République Algérienne Démocratique Et Populaire
Université Saad Dahlab Blida 1
Faculté des Sciences
Département d'informatique

Thème

Implémentation d'une Solution Segment Routing sur un environnement basé MPLS/SDN

En vue d'obtention du diplôme de Master

Domaine : MI

Filière : Informatique

Spécialité : Système informatique et réseaux

Spécialité : Sécurité des systèmes informatique

Organisme d'accueil : ERICSSON
Rapport présenté par: Latreche Mohamed ,Azouz Malik
Promoteur : Mr. Mohamed OULD-KHAOUA
Encadreur: Mme K.HAMMOUTENE

Année universitaire : 2022/2023

REMERCIEMENTS

Nous remercions toutes les personnes ayant contribuées au succès de ce projet, merci pour l'aide la patience le support et le courage

Merci à Mme HAMMOUTENE Khadidja Amina (ERICSSON) et Mr. Mohamed OULD-KHAOUA(USDB), pour leur patience, leur disponibilité et surtout à leurs judicieux conseils,qui ont contribué à alimenter notre réflexion

On remercie évidemment nos parents pour toute ces années passées à nous soutenir sans jamais faillir à leur mission de parents.

Azouz Malik / Latreche Mohamed

Table des matières

Liste Des Figures

Liste des abréviations

AS	Autonomous System
API	Application Programming Interface
BGP	Border Gateway Protocol
CE	Customer edge
CLNP	Connectionless-mode Network Service
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
IETF	Internet Engineering Task Force
IEC	International Electrotechnical Commission
IGRP	Interior Gateway Routing Protocol
IGP	Interior Gateway Protocol
IP	Internet Protocol
ISIS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
LDP	Label Distribution Protocol
LSA	Link-State Advertisement
LSP	Link-State Path
LSR	Link-State Router
MPLS	Multiprotocol Label Switching
NBI	Northbound Interface
OF	OpenFlow
ONF	Open Networking Foundation
OSPF	Open Shortest Path First
OVSBD	Open vSwitch DataBase
PCC	Path Computation CLient
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
RTP	Real-time Transport Protocol
SDN	Software Defined Network
SR	Segment Routing
TE	Traffic Engineering
TCP	Transmission Control Protocol

Résumé

Les fournisseurs IT prévoient d'offrir une variété croissante de services en informatiques pour ses clients avec l'évolution rapide de télécommunication. Les technologies qui supportent actuellement les réseaux IP/MPLS n'ont cependant pas les propriétés de flexibilité et d'évolutivité nécessaires pour réaliser une telle évolution. En revanche, une nouvelle approche s'est introduite pour optimiser la gestion de ces réseaux, qui le SDN en séparant le contrôle du trafic d'acheminement, mais ceci n'a pas pu poursuivre le développement massif de la technologie en termes d'acheminement du trafic. Dans ce document, nous présentons le routage par segment, une nouvelle architecture de réseau qui vise à répondre à cette lacune où il met en œuvre les concepts de routage et de tunneling à la source, en permettant aux nœuds de diriger les paquets sur des chemins en utilisant une séquence d'instructions (segments) placée dans l'en-tête du paquet sans entrée par flux au niveau des routeurs intermédiaires. Ce travail présente la mise en œuvre de la solution routage par segment sous le réseau MPLS/SDN dans un environnement virtualisé.
Mots clés : Routage par segment, MPLS, SDN, OpenDayLight.

Abstract

IT providers plan to offer an increasing variety of IT services to their customers with the rapid evolution of telecommunications. However, the technologies that currently support IP/MPLS networks do not have the properties of flexibility and scalability necessary to achieve such an evolution. On the other hand, a new approach has been introduced to optimize the management of these networks, which the SDN by separating the control of the routing traffic, but this could not continue the massive development of the technology in terms of routing traffic. In this paper, we present Segment Routing, a new network architecture that aims to address this shortcoming where it implements the concepts of source routing and tunneling, by allowing nodes to direct packets on paths using a sequence of instructions (segments) placed in the packet header with no per-flow entry at intermediate routers. This work presents the implementation of the segment routing solution under the MPLS/SDN network in a virtualized environment.
Keywords: Segment routing, MPLS, SDN, OpenDayLight.

Introduction Générale

Au cours de ces trois dernières décennies, la communication et l'échange d'informations dans le réseau informatique ont considérablement évolué avec l'évolution rapide des solutions informatiques, parmi lesquelles l'apparition de la technologie Internet Protocole (IP). Cette croissance a également fait apparaître des inquiétudes concernant la bonne gestion et aux performances d'acheminement du trafic dans le réseau informatique.

Les réseaux IP répondent bien aux applications et aux exigences des utilisateurs existants, mais certains éléments posent souvent des problèmes pour un ensemble d'applications de plus en plus large.

L'architecture de base des réseaux informatiques n'a pas changé. Les exigences en matière d'évolutivité et de subtilité des réseaux modernes sont si élevés que les approches traditionnellesde mise en réseau, dans lesquelles les dispositifs prennent des décisions de routage indépendantesen exécutant plusieurs protocoles distribués, sont devenues un facteur limitant en termes d'innovation et de croissance dans l'industrie.

Les ingénieurs, chercheurs et fournisseurs IT développent de nouvelles techniques, architectures et protocoles pour prendre en charge ces nouveaux moyens d'acheminement, parmi lesquels la technologie Commutation Multi Protocole par Etiquette (MPLS) qui utilise à la fois de nouvelles techniques et les techniques de routage IP existantes afin d'améliorer les performances d'acheminement du trafic.

Ces nouvelles techniques sont le protocole MPLS pour acheminer le trafic par le Chemin Commuté par Etiquette (LSP).

Le MPLS à permet d'améliorer et de résoudre les problèmes existés mais la complexité du réseau nécessite une gestion plus efficace et plus robuste.

Des nouvelles approches sont apparues parmi lesquelles le Réseau Défini par Logiciel (SDN) qui permet de séparer le contrôle d'acheminent du trafic, ceci a permis d'optimiser le réseau, de les rendre plus flexibles et une sécurité plus robuste. Les performances sont améliorées mais ils n'ont pas satisfait complètement les exigences en termes d'acheminement.

Une solution plus récente et qui est en cours d'études fait apparaître pour optimiser, simplifier et améliorer l'évolutivité des réseaux MPLS/SDN qui est le routage par segment.

C'est un nouveau paradigme qui utilise un modèle de routage basé sur la source, dans lequel un nœud de réseau dirige un paquet en fonction d'une liste d'instructions contenues dans l'en-tête du paquet appelées "segments», où chaque liste d'un segment est identifier par un ID unique appelé Segment ID (SID), et le contrôle reste au niveau du SDN.

L'organisation du mémoire est comme suit :

Chapitre 1: Introduction sur le Protocoles de routage et Multi Protocol Label Switching (MPLS)

Chapitre 2: Software Defined Network(SDN)

Chapitre 3: Simulation du réseau MPLS/SDN basé sur routage par segment

Chapitre1:Protocoles de routage et Multiprotocol Label Switching

1.1 Introduction

Le cœur de l'Internet est constitué par le trajet étonnant des paquets de données depuis le centre de données jusqu'aux périphériques, dont la gestion est complexe pour assurer un transfert efficace. Ce processus est appelé routage et de nombreux algorithmes sont mis en œuvre via des protocoles pour acheminer les paquets. Les protocoles de routage traditionnels impliquent que chaque routeur traite chaque paquet pour déterminer le prochain saut, ce qui a entraîné de nombreux problèmes tels qu'un en-tête IP large, une lenteur de routage dans la couche réseau et une considération limitée des métriques supplémentaires. Pour répondre à ces besoins, des protocoles tels que MPLS ont été développés, où les paquets sont dirigés à travers un chemin via des étiquettes plutôt que des recherches complexes dans une table de routage à chaque arrêt. MPLS gère les flux de trafic de différentes granularités et prend en charge différents protocoles. Ce chapitre traite des protocoles de routage et présente le protocole MPLS comme solution à la problématique posée.

1.2 Routage

Le routage, qui correspond à la couche 3 du modèle OSI et à la couche 2 du modèle TCP/IP1, est le processus effectué par un routeur pour transférer un paquet d'un réseau à un autre en choisissant le chemin optimal. Cela se fait via un algorithme qui utilise une métrique pour déterminer la meilleure route possible. Les informations relatives à ces routes sont stockées dans une table de routage qui contient d'autres informations importantes. La figure 1.1 illustre comment ce processus fonctionne.

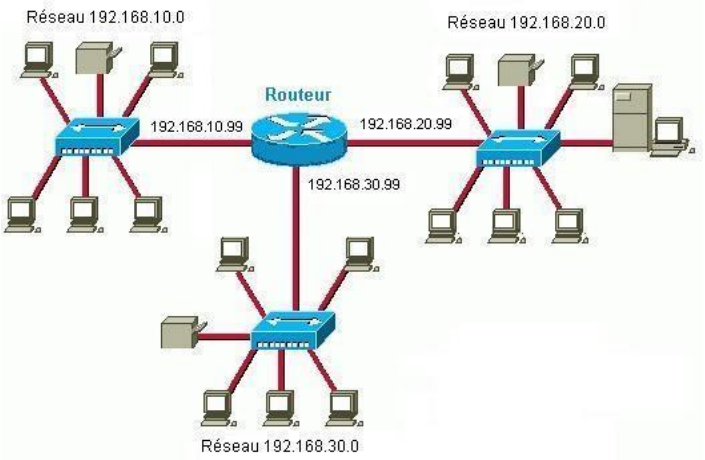

Figure 1.1 : Routage

Métrique :

La métrique s'agit d'une mesure utilisée par le protocole de routage pour calculer le meilleur chemin vers une destination donnée, lorsqu'il existe plusieurs routes vers le même réseau de destination avec la même valeur de préférence. Chaque protocole de routage utilise une métrique différente, où les plus courantes sont le saut, la bande passante, le retard, la fiabilité, la charge et le coût [15].

Table de Routage :

La table de routage est une structure hiérarchique qui se diffère d'un protocole à un autre selon l'algorithme utilisé, où chaque ligne ou entrée de la table indique la route optimale pour atteindre le réseau de destination désiré, par l'interface ou l'adresse du prochain saut avec la métrique (voir figure 1.2), alors chaque route dépend de l'algorithme utilisé où ils se divisent en trois types statique, dynamique et par défaut [1].

```
/> route
|   Destination   |     Masque      |    Passerelle    |    Interface    |
 ------------------------------------------------------------------------
| 172.12.0.2      | 255.255.255.255 | 127.0.0.1        | 127.0.0.1       |
| 172.12.0.0      | 255.255.0.0     | 172.12.0.2       | 172.12.0.2      |
| 127.0.0.0       | 255.0.0.0       | 127.0.0.1        | 127.0.0.1       |
| 0.0.0.0         | 0.0.0.0         | 172.12.255.254   | 172.12.0.2      |

/>
```

Figure 1.2 : Table de routage.

Routage statique

Il s'agit d'une route configurée manuellement qui sera inscrite dans la table de routage, obligeant les paquets à emprunter le chemin spécifié. En cas de changement dans le réseau, la route ne sera pas modifiée, sauf si quelqu'un la corrige manuellement pour détourner le trafic. Les routes statiques ont la priorité sur les routes choisies par les protocoles de routage dynamique [15].

Le routage par défaut peut être considéré comme un type particulier de routage statique. La différence entre eux est qu'une route par défaut est la route qui prend effet lorsque aucune autre route n'est disponible dans la table de routage pour une adresse de destination IP, ils ont une destination inconnue, d'où la route par défaut aide à les envoyées vers une seule adresse de saut suivant. La route par défaut dans IPv4 est désignée par 0.0.0.0/0 ou simplement 0/0. De même, dans IPv6, la route par défaut est spécifiée par : :/0. Elle est parfois appelée route de dernier recours [15][2].

Routage dynamique

Les routes sont définies par le logiciel en fonction de l'état actuel du réseau. Les modifications du réseau, telles que les ruptures de liens, les modifications du trafic et les modifications des coûts, sont échangées via des messages de mise à jour entre les routeurs pour construire une connaissance commune du réseau. Si un message indique qu'un changement de réseau s'est produit, le logiciel de routage recalcule les routes et envoie de nouveaux messages de mise à jour du routage, en utilisant des routes différentes, en fonction des conditions actuelles des circuits de communication. Le routage dynamique est préféré par rapport au routage statique, car les routeurs se mettent à jour en fonction de tout changement dans le réseau, il apporte donc une certaine robustesse face aux pannes [15].

1.2.1 Protocoles de routage dynamique

Tous les algorithmes appliqués pour le routage sont mis en œuvre dans un réseau à l'aide des protocoles où ils permettent de limiter la complexité de la table de routage, ce qui rend l'acheminement des paquets plus facile et plus rapide. Les protocoles de routage sont divisés en deux familles, Protocole de Passerelle Intérieure (IGP) et Protocole de Passerelle Extérieure (EGP) (figure 1.3).

1.2.2 Les routages IGP

Les protocoles IGPs sont utilisées pour acheminer le trafic à l'intérieur de chaque réseau distinct d'un système autonome (AS)2. Chaque AS possède un seul IGP. Des systèmes autonomes distincts peuvent utiliser des IGP différents. Il existe deux types d'IGP : le routage à vecteur de distance tel que le Protocole d'Information de Routage (RIP), le Protocole de Routage de Passerelle Intérieure (IGRP) et Protocole de Routage de Passerelle Intérieure Améliorée (EIGRP) et le routage à état de liaison tel que le protocole Ouvrir le Chemin le Plus court en Premier (OSPF) et Système Intermédiaire à Système Intermédiaire (IS-IS).

• Le protocole de routage à vecteur de distance ou Bellman-Ford, exige de chaque routeur qu'il échange avec ses voisins des informations, sur la topologie de son réseau en termes de liens et d'interfaces, ainsi la métrique, qui se fait via la table de routage, pour atteindre un nœud par l'intermédiaire de ces voisins tout en choisissant la métrique la plus faible entre les deux routeurs où elle sera ajoutée dans leur table de routage. Cette méthode aide à échanger des informations immédiates ainsi que de mettre à jour les tables de routage et elle présente une complexité de calcul et une surcharge de messages moindres que la seconde méthode.

2.Un système autonome ou Autonomous system est un groupement de réseaux qui sont tous gérés, contrôlés et supervisés par une seule entité ou organisation notamment l'IANA (Internet Assigned Numbers Authority). Un AS utilise une politique de routage commune contrôlée par l'entité [15].

• IS-IS [17]

Le protocole IS-IS est similaire au OSPF, il utilise l'algorithme SPF pour déterminer les routes. IS-IS évalue les changements de topologie et détermine s'il faut effectuer un recalcule SPF complet ou un Calcul de Route Partiel (PRC). Ce protocole a été développé à l'origine pour le routage des paquets du Protocole de Réseau Sans connexion (CLNP) de l'Organisation internationale de normalisation (ISO) par l'IETF où elle s'est définie

dans RFC 1142[17].

1.2.2.1 EGP

Les EGPs sont utilisées pour acheminer le trafic entre les routeurs qui relient des ASs distincts notamment : Protocole de Passerelle Frontalière (BGP). Le BGP définie par l'IETF dans RFC 1105 est un protocole de routage du système inter autonome.

La fonction principale d'un système BGP est d'échanger des informations d'accessibilité au réseau avec d'autres systèmes BGP. Les informations d'accessibilité sont sur les ASs que le trafic doit traverser pour atteindre ces réseaux. Ces informations sont suffisantes pour construire un graphe de connectivité des ASs à partir duquel il est possible d'élaguer les boucles de routage et d'appliquer les décisions de politique générale au niveau des AS.

BGP fonctionne sur un protocole de niveau transport fiable. Cela élimine la nécessité de mettre en œuvre la fragmentation explicite des mises à jour, la retransmission, l'accusé de réception et le séquençage [11].

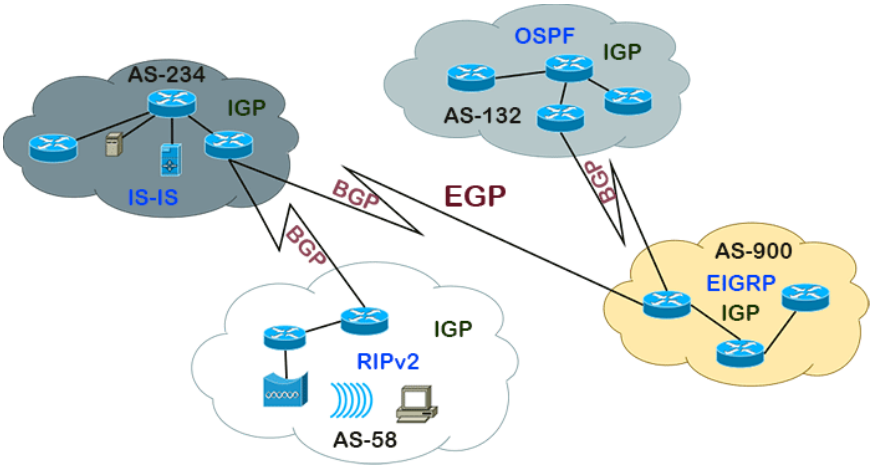

Figure 1.3: IGP et EGP.

D'autres protocoles ont été développés pour optimiser la transmission de paquets sur les réseaux étendus. Le MPLS est le protocole le plus couramment utilisé à cet effet, et nous allons le présenter dans la suite de notre étude.

1.3 MPLS

Le MPLS est une technologie de transfert de paquets défini par l'IETF dans RFC 3031 afin de remplacer la transmission IP par la commutation par étiquette, qui associe le principe de routage (couche 3) et les mécanismes de commutation (couche 2), c'est un protocole de couche 2,5. Il peut être étendu à de multiples protocoles de la couche 3 (IPv4, IPv6, etc.). MPLS n'est en aucune façon restreint à une couche 2 spécifique et peut fonctionner sur tous les types de support permettant l'acheminement de paquet de niveau 3[7][15].

1.3.1 Principe du MPLS

Il consiste à attribuer aux paquets des étiquettes courtes décrivant la manière dont ils doivent être transmis sur le réseau, indépendamment des tables de routage ou de tout protocole de routage. Il est également une technologie de tunneling, analysant que les en-têtes des paquets au niveau les routeurs situés sur les bords d'un réseau, et non à chaque saut, ce qui augmente la vitesse de transmission, ainsi, il offre un certain degré de sécurité lors de la transmission des paquets.

- Label

Le label est l'élément principal dans le réseau MPLS, qui se trouve entre l'en-tête de la couche liaison et l'en-tête de la couche réseau du paquet MPLS (voir figure 1.4) qui fait la différence entre paquet MPLS avec le paquet IP. C'est un identifiant court, de longueur fixe (4 octets), qui n'est significatif que localement qui est utilisé pour identifier un forwarding équivalence class (FEC) à laquelle un paquet appartient [9].

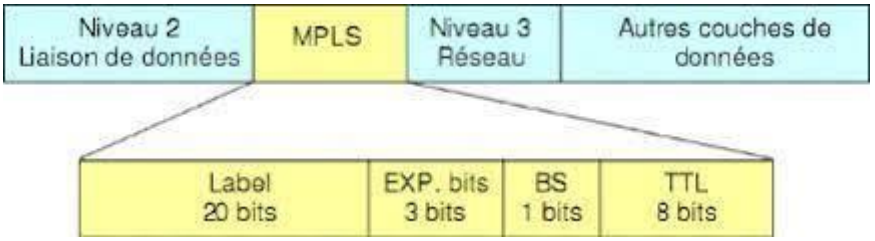

Figure 1.4: En-tête MPLS[9].

• Label : valeur de label de 20 bits.

- Exp : 3 bits, utilisé comme valeur d'extension. Généralement, ce champ est utilisé comme champ de classe de service (CoS). En cas d'encombrement, les périphériques donnent la priorité aux paquets qui ont une valeur plus grande dans ce champ
- BS : valeur de 1 bit indiquant le bas d'une pile d'étiquettes. MPLS prend en charge l'imbrication de plusieurs étiquettes. Lorsque le champ S est égal à 1, l'étiquette se trouve au bas de la pile d'étiquettes.
- TTL : Time To Live (temps de vie). Ce champ de 8 bits est le même que le champ TTL des paquets IP.

FEC est un ensemble de paquets qui sont acheminés de la même manière, sur le même chemin, avec le même traitement d'acheminement et en utilisant la même étiquette MPLS. Les FECs sont définies dans la spécification de base du protocole de distributions et peuvent être étendues par l'utilisation de paramètres supplémentaires. Dans certains cas, comme l'équilibrage de la charge, un FEC peut être associé à plusieurs étiquettes entrantes. Toutefois, chaque étiquette ne représente qu'un seul FEC sur un appareil [15]. Il peut y avoir une pile d'étiquettes "Label Stack" MPLS dans quelque application telle que VPN MPLS, dont, ces étiquettes sont organisées selon la règle du dernier entré, premier sorti. Les étiquettes sont traitées à partir du haut de la pile. Dans la figure 1.5, l'étiquette située à côté de l'en-tête de la couche 2 est le haut de la pile d'étiquettes (étiquette MPLS externe), et l'étiquette située à côté de l'en-tête de la couche 3 est le bas de la pile d'étiquettes (étiquette MPLS interne). Une pile d'étiquettes MPLS peut contenir un nombre illimité d'étiquettes [9].

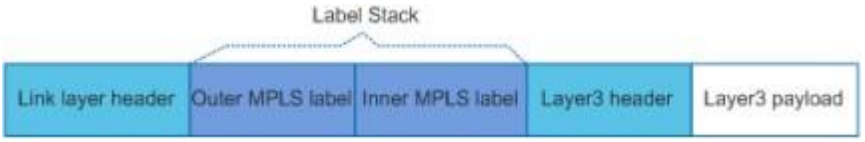

Figure 1.5: Pile d'étiquettes[9].

- **Label Switching Router**

Où dans le réseau MPLS, les routeurs sont appelés Routeur à commutation d'étiquettes (LSR), sur lequel MPLS est activé et qui peut traiter des paquets à commutation d'étiquettes en fonction de leurs étiquettes encapsulées. Il existe un autre type de routeur qui s'appelle "Edge LSR", c'est un LSR qui se situe à la périphérie ou au bord d'un réseau MPLS, faisant office de passerelle entre le réseau local et le réseau étendu ou l'Internet lui-même [9].

- **Label Switching Path**

Le chemin traversé par ces paquets étiquetés est les LSP qui sont établis par un protocole de signalisation tel que Protocole de Distribution d'Etiquettes (LDP), Protocole de Réservation des Ressources (RSVP), etc. Un LSP est une connexion unidirectionnelle, point à point, semiduplex, transportant des informations en aval du routeur entrant (premier) au routeur sortant (dernier). Les routeurs entrants et sortants ne peuvent pas être le même routeur. Le chemin est établi sur la base des critères de la FEC [9]. Toute commence lorsqu'un paquet entre dans le réseau MPLS par le E-LSR ou le nœud d'entrée qui sera affecté à une FEC, en fonction du type de paquets et de sa destination. En fonction de la FEC, le nœud d'entrée applique une étiquette au paquet et l'encapsule dans un LSP. Au fur et à mesure que le paquet passe par les "nœuds de transit" du réseau (LSR), ces routeurs continuent à diriger les paquets en fonction des instructions contenues dans l'étiquette du paquet. Ces arrêts intermédiaires sont basés sur l'étiquette du paquet et non sur des recherches IP supplémentaires. Au "nœud de sortie", ou routeur final à la fin du LSP, l'étiquette est retirée et le paquet est livré via un routage IP normal (voir figure 1.6) [9]. Où les opérations effectuées sur Label sont :

- **Pop** : est l'opération de suppression d'étiquette. Elle est effectuée par la sortie (étiquette supérieure). Elle peut également être effectuée par l'avant-dernier LSR pour diminuer le nombre d'étiquettes dans la pile d'étiquettes. Le paquet transmis est avec la pile d'étiquettes restante ou comme un paquet non étiqueté (Untagged ou No Label).
- **Swap** : est l'opération de remplacement d'étiquettes. Elle est effectuée par un nœud de transit pour remplacer une étiquette au sommet de la pile d'étiquettes dans un paquet MPLS par une autre étiquette, qui est attribuée par le prochain saut.
- **Push** : est l'opération d'ajout d'étiquettes. Elle est effectuée par les nœuds d'entrée et de transit d'un réseau MPLS. Lorsqu'un paquet IP atteint l'entrée d'un réseau MPLS, l'entrée ajoute une étiquette entre l'en-tête de la couche 2 et l'en-tête de la couche 3 du paquet avant de transmettre le paquet. Un nœud de transit à l'intérieur du réseau MPLS peut également ajouter une étiquette au sommet de la pile d'étiquettes si nécessaire.

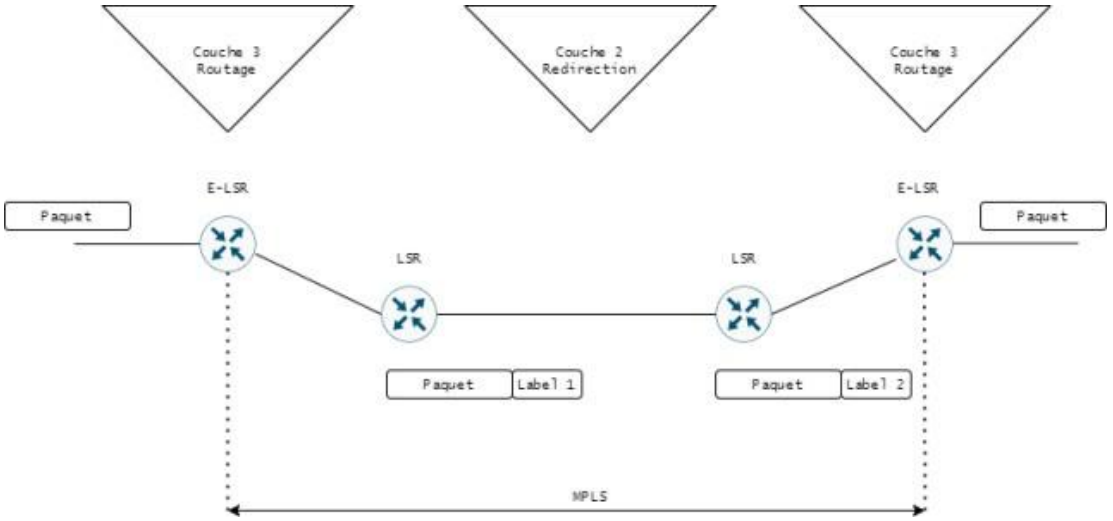

Figure 1.6: Principe de MPLS[3].

1.3.2 MPLS applications

MPLS prend en charge l'ingénierie du trafic (TE), la création de réseaux privés virtuels (VPN) et la qualité de service (QoS). L'ingénierie ou le contrôle du trafic se fait principalement par l'utilisation de protocoles de signalisation, notamment RSVP et LDP, pour établir des LSPs. La prise en charge des VPN comprend les VPN de couche 2 et de couche 3 et les circuits de couche 2.

TE (Traffic Engineering)

L'ingénierie du trafic ou "Traffic Engineering", est la capacité de contrôler le chemin emprunté par un réseau ou une partie d'un réseau en fonction d'un ensemble de paramètres de trafic (bande passante, paramètres de qualité de service, etc.), afin d'optimiser les performances des réseaux opérationnels et de leurs ressources en équilibrant la charge du trafic sur les liens, les routeurs et les commutateurs du réseau. Ce processus est particulièrement important dans les réseaux où plusieurs chemins parallèles ou alternatifs sont disponibles.

Qualité de service (Qos)

La qualité de service (QoS) représente l'ensemble des techniques nécessaires pour gérer la bande passante, le délai, la gigue et la perte de paquets du réseau. Du point de vue de l'entreprise, il est essentiel de garantir aux applications critiques les ressources réseau dont elles ont besoin, malgré les variations de la charge du trafic réseau.

Où :

- La bande passante est une mesure de la quantité d'informations qu'un réseau peut transférer.
- Le délai désigne le temps nécessaire à un paquet pour aller d'un point A à un point B.
- La gigue est un autre mot pour désigner l'incohérence. Il s'agit de la

fluctuation de la latence des paquets circulant sur le réseau, causée par la congestion du réseau ou les changements de route.

- La perte de paquet est lorsqu'un ou plusieurs paquets n'atteignent pas leur destination souhaitée.
- Toutes les implémentations de la QoS MPLS utilisent le modèle de services différenciés (DiffServ). Les routeurs utilisent trois bits, appelés bits expérimentaux (EXP bits) dans l'en-tête MPLS de chaque paquet transporté sur le réseau MPLS, donc il spécifier la QoS du paquet pour différencier le trafic [1].

1.3.3 Avantages et inconvénients MPLS

MPLS a de nombreux avantages mais aussi des inconvénients, dont nous citons :

Avantages

• Amélioration des performances de l'acheminement des paquets dans le réseau ;

• Il s'agit d'une technologie fiable ;

• Prise en charge de QoS et CoS (Class of service) pour la différenciation des services ;

• Il est flexible. Prise en charge de l'évolutivité du réseau ;

• Congestion de trafic faible ;

• Déploie des services à grande échelle.

Inconvénients

• Une couche supplémentaire est ajoutée ;

• Le routeur doit comprendre MPLS ;

• Il est optimisé pour la connectivité point à point uniquement ;

• Il n'y a pas de contrôle total sur le réseau qui contient MPLS. Le fournisseur de services configurera tous les réseaux et l'administrateur n'aidera le fournisseur que pour le routage dynamique ;

• Il est couteux lors de l'optimisation de diffusion dans les réseaux étendus.

1.4 Conclusion

Le MPLS représente une technologie essentielle pour le réseau backbone IP des futurs fournisseurs de services et opérateurs, offrant de nombreux avantages tels que l'évolutivité, la performance, une meilleure utilisation de la bande passante, la réduction de la congestion du réseau et une meilleure expérience pour l'utilisateur final. Toutefois, le MPLS présente également quelques désavantages majeurs. Le premier est qu'il est optimisé pour une connectivité point à point plutôt que point à cloud, ce qui signifie qu'il n'est pas possible d'accéder directement à chaque cloud avec le MPLS. Le deuxième est que le MPLS nécessite une optimisation du réseau étendu (WAN) pour simplifier la diffusion, ce qui peut engendrer un coût supplémentaire pour une solution déjà onéreuse. Cela ne signifie pas que le MPLS ne sera plus utilisé, mais plutôt qu'il y a un besoin croissant de nouvelles technologies pour résoudre ces problèmes. Le SDN

(Software-Defined Networking) et le routage par segment sont plus flexibles et permettent aux utilisateurs de bénéficier d'un plus grand contrôle et d'une plus grande facilité pour gérer les ressources virtuellement dans l'ensemble du plan de contrôle, ainsi que pour faciliter l'ingénierie du trafic dans les nouveaux réseaux. Ces technologies seront présentées dans le chapitre suivant. En résumé, le MPLS reste une technologie importante, mais il existe d'autres options à considérer pour répondre aux besoins actuels et futurs des réseaux étendus.

Chapitre 2 Software Defined Network(SDN)

2.1 Introduction

La mise en place et la gestion d'un réseau informatique à l'aide de l'infrastructure traditionnelle représente un défi majeur pour les ingénieurs IT. Les technologies matérielles existantes ne répondent plus aux besoins actuels des fournisseurs et des utilisateurs finaux, rendant l'approche classique inadéquate. C'est pourquoi de nouvelles approches de virtualisation ont émergé, notamment la virtualisation de la fonction réseau (NFV) et SDN, qui sont les plus recommandées

Cette nouvelle approche de virtualisation rend les réseaux plus agiles et permet l'apparition du routage par segment, ce qui résout les problèmes importants dans les réseaux IP/MPLS. L'objectif est de créer un réseau plus flexible, fiable et optimisé en termes de temps, de coûts et de ressources. Dans les sections suivantes, nous présenterons en détail l'architecture, le principe et les avantages de la technologie SDN, ainsi que la solution de routage par segment.

2.2 SDN(Software-Defined Networking)

Le SDN (Software-Defined Networking) est une technologie de virtualisation de réseau qui permet de gérer les réseaux informatiques de manière centralisée et programmatique. Contrairement aux réseaux traditionnels basés sur des équipements matériels tels que des commutateurs et des routeurs, le SDN sépare la partie de contrôle du réseau de la partie de transfert de données. Le contrôle est géré par un contrôleur SDN centralisé, qui utilise des protocoles de communication standard pour échanger des informations avec les équipements de transfert de données (switches, routeurs, etc.). Cette approche permet une gestion plus souple et plus dynamique du réseau, ainsi qu'une meilleure adaptation aux besoins changeants des applications.

En pratique, le SDN permet aux administrateurs réseau de programmer et de configurer le réseau à partir d'une interface centralisée, plutôt que de devoir gérer chaque équipement de manière individuelle. Il permet également de définir des politiques de sécurité et de qualité de service de manière plus fine, en fonction des besoins des applications et des utilisateurs. Enfin, le SDN facilite la mise en place de réseaux privés virtuels (VPN) et de réseaux dédiés (VLAN) en utilisant des techniques de virtualisation.[19]

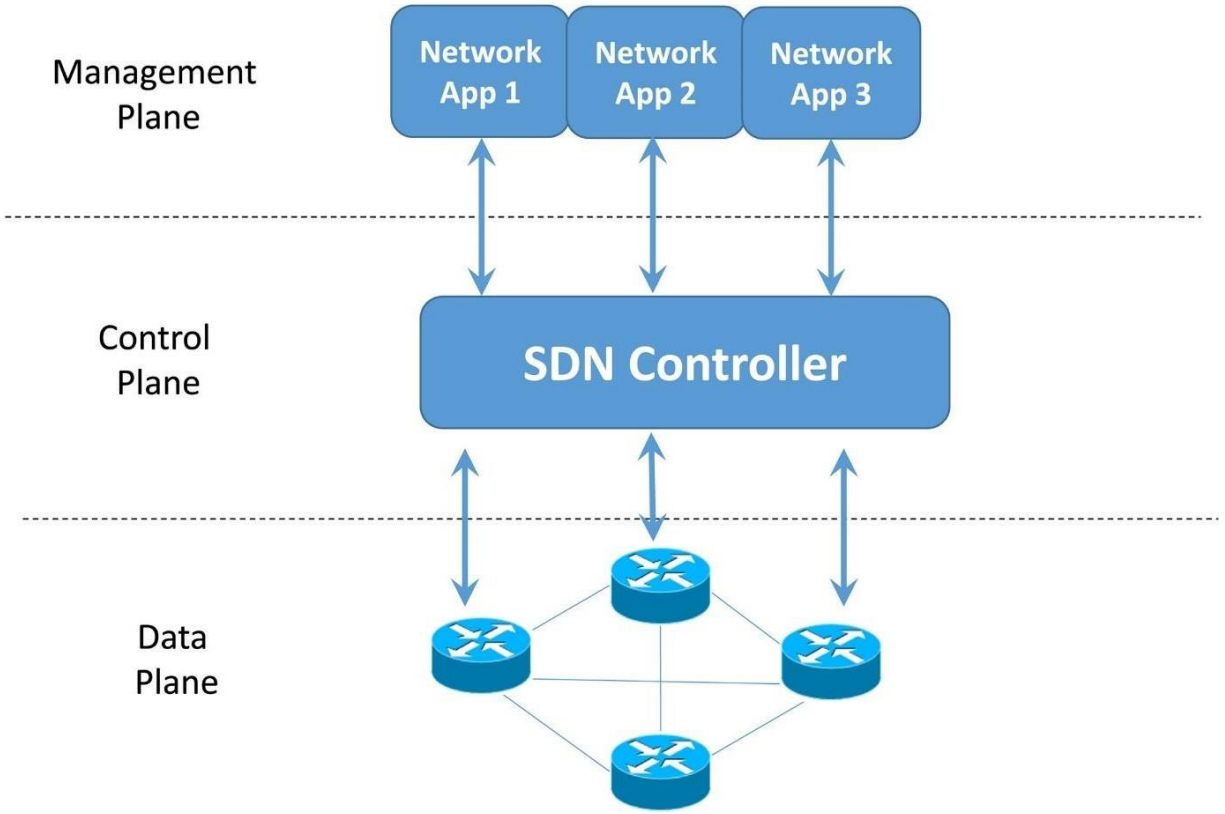

Figure 2.1 : Principe de SDN

2.3 Architecture du SDN

L'ONF a décrit une architecture de haut niveau du SDN (voir la figure 2.1), qui se divise fonctionnellement en quatre couches distinctes sont : infrastructure, contrôle, application et management [6].

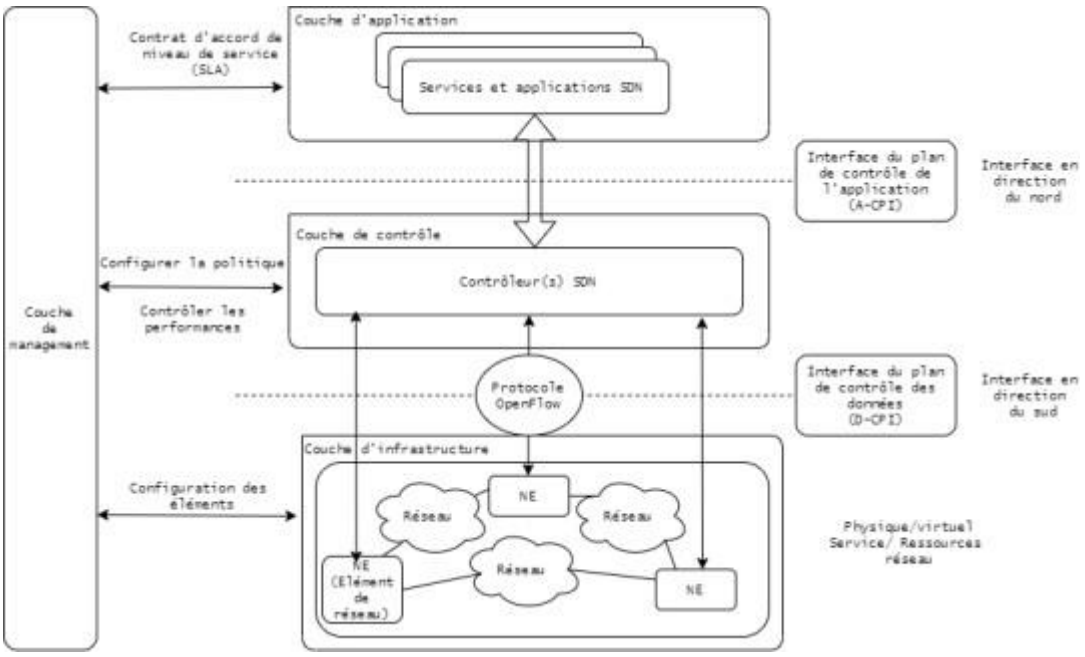

figure 2.2: Architecture de haut niveau du SDN[6]

Couche d'infrastructure

Elle représente la couche inférieure ou couche physique du SDN, qui est composée de dispositifs physiques ou virtuels appelés "Dispositifs SDN", dont le rôle est d'acheminer et de traiter le trafic en fonction des règles fournies par un contrôleur.

Couche de contrôle

La couche de contrôle dans un réseau SDN est un élément clé qui abrite la logique intelligente descontrôleurs. Elle permet de gérer l'infrastructure réseau en utilisant des règles basées sur les statistiques duréseau. Les entités communiquent entre elles via une interface appelée "est-ouest" qui est généralement mise en œuvre à l'aide de protocoles de passerelle tels que BGP, le PCE (Elément de calcul du chemin) ou lePCEP (protocole de communication). La couche de contrôle est également la couche intermédiaire qui relie la couche application à la couche infrastructure via deux types d'interfaces de programmation d'applications(API) : l'interface sud et l'interface nord.

L'interface sud permet à la couche application de communiquer ses exigences et instructions à la couche de contrôle, qui les traite avant de les transmettre aux composants du réseau via l'interface nord sous forme de messages d'application. La couche de contrôle traite également les informations pertinentes telles que les statistiques de trafic et la topologie du réseau à partir des dispositifs de mise en réseau et les transmet à l'application via l'interface sud. En résumé, la couche de contrôle est responsable de la gestion des règles et des politiques de contrôle du réseau, en communiquant avec les composants du réseau et en fournissant des informations pertinentes à la couche application pour optimiser les performances.

Couche Data

Dans le SDN (Software-Defined Networking), le plan de données est chargé du traitement des paquets porteurs de données en utilisant un ensemble de règles spécifiées par le plan de contrôle. Le plan de données peut être mis en œuvre dans des commutateurs matériels physiques ou dans des implémentations logicielles, telles que Open vSwitch. La capacité de mémoire des commutateurs matériels peut limiter le nombre de règles pouvant être stockées, tandis que les implémentations logicielles peuvent avoir une capacité plus élevée.

Couche d'application

Il s'agit d'un domaine en constante évolution qui vise à fournir ou à développer des applications et des services à la couche de contrôle en utilisant les comportements et les exigences du réseau collecté par la couche d'infrastructure.

Couche de management

D'après l'Open Networking Foundation (ONF), la couche de management est considérée comme l'un des éléments clés de l'architecture SDN. Elle doit être isolée et protégée du reste du réseau car elle est chargée des tâches sensibles telles que la gestion des pannes, la surveillance du réseau et la configuration des paramètres de celui-ci. Ces tâches sont mieux gérées en dehors des trois autres couches, pour garantir un fonctionnement optimal et une sécurité renforcée de l'ensemble du réseau.

Interface de programmation d'application

Une interface de programmation d'application ou l'API, est un intermédiaire logiciel qui permet à deux applications de communiquer entre elles à l'aide des protocoles. Or, dans les réseaux SDNs, les APIs se divisent en deux types, vers le nord et vers le sud [6].

• **APIs vers le nord (NBI) :** Elles constituent le lien entre la couche d'application et la couche de contrôle. Les NBIs ont de nombreuses fonctions notamment, les équilibreurs de charge, les pare-feu ou d'autres services

de sécurité définie par logiciel, les applications d'orchestrations ou d'automatisation sur les ressources du réseau, etc.

• **APIs vers le sud (SBI)** : Elles constituent le lien entre la couche de contrôle et la couche d'infrastructure. Où ce type d'interface permet de fournir des protocoles de virtualisation de réseau, d'interagir avec la matrice de commutation ou d'intégrer l'informatique distribuée. Les SBI les plus répandus sont le OpenFlow, Opflex, Open virtual switch database (OVSDB), etc. [6].

2.3.1 OpenFlow

Comme nous l'avons mentionné, l'OpenFlow ou OF est une API du type SBI, où elle se présente comme la première norme multifournisseur pour SDN, définie par l'ONF, qui a été développé à l'origine par une collaboration entre les chercheurs de l'université de Stanford avec l'université de Californie à Berkeley et qui est désormais gérée par l'ONF. Cette technologie a été créée, car les dispositifs de connexion, les commutateurs et routeurs physiques ou virtuels (pr hyperviseur) ont une capacité de programmation limitée, ce qui entraîne des difficultés dans la gestion et l'ingénierie du trafic pour les nouvelles infrastructures qui ont une architecture complexe, ainsi, des flux de trafic incohérents entre des matériels de réseau de différents fournisseurs. OpenFlow fournit cette cohérence en retirant le contrôle du matériel et en l'implémentant dans le logiciel. Le rôle de cette technologie sera plus compréhensif tout en présentant ces trois principaux éléments qui sont : Commutateur OpenFlow, canal OpenFlow et contrôleur OpenFlow [20] comme il est illustré dans la figure 2.2 et figure 2.3, où :

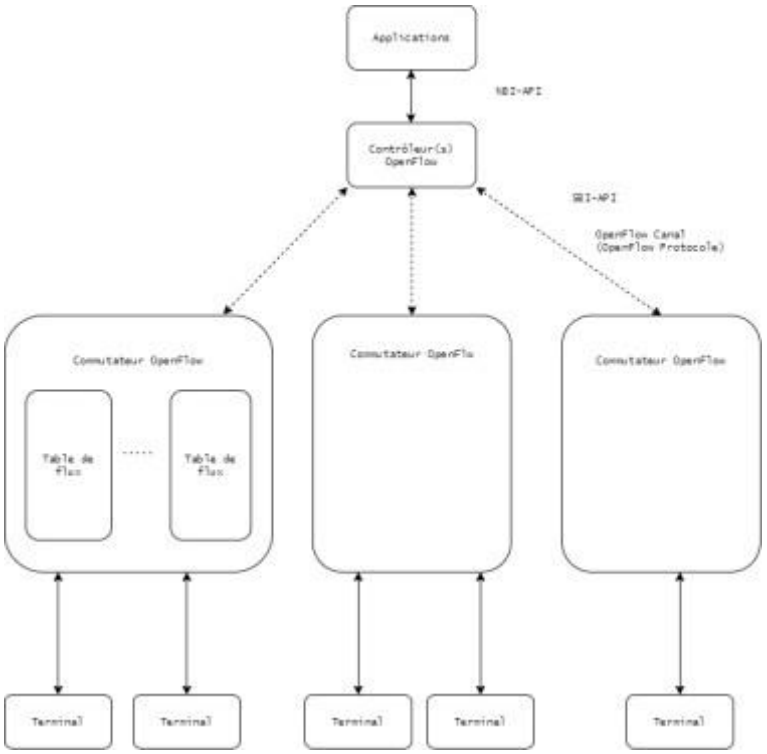

Figure 2.3: Architecture de la technologie OpenFlow[20].

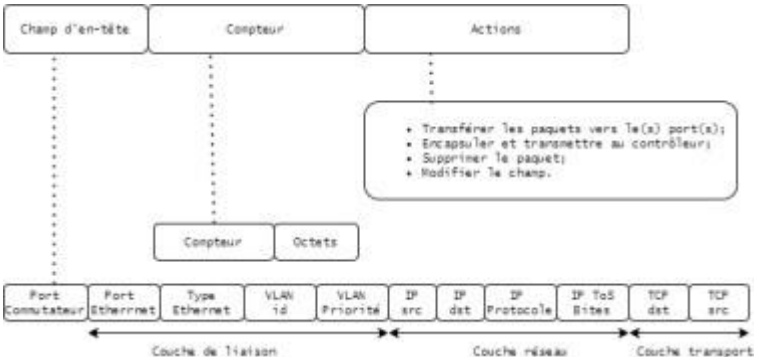

Figure 2.4: Structure de la table de flux [20].

• Les commutateurs sont des dispositifs physiques ou logiques propriétaire à openflow environnement, qui transmettent les paquets dans un environnement SDN à l'aide des tables appelées table de flux, où, il peut y avoir une ou plusieurs de cette table dans un seul commutateur OF. En particulier, une table de flux est composée d'une liste d'entrées de flux, chaque entrée contenant des champs d'en-tête, des compteurs et des actions. Les champs d'en-tête sont utilisés pour comparer les paquets et contiennent des informations telles que l'ID VLAN, les ports source et destination, l'adresse IP, etc. Les compteurs sont principalement utilisés pour conserver des statistiques sur les paquets, telles que le nombre de paquets, le nombre d'octets, etc. Les actions donnent des instructions sur la manière de traiter et de comparer les paquets dans un flux, telles que le transfert vers un port donné, le transfert vers un contrôleur et la suppression du paquet (voir la figure 2.3). Ces commutateurs sont gérés par les contrôleurs OF qui résident dans la couche de contrôle via le canal OF qui est sécurisé avec le protocole TLS (Transport Layer Security), où cette communication ou connexion se fait via un protocole propriétaire qui est le protocole de OpenFlow. Ce protocole est uniquement établi qu'entre le contrôleur et le commutateur.

• Le Contrôleur est une application centralisée qui présente le cerveau de l'environnement SDN, où il est responsable de la maintenance, la distribution et la mise à jour des politiques et des instructions aux périphériques du réseau. Il peut gérer et configure le trafic du flux, par un ajout ou suppression d'une entrée de flux dans une table de flux d'un commutateur. En outre, un commutateur OF peut se communiquer à un ou plusieurs contrôleurs OF, où, un nombre de contrôleurs élevé résulte une bonne fiabilité dans le réseau. Lorsque les opérations de la technologie OF démarrent, le commutateur doit se connecter à tous ses contrôleurs configurés en même temps, tandis que les messages pertinents ne peuvent être envoyés qu'au commutateur correspondant.

• Le canal est l'interface par laquelle le contrôleur OF et commutateurs OF peuvent se communiquer, tout en envoyant des messages via le protocole openflow, dont nous distinguons trois types de messages qui sont, contrôleur à commutateur, asynchrone et symétrique. Les messages contrôleur à commutateur sont envoyés par les contrôleurs pour gérer et inspecter correctement l'état du commutateur, notamment :

- ❖ Caractéristiques dont le commutateur les envoie au contrôleur après une demande de la part de ce dernier
- ❖ Configuration permet au contrôleur de définir et consulter les paramètres de configuration du commutateur ;

- Modifier l'état sont envoyés par le contrôleur pour gérer l'état des commutateurs. Ils sont utilisés pour ajouter/supprimer ou modifier les entrées de la table de flux ou pour définir les priorités des ports des commutateurs ;
- Envoi du paquet sont responsables de l'envoi et de la transmission des paquets de la part du commutateur à l'aide de table de flux.
- Les messages asynchrones sont envoyés sans que le contrôleur ne les sollicite auprès du commutateur. Ils sont envoyés par le commutateur pour mettre à jour le contrôleur sur les événements du réseau et les changements d'état du commutateur, nous citons :
- Paquet d'entrée transfert le contrôle du paquet au contrôleur dans le cas où les paquets qui n'ont pas d'entrée de flux correspondant ou si un paquet correspond à une entrée avec une action d'envoi au contrôleur, un message 'entrée de paquet est envoyé au contrôleur ;
- Elimination du flux informe le contrôleur de la suppression d'une entrée de flux d'une table de flux. Le message de modification de flux précise également si le commutateur doit envoyer un message de suppression de flux au contrôleur lorsque le flux expire ;
- Etat du port informe le contrôleur de l'état du port du commutateur, tel que le changement d'état du port (le cas de spanning tree IEEE 802.1D). Le commutateur doit envoyer l'état du port au contrôleur via les messages port-update ;
- Erreur aide le commutateur d'informer le contrôleur qu'il existe des problèmes.
- Les messages symétriques peuvent être initiés par le commutateur ou le contrôleur et envoyés sans sollicitation. Les trois types de messages symétriques sont les suivants :
- Hello présentent les messages d'accueil sont échangés entre le commutateur et le contrôleur lors de l'établissement de la connexion ;
- Echo sont utilisé pour indiquer la latence, la bande passante et/ou la disponibilité d'une connexion contrôleur-commutateur ;
- Vendeur offrent une fonctionnalité supplémentaire dans l'espace du type de message OpenFlow pour les futures révisions d'OpenFlow [20].

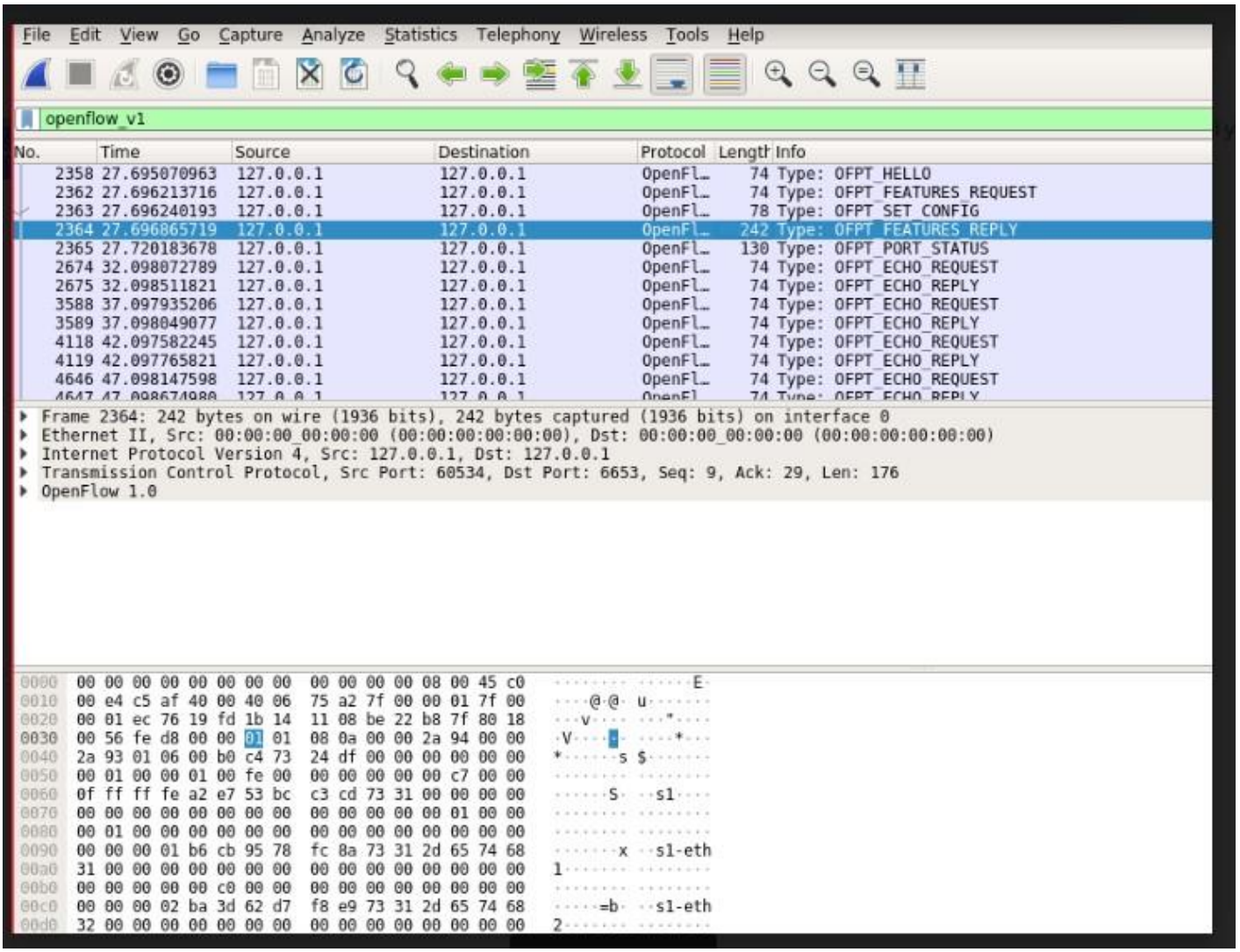

Figure 2.5 : Wireshark Capture flux

2.3.2 Avantages du SDN

Ce nouvel processus à apporter de nombreux avantages dont les plus répandus :

• Créer un réseau orchestral et contrôlable de manière centralisée ;

• Il est basé sur la programmation ce qui facilite sa configuration, la gestion du trafic, etc. Donc, avoir un réseau adaptatif, flexible, dynamique, automatique ;

• La virtualisation ;

• Une gestion simplifiée

2.3.3 SDN versus réseau traditionnel

Ce processus s'éloigne de l'architecture réseau traditionnelle où :

Réseau traditionnel

• Il est l'ancienne approche conventionnelle de mise en réseau ;

• Il est basé sur le matériel ;

• Ils travaillent en utilisant des protocoles ;

• C'est un contrôle distribué ;

• C'est une interface fermée ;

• Dans un réseau traditionnel, le plan de données et le plan de contrôle sont sur le même plan ;
• Il prend en charge la configuration statique/manuelle, ce qui prend plus de temps ;
• Ils ne sont pas utiles pour les nouvelles entreprises, ainsi, ils possèdent peu d'agilité et de flexibilité.

Réseau SDN

• Il est une approche de mise en réseau virtuelle ;
• Il est basé sur le logiciel ;
• Ils utilisent des API pour se configurer selon leurs besoins ;
• C'est un contrôle centralisé ;
• C'est une interface ouverte ;
• Dans le SDN, le plan de données et le plan de contrôle sont découplés physiquement et intègres de manière centralisée la logique du réseau au niveau du contrôleur ;
• Il prend en charge la configuration dynamique ;
• Ils aident les nouvelles entreprises grâce à sa flexibilité, son agilité et sa virtualisation.

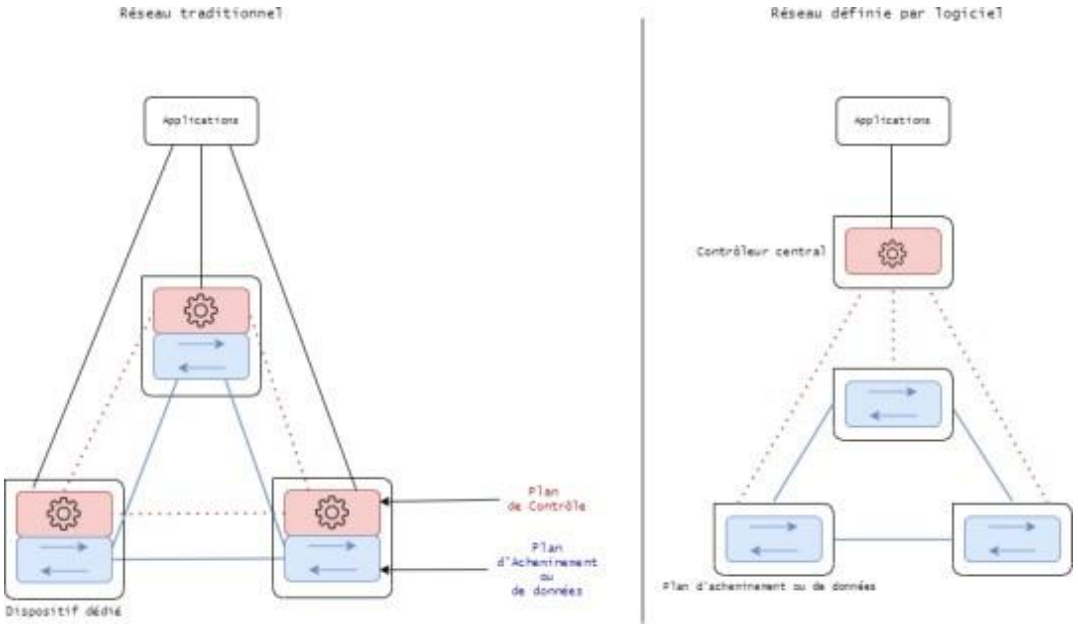

Figure 2.6: SDN versus le réseau traditionnel (architecture)

De nombreuses implémentations via cette nouvelle approche SDN existent avec les avantages qu'elles offrent, parmi lesquelles celle qui est mise en œuvre pour la solution de routage par segment (SR) dont nous allons en parler dans le chapitre suivant.

2.4 Conclusion

La mise en place d'une architecture de réseau et l'ingénierie du trafic sont devenues plus faciles à gérer, configurer et sécuriser grâce aux avantages offerts par le SDN et le routage par segments. Après avoir présenté les notions relatives à notre travail, le chapitre suivant portera sur la simulation de la solution proposée dans un environnement virtuel, qui sera présenté en détail.

Chapitre 3 : Simulation du réseau MPLS/SDN basé sur routage par segment

3.1 Introduction

La solution du routage par segment est une solution très récente et qui est en cours d'étude par de nombreuses méthodes en termes softwares et hardwares proposés par les ingénieurs, chercheurs et fournisseurs IT, parmi eux, le fournisseur Ericsson, qui nous a proposé d'implémenter cette solution par une simulation sous GNS3. Nous allons dans ce chapitre introduire l'environnement de travail ainsi que la simulation de la solution proposée étape par étape, par la suite, nous allons discuter les résultats obtenus

3.2 Routage par segment

Le routage par segment (SR) est une nouvelle technologie d'ingénierie du trafic et de commutation d'étiquettes développée par la collaboration des deux groupes de travail, Source Packet Routing in Networking (SPRING avec IPv6 de l'IETF dont son architecture a été définie sous la RFC 8402. Il s'agit d'une nouvelle méthode pour faire transiter les paquets sans les protocoles de signalisation (LDP et RSVP), basée sur le paradigme du routage à la source implémentée dans les deux plans de données, SR-MPLS et SR-IPv6, et qui permet de contrôler les chemins empruntés par le trafic dans le réseau, dans le but est de simplifier l'ingénierie et la gestion du trafic à travers les domaines du réseau et permet d'obtenir un réseau plus évolutif. Nous allons mieux comprendre le principe de fonctionnement de cette nouvelle technologie tout en commençant par la compréhension de ses composants fondamentaux dont elles seront présentées par la suite [4].

3.2.1 Composants fondamentaux du SR

Figure 3.1: Composants fondamentaux du SR [14].

• **Domaine SR**: Il s'agit d'un ensemble de nœuds qui participent aux protocoles SR. Dans un domaine SR, les nœuds peuvent être connectés à la même infrastructure physique comme le réseau d'une entreprise, ou à distance les uns aux autres comme dans le cas du VPN, où, ces nœuds qui sont gérés par la même entité administrative, peuvent exécuter des procédures d'entrée, de transit ou de sortie. Un domaine SR peut avoir plusieurs instances de protocoles qui est dans certaines souhaitable de subdiviser le réseau en plusieurs domaines SR, dont, chacun comprenant d'eux comprenant une ou plusieurs instances de protocole.

• **Chemin SR** : C'est le chemin optimal choisi et encodé dans l'en-tête du paquet comme une liste d'instructions ordonnées appelées "segments" au niveau du nœud SR source ou entrée pour atteindre le nœud SR de sortie, en d'autres termes, c'est le chemin choisi pour relier le nœud SR d'entrée avec le nœud SR de sortie.

• **Segment SR** : C'est un identifiant pour tout type d'instruction (topologique ou basée sur un service). Un segment SR peut avoir une instruction ou ensemble d'instructions qu'un nœud SR exécute pour transférer le paquet entrant dans une section de la topologie du réseau. Chaque segment est identifié par l'ID du segment (SID), qui consiste en un nombre entier non signé de 20 bits, alors, chaque routeur (nœud) et chaque lien (adjacence) est associé à un SID. SR défini de nombreux types de segments SR, dont les plus utilisés sont ceux qui sont distribués par le protocole IGP, qui sont les segments d'adjacence, de nœud et de préfixe.

• Segments d'adjacence :

Ce type de segment permet à un paquet de traverser un lien spécifique unidirectionnelle associée à une adjacence IGP entre deux nœuds, où le coût du chemin pour ce segment est généralement un saut. Il est identifié par un SID propriétaire appelée SID d'adjacence qui est distribué par les deux protocoles ISIS ou OSPF. Le segment d'adjacence est un segment local, le SID d'adjacence est donc localement unique par rapport à un routeur spécifique.

• Segments de nœud : C'est un segment préfix qui identifie un routeur spécifique. Il peut être attribué à une interface de bouclage ou à une interface non-bouclée. Il est identifié par un SID propriétaire, c'est le SID nœud.

• **Segments de préfixe** : Le segment de préfixe oriente le trafic sur le chemin le plus court vers la destination désirée en utilisant l'algorithme de routage spécifié dans le champ algorithme, dans la topologie et dans l'instance IGP où il est annoncé. Il est identifié par le ID de segment de préfixe où il est configuré manuellement et distribué par les deux protocoles ISIS ou OSPF. Ce segment est un segment global, donc, il est unique, visible et prend effet globale dans le domaine SR.

• **SR Global Block (SRGB**) : C'est l'ensemble des segments globaux du domaine SR, où, ces types de segments sont définis au niveau du domaine SR et ils seront choisis comment étant un segment typique, le segment qui a le plus court chemin vers la destination donnée dans le domaine SR. Si un nœud participe à plusieurs domaines SR, il existe un SRGB pour chaque domaine SR.

• **SR Local Block (SRLB)** : C'est une propriété locale d'un nœud SR. Si un nœud participe à plusieurs domaines SR, il y a un SRLB pour chaque domaine SR, donc les instructions seront selon le nœud SR qui distribue le segment.

• **Politique SR** : C'est une liste ordonnée de segments identifiée par un tuple, qui sont créés à partir de plusieurs méthodes (BGP, PCEP, etc), dont le nombre de segment d'une politique SR définie la profondeur de la liste des segments dans un domaine SR. Une politique SR peut être utilisée pour des raisons de TE, d'exploitation, d'administration et de maintenance (OAM) ou de réacheminement rapide (FRR)[4].

3.2.2 Principe de fonctionnement du SR

Tout commence lorsque le paquet arrive à l'entrée du domaine sr ou pour être plus précis au nœud d'entrée sr, où, ce paquet va soumis à une politique pour qu'il soit apte à le transmettre dans le domaine SR. Après

avoir vérifié si le paquet satisfait aux conditions de correspondance pour un chemin SR, le nœud d'entrée SR encapsule le paquet dans un tunnel SR qui traverse ce chemin, segment par segment. En d'autres termes, le nœud ajoute une pile de SID au niveau d'en-tête du paquet conformément aux conditions définies pour le domaine SR. Chaque chemin SR se termine par un nœud d'extrémité de segment, où, ce dernier va examiner l'en-tête pour déterminer si ce paquet est à la fin de son chemin et de trouver le prochain segment. Ce processus se poursuit jusqu'à ce que le paquet arrive à l'extrémité finale du segment, qui peut être le nœud de sortie du domaine SR. Lorsqu'un paquet arrive au nœud de sortie SR, ce nœud supprime les informations de l'en-tête SR encapsuler et transmet le paquet en fonction de son adresse IP de destination. Le SR peut être utilisé pour mettre en correspondance les paquets associés à un utilisateur final ou à une application avec des services de fonction réseaux spécifiques. Pour ce faire, il mappe un chemin vers l'endroit où le service sera appliqué et fournit des instructions sur le service et des informations supplémentaires sur le chemin entre la passerelle de service et le routeur de sortie du domaine SR. La figure ci-dessus illustre le principe du SR[4]. Le principe de SR se diffère d'une application à une autre, en terme de protocole, architecture ainsi d'autre paramètres, comme dans le cas du SR dans TE ou SR-TE se diffère des autres dans le chemin choisi pour acheminer les paquets, où il sera spécifié par les routeurs du réseau central du fournisseur au lieu du chemin le plus court calculé par le routeur ou par l'IGP, ainsi, la destination n'est pas consciente de la présence du tunnel[14].

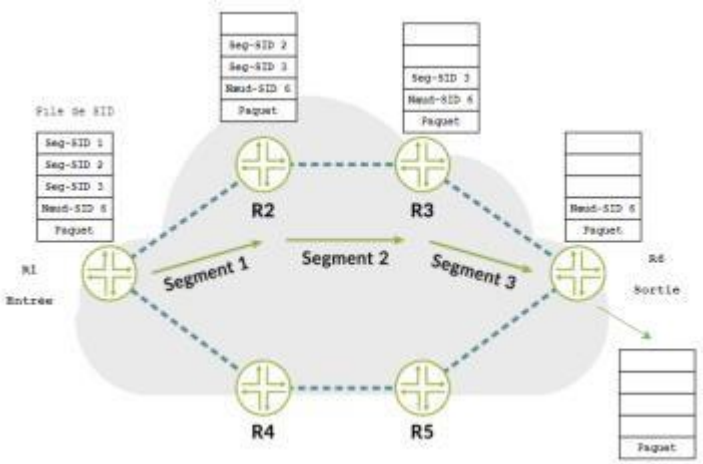

Figure 3.2: Principe du SR[14].

3.2.3 Avantages du SR

De nombreux avantages sont offert par le SR, dont nous citons :

• Optimiser, de simplifier et d'améliorer l'évolutivité des réseaux basés sur IP/MPLS, donc avoir un réseau flexible, agile à gérer ;

• Equilibrage des charges ;

• Une évolution plus fluide du réseau vers le SDN;

• Il assure une protection automatique du trafic sans aucune restriction topologique.

3.4 Protocole PCEP

Le protocole PCEP est un protocole TCP qui a été défini par l'IETF dans ses deux RFCs, RFC 5440 et RFC 4655 pour améliorer le calcul du chemin dans les réseaux qui utilisent le processus traffic engineering. Le principe de ce protocole est de permettre à un client de calcul de chemin (PCC) de demander des calculs de chemin de l'élément de calcul de chemin (PCE), par la suite, ce dernier va envoyer de sa par une réponse pour lui répondre. Le protocole PCEP permet la communication entre un PCC et un PCE, ou entre deux PCE. L'architecture de ce protocole est illustrée dans la figure 2.8, permet de mieux comprendre son principe, où :

• Le PCE est une entité du réseau, peut être un nœud, une application ou un dispositif, qui peut calculer un chemin à travers un réseau sur la base d'un graphe de réseau et en appliquant des contraintes de calcul en temps réel.

• Le PCC est toute application cliente demandant un calcul de chemin à effectuer par un PCE.

• Traffic Engineering Database (TED) est une base de données qui contient toutes les informations sur les ressources d'une topologie d'un réseau.

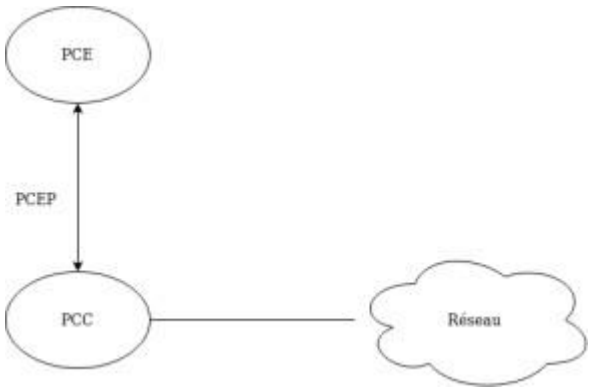

Figure 3.3: Architecture du protocole PCEP

Nous allons maintenant passer à la présentation de l'application

3.5 Environnement de travail

Nous allons présenter l'environnement logiciel et matériel que nous avons utilisé dans la simulation du réseau. Environnement matériel L'environnement matériel utilisé est un ordinateur portable avec les caractéristiques suivantes :

• Processeur : Intel(R) Core(TM) i5-1035G1
• Mémoire : 16Gb
• Disque dur : 250G SSD
• Système d'exploitation : Ubuntu 18.04

Environnement logiciel

GNS3

GNS3 (Graphical Network Simulator) est un logiciel open source qui permet de simuler et émuler des réseaux complexes tout en étant aussi proche que possible du fonctionnement des réseaux réels. GNS3 est constitué de deux composants logiciels :

• **Logiciel GNS3-all-in-one (GUI** : Il s'agit de la partie client du GNS3 et de l'interface utilisateur graphique (

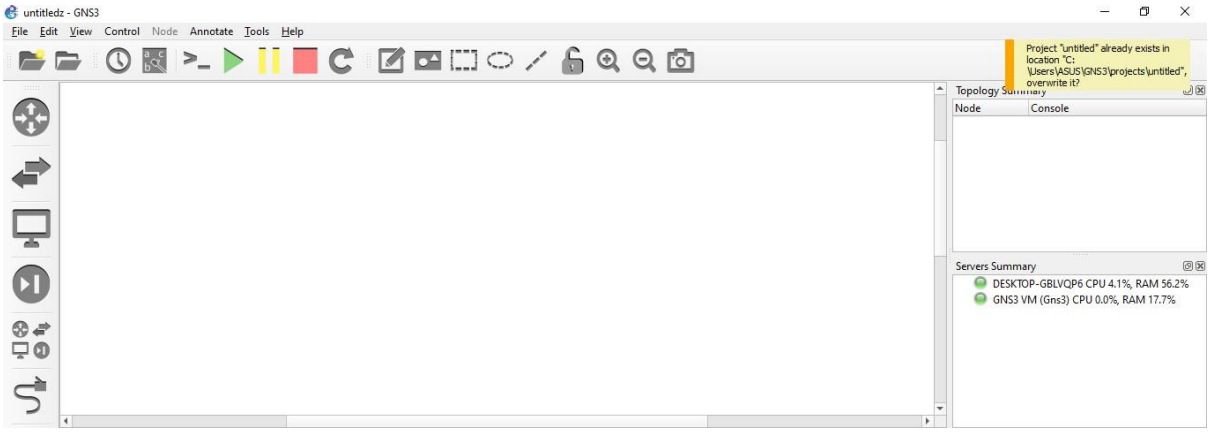

figure 3.4: GNS3 GUI

• **Serveur GNS3** : Les périphériques créés doivent être hébergés et exécutés par un processus serveur. Le serveur peut être installé comme un serveur local (standalone), une machine virtuelle local ou bien une machine virtuelle distante. Dans notre cas, nous utilisons des images Qemu et selon la documentation officielle de GNS3, Il est recommandé dans ce cas d'utiliser la machine virtuelle (voir la figure 3.2) [8].

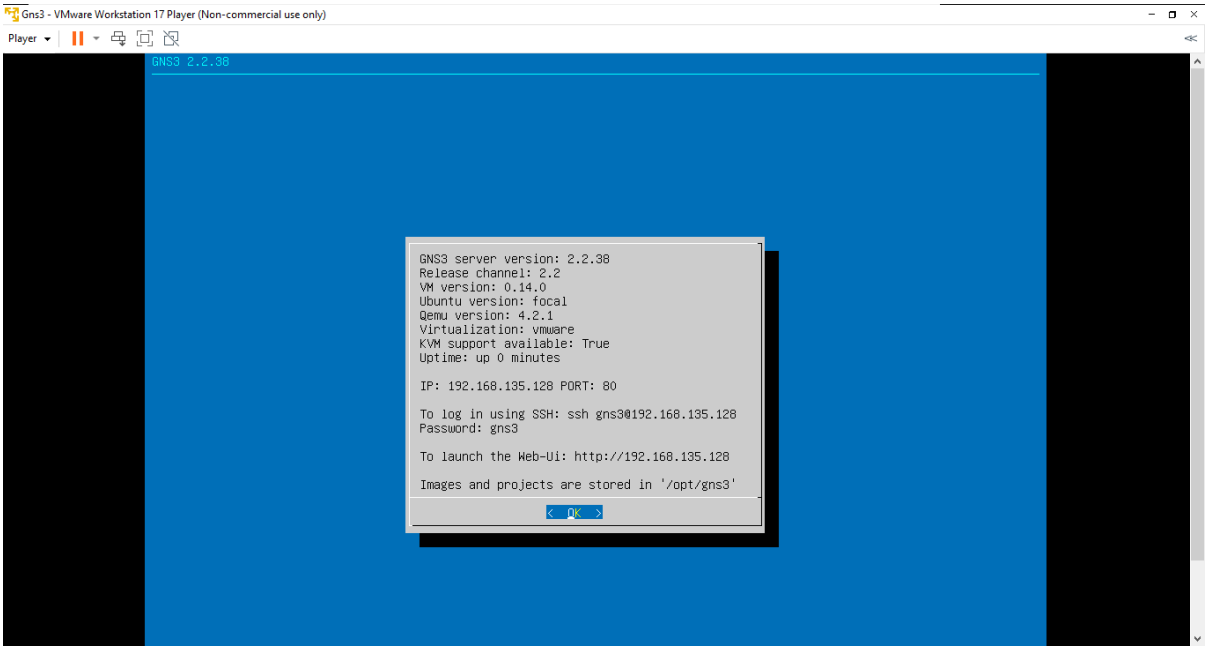

Figure 3.5 : GNS3 serveur VM (vmware)

OpenDayLight

OpenDayLight ou ODL est un contrôleur et un framework SDN open source hébergé par la fondation Linux. Il vise à développer des contrôleurs ouverts et normalisés pour les SDN. L'architecture de cet outil est illustrée dans la figure 3.3 [16].

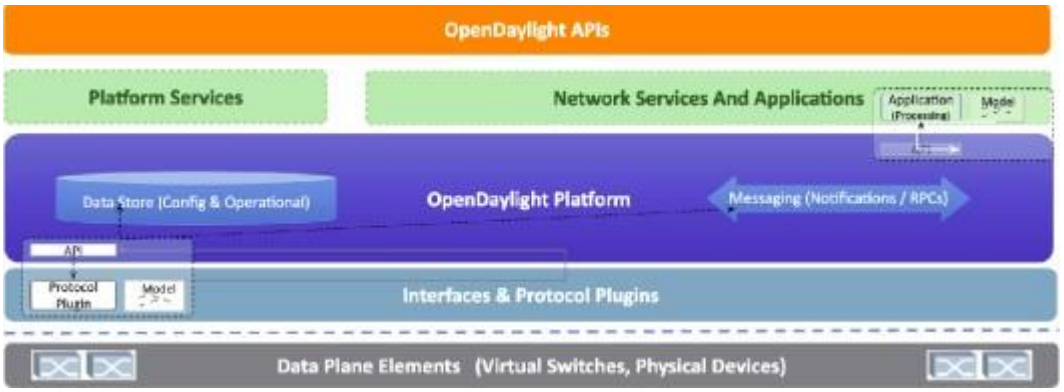

Figure 3.6: Architecture d'ODL[16].

Où :

• ODL API est une API northbound qui est utilisée pour communiquer avec la couche supérieure principalement basée sur REST. La couche d'abstraction de service orientée modèle rend les API REST conformément à la spécification RESTCONF basée sur les modèles YANG définis par les applications via des protocoles tels que HTTP. Comme chaque outil, ODL à aussi une partie d'authentification dont nous allons présenter dans la suite.

• AAA ou Authentification, l'Autorisation et la Comptabilité (Acounting) permet d'identifier les périphériques et contrôleur du réseau, où nous avons gardé l'identification par défaut, "admin" pour le nom d'utilisateur et le mot de passe, mais, nous pouvons rendre la sécurité plus robuste en modifiant cette configuration.

• Plateforme de contrôleur présente la partie cœur d'ODL, dont nous trouvons :

• NetVirt ou Network Virtualization est une plateforme de virtualisation de réseau qui permet de créer et de gérer les réseaux VPN.

• MD-SAL ou la couche d'adaptation de service pilotée par modèle est un composant middleware extensible qui fournit des fonctionnalités de messagerie et de stockage de données basées sur des modèles de données et d'interface définis par les développeurs d'applications.

• Pluggins ou enfichables permettent d'effectuer différentes tâches sur le réseau telles que, la collecte de statistiques sur le réseau.

• Les protocoles OpenFlow et OVBSD (Open vSwitch Database Management Protocol) sont présentés dans le chapitre précédent. La configuration des API d'ODL se fait dans la GUI de cette plateforme, mais il existe des applications qui ont facilité cette tâche dont celui que nous avons choisi : l'application Postman.

Application Postman

Postman est un outil de développement d'API (interface de programmation d'applications) qui permet de créer, de tester et de modifier des API. Il permet d'envoyer diffèrent types de requête HTTP qui sont GET, POST, PATCH, PUT et DELETE représentées en format JSON et XML, où ces méthodes sont envoyés via une url dont nous devons spécifier l'adresse où se trouve notre contrôleur opendaylight et le port tcp propriétaire à http (80)/https (8181)[18].

• **GET** pour récupérer des données d'un API.

• **POST** pour envoyer les nouvelles données d'un API .

•**PATCH** et **PUT** pour actualiser les données existantes.

•**DELETE** pour supprimer les données existantes.

3.3 Construction de la topologie

Dans cette section nous allons présenter l'architecture réseau de notre projet :

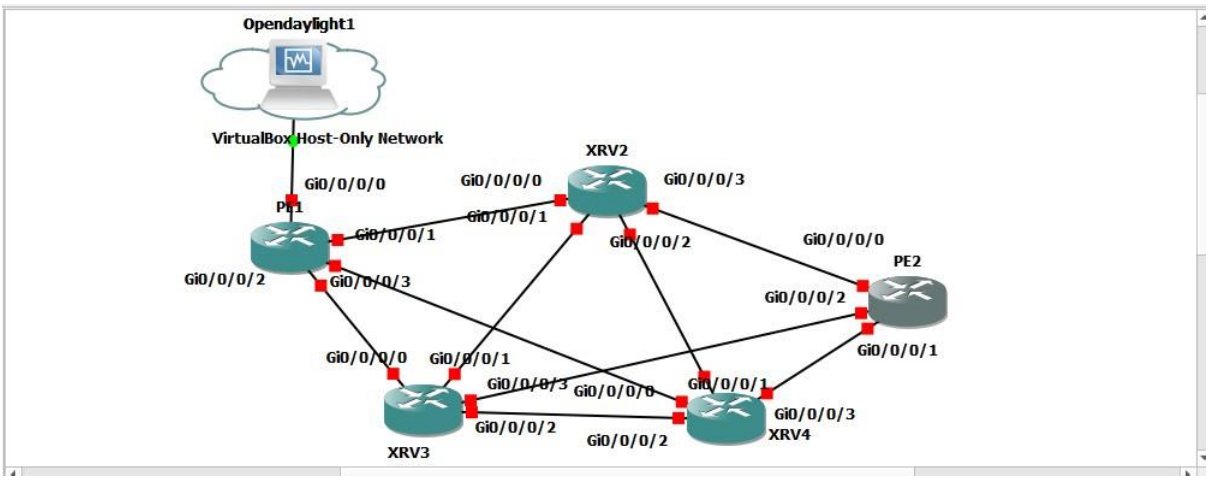

figure 3.7: Topology

Pour la topologie abordée, nous avons opté pour le routeur virtuel du fournisseur Cisco , le Cisco XRv
Cette machine virtuelle contient un seul processeur de routage (RP) avec une fonctionnalité de plan de contrôle et des interfaces réseau de carte de ligne (LC :line control) avec leurs fonctionnalités associées. . La figure 3.5 illustre l'architecture du XRV :

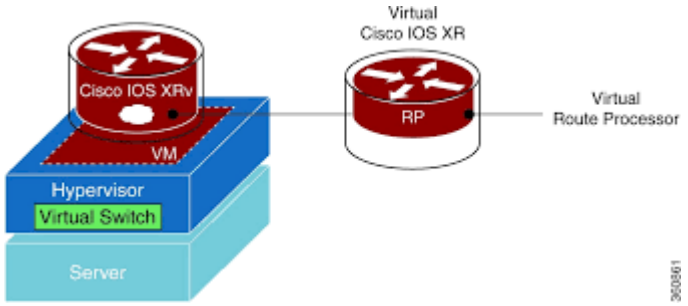

Figure 3.8 : architecture du XRV

Le routeur Cisco XRV nécessite des ressources importantes minimum 4Gb de RAM pour chaque routeur, et au vu des contraintes matérielles nous n'allons utiliser que 5 routeurs. Dans le cas d'un petit réseau ceci ne pose pas de problème car nous n'allons pas acheminer les paquets aux clients finaux [10]. Dans notre architecture nous avons utilisé un NAT Cloud connectant notre machine virtuelle (virtualbox) dans notre cas Ubuntu, dans laquelle est installé Opendaylight où nous communiqueront avec celle-ci avec Postman.

3.4 Configuration de la connectivité

Chaque routeur dispose de quatre interfaces ethernet pour lier les routeurs entre eux, à l'exception du routeur d'extrémité PE2.

Nous avons commencé par l'attribution des adresses ipv4 au contrôleur et aux interfaces des routeurs, permettant ainsi d'assurer l'acheminement du trafic entre les routeurs. 3.4.1 Plan d'adressage Nous avons attribué les adresses ip en suivant le plan d'adressage illustré dans la figure 3.6.

Nœud	Adresse de bouclage	Adresse de l'interface Ethernet de Gigabyte
Hôte odl	Pas de configuration	E0 :192.168.56.105
		E0 : DHCP
PE1	1.1.1.1	E0 :192.168.56.200
		E1 :192.168.12.1
		E2 :192.168.13.1
		E3 :192.168.14.1
XRV2	2.2.2.2	E0 :192.168.12.2
		E1 :192.168.23.1
		E2 :192.168.24.1
		E3 :192.168.25.1
XRV3	3.3.3.3	E0 :192.168.13.2
		E1 :192.168.23.2
		E2 :192.168.34.1
		E3 :192.168.35.1
XRV4	4.4.4.4	E0 :192.168.14.2
		E1 :192.168.24.2
		E2 :192.168.34.2
		E3 :192.168.45.1
PE2	5.5.5.5	E0 :192.168.25.2
		E1 :192.168.35.2
		E2 :192.168.45.2

Figure3.9 : Plan d'adressage

```
RP/0/0/CPU0:ios#show ip int br
Mon Jun  5 13:19:31.023 UTC

Interface                  IP-Address      Status          Protocol Vrf-Name
Loopback0                  1.1.1.1         Up              Up       default
tunnel-te1                 unassigned      Down            Down     default
tunnel-te2                 unassigned      Down            Down     default
tunnel-te3                 unassigned      Down            Down     default
tunnel-te4                 unassigned      Down            Down     default
MgmtEth0/0/CPU0/0          unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/0     192.168.56.200  Up              Up       default
GigabitEthernet0/0/0/1     192.168.12.1    Up              Up       default
GigabitEthernet0/0/0/2     192.168.13.1    Up              Up       default
GigabitEthernet0/0/0/3     192.168.14.1    Up              Up       default
GigabitEthernet0/0/0/4     unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/5     unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/6     unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/7     unassigned      Shutdown        Down     default
```

PE1

```
RP/0/0/CPU0:ios#show ip int br
Mon Jun  5 13:25:09.141 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      3.3.3.3         Up              Up       default
MgmtEth0/0/CPU0/0              unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/0         192.168.13.2    Up              Up       default
GigabitEthernet0/0/0/1         192.168.23.2    Up              Up       default
GigabitEthernet0/0/0/2         192.168.34.1    Up              Up       default
GigabitEthernet0/0/0/3         192.168.35.1    Up              Up       default
GigabitEthernet0/0/0/4         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/5         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/6         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/7         unassigned      Shutdown        Down     default
RP/0/0/CPU0:ios#
```

XRV3

```
RP/0/0/CPU0:ios#show ip int br
Mon Jun  5 21:06:17.966 UTC

Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      4.4.4.4         Up              Up       default
MgmtEth0/0/CPU0/0              unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/0         192.168.14.2    Up              Up       default
GigabitEthernet0/0/0/1         192.168.24.2    Up              Up       default
GigabitEthernet0/0/0/2         192.168.34.2    Up              Up       default
GigabitEthernet0/0/0/3         192.168.45.1    Up              Up       default
GigabitEthernet0/0/0/4         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/5         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/6         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/7         unassigned      Shutdown        Down     default
RP/0/0/CPU0:ios#
```

XRV4

```
Interface                      IP-Address      Status          Protocol Vrf-Name
Loopback0                      5.5.5.5         Up              Up       default
MgmtEth0/0/CPU0/0              unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/0         192.168.25.2    Up              Up       default
GigabitEthernet0/0/0/1         192.168.45.2    Up              Up       default
GigabitEthernet0/0/0/2         192.168.35.2    Up              Up       default
GigabitEthernet0/0/0/3         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/4         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/5         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/6         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/7         unassigned      Shutdown        Down     default
RP/0/0/CPU0:ios#
```

PE2

3.4.1 Configuration du protocole ISIS

Le réseau MPLS/TE que nous allons implémenter utilise le protocole ISIS pour calculer le chemin le plus court tout en permettant à chaque nœud du réseau de construire une carte de la connectivité du réseau sous la forme d'un graphe qui facilite la gestion du réseau. De ce fait nous allons configurer le protocole ISIS au niveau des 5 routeurs, et l'activer pour toutes les interfaces sauf l'interface de management qui ne nécessite par un routage et les interface de bouclage.

```
router isis 1
 is-type level-2-only
 net 49.0000.0000.0001.00
 distribute bgp-ls
 log adjacency changes
 address-family ipv4 unicast
  metric-style wide
  mpls traffic-eng level-2-only
  mpls traffic-eng router-id Loopback0
  redistribute connected
  redistribute static
  segment-routing mpls
 !
 interface Loopback0
  circuit-type level-2-only
  address-family ipv4 unicast
   prefix-sid absolute 17001
  !
 !
 interface GigabitEthernet0/0/0/1
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/2
  point-to-point
  address-family ipv4 unicast
  !
 !
 interface GigabitEthernet0/0/0/3
  point-to-point
```

Figure 3.10 : configuration isis

La vérification de cette configuration sera faite via un ping entre ces routeurs (figure 3.11).

```
RP/0/0/CPU0:ios#ping 3.3.3.3
Mon Jun  5 21:19:22.824 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/7/9 ms
RP/0/0/CPU0:ios#ping 4.4.4.4
Mon Jun  5 21:19:29.294 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms
RP/0/0/CPU0:ios#ping 5.5.5.5
Mon Jun  5 21:19:33.924 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/9 ms
RP/0/0/CPU0:ios#
```

Figure 3.11 : Vérification de la connectivité entre les routeurs.

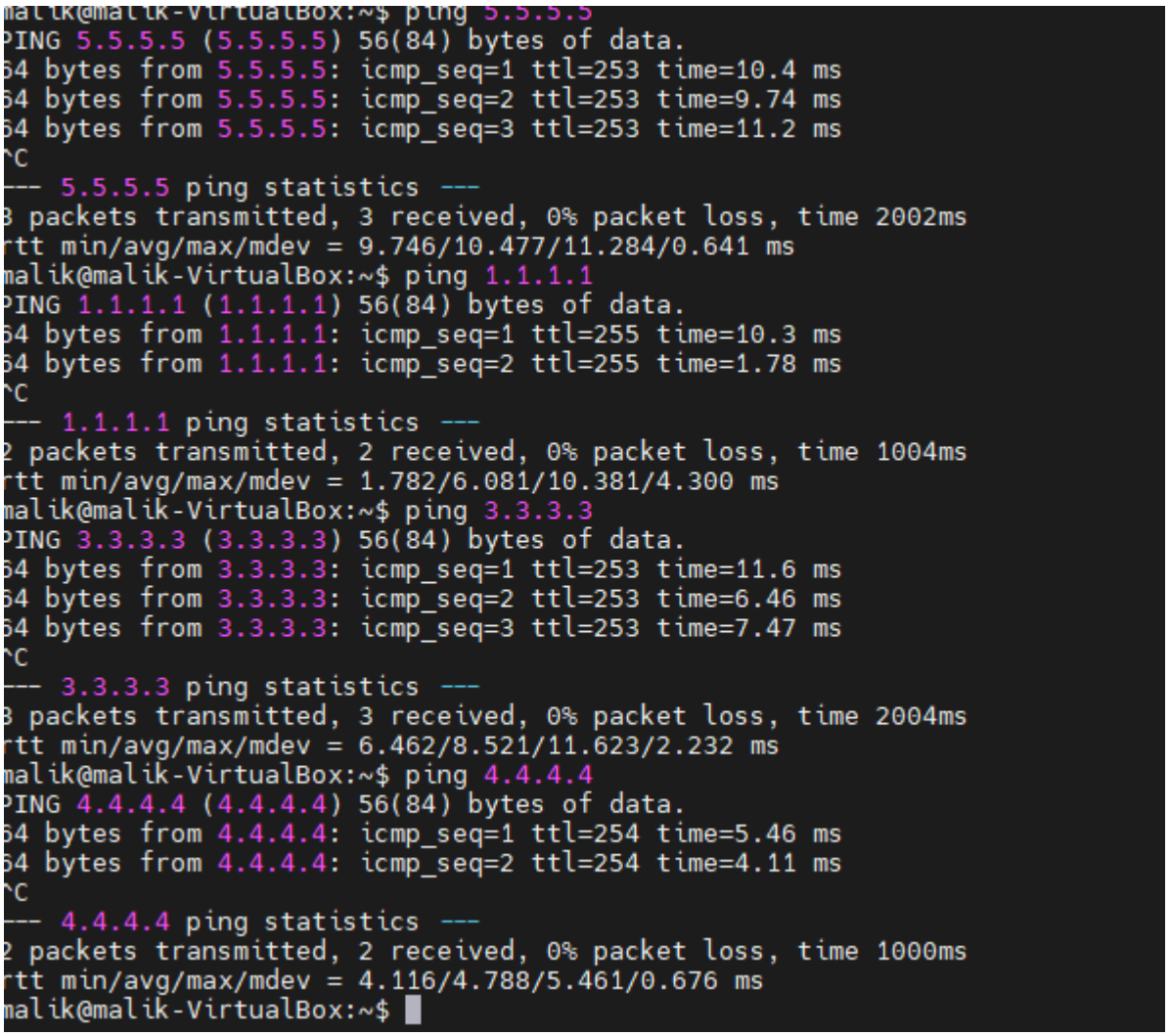

Figure 3.12: Opendaylight connectivite entre les routeurs.

La connectivité entre l'hôte est les routeurs est réussie aussi.
L'étape suivante sera la configuration du réseau MPLS/SDN.

3.5 Configuration d'un réseau MPLS/SDN

Le réseau MPLS/SDN est composé d'un routeur Provider Edge (PE) et les routeurs P, où les routeurs PEs présentent l'interface qui connectent le core du réseau avec un nœuds externe afin d'acheminer le trafic rapide et un contrôle robuste en utilisant les étiquettes LSP. Le contrôle d'un réseau MPLS est amélioré avec le réseau SDN, qui sera le nœud externe, où cette collaboration introduit le réseau MPLS/SDN . Les informations échangées entre le cœur du réseau et le nœud externe, sont disposés au niveau des routeurs PE1 et PE2 , de ce fait, il doit y avoir le protocole BGP interne entre ces deux nœuds pour transporter les informations de routage pour le réseau et les étiquettes MPLS entre eux. Ainsi un protocole de communication entre le routeur PE et le contrôleur SDN, qui est le pcep, où, le PCE est le contrôleur SDN etle PCC sont les routeurs PE. Les routeur PE dans notre cas sont les routeurs PE1/PE2, les routeurs P sont les routeurs XRV2 ,XRV3,XRV4. De ce fait, dans les sections suivantes nous allons présenter la configuration établie pour le réseau MPLS/SDN étape par étape.

Nous avons commencer par la configuration du EBGP au niveau du routeur est illustré dans la figure 3.14, où :

```
router bgp 65500
 bgp router-id 192.168.56.200
 address-family link-state link-state
 !
 neighbor 192.168.56.105
  remote-as 65500
  address-family link-state link-state
  !
 !
```

```
router bgp 65500
 bgp router-id 5.5.5.5
 address-family ipv4 unicast
 !
 address-family link-state link-state
 !
 neighbor 192.168.56.105
  remote-as 65500
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family link-state link-state
  !
```

Figure 3.13 :BGP-LS configuration PE1 et PE2

L'adresse du contrôleur opendaylight 192.168.56.105 mentionnée comme neighbor va permettre aux routeurs PE1 et PE2 de transmettre les informations de notre réseau à opendaylight.
En résumé, les informations d'état de liaison collectées par le protocole IGP dans le domaine (dans ce projet, le protocole ISIS) seront transmises par le protocole BGP-LS au pair BGP d'ODL. Pour tenir les autres routeurs à jour si un routeur voisin se connecte ou se déconnecte.

À l'intérieur de l'address-family ipv4 unicast, nous avons d'abord configuré le style de métrique "wide", ce qui signifie que l'IS (Intermediate System) ne peut recevoir que des TLV (Type-Length-Value) de nouveau style. Pour permettre à un routeur de propager des informations sur les liens MPLS (Multiprotocol Label Switching) via IS-IS dans le niveau configuré, il doit être configuré à l'intérieur de l'instance IS-IS (voir figure 3.10). Dans notre cas, nous avons configuré la distribution des liens MPLS uniquement dans le niveau 2 et nous avons spécifié l'adresse 192.168.56.200 pour PE1 et Loopback0 comme identifiant du routeur PE2. Enfin, nous avons configuré les interfaces annoncées par IS-IS. Tout d'abord, Nous avons également spécifié le Node-SID (Segment IDentifier) qui sera lié à l'adresse Loopback du routeur. Les deux autres interfaces sont configurées en tant que liens GigabitEthernet en mode point-à-point et avec une famille d'adresses unicast.

3.5.1 Configuration MPLS, PCEP et Segment routing

Cependant, en cas d'utilisation du contrôleur SDN ODL (OpenDayLight), en configurant l'adresse IP source, nous pouvons établir la connexion PCEP (Path Computation Element Protocol), mais nous ne pouvons jamais ajouter de LSP (Label Switched Path). En effet, l'adresse IP loopback du routeur utilisé pour BGP-LS doit correspondre à l'adresse IP PCEP (ici, avec cette configuration, l'adresse IP PCEP serait 192.168.56.105). C'est ainsi que les deux topologies sont liées. Donc, lorsque nous utilisons ODL, la configuration du PCE sur le routeur doit être la suivante.

```
pce
 peer source ipv4 1.1.1.1
 peer ipv4 192.168.56.105
 !
```

Figure 3.14 :peer ODL

Notez que pour que PCE-P s'établisse, votre instance ODL doit être en mesure de router vers les adresses loopback de votre réseau XRv. Par exemple, si ces adresses sont dans le réseau 10.10.0.0/16, vous pouvez faire quelque chose comme ceci sur l'hôte ODL sudo route add -net 192.168.56.200/30 gw 192.168.56.200 Si vous souhaitez que cette configuration persiste après un redémarrage, vous pouvez modifier le fichier /etc/network/interfaces (en supposant que vous utilisez Ubuntu) et ajouter up route add -net 192.168.56.200/16 gw 192.168.56.200

PCE doit être configuré sur nos routeur PE pour activer la création de chemin. De plus, dans OpenDaylight, dans le module PCE, nous allons configurer l'adresse de mise en relation. Ici, il doit être précisé que nous voulons effectuer un calcul de chemin en mode SR (Segment Routing) . Tout d'abord, en tant que stateful-client, cela signifie que le routeur ajoutera un TLV (Type-Length-Value) de capacités stateful lors de l'ouverture de la nouvelle session. De plus, nous pouvons configurer la délégation de tous les tunnels actifs au PCE. En réalité, cette commande permet au contrôleur SDN de modifier les LSP existants tout en calculant de nouveaux chemins. Cela est utile si le PCE souhaite réoptimiser les tunnels traffic engineering. L'identifiant du speaker est toujours l'ID de l'interface de bouclage du routeur. Pour configurer la plage d'IDs de tunnel à utiliser pour les demandes d'instanciation de PCE stateful , nous utilisons la commande "auto-tunnel-pcc" et nous spécifions l'ID de tunnel minimum et maximum. la ré optimisation est configurée pour le nombre de secondes spécifiée , ce qui signifie que l'installation de nouveaux LSP avec de nouveaux labels après la réoptimisassions du tunnel se produira dans le nombre de secondes spécifié

```
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
 interface GigabitEthernet0/0/0/3
 !
 pce
  peer source ipv4 1.1.1.1
  peer ipv4 192.168.56.105
  !
  segment-routing
  logging events peer-status
  stateful-client
   instantiation
   cisco-extension
  !
 !
 logging events all
 auto-tunnel pcc
  tunnel-id min 1 max 99
 !
 reoptimize timers delay installation 0
!
segment-routing
!
end
```

Figure 3.15: segment-routing/stateful-client configuration

Tous les routeurs s'ont été vu attribués un segment-ID (NODE-SID) au niveau de leurs adresses de loopback respectives

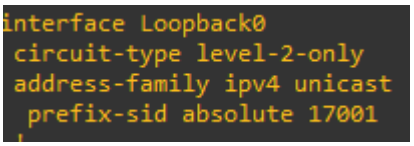

Figure 3.16 :node SID 17001 PE1

17002 pour XRV2
17003 pour XRV3
17004 pour XRV4
17005 pour PE2
Dans ce réseau, nous avons élu PE1 pour être un speaker BGP et redistribuer toutes les informations IGP vers OpenDaylight.
Les routeurs XRV2, XRV3, XRV4 ont une configuration simple des interfaces ainsi que d'ISIS. MPLS-TE doit être activé sur ces routeurs principaux.
La configuration BGP est présentée dans la figure ci-dessous. Comme pour IS-IS, tout d'abord, une instance BGP doit être ouverte. Encore une fois avons spécifié l'adresse IP de bouclage du routeur en tant qu'ID du routeur. Le voisin BGP est le contrôleur SDN, . Le contrôleur SDN appartient à un AS distant.

3.6 Configuration d'Opendaylight

Dans ODL ,le module BGP qui est placé '/etc/opendaylight/karaf' dans le fichier 41-

bgp-example.xml doit être reconfiguré. Dans la ligne 81, nous avons spécifié

L'interface de gestion du speaker BGP (PE1).

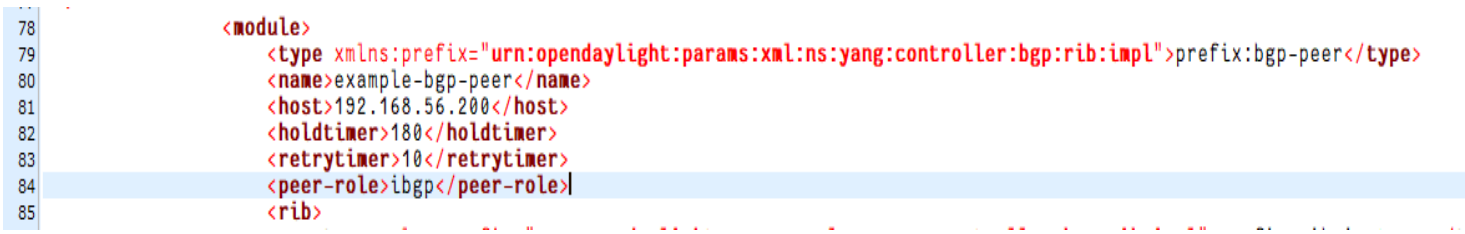

Figure 3.17 :bgp speaker pe1 address

3.7 MPLS-TE et SR-TE Tunnel Setup avec OpenDayLight

Pour l'établissement d'un nouveau tunnel MPLS, nous devons envoyer une commande REST à PCEP

Module contenu à l'intérieur d'ODL et demandez l'ajout de LSP. À l'intérieur de l'application Postman, nous devrions définir les paramètres pour établir la connexion à ODL et envoyer le bon modèle de données dans le Format XML ou JSON. L'exemple de création de tunnel PCEP est illustré ci-dessous :

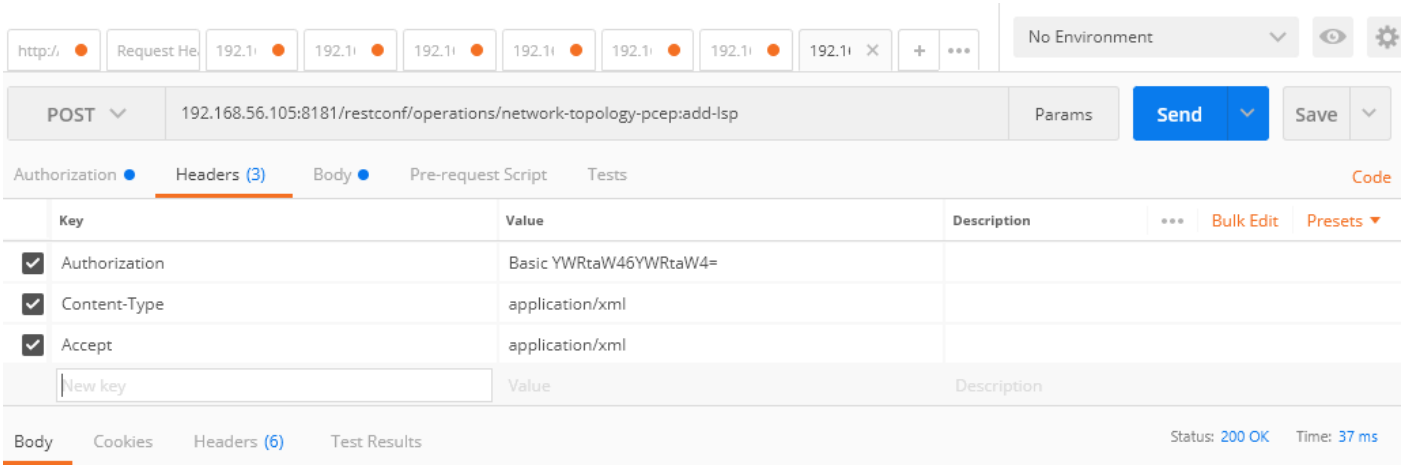

Figure 3.18 :application postman

Au début, L'action Post doit être sélectionnée et doit être envoyé en module d'odl suivant **192.168.56.105:8181/restconf/operations/network-topology-pcep:add-lsp**

Les 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> content-type : application/xml

• header > accept : application/xml

Ensuite, nous devons remplir le corps de notre message. Pour ce faire, nous devons d'abord définir quel routeur est l'entrée, quel routeur est la sortie et l'ERO (Explixit object route) signifiant quels nœuds doivent être passés pour ce LSP.L'Exemple suivant sert à ajouter un tunnel LSP du nœud PE1 au nœud PE2, en passant par un chemin explicite contenant les routeurs PE1>XRV3>XRV4>PE2.

```
<input xmlns="urn:opendaylight:params:xml:ns:yang:topology:pcep">
<node> pcc://1.1.1.1</node>
<name>test15</name>
<arguments>
<lsp xmlns="urn:opendaylight:params:xml:ns:yang:pcep:ietf:stateful">
<delegate>true</delegate>
<administrative>true</administrative>
</lsp>
<endpoints-obj>
<ipv4>
<source-ipv4-address>1.1.1.1</source-ipv4-address>
<destination-ipv4-address>5.5.5.5</destination-ipv4-address>
</ipv4>
</endpoints-obj>
<ero>
<subobject>
<loose>false</loose>
<ip-prefix>192.168.13.1/30</ip-prefix>
</subobject>
<subobject>
<loose>false</loose>
<ip-prefix>192.168.34.1/30</ip-prefix>
</subobject>
<subobject>
<loose>false</loose>
```

```
subobject>
loose>false</loose>
ip-prefix>192.168.42.1/30</ip-prefix>
/subobject>
/ero>
/arguments>
network-topology-ref xmlns:topo="urn:TBD:params:xml:ns:yang:network-topology">/topo:network-topology/topo:topology[topo:topology-id="pcep-topology"]</network
   -topology-ref>
/input>
```

Figure 3.19 :label switching path de PE1 a PE2

La ligne 2 définit l'adresse IP du nœud PCEP auquel ODL appliquera la

configuration La ligne 6 délègue le contrôle de ce LSP au PCE qui est ODL, donc

que ce tunnel ne peut pas être supprimé ou modifié par CLI sur le routeur.

Pour vérifier le tunnel à créer correctement, il faut le tester en insérant la commande

"show mpls traffic-engineering tunnels ", comme sur la figure ci-dessous :

```
Signalled-Name: pe1-to-pe2
Status:
  Admin:    up Oper:   up   Path:  valid   Signalling: connected

  path option 10, (verbatim) type explicit (autopcc_te4) (Basis for Setup)
    Protected-by PO index: 20
  G-PID: 0x0800 (derived from egress interface properties)
  Bandwidth Requested: 0 kbps  CT0
  Creation Time: Mon Jun  5 11:41:58 2023 (02:55:30 ago)
Config Parameters:
  Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
  Metric Type: TE (global)
  Path Selection:
    Tiebreaker: Min-fill (default)
  Hop-limit: disabled
  Cost-limit: disabled
  Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
  AutoRoute: disabled  LockDown: disabled   Policy class: not set
  Forward class: 0 (default)
  Forwarding-Adjacency: disabled
  Autoroute Destinations: 0
  Loadshare:          0 equal loadshares
  Auto-bw: disabled
  Fast Reroute: Disabled, Protection Desired: None
  Path Protection: Not Enabled
  BFD Fast Detection: Disabled
  Reoptimization after affinity failure: Enabled
  Soft Preemption: Disabled
Auto PCC:
  Symbolic name: pe1-to-pe2
  PCEP ID: 5
  Delegated to: 192.168.56.105
  Created by: 192.168.56.105
History:
  Tunnel has been up for: 00:02:09 (since Mon Jun 05 14:35:20 UTC 2023)
  Current LSP:
    Uptime: 00:02:09 (since Mon Jun 05 14:35:20 UTC 2023)
```


Figure 3.20 :Tunnel PE1-PE2

On vois que notre controleur ODL a realisé ce tunnel PE1-PE2

3.8 Segment Routing

Pour ajouter un nouveau tunnel de routage par segment, comme pour la création de tunnel MPLS, nous devons utiliser Postman avec les mêmes paramètres de connexion, et nous devons envoyer la configuration au même module PCEP d'ODL. La différence est que, pour ajouter des tunnels SR, nous devons définir la destination et l'itinéraire du tunnel en spécifiant numéros SID.

Ici, j'ai ajouté un chemin SR de PE1 comme source, à PE2 comme destination, forçant le tunnel à passer explicitement par XRV3. Parmi la partie centrale du réseau dans ce scénario, lorsqu'une partie centrale du routeur reçoit un paquet, car il ne prend pas en charge le routage de segment, il achemine le paquet basé sur l'algorithme du chemin le plus court vers l'adresse IP correspondante du prochain saut défini pour le tunnel. en envoyant cette configuration à opendaylight :

```
<subobject>
  <loose>false</loose>
  <sid-type xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">ipv4-node-id</sid-type>
  <m-flag xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">true</m-flag>
  <sid xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">17001</sid>
  <ip-address xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">192.168.13.1</ip-address>
</subobject>
<subobject>
  <loose>false</loose>
  <sid-type xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">ipv4-node-id</sid-type>
  <m-flag xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">true</m-flag>
  <sid xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">17003</sid>
  <ip-address xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">192.168.34.1</ip-address>
</subobject>
<subobject>
  <loose>false</loose>
  <sid-type xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">ipv4-node-id</sid-type>
  <m-flag xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">true</m-flag>
  <sid xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">17004</sid>
  <ip-address xmlns="urn:opendaylight:params:xml:ns:yang:pcep:segment:routing">192.168.45.1</ip-address>
</subobject>
</ero>
```

Figure 3.21 :création tunnel basé sur les SID

Nous avons spécifié explicitement qu'il doit passer par les routeurs ayant comme préfixe 17001-17003-17004

3.9 Conclusion

Ce travail a été mis en place pour étudier le paradigme de routage source appelé Segment Routing et le comparer à l'ingénierie du trafic MPLS. Dans un premier temps, nous avons discuté du Segment Routing : un aperçu général et ses avantages dans le contexte du réseau défini par logiciel (SDN). L'implémentation a considéré la création de réseau dans GNS3, la configuration du contrôleur SDN et sa connexion au réseau, la création de tunnels MPLS-TE et de Segment Routing à l'intérieur du réseau en utilisant le contrôleur SDN.

L'un des principaux objectifs de ce travail était l'utilisation de l'émulateur de réseau GNS3 comme plateforme gratuite avec presque aucun inconvénient par rapport aux deux autres méthodes précédemment essayées par d'autres chercheurs.
Dans ce travail, j'ai présenté deux façons différentes d'obtenir des numéros SID (Segment Identifier) : la première façon possible est d'obtenir les numéros SID directement à partir des routeurs. La deuxième façon consiste à extraire les numéros SID à partir de la base de routage BGP (Border Gateway Protocol) dans OpenDayLight. Ces numéros SID peuvent être utilisés comme principal paramètre pour toute application liée au segment routing.
En ce qui concerne les travaux futurs, on peut utiliser ces numéros SID pour créer une application graphique en tant que partie de l'orchestrateur de réseau, pour l'exploitation et la maintenance des tunnels de segment routing.
En utilisant GNS3 comme plateforme gratuite et sa fonctionnalité pour être exécuté sur différents ordinateurs en laboratoire, tout en ayant la possibilité de connecter ces ordinateurs par l'intermédiaire d'un commutateur matériel, cela offre la possibilité de créer un réseau étendu pour évaluer différents scénarios aussi larges que les réseaux réels disponibles dans le monde entier.

Conclusion Générale

L'infrastructure du réseau MPLS/SDN devient de plus en plus compliquée en raison du trafic élevé du réseau, ce qui nécessite une consommation de stockage élevée et un coût de déploiement élevé, qui ont fait apparaîtrele besoin de nouvelles technologies pour résoudre ces problèmes. Une solution très récente a été introduite par l'organisation IETF, permettant de répondre à ce besoin, c'est le routage par segment. L'architecture decette solution a permet de réunir le concept du routage par segment, les protocoles de routage IP/MPLS et les fonctionnalités SDN d'une manière unique tout en offrant un réseau optimisé, flexible, simple à implémenter et une sécurité robuste. Son principe est d'acheminer le trafic entre les routeurs du réseau Internet avec des SIDs encapsulés dans l'entête du paquet transmis en se basant sur la commutation des paquets du réseau et le contrôle MPLS/SDN . Le routage par segment est actuellement mis en œuvre par de nombreux fournisseurs informatiques avec les améliorations qu'ils offrent pour les réseaux MPLS, parmi lesquels le fournisseur Ericsson. Ce qui a fait l'objectif de notre mémoire qui consistait à étudier le routage par segment tout en comprenant son principe qui permet d'acheminer le trafic en utilisant SID et à tester ses capacités d'ingénierie du trafic dans les réseaux MPLS en construisant un environnement de test virtualisé via l'outil GNS3, qui a été atteint. Ainsi, le routage par segment et le réseau MPLS/SDN sont des technologies nouvelles, très vastes et complexes qui ont rendu difficile la compréhension de certaines notions avec la bibliographie restreinte existante, au début. Avec la poursuite et le soutien des deux encadreurs, nous avons pu acquérir et conforter des notions théoriques et pratiques sur l'acheminement du trafic dans les réseaux MPLS/SDN avec la solution routage par segment. Nous avons également appris à simuler et a programmer l'acheminement d'un réseau avec les différents types de routages sous GNS3. Nous avons appris aussi à utiliser deux nouveaux outils, Opendaylight et Postman, qui aident à optimiser la gestion du réseau simuler. Parallèlement à ce développement, un effort continu est déployé pour planifier les technologies futures ou modifier celles existantes pour satisfaire toujours les exigences des utilisateurs. En perspective, nous pensons améliorer notre travail avec des ressources plus performantes afin de s'approcher du réseau réel. Nous aimerions également tester cette simulation avec des routeurs réels pour atteindre l'objectif de cette solution dans la réalité.

Bibliographie

[1] Cisco Networking Academy. Guide d'accompagnement des protocoles de routage. Cisco Press, 2014.
[2] Cisco Networking Academy. CCNA. Cisco Press, 2020.
[3] et al Benbella Benduduh, Jean Marc Fourcade. Protocole MPLS.
[4] S. Previdi et al C. Filsfils. RFC 8402. 2018.
[5] Cisco. IP Routing. 2005.
[6] K. Pentikousis et al E. Haleplidis. RFC 7426. 2015.
[7] R. Callon E. Rosen, A. Viswanathan. RFC 3031. 2001.
[8] GNS3. Getting Started with GNS3.
[9] Huawei Technical Guide. Qu'est-ce que mpls. https://support.huawei.com/enterprise/en/doc/EDOC1100118961, 2019. Consulté le Mai 2021.
[10] Cisco. Architecture of a XRV stance.
[11] Y. Rekhter K. Lougheed. RFC 1105. 1989.
[12] G. Malkin. RFC 2453. 1998.
[13] J. Moy. RFC 2328. 1998.
[14] Juniper Networks. qu'est-ce que SR. 2019.
[15] Juniper Networks. Juniper networks glossaire. https://www.juniper.net/documentation/en_US/release-independent/glossary-topic-122763.html#symbols, 2021. Consulté le Mai 2021.
[16] OpenDayLight. OpenDaylight Downloads
[17] D. Oran. RFC 1142. 1990.
[18] Wang, An; Guo, Yang; Hao, Fang; Lakshman, T.; Chen, Songqing (2 December 2014). "Scotch: Elastically Scaling up SDN Control-Plane using vSwitch based Overlay" (PDF). ACM CoNEXT.
[19] Open Networking Foundation: Software-Defined Networking (SDN)https://www.opennetworking.org/sdn-resources/what-is-sdn/
[20] Software-Defined Networking: The New Norm for Networks https://www.cisco.com/c/en/us/solutions/data-center-virtualization/software-defined-networking-sdn.html