers ou par une combinaison des deux.
- **Cloud public** : L'infrastructure de ce type de Cloud peut être détenue et gérée par une organisation universitaire ou commerciale et offrir un service au public. Ce service peut exister sur l'infrastructure du fournisseur de nuage.
- **Cloud hybride** : Comme son nom l'indique, l'infrastructure de Cloud hybride peut composer

deux ou plusieurs infrastructures de Cloud différentes (c'est-à-dire privées, communautaires et/ou publiques) [7].

II.2.6 Les avantages et les inconvénients du Cloud

Dans ce qui suit, nous allons montrer les avantages et les inconvénients du Cloud Computing.

II.2.6.1 Les Avantages

Le Cloud Computing offre beaucoup d'avantages et de flexibilité à ses utilisateurs. L'utilisateur peut opérer n'importe où et à tout moment de manière sécurisée. Vu le nombre croissant d'appareils compatibles avec le Web qui sont utilisés aujourd'hui (par exemple, les tablettes, les téléphones intelligents,etc.), l'accès à l'information et aux données doit être rapide et plus simple. Certains de ces avantages, très pertinents concernent l'utilisation d'un Cloud peuvent être les suivants :

- Réduire le coût de gestion et de l'investissement initial : avec le Cloud les entreprises ne se soucient pas de la gestion des ressources où du personnel nécessaire à la supervision de leurs plateformes. Le Cloud minimise les risques commerciaux ;
- Fournir une infrastructure dynamique qui offre des coûts réduits et des services améliorés avec moins de coûts de développement et de maintenance ;
- Fournir des services à la demande, flexibles, évolutifs, améliorés et adaptables grâce au modèle de paiement à l'usage « *Pay-as-you-go* » ;
- Fournir une disponibilité et des performances cohérentes avec des charges maximales provisionnées automatiquement ;
- Se rétablir rapidement et améliorer les capacités de restauration pour améliorer la résilience des entreprises ;
- Fournir une capacité de traitement, de stockage, de réseau illimité, etc. de manière élastique ;
- Offrir une collaboration de groupe facile, c'est-à-dire une flexibilité pour les utilisateurs àl'échelle mondiale de travailler sur le même projet ;
- Offrir un calcul respectueux de l'environnement car il utilise uniquement l'espace serveur requis par l'application.

II.2.6.2 Les inconvénients

Certains des inconvénients lors de l'utilisation d'un Cloud sont comme suit :

- Le Cloud nécessite un réseau avec une haute vitesse de communication et une connectivité constante ;

- Les données et les applications sur un Cloud public pourraient ne pas être très sécurisées, ce qui pose le problème de la confidentialité et de la sécurité ;
- Nécessite une surveillance et une application constante des accords de niveau de service (SLA).

II.3 La virtualisation

II.3.1 Définition

Le concept de virtualisation est l'un des piliers du Cloud Computing. En fait, chronologiquement parlant, la virtualisation est apparue longtemps avant la naissance du concept du Cloud Computing. C'est un processus qui va permettre de masquer les caractéristiques physiques d'une ressource informatique de manière à simplifier les interactions entre cette ressource et d'autres systèmes, d'autres applications et les utilisateurs. Elle va permettre de percevoir une ressource physique comme plusieurs ressources logiques et, inversement, de percevoir plusieurs ressources physiques comme une seule ressource logique.

II.3.2 Machines virtuelles

Une machine virtuelle (figure II.2) peut être définie comme la représentation logique de l'ordinateur. Elle reposeprincipalement sur le découplage du matériel et du système d'exploitation pour offrir plus de flexibilité et maximiser l'efficacité de l'utilisation des ressources sous-jacentes. Dans un environnement de virtualisation,un seul matériel physique est capable d'accueillir plusieurs VM partageant les ressources de la même machine physique. Cependant, chaque VM est isolée et fonctionne indépendamment sur le même hôte physique. Par conséquent, en cas de défaillance de l'une de ces VM, cette erreur n'affecte pas les autres VM[9].

En général, les VM sont contrôlées par un hyperviseur, qui est un firmware ou un logiciel utilisé pour employer les ressources de la machine physique et gérer plusieurs VM. Il est principalement responsable de la création, de l'allocation des ressources (c'est-à-dire CPU, mémoire, stockage) et de la destruction des VM lorsqu'elles ne sont plus nécessaires. Ces hyperviseurs peuvent fonctionner au-dessus du matériel, comme VMWare ESXi, XEN et KVM, et ce type d'hyperviseurs est appelé hyperviseur natif ou bare-metal. L'autre type s'exécute au-dessus du système d'exploitation conventionnel comme tous les autres programmes et applications, et ce type est appelé hyperviseurs hébergés.

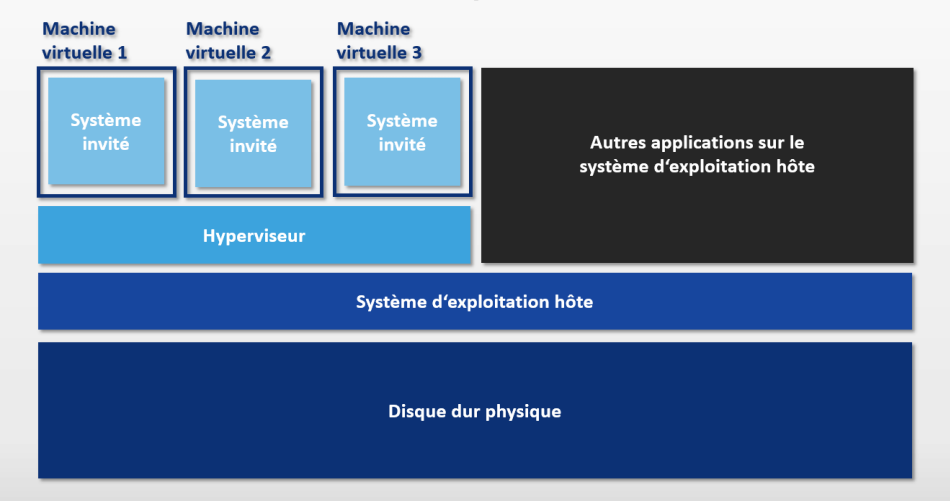


Figure II.2 : *L'architecture de la machine virtuelle.*

II.3.3 La virtualisation des réseaux

La virtualisation du réseau (figure II.3) consiste à combiner des ressources réseau matérielles et logicielles dans une seule unité administrative. L'objectif de la virtualisation du réseau est de fournir aux systèmes et utilisateurs un partage efficace, contrôlé et sécurisé des ressources réseau.

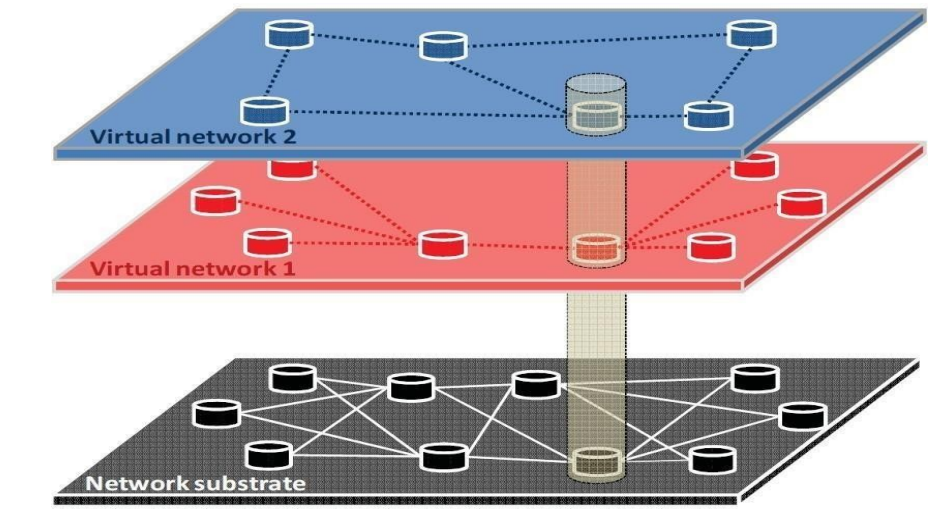


Figure II.3 : *Modèle de virtualisation du réseau [10].*

Parmi les techniques qui ont été utilisées pour la virtualisation des réseaux, il y a les réseaux locaux virtuels (VLAN), les réseaux privés virtuels (VPN), les commutations logiques, le routage logique, le pare-feu logique, l'équilibrage de charge logique. Certains de ces techniques étaient très utiles avec des équipements réseaux et serveurs physiques.

La virtualisation des serveurs ajoute une couche d'abstraction au-dessus de la couche physique, et elle peut créer des problèmes lorsque le data center repose sur des architectures réseau traditionnelles. Par

exemple, les réseaux VLAN utilisés par les machines virtuels (VMs) doivent être affectés au même port de commutation que le serveur physique exécutant l'hyperviseur. Toutefois, les machines virtuelles peuvent être déplacées, et l'administrateur réseau doit être en mesure d'ajouter, supprimer et modifier des ressources et des profils réseau. Un tel processus difficilement envisageable avec des commutateurs réseau traditionnels.

En général, les VNs se composent de deux éléments principaux, à savoir la virtualisation des nœuds et des liens. La virtualisation des liens permet de transporter un ensemble de liens virtuels sur un lien physique partagé. D'autre part, un nœud virtuel peut être hébergé sur tout nœud physique du réseau quiprend en charge les technologies de virtualisation. En d'autres termes, chaque nœud physique peut héberger plusieurs nœuds virtuels sécurisés et les gérer via un hyperviseur [10].

II.3.4 La virtualisation des fonctions réseau (NFV)

La virtualisation des fonctions réseau (NFV) représente une transformation significative pour les réseaux de télécommunications/fournisseurs de services, motivée par les objectifs de réduction des coûts, d'augmentation de la flexibilité et de fourniture de services personnalisés. La promesse de NFV se concrétisera au cours des prochaines années, et plusieurs défis doivent être relevés pour y parvenir. Les opérateurs télécoms ont transféré la plupart de leurs communications vers des réseaux IP standard et commencent maintenant à migrer la plupart de leurs ordinateurs vers des serveurs standard. NFV est une tendance récente qui s'explique par la disponibilité de la technologie qui permet désormais de produire des paquets de haute performance sur des systèmes de base, et par le désir des fournisseurs de services de virtualiser les fonctions réseau telles que les routeurs, les firewalls, la traduction d'adresses réseau (NAT), etc, NFV s'appuie sur les principes de l'informatique en nuage pour changer la façon dont les NF tels que les passerelles et les middle boxes sont ouverts.

Le concept de NFV a été lancé au sein du consortium de l'Institut européen des normes de télécommunications (ETSI). Le NFV permet aux NF existants offerts par des équipements spécialisés de fonctionner en logiciel sur du matériel générique. Les principales technologies à l'origine de cette évolution sont le Cloud Computing et la virtualisation. NFV permet de déployer des VNF dans des serveurs de base à haute performance dans le data centre d'un opérateur, avec une grande flexibilité pour faire tourner les VNF à la demande.

II.3.4.1 Architecture de la virtualisation des fonctions réseau

L'architecture NFV proposée par l'Institut européen des normes de télécommunications (ETSI) aide à définir les normes pour la mise en œuvre de la virtualisation des fonctions réseau. Chaque composant de l'architecture repose sur ces normes afin d'offrir un niveau de stabilité et d'interopérabilité supérieur.

Une architecture NFV (figure II.4) comprend les éléments suivants [11]:

- Les **fonctions réseau virtualisées (VNF)** sont des applications logicielles qui fournissent des fonctions réseau, telles que le partage de fichiers, les services d'annuaire et la configuration d'IP.

- L'**infrastructure de virtualisation des fonctions réseau (NFVi)** consiste en un ensemble de composants d'infrastructure (calcul, stockage, réseau) sur une plateforme,qui prend en charge des logiciels, par exemple un hyperviseur tel que KVM, ou une plateforme de gestion de conteneurs, nécessaire pour exécuter des applications réseau.
- Le **composant de gestion, d'automatisation et d'orchestration réseau ou MANO (Management, Automation and Network Orchestration)** fournit la structure permettant de gérer l'infrastructure NFV et l'approvisionnement de nouvelles fonctions réseau virtualisées.

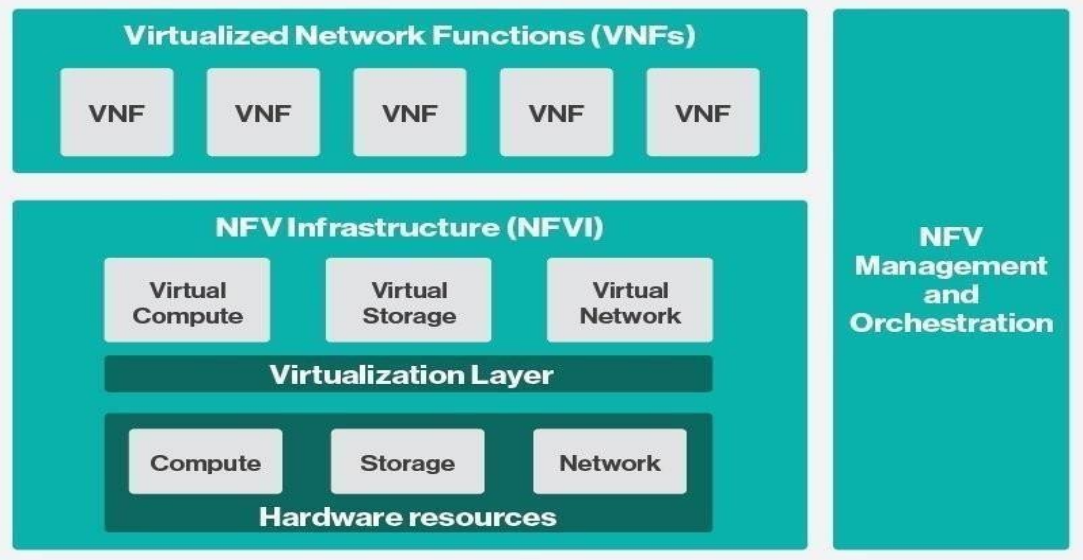


Figure II.4 : *Architecture de la virtualisation des fonctions réseau (NFV).*

II.3.4.2 Les avantages de NFV

- NFV réduit le coût des équipements de calcul et de mise en réseau en exploitant la technologie de virtualisation ;
- La virtualisation des fonctions de réseau accélère le délai de mise sur le marché. Cela peut être réalisé en réduisant l'innovation du cycle de l'opérateur de réseau [1];
- NFV garantit la disponibilité de la multi-location, ce qui permet aux opérateurs de réseaux d'utiliser une seule plateforme pour plusieurs applications et utilisateurs finaux. En d'autres termes, les opérateurs de réseaux peuvent partager les ressources physiques entre plusieurs services destinés à différents utilisateurs finaux ;
- Il s'adresse à une large population d'utilisateurs finaux et, par conséquent, les services peuvent être étendus ou réduits selon les besoins ;
- L'ouverture : Elle incite les utilisateurs finaux et les chercheurs du monde universitaire ainsi que les petits acteurs à innover rapidement des services à moindre risque [1].

Malgré tous les avantages que NFV offre à l'industrie, déterminer le nombre d'instances VNF

requises et finir les ressources disponibles dans l'infrastructure pour héberger ces instances sont des défis majeurs qui seront abordés dans la section suivante.

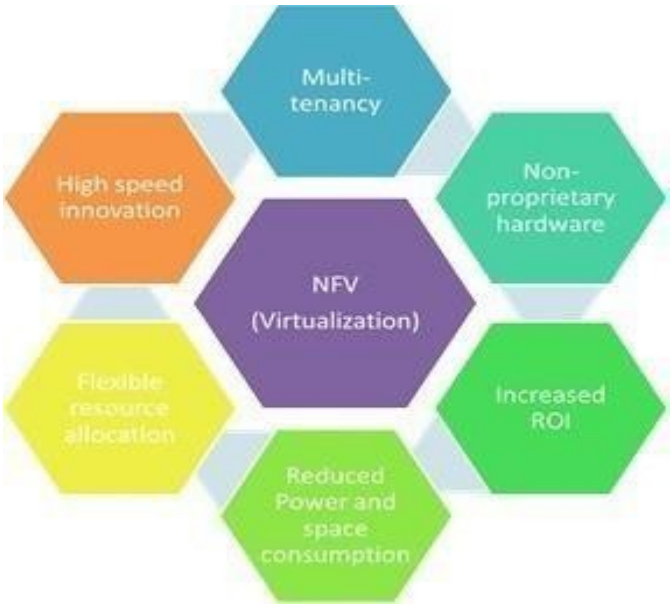


Figure II.5 *: Aperçu des avantages de la NFV [12]*

II.3.5 Software Defined Networking

La première documentation standard relative au SDN a été introduite par l'Université de Berkeleyet Stanford en 2008. Plus tard, en 2011, est née l'Open Networking Foundation (ONF), une organisation dirigée par les utilisateurs et dédiée à la promotion et à l'adoption du SDN par le développement de normes ouvertes [13].

Selon l'ONF [14], la mise en réseau définie par logiciel est une nouvelle approche de la mise en réseau, dans laquelle le plan de contrôle (qui décide de la manière de gérer le trafic) est découplé du plan de données (qui achemine le trafic en fonction des décisions prises par le plan de contrôle) et est directement programmable.

Il en résulte une architecture extrêmement dynamique, facile à utiliser, rentable et adaptable, qui offre aux administrateurs une programmabilité sans précédent, l'automatisation et le contrôle.

En résumé, les principales caractéristiques du SDN sont les suivantes :

- Directement programmable: le découplage des fonctions d'acheminement permet la programmation directe du contrôle du réseau.
- Agile : cette séparation permet aux administrateurs d'adapter dynamiquement le trafic aux besoins changeants.
- Gestion centralisée : l'intelligence du réseau est assurée par les contrôleurs SDN qui ont

une vue générale du réseau et sont considérés par les applications de niveau supérieur comme un commutateur unique.

- Configuration programmée : les gestionnaires de réseau sont en mesure de configurer, de gérer, de sécuriser et d'optimiser efficacement les réseaux grâce à des programmes SDN automatisés qu'ils peuvent, en outre, écrire eux-mêmes en ne dépendant pas de logiciels propriétaires.
- Fondé sur des normes ouvertes et neutre vis-à-vis des fournisseurs : le fait d'être mis en œuvre par le biais de normes ouvertes permet de simplifier l'exploitation et la conception des réseaux.

Pour être en mesure de fournir ces caractéristiques, l'architecture SDN se compose de trois couches distinctes (figure II.6) :

Couche d'application : elle est composée des applications qui communiquent directement avec le contrôleur et demandent les ressources dont elles ont besoin. Elles peuvent avoir des formes et des objectifs très variés.

Couche de contrôle : elle fournit la fonction de contrôle logiquement centralisée qui supervise le comportement du réseau par le biais d'interfaces ouvertes. Cette fonction est généralement assurée par une entité logicielle capable de traduire les besoins des applications en règles pour le plan de données et deleur fournir des informations sur le réseau.

Couche infrastructure : elle est constituée de tous les éléments et dispositifs de réseau de bas niveau qui assurent la commutation et la transmission des paquets.

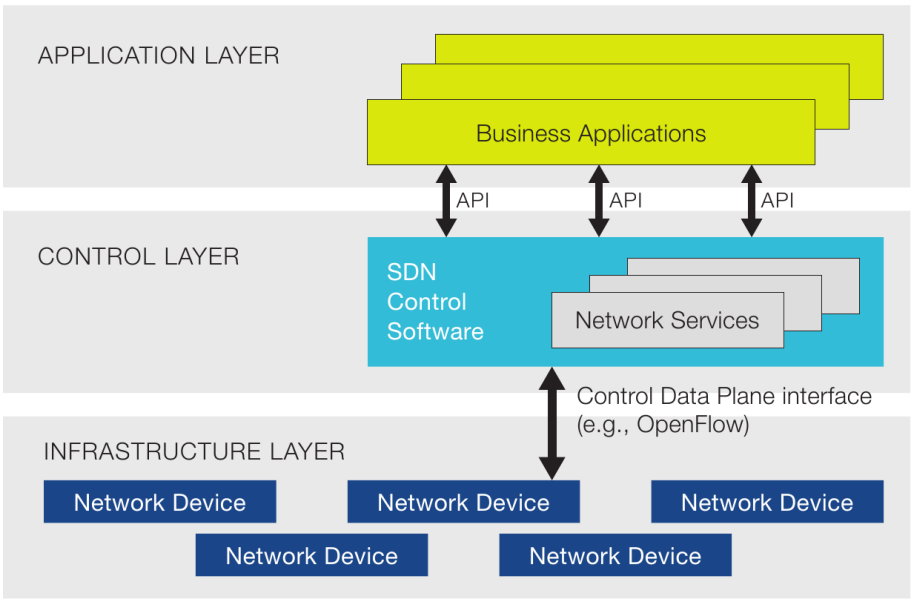


Figure II.6 *: Vue d'ensemble de l'architecture SDN [15].*

Il semble donc que, grâce à son architecture, le SDN soit capable d'apporter au réseau l'innovation nécessaire pour passer d'un réseau statique à une plateforme dynamique capable de s'adapter efficacement

aux besoins croissants du secteur et des utilisateurs.

II.3.6 Relation entre NFV et SDN

Après avoir examiné plus en détail ces deux paradigmes, nous pouvons conclure que, bien qu'il s'agisse de concepts indépendants, ils se combinent très bien et se complètent. Mais pour être plus clair, nous pouvons les résumer comme suit.

SDN : il concerne les fonctionnalités du réseau. Découple les plans de contrôle et de données et fournit un contrôleur centralisé et une programmabilité du réseau.

NFV : il s'agit du concept de transfert des fonctions réseau des dispositifs matériels dédiés vers des applications logicielles et découple les fonctions réseau du matériel propriétaire sans en affecter la fonctionnalité.

De cette façon, nous pouvons conclure que :

Le SDN sert NFV en fournissant la connectivité programmable entre les VNF. Des connexions qui seront orchestrées par le contrôleur SDN.

NFV sert le SDN en implémentant ses fonctions de réseau de manière logicielle, permettant au contrôleur SDN d'être virtualisé et exécuté sur un nuage, qui peut être migré en fonction des besoins du moment.

II.3.7 Les fonctions de réseau virtuel (VNFs)

La virtualisation des fonctions réseau (NFV) vise à remplacer les équipements matériels par des fonctions réseau (NF) logicielles. Elle permet de mettre en œuvre et d'exécuter des fonctions réseau sur des serveurs prêts à l'emploi en utilisant des langages de programmation, des cadres et des techniques de virtualisation de base. Les services NFV (applications) sont généralement développés à l'aide d'une architecture basée sur les VNFs où chaque service est constitué d'un ou plusieurs VNFs. Ces VNFs sont enchaînés logiquement pour créer la chaîne de services (SFC) telle que décrite dans le VNFFG. Les fonctionnalités des VNFs sont combinées pour fournir des services abstraits de haut niveau. Comme le décrit le VNFFG, les VNFs participants à la chaîne de fonctions de service sont configurés pour représenter les dépendances fonctionnelles et former les chemins de calcul du service. La figure II.7 illustre la chaîne de fonctions de service de VNF.

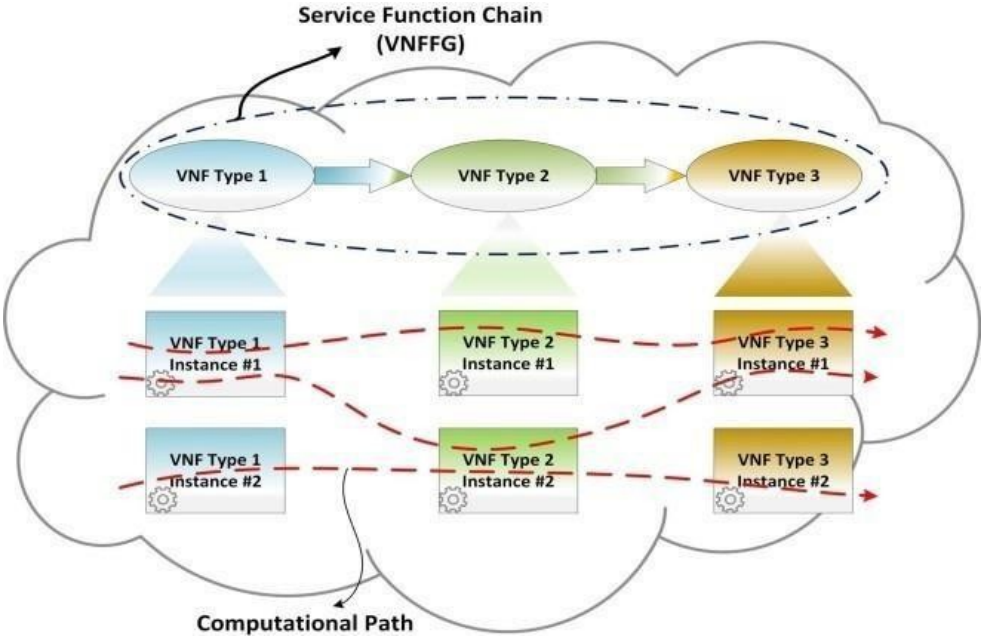


Figure II.7 : *Chaîne de fonctions de service et chemin de calcul de NFV de différents types de VNF.*

Les VNFs sont hébergés dans un environnement en nuage où ils sont exécutés soit dans des machines virtuelles (VM), soit dans des containers. L'allocation de l'environnement d'exécution des VNFs sur les serveurs d'hébergement dans les data center affecte directement la qualité du service fourni par ces VNFs. Par conséquent, il est essentiel de disposer d'une allocation optimale pour les VNFs afin de satisfaire aux exigences de qualité des opérateurs. Dans cette optique, le placement des VNFs et le chaînage des services restent des défis importants qui doivent être étudiés plus en profondeur afin d'obtenir les avantages escomptés des NFV, tels que la réduction des dépenses d'exploitation et d'investissement.

II.2.7 Description du problème de placement

Dans cette section, nous démontrons le problème de placement des instances VNFs. Ce problème est principalement divisé en deux sous-problèmes. Le premier est de déterminer le nombre d'instances VNF nécessaires pour traiter une certaine quantité de demande de trafic. Le second consiste à trouver le placement le plus approprié pour les instances VNF dans l'infrastructure de telle sorte que le coût opérationnel des demandes et les coûts de synchronisation entre chaque paire d'instances VNF du même type soient minimisés.

Nous commençons d'abord par définir la chaîne de fonctions de service (SFC) ou la demande de service est un ensemble de VNF qui traverse le chemin des sources vers une destination unique. Les SFC sont soit composés d'un seul ou de plusieurs VNF de différents types comme le montre la figure II.8, car il peut commencer par un équilibreur de charge suivi d'un firewall et se terminer par un IDS.

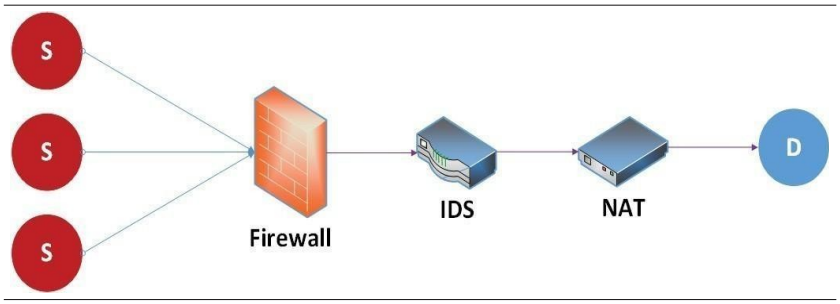


Figure II.8 : *Chaîne fonctionnelle des services.*

Les VNFs de chaque SFC peuvent contenir une ou plusieurs instances du même type. Par exemple, une chaîne de services est composée d'un firewall, d'un équilibreur de charge et d'un NAT. Un seul VNF de chaque fonction de firewall et d'équilibreur de charge peut suffire la quantité de trafic reçue.Cependant, deux instances de NAT sont nécessaires pour traiter le même trafic. Cet exemple montre que le nombre d'instances de chaque VNF varie en fonction de la capacité de traitement du VNF dans chaque requête. Jusqu'à présent, nous apprenons qu'en déterminant le nombre d'instances requises pour les VNFs, nous assurons la continuité d'une demande de service. Cela ne peut être réalisé qu'en connaissant la capacité de traitement maximale de VNF à laquelle l'utilisation des ressources est maximisée tout en évitant la perte de paquets. Une fois que les exigences de la demande de service ont été définies, le fournisseur de services sera responsable d'allouer des ressources pour ces VNF dans l'infrastructure de Cloud.

Les opérateurs de réseau, quant à eux, sont chargés de déterminer le nombre d'instances nécessaires pour chaque VNF dans un SFC. De plus, comme ils ont une vue globale de leur infrastructure, ils doivent finir l'emplacement le plus approprié dans l'infrastructure pour intégrer les instances de VNF de telle sorte que l'intégration ne viole pas l'ordre du SFC et qu'elle maximise le coût opérationnel et le débit de données échangé entre chaque paire d'instances. L'allocation de ressources pour les instances VNF et leur intégration dans l'infrastructure physique est une tâche difficile en raison du fait que l'utilisation d'un nombre fixé d'instances VNF peut sous utiliser ou épuiser les sources physiques. Par conséquent, il est nécessaire d'instancier plusieurs instances en fonction du trafic entrant reçu. Si ce trafic est plus important que la capacité de traitement d'une instance, alors une autre instance du même type de VNF est nécessaire afin d'obtenir la continuité du SFC. Pour déterminer le nombre d'instances de chaque VNF pour provisionner une chaîne de fonctions de service, nous prenons en compte la prédiction du trafic futur, le temps de traitement, le délai maximal de bout en bout.

- **Demande de trafic future** : étant donné qu'une forte demande de trafic maximise l'utilisation des ressources VNF, cela soulève la question de savoir si un seul VNF est suffisant pour servir plusieurs demandes. Si l'utilisation des ressources, spécifiquement le CPU, atteint son pic, alors le VNF abandonnera les futures requêtes. Pour cette raison, les performances des VNFs doivent être évaluées afin d'estimer le nombre d'instances nécessaires pour servir la demande et minimiser l'abandon des requêtes par manque de ressources, tout en utilisant les ressources de manière plus efficace.

- **Délai de bout en bout** : il s'agit du temps total que prend un certain trafic pour traverser le chemin, qui est composé d'un ensemble d'instances VNF, des sources à la destination. De plus, le nombre d'instances VNF de chaque chaîne de services varie en fonction du trafic entrant à la source. Cela signifie que chaque VNF a une capacité de traitement différente qui varie en termes de temps de traitement, ce qui affecte à son tour les délais totaux.

- **Coût opérationnel** : Cette métrique est définie comme le coût du déploiement et du transfert d'une image VNF d'un emplacement à un autre au sein de l'infrastructure et de son exécution pendant un certain temps (c'est-à-dire la durée de vie de la demande) où chaque demande a une durée de vie différente. Le déploiement de VNF dépend du type de VNF, du système d'exploitation et du coût de la licence de VNF.

- **Synchronisation des instances VNF** : La synchronisation de VNF est le taux auquel les données sont échangées entre plusieurs instances VNF du même type. Ce taux varie en fonction de la fonctionnalité de la fonction réseau.

II.4 Conclusion

Dans ce chapitre, nous avons présenté le Cloud Computing et ses modules, la virtualisation, la virtualisation des réseaux et finalement introduit la virtualisation des fonctions réseau (NFV). Nous avons également énuméré les avantages et les inconvénients de chaque technologie, plus précisément, comment la virtualisation réduit les coûts d'investissement et de dépenses.

La NFV a été introduite au cours des dernières années. Bien que de nombreuses propositions abordent le problème du placement des VNFs, aucune d'entre elles n'a évalué les performances des VNFs afin de déterminer le nombre d'instances nécessaires pour satisfaire une certaine quantité de trafic. Sans compter que le placement de VNF devient encore plus difficile en tenant compte du coût de synchronisationentre chaque paire d'instances VNF dans le SFC.

Dans le chapitre suivant nous abordons certaines des solutions existantes concernant le problème du placement des fonctions de réseau virtualisées (VNFs).

Chapitre III : Revue de laLittérature sur le placement de VNF

III.1 Introduction

Dans ce chapitre, nous abordons certaines des solutions existantes concernant le problème du placement des fonctions de réseau virtualisées (VNF). Tout en présentant ces solutions, nous mettons en avant leurs objectifs et leurs limites. Nous avons classifié les travaux existants en deux catégories principales en fonction de leurs objectifs. Ces deux catégories sont le placement et la réduction des coûts des VNFs, et le routage du trafic des chaînes de services.

III.2 Travaux antérieurs et solutions existantes

La chaîne de fonctions de service (SFC) [16] consiste à connecter différentes fonctions de réseau dans un ordre spécifique et à fournir le service correspondant aux utilisateurs. Les fonctions de réseau dans la SFC sont réalisées comme différentes fonctions de réseau virtualisées (VNF). Dans un réseau réel,le SFC peut être configuré et ajusté en fonction des différentes demandes de trafic. Le processus de configuration comporte deux aspects : le placement des VNFs et la gestion du trafic entre les différentes VNFs.

En termes de placement de VNF, les opérateurs de réseau (ou les fournisseurs de services Internet) doivent sélectionner l'emplacement de l'instance VNF (VNFI), qui peut exécuter VNF et allouer les ressources (CPU, mémoire, etc.) pour chaque VNFI. En ce qui concerne l'orientation du trafic (routage), il faut déterminer le chemin utilisé pour transmettre le trafic à travers des VNFs spécifiques du SFC. Une configuration appropriée du SFC peut être utile pour améliorer les performances du réseau et réduire les coûts opérationnels.

Dans un environnement de réseau réel, les utilisateurs et les opérateurs de réseau ont leurs propres exigences de performance pour les fonctions du réseau. Pour les opérateurs de réseau, les exigences peuvent être de réduire le coût de placement des VNFs et d'améliorer l'utilisation des ressources. Et pour les utilisateurs communs, les exigences peuvent être d'augmenter le débit du réseau et de réduire les délais de transmission du trafic. Ces exigences de performance doivent être satisfaites en adoptant une configuration SFC appropriée (y compris le placement de VNF et le routage du trafic).

Au cours du processus de modélisation, afin d'obtenir de meilleures performances du réseau, le problème du placement des VNFs et celui du routage du trafic peuvent être considérés conjointement. L'objectif d'optimisation peut être la combinaison des objectifs d'optimisation du placement et du routage.

Actuellement, de nombreuses recherches se concentrent sur le problème d'optimisation du placement et du routage pour le SFC. Elles utilisent différentes méthodes pour modéliser le problème d'optimisation et développent des algorithmes correspondants pour résoudre efficacement le problème. Les algorithmes tentent de trouver des configurations SFC optimales afin de fournir de meilleurs services de réseau aux utilisateurs avec un coût moindre.

Dans ce qui suit, nous abordons certaines des solutions existantes concernant le problème du

placement de VNFs. Les travaux existants sont classés en deux catégories principales comme suite :

i. Le placement et la réduction des coûts des VNFs,

ii. Le routage du trafic des chaînes de services.

III.2.1 Placement de VNF et réduction des coûts

Dans cette section, nous discutons des travaux les plus significants qui ont été réalisés vers le placement de VNF avec un objectif de réduction des coûts. Nous mettons en évidence les approches de placement de VNF les plus significants en prenant en compte différentes métriques.

- **Clayman et al. (2014)** [17], proposent une architecture multicouche pour la gestion et l'orchestration qui prend en charge (SDN) et (NFV). Cette architecture se compose de quatre couches : application, orchestration, abstraction et infrastructure. L'architecture et le schéma de gestion assurent une gestion automatique l'allocation de services réseau et le placement de VNF où les VNFs et les liens virtuels sont placés dynamiquement en cas de besoin.

  Les résultats montrent que chaque moteur de placement configurable démontre un comportement différent, et fournit différentes stratégies de placement pour les fonctions réseau virtuelles. Cependant, cette étude ne prend pas en compte d'autres types de VNF tels que le pare-feu, le DPI et l'IDS pour le placement. Le chaînage des VNFs n'est pas pris en compte dans ce travail. Enoutre, les auteurs n'utilisent pas un nombre réaliste d'hôtes dans les expériences pour évaluer la demande de fonctions réseau dans l'environnement de production.

- **Mijumbi et al. (2015b)** [18], abordent le problème de l'intégration et de l'ordonnancement de VNF sur un réseau virtuel. Ils formulent le problème d'intégration et d'ordonnancement de VNF en ligne à l'aide d'ILP et proposent trois algorithmes avides et une heuristique basée sur la recherche Tabou pour résoudre ce problème. Les auteurs suggèrent que les VM sont déjà mappées dans le réseau physique et que chaque VM peut traiter plusieurs VNFs en utilisant des conteneurs, en supposant que la taille des VNF est considérablement légère. Sans oublier que chaque VNF doit être traité dansun ordre specifique. Pour l'ordonnancement, ils supposent que chaque demande de service a une exigence de délai s'il est dépassé, la demande sera rejetée.

  Les algorithmes gourmands intègrent les demandes au fur et à mesure qu'elles arrivent, puis les classent en fonction de critères gourmands (c'est-à-dire le moins de charge, le moins de temps de traitement et les files d'attente de VM les plus courtes). TabuSearch vise à finir une solution ensuite d'optimiser cette solution en explorant les voisins.

  Les résultats montrent que l'algorithme proposé basé sur la recherche tabou surpasse les algorithmes gourmands en termes de taux d'acceptation, de coût total et de revenu total. L'utilisation de conteneurs pour accueillir des VNF dans une seule VM peut convenir à certaines fonctions, mais pasà toutes. Cependant, les VMs offrent une isolation pour chaque VNF malgré le fait qu'elles utilisent

plus de ressources que les conteneurs. De plus, les algorithmes proposés sont incapables de s'adapter aux changements des conditions du réseau. De plus, l'hébergement de VNF dans une seule VM peut conduire à une sur-utilisation imprévisible des ressources (c.-à-d. CPU, mémoire et stockage). Sans compter que ces travaux ne prennent pas en compte la nécessité d'intégrer plusieurs instances VNFs pour répondre à une demande de trafic élevée.

- **Bari et al. (2015b)** [19], abordent le problème de l'orchestration des VNFs (VNF-OP). Plus en détail, le problème est divisé en coût de déploiement de VNF tout en provisionnant la chaîne de services, en coût d'énergie pour le fonctionnement des VNFs, et en transfert de trafic de/vers les VNFs. Les auteurs formulent VNF-OP à l'aide d'un modèle ILP pour objectif de minimiser les coûts de déploiement, d'énergie et de transfert de trafic. Ils approfondissent ensuite leur solution en proposant une heuristique qui trouve une intégration faisable des chaînes de services qui minimisele coût opérationnel et l'utilisation des ressources à plus grande échelle.

  De plus, ils comparent leur heuristique proposée avec le modèle ILP utilisant CPLEX et les résultats montrent que l'algorithme proposé trouve un encastrement pour les chaînes de services 1,3fois plus vite que la solution CPLEX. Sans oublier que le temps d'exécution est 56 à 3500 fois plus rapide que la solution CPLEX. Les auteurs supposent qu'un VNFs pour chaque fonction de la chaîne de services est suffisant pour traiter la demande de trafic de n'importe quelle taille, mais cela n'est pas pratique dans un environnement de production.

  En outre, les auteurs considèrent que chaque VNF a des exigences distinctes en matière des ressources cependant, cela conduit à un gaspillage de ressources aux moments où un faible taux de charge de trafic est reçu. Ils n'ont pas non plus envisagé d'instancier plusieurs instances au lieu d'avoir une seule grande instance VNF pour réduire le coût énergétique et la consommationd'énergie.

- **Luizelli et al. (2015)** [20], proposent un modèle d'optimisation basé sur la programmation linéaire en nombres entiers (ILP) qui aborde le problème du placement de VNF, ainsi qu'un algorithme qui fait face aux grandes infrastructures. Le problème est principalement divisé en trois phases : placement, affectation et chaînage. La phase de placement, quant à elle, vise à déterminer le nombre de VNF nécessaires et l'endroit où les placer. La phase d'affectation détermine quelle VNF sera responsable de chaque flux.

  Dans la phase de chaînage, des chemins de réseau sont créés pour interconnecter les VNFs. Le modèle proposé réduit les délais de bout en bout et empêche le surprovisionnement des ressources. Cependant, le placement ne prend pas en compte le chemin le plus court entre la source et le POP dans lequel les VNFs sont placés. Le nombre estimé d'instances pour chaque VNF dans un SFC n'est pas considéré dans ce travail.

- **Wang et al. (2016)** [21], abordent le problème du déploiement en ligne d'instances VNFs multiples évolutives afin de traiter le taux de trafic reçu aux VNFs pour atteindre le coût minimal de

provisionnement des ressources. Les auteurs proposent deux algorithmes. L'un pour la chaîne de services unique et l'autre pour les chaînes de services multiples. Le premier algorithme est divisé en deux phases. La première phase est appelée phase de pré-planification. Dans cette phase,toutes les instances VNF sont interdites de migration entre les serveurs. Dans la deuxième phase, l'algorithme s'adapte à l'algorithme de ski-location aléatoire pour atteindre le ratio optimal. Ce qui distingue le deuxième algorithme proposé du premier est qu'il considère l'existence d'un vecteur de taux de flux d'entrée maximal, qui détermine le nombre maximal d'instances VNFs à déployer pour chaque chaîne de services.

Les résultats montrent que l'algorithme randomisé (Algorithme 1) peut réduire jusqu'à 70% du coût opérationnel par rapport au Receding Horizon Control (RHC). Cependant, l'auteur affirme que moins le nombre d'instances déployées est élevé, plus la réduction de coût est importante. Bien que cette affirmation soit vraie, elle n'est pas réaliste car le nombre d'instances peut augmenter ou diminuer en fonction de la demande.

- Dans (**Wang et al. (2017)**) [22], les auteurs abordent le problème du déploiement optimal de VNF dans une infrastructure cloud. Les auteurs proposent dans ce travail un algorithme de provisionnement en ligne et un schéma d'optimisation de type bandit multi-armed. L'algorithme de provisionnement en ligne minimise le coût d'instanciation tandis que le schéma d'optimisation détermine le placement des instances VNFs sur les serveurs.

  L'algorithme de provisionnement en ligne est une heuristique en ligne aléatoire adaptée basée sur l'algorithme ski-rental (un algorithme de prise de décision d'achat/location en ligne), qui estime dynamiquement le nombre d'instances VNFs nécessaires pour minimiser le coût global en incluant potentiellement le coût d'instanciation pour les nouvelles demandes, ou en maintenant les instances en fonctionnement, et en payant le coût opérationnel.

  D'autre part, le schéma d'optimisation par bandit à plusieurs bras comprend un algorithme d'apprentissage en ligne basé sur le bandit pour le placement des instances VNFs qui réduit la congestion du trafic au sein des centres de données en prédisant les charges congestionnées et en les évitant alors que le placement a pris place. Les auteurs évaluent les performances de leur algorithme à l'aide d'une simulation basée sur les traces et des expériences à petite échelle.

  Les résultats montrent que les algorithmes proposés réalisent une réduction significante en termes de rapport concurrentiel, de coûts opérationnels et d'instanciation. Les auteurs ont abordé différents types de charges de travail. Cependant, ces travaux n'identifient pas clairement comment les charges de travail de fond affectent la décision de placement de VNF. Ils considèrent également plusieurs instances de VNF sans évaluer la capacité de traitement des VNFs. Ils supposent que les équilibreurs de charge, les firewalls et les NATs ont la même capacité de traitement alors que les fonctions réseau IDS traitent un taux de trafic inférieur, ce qui n'est pas réaliste.

III.2.2 Le routage de trafic des chaînes de services

Nous démontrons ici les propositions les plus importantes en ce qui concerne le routage du trafic des chaînes de services dans le but d'optimiser l'utilisation de la bande passante des liens et de minimiser le délai de bout en bouts.

- **Mehraghdam et al.** (2014) [23], proposent un modèle pour formaliser le chaînage des fonctions réseau en utilisant un langage sans contexte, qui contient l'ordre des fonctions réseau à placer. Ils proposent une heuristique qui réduit le temps d'exécution du processus en cas de déploiement multiple. les possibilités d'ordonnancement des requêtes. De plus, les auteurs formulent un modèle d'optimisation pour le meilleur placement des fonctions réseau chaînées dans plusieurs centres de données en tenant compte de certains paramètres tels que le débit de données, le nombre de nœuds utilisés et la latence.

  Les résultats de ce travail montrent des compromis entre le débit total restant, le nombre de nœuds utilisés et la latence. De plus, il montre le potentiel de finir un placement qui optimise les trois métriques si les ressources sont suffisantes. Cependant, ce travail ne prend pas en compte le chaînage des NFs de manière dynamique en fonction de la demande des locataires.

- **Moens et De Turck (2014)** [24], abordent le problème du placement des VNFs. Le problème est principalement axé sur le scénario de placement de VNF où la charge de base est traitée par des boîtes intermédiaires basées sur le matériel, et la charge restante est traitée par des VNF lorsque les boîtes intermédiaires sont entièrement utilisées. Les auteurs de ce travail proposent un algorithme de placement des fonctions de réseau virtuel (VNF-P) pour l'allocation des ressources des réseaux NFV hybrides.

  Les résultats montrent que le modèle fonctionne de manière efficace dans les petits réseaux de fournisseurs de services en termes de vitesse d'exécution. Cependant, les auteurs dans ce travail ne prennent pas en compte de remplacer les boîtes intermédiaires basées sur le matériel par l'instanciation des instances VNF pour minimiser le coût opérationnel. En outre, les auteurs n'ont pas discuté en détail la manière dont la capacité de traitement d'une middle-box est déterminée.

- **Huang et al. (2015)** [25], s'attaquent au problème du mélange de flux pour des chaînes de services multiples. Le problème dans ce travail est détaillé en deux sous-problèmes : Le problème de contention intra-chaîne et le problème de contention inter-chaîne. Le problème de contention intra-chaîne est fondamentalement causé lorsque le trafic flux d'une seule chaîne de service utilise le même ensemble de liens plusieurs fois alors que la contention inter-chaîne est conduite lorsque plusieurs chaînes de service passent par le même ensemble de liens plusieurs fois.

  Pour équilibrer le routage du trafic entre plusieurs chaînes de services, les auteurs proposent l'algorithme Network-Aware Chains Orchestration (NACHOS), qui applique la programmation linéaire et dynamique afin de maximiser la bande passante disponible du réseau. De plus, la programmation dynamique résout ce problème en utilisant la rétro-induction, qui enregistre le nombre de fois qu'une chaîne de services passe par les mêmes liens.

Les auteurs comparent NACHOS, avec un mécanisme de chaîne de services unique (SSC) avec différentes tailles de réseau de centre de données pour mesurer l'utilisation des ressources du réseau en termes de nombre d'utilisateurs servis et de taux de réduction. NACHOS montre son efficacité en termes de taux d'acceptation et de taux de réduction des violations de SLA.Cependant, ces travaux se concentrent uniquement sur le routage du trafic à l'intérieur des réseaux de centres de données sans prendre en compte le placement des chaînes de services. Ils ne considèrent pas non plus l'instanciation de plusieurs instances et les mécanismes de routage du trafic entre les instances synchronisées.

❖ **Beck & Botero (2015)** [26] traitent du placement et du chaînage des VNF et introduisent CoordVNF pour résoudre ce problème. La solution proposée vise à réduire l'utilisation sur les liens. Elle s'adapte aux conditions changeantes de l'infrastructure une option de chaînage réalisable pour les VNF. Les instances VNF ne sont considérées que dans certains cas d'utilisation lorsque vDPI divise le trafic en trafic TCP et non TCP. L'allocation dynamique des ressources et la demande future de trafic ne sont pas prises en compte dans ce travail. CoordVNF adopte le concept de backtracking où il tente à plusieurs reprises de trouver les options d'affectation des instances VNF. Si pour une raison quelconque, les algorithmes sont incapables de finir des solutions réalisables pour l'enchainement, il tente itérativement d'enchainer d'autres options d'enchainement. Cette proposition ne considère les instances multiples que lorsqu'une inspection virtuelle approfondie des paquets (vDPI) divise le trafic en trafic TCP et non TCP.

❖ Dans **(Vizarreta et al. (2017))** [27], les auteurs abordent le problème de la performance des VNFs par rapport aux middleboxes matérielles en essayant de garantir le débit minimal, la latence maximale de bout en bout, la disponibilité des ports et l'évitement des pertes de paquets. Les auteurs proposent un ILP qui trouve le placement des VNFs tout en garantissant les exigences SLA (c'est-à-dire le débit minimum et la disponibilité du service).

L'objectif principal de cette proposition est de minimiser le coût des ressources tout en tenant compte de la bande passante, des délais de propagation et de traitement, ainsi que de la disponibilité du service. Les auteurs proposent trois heuristiques gourmandes. Le premier algorithme trouve un encastrement réalisable pour la chaîne de services en explorant les chemins et sélectionne ensuite le chemin contraint par la QoS. Le deuxième algorithme évalue les nœuds candidats dans lesquels les VNFs sont intégrés, et sélectionne le nœud présentant le coût supplémentaire le plus faible (c'est-à-dire le coût d'installation de la licence). Le troisième algorithme trouve le chemin le plus court entre la source et la destination de la chaîne de services tout en tenant compte des contraintes de QoS. En outre, les auteurs évaluent les performances de l'heuristique proposée avec un algorithme de base qui trouve à peine les plus courts chemins sous contrainte de QoS entre la source et la destination.

Les résultats montrent que la disponibilité du service peut être garantie, mais que le risque de violation du SLA peut encore être élevé lorsque la disponibilité du service est négligée.

III.3 Analyse de solutions présentées dans la littérature

L'analyse de la littérature actuelle nous permet d'observer que cette dernière est assez peu fournie en ce qui concerne le problème du placement et du chaînage de VNF. En effet, nombreux sont les efforts sur le sujet du placement et du chaînage de VNF. Les travaux proposés se sont alors orientés sur plusieurs modèles de VNF, dans le but de déterminer le nombre d'instances VNFs nécessaires pour exploiter des SFC répondant à la demande de trafic élevé, à réduire les coûts totaux d'exploitation et de synchronisationen tenant compte des différents prix de l'énergie et des données échangées entre les instances VNFs mappées, et à améliorer les performances du réseau et augmenter le taux d'acceptation des demandes mappées. Les solutions de placement de VNF existantes modélisent le problème d'optimisation par une programmation linéaire en nombres entiers et utilisent des algorithmes heuristiques pour une résolution rapide.

III.3.1 À l'égard du problème de placement de VNF et des coûts

Les chercheurs dans [17], montrent que chaque moteur de placement configurable démontre un comportement différent, et fournit différentes stratégies de placement pour les fonctions réseau virtuelles. Cependant, le chaînage des VNF n'est pas pris en compte dans ce travail. En outre, les auteurs n'utilisent pas un nombre réaliste d'hôtes dans les expériences pour évaluer la demande de fonctions réseau dans l'environnement de production.

Les résultats dans [18], montrent que l'algorithme proposé basé sur la recherche Tabou surpasse les algorithmes gourmands en termes de taux d'acceptation, de coût total et de revenu total. Cependant, les algorithmes proposés sont incapables de s'adapter aux changements des conditions du réseau. De plus, l'hébergement de VNF dans une seule VM peut conduire à une sur-utilisation imprévisible des ressources. Sans compter que ces travaux ne prennent pas en compte la nécessité d'intégrer plusieurs instances VNFs pour répondre à une demande de trafic élevée.

Dans [20], Le modèle proposé réduit les délais de bout en bout et empêche le surprovisionnement des ressources. Cependant, le placement ne prend pas en compte le chemin le plus court entre la source et le POP dans lequel les VNF sont placés, et le nombre estimé d'instances pour chaque VNF dans un SFC n'est pas considéré dans ce travail.

Et les résultats dans [22], montrent que les algorithmes proposés réalisent une réduction significante en termes de rapport concurrentiel, de coûts opérationnels et d'instanciation. Cependant, ces travaux n'identifient pas clairement comment les charges de travail de fond affectent la décision de placement de VNF. Ils considèrent également plusieurs instances de VNF sans évaluer la capacité de traitement des VNFs.

III.3.2 Par rapport au problème de routage du trafic des chaines de services

Dans [25], la programmation dynamique résout le problème en utilisant l'induction inverse, qui enregistre le nombre de fois qu'une chaîne de services passe par les mêmes liens. Cependant, ces travaux

se concentrent uniquement sur le routage du trafic à l'intérieur des réseaux de centres de données sans prendre en compte le placement des chaînes de services.

Les résultats de travail [23], montrent des compromis entre le débit total restant, le nombre de nœuds utilisés et la latence. De plus, ils montrent la possibilité de trouver un placement qui optimise les trois paramètres si les ressources sont suffisantes. Mais, ce travail ne prend pas en compte le chaînage des NFs de manière dynamique en fonction de la demande des locataires.

Les résultats dans [27], montrent que la disponibilité du service peut être garantie, mais que le risque de violation du SLA peut encore être élevé lorsque la disponibilité du service est négligée.

Au regard de l'état actuel de la littérature sur le sujet émergent du placement et du chaînage de VNF, nous nous efforcerons dans ce travail à intégrer quelque aspects précédemment mentionnés. Le cadre du placement et du chaînage de VNF que nous utilisons dans ce travail permettra d'avoir trouvé un placement pour les VNF. Le but sera, plus précisément, d'optimiser les performances, tout en minimisant les couts.

III.4 Projets et plate-formes NFV

Ces dernières années, une multitude de plates-formes de déploiement et d'expérimentation NFV open-source ont émergé, apportant de nouvelles fonctions pour le plan de données, de contrôle et de gestion. Ces plates-formes modifient considérablement les architectures réseau existantes. Nous présenterons dans cette section les plus pertinentes d'entre elles de notre point de vue.

OPNFV Open Platform for Networks FunctionsVirtualization (OPNFV) [28] est un projet open-source fondé, mis en place et hébergé par la fondation linux (Linux Foundation), et composé de fournisseurs d'équipements et de Fournisseur d'Accès à Internet (FAI). L'objectif est d'établir une référence open-source intégrée, de qualité opérateur, qui peut être utilisée pour valider des solutions NFV interopérables multi-fournisseurs. OPNFV vise à valider les spécifications des normes existantes, d'apporter des améliorations aux projets open source pertinents en amont et de développer les nouvelles fonctionnalités nécessaires dans les projets OPNFV. En particulier, il se concentre sur la mise en œuvre des exigences du NFV [29] fournies par l'ETSI.

OpenMANO [30,31] est un projet open source mené par Telefonica, qui vise à mettre en œuvre le groupe de travail ETSI NFV MANO et à traiter les aspects liés aux performances et à la portabilité des fonctions réseau virtualisées. Son architecture est composée de trois modules logiciels :

Openmano (composant clé) : C'est une implémentation de la référence de l'orchestration des NFV (Network Functions Virtualisation Orchestrator (NFV-O)) tel que le définit l'ETSI, qui permet la création de scénarios de réseaux virtuels complexes. Il s'interface avec un VIM pour NFV par le biais de son Application Programming Interface (API) et offre une interface vers le nord, basée sur REST, où des services NFV sont offerts, y compris la création et la suppression de services de réseau ou VNF.

Openvim : C'est une implémentation de la référence NFV-PER001 [32] de l'ETSI, avec support pour des performances élevées et prévisibles. OpenVIM s'interface avec les nœuds de calcul de l'infrastructure et un contrôleur SDN pour offrir des capacités de calcul et de mise en réseau pour déploiement des machines virtuelles. Il offre aussi une interface OpenStack [33] (OpenVim API), où des services Cloud améliorés sont offerts, y compris la création, la suppression et la gestion des images, des instances et des réseaux.

Openmano-gui: C'est l'interface graphique (web) pour interagir avec l'API Openmano d'une manière graphique, une interface en ligne de commande est également disponible.

ONAP Open Network Automation Platform (ONAP) [34], c'est un autre projet open-source également soutenu par la fondation Linux. C'est le résultat de la fusion de deux autres initiatives open-source sur le Management Orchestration (MANO) OPEN-O et OpenECOMP. La première version (Release 1.0.0 Amsterdam) a été publiée en Novembre 2017, la version actuelle (premier semestre 2019) est la release Dublin. C'est une plate-forme complète pour l'orchestration et l'automatisation en temps réel des fonctions de réseaux physiques et virtuels, qui permettra aux fournisseurs et développeurs de logiciels,de services réseaux et de services Cloud d'automatiser rapidement de nouveaux services et de prendre en charge la gestion du cycle de vie complet. Il existe plusieurs autres projets et plates-formes telsque CloudNFV, X-MANO, XOS dont l'objectif est de proposer une seule fonction réseau virtualité comme un routage, un DeepPacket Inspection (DPI) ou encore un contrôle d'accès.

OpenDaylight est un projet collaboratif open-source hébergé par la Linux Fondation. Avec une plateforme de contrôleur modulaire et flexible au cœur du projet. Ce contrôleur est implémenté entièrement en logiciel, et il est contenu dans sa propre machine virtuelle Java. Pour cette raison, il peut être déployé sur tout matériel ou système d'exploitation prenant en charge Java [35]. L'objectif du projet est d'accélérer l'adoption de SDN et de créer une base solide pour NFV [36].

C'est le contrôleur le plus riche en fonctionnalités parmi les projets open-sources du moment. En s'appuyant sur OSGi (Open Services Gateway initiative (OSGi)), ce contrôleur est très modulaire et a d'excellents temps d'exécution pour le chargement des paquets. Ce projet a l'avantage d'une interface graphique assez conviviale pour les développeurs d'applications avec une large communauté de contributeurs et une documentation abondante. En outre, il propose une architecture distribuée idéale pour le déploiement du SDN dans un environnement réaliste. Enfin, ils offrent une interface avec OpenStack via le module Neutron permettant ainsi l'orchestration des NFV et cloud qui reste un impératif majeur pour la gestion des ressources virtuelles.

III.5 Conclusion

Ce chapitre a mené une étude pour les principales approches qui traitent du placement et du chaînage de VNF. La littérature a été classée en fonction de l'orientation principale des travaux portant sur deux parties du placement de VNF, à savoir la réduction des coûts de placement et de chaînage de VNF et le placement de VNF et le routage du trafic de services.

Le chapitre suivant sera consacré à la réalisation d'un réseau virtuel Multi ISP Cloud sur GNS3 et l'intègrer avec OpenDaylight comme un projet SDN open source pour prouver les avantages des VNFs.

Chapitre IV : Implémentation de la solution de placement de VNF

IV.1 Introduction

Dans ce chapitre nous allons commencer par l'implémentation d'un réseau virtuel Multi ISP Cloud sur lelogiciel GNS3. Ensuite, on l'intègre avec OpenDaylight comme un projet SDN open source prenant en charge ces protocoles tels que BGP-LS. Enfin, nous avons abordé l'intégration d'OpenDaylight et le réseauvirtuel qu'on a réalisé pour prouver les avantages des VNFs.

IV.2 Réseaux définis par logiciel

Mise en réseau définie par logiciel (SDN) [37]. En bref, le SDN peut être défini comme "*une architecture de réseau émergente où le contrôle du réseau est découplé et séparé du mécanisme de transmission et est directement programmable*". L'architecture SDN apporte un contrôle logiquement centralisé, appelé contrôleur SDN, qui a une vue globale du réseau sous-jacent. Les dispositifs de bas niveau deviennent strictement éléments de transmission sans aucune fonction de contrôle. Ils reçoivent toutes les instructions du contrôleur via une interface spécialisée. Les nouveaux protocoles sont définis pour la communication entre le contrôleur et les commutateurs configurables. L'un des protocoles les plus connus utilisés par les contrôleurs SDN est OpenFlow.

La couche d'acheminement ou de plan de données est placée au bas de l'architecture SDN. La couche plan de données est constituée de dispositifs de commutation connectés de manière filaire ou sans fil. Les dispositifs du réseau effectuent un ensemble d'opérations élémentaires d'acheminement. Ce sont des dispositifs programmables et ils se comportent en fonction des instructions envoyées par le contrôleur.

La communication entre le contrôleur SDN et les commutateurs programmables est rendue possible par les interfaces de programme d'application (API) sud [37]. Ces interfaces facilitent le contrôle efficace du réseau et permettent au contrôleur SDN d'apporter des modifications dynamiques au plan de transfert en temps réel. Par exemple, le contrôleur SDN peut ajouter ou supprimer une entrée dans la table de transfertpar le biais de l'interface sud.

Le contrôleur SDN est le "cerveau" du réseau. Il s'agit d'un point de contrôle centralisé qui gère le contrôle du flux vers les dispositifs du réseau en dessous et la logique des applications au-dessus. Le contrôleur SDN crée une vue abstraite du réseau, y compris des statistiques et l'état du réseau, et l'envoie au niveau de l'application. Le plan de données peut être contrôlé à partir de niveau de l'application. Une fois que l'instruction du niveau supérieur est envoyée, le contrôleur la prend et la transmet aux dispositifs de niveau inférieur. L'API nord présente une interface d'abstraction de réseau aux applications qui se trouvent au sommet de la pile SDN. Cette interface permet de programmer le réseau au niveau de l'application. L'API nord est certainement la partie la plus critique de l'architecture SDN. Le contrôleur SDN est apprécié pour les applications innovantes qu'il peut prendre en charge, et l'API nord doit répondreaux exigences des applications.

Les API Northbound sont également utilisées pour connecter le contrôleur SDN à la pile d'automatisation et aux plateformes d'orchestration.

La couche application comprend l'ensemble des applications qui exploitent les fonctions offertes par l'API nord pour mettre en œuvre la logique opérationnelle et le contrôle du réseau. Depuis le niveau applicatif, on peut surveiller le réseau physique et contrôler le routage, les pare-feu, les équilibreurs de charge, etc. Toutes les commandes provenant de la couche applicative sont traduites en instructions orientées vers le sud qui programment le comportement des dispositifs de transfert.

IV.3 Environnement logiciel

- **OpenDaylight** : Le projet OpenDaylight (contrôleur OpenDaylight, ODL) est un projet SDN open source régi par la Fondation Linux [38]. Les contrôleurs SDN open source permettent de tester facilement le réseau et prennent en charge la virtualisation du réseau. L'architecture des solutions open source est généralement modulaire, ce qui signifie que le contrôleur est constitué de modules enfichables qui exécutent différentes fonctions réseau. Les projets open source offrent des possibilités de développement et de personnalisation. Aujourd'hui, il existe de nombreux projets open source lancés pour un développement plus poussé, tels que ONOS, Pox, Ryu, etc.

  Le projet OpenDaylight a été annoncé dès 2013 dans le but d'accélérer le développement du SDN et son adoption par l'industrie. ODL est basé sur le langage de programmation Java et prend en charge la norme OpenFlow [39]. Certaines des entreprises qui contribuent au développement d'ODL sont Cisco, Juniper Networks, VMware, Microsoft, Ericsson, etc. Quinze versions sont actuellement disponibles :

  Hydrogène (4 février 2014), Hélium (29 septembre 2014), Lithium (29 juin 2015), Béryllium (22 février 2016), Bore (21 septembre 2016), Carbone (26 mai 2017), Azote (7 septembre 2017), Nitrogen (May,2018), Oxygen (Dec,2018), Flourine (Jun,2019), Neon (Dec,2019), Sodium (Aug,2020), Magnesium (Jul,2020), Aluminum (Nov,2020).

Architecture ODL

L'architecture détaillée d'OpenDaylight varie selon les versions [40]. L'architecture ODL simplifie présentée dans la figure IV.1 est commune à toutes les versions :

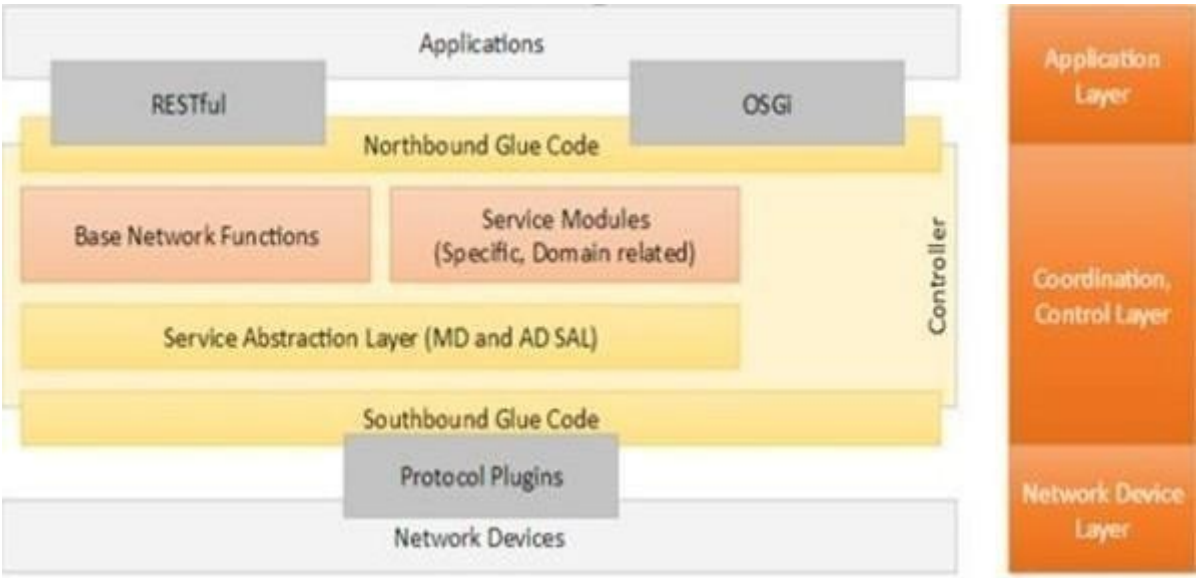


***Figure IV.1**: Architecture détailler d'opendaylight.*

Comme tous les contrôleurs SDN, ODL se compose de trois parties principales :

1. API en direction du sud
2. Couche de fonction de contrôle
3. API en direction du Nord

- **GNS3 :** Graphical Network Simulator-3 (abrégé en GNS3) est un émulateur de logiciel de réseau publié pour la première fois en 2008[41]. Il permet la combinaison de dispositifs virtuels et réels, utilisés pour simuler des réseaux complexes. Il utilise le logiciel d'émulation **Dynamips** pour simuler **Cisco IOS**. [41] GNS3 est utilisé par de nombreuses grandes entreprises, notamment Exxon, Walmart, AT&T et la NASA**.** et également populaire pour la préparation des examens de certification professionnelle des réseaux. En 2015, le logiciel a été téléchargé 11 millions de fois [42].

  Dans ce travail, la topologie du réseau est implémentée dans un environnement GNS3, composé de routeurs CSR1000v et de routeurs Cisco IOU L3. Les détails de l'installation des routeurs et de la configuration de l'environnement seront présentés dans la partie suivante.

- **La machine virtuelle GNS3** : est une machine virtuelle qui exécute GNS3 et à laquelle le client GNS3 installé sur notre PC se connecte. C'est une fonctionnalité introduite dans GNS3 à partir de la version 1.4, mais la machine virtuelle n'est pas vraiment quelque chose de nouveau. La fonctionnalité de machine virtuelle GNS3 facilite simplement les choses, élimine la complexité d'avoir à installer une machine virtuelle, à configurer le réseau et à intégrer cette machine virtuelle avec GNS3 en tant que machine distante. Avec la VM GNS3, le client GNS3 établit la « connexion» à virtualbox ou vmware (figure IV.2).

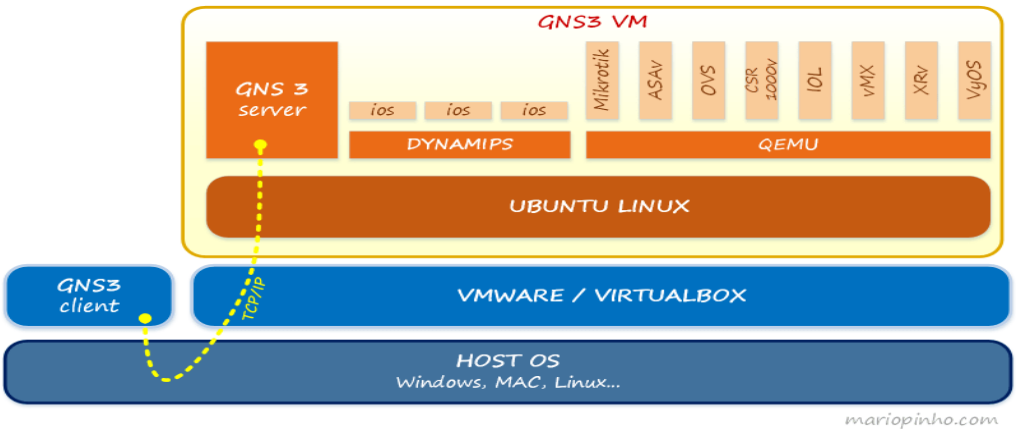


***Figure IV.2**: Architecture montre GNS3 et GNS3vm.*

- **Le Cisco Cloud Services Router 1000V (CSR 1000V)** [43] est un routeur et une plate- forme de services réseau dans un facteur de forme virtuel qui est destiné à être déployé dans les centres de données virtuels et en nuage. Il est optimisé pour servir de passerelle WAN à locataire unique ou à locataires multiples. Grâce aux fonctions de sécurité et de mise en réseau éprouvées du logiciel Cisco IOS® XE, le CSR 1000V permet aux entreprises d'étendre de manière transparente leur réseau étendu aux nuages hébergés par des fournisseurs externes et aux fournisseurs de nuages d'offrir à leurs locataires des services de mise en réseau de classe entreprise.

- **Cisco IOU L3** est un routeur pur et simple. L2 est un commutateur qui est capable d'agir comme un commutateur L3 mais qui est initialement appelé un dispositif L2 et qui est donc un commutateur.

- **Docker UBUNTU** c'est un conteneur Linux dispose d'une interface en ligne de commande simple qui facilite la prise en main pour les nouveaux utilisateurs. Il contient un environnement de virtualisation au niveau du système d'exploitation qu'il est possible d'installer sur de nombreux systèmes basés sur Linux.

- **VMware Workstation Pro** : Permet aux professionnels techniques de développer, tester, démontrer et déployer des logiciels en exécutant simultanément plusieurs systèmes d'exploitation en tant que machines virtuelles (VM) sur un seul PC. Il peut allouer plusieurs cœurs de processeur, des giga-octets de mémoire principale et de mémoire graphique à chaque machine virtuelle, que la machine virtuelle réside sur un PC personnel ou sur un Cloud d'entreprise privé.

- **Postman** : est un environnement de développement d'API complet qui vous aide à gérer vos API à chaque étape du développement, de la conception et des tests à la publication de la documentation et de la surveillance des API. Postman est rapidement devenu l'un des outils d'API les plus utilisés par les développeurs du monde entier. un outil qui permet de construire et de tester rapidement des requêtes http directement depuis une interface graphique.

IV.4 Préparation de l'environnement de travail

Pour réaliser notre travail nous avons d'abord passé par plusieurs étapes :

IV.4.1 Installation de GNS3

Premièrement, nous avons téléchargé et installé GNS3, toutes les versions de l'émulateur sont disponibles sur la page d'accueil de GNS3. La version installée dans notre cas est la 2.1.3, GNS3 existe pour Windows, Linux et Mac. Vous devriez donc trouver ce qu'il vous faut. En ce qui nous concerne, nous somme sous Windows 10. La configuration recommandée par GNS3 est de 8GB de RAM, un disque dur SSD et Un processeur 4 cœurs. Évidemment, GNS3 utilisant les ressources de votre machine, plus votre configuration système sera puissante, plus vous pourrez ajouter de nouveaux routeurs, Switchs et autres serveurs virtuels et donc effectuer des tests à plus grande échelle.

IV.4.2 Installation de GNS3 VM

Il nous faut donc installer GNS3 VM et pour cela, nous avons besoin d'un hyperviseur. Il est possible d'installer GNS3 VM sur plusieurs hyperviseurs, dans notre cas on a utilisé VMware Workstation Pro (version 16).

Apres avoir téléchargé GNS3vm de même version que GNS3 et sur le même site, nous importons donc la machine virtuelle GNS3 dans le programme VMware comme le montre la figure IV.3 :

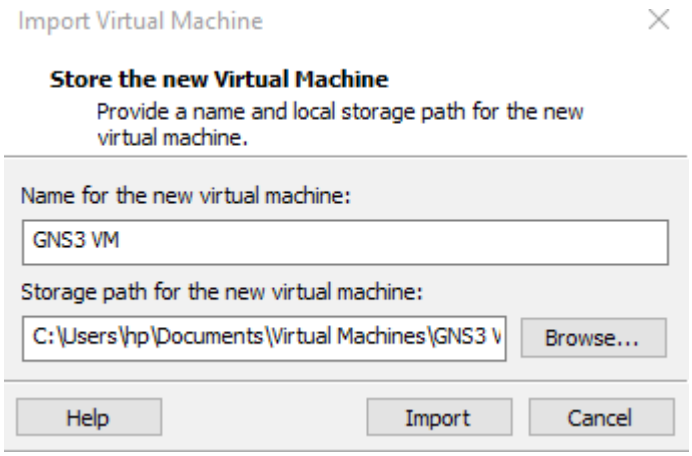


Figure IV.3 : *Importer la machine virtuelle GNS3.*

Pour faire le lien avec GNS3 il faut :

- Ouvrir GNS3.
- Cliquant sur Cancel dans les "Setup wizard"
- Cliquant sur Edit -> Préférences -> GNS3 VM
- Cliquant sur "Enable the GNS3 VM" et on sélectionne "VMware Worstation /Player" dans le champ Virtualize engine.
- Cliquant sur Refresh et GNS3 VM apparaît dans VM name. S'il n'apparait pas, c'est que GNS3 n'arrive pas à communiquer avec VMware. Dans ce cas, le mieux est de redémarrer votre ordinateur et de retenter.
- Cliquant sur Apply.

Si la configuration est bonne, GNS3 VM se lance automatiquement.

IV.4.3 Vérification de connectivité entre GNS3 et GNS3VM

Dès lors que le « feu vert » est affiché (au milieu à droite de votre écran), cela veut dire que GNS3 à bien détecté et va donc utiliser GNS3 VM pour émuler les IOS (figure IV.4).

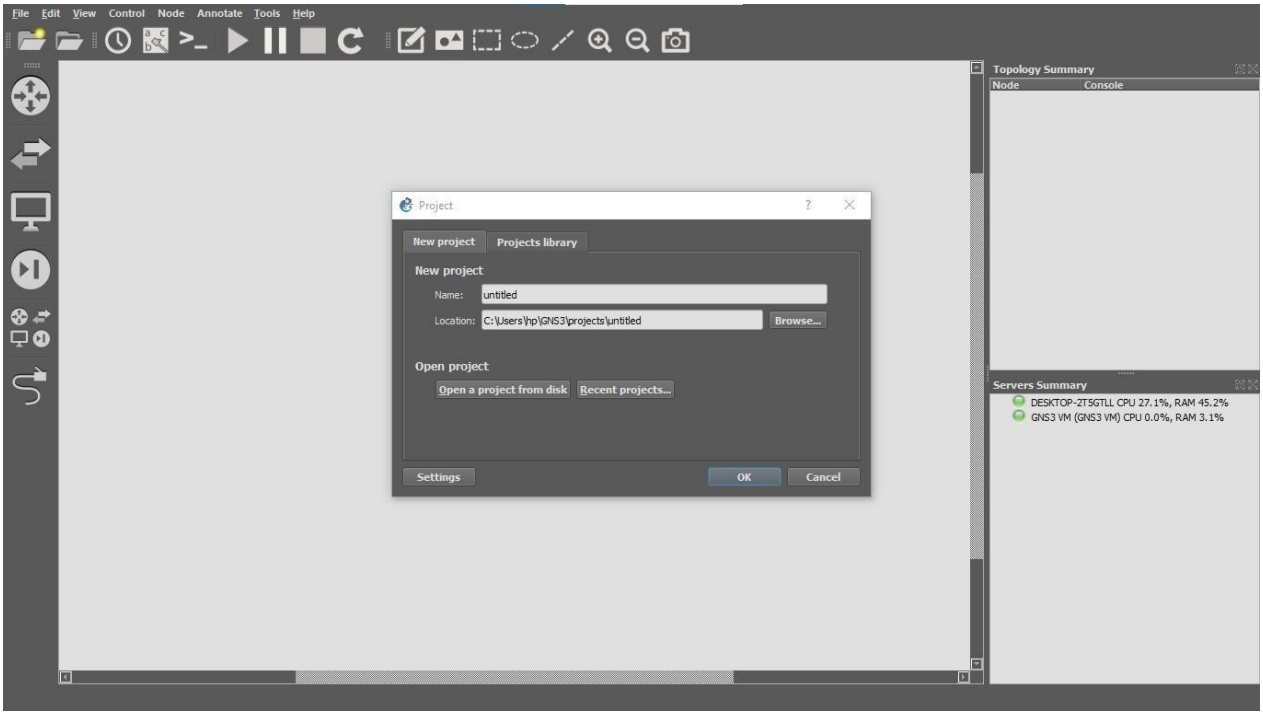


Figure IV.4 : *Vérification de connectivite entre GNS3 et GNS3 VM.*

IV.4.4 Installation des images du routeur CSR1000v et de routeur Cisco IOU L3 sur le GNS3

Afin d'exécuter l'image de Cisco CSR100v dans GNS3, d'abord nous devons télécharger l'image et l'importer à GNS3. Dans notre cas on a travaillé avec la version (16.12.03), on a décidé de lui donner 3Go de RAM à la machine virtuelle qui devrait être une exigence minimale pour exécuter le CSR1000v.

La même chose pour le routeur Cisco IOU L3. Le téléchargement d'Appliance à partir de GNS3 Marketplace: on a accéder à la place de marché GNS3 et télécharger l'Appliance GNS3 à partir de navigateur Web. Ensuite, on a l'importer dans GNS3 et suivons l'assistant d'installation. GNS3 vous guidera sur les fichiers dont vous avez besoin et ce qu'il faut faire pour le faire fonctionner.

IV.4.5 Préparation d'OpenDaylight

OpenDaylight a la possibilité d'installer des fonctionnalités spécifiques nécessaires à la mise en œuvre d'un réseau défini par logiciel. Pour l'intégration avec le réseau multi ISP, ODL doit prendre en charge le protocole BGP-LS et les fonctionnalités de traitement des requêtes API Neutron. Les étapes suivantes montrent comment installer OpenDaylight sur Ubuntu LTS 16.04 :

Tout d'abord, on a téléchargé et installé le contrôleur OpenDaylight Magnésium SDN sur Linux Ubuntu (avant d'exécuter l'ODL, le kit de développement Java doit être installé sur le système). D'autres programmes nécessaires doivent être installés sur Linux, par exemple, Google Chrome, Postman (comme serveur REST). Les étapes suivantes montrent comment installer OpenDaylight sur Ubuntu LTS 16.04 :

1. Préparer le système d'exploitation
2. Installer le JRE Java
3. Télécharger OpenDaylight
4. Installer OpenDaylight
5. Démarrez le contrôleur avec la commande suivante :

```
root@ODL:~# cd karaf-0.12.2
root@ODL:~/karaf-0.12.2# ./bin/karaf
```

6. Une fois la console ODL ouverte, Installer la fonctionnalité nécessaire à OpenDaylight.

À ce stade, ODL est prêt à traiter les demandes de mise en réseau.

IV.5 Implémentation et mise en place des réseaux

IV.5.1 Travail à réaliser

GNS3 est maintenant fonctionnel, il nous est dorénavant possible de glisser/déposer des routeurs de la série CSR1000v et Cisco IOU L3 sur le plan de travail.

Nous avons donc ajouté quatre routeurs CSR1000v R1 à R4, et deux routeurs de la gamme Cisco IOU L3 et deux Customer. Nous connectons tous les nœuds par un câblage virtuel disponible sur GNS3

de manière très similaire aux réseaux ISP de conception Cisco. Cette topologie est montrée à la figure IV.5.

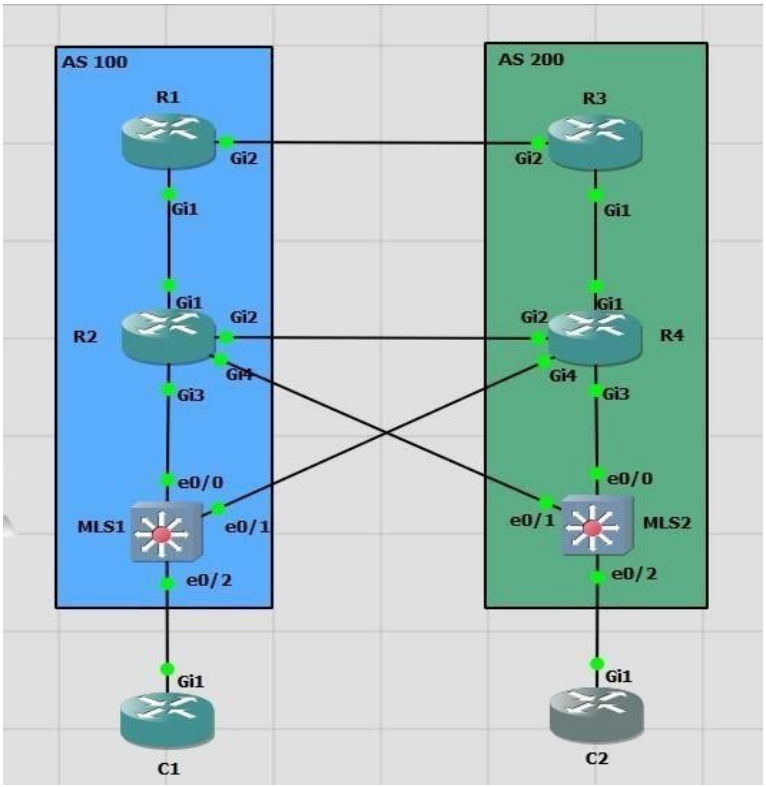


Figure IV.5 : *La topologie de réseau multi ISP.*

IV.5.2 Plan d'adressage

Le tableau IV.1 résume le plan d'adressage de notre réseau multi ISP.

Routeurs	Interfaces	Adresse IP	Masque
R1	Loopback	1.1.1.1	255.255.255.255
	Gi1	20.20.20.1	255.255.255.0
	Gi2	10.10.10.0	255.255.255.0
R2	Loopback	2.2.2.2	255.255.255.255
	Gi1	20.20.20.2	255.255.255.0
	Gi2	40.40.40.1	255.255.255.0
	Gi3	50.50.50.1	255.255.255.0
	Gi4	60.60.60.1	255.255.255.0
R3	Loopback	3.3.3.3	255.255.255.255
	Gi1	30.30.30.1	255.255.255.0
	Gi2	10.10.10.2	255.255.255.0
R4	Loopback	4.4.4.4	255.255.255.255
	Gi1	30.30.30.2	255.255.255.0
	Gi2	40.40.40.2	255.255.255.0
	Gi3	80.80.80.1	255.255.255.0
	Gi4	70.70.70.1	255.255.255.0
	loopback	5.5.5.5	255.255.255.255

MLS1	e0/0	50.50.50.2	255.255.255.0
	e0/1	70.70.70.2	255.255.255.0
	e0/2	192.168.1.2	255.255.255.0
MLS2	loopback	6.6.6.6	255.255.255.255
	e0/0	80.80.80.2	255.255.255.0
	e0/1	60.60.60.2	255.255.255.0
	e0/2	192.168.2.2	255.255.255.0
C1	Loopback	7.7.7.7	255.255.255.255
	Gi1	192.168.1.1	255.255.255.0
C2	loopback	8.8.8.8	255.255.255.255
	Gi1	192.168.2.1	255.255.255.0

Tableau IV.1 : *Plan d'adressage de la topologie.*

IV.5.3 Configuration

Tout d'abord, on a configuré les interfaces comme sur la figure ci-dessous. A titre d'exemple de configuration d'interface, voici la configuration de l'interface du routeur.

Après avoir eu un réseau connecté, nous avons poussé quelques commandes pour obtenir les paramètres initiaux sur les périphériques. Tout d'abord, sur chaque périphérique de ce réseau, nous avons configuré les interfaces comme sur la figure ci-dessous. Chaque routeur possède une adresse Loopback et des interfaces Gigabit Ethernet pour l'interconnexion des routeurs. A titre d'exemple de configuration d'interface, voici la configuration de l'interface du routeur a été utilisé pour tous les routeurs (figure IV.6):

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface Loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)#
R1(config-if)#interface GigabitEthernet1
R1(config-if)# ip address 20.20.20.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
R1(config-if)#interface GigabitEthernet2
R1(config-if)# ip address 10.10.10.1 255.255.255.0
R1(config-if)#no sh
```

Figure IV.6 : *Configuration des adresses aux interfaces de routeur.*

L'acheminement de la couche 3 doit alors être activé globalement sur les Multilayer Switch:

```
MLS1(config)# iprouting
```

a) Configuration du routage statique

Pour que les routeurs puissent communiquer avec tous les réseaux, on doit configurer le routage statique, en appliquant ceci sur R1 on obtient (figure IV.7) :

```
R1(config)#ip route 2.2.2.2 255.255.255.255 20.20.20.2
R1(config)#ip route 3.3.3.3 255.255.255.255 10.10.10.2
R1(config)#ip route 4.4.4.4 255.255.255.255 10.10.10.2
R1(config)#ip route 5.5.5.5 255.255.255.255 20.20.20.2
R1(config)#ip route 6.6.6.6 255.255.255.255 10.10.10.2
R1(config)#ip route 7.7.7.7 255.255.255.255 20.20.20.2
R1(config)#ip route 8.8.8.8 255.255.255.255 192.168.1.2
R1(config)#ip route 8.8.8.8 255.255.255.255 20.20.20.2
R1(config)#
```

Figure IV.7 : *Configuration de routage statique.*

La même chose pour les routeurs R2, R3, R4 et les multilayer switchs MLS1 et MLS2.

Pour les deux Customers on configure la route par défaut pour permettre aux utilisateurs d'accéder aux ISP (figure IV.8).

```
C1(config)#
C1(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.2
C1(config)#
```

Figure IV.8 : *Configuration de la route par défaut.*

b) Configuration du protocole OSPF

Ensuite, nous avons configuré le protocole IGP dans le réseau. Nous avons utilisé OSPF. En prennent toujours R1 comme exemple d'application de notre configuration, on obtient (figure IV.9) :

```
R1(config)#router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 1.1.1.1 0.0.0.0 area 0
R1(config-router)# network 20.20.20.0 0.0.0.255 area 0
*Aug  7 14:45:03.959: %OSPF-6-DFT_OPT: Protocol timers for fast convergence are Enabled.
R1(config-router)#
```

Figure IV.9 : *Configuration de protocole OSPF.*

c) Configuration du protocole BGP

Nous allons configurer le BGP sur tous les routeurs (R1, R2, R3, R4) et les Multilayer Switchs (MLS1, MLS2) pour obtenir une connectivité complète entre les routeurs. La figure IV.10 vous montrer la configuration complète pour R1 :

```
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 100
R1(config-router)# bgp router-id 1.1.1.1
R1(config-router)# neighbor 2.2.2.2 remote-as 100
R1(config-router)# neighbor 2.2.2.2 update-source Loopback0
R1(config-router)#
R1(config-router)# neighbor 3.3.3.3 remote-as 200
R1(config-router)# neighbor 3.3.3.3 ebgp-multihop 2
R1(config-router)# neighbor 3.3.3.3 update-source Loopback0
R1(config-router)#
R1(config-router)# neighbor 5.5.5.5 remote-as 100
R1(config-router)# neighbor 5.5.5.5 update-source Loopback0
R1(config-router)#
R1(config-router)#  network 10.10.10.0 mask 255.255.255.0
R1(config-router)#  network 20.20.20.0 mask 255.255.255.0
R1(config-router)#
```

Figure IV.10 : *Configuration de protocole BGP.*

d) Test de connectivité

Pour assurer la connectivité entre tous les composants de notre topologie, un ping a été envoyé entre C1 et C2 et réussi comme le confirme la figure IV.11:

C1

```
C1#PING 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 11/149/678 ms
C1#
```

Figure IV.11 : *Test de connectivité.*

Enfin, notre Lab est validé, passant alors à intégrer cette topologie avec le contrôleur OpenDaylight.

IV.6 Intégration d'OpenDaylight avec GNS3

IV.6.1 Configuration de la topologie du réseau

Après avoir installé le conteneur Docker dans le GNS3, nous l'avons fait glisser vers l'espace de travail, puis nous avons fait glisser un commutateur Ethernet avec le nœud NAT qui a permis au conteneur Docker d'obtenir une adresse IP, et d'avoir une connectivité internet à partir de l'ordinateur local, car par défaut le nœud NAT exécute un serveur DHCP avec un pool prédéfini dans la plage 192.168.122.0/24. La figure IV.12 montre les composants de cette topologie.

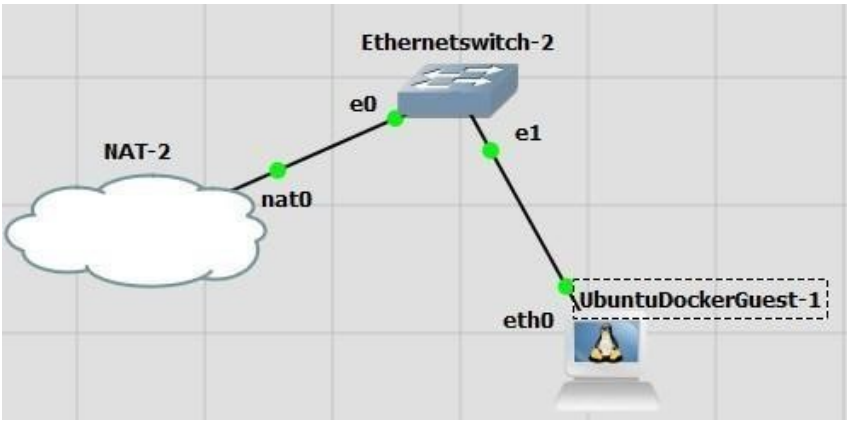


Figure IV.12 : *Conteneur Docker connecté au nœud NAT sur l'espace de travail dans GNS3.*

Lorsque nous avons démarré la topologie, nous avons ouvert la console depuis le conteneur Docker une fois qu'il a démarré, puis nous l'avons configuré pour utiliser le protocole DHCP. Donc la commande **`nano/etc/network/interfaces`** a été passée dans la console afin d'éditer ce fichier et de décommenter les deux dernières lignes puis de l'enregistrer comme nous pouvons le voir dans la Figure IV.13.

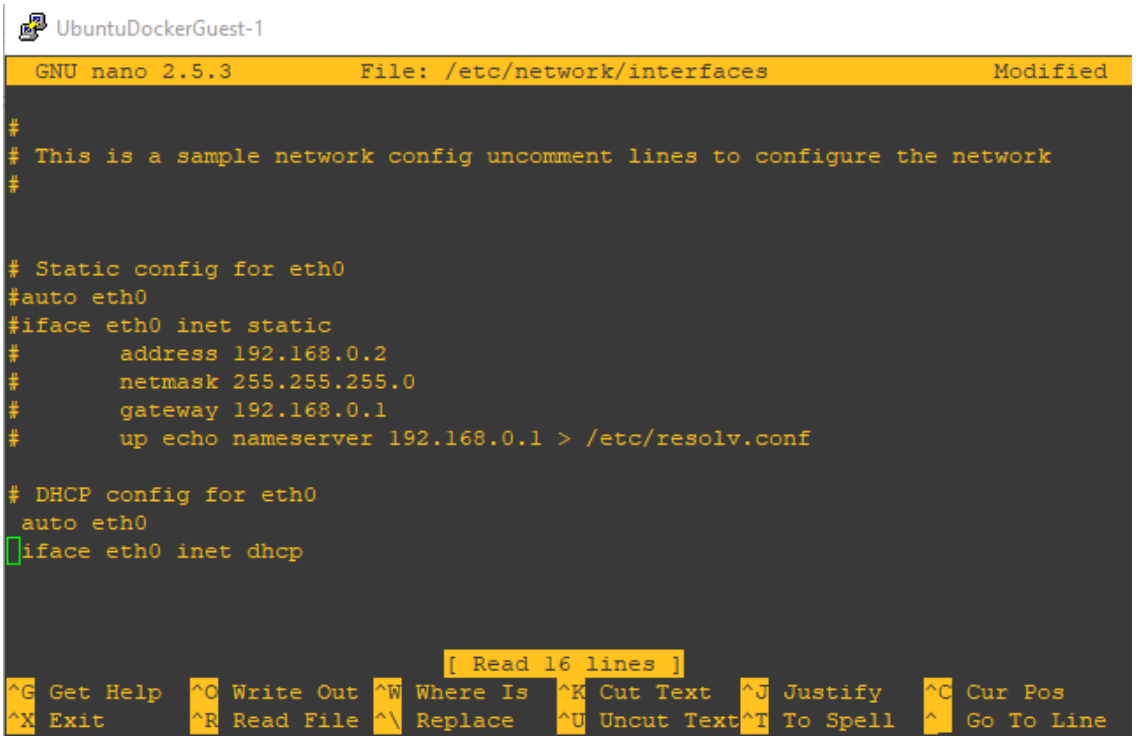


Figure IV.13 : *Configuration de l'interface réseau de Docker à l'aide de l'éditeur Nano.*

Après cela, nous avons rechargé le nœud Docker, afin de permettre au client DHCP sur celui-ci d'envoyer des messages `Discover` au nœud NAT qui lui offrira une adresse IP, comme nous pouvons le voir sur la Figure IV.14. Pour confirmer la commande `ifconfig` doit être passée dans la console.

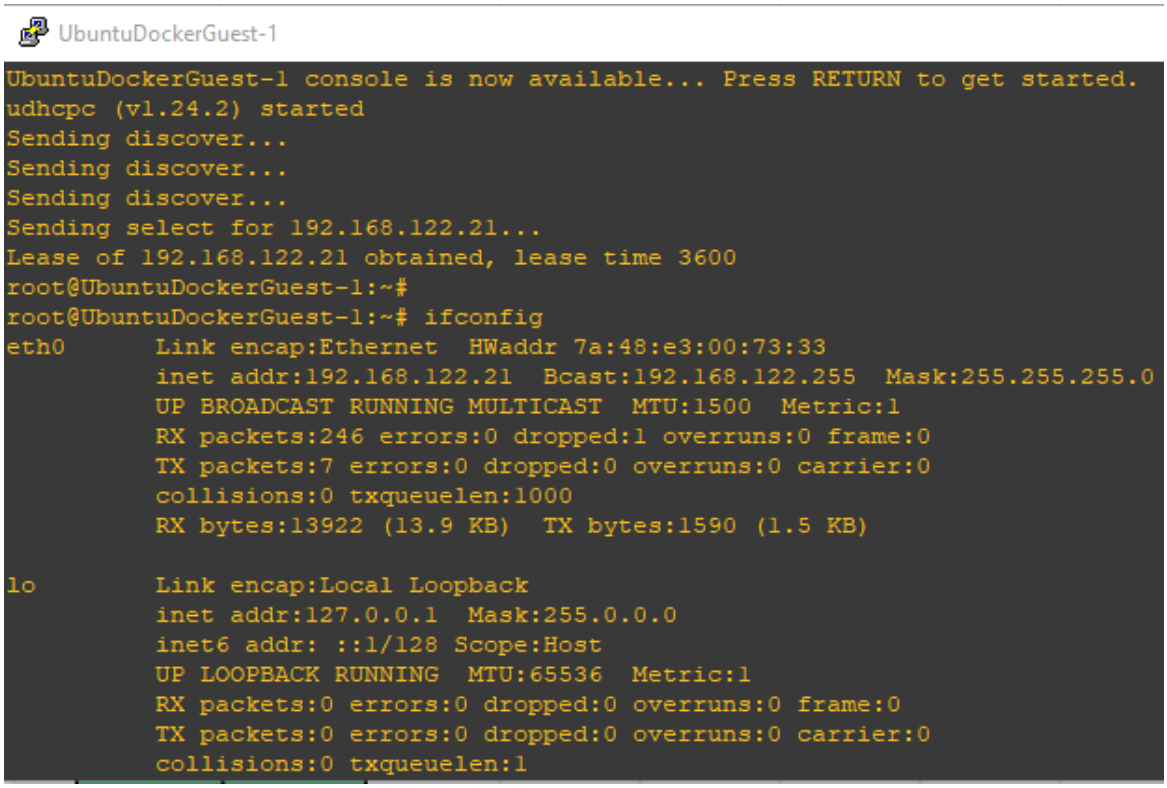


Figure IV.14 : *L'obtenu d'une adresse IP pour Docker à partir de DHCP.*

IV.6.2 Les fonctionnalités nécessaires pour l'intégration avec OpenDaylight

- Télécharger OpenDaylight à partir de [44]. Dans ce travail, nous avons utilisé la distribution Magnesium, qui par défaut n'est livrée avec aucune fonctionnalité. La raison pour laquelle nous avons choisi cette version qui contient déjà de nombreuses applications compatibles avec cette version. En installant ODL, on peut personnaliser l'environnement et ajouter les fonctionnalités dont on a besoin. Pour installer ODL, on peut suivre ce guide d'installation [44].
- Exécuter le conteneur karaf bin/karaf
- Installer les fonctionnalités nécessaires en tapant *feature:install odl-bgpcep-bgp, odl- bgpcep-pcep, odl-restconf et odl-netconf-topology***.** La fonctionnalité odl-restconf active l'accès de l'API REST au MD-SAL, y compris le magasin de données et la fonctionnalité odl-bgpcep-bgp Prend en charge le protocole Border Gateway (y compris la distribution d'état de liaison) en tant que source d'informations de topologie L3. La figure IV.15 Représente les fonctionnalités nécessaires pour l'intégration.

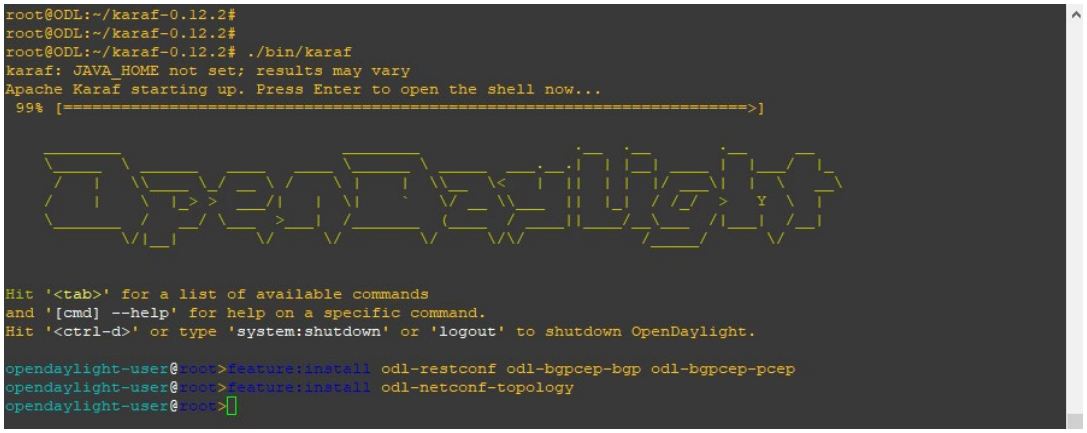


***Figure IV.15**: les fonctionnalités nécessaires pour l'intégration.*

- Lors de l'installation de la fonctionnalité, un certain nombre de fichiers xml seront générés dans etc/opendaylight/karaf et seront reconfigurés.
- Configurez l'adresse IP d'OpenDaylight par sudo nano /etc/network/interfaces. L'adresse IP que nous avons utilisé pour ODL est 192.168.152.20.

Maintenant nous assignons l'adresse IP à l'interface gi3 de R1 pour interconnecter ODL avec R1 . On choisit le R1 parce qu'il est le routeur désigné DR (Designated Router).

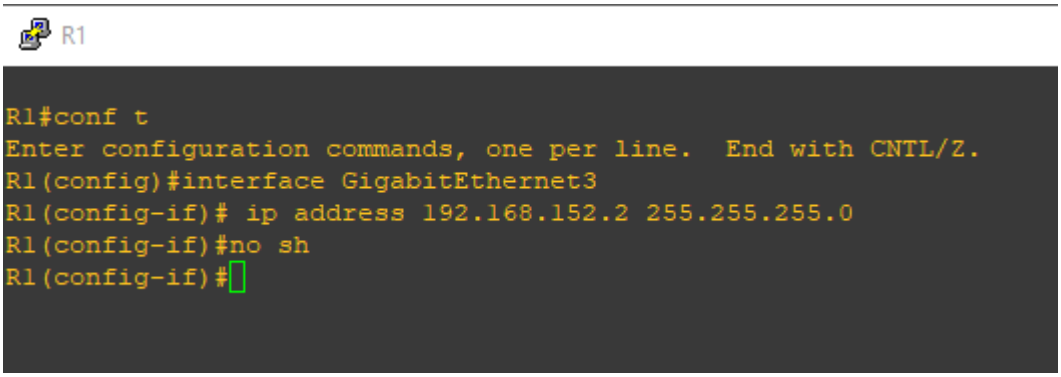


Figure IV.16 : *Configuration de l'interface Gi3*

Ensuite, nous relions l'interface Gi3 du routeur au Switch connecté à ODL comme il est indiqué à la topologie de la figure IV.17.

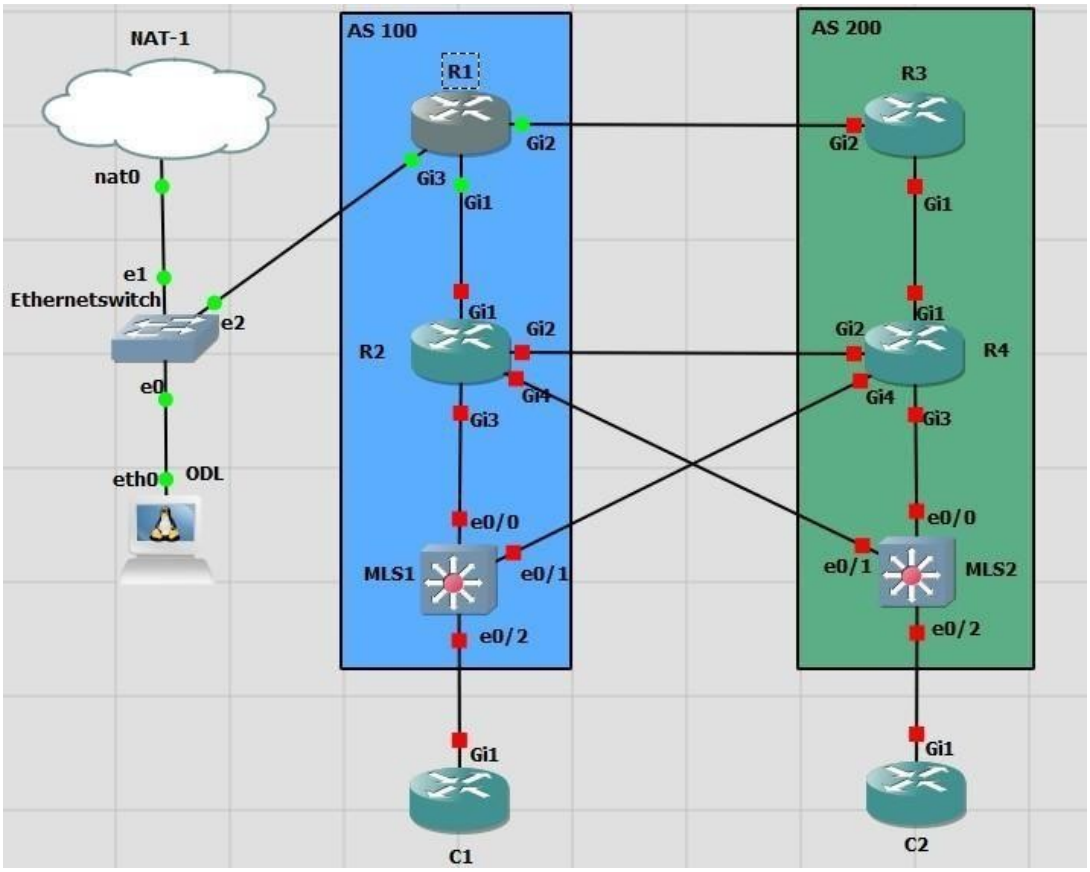


Figure IV.17 : *Interconnexion de R1 avec le contrôleur ODL.*

IV.6.3 Teste de la connectivité

Le teste de la connectivité entre le contrôleur OpenDaylight et le routeur désigné R1 est représenté par la figure IV.18.

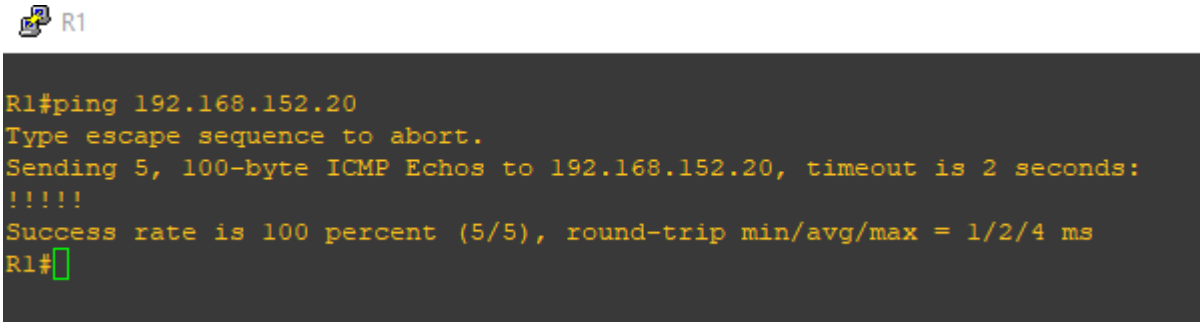


Figure IV.18 : *Teste de connectivité entre R1et ODL.*

IV.7 La configuration BGP-LS

IV.7.1 Types des interfaces de programmation d'application

Les applications API Southbound peuvent être ouvertes ou propriétaires. Elles facilitent un contrôle efficace du réseau et permettent au contrôleur SDN d'apporter des modifications dynamiques en fonction des demandes et des besoins en temps réel. Il existe de nombreuses API, standards ou propriétaires,

pouvant agir sur différents éléments de l'équipement telles que SNMP, NETCONF, YANG, BGP-LS, PCE, OpenFlow…

Le protocole BGP-LS rassemble toutes les informations relatives à la topologie du réseau à partir d'un protocole de routage IGP (ISIS ou OSPF) dans un domaine de réseau, et transmet ces informations au contrôleur SDN.

IV.7.2 Client API REST basée sur le WEB

OpenDaylight utilise l'interface RESTful northbound pour communiquer avec les couches supérieures. Pour être en mesure de recevoir une topologie abstraite et de donner des commandes au contrôleur SDN, il faut utiliser un client RESTful. Il y a beaucoup de clients RESTful basés sur le web disponibles aujourd'hui, cependant nous avons utilisé Postman.

Postman a une interface facile à utiliser. L'action de la demande peut être n'importe quel HTTP classique, les commandes GET, POST, PUT et DELETE.

Les données peuvent être représentées au format JSON et XML. Pour connecter le serveur au client, il faut spécifier l'adresse de localisation du serveur.

Dans ce travail, Postman a été utilisé pour récupérer la topologie du réseau depuis le contrôleur, On peut facilement voir les informations que le contrôleur reçoit réellement du réseau sous-jacent, comme les paramètres des liens et des nœuds (LSDB). Nous avons utilisé Postman pour configurer BGP-LS dans le réseau. Cela se fait en postant une requête XML au contrôleur via l'interface de Postman.

IV.7.3 Configuration de BGP-LS

BGP-LS n'a besoin que d'une seule session entre PCE et PCC, PCC étant le routeur et PCE l'ODL. Tout d'abord, nous configurons le BGP sur le routeur R1. Nous escortions les étapes suivantes:

1. **Configuration de BGP Link-State avec OSPF**

Dans ce réseau, nous avons choisi R1 pour être un locuteur BGP et pour redistribuer toutes les informations IGP à OpenDaylight (figure IV.19).

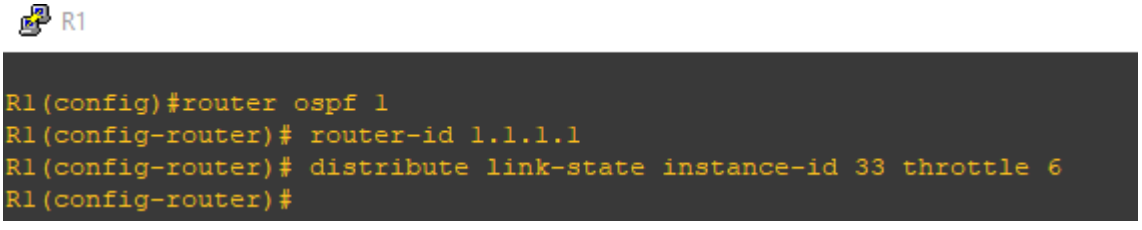


Figure IV.19: *Configuration de BGP Link-State avec OSPF.*

La ligne 3 copie les informations de l'état des liens OSPF dans l'état des liens BGP. Cette escortions commande n'est insérée que dans le routeur qui fonctionne comme locuteur BGP et qui parle au pair BGP d'OpenDaylight. En bref, les informations sur l'état des liens recueillies par le protocole IGP dans le

domaine (qui dans ce projet est le protocole OSPF) seront transportées par le protocole BGP-LS vers le Peer BGP d'ODL.

2. Configuration de BGP Link-State avec BGP

La configuration de BGP est présentée dans la figure IV.20.

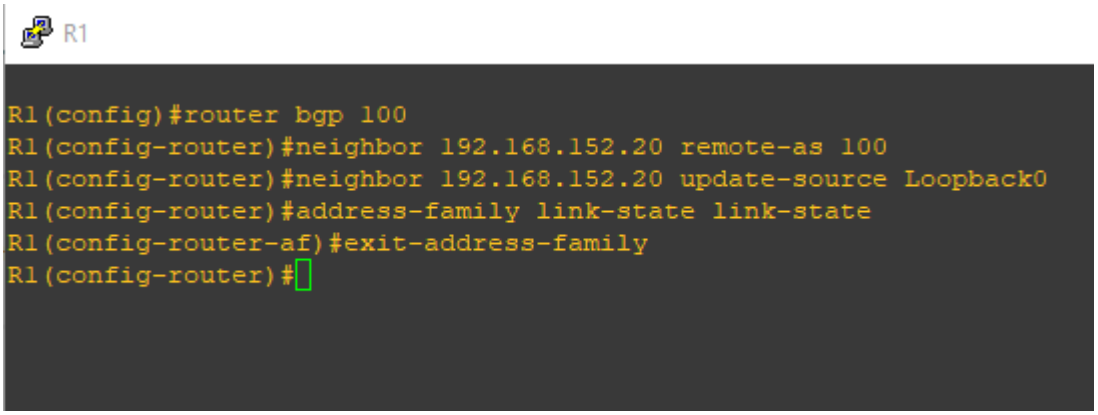


Figure IV.20: *Configuration de BGP Link-State avec BGP.*

Il faut d'abord ouvrir une instance BGP. Ensuite, nous avons spécifié l'adresse Loopback du routeur comme ID du routeur. Le voisin BGP est le contrôleur SDN, spécifiéà la ligne 2. Le contrôleur SDN appartient à un AS distant.

3. Vérification des configurations

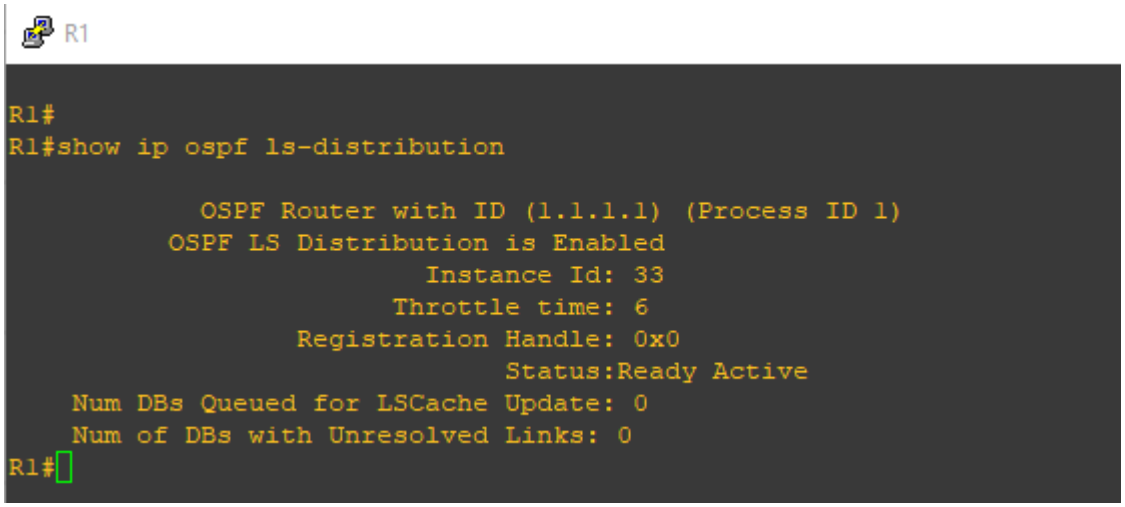


Figure IV.21: *Vérification de configuration OSPF avec BGP-LS.*

R1

```
     Network          Next Hop            Metric LocPrf Weight Path
 *>    [V][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1]]
                      0.0.0.0                  0         32768 i
 *>    [V][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]]
                      0.0.0.0                  0         32768 i
 *>    [V][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5]]
                      0.0.0.0                  0         32768 i
 *>    [V][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1d20.20.20.1]]
                      0.0.0.0                  0         32768 i
 *>    [V][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5d50.50.50.2]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1]][R[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1d20.20.20.1]][L
[i20.20.20.1][n20.20.20.1]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]][R[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1d20.20.20.1]][L
[i20.20.20.2][n20.20.20.1]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]][R[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5d50.50.50.2]][L
[i50.50.50.1][n50.50.50.2]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5]][R[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5d50.50.50.2]][L
[i50.50.50.2][n50.50.50.2]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1d20.20.20.1]][R[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1]][L
[i20.20.20.1][n20.20.20.1]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1d20.20.20.1]][R[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]][L
[i20.20.20.1][n20.20.20.2]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5d50.50.50.2]][R[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]][L
[i50.50.50.2][n50.50.50.1]]
                      0.0.0.0                  0         32768 i
 *>    [E][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5d50.50.50.2]][R[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5]][L
[i50.50.50.2][n50.50.50.2]]
                      0.0.0.0                  0         32768 i
 *>    [T4][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][rl.1.1.1]][P[o0x01][pl.1.1.1/32]]
                      0.0.0.0                  0         32768 i
 *>    [T4][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r2.2.2.2]][P[o0x01][p2.2.2.2/32]]
                      0.0.0.0                  0         32768 i
 *>    [T4][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5]][P[o0x01][pl92.168.1.0/24]]
                      0.0.0.0                  0         32768 i
 *>    [T4][O][I0x21][N[c100][b0.0.0.0][a0.0.0.0][r5.5.5.5]][P[o0x01][p5.5.5.5/32]]
                      0.0.0.0                  0         32768 i
```

Figure IV.22: *Vérification de configuration BGP avec BGP-LS.*

Apres nous passons à la configuration BGP-LS sur ODL, Premièrement nous vérifions la connectivité entre ODL et la machine physique Windows (figure IV.23) à partir de laquelle nous exécuterons les appels d'API REST à Opendaylight à l'aide de l'application POSTMAN.

Invite de commandes

```
Microsoft Windows [version 10.0.19042.1110]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\hp>ping 192.168.152.20

Envoi d'une requête 'Ping'  192.168.152.20 avec 32 octets de données :
Réponse de 192.168.152.20 : octets=32 temps<1ms TTL=64
Réponse de 192.168.152.20 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.152.20 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.152.20 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.152.20:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Users\hp>
```

Figure IV.23: *Vérification de connectivité entre le PC et l'ODL.*

Ensuite, en utilisant l'API REST, nous activerons l'ID de routeur BGP avec la famille Link State. Après avoir démarré Postman, cliquant simplement sur un nouvel onglet (avec un signe +), puis nous commençons par sélectionner **PUT** dans la liste déroulante. Ensuite, juste à droite de PUT, nous ajoutons l'URL du service Web, dans ce cas :

http://192.168.152.20:8181/restconf/config/openconfig-network-instance:network-instances/network-instance/global-bgp/openconfig-network-instance:protocols

Paramètres de connexion doivent être définis comme suit :

- Autorisation : nom d'utilisateur et mot de passe de l'ODL qui est admin/admin par défaut.
- Header> type de contenu : application/xml
- En-tête>accept : application/xml

Ensuite, nous devons remplir la partie corps de notre message.

```
{
    "protocols": {
        "protocol": [
   {
                "identifier": "openconfig-policy-types:BGP",
                "name": "bgp-example",
                "bgp-openconfig-extensions:bgp": {
"global": {
                        "afi-safis": {
                            "afi-safi": [
{
                                "afi-safi-name": "bgp-openconfig-extensions:LINKSTATE"
                                }
                            ]
                        },
                        "config": {
                            "router-id": "192.168.152.20",
                            "as": 100
                        }
                    },
                    "neighbors": {
                        "neighbor": [
                            {
                                "neighbor-address": "1.1.1.1",
                                "transport": {
                                    "config": {
                                        "remote-port": 179,
                                        "passive-mode": false
                                    }
},
                                "config": {
                                    "peer-type": "INTERNAL"
                                },
                                "timers": {
                                    "config": {
                                        "hold-time": 90,
                                        "connect-retry": 10
                                    }
```

```
                                },
                                "afi-safis": {
                                    "afi-safi": [
                                        {
                                  "afi-safi-name":"openconfig-bgp-types:IPV4-UNICAST"
},
                                        {
                                "afi-safi-name":"bgp-openconfig-extensions:LINKSTATE"
                                        }
                                    ]
                                }
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```

Ici, nous avons créé une instance BGP pour le nœud ODL. L'ID du routeur est l'IP de notre ODL où l'ODL fonctionne.

L'adresse du voisin est l'endroit où mes nœuds GNS fonctionnent, nous n'avons pas besoin de donner tous les nœuds, juste DR est suffisant (figure IV.24).

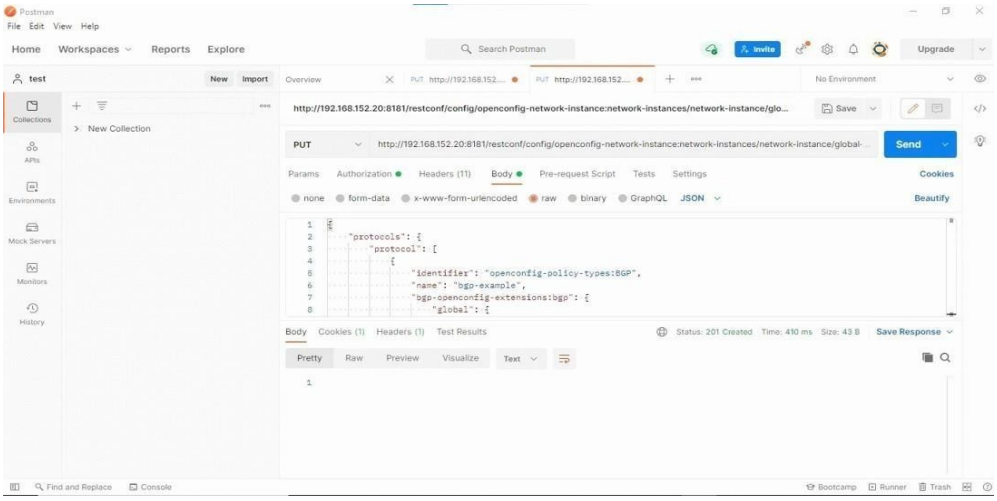


***Figure IV.24**: création d'une instance BGP pour le nœud ODL.*

Maintenant, cliquant sur **Send** pour envoyer le RPC ci-dessus afin de créer une instance de BGP dans ODL.

On fait la même chose pour créer un autre nœud, nous sélectionnons **PUT** dans la liste déroulante. Ensuite, nous ajoutons l'URL du service Web, dans ce cas :

http://192.168.152.20:8181/restconf/config/network-topology:network-topology/

Après nous cliquons sur l'onglet **Corps** et collons la requête XML suivante:

```
<network-topology xmlns="urn:TBD:params:xml:ns:yang:network-topology">
<topology>
```

```
<topology-id>topology-netconf</topology-id>
<node>
<node-id>R1</node-id>
<host xmlns="urn:opendaylight:netconf-node-topology">192.168.152.20</host>
<password xmlns="urn:opendaylight:netconf-node-topology">cisco</password>
<username xmlns="urn:opendaylight:netconf-node-topology">cisco</username>
<port xmlns="urn:opendaylight:netconf-node-topology">830</port>
<tcp-only xmlns="urn:opendaylight:netconf-node-topology">false</tcp-only>
<keepalive-delay xmlns="urn:opendaylight:netconf-node-topology">10</keepalive-delay>
</node>
</topology>
</network-topology>
```

Dans le champ **<node-id>** on a donné le nom de notre nœud R1. ODL connaîtra notre nœud avec ce nom. Le champ **<hostname>** sera le nom d'utilisateur pour notre nœud, et dans le champ **<password>,** on a mis le mot de passe pour le même nœud. Et on a gardé le port 830 tel qu'il est...

Ce RPC va créer un autre nœud sous Network-topology (figure IV.25), Apres envoyer le RPC ci-dessus onobtient ça :

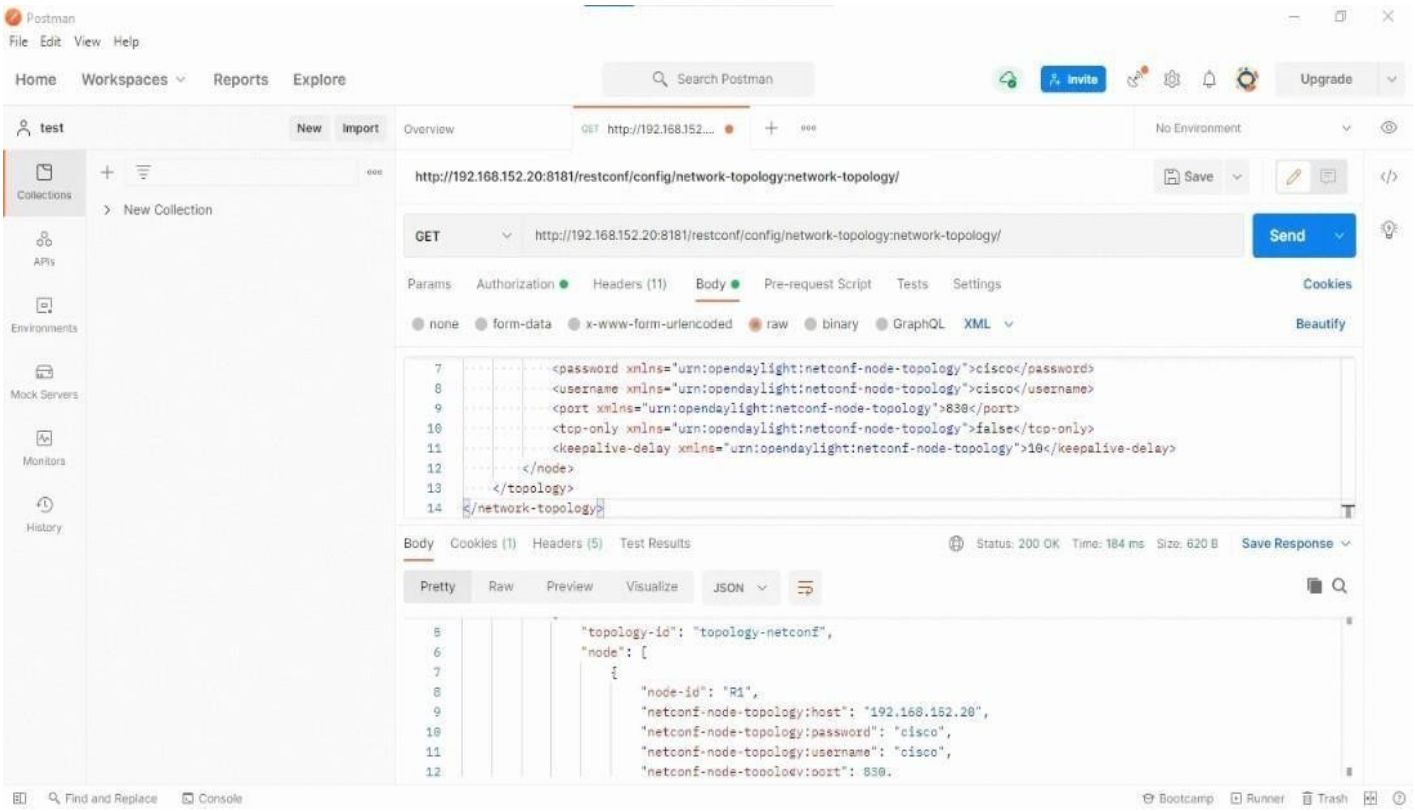


Figure IV.25: *Création d'un autre nœud.*

Pour vérifier que le nœud R1 a été créé, on tape cette adresse URL http://192.168.152.20:8181 /rests/data/network-topology:network-topology/topology=topology-netconf/node=R1 dans le navigateur et on obtient l'arbre de document affiché par la figure IV.26.

Figure IV.26: *Vérification de la création de R1.*

IV.8 Conclusion

Ce travail a été mis en place pour étudier le paradigme de virtualisation des fonctions réseaux NFV. Dans un premier temps, nous avons réalisé la topologie du réseau l'implémentation a pris en compte la création du réseau multi ISP dans GNS3, la configuration du contrôleur SDN et sa connexion au réseau.

L'utilisation de GNS3 comme plateforme gratuite, et sa possibilité d'être exécutée sur différents ordinateurs dans le laboratoire, ainsi que le contrôleur OpenDaylight C'est le contrôleur le plus riche en fonctionnalités parmi les projets open-sources du moment.

Conclusion générale

Pour conclure, nous présentons dans ce qui suit une brève revue de ce que nous avons réalisé au cours de notre mémoire. Ensuite nous présentons les limites de notre implémentation.

Dans ce travail, nous avons étudié l'un des principaux défis auxquels sont confrontés les fournisseurs de services. Dans les environnements Cloud, les fournisseurs de Cloud offrent leur infrastructure, composée d'un centre de données (CD) composés d'équipements de calcul et de réseautage. La virtualisation des fonctions de réseau (NFV), en tant que technologie émergente, suit le même concept et l'emmène à un niveau supérieur en tirant parti technologie de virtualisation pour virtualiser les fonctions réseau. Ce défi consiste à trouver le placement le plus approprié de ces VNF dans le l'infrastructure Cloud.

Le problème de placement VNF, un peu d'efforts ont été faits pour évaluer la capacitéVNF pour traiter de grandes quantités de trafic, en particulier avec la croissance du nombre d'utilisateurs finaux l'adoption des services Cloud. Un seul VNF pourrait suffire à traiter un certain nombre de demandes avant qu'il ne devienne surchargé. Cela se produit surtout lorsqu'ily a une augmentation des demandes pour un certain service. Dans laquelle nous avons abordé quelques solutions pour répondre à ce problème.

Le travail de recherche présenté dans ce mémoire a débuté par généralités sur les réseaux informatiques. Ensuite, nous avons entré dans le contexte technique dont nous avons illustré les concepts de base tandis qu'on a décrit le problème du placement de VNF de manière très approfondie. Puis dans le chapitre suivant nous avons concentré sur les travaux connexes. Tout au long de ce chapitre, nous avons discuté des solutions proposées pour le placement des VNFs nous mettons en avant leurs objectifs et leurs limites. Nous avons classifié les travaux existants en deux catégories principales en fonction de leurs objectifs. Ces deux catégories sont le placement et la réduction des coûts des VNFs, et le routage du trafic des chaînes de services.

Enfin nous finalisons par une implémentation d'un réseau virtuel Multi ISP Cloud sur GNS3 et l'intégrer avec OpenDaylight comme un projet SDN open source pour prouver les avantages des VNFs.

NFV est une technologie émergente qui vise à découpler le logiciel du matériel et offre plusieurs avantages pour l'industrie. Malgré les progrès considérables réalisés dans la résolution des problèmes liésà la NFV, aucune des solutions existantes n'aborde le placement des chaînes de fonctions de services (SFC) lorsque plusieurs instances de VNF sont nécessaires pour faire face à la demande croissante de trafic.

Le thème de notre projet est très intéressant et nous a poussé à réfléchir à le concrétiser réellement mais hélas, lescapacités de recherche ont différés nos souhaits, mais il reste toujours un très bon projet réalisable à l'avenir.

Bibliographie

[1] M. Chiosi, S. Wright, and others. (2012). *Network Functions Virtualisation (NFV). SDN and OpenFlow* , World Congress.

[2] https://www.cours-gratuit.com/cours-reseau/cours-complet-sur-les-reseau-en-pdf?fbclid=IwAR2HKyYb3PFEJ1okH1uCQ3ZTR9hiPe_ljMlbLCGf-duv0y7rLSv0SliksaI, Consulté le 23/02/2021.

[3] https://www.institut-numerique.org/la-securisation-des-serveurs-des-donnees-de-lentreprise-sousisa-serveur-2006-525681a31e9d4?fbclid=IwAR3vpNWbLFRDVh-al0cqdWJjierRJUBJh, Consulté le 23/02/2021.

[4] Xiyang, Z., & Chuanqing, C. (2009). *Research on VLAN Technology in L3 Switch*. 2009 Third International Symposium on Intelligent Information Technology Application. doi:10.1109/iita.2009.498.

[5] https://waytolearnx.com/2019/06/qu-est-ce-qu-un-commutateur-reseau-switch.html?fbclid=IwAR2h5cThlz-OKsm8FPk6xrTncxLR3YnkcZkoC2WOD_BwIkCDEfaokAns6zQ, Consulté le 27/02/2021.

[6] https://www.memoireonline.com/04/12/5743/m_Virtualisation-d-un-reseau-intranet.html, Consulté le 03/03/2021.

[7] Mell, P. M. & Grance, T. (2011). SP 800-145. *The NIST Definition of Cloud Computing*. Gaithersburg, MD, United States: National Institute of Standards & Technology.

[8] Zhang, Q., Cheng, L. & Boutaba, R. (2010). *Cloud computing: state-of-the-art and research challenges*. Journal of internet services and applications, 1(1), 7–18.

[9] IBM. (2007). *Virtualization in Education*. Consulted at http://www-07.ibm.com/solutions/in/education/download/VirtualizationinEducation.pdf. Consulté le 13/03/2021.

[10] Carapinha, J. & Jiménez, J. (2009). *Network Virtualization: A View from the Bottom*. Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, (VISA '09), 73–80. Consulted at http://doi.acm.org/10.1145/1592648.1592660.

[11] La virtualisation des fonctions réseau, qu'est-ce que c'est ? (redhat.com), Consulté le 16/03/2021.

[12] http://community.brocade.com/t5/SDN-NFV/An-Introduction-ToThe-Growing-NFV-Movement/ba-p/25, Consulté le 02/04/2021.

[13] https://en.wikipedia.org/wiki/Software-defined_networking, Consulté le 11/04/2021.

[14] https://www.opennetworking.org/images/stories/downloads/sdnresources/white-papers/wp-sdn-newnorm.pdf, Consulté le 21/04/2021.

[15] https://www.sdxcentral.com/resources/sdn/inside-sdn-architecture/, Consulté le 22/04/2021.

[16] J.Halpern, C.Pignataro et al, "*Service function chaining (sfc) architecture*," in RFC 7665,2015.

[17] Clayman, S., Maini, E., Galis, A., Manzalini, A. & Mazzocca, N. (2014). *The dynamic placement ofvirtual network functions*. IEEE Network Operations and Management Symposium (NOMS).

[18] Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., De Turck, F. & Davy, S. (2015b, April). *Design and evaluation of algorithms for mapping and scheduling of virtual network functions*. IEEE Conference on Network Softwarization (NetSoft), pp. 1-9.

[19] Bari, M., Chowdhury, S. R., Ahmed, R. & Boutaba, R. (2015b, Nov). *On orchestrating virtual network functions*. IEEE International Conference on Network and Service Management (CNSM), pp. 50-56.

[20] Luizelli, M., Bays, L., Buriol, L., Barcellos, M. & Gaspary, L. (2015). *Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions*. IEEE International Symposium on Integrated Network Management (IM).

[21] Wang, X., Wu, C., Le, F., Liu, A., Li, Z. & Lau, F. (2016). *Online VNF scaling in datacenters.* IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 140–147.

[22] Wang, X., Wu, C., Le, F. & Lau, F. C. (2017). *Online Learning-Assisted VNF Service Chain Scaling with Network Uncertainties*. IEEE 10th International Conference on Cloud Computing (CLOUD), 2017, pp. 205–213.

[23] Mehraghdam, S., Keller, M. & Karl, H. (2014). *Specifying and placing chains of virtual network functions*. IEEE International Conference on Cloud Networking (Cloud Net).

[24] Moens, H. & De Turck, F. (2014, Nov). *VNF-P: A model for efficient placement of virtualized network functions*. International Conference on Network and Service Management (CNSM), pp. 418-423.

[25] Huang, P.-H., Li, K.-W. & Wen, C.-P. (2015, Oct). *NACHOS: Network-aware chains orchestration selection for NFV in SDN datacenter*. IEEE International Conference on Cloud Networking (CLOUDNET).

[26] Beck, M. T. & Botero, J. F. (2015, Dec). Coordinated Allocation of Service Function Chains. IEEE Global Communications Conference (GLOBECOM), pp. 1-6.

[27] Vizarreta, P., Condoluci, M., Machuca, C. M., Mahmoodi, T. & Kellerer, W. (2017). *QoS driven function placement reducing expenditures in NFV deployments*. IEEE International Conference on Communications (ICC), pp. 1–7.

[28] Opnfv. https://www.opnfv.org/. [Online], Consulté le 15/06/2021.

[29] Maryam Tahhan, Billy O'Mahony, and Al Morton. *Benchmarking Virtual Switches in the Open Platform for NFV (OPNFV)*. RFC 8204, September 2017.

[30] Github openmano. https://github.com/nfvlabs/openmano/. [Online]. Consulté le 20/06/2021.

[31] Openmano. http://www.tid.es/long-term-innovation/ network-innovation/telefonica-nfv-referencelab/openmano. [Online]. Consulté le 22/06/2021.

[32] Onap. https://www.onap.org/. [Online]. Consulté le 23/06/2021.

[33] Anat Bremler-Barr, Yotam Harchol, David Hay, and Yaron Koral. *Deep packet inspection as a service*. In Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, CoNEXT '14, pages 271–282, New York, NY, USA, 2014. ACM.

[34] Lionel Bertaux, Samir Medjiah, Pascal Berthou, Slim Abdellatif, Akram Hakiri, Patrick Gelard, Fabrice Planchou, and Marc Bruyère. *Software Defined Networking and Virtualization for Broadband Satellite Networks*. IEEE Communications Magazine, 53(3):pp. 54–60, March 2015.

[35] http://www.opendaylight.org/project/technical-overview, Consulté le 20/07/2021.

[36] https://en.wikipedia.org/wiki/OpenDaylight_Project, Consulté le 20/07/2021.

[37] Mise en œuvre du routage par segment et de l'ingénierie du trafic MPLS dans un réseau défini par logiciel basé sur l'émulateur de réseau GNS3 et le contrôleur OpenDaylight SDN

[38] https://www.sdxcentral.com/resources/sdn/sdn-controllers/opendaylightcontroller/, Consulté le 02/08/2021.

[39] Khattak, Zuhran Khan, Muhammad Awais, et Adnan Iqbal. (April 2015). *Performance evaluation of OpenDaylight SDN controller*, 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS). doi: 10.1109/PADSW.2014.7097868.

[40] http://thenewstack.io/sdn-series-part-vi-opendaylight/, Consulté le 08/08/2021.

[41] Hassine, Jameleddine ; Hamou-Lhadj, Abdelwahab (Sept. 2014). "*Toward a UCM- Based Approach for Recovering System Availability Requirements from Execution Traces*". In Amyot,

[42] Daniel ; Pau Fonseca i Casas ; Mussbacher, Gunter. *Analyse et modélisation des systèmes : Modèles et réutilisabilité*. 8e conférence internationale, SAM 2014. 8769. Valence, Espagne : Springer. pp. 48– 63. ISBN 9783319117430.

[43] Fogarty, Susan. "*Simulateur de réseau GNS3 améliore son jeu".* Network Computing. UBM Tech. Consulté le 10/08/2021

[44] How to install OpenDaylight as a Service on Ubuntu 18.04 LTS (soban.ski), Consulté le 12/08/2021.

Annexes

• Annexe 1: installation de GNS3

Exigences recommandées

Voici la configuration recommandée pour un environnement Windows GNS3 :

Article	Exigence
Système d'exploitation	Windows 7 (64 bits) ou version ultérieure
Processeur	4 cœurs logiques ou plus - AMD-V / RVI Series ou Intel VT-X / EPT
Virtualisation	Extensions de virtualisation requises. Vous devrez peut-être l'activer via le BIOS de votre ordinateur.
Mémoire	16 Go de RAM
Stockage	Disque SSD (SOLID-STATE Drive) avec 35 Go d'espace disponible
Notes complémentaires	La virtualisation des périphériques est gourmande en processeur et en mémoire. Plus c'est mieux, mais un appareil correctement configuré l'emporte sur la RAM et la puissance de traitement.

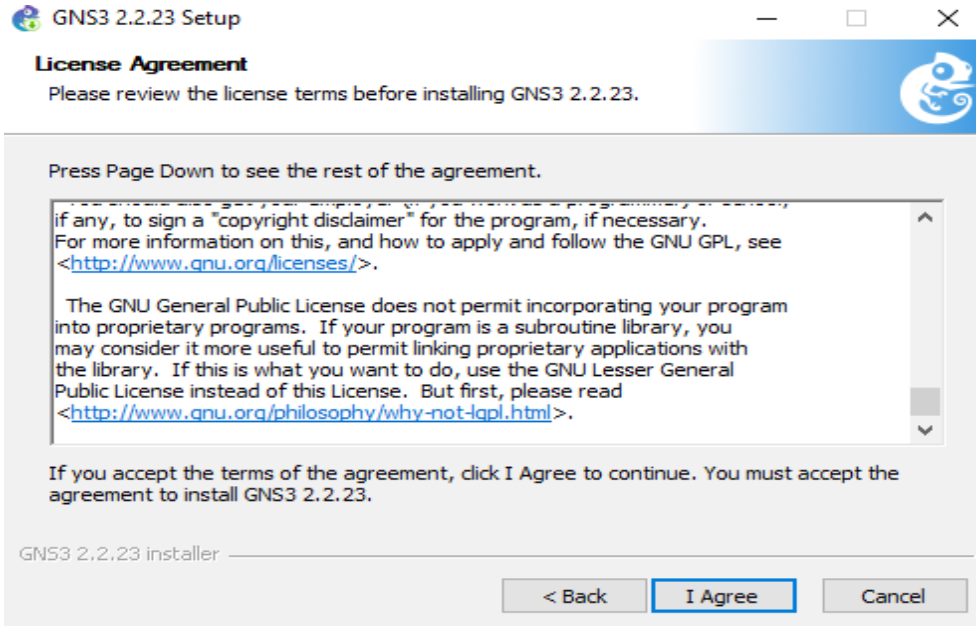
Etape1 : Téléchargement de GNS3

Utilisé le lien Software | GNS3. Pour accéder au page téléchargement et cliquer sur le bouton vert (Free Download)

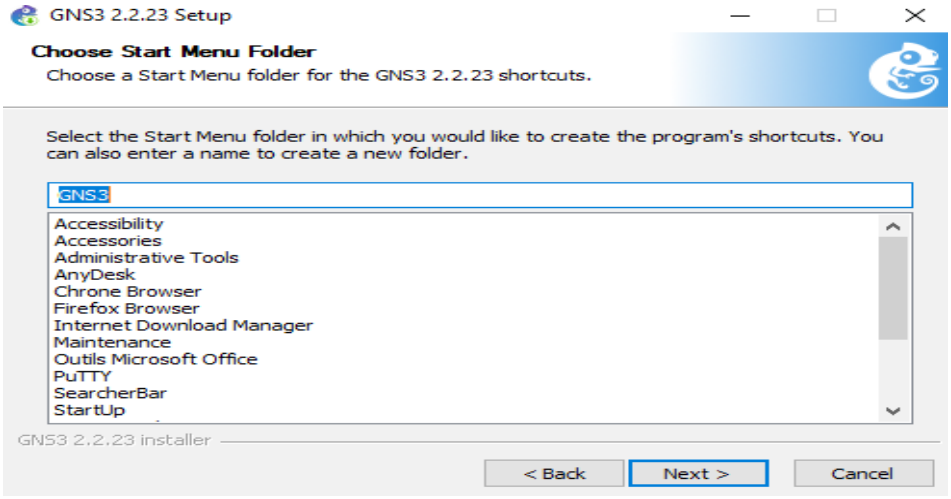
Etape 2 : Installation de GNS3

1. L'Assistant Installation de GNS3 s'affiche. Cliquant sur **Suivant** pour démarrer l'installation

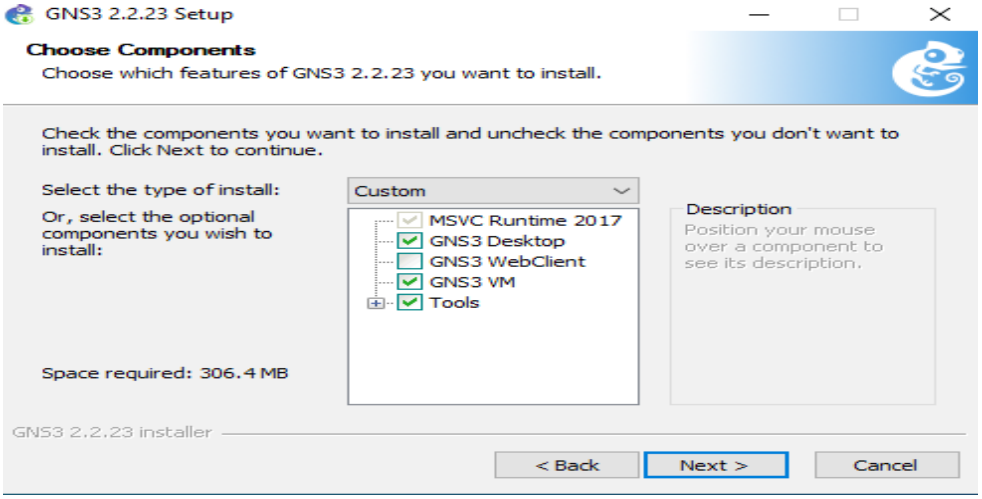
2. GNS3 est un logiciel open source gratuit distribué sous la licence publique générale GNU Version 3, cliquant sur le bouton **J'accepte** pour poursuivre l'installation :



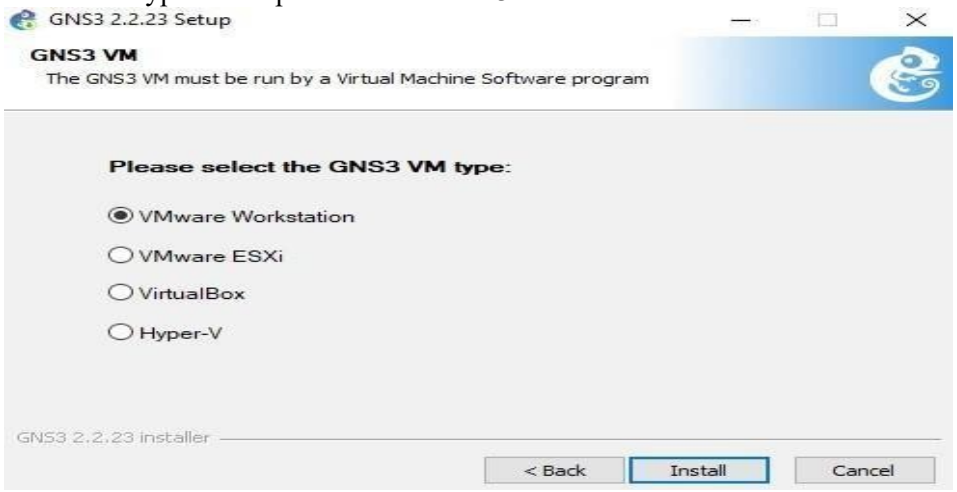
3. Autorisé GNS3 pour créer un dossier Menu Démarrer avec le nom par défaut GNS3 en cliquant sur le bouton Suivant.



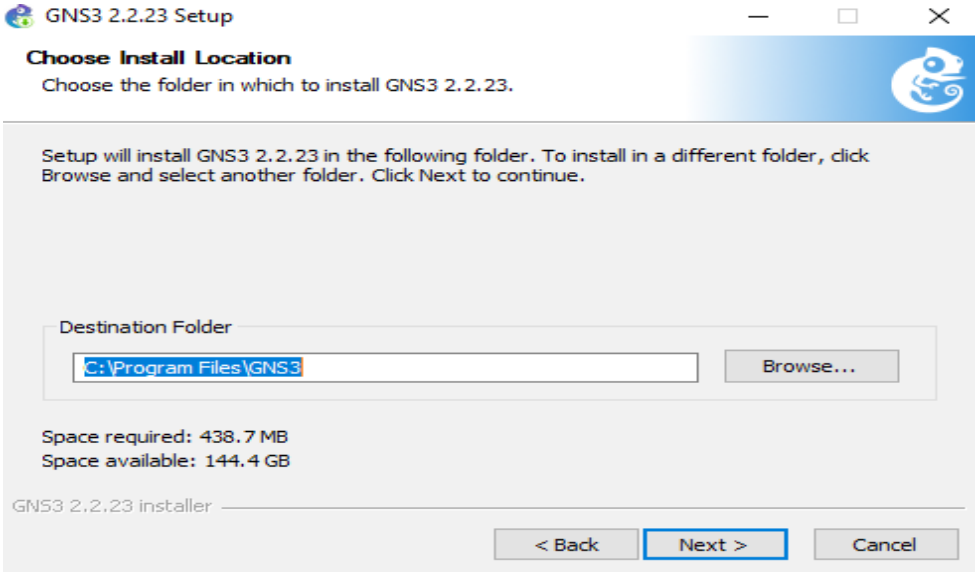
4. GNS3 est livré avec divers logiciels prérequis et optionnels. Par défaut, la plupart des logiciels sont sélectionnés pour l'installation.



5. On choisit l'hyperviseur pour installer GNS3 VM.



6. Un emplacement par défaut est choisi pourGNS3. Cliquant sur le bouton Installer pour accepter l'emplacement par défaut et pour commencer l'installation proprement dite des fichiers.



7. Lorsque l'Assistant a terminé, nous pouvons décocher **start GNS3**, puis cliquons sur le bouton Terminer.

- **Annexe 2 : Installation de GNS3vm**

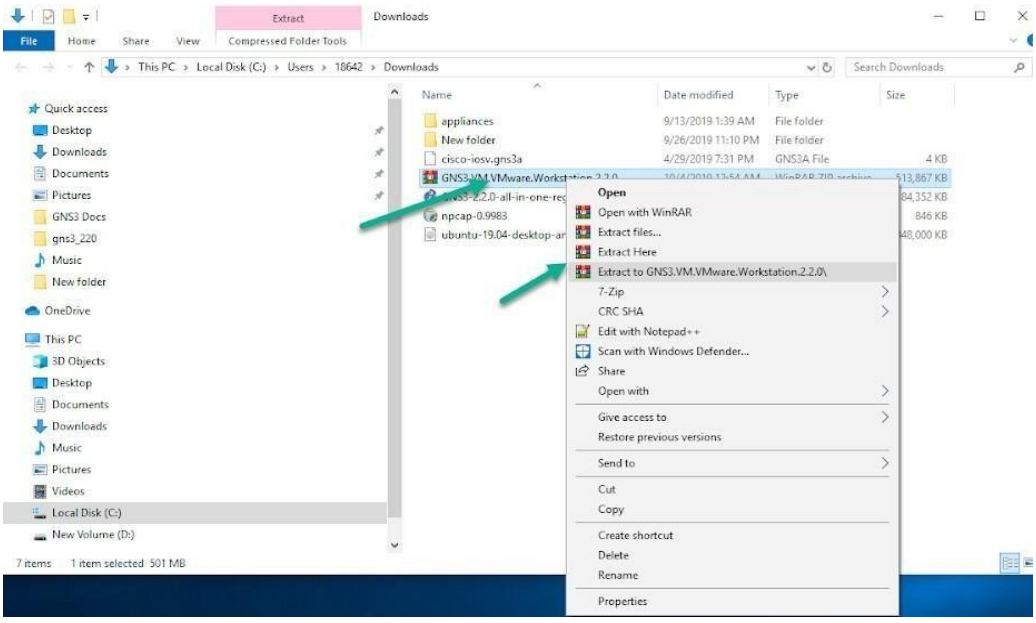
ETAPE 1 : TÉLÉCHARGEMENT DE LA MACHINE VIRTUELLE

Nous pouvons obtenir le GNS3-VM de plusieurs façons. Dans notre cas on a téléchargé via www.gns3.com en sélectionnant ce lien sous le lien vers l'application GNS3 principale elle-même.

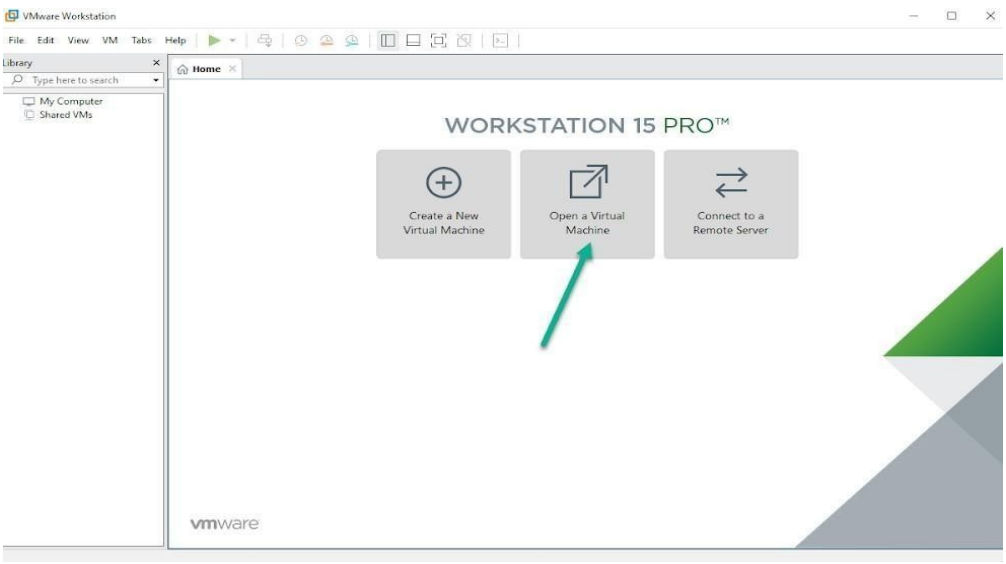
ETAPE 2 : IMPORTER UNE MACHINE VIRTUELLE GNS3 DANS VMWARE WORKSTATION

On a importé la machine virtuelle GNS3 dans VMware Workstation sur notre PC Windows local.

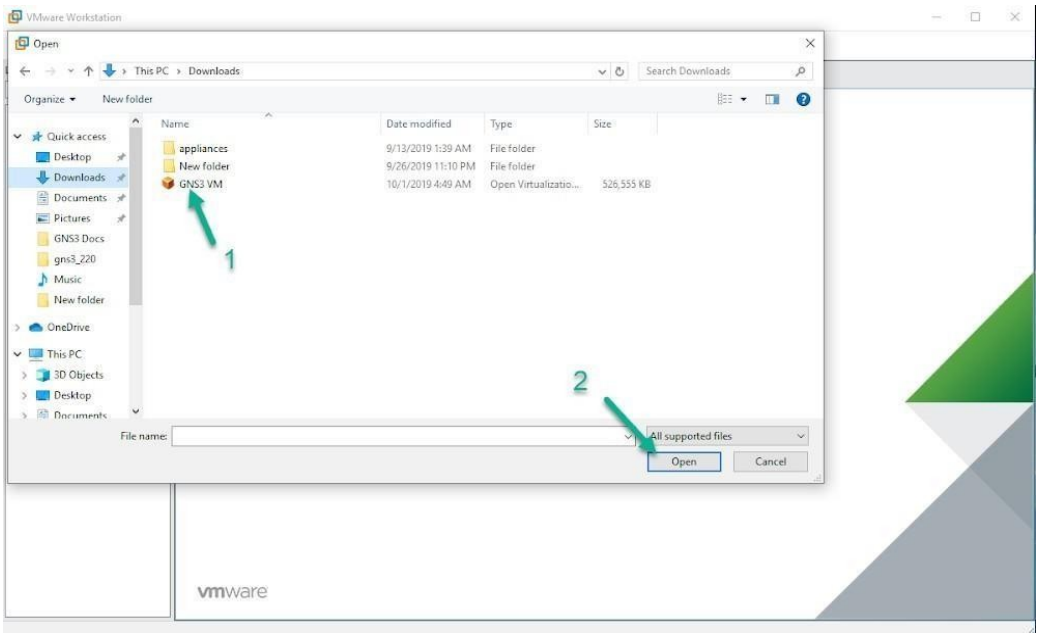
1. Extrayez l'archive .zip téléchargée :



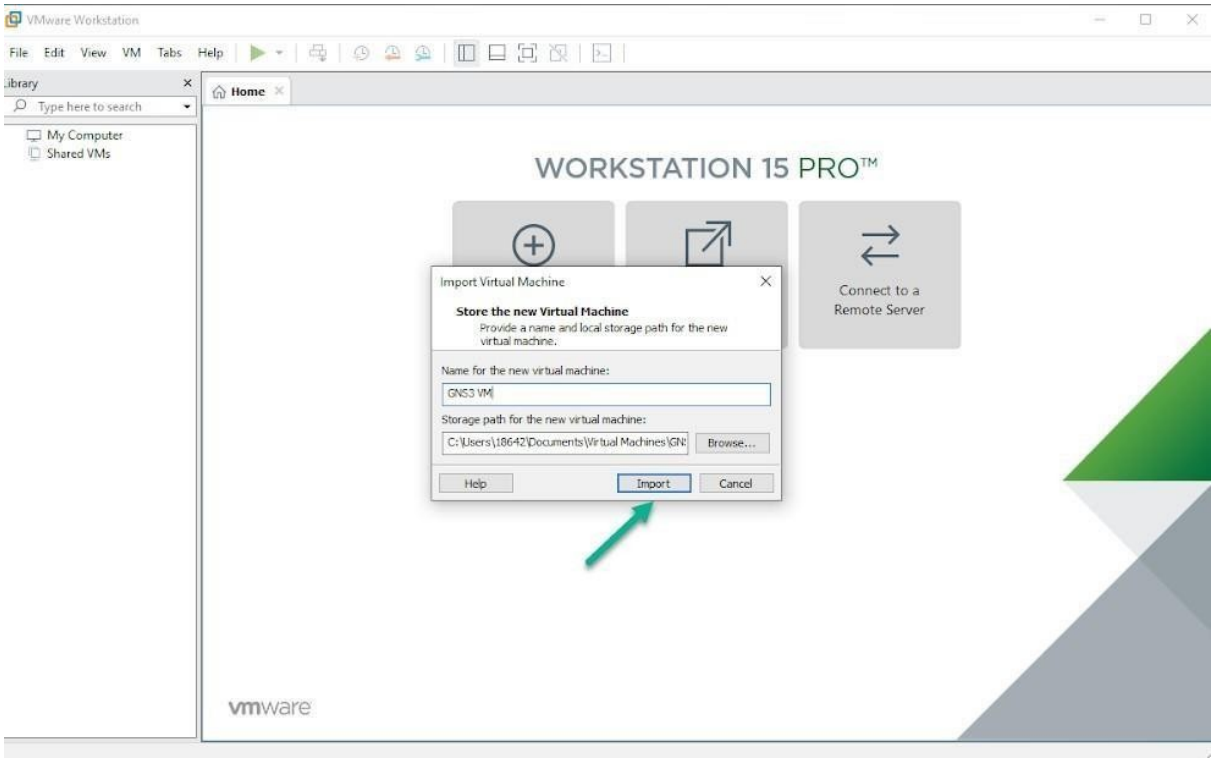
2. Dans VMware Workstation, cliquez sur« **Ouvrir une machine virtuelle**» :



3. Accédez au répertoire où se trouve le **fichier VM.ova GNS3** extrait, puis cliquez sur «**Ouvrir**»:

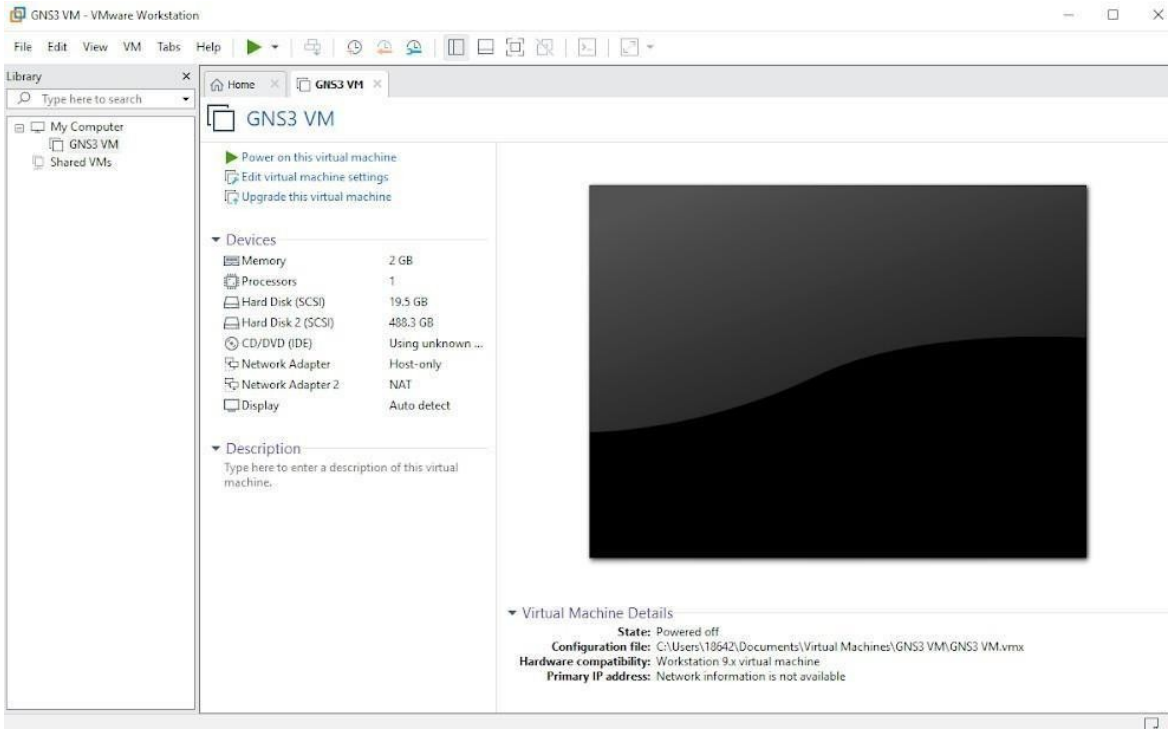


4. Laissez le nom de la machine virtuelle en tant que '**GNS3 VM**', puis cliquez sur '**Importer**':



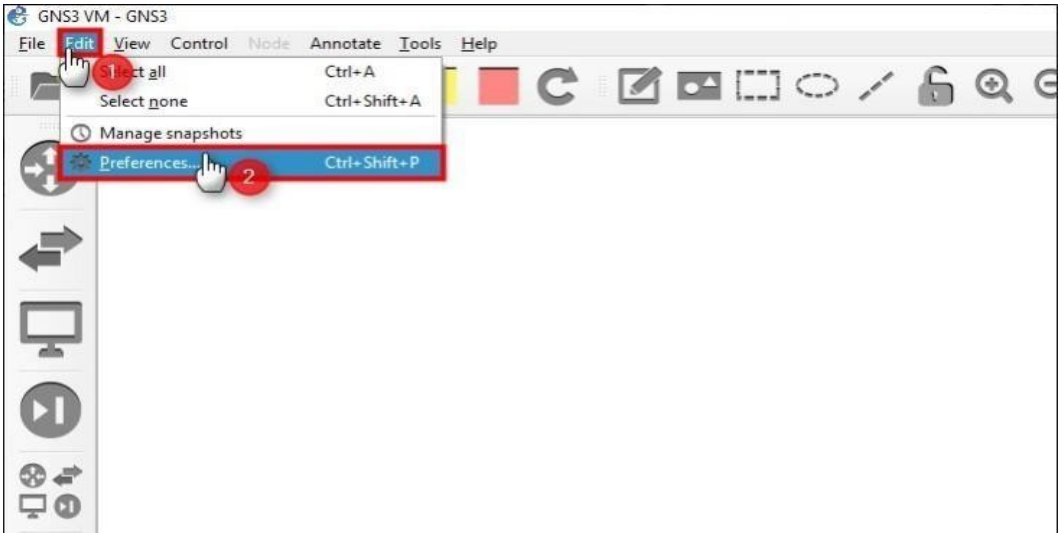
5. La machine virtuelle GNS3 s'affiche comme disponible dans VMware Workstation.

Laissez tous les paramètres à leurs valeurs par défaut :

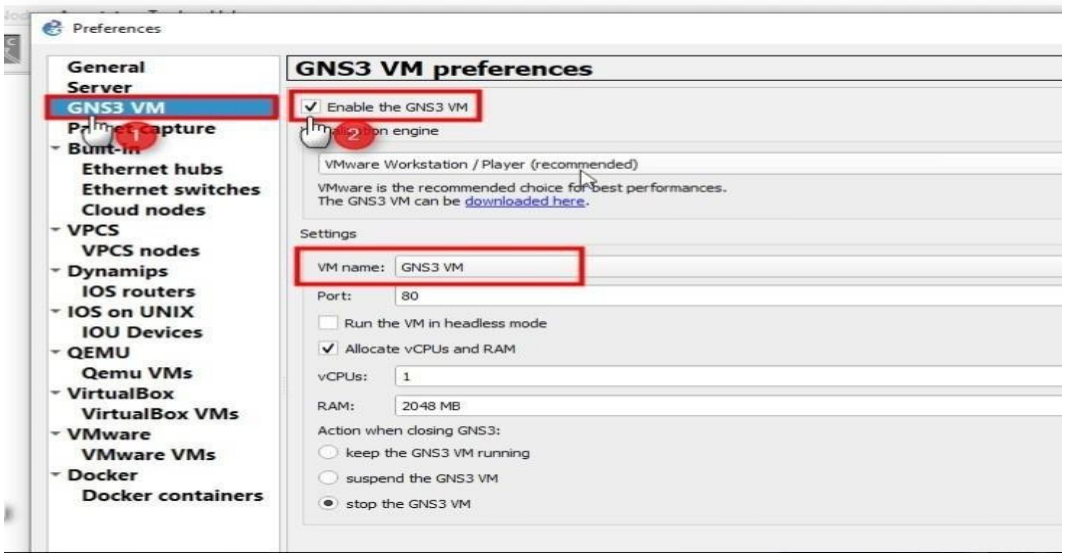


- **Annexe 3 : Intégration GNS3 avec la machine virtuelle GNS3.**

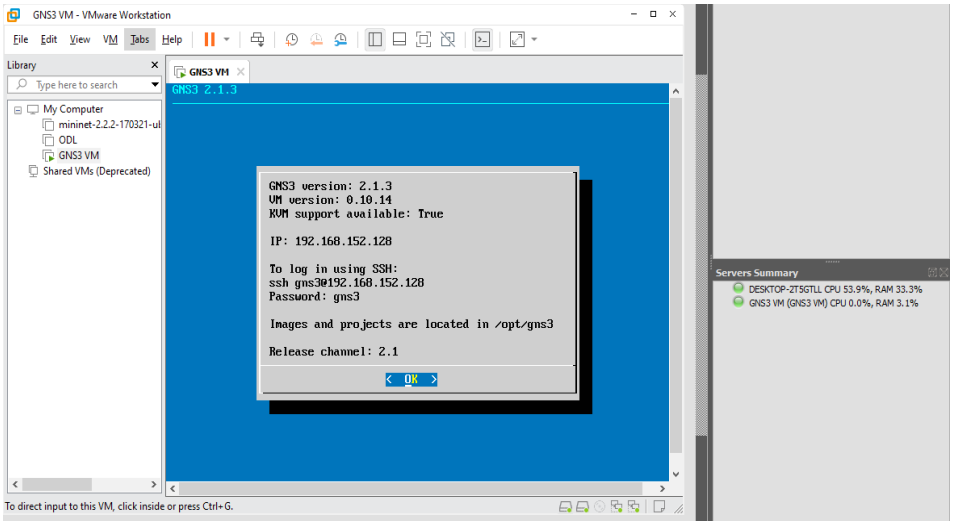
Etape 1 : ouvrant le programme GNS3 et cliquant sur Fichier après Préférence.



Etape 2 : Tout d'abord, nous cliquons sur GNS3 VM, pous nous sélectinnos l'option <<Enable the GNS3 VM >> dans les paramètres de la section de droite et cliquant sur le bouton OK.



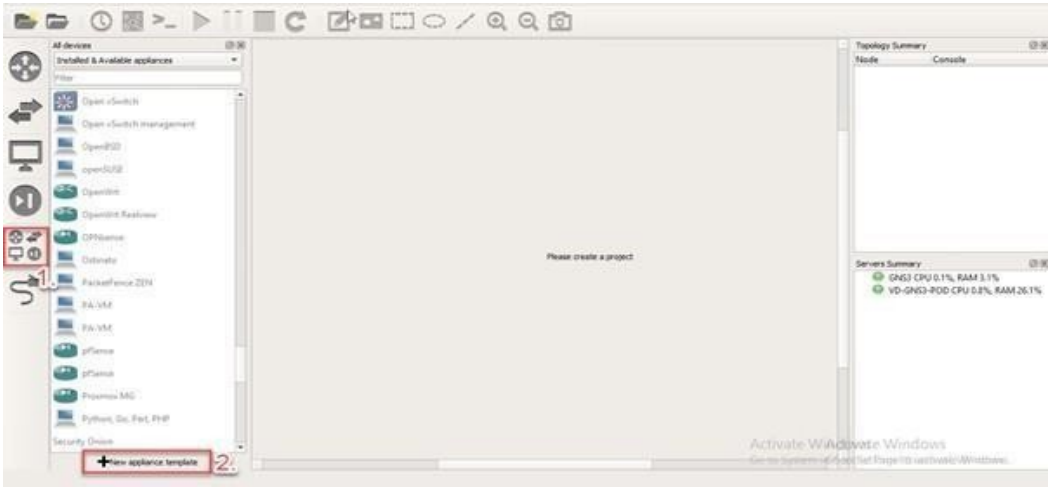
Lorsque nous activons l'option appropriée pendant l'exécution de serveur GNS3, on peut voir que le serveur virtuel a démarré activement dans la section résumé des serveurs.



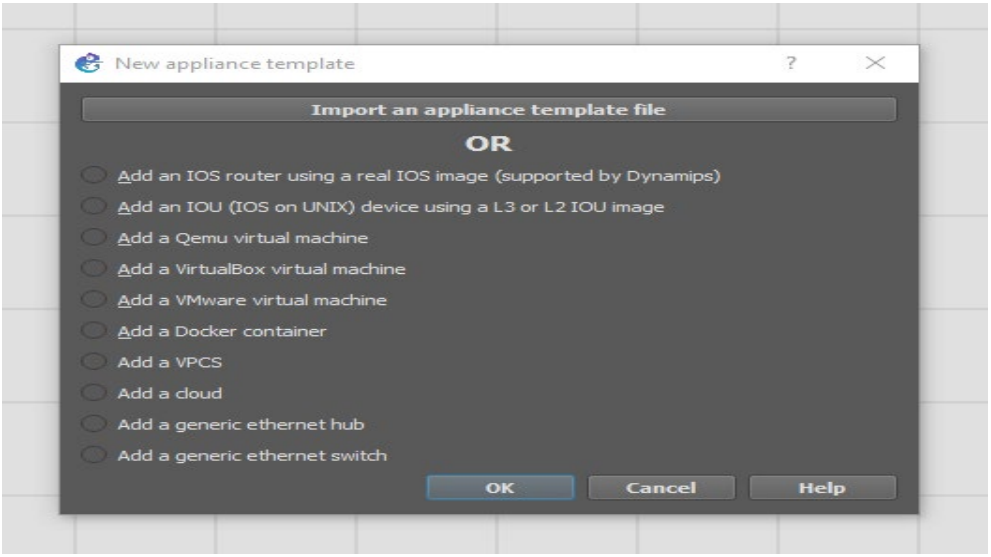
• Annexe 4 : Comment ajouter un appareil et une image sur le GNS3

Pour le routeur CSR1000v :

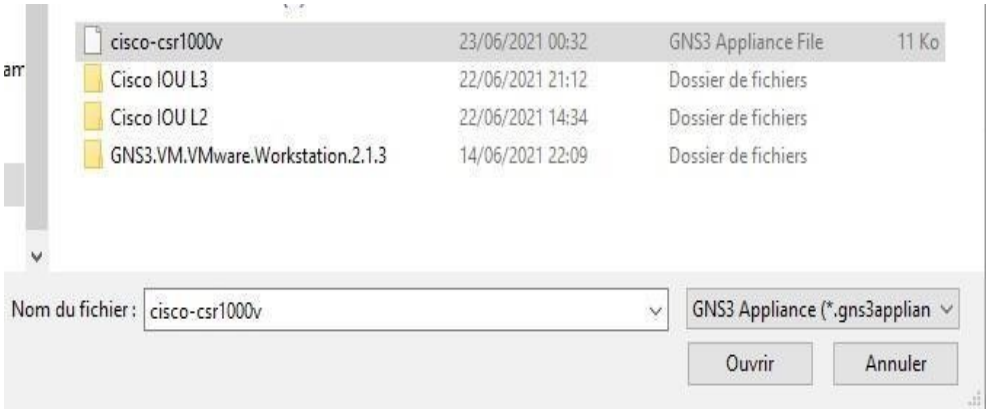
Étape 1 : on Clique sur "browse all devices" puis sélectionnez "new appliance template button".



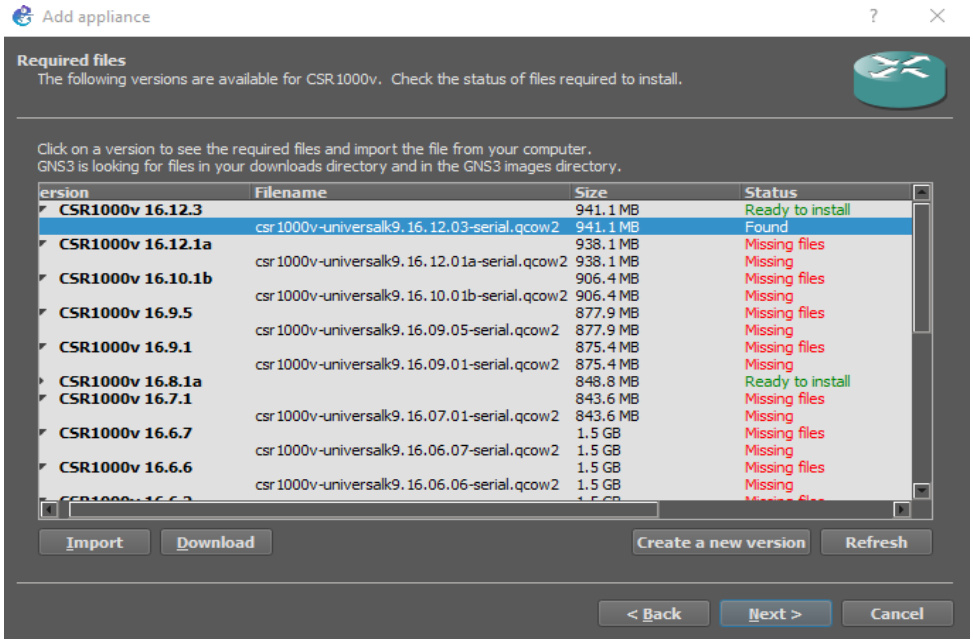
Étape 2 : Sélectionnez "importer un fichier de modèle d'appareil".



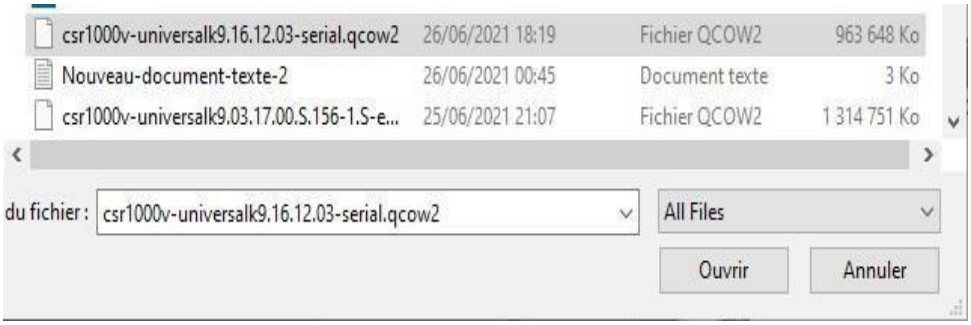
Étape 3 : Sélectionnez le fichier de l'appareil. Puis cliquez sur suivant.



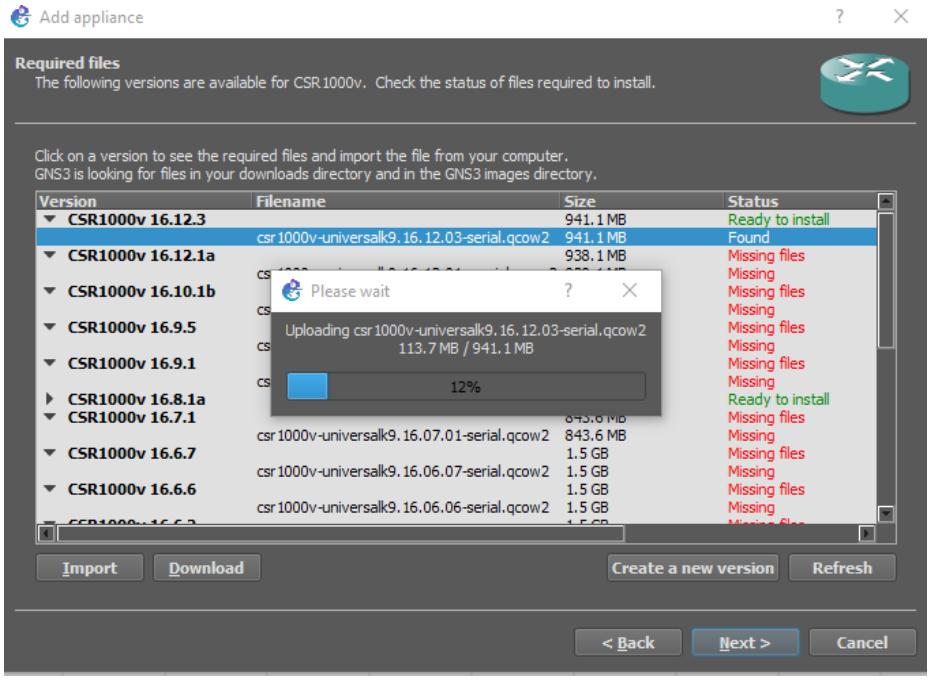
Étape 4 : Sélectionnez le fichier existant, puis cliquez sur l'option d'importation.



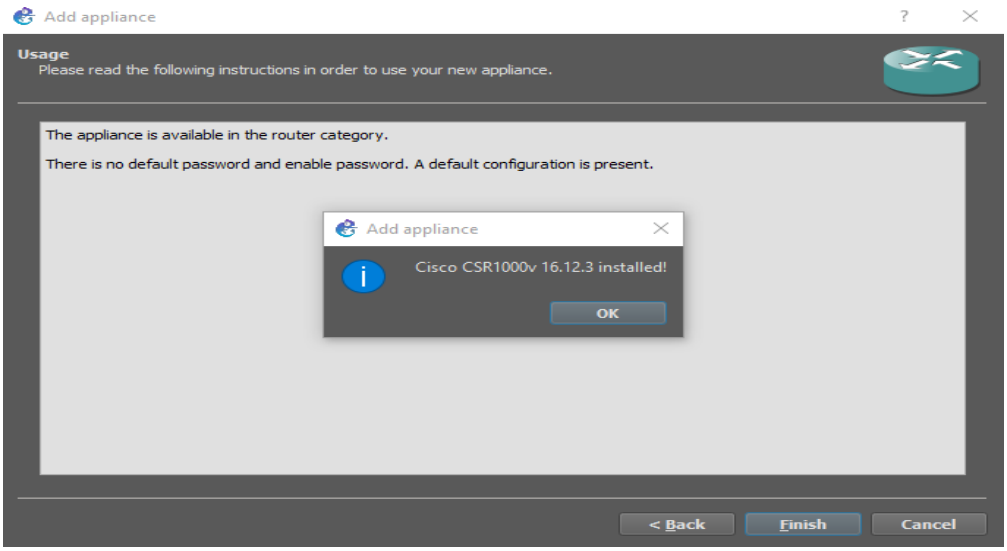
Étape 5 : Sélectionnez le fichier image téléchargé dans le dossier.



Étape 6 : Une fois que le fichier est importé avec succès, cliquez sur suivant.



Étape 7 : Enfin, vous pouvez voir l'image dans l'onglet de l'appareil installé.



On suit les mêmes étapes pour importer Cisco IOU L3 et Docker Ubuntu.

• Annexe 5: installation OpenDaylight version Magnésium

Pour l'installation de opendaylight il faut savoir d'abord que y'a trois types de plate-forms, pour l'utilisateur y'a une seule pour les développeurs y'a deux :

- **Opendaylight distributions-base :** cette plateforme c'est la première version du code logiciel pour le contrôleur OpenDaylight qui s'appelle Hydrogen aide les utilisateurs à démarrer. plateforme Hydrogen.

- **Opendaylight Docker Image :** utiliser par les développeurs c'est une image de conteneur utilisé dans les docker.

- **Opendaylight's apache karaf-based distribution:** utiliser par les développeurs comme la plateform Beryllium, Boron, Carbon, Magnesium…

Dans notre cas dans un container docker Ubuntu 16.04 nous avons commencé par installer la plateforme Magnesium, les étapes d'installation comme suit:

Etape 1: préparation du système d'exploitation :

Dans cette étape nous avons exécuté une mise à jour apt-get pour nous assurons que notre serveur reçoit tous les packages de sécurité et d'application les plus récents :

$ apt-get update

Apres nous avons utilisé Wget qui est un outil utilitaire libre en ligne de commande GNU utilisé pour télécharger des fichiers depuis Internet pour installer les packages de commodité comme vim qui est un éditeur de texte, Il peut être utilisé pour éditer toutes sortes de texte brut afin de faciliter la tâche :

$ sudo apt-get –y install unzip vim wget

Etape 2 : installation de Java JRE :

L'installation d'OpenDaylight via l'archive zip de sortie nécessite l'environnement d'exécutionJAVA 11, dans cette étape nous expliquons les commandes utiliser pour installer le JDK.
D'abord nous avons exécuté les commandes suivantes pour installer le JRK :

$ apt install software-properties-common

$ add-apt-repository ppa:openjdk-r/ppa

Ensuite, Installant la machine virtuelle Java en exécutant la commande suivante :

$ apt install openjdk-11-jdk

En vérifie par la commande suivante :

$ java –version

```
root@ODL:~# java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.16.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.16.04, mixed mode, sharing)
root@ODL:~#
```

Etape 4 : Téléchargement de l'archive Zip Opendaylight.

Dans cette étape nous avons exécuté la commande *wget* avec le lien de la release ODL de notre choix.

$wget https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integ

Apres nous Décompressons l'archive en exécutant la commande suivante :

$ unzip karaf-0.12.2.zip

Nous Démarrons le contrôleur et installons les outils requis avec la commande suivante :

$ cd karaf-0.12.2

$ ./bin/karaf

Le design ODL est apparu ça veut dire que l'installation a été réussite.

Nous installons la fonctionnalité nécessaire à OpenDaylight:

opendaylight-user@root>feature:install odl-restconf odl-bgpcep-bgp odl-bgpcep-pcep odl-netconf-topology

Une fois les fonctionnalités installées, déconnectez-vous du *karaf :*

opendaylight-user@root>logout