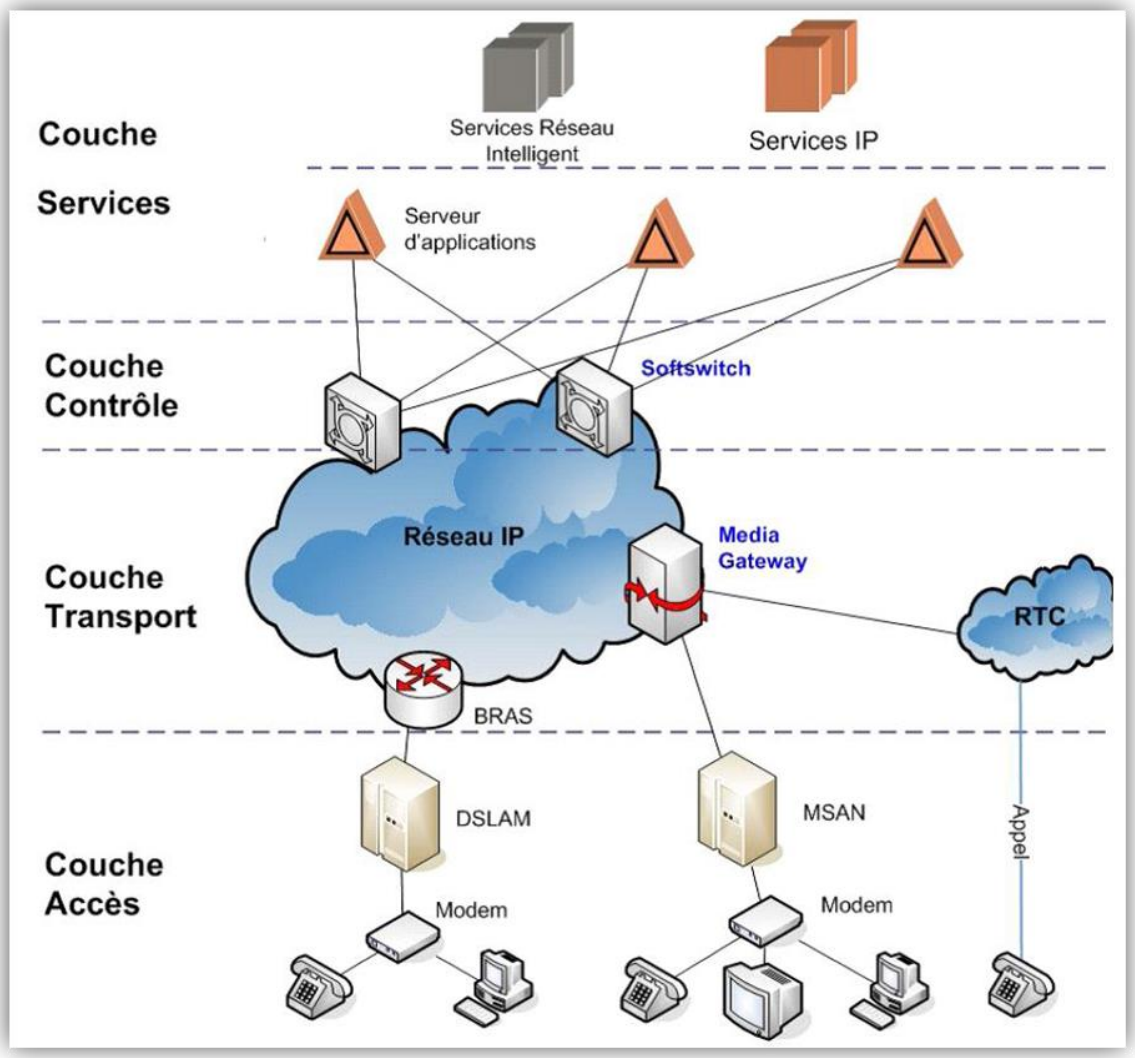
Maquette de Configuration de Modèle BackBone IP/MPLS

Instructeur : Mr Tarek Hdiji

Introduction NGN

Un réseau *Next Generation Network* (NGN) est un réseau multiservices prenant en charge les données, la voix et la vidéo. Une telle architecture permet d'offrir un réseau cœur à commutation de paquets pouvant transporter toutes les informations via de multiples technologies et ce avec une QoS garantie. De plus, elle prend en charge la notion de la mobilité généralisée ; les réseaux NGN assurent l'accès aux services et la cohérence entre ces derniers quelque soit le lieu où la technologie d'accès fixe ou mobile.

Parmi les points les plus importants qu'un réseau NGN peut offrir est la séparation entre le plan de commande à savoir la signalisation et les commandes et celui de transport et commutation. Ainsi, une architecture NGN est structurée selon un modèle en couches décrit par la figure ci-dessous.



Modèle NGN

L'architecture NGN se compose principalement de quatre couches à savoir :

- **Couche accès :** cette couche est formée par les réseaux d'accès existants fixe (xDSL, PSTN), mobile (GSM, EDGE, etc.) et sans fil (WLAN, etc.) permettant la connexion des utilisateurs au réseau NGN selon la technologie d'accès implémentée.

  Les composants : Terminaux IP, *Session Border Controller* (SBC), *Media Gateway* (MG)

- **Couche transport :** c'est la partie responsable du transfert du trafic média (data ou voix) et des informations de contrôle (signalisation) dans le cœur du réseau IP ; ainsi sa fonction principale est la commutation des paquets IP.

  Cette couche fournit d'une part, la convergence voix/données et d'une autre part la convergence fixe/mobile et ce, en offrant un réseau de transport commun et tout IP à tous les types de trafics à toutes les technologies d'accès.

  Les composants : Les routeurs situés dans le Backbone et les réseaux WAN.

- **Couche contrôle :** elle assure le contrôle des services. C'est à cette couche qu'il incombe d'établir, maintenir et libérer les sessions multimédias dans le réseau cœur IP d'une part et d'une autre part, d'à allouer les réseaux dans le réseau transport ; Sa fonction principale étant le pilotage de la couche Transport.

  Les composants : contrôleur d'appel ou *SoftSwitch*

- **Couche service :** cette couche regroupe des plateformes d'exécution des services et de la diffusion du contenu ; elle offre des services à valeur ajoutée.

  Les composants : serveur d'application, serveur de média ect…..

Partie I :

Etude de la technologie MPLS

1. Introduction

La technologie *Multi Protocol Label Switching* (MPLS) est une architecture plutôt qu'un protocole ; en fait, elle est composée de plusieurs protocoles et mécanismes coopérant tous ensemble dans le but de faciliter le transport des paquets IP entre les différentes entités.

« *Ce serait bien de pouvoir acheminer les paquets sur un chemin déterminé par le réseau sans devoir réévaluer la Forwarding Equivalence Class (FEC) à chaque saut* ». Cet acheminement préconisé peut être, à vrai dire, réalisé grâce à l'établissement d'un itinéraire « logique » sur lequel les paquets seront envoyés et ce en utilisant un identificateur unique par FEC appelé Label; chaque paquet entrant dans un réseau présente dans son entête le FEC. Ce paquet n'est pas obligatoirement un paquet IP ce qui nous renvoie à la notion de Multi Protocoles de MPLS.

2. Principe de fonctionnement

Le réseau MPLS est une technique servant à implémenter en même temps les principes de commutation utilisé en Frame Relay et ATM (niveau 2) et ceux de routage (niveau 3). De ce fait, MPLS est indépendant de la couche 2 du modèle OSI.

MPLS fonctionne en mode connecté (basé sur les labels) du fait que les tables de commutation sont déduites à partir des instructions issues de la « coopération » des protocoles de contrôle avec ceux de routage IP.

Une architecture MPLS contient principalement deux types d'équipements comme le montre la figure 1, à savoir :

- *Label Switch Router* (LSR) : ce sont des routeurs ou commutateurs existant à l'intérieur du domaine MPLS.

- *Label Edge Router* (LER) : c'est un routeur servant d'interface entre le nuage MPLS et le réseau externe. En général, le LER possède des interfaces supportant le protocole MPLS et d'autres supportant un protocole de type IP.

Du fait qu'il est à la frontière, on trouve deux catégories de LER :

- *Ingress LER* : point d'entrée dans le réseau MPLS
- *Egress LER* : point de sortie du le réseau MPLS

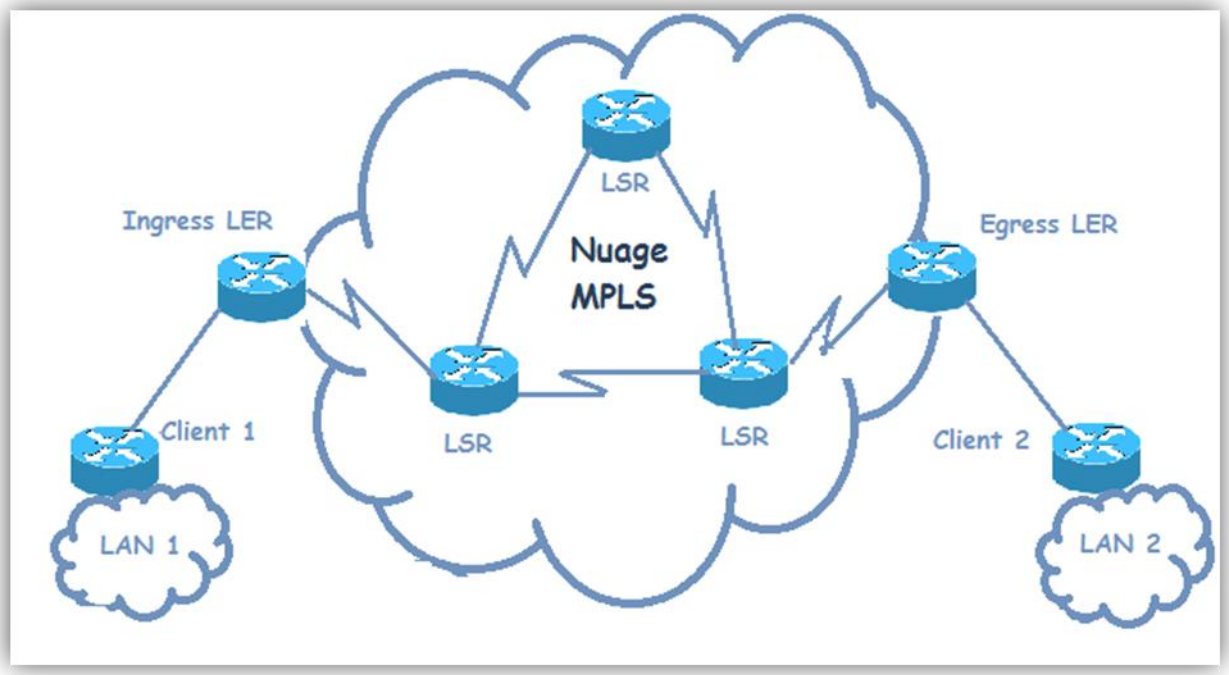


Figure 1 : Architecture MPLS

2.1. Commutation des labels

2.1.1. Principe général

Les réseaux MPLS se reposent sur la notion d'établissement d'itinéraires entre deux équipements appelés *Label Switched Path* (LSP). Les paquets circulant sur ce chemin seront commutés par examen du label de l'entête MPLS ajouté entre la couche 2 (Ethernet généralement) et la couche 3 (IP).

La commutation de paquet en MPLS peut être résumée en, principalement, trois étapes :

- *Etape 1 :* Lorsque le paquet arrive au réseau MPLS, l'Ingress LER examine sa table de commutation, lui affecte un label et le transmet ensuite au LSR suivant.

- *Etape 2 :* Une fois le paquet MPLS arrivé au LSR interne du réseau MPLS, le protocole de routage implémenté sur ce dernier consulte la *Label Base Information* (LIB) et détermine à partir de cette dernière le label suivant à utiliser pour acheminer le paquet vers sa destination. Ainsi, le LSR en question, sur lequel le protocole de routage IP n'est plus utilisé, modifie l'entête MPLS en permutant les labels et mettant à jour les champs TTL et le bit S, et ce avant de passer le paquet au nœud suivant que ce soit un LSR ou un Egress LER.
- *Etape 3 :* Comme dernière étape, lorsque le paquet MPLS arrive à l'Egress LSR, ce dernier élimine « toute trace MPLS » et le transporte vers le monde IP.

Ainsi, la commutation de paquets peut être schématisée comme suit (voir figure 2)

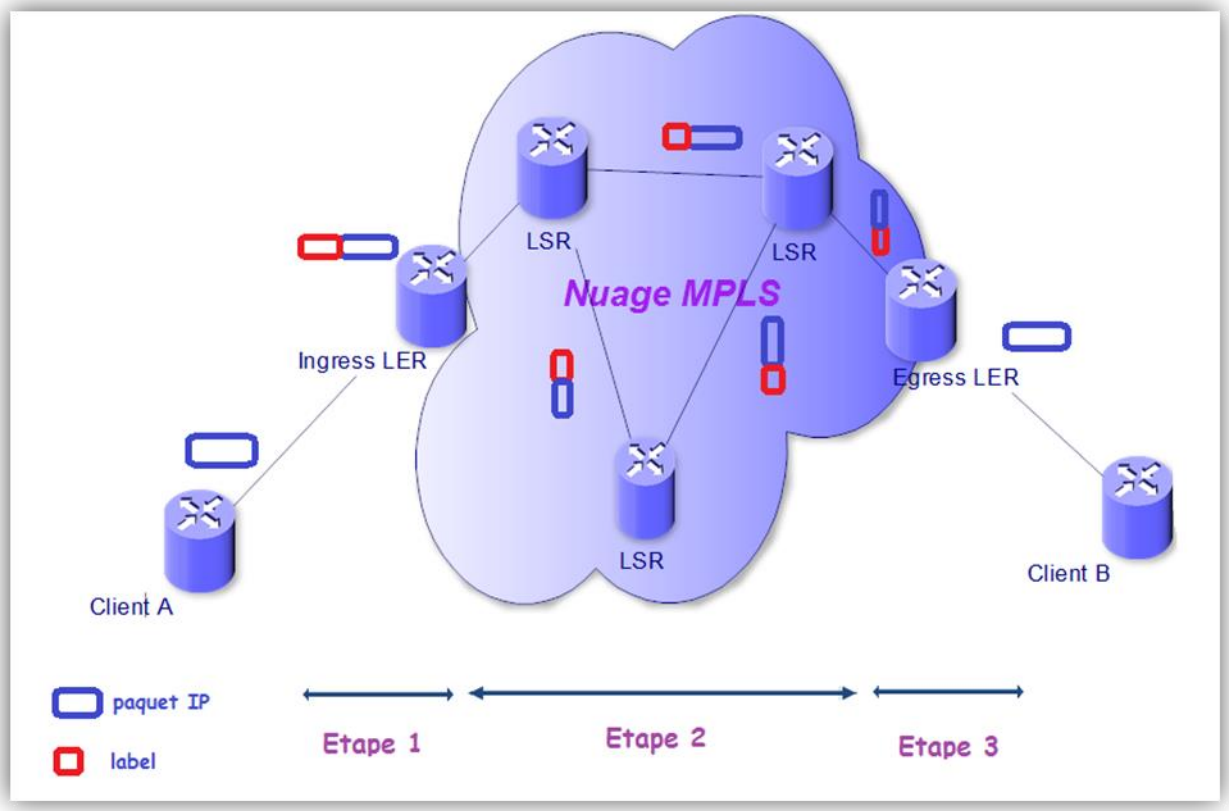


Figure 2 : commutation de paquets dans MPLS

2.1.2. Notion des tables de commutation

L'acheminement des paquets dans MPLS met en œuvre principalement trois tables :

- *Label Information Table (LIB)* : c'est la table primordiale à construire par le routeur MPLS. Elle contiendra la liste des labels affectés par les LSR voisins pour chaque sous-réseau. Ainsi, grâce à cette table on peut connaitre tous les chemins possibles pour atteindre une destination.
- *Label Forwarding Information Base (LFIB) :* c'est une table déduite à partir de la LIB et de la table de routage IP. Elle est élaborée par le routeur qui y met les labels du meilleur prochain saut (Next Hop) utilisé pour l'acheminement des paquets étiquetés.
- *Forwarding Information Base* (FIB) : c'est une base de données permettant l'acheminement des paquets ne portant pas de labels.

Pour aboutir aux différentes tables, le routeur LSR doit passer, obligatoirement, par les étapes suivantes :

- Etablir les tables de routage par un protocole de routage.
- Assigner un label à chaque destination existante dans la table de routage.
- Enregistrer dans la LIB des labels avec signification locale.
- Modifier la LFIB en y mettant les labels précédemment alloués en correspondance avec leur prochain saut tout en mentionnant l'action.
- Envoyer les informations existant sur la LIB à ses voisins

Après avoir entamé l'envoi, chaque LSR ayant reçu ces informations les enregistre dans sa LIB et les prochains sauts dans sa FIB.

Ainsi, on peut schématiser la procédure par l'exemple de la figure 3.

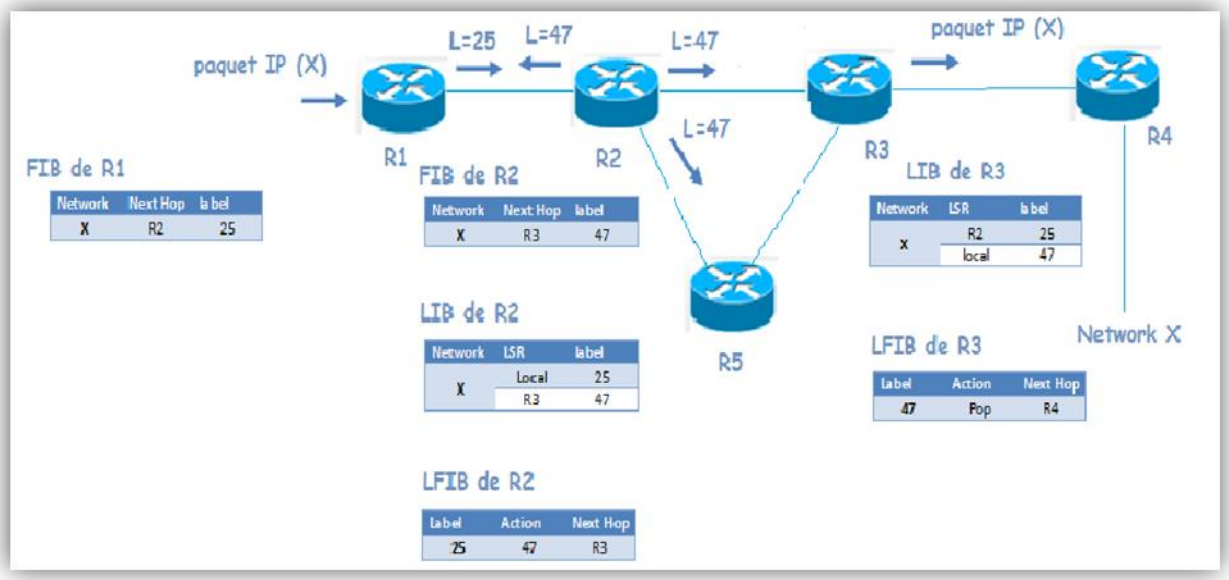


Figure 3 : Exemple construction table de commutation

2.2. La notion des labels

Il est crucial d'examiner la pile protocolaire mise en jeu et l'élément de base étudié, le label. En fait, un label est un identifiant unique d'une FEC ayant une signification locale entre 2 LSR adjacents. Il sert de moyen de traçage de flux entre un LSR et son successeur.

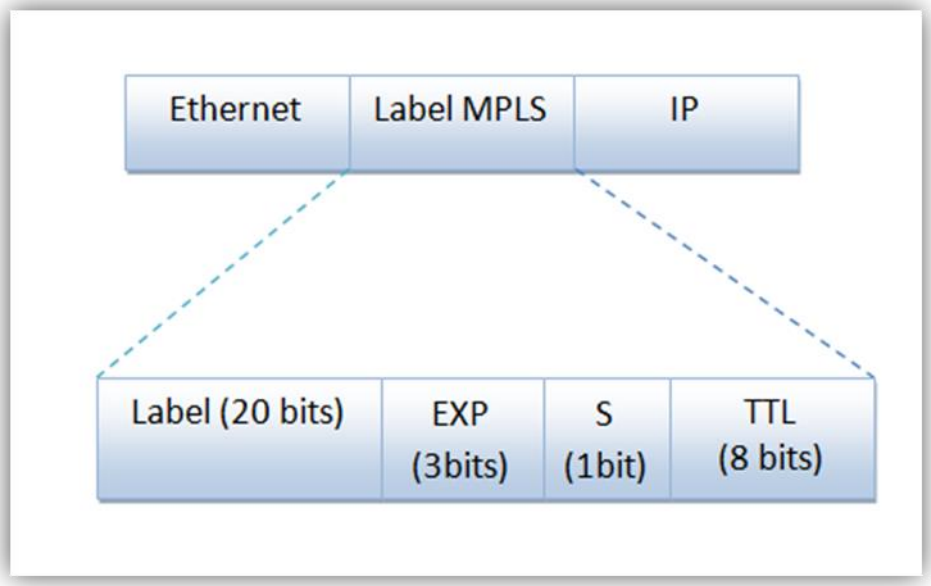


Figure 4: label MPLS

Comme est indiqué dans la figure 4, le label MPLS contient principalement quatre champs :

- Label (20 bits) : qui contient le label.
- EXP (3 bits) : c'est un champ indiquant la classe du trafic ; il est utilisé généralement comme indicateur de QoS.
- S (1 bit): bottom of stack indicator. MPLS offre plusieurs labels à empiler, ce champ indique si ce label est le dernier à insérer dans le paquet ou non ; le bit S est à 1 si ce label est au sommet de la pile de labels.
- TTL (8bits): Time To Live. C'est un champ indiquant la durée de vie du paquet ; autrement dit, il renseigne sur le nombre de routeurs que le paquet peut franchir. Ce champ permet l'évitement des boucles de routage. A vrai dire, dans le cas courant, le TTL est une copie du TTL de l'entête IP et ce pour que les routeurs du cœur MPLS ne se voient pas obligés d'aller consulter le TTL du paquet IP.

2.3. La notion de FEC

Pour mettre en œuvre MPLS, il est crucial de se baser sur des caractéristiques en commun entre un ensemble de paquets, desquelles dépendra l'acheminement de ces derniers ; c'est la notion de la FEC.

Les paquets pénétrant dans le réseau MPLS sont associés à une FEC qui va définir la manière avec laquelle ils seront transportés ; cette FEC prend en compte différents critères à l'instar de l'adresse destination, adresse source, application, QoS (débit, délai…).

Les paramètres intervenant dans la classification d'un paquet dans une FEC dépendent du protocole de distribution de label utilisé : *Label Distribution Protocol* (LDP) ou RSVP-TE. En revanche, seul RSVP-TE, permet de classifier un paquet dans une FEC selon des paramètres de QoS.

Pour assigner une FEC à un paquet, MPLS se base sur le protocole de routage utilisé dans le réseau IP. LDP, par exemple, donne une FEC par préfixe réseau figurant dans la table de routage du routeur. Ainsi, on peut remarquer des correspondances FEC/label appelées en terminologie MPLS des Bindings ; le routeur saura donc le label, à assigner au paquet, correspondant à la FEC.

3. Fonctionnalités apportées par MPLS

De nos jours, MPLS n'est plus considéré comme une simple technologie permettant la réduction du temps de traitement de paquets, mais plutôt un moyen servant à multiples applications qui étaient presque impossible à mettre en place avec l'IP traditionnel.

Parmi les fonctionnalités majeures que MPLS offre, on trouve le support de la virtualisation, l'AToM et la gestion de la QoS.

3.1. Support de la QoS

La technologie MPLS permet une meilleure gestion de la QoS ; cette dernière est mise en œuvre de trois manières à savoir l'ordonnancement du trafic, le marquage des paquets et le *Traffic Engineering*.

3.1.1. Ordonnancement du trafic

MPLS permet la mise en attente des flux circulés sur différentes files d'attente ; les trafics d'un même *Label Switched Path* (LSP) seront éparpillés sur différentes files de plusieurs LSR et ce selon la valeur du champ EXP de l'entête MPLS.

3.1.2. Marquage des paquets

Le marquage des paquets est une notion de QoS permettant la différenciation du trafic et ce en implémentant les classes de services : chaque routeur périphérique (LER) applique sur les paquets une classification et un marquage, par insertion d'un champ *Differentiated Services Code Point* (DSCP) dans l'entête IP, qui seront utilisés par les nœuds centraux

(LSR). En fait, comme est indiqué dans la figure 5, l'entête IPv4 contient un champ appelé *Type Of Service* (ToS) s'étendant sur un octet (8bits). Selon la RFC-3168, le ToS se compose principalement de deux parties comme suit :

- **DSCP :** une valeur numérique occupant les six premiers bits. Ce champ est subdivisé en deux sous champs :
  - Class Selector Code Point (3 bits) : permet la classification du paquet en spécifiant la classe à laquelle il appartiendra.
  - Drop Precedence (3 bits) : définit la priorité avec laquelle les paquets seront rejetés.

Le champ DSCP donne 64 *codePoints* permettant, chacun, de définir une classe de service *Per-Hop Behavior* (PHB). En effet, il existe trois PHB **:**

- *Assured Forwarding* (AF) : on utilise cette PHB pour les trafics à bande passante limitée ; le flux excédent sera rejeté, se basant sur un mécanisme de priorité et ce, compte tenu de quatre classes de trois priorités de rejet chacune.
- *Expedited Forwarding* (EF) : cette classe correspond aux services premium nécessitant une bande passante garantie et ne tolérant que des faibles taux de pertes de paquets ; le DSCP correspondant à cette classe est 101110.
- *Class Selector* (CS) : selon la RFC-791 et 1349, on trouve un champ IP Precedence sur trois bits. Alors, pour assurer la compatibilité entre ce dernier et le champ DSCP, on définit huit CS allant de CS0 à CS7.

- **Explicit Congestion Notification (ECN) :** une valeur numérique sur 2 bits permettant la prévention de la congestion du réseau avant que la perte des paquets ne commence. Les protocoles de transport à l'instar de TCP tendent, en cas de fonctionnement normal, à accroître le débit de transport du flux qui n'est diminué qu'en cas de perte de paquets. A l'aide d'ECN, le routeur peut mentionner le début de congestion ce qui permet d'éviter la perte de paquets.

  Vu qu'il s'étend sur 2 bits, le champ ECN peut avoir les valeurs suivantes :
  - 00 : Transport incapable de gérer l'ECN (non-ECT « non-ECN Capable Transport »)
  - 10 : Transport capable de gérer l'ECN (ECT(0))
  - 01 : Transport capable de gérer l'ECN (ECT (1))
  - 11 : Congestion existante : *Congestion Experienced (CE).*

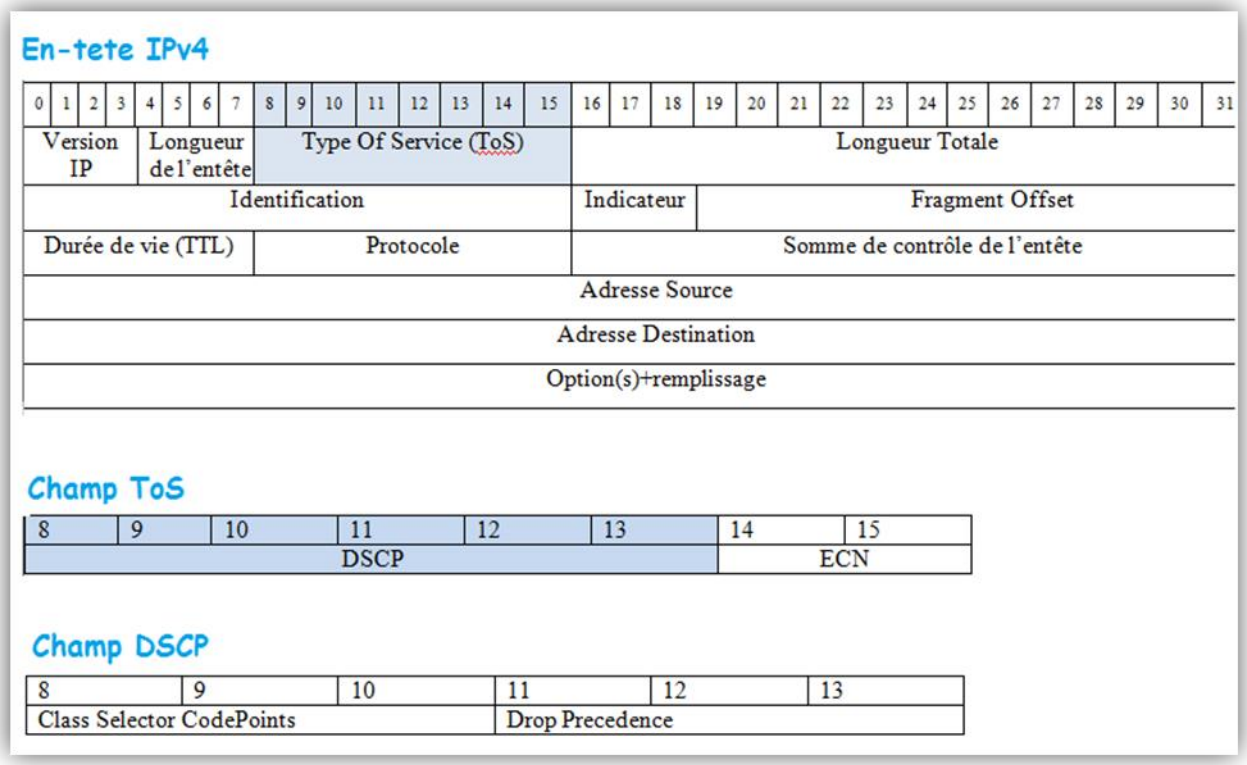


Figure 5: champ DSCP

Le marquage DSCP est un marquage de niveau 3. Ce dernier suit toujours un marquage de niveau de champ EXP de l'entête MPLS. Pour cette raison, une table de mapping EXP<=>DSCP doit exister. Il n'y a pas de normalisation dans cette correspondance ; tout simplement, il faut faire un choix d'association EXP<=>DSCP cohérent sur tous les Gateway de Backbone IP/MPLS. En revanche, il faut noter, quand même, qu'il existe quelques valeurs par défaut de mapping EXP<=>DSCP comme le montre le tableau ci-desous.

Tableau : mapping EXP/DSCP

EXP	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
DSCP	0	8	16	24	32	40	48	56

3.1.3. Trafic Engineering

Le trafic engineering est un concept en relation étroite avec la mise en œuvre de la QoS dans un réseau MPLS, ayant comme résultat direct l'amélioration et l'optimisation des

ressources radio à l'instar de la gigue, bande passante permettant ainsi l'évitement de la congestion.

MPLS TE est vu comme une solution visant l'augmentation des performances d'un réseau tout en jouant sur la répartition équitable des trafics dans un réseau ; c'est pour cette raison qu'il est considéré comme étant une « application à part entière » par la plupart des industriels et non pas une simple technique de réservation de ressources.

3.2. AToM

L'*Any Transport Over MPLS* est un service facilitant le transport du trafic de la couche 2 à travers un nuage MPLS ; c'est une application qui met en relief l'indépendance de la technologie MPLS des protocoles implémentés dans les couches 2 et 3.

3.3. Le support de la virtualisation

3.3.1. Les réseaux privés virtuels

L'une des applications les plus importantes que fournit MPLS est la création des *Virtual Private Network* (VPN). « Un VPN est un ensemble de sites d'un client qui sont interconnectés ensemble à partir d'une infrastructure réseau partagée et qui n'ont pas conscience de la présence d'éventuels autres sites connectés eux aussi sur cette infrastructure.

Pour l'isolation des trafics les uns des autres, MPLS met à disposition deux labels : un label extérieur identifiant le chemin vers le LSR destination changeable à chaque pas et un label intérieur spécifiant l'identificateur du VPN et non modifiable tout au long du LSP.

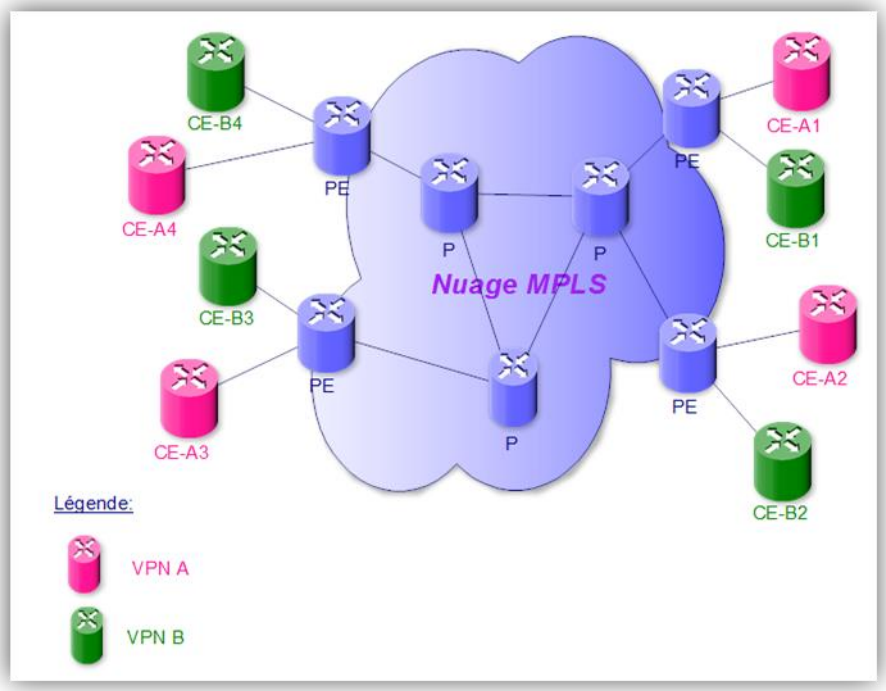


Figure 6: composantes des VPN/MPLS

Comme le montre la figure 6, un VPN/MPLS est constitué principalement de trois types d'équipements :

- *Customer Edge* (CE) : c'est le routeur client responsable du routage IP entre le site client et le Backbone c'est pourquoi il est connecté à ce dernier par une Ligne Spécialisée, nuage Frame Relay, etc.
- *Provider Edge* (PE) ou encore LER : c'est le routeur qui est directement relié au CE ; de ce fait il est situé sur le bord du nuage MPLS pour assigner un CE à un VPN bien particulier. Il a principalement deux rôles à savoir la gestion des VPN en collaboration avec les autres PE et la commutation du trafic vers les P.
- *Provider* (P) ou encore LSR : c'est un routeur ou commutateur existant au cœur du nuage MPLS ayant pour fonction l'acheminement du trafic.

3.3.2. Les tables de routage virtuelles

3.3.2.1. Principe de fonctionnement

La mise en place des VPN impose la création de « barrière » entre les flux au niveau des sites clients n'appartenant pas au même VPN. Pour ce faire, le *Provider Edge* assigne statiquement une *Virtual Routing and Forwarding table* (VRF) à chaque partie utilisateur existante ce qui lui permet de manipuler plusieurs tables de routage à la fois. Cette gestion

« distribuée » permet au PE d'administrer un plan d'adressage par sites même en cas de recouvrement d'adresse entre VPN différents *(overlapping).*

Une VRF est constituée d'une table de routage, une FIB et une table CEF spécifique. Elle est désignée par un nom significatif localement sur chaque PE ; ainsi, chaque interface du PE raccordée avec un site client est reliée à une VRF bien déterminée. Donc, en recevant un paquet IP sur une interface client, le PE examine la VRF à laquelle est associée l'interface en question sans consulter sa table de routage globale.

Pour qu'ils puissent construire leurs tables VRF, les *Provider Edge* échangent entre eux les routes des différents VPN mis en œuvre ; ainsi, l'acheminement des paquets destinés à un PE-1 attaché à un site CE-1 impose la connaissance des routes VPN de PE-1 par PE-2. En effet, cet échange est réalisé par le protocole MP-BGP ; Les configurations des VRF ne comportant que des paramètres relatifs à MP-BGP (notamment pour l'export et l'import des routes).

La table CEF permet de déterminer le prochain saut, l'interface de sortie et les labels utilisés pour atteindre un sous réseau particulier.

3.3.2.2. Famille d'adresses VPN-IPv4

a) Structure des adresses VPN-IPv4

L'utilisation du protocole MP-BGP rend BGP « compatible » avec différentes familles d'adresses à l'instar de VPN-IPv4, famille créée pour les VPN BGP MPLS.

Une adresse de ce type s'étend sur 96 bits, à savoir 12 octets. Cette adresse commence par un identifiant de route unique, le *Route Distinguisher* (RD) sur 8 octets et se termine par une adresse IPv4 sur 4 octets.

Structure adresse VPN-IPv4

Adresse VPN-IPv4 (12 octets)	
Route Distinguisher (8 octets)	Adresse IPv4 (4 octets)

b) Structure des RD

Le RD est un champ administrable par l'opérateur. Il permet de garantir l'unicité des routes VPN-v4 échangées entre les PE. Pour qu'il soit facilement administré, le champ RD est structuré comme le montre ci-dessous.

Structure des RD

Route Distinguisher (8 octets)		
Type (2 octets)	Admin (X octets)	N° assigné (Y octets)

La structure du RD dépend du type utilisé ; ainsi, il existe principalement trois types à savoir type 0, 1 et 2.

c) Redistribution des routes

Les RD ne permettent pas l'identification de la manière avec laquelle les routes sont insérées dans les VRF des routeurs PE ; l'importation et l'exportation des routes sont gérées grâce à une communauté étendue BGP (*extended community*) associée à une VRF appelée *Route Target* (RT). Cet attribut, indiquant l'adhésion d'une route à un VPN, est véhiculé par BGP comme étant l'attribut de la route, ce qui lui permet d'être placée dans les VRF qui sont utilisés pour router le trafic reçu de ce site.

Puisque les routes peuvent être soit envoyées soit reçues par un routeur, on définit deux « méthodes » pour la RT à savoir :

- Import Target : le routeur PE assigne un RT à une route reçue d'un site
- Export Target : le routeur PE utilise le RT pour vérifier que la route reçue d'un autre PE est valide et l'ajoute aux VRF.

3.3.2.3. Avantages

Les VRF sont utilisés pour partitionner l'infrastructure réseau et faciliter la gestion de ce dernier. Ses avantages sont les suivants:

- Fournir des accès sécurisés
- Permettre aux différentes entités d'utiliser des adresses IP en *overlapping* ce qui n'est pas supporté par le routage IP global.

▪

Partie 2 :

Etude de cas

1. Introduction

Dans cette partie, nous allons nous intéresser à la partie routage au niveau du Backbone IP/MPLS ; nous nous sommes dans la couche transport du modèle NGN.
Ainsi, on argumentera le choix du simulateur choisi et des protocoles implémentés lors de la simulation.

2. Environnement du travail

Actuellement, il existe plusieurs logiciels de simulation des réseaux informatique dont les fonctionnalités diffèrent. Pour notre cas, on a cherché un simulateur supportant MPLS et ses dérivés (VPN et VRF) ; ainsi, notre choix s'est focalisé sur GNS3 pour plusieurs raisons. En effet, GNS3 est un logiciel open Source permettant l'émulation des équipements Cisco à savoir les routeurs quelle que soit leur gamme, les firewalls, les switchs (Layer 2 et Layer 3), etc. En outre, ce simulateur permet la construction des architectures comme est le cas dans la réalité exactement. Enfin, il possède une fonctionnalité très importante à savoir d'une part le support de la connexion aux réseaux réels et d'autre part la possibilité d'intégration de machines virtuelles dans le réseau à simuler.
L'une des particularités de GNS 3 est qu'il charge de véritables images IOS des routeurs Cisco à simuler. Ainsi, et pour permettre des simulations des réseaux compliqués, il est crucial qu'il soit lié à ces deux logiciels :

- *Dynamips :* il sert à l'émulation d'une machine virtuelle Cisco, supporté sous Linux et windows.
- *Dynagen :* une interface supplémentaire écrite en Python permettant la gestion des interconnexions entre les différentes machines virtuelles.

3. Maquette à réaliser

Dans notre cas, nous allons limiter notre étude sur 2 P (P1 et P2), 2 PE (PE1 et PE2) et 4 CE (CE11, CE12, CE21 et CE22) mis à part la simulation, partie décisive dans notre architecture.

CEi,j : (i : représente le Client et j : représente le numéro de site)

Par exemple CE12 : Client 1 et Site 2

3.1.Présentation de la maquette

Pour mener à bien notre simulation, nous nous intéressons à l'architecture présentée par la figure ci-dessous :

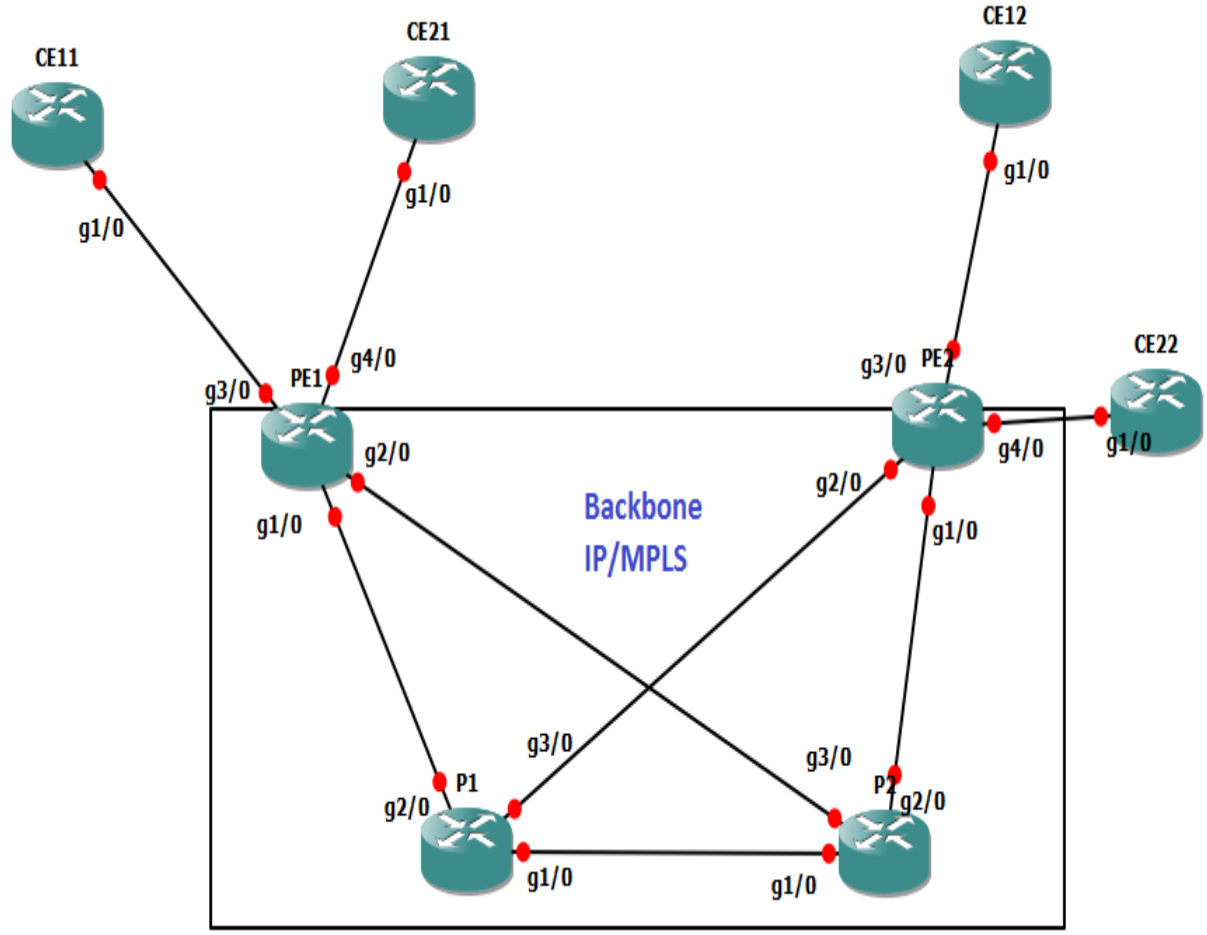


Figure 7: Maquette de Simulation

Comme est montré par la figure 7, l'architecture du Backbone repose sur la notion de redondance permettant d'être à l'abri de tout dysfonctionnement d'un provider ; chaque PE est doublement raccordé sur deux providers.

3.2. Plan d'adressage

Pour envisager le plan d'adressage, nous devons nous intéresser principalement à quatre éléments importants à savoir :

- *Backbone :* Le plage d'adresses 10.1.1.0/24. Ainsi, nous utiliserons pour les sous réseaux du backbone des adresses appartenant à la dite plage. Donc la distribution se fera comme suit :
  - PE1 – P1 : 10.1.1.0/30
  - PE1 – P2 : 10.1.1.4/30
  - PE2 – P2 : 10.1.1.8/30
  - PE2 – P1 : 10.1.1.12/30
  - P1 – P2 : 10.1.1.20/30
  - Loopback0 de PE1 : 1.1.1.1/32
  - Loopback0 de PE2: 2.2.2.2/32
  - Loopback0 de P1 : 3.3.3.3/32
  - Loopback0 de P2 : 4.4.4.4/32

- *Coté Client :* nous utiliserons la plage d'adresse 172.16.0.0/16 pour les LAN et la plage d'adresse 192.168.1.0/24 pour le WAN. La distribution d'adresses sera comme suit :

  - CE11 –PE1 : 192.168.1.0/30
  - CE21 –PE1 : 192.168.1.4/30
  - CE12 –PE2 : 192.168.1.8/30
  - CE22 –PE2 : 192.168.1.12/30
  - Loopback0 de CE11 : 172.16.11.11/32
  - Loopback0 de CE12 : 172.16.12.12/32
  - Loopback0 de CE21 : 172.16.21.21/32
  - Loopback0 de CE22 : 172.16.22.22/32

Le tableau ci-dessous récapitulera l'adressage des différentes interfaces des routeurs implémentés :

Tableau : Adressage de la maquette

	Interface	**Adresse IP**
CE11	G1/0 Connect to PE1	192.168.1.2/30
	Loopback0	172.16.11.11/32
CE12	G1/0 Connect to PE2	192.168.1.10/30
	Loopback0	172.16.12.12/32
CE21	G1/0 Connect to PE1	12.168.1.6/30
	Loopback0	172.16.21.21/32
CE22	G1/0 Connect to PE2	192.168.1.14/30
	Loopback0	172.16.22.22/32
PE1	Loopback0	1.1.1.1/32
	G1/0 Connect to P1	10.1.1.1/30
	G2/0 Connect to P2	10.1.1.5/30
	G3/0 Connect to CE11	192.168.1.1/30
	G4/0 Connect to CE21	192.168.1.5/30
PE2	Loopback0	2.2.2.2/32

	G1/0 Connect to P2	10.1.1.9/30
	G2/0 Connect to P1	10.1.1.13/30
	G3/0 Connect to CE12	192.168.1.9/30
	G4/0 Connect to CE22	192.168.1.13/30
P1	Loopback0	3.3.3.3/32
	G1/0 connect to P2	10.1.1.21/30
	G2/0 connect to PE1	10.1.1.2/30
	G3/0 connect to PE2	10.1.1.14/30
P2	Loopback0	4.4.4.4/32
	G1/0 connect to P1	10.1.1.22/30
	G2/0 connect to PE2	10.1.1.10/30
	G3/0 connect to PE1	10.1.1.6/30

3.3. Configuration de base de l'adressage de chaque Interface

Dans une première étape, nous devons configurer les différentes interfaces des routeurs à utiliser. Ci-dessous, un exemple de configuration de quelques interfaces du routeur PE1 est illustré. **(Les détails voir le Vidéo)**

```
PE1# conf t

PE1(config)# interface Loopback 0

PE1(config-if)#ip address 1.1.1.1 255.255.255.255

PE1(config-if)#interface g1/0

PE1 (config-if)#ip address 10.1.1.1 255.255.255.252

PE1(config-if)#no shutdown
```

4. Configuration des protocoles de routage

4.1.Routage OSPF de Backbone IP/MPLS

Au niveau du backbone, nous avons choisi d'implémenter OSPF comme protocole de routage dynamique et ce, pour plusieurs raisons à savoir :

- Convergence rapide.
- Réduction des mises à jour et ce grâce à l'utilisation des areas et aux mises à jour incrémentales.
- Réduction de la taille des tables de routage par segmentation en areas et implémentation des résumés de routes.
- Support du Trafic Engineering (OSPF-TE).

Pour l'implémentation d'OSPF, on a segmenté l'architecture globale en différentes zones (areas) comme suit :

- Area 0 : c'est la zone backbone ; elle contient les routeurs providers « P » et les routeurs Provider Edge « PE ».
- Area ij : chaque site « ij » aura une area « ij» contenant le WAN et LAN (dans notre cas Loopback 0 de CEij) du site.

Ainsi, l'architecture OSPF sera illustrée comme suit :

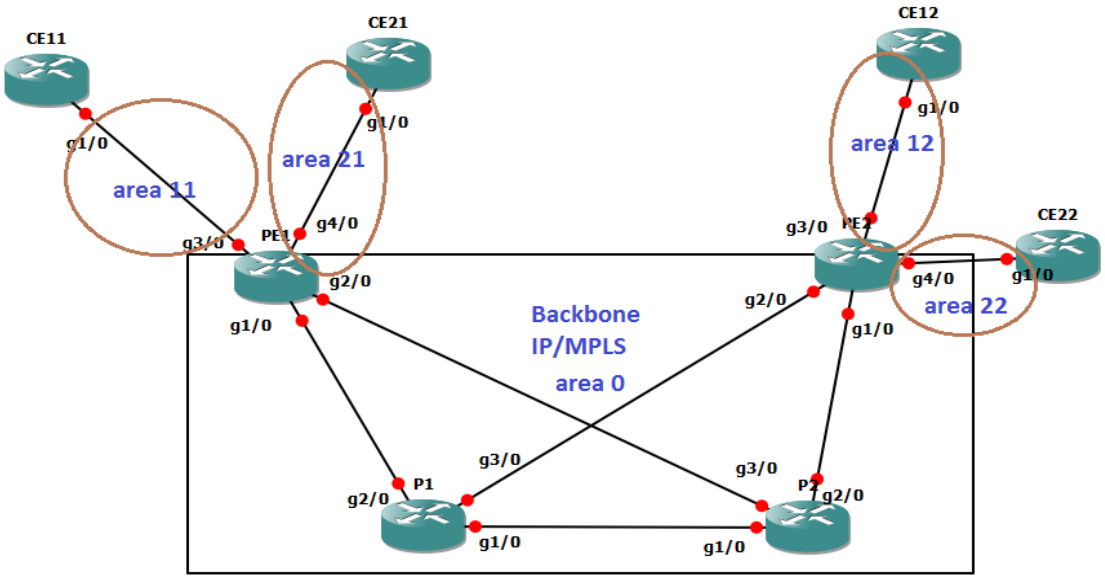


Figure 8: architecture OSPF

Nous appliquerons le protocole OSPF sur tous les routeurs du Backbone IP/MPLS tout en prenant en considération Area 0.

Nous donnerons un exemple de configuration ci-dessous le routeur PE1 **(les détails de configuration voir le Vidéo)**.

```
PE1(config)# router ospf 1

PE1(config-router)# network 10.1.1.0 0.0.0.3 area 0

PE1 (config-router)# network 10.1.1.4 0.0.0.3 area 0

PE1(config-router)# network 1.1.1.1  0.0.0.0 area 0
```

Afin de vérifier le bon fonctionnement du routage OSPF, nous tapons la commande ci-dessous :

1- **show ip ospf neighbor :** Affichage la table de Voisinage OSPF

2- **show ip route ospf** : Affichage la table de routage OSPF

4.2. Configuration de MPLS

La technologie MPLS fonctionne par commutation des labels. Ainsi, il est obligatoire d'activer le protocole MPLS sur les routeurs du Backbone tout en prenant en considération les paramètres exigés. Pour ce faire, nous procédons comme il est noté ci-dessous (Exemple de configuration PE1) en tapant les dites commandes sur toutes les routeurs appartenant au Backbone MPLS/IP **(les détails de configuration voir le Vidéo)**.

```
PE1# conf t

PE1(config)# ip cef

PE1(config)# mpls ip

PE1(config-if)# mpls label protocol ldp

PE1(config-if)# mpls ldp router-id loopback 0 force

PE1(config)#interface g 1/0

PE1(config-if)# mpls ip

PE1(config-if)# interface g 2/0

PE1(config-if)# mpls ip
```

Afin de vérifier le bon fonctionnement de la commutation des paquets au sein du Backbone, nous pouvons utiliser les commandes suivantes :

- **Show mpls ldp neighbors :** affichage des voisins
- **Show mpls forwarding table:** vérification de la LFIB
- **Show mpls ip binding:** affichage des bindings des labels MPLS récupérés par LDP.

5. Mise en place des VPN (Tekup)

5.1. Configuration de VRF

Afin d'isoler les trafics et implémenter la fonctionnalité de virtualisation de MPLS, nous implémentons les VRF sur nos routeurs PE. A cet égard, nous commençons dans cette partie par la création deux VRF : VPN_Customer1 et VPN_Customer2.

Ainsi, sur tous les providers Edge (PE1 et PE2) du Backbone, on crée les deux VRF VPN_Customer1 et VPN_Customer2. La configuration du VRF se fera en deux étapes à savoir : **(les détails de configuration voir le Vidéo)**.

- **Création du VRF**

```
PE1(config)# ip vrf    VPN_Customer1

PE1(config-vrf)#rd 100 :1

PE1(config-vrf)# route-target both 100:1

PE1(config)# ip vrf    VPN_Customer2

PE1(config-vrf)#rd 100 :2

PE1(config-vrf)# route-target both 100:2
```

```
PE1(config)# ip vrf    VPN_TEKUP

PE1(config-vrf)#rd 100 :3

PE1(config-vrf)# route-target both 100:3
```

- **Activation du VRF sur les interfaces**

Sur les deux PE sur lesquels nous avons attaché des sites clients, nous configurons le VRF précédemment créé et ce sur l'interface raccordée avec le client. Ainsi, nous donnons dans ci-dessous la configuration du routeur PE1 **(les détails de configuration voir le Vidéo)**.

```
PE1(config)#interface g3/0
PE1(config-if)#ip vrf forwarding VPN_TEKUP
PE1(config-if)#ip address 192.168.1.1  255.255.255.252
PE1(config-if)#no shutdown
PE1(config)#interface g4/0
PE1(config-if)#ip vrf forwarding VPN_TEKUP
PE1(config-if)#ip address 192.168.1.5  255.255.255.252
PE1(config-if)#no shutdown
```

NB : Idem pour PE2 (juste @IP)

5.2 Configuration les Interfaces et OSPF au niveau CE.

Nous appliquerons le protocole OSPF sur tous les routeurs du CEij tout en prenant en considération Area ij.

Nous donnerons un exemple de configuration ci-dessous le routeur CE11 **(les détails de configuration voir le Vidéo)**.

```
CE11# conf t
CE11(config)# interface Loopback 0
CE11(config-if)#ip address 172.16.11.11 255.255.255.255
CE11(config-if)#interface g1/0
CE11 (config-if)#ip address 192.168.1.2  255.255.255.252
CE11(config-if)#no shutdown
```

```
CE11(config)# router ospf 1

CE11(config-router)# network 192.168.1.0 0.0.0.3 area 11

CE11(config-router)# network 172.16.11.11  0.0.0.0 area 11
```

5.3 Configuration de MP_BGP

Pour que le VRF fonctionne, nous distribuons le chemin vers tout le réseau. Les commandes ci-dessous seront exécutées sur PE1 ayant les interfaces sur laquelle est attaché le VRF. (**(Les détails de configuration voir le Vidéo)**.

```
PE1#conf t

PE1(config)# router bgp 100

PE1(config-router)#no bgp default ipv4-unicast

PE1(config-router)# neighbor 2.2.2.2  remote-as 100

PE1(config-router)# neighbor 2.2.2.2  update-source loopback 0

PE1(config-router)# address-family vpnv4 unicast

PE1(config-router-af)# neighbor 2.2.2.2 activate

PE1(config-router-af)# neighbor 2.2.2.2 send community both

PE1(config-router-af)#address-family ipv4 vrf VPN_Customer1

PE1(config-router-af)#redistribute ospf 100 vrf VPN_Customer1

PE1(config-router-af)#address-family ipv4 vrf VPN_Customer2

PE1(config-router-af)#redistribute ospf 200 vrf VPN_Customer2

PE1(config-router-af)#address-family ipv4 vrf VPN_TEKUP
```

```
PE1(config-router-af)#redistribute ospf  300 vrf VPN_TEKUP

PE1(config-router-af)#exit

PE1(config-router)#exit

PE1(config)# no router ospf 100 vrf  VPN_Customer1

PE1(config)#no router ospf 200 vrf  VPN_Customer2

PE1(config)# router ospf 300 vrf  VPN_TEKUP

PE1(config-router)# redistribute bgp 100 subnets

PE1(config-router)# network 192.168.1.4 0.0.0.3 area 21

PE1(config-router)# network 192.168.1.0  0.0.0.3 area 11
```

```
PE2#conf t

PE2(config)# router bgp 100

PE2(config-router)#no bgp default ipv4-unicast

PE2(config-router)# neighbor 1.1.1.1  remote-as 100

PE2(config-router)# neighbor 1.1.1.1  update-source loopback 0

PE2(config-router)# address-family vpnv4 unicast

PE2(config-router-af)# neighbor 1.1.1.1  activate

PE2(config-router-af)# neighbor 1.1.1.1  send community both

PE2(config-router-af)#address-family ipv4 vrf VPN_Customer1

PE2(config-router-af)#redistribute ospf 100 vrf VPN_Customer1

PE2(config-router-af)#address-family ipv4 vrf VPN_Customer2

PE2(config-router-af)#redistribute ospf 200 vrf VPN_Customer2
```

```
PE2(config-router-af)#address-family ipv4 vrf VPN_TEKUP

PE2(config-router-af)#redistribute ospf  300 vrf VPN_TEKUP

PE2(config-router-af)#exit

PE2(config-router)#exit

PE2(config)# no router ospf 100 vrf  VPN_Customer1

PE2(config)#no router ospf 200 vrf  VPN_Customer2

PE2(config)# router ospf 300 vrf  VPN_TEKUP

PE2(config-router)# redistribute bgp 100 subnets

PE2(config-router)# network 192.168.1.8  0.0.0.3 area 12

PE2(config-router)# network 192.168.1.12  0.0.0.3 area 22
```

Pour vérifier le fonctionnement du VPN, nous tapons les commandes suivantes au niveau PE1 et PE2 :

1- **Show ip bgp vpnv4 vrf VPN_Customer1**: Affichage Instance BGP
2- **Show ip bgp vpnv4 vrf VPN_Customer2:** Affichage Instance BGP
3- **Show ip route vrf VPN_Customer1:** Affichage la table de routage de VRF de VPN_Customer1
4- **Show ip route vrf VPN_Customer2:** Affichage la table de routage de VRF de VPN_Customer2

Pour vérifier la connexion entres les différents sites de clients, nous tapons les commandes suivantes au niveau CEij :

1- **Show ip route** : Affichage Table de routage
2- **Ping @IP: (example:** CE11#ping 172.16.12.12)