REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

Département d'informatique

MEMOIRE

DE FIN D'ETUDE

En vue de l'obtention du diplôme de Master Académique en Informatique.

Option : Réseaux Mobilité et Systèmes Embarqués.

Titre du mémoire :

Implémentation de MPLS sur le backbone d'un fournisseur de service

Encadré par **:**

Mr. DJAMAH.B
Mr. Ben Chaabane.M

par **:**

ZIDELMAL Samy
TOUMI Mohamed

Promotion 2017-2018

Préface

MultiProtocol Label Switching (MPLS) s'est imposé depuis les années 90s comme une technologie fondamentale dans les grands réseaux de données. Etant donnée la vitesse à laquelle la demande en réseaux de données a évolué et continue à évoluer, ainsi que la variété de types de services qui leur en sont requis. Ces réseaux sont des ressources critiques pour bon nombre d'entreprises. La disponibilité doit être extrêmement élevé, les temps de convergence en cas de panne doivent être infiniment petits. Il se trouve que MPLS répond assez bien à ces exigences.

Les fournisseurs de services qui offrent comme service le transport de données, appelés « data carrier », ont été les premiers à adopter MPLS et ce fut rapidement un très grand succès, les entreprises n'ont pas tardé par la suite à utiliser MPLS. Le succès de MPLS réside sans doute dans le fait qu'il permet de transporter à travers le réseau tout type de données existantes. Ainsi MPLS a permis de consolider les différents réseaux pour chaque type de données en un seul grand réseau.

MPLS a beaucoup évolué depuis ses débuts, des applications basées sur MPLS ont vu le jour tel que L2VPN, L3VPN, Pseudowires (PW), Traffic Engineering (TE), la Qualité de Service (QoS).

Sommaire

I. Les méthodes d'acheminement des données réseau

Le modèle OSI divise les fonctions de communication réseau en sept couches. L'information ou la donnée change de nom selon les encapsulations qu'elle possède.

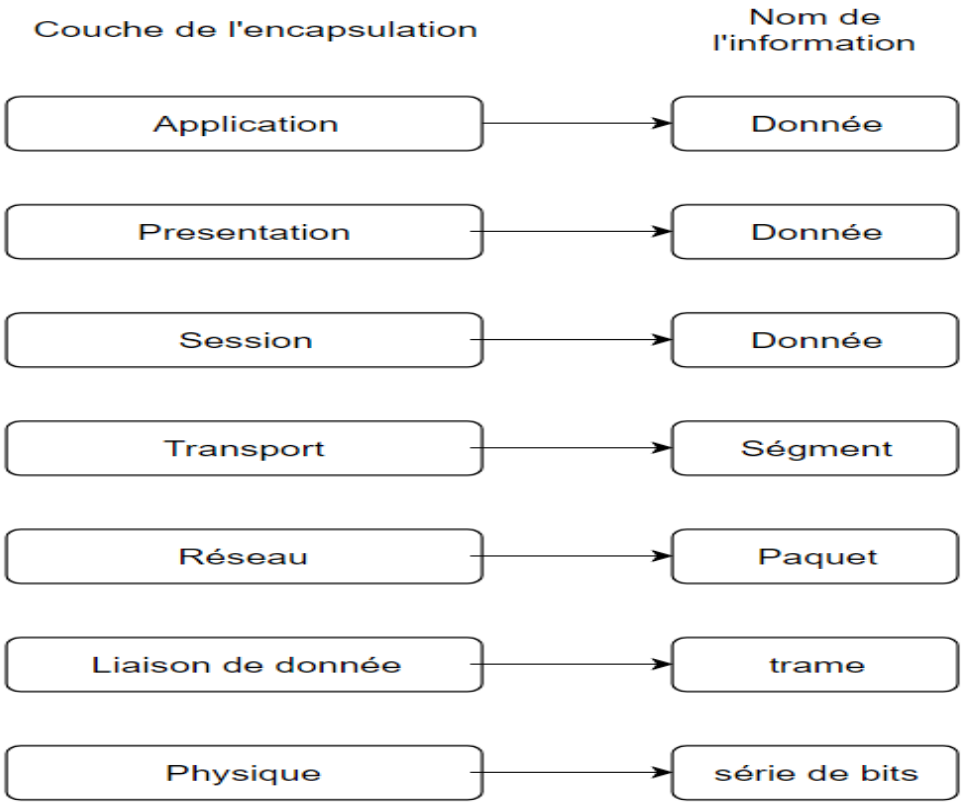

Figure 1-1 noms de l'information selon son encapsulation.

La méthode d'acheminement des données réseau diffère aussi selon l'encapsulation de la donnée. Pour les trames il s'agit de commutation, il existe plusieurs protocoles de couche liaison de donnée et les techniques de commutation varient selon le protocole utilisé. Pour les paquets il s'agit de routage, contrairement à la commutation il n'y a qu'une seule technique de routage pour tous les protocoles de couche réseau, qui consiste à regarder une table pour trouver une interface de sortie qui mène à une certaine destination.

```
Gateway of last resort is 204.59.3.37 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 204.59.3.37, GigabitEthernet0/0/0
      1.0.0.0/8 is variably subnetted, 2596 subnets, 14 masks
B        1.0.0.0/24 [200/100] via 193.251.245.1, 03:27:15
B        1.0.4.0/22 [200/100] via 193.251.245.1, 07:11:21
B        1.0.4.0/24 [200/100] via 193.251.245.1, 07:10:57
B        1.0.5.0/24 [200/100] via 193.251.245.1, 07:11:32
B        1.0.6.0/24 [200/100] via 193.251.245.1, 07:11:33
B        1.0.7.0/24 [200/100] via 193.251.245.1, 07:10:57
B        1.0.16.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.64.0/18 [200/100] via 193.251.245.1, 5w1d
B        1.0.128.0/17 [200/100] via 193.251.245.1, 04:44:37
B        1.0.128.0/18 [200/100] via 193.251.245.1, 04:44:37
B        1.0.128.0/19 [200/100] via 193.251.245.1, 04:44:37
B        1.0.128.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.129.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.131.0/24 [200/100] via 193.251.245.1, 5d12h
B        1.0.132.0/22 [200/100] via 193.251.245.1, 1w3d
```

```
B        1.0.136.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.138.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.139.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.142.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.143.0/24 [200/100] via 193.251.245.1, 1w3d
B        1.0.144.0/20 [200/100] via 193.251.245.1, 04:44:28
B        1.0.160.0/19 [200/100] via 193.251.245.1, 04:44:37
```

Ceci est une fraction de la table de routage pour le protocole IPv4 du routeur public 'OpenTransit' situé en France, on peut y accéder sur telnet://route-server.opentransit.net. La table de routage est une liste d'information sur une destination avec l'interface de sortie ou le prochain saut qui mène à cette destination.

1. Les fonctionnalités de routages :

Les fonctionnalités de routage sont divisées en trois plans d'opération :

- Plan de gestion : C'est tout ce qui concerne la gestion du routeur (accès utilisateurs, moyens d'accès, privilèges des utilisateurs …)
- Plan de contrôle : le plan de contrôle permet de prendre les décisions de routage des paquets qui traversent le routeur, les protocoles de routage remplissent la fonction du plan de contrôle.
- Plan de donnée : C'est ce qui concerne le transite des paquets à travers le routeur (quelle interface de sortie utiliser pour transférer le paquet).

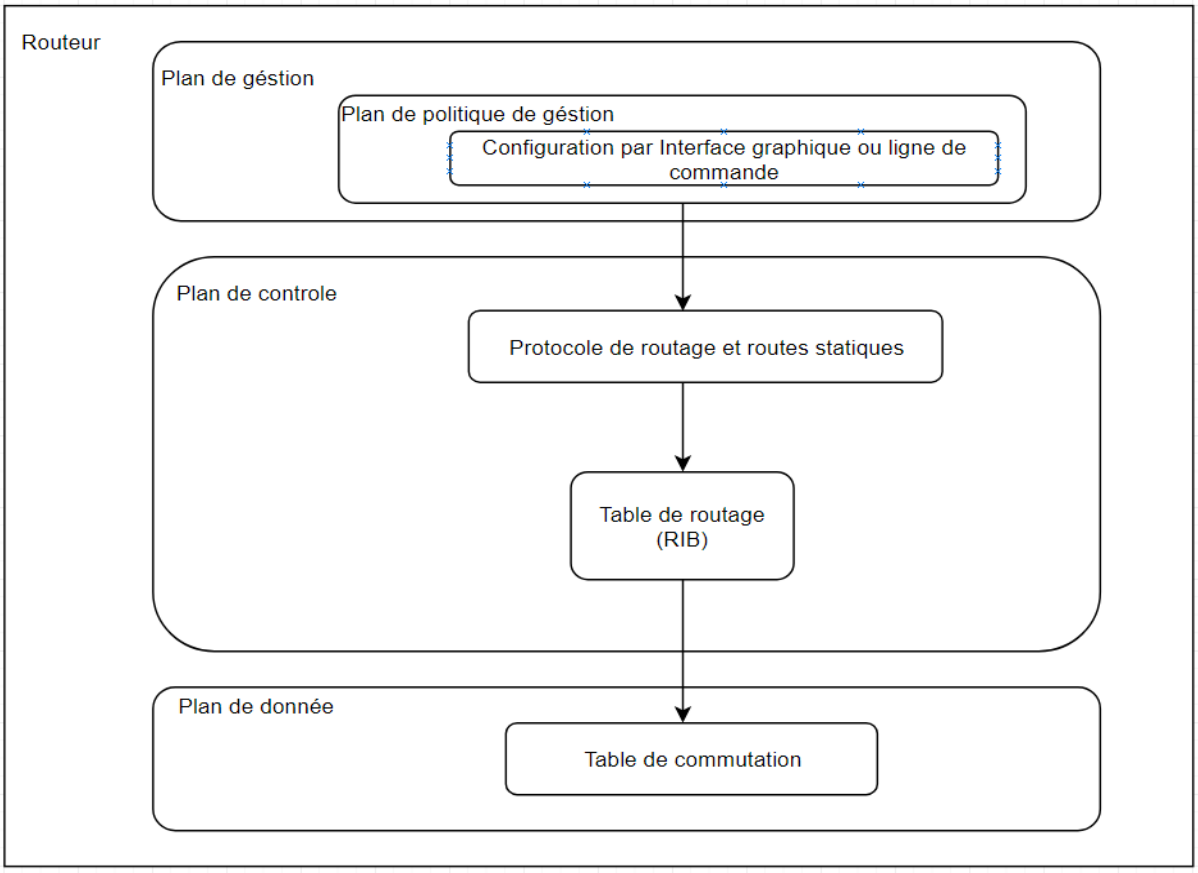

Figure 1-2 Fonctionnalités de routage

2. Les protocoles de routage

Les protocoles de routage visent à faciliter la gestion d'un réseau en automatisant la construction et maintenance des tables de routage. Les routeurs s'échanger leurs informations d'accessibilité avec d'autre routeurs du réseau dans le but de rendre accessible les réseaux auxquels ils ont un accès direct à d'autres routeurs plus éloignés. Les protocoles de routage sont deux catégories majeures : les protocoles de routage internes à un domaine et les protocoles externes entre différents domaines.

Les protocoles de routages internes se divisent en deux familles

- Les protocoles à vecteur de distance : Les protocoles à vecteur de distance (RIPv1, RIPv2, RIPng, IGRP) utilise comme métrique le nombre de routeurs par lesquels un paquet doit passer pour atteindre une certaine destination. Il existe cependant un protocole à vecteur de distance plus récent et plus évolué (EIGRP) qui utiliser d'autres métriques plus avancées.
- Les protocoles à états de lien : Les routeurs utilisant un protocole à état de lien (OSPF, IS-IS, NLSP, DNA v5) ont une vision complète du réseau. Ils construisent à partir des informations reçues de leurs voisin un graphe représentant les liens entre les nœuds du réseau. Une fois le graphe construit, chaque routeur calcule indépendamment le chemin le moins couteux vers chaque destination.

Les informations de routage des premiers protocoles à vecteur de distance sont extrêmement rudimentaires, ces informations sont la direction et le nombre de saut pour y arriver. Avec des informations aussi limitées ces protocoles font assez souvent de mauvaises décisions de routage on appelle cela le 'routage par rumeur'.

Les protocoles à état de lien ont une vision totale du réseau, les informations sur les liens qui leur sont transmise sont de première main par les nœuds qui y sont directement connectés à ces liens.

3. Les protocoles à vecteur de distance

Le nom vecteur de distance vient du fait que les routes sont annoncées comme des vecteurs de type (distance, direction), la distance est une métrique définie par le nombre de sauts (passage par un routeur) requis pour atteindre la destination. Chaque routeur apprend des routes de la perspective de ses voisins, et annonce ses routes à d'autres voisins selon sa perspective à lui. L'algorithme des protocoles à vecteur de distance annonce périodiquement avec des messages de diffusions l'intégralité de sa table de routage.

Le fait d'annoncer l'intégralité de sa table de routage périodiquement en message de diffusion, disqualifie les protocoles à vecteur de distance pour un usage sur de très larges réseaux où les tables de routages sont particulièrement grandes. Le protocole à lui seul peut saturer les ressources du réseau avec ses annonces.

Un routeur utilisant un protocole à vecteur de distance repose sur les informations reçues de ses voisins qu'eux-mêmes ont reçu de leurs voisins, ce qui détériore la précision de l'information, ce type de propagation de l'information de routage est appelé 'routage par rumeur', ceci peut induire les routeurs en erreur et provoquer des inconsistances dans le routage et des boucles de routage.

4. Les protocoles à état de lien

Les algorithmes des protocoles de routage à état de lien utilisent une base de données répliquée et distribuée, Chaque nœud (routeur) participant au routage avec le même protocole contribue à la création de la base de données à état de lien en y indiquant les informations le concernant et concernant son environnement. Les informations produites par un nœud sont propagées à tous les

autres nœuds. Une fois que chaque nœud ait fini de propager ses informations, les nœuds commencent à constituer chacun sa base de données localement. La base de données de tous les nœuds doit être identique, elle décrit la topologie du réseau avec ses liens et leur cout, à partir ses informations un nœud peut construire un graphe des nœuds et liens qui les relient. Chaque routeur calcule en utilisant l'algorithme du chemin le plus court de la théorie des graphes proposé par Dijkstra, le chemin le moins couteux depuis sa perspective pour se rendre vers chaque destination.

A. Open Shortest Path First ou OSPF

Open Shortest Path First (OSPF) a été développé par l'Internet Engineering Task Force (IETF) comme remplacement pour les protocoles à vecteur de distance qui avaient des fonctionnalités extrêmement limitées. Le protocole à vecteur de distance utilisé l'époque (RIP) avait atteint ses limites avec la croissance fulgurante des réseaux des années 80s. Les problèmes constatés avec ce protocole sont :

- Saturation des ressource réseau par le protocole de routage lui-même.
- Les temps de convergence été trop longs.
- Les métriques utilisées n'été pas fiables.
- Le nombre de saut qu'une route peut avoir été limitée.
- Possibilité de boucle de routage.

OSPF a été la solution aux problèmes constatés sur RIP, il a aussi apporté de nouvelles fonctionnalités aux protocoles de routages internes.

a. Fonctionnement d'OSPF

Dans les protocoles à état de lien, chaque routeur maintien sa base de données décrivant la topologie du réseau. On appelle cette base de données 'la base de données d'état de lien'. Chaque nœud participant au routage à une base de données d'état de lien identique aux autres nœuds. Chaque nœud contribue à la construction de la base de données d'état de lien. À partir de la base de données d'état de lien un nœud construit l'arbre des chemins les plus courts ayant soi-même comme racine de l'arbre. (J. Moy, 1998)

b. Les Link State Advertisement ou LSA

Chaque nœud du domaine OSPF crée des Link State Advertisement (LSA). Ces LSA sont la description des liens du nœud. Une fois les LSA crées le nœud les propagent vers les autres nœuds OSPF du domaine. La collection de toutes les LSA créés par le nœud et les LSA valides reçues d'autres nœuds du domaine vont constituer la base de données d'état de lien. Ces LSA vont être utilisées pour former le graphe de la topologie du réseau.

Les LSA stockées dans la base de données à état de lien ont une durée de vie de 60 minutes. Quand l'âge d'une LSA atteint 60 minutes elle doit être retirée de la base de données d'état de lien. Le nœud auteur d'une LSA doit actualiser celle-ci toutes les 30 minutes, ceci sert de mécanisme keep-alive pour des LSA individuelles. Les nœuds OSPF remplacent l'ancienne LSA à chaque fois qu'ils reçoivent une nouvelle copie.

- Toutes LSA commence par un entête de 20 octets, L'entête LSA contient les champs suivants :

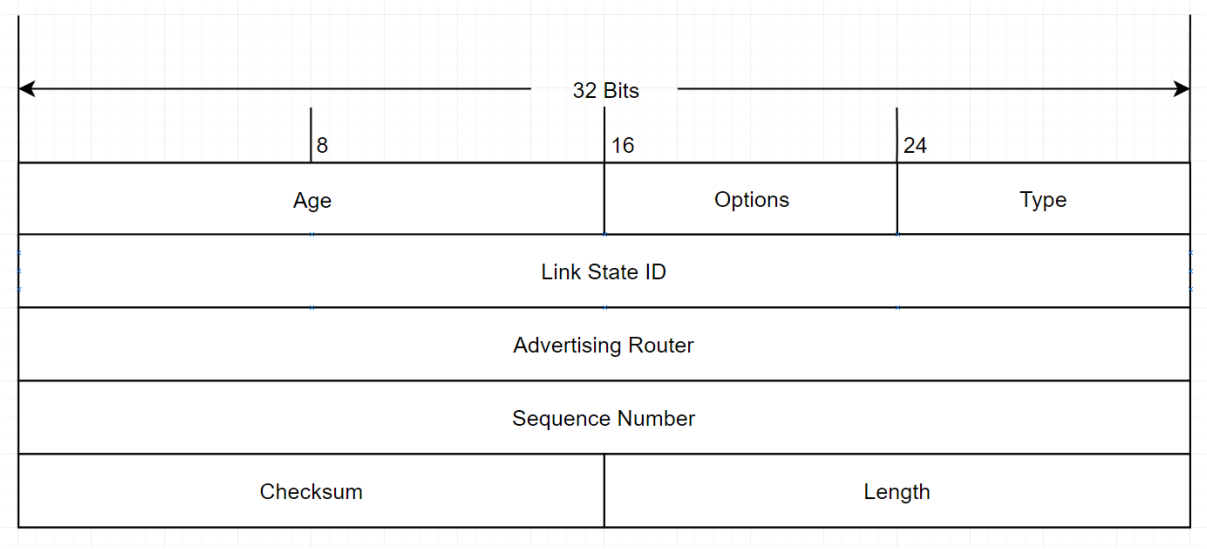

Figure 1-3 entête LSA

Toutes les LSA dans la base de données d'état de lien commencent par un entête LSA, c'est aussi utilisé dans les messages descriptifs de la base de données et les messages d'acquittement.

- Age : Champ de deux octets, indique le temps en second écoulé depuis la création de la LSA.
- Options : Champ d'un octet, indique les capacités optionnelles que possède le domaine OSPF décrit par la LSA.
- Type : champ d'un octet, indique le type d'LSA.
- Link State ID : Champ de quatre octets, identifie la portion du domaine OSPF décrite par la LSA. Ce champ à un usage variable selon le type d'LSA.
- Advertising Router : Champ de quatre octets, indique le nœud qui est l'auteur de la LSA.
- Sequence Number : Champ de quatre octets, indique la séquence de l'instance de la LSA, chaque fois qu'une nouvelle instance de la LSA est créé sa séquence est incrémenté. Cela permet aux nœuds de distinguer l'instance la plus récente.
- Checksum : Champ de deux octets, c'est un hash du message avec l'entête inclus. Il sert à déterminer s'il y a eu des erreurs lors de la transmission.
- Length : Champ de deux octets, indique la taille en octet de la LSA, l'entête est inclus.

Il existe plusieurs types de nœud OSPF, il est nécessaire d'utiliser un type d'LSA spécifique pour chaque type. Les types LSA sont les suivantes :

Code de type	Description
1	LSA de nœud
2	LSA de réseau
3	LSA de résumé de réseau
4	Résumé d'un réseau d'une autre zone
5	LSA d'un AS externe
6	LSA d'appartenance à un groupe
7	LSA NSSA externe
8	LSA à attributs externes
9	LSA opaque qui a comme portée la liaison de donnée
10	LSA opaque qui a comme portée la zone OSPF
11	LSA opaque qui a comme portée le système autonome

Après l'entête LSA on retrouve le corps de la LSA, chaque type d'LSA possède son propre format. Pour notre étude nous nous intéressons seulement au premier type d'LSA.

i. LSA de nœud

Cette LSA est créé par chaque nœud OSPF. Elle contient la liste des lien et interface du nœud, elle inclut aussi l'état et le cout des liens ainsi qu'une liste de voisins connus sur le lien. Le champ 'Link state ID' de l'entête LSA contient l'identifiant du nœud qui est l'auteur de la LSA. L'LSA de nœud ajoutes à l'entête LSA les champs suivants :

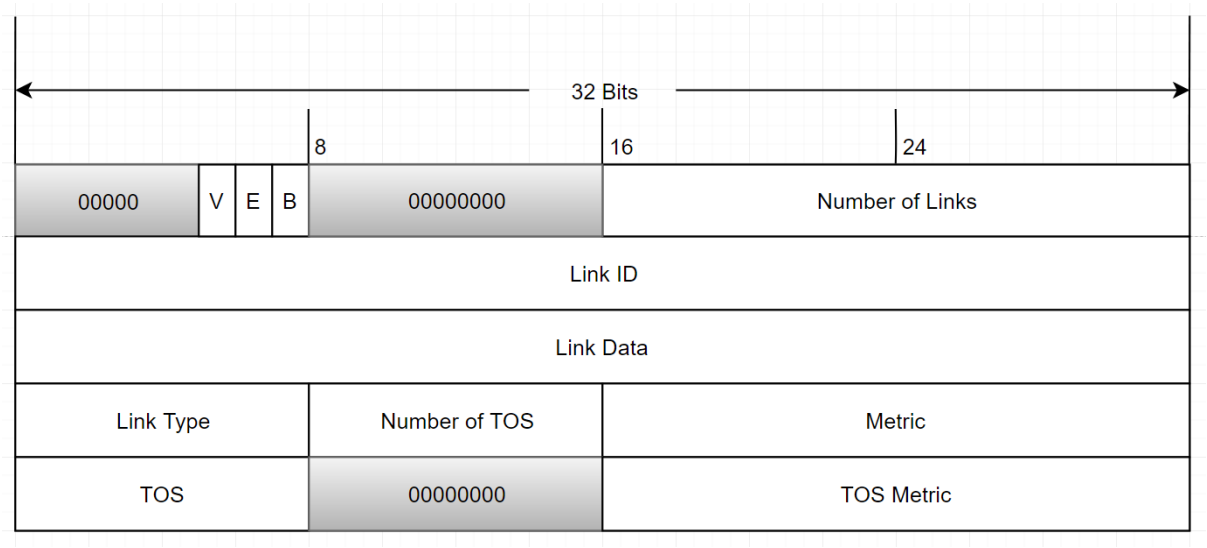

Figure 1-4 Champs du corps d'une LSA de nœud

- V ou Virtual Link Endpoint : Ce bit est mis à 1 si l'auteur de la LSA est un nœud qui est à l'extrémité d'un ou plusieurs liens virtuels.
- E ou External : Ce bit est mis à 1 si l'auteur de la LSA est un nœud qui est à la frontière d'une du domaine réseau.
- B ou Border : Ce bit est mis à 1 si l'auteur de la LSA est un nœud qui est à la frontière d'une zone OSPF.
- Number of Links : Champ de deux octets, indique le nombre de lien décrit dans cette LSA.

Les champs suivants décrivent chaque lien de la LSA individuellement, ces champs peuvent apparaitre une ou plusieurs fois selon le nombre de lien décrit dans la LSA.

- Link ID : Champ de quatre octets, identifie les objets reliés au lien. La valeur de ce champ dépend du champ suivant 'Link Type'. Le tableau suivant montre ce qu'on peut trouver dans le champ 'Link ID' selon la valeur du champ 'Link Type' :

Valeur du champ 'Link Type'	Valeur du champ 'Link ID'
1	L'identifiant du nœud voisin
2	L'adresse IP de l'interface du nœud élu comme nœud principal
3	Adresse IP de réseau du lien
4	L'identifiant du nœud voisin

- Link Data : Champ de quatre octets, indique l'adresse IP de l'interface du nœud associé. Ce champ dépend de la valeur du champ suivant 'Link Type'. Le tableau suivant montre ce qu'on peut trouver dans le champ 'Link ID' selon la valeur du champ 'Link Type'

Valeur du champ 'Link Type'	Valeur du champ 'Link Data'
1	L'adresse IP de l'interface du nœud reliée au lien
2	L'adresse IP de l'interface du nœud reliée au lien
3	Adresse IP de réseau du lien
4	L'identifiant du nœud voisin

- Link Type : Champ d'un octet, indique le type de connexion de ce lien, le tableau suivant présente les valeurs possibles ainsi que le type de lien associé à ces valeurs

Valeur	Type de connexion du lien
1	Connexion point à point d'un nœud à un autre
2	Connexion à un réseau de transite
3	Connexion à un réseau de bout 'stub'
4	Lien virtuel

- Number of TOS : Champ d'un octet, indique le nombre de métrique 'Type of Service' inclus pour le lien. Ce champ n'est plus utilisé dans les version récente d'OSPF mais il est toujours inclus pour des raisons de rétrocompatibilités.
- Metric : Champ de deux octets, indique le cout du lien.

Les deux champs suivants avec l'espace inutilisé de 8 bits vont se répéter pour un lien au tant de fois qu'indiqué dans le champ précédent 'Number of TOS'

- TOS : Champ d'un octet, indique le Type of Service.
- TOS Metric : Champ de deux octets, indique la métrique associée au Type of Service.

c. Les messages OSPF

Les nœuds OSPF utilisent plusieurs types de message pour communiquer entre eux. OSPF commence par créer des relations de voisinage fonctionnelles entre des nœuds voisins qui partagent la même liaison de donnée, pour cela il utilise le protocole Hello, ce protocole est responsable de la formation et la maintenance des relations de voisinage fonctionnelles entre nœuds OSPF.

Une fois les relations de voisinage fonctionnelles entre les nœuds du réseau créées, ces nœuds doivent synchroniser leur base de données d'état de lien. Il est requis que tout nœuds avec une relation de voisinage fonctionnelle aient des bases de données synchronisées. Un nœud décrit sa base de données en envoyant une série de paquets descriptifs de la base de données. Le processus d'envoie et de réception de paquet descriptifs est appelé 'processus d'échange de base de données'

Les messages OSPF commencent tous par un même entête de base, voici le format paquet de l'entête de base OSPF :

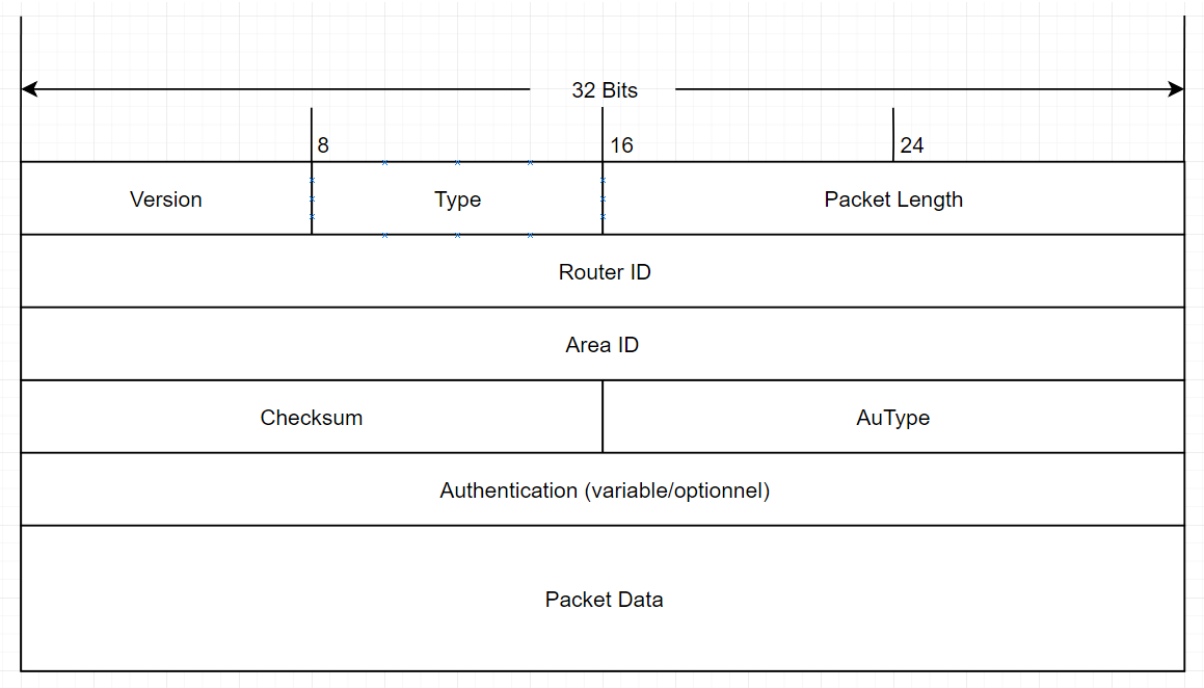

Figure 1-5 entête des messages OSPF

Les champs de l'entête sont :

- Version : Champ d'un octet, indique la version du protocole OSPF utilisé par le nœud.
- Type : Champ d'un octet, c'est un code indiquant le type de message qui va être contenu ensuite dans le champ 'Packet Data'. Il y a cinq types de message OSPF le tableau suivant montre le code utilisé pour chaque message OSPF

Code	Message correspondant
1	Hello
2	Database Description
3	Link State Request
4	Link State Update
5	Link State Acknowledgment

- Packet Length : Champ de deux octets, Il indique la taille totale du message, l'entête est inclus.
- Router ID : Champ de quatre octets, il identifie de façon unique le nœud qui est l'auteur du message.
- Area ID : Champ de quatre octets, il identifie la zone où est situé le lien du nœud qui est l'auteur du message.
- Checksun : Champ de deux octets, c'est un hash du message avec l'entête inclus. Il sert à déterminer s'il y a eu des erreurs lors de la transmission.
- AuType : Champ de deux octets, c'est un code qui indique le mode d'authentification utilisé. Il existe trois modes d'authentification, ils sont présentés dans le tableau suivant

Code	Mode d'authentification
0	Aucune authentification
1	Authentification avec mot de passe en clair.
2	Authentification avec mot de passe crypté

- Authentication : Champ optionnel et à longueur variable, ce champ contient l'information d'authentification nécessaire au paquet pour être authentifié.
- Packet Data : Champ à longueur variable, contient le corps du message OSPF.

i. Le Message HELLO

Le protocole Hello est essentiel au fonctionnement d'OSPF, pour un nœud OSPF il sert les rôles suivants :

a) La découverte de nœuds voisins (partageant une même liaison de données) et utilisant OSPF.
b) L'annonce des paramètres prérequis à la formation d'une relation de voisinage.
c) Mécanisme keep-alive, permet de faire savoir aux autres nœuds que le nœud est toujours opérationnel est que la liaison n'a pas été interrompu.
d) Assure que la communication est fonctionnelle dans les deux directions pour deux nœuds avec une relation de voisinage fonctionnelle.
e) Mécanisme d'élection pour former une hiérarchie entre les nœuds quand plusieurs nœuds OSPF sont reliés à une même liaison de donnée.

Les messages HELLO commencent par l'entête OSPF et contiennent les champs additionnels suivants :

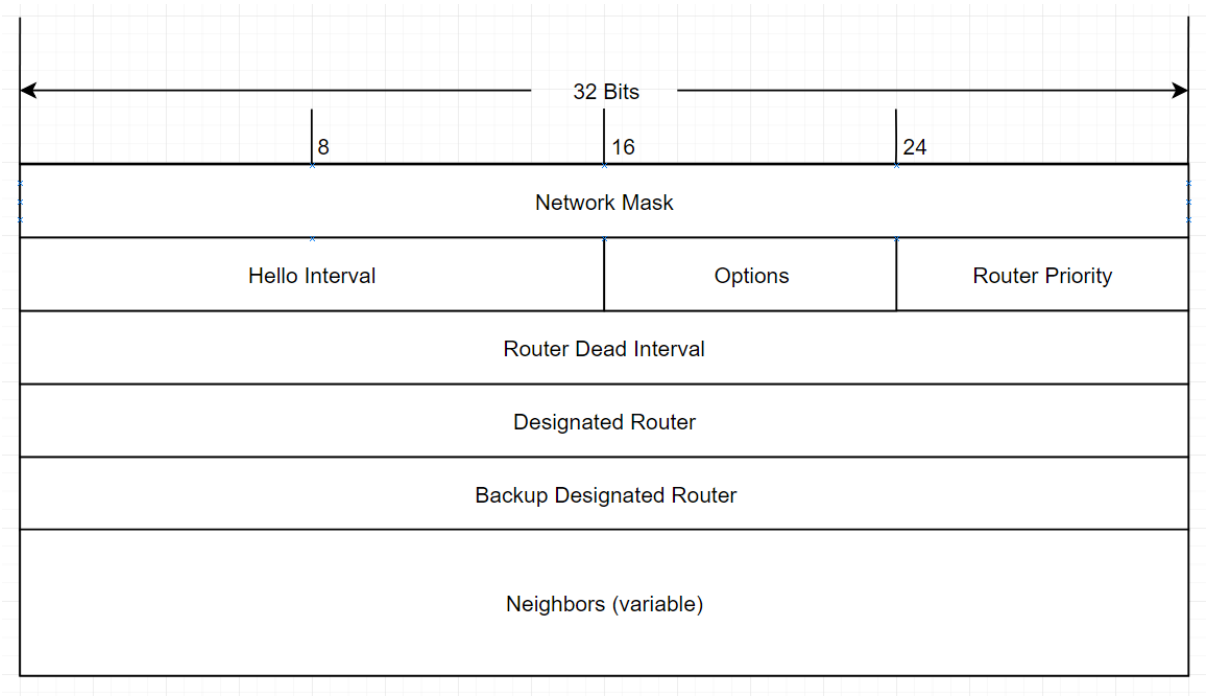

Figure 1-6 Champs additionnels du message HELLO

- Network Mask : champ de quatre octets, contient le masque réseau de l'interface qui a émis le message Hello. Le nœud ayant reçu le message HELLO doit s'assurer que le masque réseau du message correspond au masque réseau de l'interface où le message a été reçu. Dans le cas où les masque réseau ne sont pas identique le message doit être détruit.
- Hello Interval : champ de deux octets, c'est un entier non signé qui indique la fréquence en secondes à laquelle les messages HELLO doivent être envoyés. Ceci est utile pour le mécanisme keep-alive. Il est important que deux voisins aient le même paramètre Hello Interval, sans cela la relation de voisinage fonctionnelle ne peut pas se former.

- Options : champs d'un octet, ce champ inclus les capacités d'un nœud OSPF.
- Router Priority : Champ d'un octet, c'est un entier non signé qui indique la priorité du nœud, cette priorité est utilisée lors du processus d'élection des nœuds pour une hiérarchie.
- Router Dead Interval : Champ de quatre octets, c'est un entier non signé qui indique un délai en secondes qu'un nœud devrait attendre sans recevoir de message HELLO d'un autre nœud avant de le déclarer mort.
- Designated Router : Contient l'adresse IP de l'interface connectée à la même liaison de donnée du nœud qui a été élu comme nœud principal, ceci est nécessaire que pour les liaisons de donnée qui relient plus de deux nœuds (liaison de donnée point-à-multipoint), dans le cas où il y a que deux nœuds dans la liaison de donnée (liaison de donnée point-à-point), tous les bits de ce champ sont à 0.
- Backup Designated Router : Contient l'adresse IP de l'interface connectée à la même liaison de donnée du nœud qui a été élu comme nœud principal de secours, ceci est nécessaire que pour les liaisons de donnée qui relient plus de deux nœuds (liaison de donnée point-à-multipoint), dans le cas où il y a que deux nœuds dans la liaison de donnée (liaison de donnée point-à-point), tous les bits de ce champ sont à 0.
- Neighbors : c'est un champ de quatre octets, ce champ peut apparaitre plusieurs fois dans un même message. Un nœud liste les identifiants (Router ID) de tous nœuds voisins dont il a reçu un message HELLO dans ses messages HELLO.

ii. Les messages descriptifs de la base de données (Database Description ou DD)

Quand une relation de voisinage fonctionnelle entre deux nœuds OSPF se forme, les messages descriptifs de la base de données sont utilisés pour annoncer les liens présents dans la base de données d'un nœud dans le but de synchroniser les bases de données des deux nœuds.

Un message descriptif de la base de données commence par l'entête de base OSPF et contient les champs additionnels suivants :

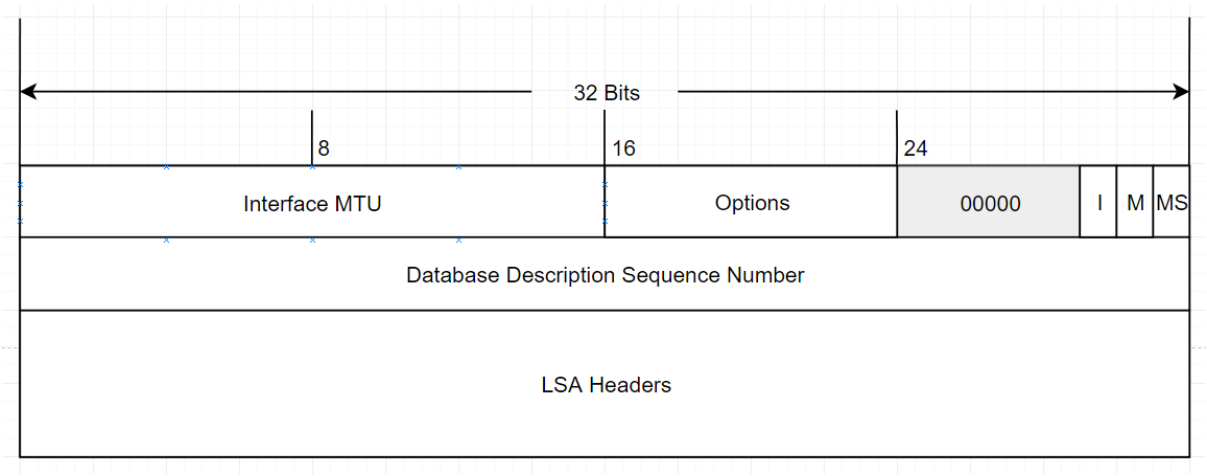

Figure 1-7 champs additionnels du message descriptif de base de données

- Interface MTU : Champ de 2 octets, c'est un entier non signé qui indique la taille maximal qu'un paquet doit avoir pour être envoyé sans aucun besoin de le fragmenter.
- Options : Champ d'un octet, ce champ est utilisé pour indiquer à un nœud de ne pas annoncer certaines annonces à d'autres nœuds qui ne possèdent pas une certaine capacité.
- 5 bits inutilisés toujours à 0.

- I : Le bit I aussi appelé bit Initial, Si le bit est à 1 cela signifie que ce message descriptif est le premier paquet d'une série de messages descriptifs, tous les messages suivants devront avoir le bit Initial à 0
- M : Le bit M aussi appelé bit More, Si le bit est à 1 cela signifie que ce message descriptif n'est pas le dernier de la série de messages. Le dernier message de la série doit avoir le bit More à 0
- MS : Le bit MS aussi appelé bit Master/Slave, Si le bit est à 1 cela signifie que l'auteur du message est le maitre durant le processus de synchronisation des bases de données. L'esclave à le bit MS à 0.
- Database Description Sequence Number : Champ de 4 octets, c'est un entier non signé, il indique un numéro de séquence pour les messages descriptifs de la base de données. Cela assure que toute la série a été envoyée et reçue avec succès. La séquence du premier message est déterminée par le maitre, la séquence est incrémentée pour chaque message descriptif de la base de données suivant.
- LSA Headers : Champs de 30 octets, ce champ peut apparaitre plusieurs fois, c'est la liste de quelques ou tous les entêtes LSA contenus dans la base de données d'état de lien de l'auteur.

iii. Messages de requête d'état de lien (Link State Request ou LSR)

Lors du processus de synchronisation des bases de données d'état de lien, les nœuds s'envoient des messages descriptifs de leur base de données. Chaque nœud prend note de toutes les LSA des messages descriptifs de base de données qu'il ne possède pas déjà. Ces LSA qu'il ne possède pas sont ajoutées à une liste appelé 'Link State Request List'. Le nœud envoie ensuite des messages de requête 'Link State Request' pour demander une copie de ces LSA manquantes.

Les messages de requête d'état de lien commencent par l'entête OSPF et contiennent les champs additionnels suivants :

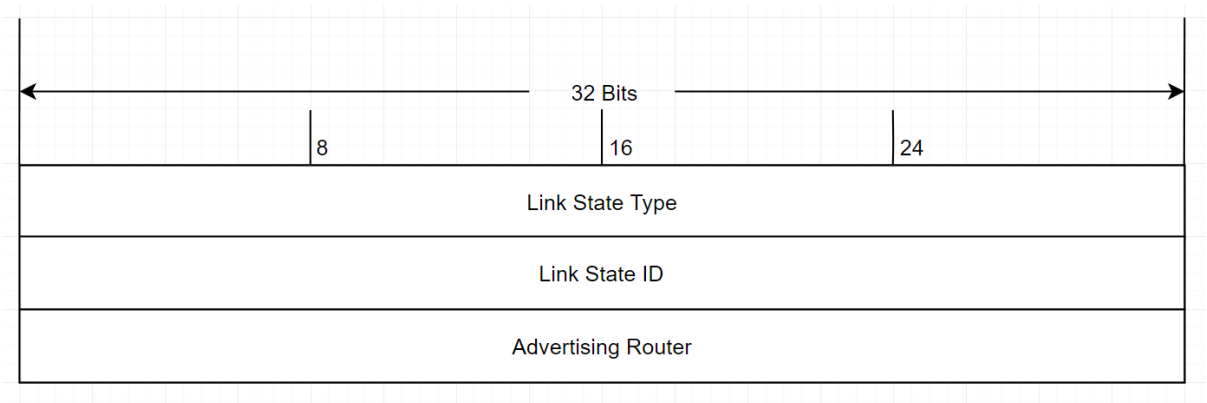

Figure 1-8 champs additionnels du message de requête d'état de lien

- Link State Type : Champ de 4 octets, indique le type d'LSA.
- Link State ID : Champ de 4 octets, le sens de ce champ dépend de la valeur du champ précèdent.
- Advertising Router : Champ de 4 octets, indique l'identifiant du nœud qui est l'auteur de la LSA.

Les champs additionnels du message de requête d'état de lien peuvent être présents plusieurs fois. Dans un seul message de requête d'état de lien on peut avoir plusieurs LSA demandées en un seul message, chaque LSA est identifiée par la combinaison des trois champs (Link State Type, Link State ID, Advertising Router).

iv. Message de mise à jour d'état de lien (Link State Update ou LSU)

Ce message est utilisé pour propager une LSA en réponse à un message de requête d'état de lien à travers le domaine OSPF.

Les messages de mise à jour d'état de lien commencent par l'entête OSPF et contiennent les champs additionnels suivants :

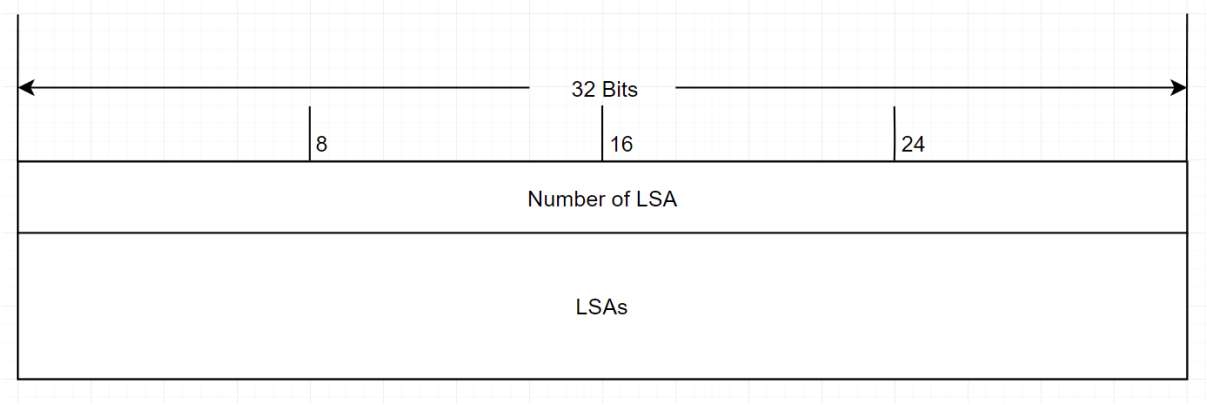

Figure 1-9 Champs additionnels du message de mise à jour d'état de lien

- Number of LSA : Champ de quatre octets, spécifie le nombre d'LSA contenues dans le message de mise à jour d'état de lien
- LSAs : Champ à longueur variable, contient des LSA complètes (entête et corps).

v. Message d'acquittement d'état de lien

Chaque LSA reçu par un nœud OSPF doit être explicitement acquittée. La LSA acquittée doit avoir son entête inclus dans le message d'acquittement, un message d'acquittement peut acquitter plusieurs LSA à la fois.

Les messages d'acquittement d'état de lien commencent par l'entête OSPF suivi des entêtes LSA à acquitter.

d. Les relations de voisinage fonctionnelles OSPF

Pour qu'OSPF commence à opérer, il doit d'abord découvrir des nœuds voisins et établir des relations de voisinage fonctionnelles avec eux. Les nœuds voisins seront enregistrés dans ce qu'on appelle une 'table de voisinage', cette table associe chaque nœud voisin avec l'interface par laquelle il a été découvert.

i. Les interfaces OSPF

Avant d'envoyer des messages HELLO et former des relations de voisinage fonctionnelles, un nœud OSPF doit d'abord découvrir ses interfaces. La RFC 2328 défini une structure de données pour chaque interface, cette structure contient les informations suivantes (J. Moy, 1998) :

- Type : Indique le type de liaison de donnée de l'interface. (Point à point, point à multipoint, lien virtuel, broadcast …)
- État : Indique le niveau de fonctionnement de l'interface.
- Adresse IP : Indique l'adresse IP associée à l'interface.

- Masque réseau : Indique le masque réseau associée à l'interface.
- Zone : Indique la zone OSPF à laquelle l'interface est attachée.
- HelloInterval : Indique le délai en secondes, qu'un nœud doit attendre entre l'envoi de deux messages HELLO sur l'interface. Cet intervalle sera indiqué au nœud voisins dans les messages HELLO.
- RouterDeadInterval : Indique le délai d'attente en seconde, que le nœud voisin doit attendre s'il ne reçoi pas de message HELLO de cette interface avant de déclarer le nœud mort.
- InfTransDelay : Indique le temps estimé en secondes, de transmission d'un message de mise à jour d'état de lien à travers le lien de l'interface. Les LSA contenues dans un message de mise à jour d'état de lien vont avoir leur âge incrémenté avec ce temps avant d'être envoyés par cette interface.
- Priorité : Indique la priorité du nœud pour devenir nœud principal sur cette interface.
- Compteur Hello : C'est un compteur à intervalle (l'intervalle utilisé est HelloInterval), à chaque intervalle un message HELLO est envoyé sur l'interface.
- Compteur Wait : C'est un compte à rebours qui enclenche la sortie de l'état 'Waiting'.
- Liste des nœuds voisins : Liste des nœuds voisins sur cette interface, cette liste est créée par le protocole Hello.
- Nœud élu principal : Indique l'adresse IP du nœud élu comme nœud principal pour la liaison de donnée à laquelle l'interface est attachée.
- Nœud élu principal de secours : Indique l'adresse IP du nœud élu comme nœud principal pour la liaison de donnée à laquelle l'interface est attachée.
- Cout de l'interface pour l'envoi : Indique le cout d'envoi des données par l'interface.
- RxmtInterval : Indique le temps d'attente en secondes, pour la retransmission d'LSA pour les relations de voisinage sur cette interface.
- AuType : Indique le mode d'authentification utilisé sur la liaison de donnée à laquelle est attachée l'interface.
- Donnée d'authentification : contient la clé d'authentification à utiliser pour communiquer sur la liaison de donnée à laquelle est attachée l'interface.

(1). L'automate d'états des interfaces OSPF

Avant de devenir fonctionnelle une interface OSPF doit d'abord transiter par plusieurs états. Le schéma suivant représente l'automate des états d'une interface OSPF (J. Moy, 1998) :

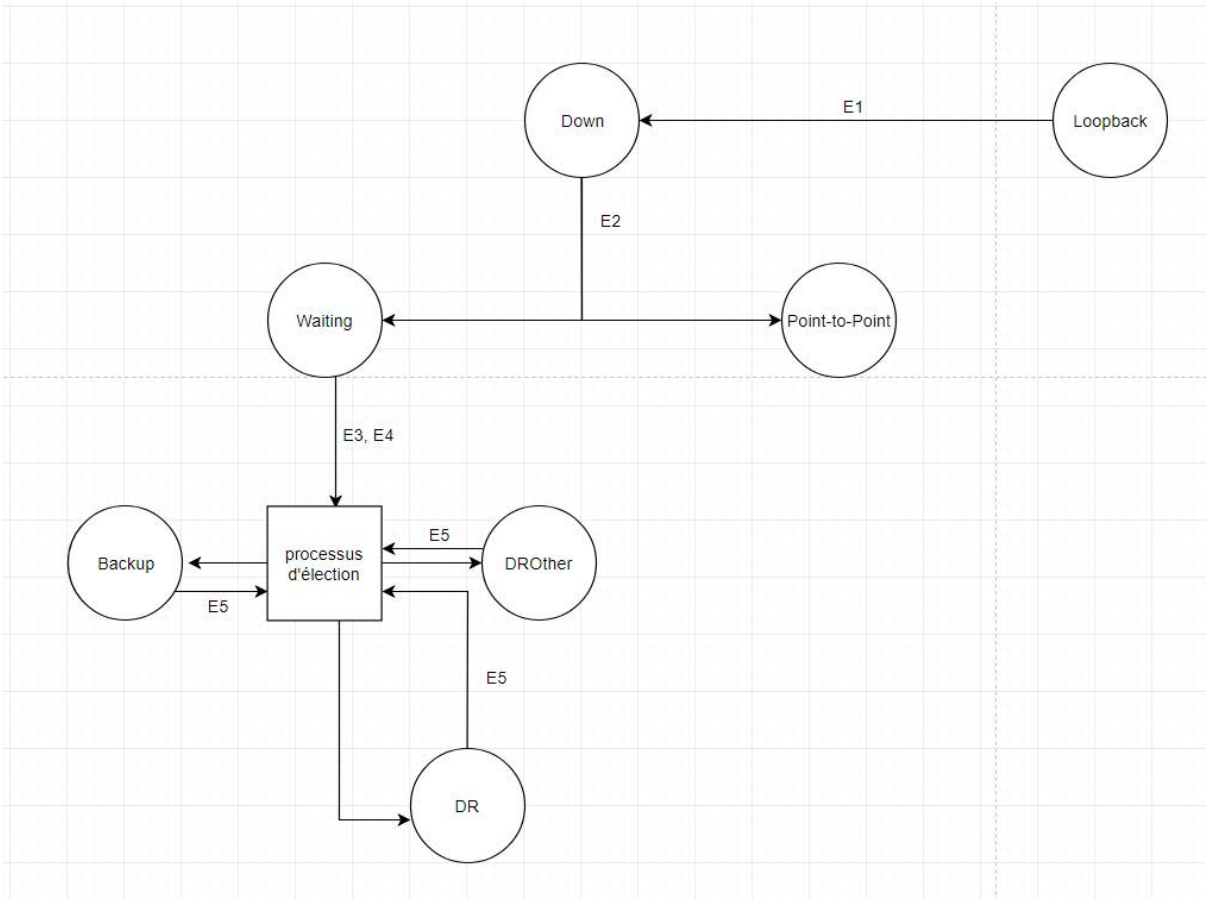

Figure 1-10 automate des états d'une interface OSPF

Les états d'une interface OSPF sont :

- Loopback : L'interface est bouclée (looped back) avec le logiciel ou de façon matérielle. Une interface bouclée ne peut pas être utiliser pour l'acheminement de donnée. L'interface se comporte comme un hôte réseau standard.
- Waiting : Cet état s'applique seulement aux interfaces connectées à une liaison de donnée de type Broadcast ou NBMA (Non-Broadcast Multiple Access). Quand une interface transite vers cet état elle commence à envoyer et recevoir des messages HELLO, elle enclenche aussi le compte à rebours du compteur Wait. Le nœud va tenter d'identifier les nœuds élus comme nœud principal et nœud principal de secours
- Point-to-point : Cet état s'applique seulement aux interfaces connectées à une liaison de donnée de type point-à-point ou lien virtuel. Quand une interface transite vers cet état elle immédiatement opérationnelle. Elle commence à envoyer des messages HELLO à intervalle régulier. Après réception de message HELLO le nœud va tenter de former une relation de voisinage avec le nœud de l'autre côté du lien.
- DROther : Dans cet état le nœud n'est pas élu comme nœud principal ou principal de secours. Le nœud va former des relations de voisinage avec les nœuds reliés à la liaison de donnée à accès multiple.
- Backup : Dans cet état le nœud est élu comme nœud principal de secours. Le nœud va former des relations de voisinage avec les nœuds reliés à la liaison de donnée à accès multiple.
- DR : Dans cet état le nœud est élu comme nœud principal. Le nœud va former des relations de voisinage avec les nœuds reliés à la liaison de donnée à accès multiple.

Les événements qui causent le changement d'état de l'interface, représentés comme des arcs dans la figure 1-10 sont expliqués dans le tableau suivant :

Événement	Nom et Description
E1	UnLoopInd : Une indication que l'interface n'est plus bouclée a été reçue
E2	InterfaceUp : Un protocole de couche inférieur a indiqué que l'interface est opérationnelle.
E3	WaitTimer : Indique que le compte à rebours du compteur Wait terminé. La période d'attente pour l'élection des nœuds est terminée.
E4	BackupSeen : Le nœud a détecté la présence ou l'absence d'un nœud principal de secours sur la liaison de donnée.
E5	NeighborChange : Un changement de nœuds voisins sur la liaison de donnée s'est produit, un nouveau nœud principal de secours doit être élu.

Il existe deux événements additionnels qui ne sont pas représentés dans la figure 1-10, il important de mentionner.

- InterfaceDown : Un protocole de couche inférieur a indiqué que l'interface n'est pas opérationnelle. Cet événement force n'importe quel état sauf 'Loopback' à passer vers l'état 'Down'.
- LoopInd : Une indication que l'interface est bouclée a été reçue. Cet événement force n'importe quel état à passer vers l'état 'Loopback'.

Dans notre étude, on s'intéresse seulement aux liens point-à-point, nous n'utiliserons pas de liens de liaison de donnée à accès multiple. Pour la suite nous allons étudier seulement ce qui concerne les liaisons de donnée point-à-point.

(2). Formation des relations de voisinage fonctionnelles

OSPF défini plusieurs états pour un voisin (l'état de la conversation avec le voisin). Pour qu'une relation de voisinage ne soit totalement opérationnelle, un voisin doit passer de l'état inopérant à des états intermédiaires avant d'arriver à un état fonctionnel.

Une fois l'interface d'un nœud OPSF opérationnelle, il peut commencer à envoyer et recevoir des messages HELLO avec et ainsi former des relations de voisinage. Chaque message HELLO contient les informations suivantes concernant l'auteur du message :

- Identifiant du nœud.
- Identifiant de la zone du nœud.
- Masque réseau de l'interface par laquelle le message a été émis.
- Le mode d'authentification et l'information d'authentification.
- Les temps à utiliser pour les compteurs HelloInterval et RouterDeadInterval.
- Priorité du nœud.
- Les nœuds désigné comme nœud principal et nœud principal de secours.
- Capacités optionnelles que le nœud possède.
- Les identifiants des nœuds dont un message HELLO a été reçu sur l'interface

Pour pouvoir former une relation de voisinage fonctionnelle deux nœuds doivent avoir les mêmes paramètres en ce qui concerne :

- L'identifiant de la zone du nœud.
- Le masque réseau.
- Le mode d'authentification ainsi que ces paramètres.
- Les temps des compteurs

Si ces paramètres sont identiques alors le message HELLO est déclaré valide. Les messages HELLO inclus une liste de voisins, si un nœud retrouve son propre identifiant dans cette liste de voisin alors il sait que la communication est fonctionnelle dans les deux directions.

L'automate d'état de la figure 1-11 montre ces états (J. Moy, 1998) :

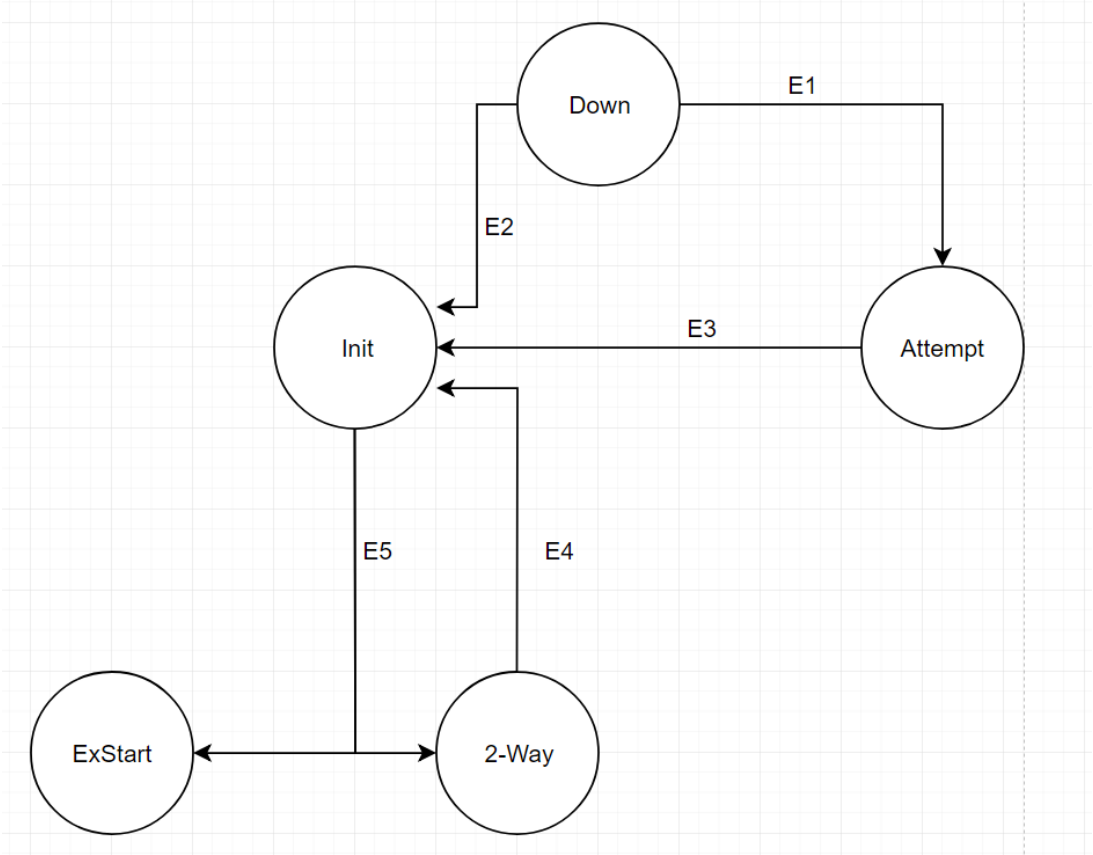

Figure 1-11 Automate d'état de voisinage OSPF

Les états de voisinage sont :

- Down : C'est l'état initial de la conversation avec le voisin. Cet état indique qu'aucune information récente de la part du voisin n'a été reçu.
- Attempt : Cet état s'applique seulement aux voisins attachés à un lien de liaison de donnée à accès multiple. Cet état indique qu'aucune information récente de la part du voisin n'a été reçu.
- Init : Cet état indique qu'un message HELLO de la part du voisin a été récemment reçu. Cependant la communication bidirectionnelle n'a pas encore été établie avec le voisin. (Les voisins dans l'état Init sont désormais inclus dans la liste des voisins des messages HELLO)
- 2-Way : Cet état indique que la communication bidirectionnelle avec le voisin est établie.
- ExStart : Cet état est la première étape dans la création d'une relation de voisinage fonctionnelle entre les deux nœuds voisins. Dans cet état le nœud maitre, le numéro de séquence pour les messages descriptifs de la base de données d'état de lien sont déterminés.

Les événements qui causent le changement d'état du voisin, représentés comme des arcs dans la figure 1-11 sont expliqués dans le tableau suivant :

Événement	Nom et Description
E1	Start : Cet événement est une indication que le nœud est autorisé à envoyer des messages HELLO sur l'interface à intervalle de HelloInterval. Cet événement est généré seulement pour les interfaces reliées à une liaison de donnée point-à-multipoint.
E2	Hello Received : Un message HELLO de la part du voisin a été reçu.
E3	1-Way Received : Un message HELLO de la part du voisin a été reçu. Le nœud local n'a pas été mentionné dans la liste des voisins du message. Ceci indique que la communication n'est pas bidirectionnelle.
E4	2-Way Received : Un message HELLO de la part du voisin a été reçu. Le nœud local figure dans la liste des voisins du message. La communication bidirectionnelle a été établie.

(3). Synchronisation des bases de données d'état de lien

Une fois la relation de voisinage fonctionnelle, les nœuds OSPF passe à des états de synchronisation des bases de données d'état de lien. L'automate présenté dans la figure 1-12 va de l'état 'ExStart' présenté dans la figure 1-11 vers un état de synchronisation complet.

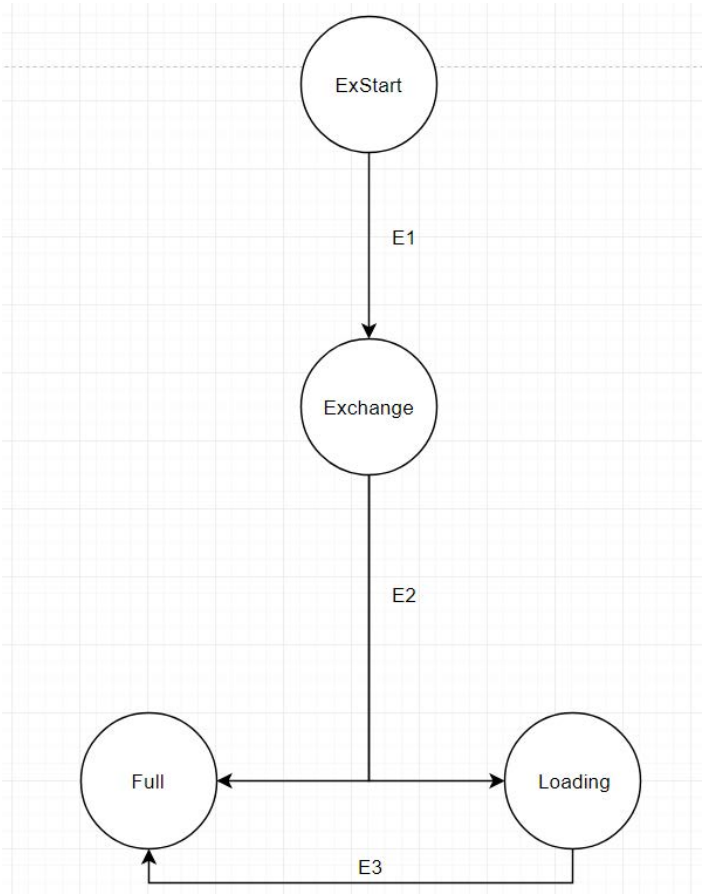

Figure 1-12 Automate d'états de synchronisation des bases de données d'état de lien

Les états de synchronisation sont :

- ExStart : Cet état est la première étape dans la création d'une relation de voisinage fonctionnelle entre les deux nœuds voisins. Dans cet état le nœud maitre, le numéro de séquence pour les messages descriptifs de la base de données d'état de lien sont déterminés.

- Exchange : Dans cet état le nœud décrit la totalité de sa base de données d'état de lien au nœud voisin avec des messages descriptifs de base de données.
- Loading : Dans cet état le nœud envoie des messages de requête de base de données au voisin pour demander les LSA qui lui manque.
- Full : Dans cet état les bases de données sont totalement synchronisées.

Les événements qui causent le changement d'état du voisin, représentés comme des arcs dans la figure 1-12 sont expliqués dans le tableau suivant :

Événement	Nom et Description
E1	NeogotiationDone : La relation maitre/esclave a été négociée. La séquence pour les messages descriptifs a été échangée.
E2	ExchangeDone : Les deux nœuds ont terminé de transmettre la séquence complète de leur base de données d'état de lien avec les messages de description de base de données.
E3	LoadingDone : Des messages de mise à jour d'état de lien ont été reçus pour toutes les LSA demandées.

II. Le routage inter domaine et systèmes autonomes

ARPANET, le précurseur d'Internet a commencé à grandir extrêmement vite à la fin des années 70s. Au fur et à mesure que le réseau grandissait, gérer le routage de ce réseau avec un simple protocole de routage devint presque impossible. La visibilité de chaque sous réseau par toutes les passerelles du réseau pouvait rendre la stabilité impossible et la sécurité était totalement compromise.

Pour résoudre le problème on devait trouver un moyen adaptatif pour la gestion des réseaux. L'idée à laquelle sont parvenu les ingénieurs est celle des domaines administratifs qui furent appelés des systèmes autonomes (en anglais 'Autonomous Systems' ou AS), Qui définit les frontières d'un réseau sous une même administration et un protocole de routage qui diffuse les informations de routage entre ces domaines. On dit que le protocole est externe aux domaines administratifs.

1. Exterior Gateway protocol (EGP)

EGP était le premier protocole de routage entre les AS. Ce protocole est totalement obsolète depuis les années 90s mais nous allons le citer pour des raisons historiques, mais aussi pour tirer une leçon des erreurs qu'avaient commises les ingénieurs lors de la conception de ce protocole.

Au début des années 80s ARPANET était constitué de passerelles qui diffusaient leurs routes avec un protocole à vecteur de distance appelé Gateway-to-Gateway Protocol (GGP). Grâce à ce protocole, chaque passerelle (Gateway) connaissait un chemin vers chaque sous réseau.

Les problèmes que les ingénieurs ont rencontrés étaient liés à l'extensibilité du réseau. Ils ont réalisé que les protocoles qu'ils avaient n'étaient guère adaptés à un réseau qui ne cesse de grandir.

Parmi les problèmes qu'ils ont rencontrés les plus importants sont (Rosen & Inc., 1982) :

- Les algorithmes de routage des passerelles étaient inadaptés à un large réseau ; ils généraient beaucoup de surcharge sur le réseau. Avec l'énorme taille des tables de routage chaque mise à jour des informations de routage consommait entièrement les ressources du réseau.

- GGP a été implémenté sur une large variété de plateformes physiques. Les implémentations sont devenues tout aussi diverses ce qui conduisait à des problèmes de compatibilité.
- GGP requière une bonne coordination entre les administrateurs des différentes passerelles constituant le réseau, chose qui devient de plus en plus difficile à mesure que le réseau grandit.

La solution qui a été proposé était de diviser le réseau ARPANET, qui était un réseau unique, en plusieurs sections réseau interconnectés 'inter-networks'. Chaque réseau était régi par une autorité administrative indépendante qui avait une liberté totale en ce qui concerne la gestion de son réseau.

Les AS étaient désormais libres de distribuer leurs informations de routage interne comme ils voulaient, cela a permis le développement de nouveaux protocoles de routage externes.

Les AS partagent leurs informations de routage à travers leurs passerelles de frontière, ils utilisent pour cela un protocole externe indépendant du protocole de routage interne. Le premier protocole à avoir été utilisé entre des AS était EGP qui est un protocole à vecteur de distance, ce n'est pas tout à fait un protocole de routage car il ne partageait pas d'informations sur les routes mais des informations d'accessibilité (en Anglais Reachability Information ou RI). Ces RI constituent une liste des réseaux ainsi que les cheminements permettant de les atteindre.

2. Topologie autorisée par EGP

Pour le fonctionnement d'EGP la topologie en étoile est nécessaire. On peut distinguer deux types de passerelles EGP : les passerelles centrales (Core Gateway) situées dans le système autonomes centrale (ou Core AS), et les passerelles de bout (Stub Gateway) situées dans les systèmes autonomes de bout (AS Stub).

Il ne peut y avoir qu'un seul AS Core, et tout autre AS appelé Stub doit être relié directement au Core. On obtient ainsi une topologie étoilée.

Cette hiérarchie à deux niveaux (un niveau Core et un niveau Stub) est nécessaire car EGP n'a aucun mécanisme de prévention des boucles ce qui conduit à la nécessité d'une topologie sans boucle.

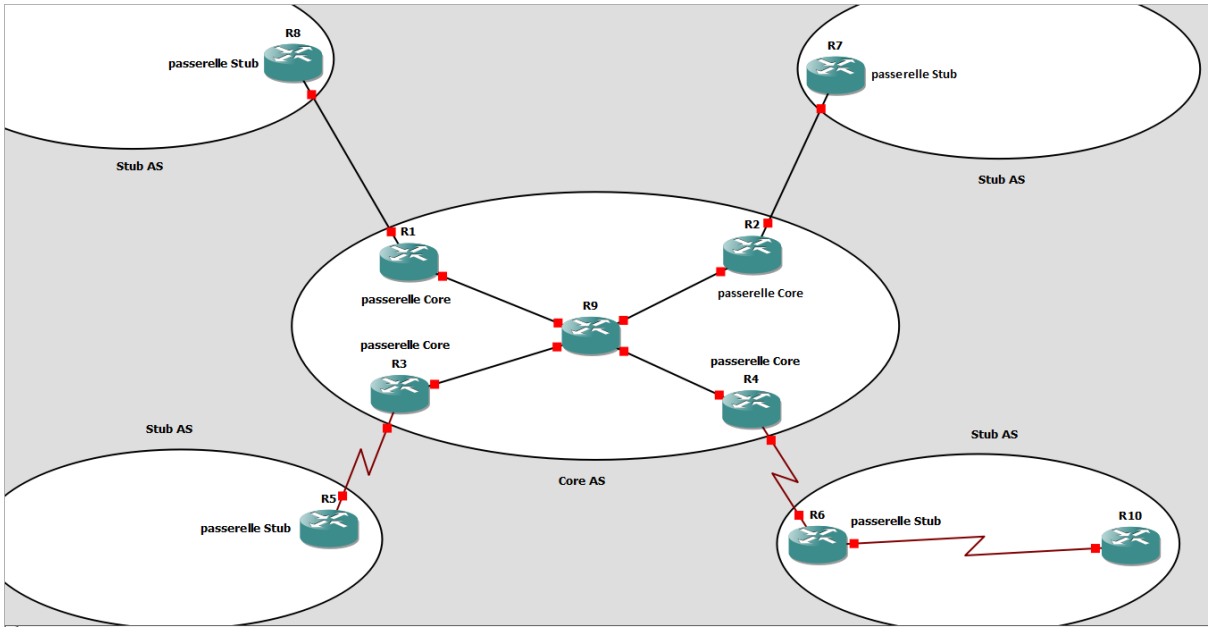

Figure 2-1 Topologie sans boucle représentant les Core et Stub AS

Les rôles des passerelles Core et Stub diffèrent. Une passerelle Stub ne peut envoyer que des RI qui concernent seulement son réseau interne, alors qu'une passerelle Core peut transmettre des RI reçues des passerelles Stub. (Rosen & Inc., 1982)

Au début des années 80s la nécessité d'une topologie limitée n'était pas un problème pour ARPANET, mais pour continuer à évoluer vers ce qu'est aujourd'hui Internet (un réseau constitué d'AS sans aucune structure) il est impossible de produire un environnement sans boucle entre les AS pour que le protocole EGP puisse fonctionner correctement.

Un autre problème a surgi avec EGP ; les différents systèmes autonomes du réseau Internet organisent des politiques qui concernent le trafic de données qui traversent leur système autonome, ainsi ils déterminent ce qui peut passer à travers leur réseau. Or EGP ne permet pas un routage basé sur la politique d'ingénierie du trafic.

3. Le passage vers Border Gateway Protocol (BGP)

Les ingénieurs ont essayé d'améliorer EGP mais ça a été un échec, EGP est un protocole d'accessibilité et n'a jamais été un protocole de routage. Transformer EGP en un protocole de routage reviendrait à le reconcevoir en partant de zéro, et le résultat final n'aurait plus rien à voir avec l'initial EGP, et ce fut le cas.

Les ingénieurs ont développé un nouveau protocole de routage inter domaine capable de faire tout ce que EGP ne pouvait pas faire. Ce protocole nommé BGP fut introduit pour la première fois en 1989 et remplaça définitivement EGP et c'est aujourd'hui le seul protocole de routage inter domaine utilisé sur Internet.

A. Présentation de BGP

BGP établit des connexions 'peer-to-peer' manuellement entre deux routeurs BGP. BGP ne possède pas de protocole ou mécanisme de transport propre à lui mais utilise TCP sur le port 179 pour l'établissement de la connexion, l'envoi d'information et l'acquittement.

BGP est un protocole à vecteur de chemin car pour calculer le cout d'une route il utilise comme métrique une liste d'AS par lesquels un paquet doit transiter pour atteindre une destination.

Quand un routeur BGP annonce une route à un autre routeur BGP, il lui envoie un tableau d'attributs concernant cette route. Parmi ces attributs on retrouve un attribut appelé 'AS_PATH' qui est une liste d'AS par lesquels on doit transiter pour arriver à une destination. Pour qu'un routeur annonce une route BGP qu'il vient de recevoir il doit ajouter son numéro d'AS dans l'AS_PATH avant de l'annoncer.

Si un routeur reçoit une nouvelle route et s'aperçoit que son numéro d'AS figure déjà dans l'AS_PATH il détermine alors qu'une boucle s'est produite et que la route reçue est erronée.

Avec BGP un paquet peut transiter par un ou plusieurs AS pour atteindre sa destination. Cela change entièrement la définition des types d'AS d'EGP où le seul AS par lequel les paquets avaient le droit de transiter est le Core AS. Dans BGP il y a deux types d'AS

- Les 'Stub AS' qui sont des systèmes autonomes par les quels aucun trafic ne transite, chaque paquet qui rentre dans l'AS est destiné à cet AS et chaque paquet qui en sort a été émis par l'AS
- Les 'Transit AS' qui sont des systèmes autonomes qui permettent à un trafic dont ils ne sont ni les destinataires ni les émetteurs de transiter par leur réseau.

On peut distinguer deux formes assez similaires de BGP mais avec quelques différences non négligeables dans leur fonctionnement :

- Interior BGP ou IBGP, est utilisé entre deux routeurs BGP qui appartiennent au même AS.
- Exterior BGP ou EBGP, est utilisé entre deux routeurs qui appartiennent à deux AS différents.

Pour les différences de fonctionnement, deux règles qui s'appliquent à IBGP :

- Les routeurs n'ajoutent pas leurs numéros d'AS dans l'attribut 'AS_PATH' quand ils annoncent une route via IBGP.
- Une route apprise via IBGP n'est jamais annoncée à un autre routeur IBGP sauf si configuré explicitement pour le faire.

IBGP est principalement utilisé dans les AS de transit, il est beaucoup plus rare de voir IBGP sur un 'Stub AS', mais avec l'arrivée de nouvelles technologies telles que 'MPLS VPN' ou les multicast il devient plus fréquent de voir un 'Stub AS' adopter IBGP.

B. Qu'est ce qui justifie l'usage de BGP

- Une connexion à un domaine non sécurisé

Une supposition sous-jacente des protocoles de routage interne est que tous les voisins sont sous la même administration et sont donc fiables.

Mais pour ce qui est du routage inter domaine on ne peut pas supposer que les voisins ne sont pas malveillants, qu'ils ont été correctement configurés et qu'ils nous transmettent des informations

fiables. BGP a été conçu dans l'idée de partager des informations de routage entre des participants qui ne se font pas confiance.

- Plusieurs points de sorties de l'AS

Si un réseau possède plusieurs points de sortie vers un seul et même AS ou différent AS alors BGP n'est pas nécessaire mais peut être utilisé pour définir quels sont les points de sortie et d'entrées optimales pour un trafic précis, mais avec une complexité de routage et de conception ajoutées, ainsi que la nécessité de ressources matérielles supplémentaires.

- Usage des politiques de routage

Quand il existe une certaine redondance de liens vers l'extérieur, il est possible d'avoir des préférences sur les chemins à utiliser pour certains trafics bien précis. BGP offre des outils permettant de communiquer ces préférences.

(Beijnum, 2002)

C. Les Dangers de BGP

Convenir d'une association BGP implique une intéressante combinaison de confiance et de méfiance, lors de la manipulation de la configuration BGP une grande prudence est de rigueur. En effet, une simple erreur humaine peut avoir des conséquences désastreuses. Le résultat d'une mauvaise manipulation (volontaire ou involontaire) est ce qu'on appelle 'BGP Hijacking' qui est un détournement de préfixe d'adresse IP provoquant un déni de service par la redirection du trafic vers un destinataire autre que le destinataire initial.

Un exemple récent de détournement de préfixe est celui du 8 avril 2010, China Telecom, le plus grand fournisseur d'accès internet en Chine a annoncé avec BGP des préfixes d'adresse IP qui ne lui appartenait pas. Le résultat de cet incident a été une perte de connexion Internet mondiale.

(Goodin, 2010)

4. Fonctionnement de BGP

Dans ce chapitre les mécanismes de fonctionnement BGP seront détaillés.

Les routeurs BGP ne communiquent entre eux qu'avec des messages à destinataire unique (unicast) et forment uniquement des liaison point à point. BGP est un protocole de la couche 7 (application), pour son transport il utilise TCP sur le port 179 et repose sur les propriétés de TCP pour les fonctions basiques tel que la maintenance d'une liaison ou l'acquittement des paquets.

BGP est un protocole à vecteur de distance bien qu'il soit appelé protocole à vecteur de chemin. Il voit le réseau comme une suite de sauts à travers des AS au lieu de routeurs.

A. La communication avec BGP

Les routeurs BGP utilisent un ensemble de messages BGP bien défini pour communiquer.

Avant de commencer à communiquer deux routeurs doivent établir une session TCP. Une fois la session établie on peut avoir un des cinq messages de base (Narbik Kocharians, 2015) suivants :

- Message OPEN

Ce message est utilisé pour identifier le routeur émetteur. Un message Open contient les champs suivants :

- Version de BGP

Ce champ spécifie la version de BGP que l'émetteur utilise, ce champ peut aussi être utilisé pour négocier quelle version les deux voisins vont utiliser, pour que l'association 'peer-to- peer' soit établie, il est nécessaire que les deux voisins utilisent la même version.

- Numéro de l'AS

Contient le numéro du système autonome auquel l'émetteur appartient, ce champ permet de déterminer s'il faut utiliser EBGP ou IBGP.

- Hold Time

C'est le temps maximum qu'un routeur doit attendre pour déclarer un autre routeur mort s'il ne reçoit pas de lui un message KEEPALIVE ou UPDATE durant cette période. La valeur de ce compteur peut être de 0 pour indiquer que les KEEPALIVE sont désactivés, ou une valeur minimum de 3 secondes.

Si les deux routeurs sont configurés avec un temps Hold Time diffèrent alors ils utiliseront le plus petit des deux temps.

- Identifiant BGP

C'est une adresse IPv4 qui est sensée identifier un routeur de manière unique.

- Paramètres optionnels

Ce champ indique quels paramètres optionnels sont supportés (par exemple l'authentification ou le MultiProtocol BGP).

- Message KEEPALIVE

Si des routeurs parviennent à un accord de voisinage après échange de message OPEN et qu'ils s'associent, ils vont commencer à s'envoyer des messages KEEPALIVE sur une période égale à un tiers du temps du 'hold time' (exemple, si le 'hold time' est 60 secondes, ils doivent s'envoyer des KEEPALIVE toutes les 20 secondes.

- Message UPDATE

Les messages UPDATE sont utilisés pour transférer des informations de routage entre une paire de routeurs BGP. Les informations contenues dans un message UPDATE peuvent être utilisées pour construire un graphe qui décrit la relation entre les différents systèmes autonomes. En appliquant certaines règles, des boucles de routages et autres anomalies peuvent être détectées et retirées du routage inter AS.

Le message UPDATE est utilisé pour annoncer une route praticable qui partage un attribut chemin commun vers un pair. Ou pour retirer des routes devenues impraticables. Le même message peut aussi annoncer et retirer des routes en même temps.

Le message UPDATE inclut les champs suivants :

- Information d'accessibilité de la couche réseau (Network Layer Reachability Information ou NLRI)

Constitué d'un ou plusieurs tuples (Longueur, préfixe) qui annoncent les destinations avec leur longueur et préfixe

- Attributs du chemin (Path Attributes)

Donne les caractéristiques du chemin des NLRI annoncés, ces attributs fournissent les informations nécessaires à BGP pour choisir le meilleur chemin, détecter les boucles et déterminer les politiques de routage.

- Routes retirées

Un ou plusieurs tuples (Longueur, préfixe) qui décrivent des destinations qui sont devenues inaccessibles.

Un message UPDATE peut inclure plusieurs préfixes dans un NLRI, mais chaque message update décrit seulement un seul chemin BGP. En d'autres termes le champ 'attributs d'un chemin' décrit un seul chemin mais ce chemin peut mener à plusieurs destinations différentes.

- Message NOTIFICATION

Le message de NOTIFICATION est toujours envoyé quand une erreur est détectée et est toujours suivi d'une fermeture de la connexion BGP.

Le message NOTIFICATION contient un code d'erreur et un sous code d'erreur pour identifier l'erreur.

- ROUTE REFRESH

Utilisé par un routeur BGP pour dynamiquement demander à un autre pair BGP une route vers une destination, afin de rafraichir cette route.

5. Les annonces de routes BGP

Après que l'association 'peer-to-peer' BGP a été établie, les deux routeurs commencent à annoncer leurs informations de routage à travers des messages UPDATE. Le contenu du message UPDATE est détaillé dans la prochaine section.

A. Les attributs de chemin (Path Attributes)

Dans chaque protocole de routage, une route annoncée vers un préfixe de destination doit avoir ses caractéristiques quantitatives incluses dans l'annonce. Ce qui permet au routeur ayant reçu l'annonce de comparer cette route aux autres routes vers cette destination.

BGP utilise une liste d'attributs pour caractériser une route annoncée. Ces attributs sont classés en quatre catégories (Y. Rekhter, T. Li, S. Hares, 2006):

- Well-known mandatory (Conventionnel Obligatoire)
- Well-known discretionary (Conventionnel Facultatif)
- Optional transitive (Optionnel transitive)
- Optional nontransitive (Optionnel non-transitive)

‘Well-Known’ signifie que l’attribut doit être reconnu par toute implémentation logicielle BGP, à l’opposé ‘Optional’ l’attribut est juste optionnel il peut être inclus dans une implémentation logicielle tout comme il peut ne pas l’être.

Les attributs ‘Well-Known’ sont soit ‘mandatory’ ils doivent être inclus dans chaque message UPDATE, ou bien ‘’discretionary’’ ils sont optionnels dans un message UPDATE.

Les attributs ‘Optional’ sont soit ‘transitive’, (dans ce cas le processus BGP doit accepter les messages UPDATE même s’ils contiennent des attributs ‘Optional transitive’ qui ne sont pas implémentés dans son logiciel et les relayer à d’autres pairs), soit ‘nontransitive’ (le processus BGP peut ignorer le message UPDATE et le détruire).

Le tableau suivant présente les attributs les plus connus ainsi que les RFC où ils ont été définis :

Attribut	Catégorie	Défini dans
ORIGIN	Well-Known mandatory	RFC 4271
AS_PATH	Well-Known mandatory	RFC 4271
NEXT_HOP	Well-Known mandatory	RFC 4271
LOCAL_PREF	Well-Known discretionary	RFC 4271
ATOMIC_AGGREGATE	Well-Known discretionary	RFC 4271
AGGREGATOR	Optional transitive	RFC 4271
COMMUNITIES	Optional transitive	RFC 1997
EXTENDED COMMUNITIES	Optional transitive	RFC 4360
MULTI_EXIT_DISC	Optional nontransitive	RFC 4271
ORIGINATOR_ID	Optional nontransitive	RFC 4456
CLUSTER_LIST	Optional nontransitive	RFC 4456
AS4_PATH	Optional transitive	RFC 6793
AS4_AGGREGATOR	Optional transitive	RFC 6793
Multiprotocol Reachable NLRI	Optional nontransitive	RFC 4760
Multiprotocol Unreachable NLRI	Optional nontransitive	RFC 4760

a. L’attribut ORIGIN

L’attribut ORIGIN est un attribut ‘’Well-Known mandatory’’ qui spécifie l’origine de la route contenue dans le message UPDATE. Cet attribut est généré par le routeur qui est l’auteur de l’information de routage. Cette valeur ne doit pas être changée par les routeurs qui propagent la route. Il peut y avoir trois origines possibles :

- IGP : Quand la NLRI est interne au système autonome qui en est l’auteur.
- EGP : Quand BGP apprend la NLRI du protocole EGP
- Incomplète : Les informations dont dispose BGP sont incomplète pour déterminer l’origine par laquelle BGP a appris la route.

(Y. Rekhter, T. Li, S. Hares, 2006)

b. L’attribut AS_PATH

AS_PATH est un attribut ‘Well-Known mandatory’. Cet attribut identifie les systèmes autonomes par lesquels l’information de routage contenue dans le message UPDATE a transitée. Les composants de cette liste sont des segments chemin de type AS_SET ou AS_SEQUENCE où :

- AS_SEQUENCE : C'est une liste ordonnée d'AS par lesquels l'annonce a transité au début de la liste l'AS le plus récent et à la fin l'AS a l'origine de la route.
- AS_SET : C'est une liste non ordonnée des AS sur le chemin vers la destination. Ils sont utilisés par exemple lors d'agrégation de plusieurs destinations en une seule.

Un des rôles majeurs de l'attribut AS_PATH est la prévention de boucle de routage. Quand un routeur BGP reçoit un message UPDATE d'un de ses pairs, il vérifie le contenu de l'attribut AS_PATH s'il y trouve son numéro d'AS cela signifie qu'une boucle s'est produite et que l'information de routage reçu est invalide (Y. Rekhter, T. Li, S. Hares, 2006).

- Quand un routeur annonce une route qu'il a apprise via BGP il modifie l'attribut AS_PATH selon la localisation de l'interlocuteur BGP à qui la route doit être annoncée.

a) Si le routeur BGP annonce la route à un pair interne situé dans le même système autonome alors le routeur ne doit pas modifier l'attribut AS_PATH
b) Si le routeur BGP annonce la route à un pair externe situé dans un autre système autonome alors l'attribut AS_PATH doit être modifié comme suit
    a. Si le premier segment chemin de l'AS_PATH est de type AS_SEQUENCE, alors le processus BGP ajoute son propre numéro d'AS comme dernier élément de la séquence. Si le segment (limité à 255 AS) est déjà plein, alors il doit ajouter un nouveau segment chemin de type AS_SEQUENCE avec son numéro dans le nouveau segment.
    b. Si le premier segment de l'AS_PATH est de type AS_SET, le processus BGP doit ajouter un nouveau segment chemin de type AS_SEQUENCE avec son numéro d'AS dans le segment.
    c. Si l'attribut AS_PATH est vide, le processus BGP doit ajouter un nouveau segment chemin de type AS_SEQUENCE avec son numéro d'AS dans le segment.

- Quand BGP annonce une route dont il est l'auteur

a) Le routeur BGP qui est l'auteur de la route ajoute son propre numéro d'AS dans un segment de chemin de type AS_SEQUENCE pour l'envoyer à un routeur BGP pair externe. Dans ce cas de figure il n'y a qu'un seul segment chemin dans l'attribut AS_PATH et dans ce segment il n'y a qu'un seul AS.
b) Le routeur BGP qui est l'auteur de la route inclut un attribut AS_PATH vide dans ses messages UPDATE quand ils sont envoyés à un pair BGP interne.

Il est à noter que lors de la modification de l'attribut AS_PATH en ajoutant le numéro d'AS auquel appartient le routeur BGP, ce dernier peut ajouter plus d'une instance de son numéro d'AS à la séquence de l'AS_PATH. Ceci peut être contrôlé par une configuration locale dans le but d'influencer les décisions de routage d'autres routeurs.

c. L'attribut NEXT_HOP

L'attribut NEXT_HOP est un attribut 'Well-Known mandatory' qui définit l'adresse IP du routeur qui doit être utilisé comme prochain saut vers la destination présenté dans le message UPDATE. Cet attribut est calculé comme suit :

- Quand le message UPDATE est envoyé à un pair interne, si la route n'a pas été générée en interne alors le processus BGP NE DOIT pas modifier l'attribut NEXT_HOP sauf s'il a été

configuré explicitement pour annoncer sa propre adresse IP comme la valeur de NEXT_HOP. Si la route a été générée en interne le processus BGP doit utiliser l'adresse de l'interface par laquelle la route annoncée est accessible pour le routeur émetteur comme valeur de NEXT_HOP.

- Quand le message UPDATE est envoyé à un pair externe, et que le pair est à un saut de l'émetteur :
    a) Si la route à annoncer a été apprise de la part d'un pair interne ou a été généré localement, le routeur annonceur peut utiliser l'adresse de l'interface du pair interne par laquelle la route à annoncer est accessible au routeur annonceur comme valeur NEXT_HOP, à condition que le routeur externe partage une liaison de couche 2 avec cette adresse. Ceci est une forme d'attribut NEXT_HOP de partie tierce.
    b) Si la route à annoncer a été apprise de la part d'un pair externe, l'annonceur peut utiliser une adresse IP de n'importe quel routeur adjacent que l'annonceur utilise lui-même pour le calcul des routes locales comme attribut NEXT_HOP, à condition que le routeur externe partage une liaison de couche 2 avec cette adresse. Ceci est une autre forme d'attribut NEXT_HOP de partie tierce.
    c) Si le pair externe à qui la route doit être annoncée partage une liaison de couche 2 avec une des interfaces de l'annonceur, l'annonceur peut utiliser l'adresse IP d'une telle interface comme attribut NEXT_HOP
    d) Si aucune des conditions au-dessus n'est satisfaite, alors l'annonceur doit utiliser l'adresse IP par laquelle il établit la connexion BGP à ce routeur externe pour qui la route doit être annoncée.

- Quand le message UPDATE est envoyé à un pair externe, et que le pair est à plusieurs sauts de l'émetteur :

    ~~b)~~a) L'annonceur PEUT être configuré pour propager l'attribut NEXT_HOP, quand il annonce la route qu'il a apprise d'un de ses pairs, il l'annonce avec le même attribut NEXT_HOP que quand il l'a apprise. L'annonceur ne modifie pas l'attribut NEXT_HOP

    ~~c)~~b) Sinon par default l'annonceur DOIT utiliser l'adresse IP de l'interface avec laquelle il établit la connexion au pair BGP à qui il doit annoncer la route.

Un routeur BGP ne doit pas annoncer une route à un pair ou l'adresse du prochain saut est celle du routeur à qui il l'annonce, et un routeur BGP ne doit pas installer une route ou il est le prochain saut pour cette route.

L'attribut NEXT_HOP est utilisé par un routeur BGP pour déterminer l'interface de sortie pour transmettre le trafic transitant vers la destination.

(Y. Rekhter, T. Li, S. Hares, 2006)

d. L'attribut MULTI_EXIT_DISC

L'attribut MED ou encore ''multi-exit discriminator'' est un attribut ''optional non-transitive'' qui a été prévu pour être utilisé sur les liens externes pour favoriser des points d'entrées ou de sorties du système autonome voisin. Cet attribut est un champ de 4 octets contenant un nombre non-signé appelé métrique. Pour plusieurs sorties vers une même destination avec tous les facteurs à égalités, la sortie avec la métrique le plus faible est la préférée.

Si l'attribut a été reçu via EBGP, il peut être propagé à travers IBGP vers les pairs internes. L'attribut MED reçu d'un AS voisin ne doit pas être propagé aux autres AS voisins.

(Y. Rekhter, T. Li, S. Hares, 2006)

e. L'attribut LOCAL_PREF

C'est un attribut ''well-known discretionary'' qui doit être inclus dans tout message UPDATE qu'un émetteur BGP envoi a un pair interne. Un routeur BGP doit calculer le degré de préférence pour chaque route externe en se basant sur la politique interne configurée, il inclut ce degré de préférence calculé quand il annonce ces routes à ses pairs internes.

Le degré de préférence le plus élevé est le préféré.

Un émetteur BGP ne doit pas inclure l'attribut LOCAL_PREF dans ses messages UPDATE quand ils sont destinés à des pairs externes, sauf dans le cas d'une confédération BGP.

Si un router BGP reçoit un message UPDATE contenant l'attribut LOCAL_PREF d'un pair externe cet attribut DOIT être ignoré sauf en cas de confédération BGP.

(Y. Rekhter, T. Li, S. Hares, 2006)

f. L'attribut ATOMIC_AGGREGATE

C'est un attribut ''well-known discretionary''.

Quand un routeur BGP agrège plusieurs routes dans le but d'annoncer une seule route a un pair, l'attribut AS_PATH des routes agrégées inclut un segment chemin de type AS_SET qui contient les numéros d'AS déjà présents dans les routes qui ont été agrégées. L'administrateur doit déterminer si l'agrégat peut être annoncé sans l'AS_SET sans causer de boucles de routage, sinon l'AS_SET doit être inclus.

Si l'agrégat exclu des numéros d'AS présents dans les routes agrégées ou exclu carrément l'AS_SET, la route agrégée doit inclure l'attribut ATOMIC_AGGREGATE quand annoncé a des pairs.

Un routeur BGP qui reçoit une route avec l'attribut ATOMIC_AGGREGATE, ne doit pas enlever l'attribut quand il propage la route à d'autres pairs.

Un routeur BGP qui reçoit une route avec l'attribut ATOMIC_AGGREGATE, NE DOIT PAS créer des routes plus spécifiques à partir de cette route quand il l'annonce à un autre pair.

Un routeur BGP qui reçoit une route avec l'attribut ATOMIC_AGGREGATE, doit savoir que le chemin vers les destinations spécifiées dans les NLRI de la route peut ne pas être spécifié dans l'attribut AS_PATH

(Y. Rekhter, T. Li, S. Hares, 2006)

g. L'attribut AGGREGATOR

C'est un attribut ''optional transitive'', il peut être inclus dans les messages UPDATE contenant des routes formées par agrégation.

Lorsqu'un routeur BGP effectue une agrégation de route il PEUT ajouter l'attribut AGGREGATOR qui doit contenir son propre numéro d'AS ainsi que son adresse IP.

Cet attribut permet aux routeurs ayant reçu les informations de routage de localiser le point ou l'agrégation s'est produite.

(Y. Rekhter, T. Li, S. Hares, 2006)

~~*i.*~~*h. L'attribut Weight (ou POIDS)*

L'attribut weight n'est pas présent dans le tableau des attributs qui a été présenté précédemment, parce que l'attribut weight n'est pas un standard ouvert. C'est un attribut spécifique à l'entreprise 'Cisco Systems' et leurs équipements. Pour ce projet, l'entreprise 'Icosnet' utilise principalement les routeurs 'Cisco' dans son backbone, donc il est important de connaître cet attribut.

L'attribut weight est une valeur entre 0 et 65535 assignée a une route et a une signification locale au routeur. Elle n'est pas communiquée aux autres routeurs associés BGP.

Quand un routeur BGP est sur le point de choisir le meilleur chemin, le processus de décision considère la valeur weight au-dessus de toute autre caractéristique. Weight est utilisé pour influencer les décisions de routage d'un routeur.

(Narbik Kocharians, 2015)

~~B.~~

~~C.~~

~~D.~~B. Le format paquet des messages BGP

Les messages BGP sont envoyés à travers des sessions TCP. Un message BGP n'est traité qu'une fois tous ses segments reçus. La longueur maximale d'un message est de 4 096 octets et toutes les implémentations BGP sont obligées de supporter cette longueur maximale. La longueur minimale d'un message BGP qui est constitué seulement de l'entête BGP sans la partie donnée est de 19 octets.

a. Le format de l'entête BGP

Chaque message a un entête de taille fixe. La partie donnée du message vient après l'entête mais elle reste optionnelle. Voici ci-dessous le format de l'entête :

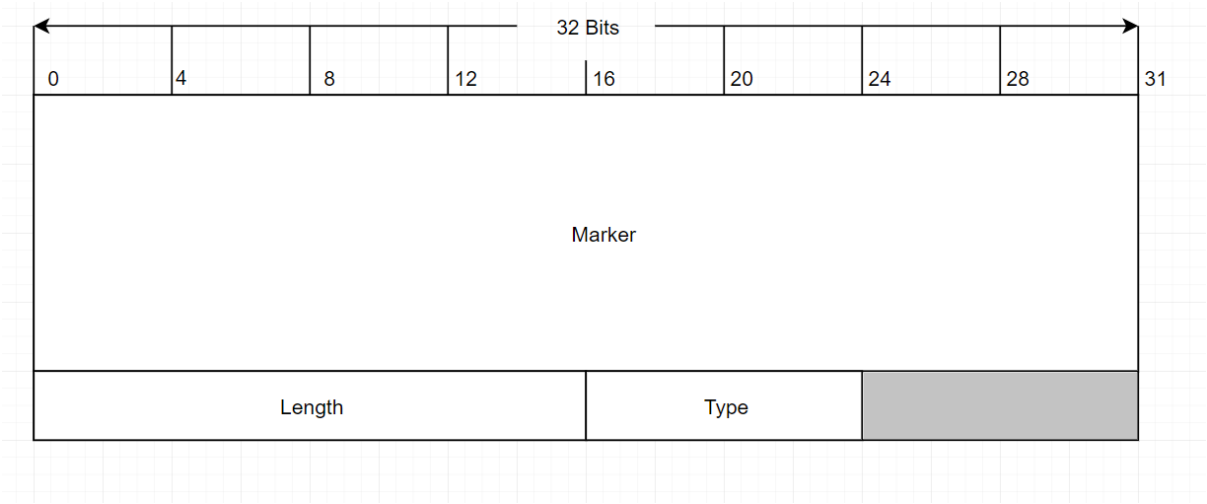

Figure 2-2 Entête BGP

L'entête de 19 octets est constitué de trois champs qui sont :

- Marker

Un champ de 16 octets, ce champ a été utilisé pour détecter la désynchronisation entre les pairs BGP. Cependant la RFC-4271 désapprouve son usage, mais le garde toujours dans l'entête BGP pour des raisons de rétrocompatibilité. Tous les bits de ce champ doivent être à 1.

- Length

Ces 2 octets représentent un entier compris entre 19 et 4096, qui indique la longueur totale du message BGP en octets avec l'entête inclus. Il permet de localiser la fin du message et le début d'un nouveau message.

- Type

Champ d'un octet qui contient un entier, indiquant le code du type de message. Les codes sont les suivants :

  - 1 – OPEN
  - 2 – UPDATE
  - 3 – NOTIFICATION
  - 4 – KEEPALIVE
  - 5 – ROUTE REFRESH

(Y. Rekhter, T. Li, S. Hares, 2006)

b. Le format du message OPEN

Le message OPEN est le premier message à être envoyé par chaque côté une fois la session TCP établie. Si le message OPEN est accepté, un message KEEPALIVE est envoyé pour confirmer le message OPEN reçu.

La longueur minimale du message OPEN avec l'entête inclus est de 29 octets. Le message OPEN additionne à l'entête les champs suivants :

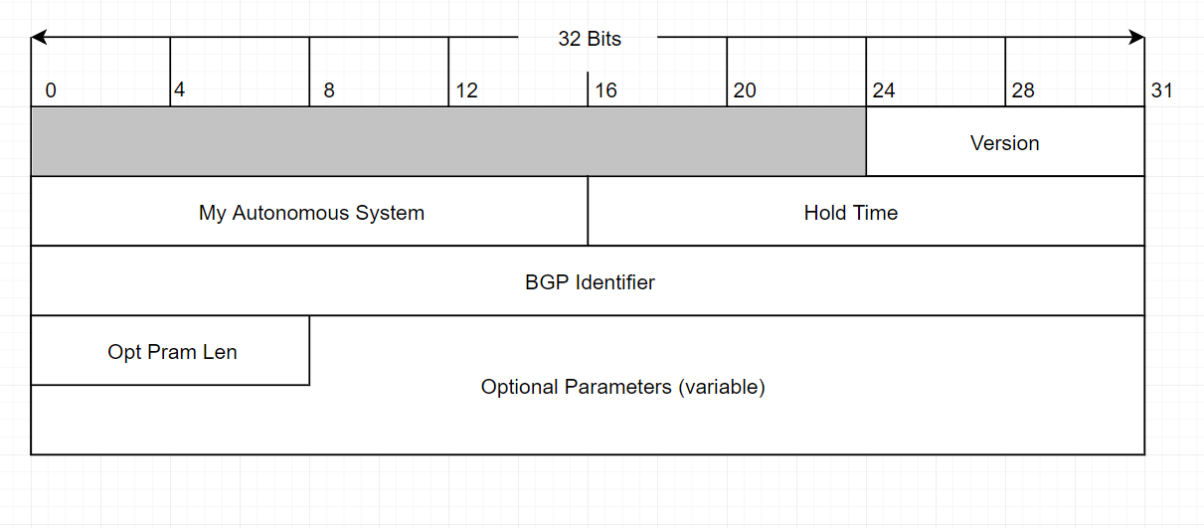

Figure 2-3 champs additionnels du message OPEN

- Version

Champ d'un seul octet qui indique la version du protocole BGP.

- My Autonomous System

Champ de 2 octets qui représente un entier indiquant le système autonome de l'émetteur du message.

- Hold Time

Champ de 2 octets qui représente un entier indiquant le nombre de seconde que l'émetteur propose d'utiliser comme temps du compteur Hold Time.

- BGP Identifier

Champ de 4 octets qui représente l'identifiant BGP de l'émetteur.

- Option Parameters Length

Champ d'un octet qui représente un entier indiquant la longueur totale en octet du champ 'Option Parameters', si la valeur est de 0, le champ suivant 'Optional Parameters' est absent.

- Optional Parameters

Ce champ à longueur variable contient la liste des paramètres optionnels, ou chaque paramètre est encodé comme suit :

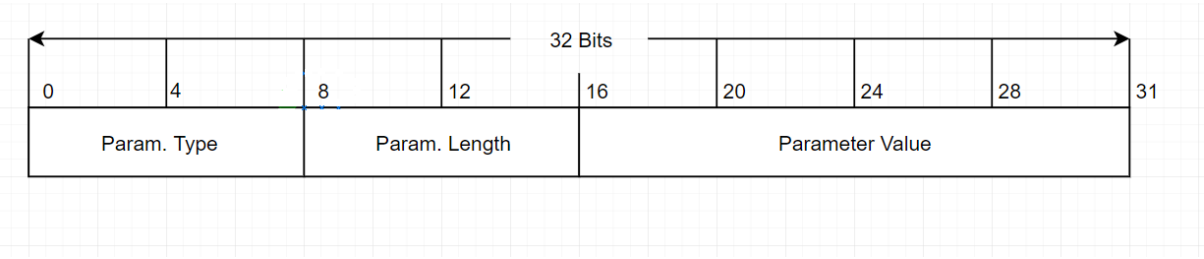

Figure 2-4 structure des champs imbriqués dans le champ 'Optional Parameters'

- Parameter Type : Champ d'un octet qui identifie un paramètre individuel sans aucune ambiguïté
- Parameter Length : Champ d'un octet qui indique la longueur en octets du champ Parameter Value.
- Parameter Value : Champ de longueur variable qui est interprété selon le contenu du champ Parameter Type.

(Y. Rekhter, T. Li, S. Hares, 2006)

c. Le format du message UPDATE

Les messages UPDATE sont utilisés pour transférer des informations de routage entre des pairs BGP, un message UPDATE peut annoncer une route vers des destinations qui partagent des attributs communs. Un message UPDATE peut simultanément annoncer une seule route praticable et retirer plusieurs routes devenues impraticable.

La longueur minimale d'un message UPDATE est de 23 octets. Le message UPDATE additionne à l'entête les champs suivants :

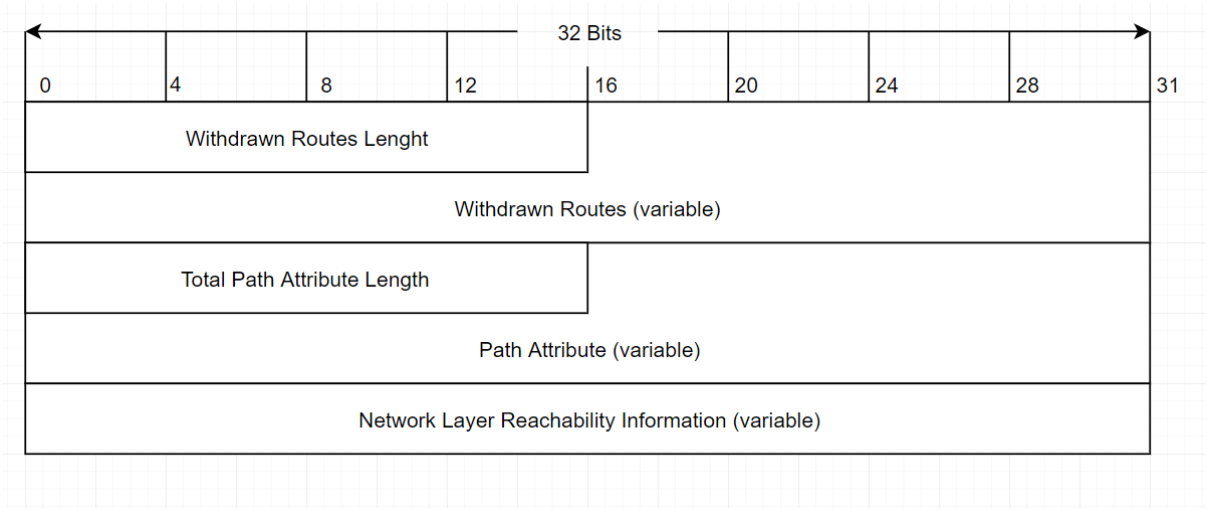

Figure 2-5 champs additionnels du message UPDATE

- Withdrawn Routes Length

Un champ de 2 octets qui représente un entier indiquant la longueur totale en octet du champ Withdrawn Routes (Routes retirées). Cette valeur est utilisée pour déterminer la longueur du champ Network Layer Reachability Information.

Une valeur égale à 0 indique qu'aucune route n'est à retirer, et que le champ Withdrawn Routes est absent.

- Withdrawn Routes

C'est un champ a longueur variable, il contient une liste de préfixe d'adresse IP pour les routes qui sont retirées. Chaque préfixe d'adresse IP est encodé sous forme de tuple (Longueur, Préfixe) avec les deux champs décrit comme suit

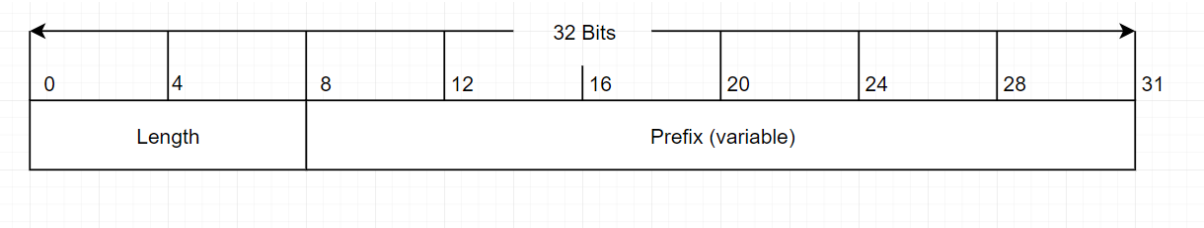

Figure 2-6 Structure des champs imbriqués dans le champ Withdrawn routes

- Length : ce champ d'un octet indique la longueur du préfixe d'adresse IP en bit.
- Prefix : ce champ à longueur variable contient un préfixe d'adresse IP suivi d'un nombre minimum de bits nécessaires pour remplir un octet.

- Total Path Attribute Length

Ce champ de 2 octets représente un entier indiquant la longueur totale en octet du champ Path Attributes. Cette valeur est utilisée pour déterminer la longueur du champ Network Layer Reachability Information.

Une valeur égale à 0 indique que les champs Path Attributes et Network Layer Reachability Information sont absents.

- Path Attributes

Ce champ est une séquence à longueur variable d'attributs. Il est présent obligatoirement dans chaque message UPDATE qui annonce des routes. Chaque attribut est un triplet de longueur variable de la forme (type d'attribut, longueur d'attribut, valeur d'attribut).

Le type d'attribut est un champ a deux octets, un octet contient les flags de l'attribut et le deuxième octet le code du type d'attribut

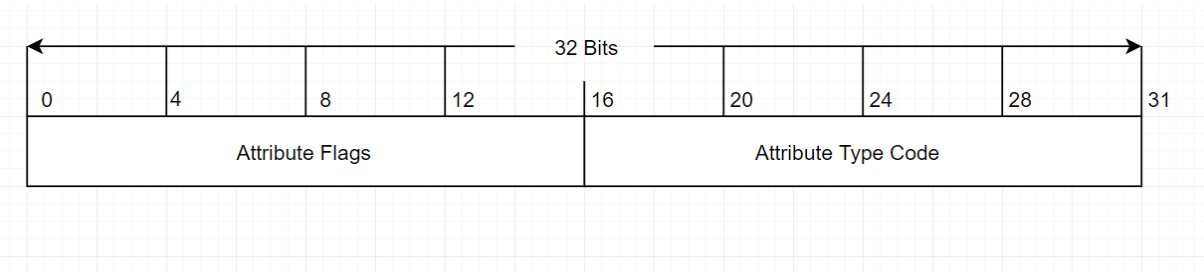

Figure 2-7 structure des champs imbriqués dans le champ Path Attributes

- Attributes Flags

Chaque bit de cet octet représente un flag, le tableau suivant illustre la position des flags dans l'octet

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|O|T|P|E|U|U|U|U| Attr.Type Code|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- O (bit 0) : Ce flag est le bit « Optional ». Si le flag est à 1 l'attribut est 'Optional', à 0 l'attribut est 'Well-known'
- T (bit 1) : Ce flag est le bit « Transitive ». Si le bit est à 1 l'attribut est transitif, à 0 l'attribut est non transitif. Pour les attributs 'Well-known' le bit doit toujours être à 1.
- P (bit 2) : Ce flag est le bit « Partial ». Il indique si l'information contenue dans l'attribut optionnel transitif est partielle (P=1), sinon complète (P=0). Pour les attributs 'Well-known' et 'Optional nontransitive' P doit toujours être à 0.
- E (bit 3) : Ce flag est le bit « Extended Length ». Il indique si la longueur est d'un octet (E=0) ou deux octets (E=1).
- U (bit 4 à 7) : Ces bits sont inutilisés, ils sont ignorés à la réception.

- Attribute Type Code

C'est un champ d'un octet indiquant le code d'un type d'attribut, le champ valeur d'attribut est un code qui dépend immédiatement du champ type d'attribut (flags, code), le tableau suivant montre les combinaisons (code attribut, valeur attribut) possibles

Type d'attribut - Code	**Signification Code**	**Valeurs Attribut – Code possibles**	**Signification code**
1	ORIGIN	0	IGP
		1	EGP
		2	Incomplet
2	AS_PATH	1	AS_SET
		2	AS_SEQUENCE
		3	AS_CONFED_SET
		4	AS_CONFED_SEQUENCE
3	NEXT_HOP	0	Adresse IP NEXT_HOP
4	MULTI_EXIT_DISC	0	4 octets pour MED
5	LOCAL_PREF	0	4 octets pour LOCAL_PREF
6	ATOMIC_AGGREGATE	0	/
7	AGGREGATOR	0	Numéro d'AS et adresse IP de l'agrégateur
8	Communauté	0	4 octets pour identifiant communautaire
9	ORIGINATOR_ID	0	4 octets pour identifiant routeur de l'initiateur
10	CLUSTER_LIST	0	Liste à longueur variable des identifiants de cluster
14	MP_REACH_NLRI	0	NLRI a longueur variable pour MP-BGP
15	MP_UNREACH_NLRI	0	NLRI a longueur variable pour MP-BGP
16	Communautés étendues	0	16 octets d'identifiants de communautés étendues

17	AS4_PATH	0	AS_PATH avec des numéros d'AS de 4 octets
18	AS4_AGGREGATOR	0	Numéro d'AS a 4 octets et adresse IP de l'agrégateur.

(Y. Rekhter, T. Li, S. Hares, 2006)

- Network Layer Reachability Information (NLRI)

C'est un champ a longueur variable, il contient une liste de préfixes d'adresses IP.

La longueur du champ NLRI n'est pas explicitement codée mais doit être calculée avec la formule suivante :

Longueur du champ NLRI = Longueur du message UPDATE (trouvée dans l'entête BGP) - 23 (longueur de l'entête + les deux champs de longueur) - Longueur du champ Path Attributes - Longueur du champ Withdrawn Routes

L'information d'accessibilité « Reachability Information » RI est codée en tuple de la forme (Longueur, Préfixe)

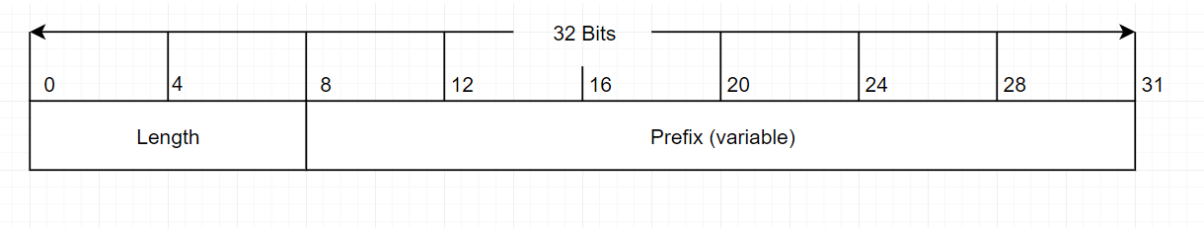

Figure 2-8 structure des champs imbriqués dans le champ NLRI

- Length : champ d'un octet qui indique la longueur en bit du préfixe d'adresse IP.
- Prefix : ce champ à longueur variable contient un préfixe d'adresse IP suivi d'un nombre minimum de bits nécessaire pour remplir un octet si nécessaire.

(Y. Rekhter, T. Li, S. Hares, 2006)

d. Le format du message KEEPALIVE

Le message KEEPALIVE est constitué seulement de l'entête BGP de 19 octets sans aucune addition.

Pour maintenir une session avec un pair, BGP n'utilise pas le mécanisme keep-alive inclus dans TCP. A la place il échange des messages KEEPALIVE avec ses pairs BGP pour réinitialiser le compteur Hold Time et éviter la fin de session automatique s'il arrive à expiration. Un intervalle raisonnable pour envoyer les messages KEEPALIVE est un tiers du temps du Hold Time. Les messages KEEPALIVE ne doivent pas être envoyés à une fréquence supérieure à un KEEPALIVE par seconde.

Si le Hold Time négocié est égal à 0 alors aucun message KEEPALIVE ne doit être envoyé.

(Y. Rekhter, T. Li, S. Hares, 2006)

e. Le format du message NOTIFICATION

Les messages NOTIFICATION sont des messages envoyés quand une condition d'erreur est détectée. Et cela résulte en une fermeture immédiate de la connexion BGP.

En addition à l'entête BGP le message de notification contient les champs suivants :

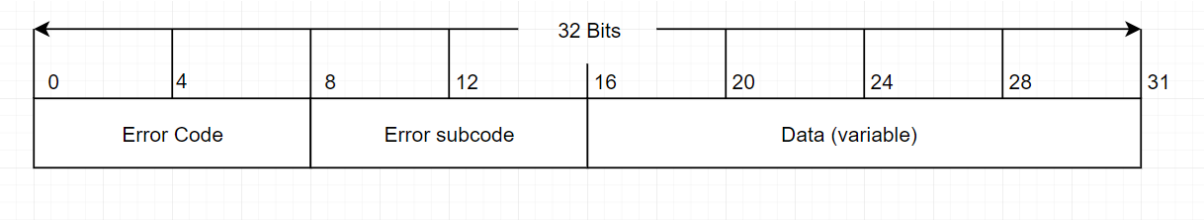

Figure 2-9 champs additionnels du message NOTIFICATION

- Error Code

Champ d'un octet qui contient un entier indiquant le type de NOTIFICATION.

- Error Subcode

Champ d'un octet qui contient un entier fournissant plus de détails concernant la nature de l'erreur signalée. Chaque code d'erreur peut avoir un ou plusieurs sous-code d'erreur associés. Si aucun sous-code approprié n'est défini pour l'erreur alors la valeur non spécifié 0 est utilisé.

- Data

Champ de longueur variable utilisé pour diagnostiquer la raison de la NOTIFICATION. Le contenu de ce champ dépend des deux champs précédents.

La longueur du champ est calculée avec la formule

Longueur du champ data = Longueur du message – 21 (Entête + Error code + Error subcode)

Le tableau suivant montre les différents codes et sous-codes des messages NOTIFICATION BGP

Error Code	Nom Anglais	Error Subcode	Details du sous code
1	Message Header Error	1	Connexion non synchronisé
		2	Mauvaise longueur du message
		3	Mauvais type de message
2	Open Message Error	1	Version BGP non supportée
		2	Numéro d'AS du pair faux
		3	Mauvaise identifiant BGP
		4	Paramètre optionnel non supporté
		5	Echec de l'authentification
		6	Hold Time inacceptable
3	Update Message Error	1	Liste d'attributs mal formée
		2	Attribut 'Well-known' non reconnu
		3	Attribut 'Well-known' manquant
		4	Erreur dans les flags d'attribut
		5	Erreur dans la longueur d'attribut
		6	Attribut ORIGIN invalide

		7	Boucle de routage (Obsolète, retirée par la RFC4271)
		8	Attribut NEXT_HOP invalide
		9	Erreur dans un attribut optionnel
		10	Champ network invalide
		11	AS_PATH mal formé
4	Hold Time Expired	0	/
5	Finite State Machine Error	0	/
6	Cease	0	/

(Y. Rekhter, T. Li, S. Hares, 2006)

6. Les décisions de routage BGP

Pour orienter un trafic interne les routeurs utilisent leur table de routage qui contient la liste des réseaux accessibles et l'interface de sortie par laquelle ils le sont. BGP utilise sa propre table de routage indépendante de la table de routage interne. La table de routage de BGP s'appelle BGP 'Routing Information Base' où RIB et elle est constituée de trois parties :

- Adj-RIBs-In

BGP Enregistre les informations de routage apprises des messages UPDATE entrants envoyés par des pairs dans la Adj-RIBs-In. Leur contenu représente des routes qui peuvent être sélectionnées pour être utilisées dans le processus de routage. Ces routes sont appelées en anglais ''feasible routes'' des routes faisables ou réalisable.

- Loc-RIB

Contient les informations de routage que le routeur BGP a sélectionné en appliquant des règles et politiques aux informations de routage contenues dans l'Adj-RIBs-In. Ces informations de routage peuvent être insérées dans la RIB pour être utilisées dans le processus de routage.

- Adj-RIBs-Out

Contient les informations de routage sélectionnées par le processus BGP pour être annoncées à ses pairs. Les informations de routage stockées dans l'Adj-RIBs-Out sont annoncées dans des messages UPDATE.

(Y. Rekhter, T. Li, S. Hares, 2006)

A. Sélection de route BGP par le processus de décision

Le processus de décision BGP sélectionne des routes en appliquant des règles et politiques de routage contenues dans la Base d'information des politiques (Policy Information Base ou PIB) aux informations de routage présentes dans l'Adj-RIBs-In puis les insère dans la Loc-RIB et ensuite les annonces à d'autres pairs BGP.

Ce processus de décision BGP est conceptuel, les implémentation BGP ne sont pas obligées de respecter à la lettre le modèle proposé par la RFC 4271 tant que l'implémentation supporte les fonctionnalités et les comportements qui y sont décrits.

Le processus de décision BGP a trois responsabilités :

- Sélectionner des routes qui seront utilisées localement par le routeur.
- Sélectionner des routes qui seront annoncées a d'autres pairs BGP.
- L'agrégation des routes et la réduction de l'information de routage.

Ce processus de sélection passe par trois phases :

- Phase 1 : Le calcul du degré de préférence

La phase 1 est invoquée à chaque fois que le routeur reçoit un message UPDATE qui contient une nouvelle route, un changement de route ou un retrait de route.

La phase 1 verrouille la Adj-RIBs-In avant d'effectuer toute opération dessus. Et la déverrouille une fois toutes les routes présentes traitées.

Pour chaque route changée ou nouvellement reçue le routeur local doit déterminer un degré de préférence comme suit :

- Si la route a été apprise d'un pair BGP interne, alors soit la valeur incluse dans l'attribut LOCAL_PREF est directement utilisé comme degré de préférence, ou le routeur local peut calculer le degré de préférence pour la route on se basant sur des politiques préconfigurées (il est très déconseillé de recalculer le degré de préférence car cela peut causer de boucles de routage).
- Si la route a été apprise d'un pair BGP externe, alors le routeur local doit calculer le degré de préférence en se basant sur les politiques préconfigurées. Si la valeur obtenue indique que la route est inéligible pour être sélectionnée alors la route ne sera pas utilisée pour la seconde phase. Sinon la valeur sera utilisée comme attribut LOCAL_PREF dans toute annonces IBGP.

- Phase 2 : La sélection des routes

La phase 2 est invoquée à la complétion de la phase 1.

Le processus de la phase 2 prend en considération toutes les routes éligibles de l'Adj-RIBs-In.

La phase 2 verrouille l'Adj-RIBs-In avant de commencer sa fonction et la déverrouille une fois achevée.

Si l'attribut NEXT_HOP d'une route éligible est une adresse qui n'est pas accessible, ou qui peut devenir inaccessible si la route est installée dans la table de routage, le processus BGP doit exclure cette route de la fonction de décision de la phase 2.

Si l'attribut AS_PATH d'une route éligible contient une boucle AS (le numéro d'AS local a été retrouvé dans l'attribut AS_PATH), la route doit être exclue de la fonction de décision de la phase 2.

Pour toutes destinations pour lesquelles plusieurs routes praticables existent dans la Adj-RIBs-In, le routeur BGP local doit identifier grâce à une suite de règles comparatives la meilleure route vers cette destination. La RFC 4271 contient la version basique de cette suite de règles comparatives, il en existe plusieurs implémentations différentes. La prochaine section présente la version qui sera utilisé au cours de notre étude.

Une fois les routes sélectionnées elles seront installées dans la Loc-RIB, toute route déjà existante vers la même destination qu'une route nouvellement sélectionnée est remplacée.

L'adresse du prochain saut pour la route sélectionnée doit être déterminée depuis l'attribut NEXT_HOP. Si le prochain saut ou son coût selon l'IGP changent, alors il devient nécessaire de refaire la phase 2.

Les routes inaccessibles doivent être supprimées de la Loc-RIB et de la table de routage, mais doivent être gardées dans l'Adj-RIBs-In dans le cas où elles redeviennent accessibles de nouveau.

- Phase 3 dissémination de routes :

La phase 3 est invoquée à la complétion de la phase 2, ou quand un des événements suivants se produit :

a) Quand des routes dans la Loc-RIB pour des destinations locales changent.
b) Quand des routes générées localement et qui ont été apprises par d'autres moyens que BGP changent
c) Quand une nouvelle session BGP est établie.

La phase 3 ne peut pas démarrer si la phase 2 est toujours en cours.

Les agrégations des routes et les techniques de réduction de l'information de routage sont appliquées lors de cette phase.

a. La route la plus spécifique avant toute chose

Il est très important de noter qu'en dépit du degré de préférence les routeurs choisissent toujours la route la plus spécifique vers une destination.

Quand un routeur BGP possède plusieurs routes également spécifiques vers une destination avec le même degré de préférence, des règles sont nécessaires afin de les départager pour sélectionner une route parmi ces routes.

B. La phase 2 selon les implémentations Cisco

Le processus de décision vu au-dessus est comme mentionné précédemment conceptuel. Les développeurs des implémentations BGP sont libres dans leur façon de faire tant que les fonctions et comportements décrits dans le modèle conceptuel sont implémentés dans les normes.

Les règles de sélection varient beaucoup selon l'implémentation de BGP sur le matériel et aussi selon la version de l'implémentation. Pour ce projet, la dernière implémentation de ce processus de sélection de route sur les routeurs 'Cisco IOS' a été utilisée.

Le processus de sélection des routes BGP qui correspond à la phase 2, est une suite de règles comparatives, pour départager les routes et définir la meilleure route en comparant les attributs des routes du plus important au moins important, en cas d'égalité des attributs le processus passe à la règle suivante jusqu'à ce qu'une inégalité soit trouvée.

Les règles sont les suivantes :

1. La route avec l'attribut WEIGHT le plus élevé est la préférée.
2. Si plusieurs routes ont le même attribut WEIGHT alors la route avec la valeur de l'attribut LOCAL_PREF la plus élevée est la préférée.
3. Si la valeur de l'attribut LOCAL_PREF est égale, alors les routes qui ont été créés localement sur le routeur ou ont été apprises par un IGP, puis ont été injectées dans BGP sont les préférées

4. Si la valeur de l'attribut LOCAL_PREF est égale, et qu'aucune route n'a été créé localement. La route avec l'attribut AS_PATH le plus court est la préférée.
5. Si l'attribut AS_PATH est de même longueur, alors la route avec le code de l'attribut ORIGIN le plus bas est la préférée.
6. Si le code de l'attribut ORIGIN est le même, alors la route avec la valeur de l'attribut MULTI_EXIT_DISC le plus bas est la préférée.
7. Si la valeur de l'attribut MULTI_EXIT_DISC est la même, alors les routes EBGP sont préférées par rapport aux routes EBGP confédérées, et les routes EBGP confédérées sont préférées par rapport aux routes IBGP.
8. Si les routes sont égales, alors la route avec le cout le plus faible vers le routeur BGP de l'attribut NEXT_HOP est la route préférée.
9. Si les routes sont égales, et que les routes sont du même AS voisin, et que BGP multipath est activé alors installer toutes les routes à cout égal dans la Loc-RIB.
10. Si les routes sont égales et sont externes, alors la route qui a été reçu en premier est la préférée. Cela permet d'éviter les changements fréquents. Cette règle peut être ignorée.
11. Si les routes sont égales et que BGP multipath est désactivé, alors la route vers le Router ID le plus bas est préférée, S'il s'agit d'une réflexion de route alors la route vers l'ORIGINATOR_ID le plus bas est la préférée.
12. Si les routes sont égales dans le cas des réflexions de route, alors la route avec la CLUSTER_LIST la plus courte est la préférée.
13. Si les routes sont égales, alors la route de l'annonceur avec l'adresse IP la plus basse est la préférée.

(Jeff Doyle, 2001)

7. L'automate d'état fini BGP

La RFC 4271 qui définit la dernière version de BGP propose un automate d'état fini qui aide les pairs BGP à savoir à quelle étape ils en sont pour chaque liaison BGP et leurs dicte quoi faire par la suite.

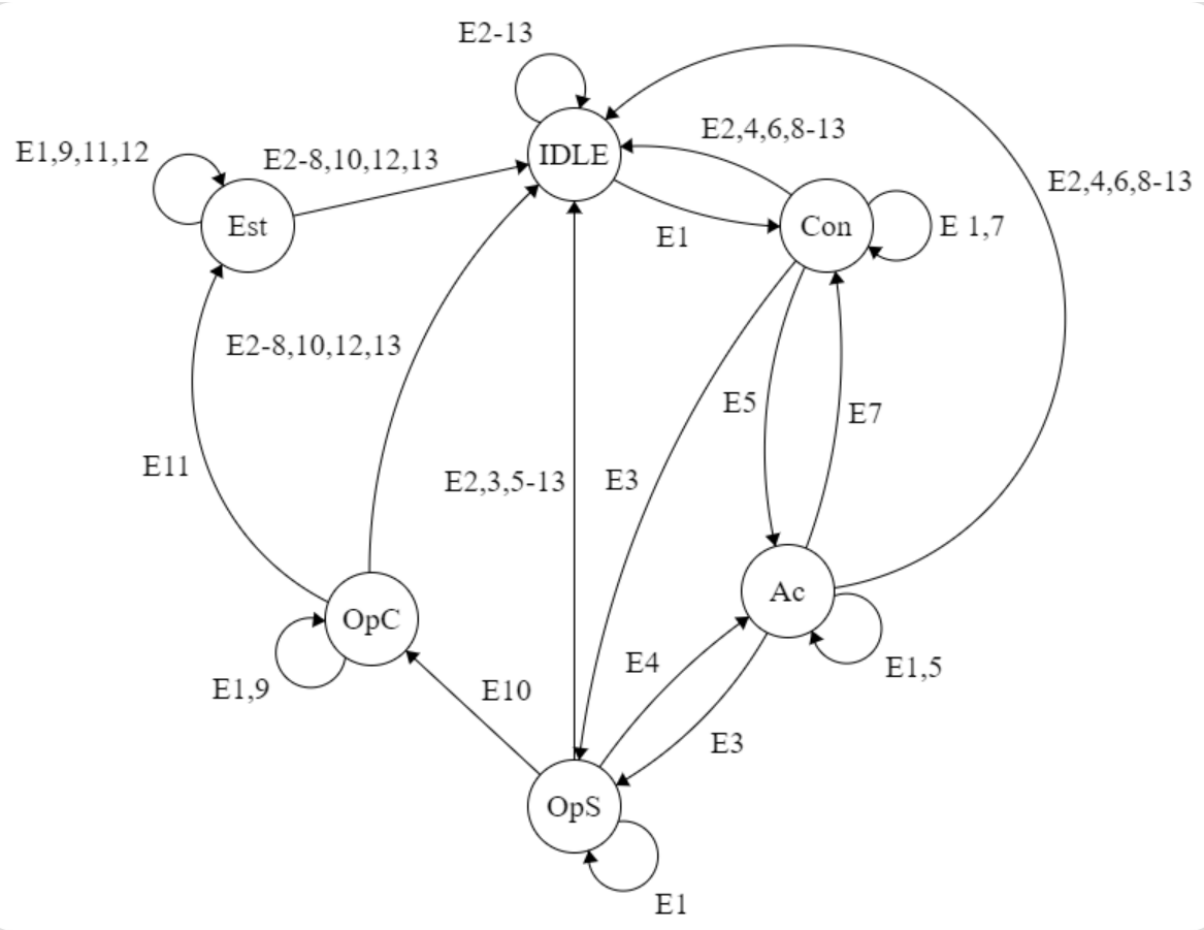

Figure 2-10 Automate d'état fini BGP

Les changements d'état de cet automate sont régis par les événements d'entrées présentés dans le tableau suivant

EVENEMENT D'ENTREE	DESCRIPTION
1	Démarrage de BGP
2	Arrêt de BGP
3	Ouverture de la session transport BGP
4	Session de transport BGP fermé
5	Echec de l'ouverture de la session transport BGP
6	Erreur fatale dans la session de transport BGP
7	Temps du compteur ConnectRetry écoulé
8	Temps du compteur Hold Time écoulé
9	Temps du compteur KEEPALIVE écoulé
10	Réception d'un message OPEN
11	Réception d'un message KEEPALIVE
12	Réception d'un message UPDATE
13	Réception d'un message NOTIFICATION

a. Description des états de l'automate

- L'état IDLE

C'est l'état initial avec lequel BGP démarre. Quand BGP est dans cet état toute tentative de connexion BGP entrante est rejetée, une fois que l'évènement 1 s'est produit BGP initialise toutes ses ressources puis démarre le compteur ConnectRetry et tente d'ouvrir une session TCP avec un pair ou attend que ce pair le fasse puis transite vers l'état connecté.

N'importe quelle erreur ou problème fait transiter BGP de n'importe quel état vers l'état IDLE d'où il tente à nouveau de se reconnecter au pair après le déclenchement de l'évènement 1, mais cependant une limitation est imposée pour éviter que le routeur n'essaye de se reconnecter constamment alors qu'une erreur est persistante, BGP ne peut transiter de l'état IDLE vers l'état connecté que si le temps compteur ConnectRetry est expiré. Le temps du compteur ConnectRetry est à chaque échec de connexion le double du temps précèdent.

- L'état connecté (Con)

Dans cet état BGP ne peut pas initialiser la liaison TCP, il doit attendre que son pair initialise la liaison TCP, Si la connexion se fait avec succès alors BGP efface le compteur ConnectRetry et envoie un message OPEN au pair puis transite vers l'état OpenSent.

BGP continue d'attendre son pair d'initialiser une connexion TCP, si le compteur ConnectRetry expire BGP passe à l'état Actif.

- L'état Actif (Ac)

Dans cet état BGP initialise la liaison TCP au pair. Si une session est établie avec succès alors BGP efface le compteur ConnectRetry, envoie un message OPEN au voisin puis transite vers l'état OpenSent.

Si aucune session TCP initialisée par le voisin n'est établie avec succès pendant le temps du compteur ConnectRetry, à la fin de ce dernier, BGP transite à nouveau vers l'état Connecté relance le compteur ConnectRetry.

- L'état OpenSent (OpS)

Après que le processus BGP a envoyé un message OPEN au routeur pair, il attend de recevoir un message OPEN de ce dernier. Quand le message OPEN est reçu tous ses champs sont vérifiés.

S'il y a un problème alors un message NOTIFICATION est envoyé et BGP transite vers l'état IDLE.

Si aucun problème n'est trouvé et que tout est conforme dans le message OPEN reçu alors un message KEEPALIVE est envoyé et les compteurs Keepalive et Hold Time sont initialisés selon ce qui est défini dans le message OPEN. BGP transite vers l'état OpenConfirm

Si la session BGP est fermée ce qui correspond à l'évènement 4 alors BGP termine la connexion avec le pair relance le compteur ConnectRetry et transite vers l'état Actif.

Tout autre évènement renvoi BGP vers l'état IDLE

- L'état OpenConfirm (OpC)

Dans cet état BGP attend la réception d'un message KEEPALIVE ou un message NOTIFICATION. Si un message KEEPALIVE est reçu BGP transite vers l'état Établie. Si un message NOTIFICATION est reçu BGP transite vers l'état IDLE

Si la session TCP est fermée ou le temps du compteur Hold Time s'écoule, ou qu'un autre évènement se produit alors il transitionne vers l'état IDLE.

- L'état Établie (Est)

Dans cet état la connexion entre la paire de routeur est totalement établie, les deux routeurs peuvent commencer à échanger des messages UPDATE.

Si BGP ne reçoit pas de message KEEPALIVE ou UPDATE et que le temps du compteur Hold Time s'écoule BGP transite vers l'état IDLE. A chaque fois qu'un message KEEPALIVE ou UPDATE est reçu le compteur est réinitialisé.

Tout évènement autre que les évènements 1, 9, 11 ,12 forcent BGP à envoyer un message NOTIFICATION et la transition vers l'état IDLE.

(Jeff Doyle, 2001)

8. Les spécificités d'IBGP

Le comportement de BGP change selon le système autonome du pair avec lequel il interagit. Une relation BGP interne est une relation entre deux routeurs BGP qui appartiennent au même système autonome, alors qu'une relation BGP externe est une relation entre deux routeurs qui appartient à deux différents systèmes autonomes.

Il y a des différences entre IBGP et EBGP qui sont subtiles comme par exemple :

- Distance administrative, pour les routeurs Cisco EBGP à une distance administrative de 20 alors que EBGP à une distance administrative de 200
- Distance entre deux pairs : par default deux pairs EBGP doivent être directement connectés l'un à l'autre, alors que les pairs IBGP peuvent être à plusieurs sauts de distance.
- Quand EBGP annonce une route BGP il change l'attribut NEXT_HOP par sa propre adresse IP, alors que IBGP ne change pas le contenu de l'attribut NEXT_HOP.
- Quand un routeur annonce une route avec IBGP il doit obligatoirement ajouter l'attribut LOCAL_PREF, alors qu'une annonce EBGP ne doit pas contenir l'attribut LOCAL_PREF.

Il y a une différence plus pertinente pour les ingénieurs et architectes réseaux qui affecte la façon de concevoir les réseaux internes. Quand IBGP annonce une route il n'ajoute pas son numéro d'AS dans l'attribut AS_PATH contrairement à EBGP. Le seul moyen dont BGP dispose pour détecter les boucles de routage est l'attribut AS_PATH, s'il retrouve son propre numéro d'AS dans l'attribut AS_PATH alors il sait qu'une boucle s'est produite. Ajouter son numéro d'AS pour annoncer une route à un pair interne n'a aucun sens car ça déclencherait le mécanisme de prévention de boucles de routage.

A. Exemple de boucles de routage IBGP

L'attribut AS_PATH n'a aucun sens à l'intérieur d'un même système autonome, donc IBGP n'a aucun mécanisme de prévention de boucles de routage.

L'exemple suivant montre comment se forme les boucles de routage avec IBGP

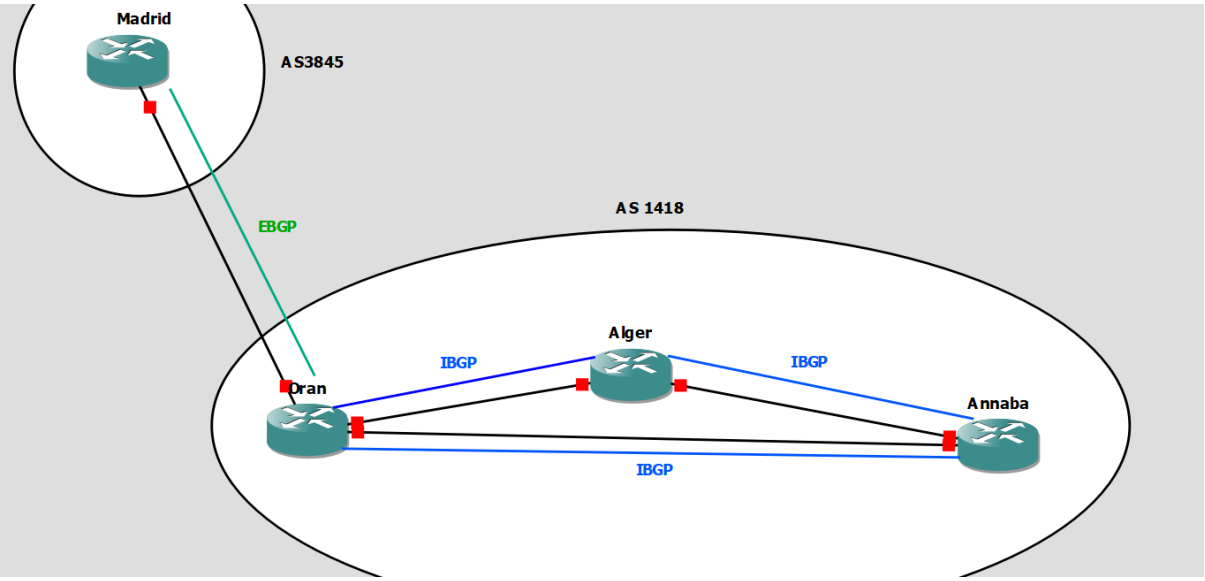

Figure 2-11 formation d'une boucle de routage

La figure au-dessus illustre deux systèmes autonomes reliés logiquement par EBGP.

Le système autonomes 1418 contient 3 routeurs tous physiquement reliés les uns aux autres, et logiquement reliés avec IBGP.

Dans cet exemple Madrid va annoncer une route vers une destination X à Oran via EBGP. Une fois la route reçue par Oran est traitée et ajoutée à sa table de routage il va l'annoncer à Alger via IBGP, Alger à son tour traite la route et l'ajoute dans sa table de routage et puis l'annonce a son tour.

En ce qui concerne Alger annonçant la route vers X, il va l'annoncer seulement à Annaba, parce que le mécanisme Split Horizon empêche les routeurs d'annoncer une route par la même interface par laquelle ils l'ont apprise.

Une fois qu'Annaba ait reçu l'annonce d'Alger il ajoute aussi la route dans sa table de routage puis l'annonce à Oran.

La même chose se passe aussi dans l'autre sens où Alger annonce à Oran la route vers la destination X reçue d'Annaba.

Au final Oran va avoir 3 routes vers la destination X, une apprise via EBGP passant par Madrid, et deux apprise via IBGP passant par Alger et Annaba.

La boucle de routage survient si la connexion physique entre Oran est Madrid est interrompue. Alors les paquets vers la destination X vont circuler indéfiniment dans la boucle Oran, Alger, Annaba jusqu'à ce que les routes IBGP soient retirées.

B. La prévention des boucles de routage avec IBGP

Pour régler le problème des boucles de routage avec IBGP, la solution consiste en une simple règle :

- Un routeur BGP ne doit pas annoncer à un routeur interne (via IBGP) une route apprise d'un routeur interne (via IBGP), sauf si explicitement configuré pour le faire.

Cette solution règle de manière définitive le problème des boucles de routage en limitant la propagation des routes dans le réseau interne. Ainsi laissant la responsabilité de propager chaque route externe seulement au routeur qui l'a apprise via EBGP.

C. Les trous noirs dans les chemins de transfert à travers le réseau interne

La règle précédente règle le problème des boucles de routage mais introduit un nouveau problème, celui des trous noirs ou Blackhole.

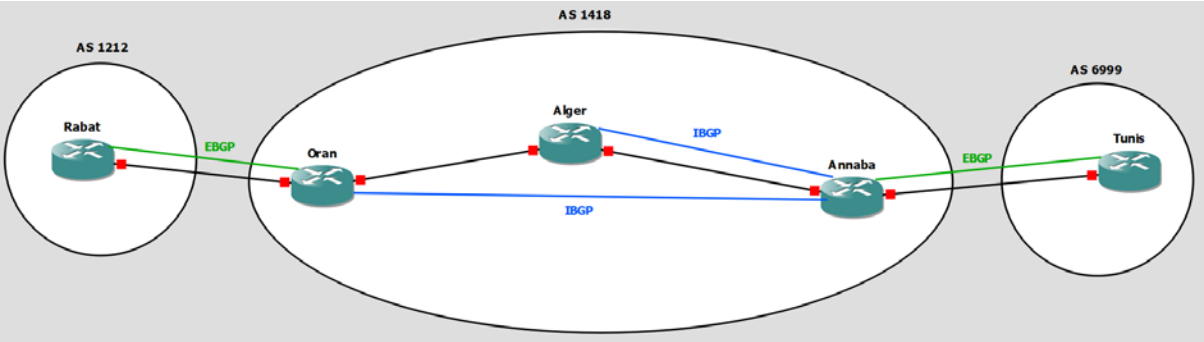

Figure 2-12 Formation d'un blackhole

Dans la figure 2-12 le système autonome 1418 est un AS de transit entre les AS 1212 et 6999.

Le routeurs Rabat et Tunis annoncent au système autonomes 1418 les routes vers leur propre système autonomes via EBGP.

Les routeurs Oran et Annaba, Annaba et Alger s'échangent les routes qu'ils ont apprises par EBGP via IBGP.

Le routeur Oran annonce à Rabat les routes du AS 1418 injectées localement et les routes de l'AS 6999 apprises par les annonces IBGP d'Annaba.

Le routeur Annaba annonce à Tunis les routes du AS 1418 injectées localement et les routes de l'AS 1212 apprises par les annonces IBGP d'Oran.

A ce stade Rabat connait les routes vers l'AS 6999 et Tunis connait les routes vers l'AS 1212, ils peuvent désormais commencer à s'échanger des paquets.

Cependant il y a un problème. Il n'y a aucune liaison directe d'Oran vers Annaba, donc le trafic entre l'AS 1212 et l'AS 6999 doit obligatoirement passer à travers Alger.

Alger connait les routes locales du système autonome 1418 et les routes du système autonome 6999 apprise d'Annaba via IBGP. Cependant il n'a aucune route vers l'AS 1212, son seul pair BGP qui est Annaba connait ces routes vers l'AS 1212 mais n'est pas autorisé a les lui annoncées.

Quand un routeur reçoit un paquet pour une destination vers laquelle il ne possède aucune route, ce paquet est immédiatement détruit. Dans le cas de la figure 2-12 Alger possède des routes vers l'AS 6999 mais pas de route vers l'AS 1212, par conséquent les paquets vers l'AS 6999 seront acheminés sans aucun soucis, alors que les paquets vers l'AS 1212 seront détruits en chemin par Alger. Ce qui constitue un trou noir qui absorbe tous les paquets à destination de l'AS 1212.

Pour résoudre ce nouveau problème crée par la règle précédente, une nouvelle règle a dû être ajoutée cette nouvelle règle stipule :

- Chaque routeur BGP à travers un chemin d'acheminement doit absolument connaitre les routes BGP utilisées par les routeurs qui ont des liaisons externes.

D. La gestion des préfixes avec IBGP

Pour faire connaitre les routes externes aux routeurs internes participants à l'acheminements des paquets externes il existe plusieurs solutions.

Une solution assez simple consiste à créer une topologie IBGP totalement maillée. Chaque routeur participant dans l'acheminement de paquets externes doit avoir une relation IBGP avec tous les autres routeurs participant dans l'acheminement de paquets externes.

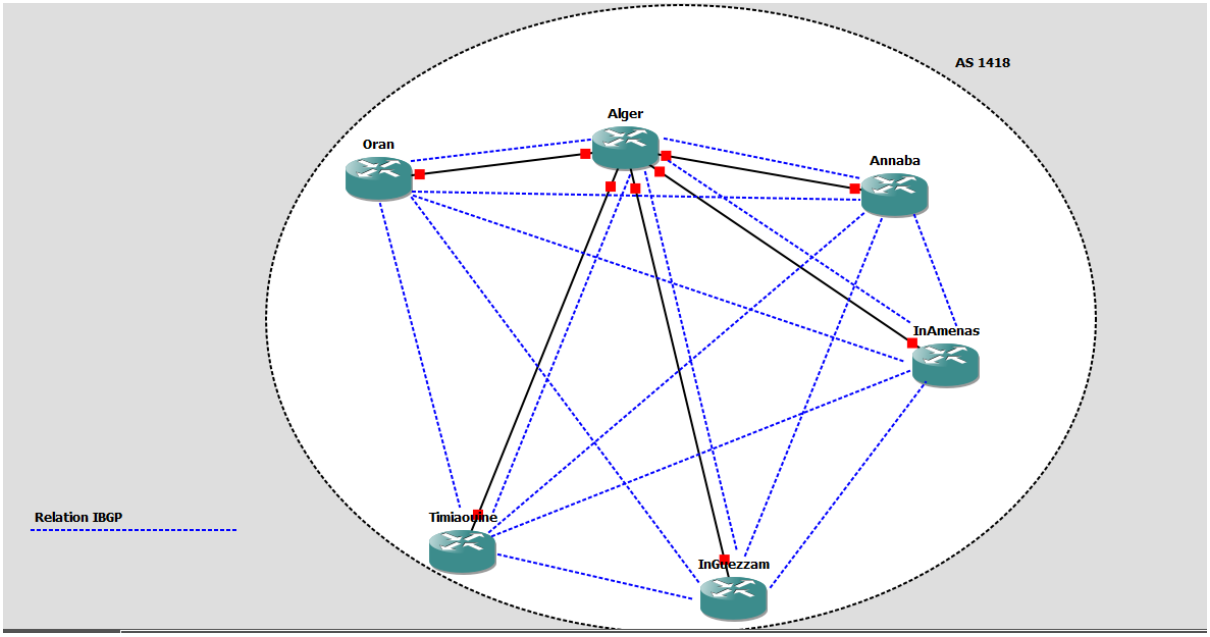

Figure 2-13 Topologie IBGP totalement maillée

La figure 2-13 représente 6 routeurs appartenant au même système autonome. Chaque routeur est physiquement relié à Alger et uniquement Alger.

Les lignes pointillées blues représente une connexion IBGP entre deux routeurs. Chaque routeur possède une connexion IBGP vers tous les autres routeurs du système autonomes.

Il est important de noter que cette topologie maillée est logique et non pas physique.

Cette topologie répond très bien à la deuxième règle. Chaque routeur peut annoncer ses préfixes externes aux autres routeurs. Ainsi les préfixes présents dans tous les routeurs de l'AS seront identiques.

E. La redistribution des routes BGP et la synchronisation BGP

Il existe une autre solution pour éviter la formation de trous noir ou encore l'annonce d'un trou noir à un pair externe.

Cette solution est la règle de synchronisation, elle stipule :

- Avant qu'une route reçue par IBGP ne soit ajoutée à la table de routage ou annoncée à un pair externe, cette route doit d'abord être connue par un IGP.

Pour étudier la synchronisation nous allons utiliser l'exemple de la figure 2-12.

Le routeur Oran a appris les préfixes de l'AS 1212, puis les a annoncés à Annaba, Annaba à son tour a annoncé ces préfixes à Tunis. Le seul chemin pour Annaba et l'AS 6999 pour atteindre l'AS 1212 est de passer par Alger mais ce dernier ne possède pas de route vers ces préfixes, ce qui va constituer un trou noir

Si on active la synchronisation BGP, lorsque Oran reçoit l'annonce de Rabat pour les préfixes de l'AS 1212, il les ajoute à sa table de routage. Oran va annoncer les préfixes de l'AS 1212 à Annaba, avant d'ajouter les préfixes Annaba, il va d'abord vérifier sa table de routage pour voir si ces préfixes sont déjà connus par un IGP, pour le moment ce n'est pas le cas. Annaba n'ajoutera pas les préfixes de l'AS 1212 à sa table de routage et ne les annoncera pas à Tunis. Ceci empêche la formation du trou noir.

Cependant la synchronisation seule n'est pas une solution complète. Il existe bel et bien un chemin d'Annaba vers l'AS 1212 qui passe par Alger mais ce chemin ne sera pas exploité.

Pour pouvoir utiliser le chemin existant d'Annaba vers l'AS 1212 il faudrait faire connaitre à un IGP les préfixes externes de l'AS 1212, pour cela on utilise la redistribution de route.

La redistribution consiste à injecter des routes d'un protocole de routage dans un autre protocole de routage, dans notre cas de BGP vers un IGP (IS-IS, OSPF, EIGRP, RIP).

Une fois les routes externes distribuées dans l'IGP, ce dernier se chargera de propager ces routes vers tous les routeurs internes utilisant cet IGP notamment Alger.

Après la redistribution, Annaba pourra ajouter les préfixes de l'AS 1212 à sa table de routage et les annoncer à Tunis. Alger pourra acheminer le trafic qui a comme destination l'AS 1212.

Bien que cela fonctionne il est à noter que la redistribution de BGP vers un IGP est une très mauvaise pratique. BGP et les IGPs ont été conçu avec différentes philosophies. Un AS voisin est sous une autre administration, BGP assume qu'un voisin externe n'est pas digne de confiance et les informations qu'il lui envoi doivent être régulées et soumises à des politiques de routage, alors que les IGPs sont sensés n'interagir qu'avec des routeurs internes sous la même administration qu'eux et ne possèdent donc pas d'outils pour réguler et contrôler les informations reçues.

Un IGP n'es pas conçu pour contenir le très grand nombre de routes que peut contenir BGP. Dans les larges réseaux cela provoquerait des pertes de performance, instabilités dans le réseau et même des crashs dû à la surcharge de l'IGP.

La topologie IBGP maillée est la solution conseillée, il est aussi conseillé de désactiver la synchronisation BGP et de ne jamais redistribuer BGP dans un IGP.

F. Problèmes rencontrés dans les larges réseaux internes

Pour assurer l'acheminement de paquet à destination externes il faut que chaque routeur sur le chemin du paquet connaisse les destinations externes. Pour ce faire, la meilleure solution pour le moment est la topologie IBGP totalement maillée.

a. Le très grand nombre de relations IBGP

À petite échelles, créer une topologie IBGP totalement maillée est assez simple. Mais les réseaux de transite ne sont jamais des petits réseaux, c'est souvent de très larges entreprises d'acheminement.

Sur un réseau interne qui possède 5,000 routeurs sur les voies d'acheminements, il faudrait gérer 12,497,500 relations IBGP nécessaires à la formation de la topologie maillée. Ce qui est un travail surhumain.

Pour calculer le nombre de relations IBGP nécessaires à un réseau de n routeur on utilise la formule suivante : $\mathbf{relations} = \frac{1}{2}(n^2 - n)$

Aucune des deux solutions vues précédemment ne convient à un très large réseau d'acheminent de paquet.

De nouvelles solutions ont dû être inventées pour résoudre le problème de propagation des routes externes dans les très large réseau internes.

Parmi ces solutions on compte Les confédérations, Les Routes Reflectors et MPLS.

Les confédérations devisent l'AS en plusieurs plus petits sous-systèmes autonomes.

Les Route Reflectors enfreignent d'une manière très ingénieuse la règle de propagation des routes IBGP.

MPLS libère les routeur internes de BGP, il devient nécessaire d'utiliser BGP seulement sur les routeurs externes.

b. Les confédérations

Une confédération est un système autonome avec un très grand nombre de routeurs BGP divisé en domaines plus petits dans le but de faciliter la gestion de ce système autonome. Une confédération est donc une collection de système autonomes représentés et annoncés comme un seul système autonome aux pairs BGP qui ne sont pas membre de la confédération. (P. Traina, 2007)

Les sous-systèmes autonomes sont aussi appelés AS membre d'une confédération.

Les membre BGP du même sous-système autonome utilisent IBGP pour communiquer entre eux. Pour communiquer avec un membre d'un autre sous-système autonome une version spécial de EBGP est utilisée, appelée ''Confederation EBGP''.

La confédération possède un ID de confédération. Cet ID de confédération est utilisé lors des communications avec des pairs externes à la confédération comme numéro d'AS.

Les pairs externes à la confédération ne doivent pas voir la structure interne de la confédération, la confédération doit être vue de l'extérieur comme un seul AS, pour réaliser cela des numéros d'AS privés doivent être utilisés pour les sous-système autonomes et ces numéros privés ne seront jamais transmis à l'extérieur.

i. Les numéros d'AS privés

Les numéros d'AS privés ont été introduit par la RFC 1930, ces numéros vont de 64512 à 65534. Le but des numéros d'AS privés est d'éviter l'épuisement des numéros d'AS disponibles.

Certains larges réseaux peuvent avoir besoin de plus d'un numéro d'AS pour gérer leur réseau internes. La solution est d'utiliser des numéros d'AS privés qui n'ont de signification que dans les

réseau internes et sont automatiquement retirés de l'attribut AS_PATH lors des annonces vers l'extérieur.

ii. L'épuisement des numéros d'AS disponibles

À l'origine les numéros d'AS sont des entiers non signés sur 2 octets. Avec le développement fulgurant d'Internet, 65 535 numéros d'AS sont devenus insuffisants.

Pour répondre à cette demande de numéros d'AS toujours croissante des numéros d'AS sur 4 octets ont été développés, ce qui fait à présent 4 milliards de numéros d'AS.

Ces numéros d'AS sur 4 octets ont un nouveau format d'écriture pour les distinguer facilement des numéros d'AS sur 2 octets.

Les numéros d'AS à 4 octets commencent là où se termine les numéros d'AS à 2 octets.

	AS sur 2 octets	AS sur 4 octets
Ensemble de numéro	1 - 65 535	65 536 - 4 294 967 295
Le format d'écriture	1 - 65535	1.0 - 65535.65535

iii. Illustration d'une confédération

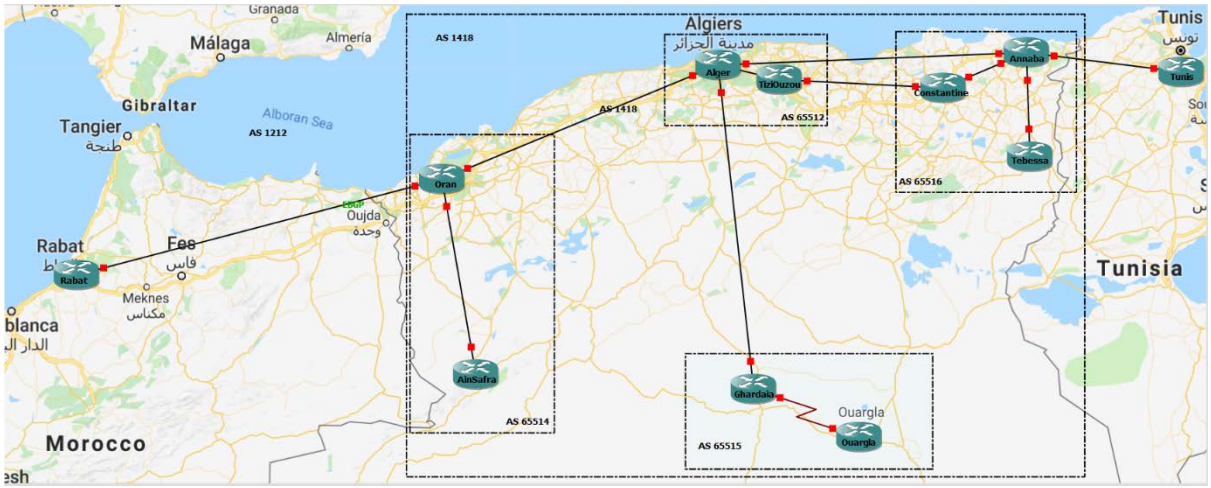

Figure 2-14 Confédération BGP

La figure 2-14 représente une confédération BGP. Le système autonome 1418 a été devisé en 3 sous-systèmes autonomes avec des numéros d'AS privés qui sont appelés numéro de membre-AS.

La topologie IBGP totalement maillée est désormais requise seulement entre les routeurs BGP du même sous-système autonome. Cette confédération requière seulement 6 relations IBGP pour former des topologie IBGP totalement maillée. Sans la confédération il aurait fallu 36 relations IBGP.

iv. Comportement d'un routeur BGP membre d'une confédération (P. Traina, 2007)

Un membre d'une confédération BGP doit utiliser son ID de confédération pour toute interaction avec un pair qui n'est pas membre de la confédération. Cet ID de confédération est le numéro d'AS visible de l'extérieur.

Un membre d'une confédération doit utiliser son numéro de membre-AS pour toute interaction avec un pair membre d'un autre sous-système autonome de la confédération.

Un routeur membre qui reçoit un message UPDATE avec l'attribut AS_PATH qui contient son confédération ID doit traiter ce message comme un message qui contient son numéro d'AS.

Un routeur membre qui reçoit un message UPDATE avec l'attribut AS_PATH qui contient son numéro de membre-AS doit traiter ce message comme un message qui contient son numéro d'AS.

v. Les attributs utilisés dans les confédérations

Les sections précédentes ont introduit l'attribut AS_PATH. L'attribut AS_PATH est un ensemble de vecteurs qui contiennent les numéros des systèmes autonomes par lesquels il faut passer pour atteindre une destination. Ces vecteurs sont de deux types les AS_SEQUENCE et les AS_SET.
Les confédération introduisent deux nouveaux vecteurs dans l'AS_PATH

- AS_CONFED_SEQUENCE : tout comme l'AS_SEQUENCE ce vecteur représente une liste ordonnée de numéros de systèmes autonomes que le chemin traverse vers la destination, à l'exception des numéros d'AS qui appartiennent à la confédération.
- AS_CONFED_SET : tout comme l'AS_SET ce vecteur représente une liste désordonnée de numéros de systèmes autonomes que le chemin traverse vers la destination, à l'exception des numéros d'AS qui appartiennent à la confédération.

L'attribut AS_PATH avec ces deux nouveaux vecteurs est utilisé entre les AS membres à travers les messages UPDATE comme mécanisme de détection des boucles de routage.

Quand un message UPDATE est envoyé à un pair externe à la confédération les vecteurs AS_CONFED_SEQUENCE et AS_CONFED_SET doivent être retirés de l'attribut AS_PATH.

vi. Les spécificités de Confederation EBGP

Confederation EBGP adopte certains comportements de EBGP et IBGP. Voici les comportements spécifiques de Confederation EBGP (P. Traina, 2007):

- L'attribut NEXT_HOP des routes externes est préservé à travers la confédération.
- L'Attribut LOCAL_PREF est préservé à travers toute la confédération
- L'attribut MULTI_EXIT_DESC des routes annoncées à la confédération est préservé à travers toute la confédération
- Les numéros d'AS membres de la confédération sont ajoutés à l'AS_PATH à travers la confédération mais sont retirés pour les pairs externes à la confédération.

vii. L'attribut AS_PATH dans la sélection de route

Les critères de sélection de route pour les informations de routage reçues de la part de membres internes à la confédération doivent suivre les mêmes règles que pour les informations reçues de membres du même système autonomes.

En addition à ces règles voici des règles supplémentaires qui doivent être appliquées lors de la sélection de routes (P. Traina, 2007) :

1. Si l'AS_PATH est interne à la confédération, il faut considérer l'AS voisin comme étant l'AS local.
2. Sinon, si le premier segment de l'AS_PATH n'est ni un AS_CONFED_SEQUENCE ou AS_CONFED_SET, alors considérer le numéro d'AS le plus à gauche dans l'AS_SEQUENCE comme AS voisin.
3. Lors de la comparaison des routes selon la longueur de leur AS_PATH, AS_CONFED_SEQUENCE et AS_CONFED_SET ne doivent pas être comptés.

4. Lors de la comparaison des routes selon le protocole par lequel elles ont été apprise IBGP (interne) ou EBGP (externe), les routes apprises d'un pair de la même confédération sont considérées comme internes.

c. Les Route Reflectors ou RR

Les Routes Reflectors sont une autre solution pour réduire le nombre de liaisons IBGP nécessaires pour propager l'information de routage externe à l'AS vers tous les routeurs de l'AS. Typiquement tous les routeurs BGP doivent être reliés avec une topologie IBGP totalement maillée même dans le cas des confédérations. Les membres d'un même sous-système autonomes doivent être tous reliés les uns aux autres.

Les Routes Reflectors dérogent à la loi qui stipule qu'une route apprise avec IBGP ne doit pas être propagée de nouveau avec IBGP. Et introduisent aussi deux nouveaux attributs 'optional nontransitive' pour prévenir les boucles de routage.

i. Démonstration d'un Route Reflectors

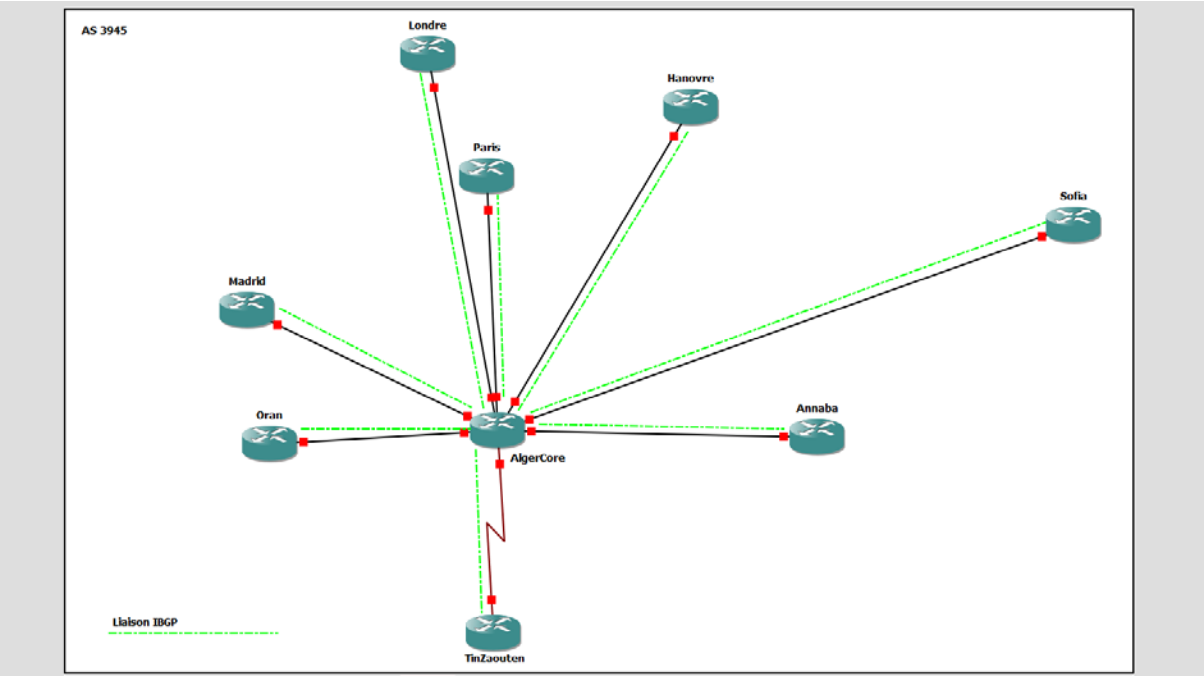

Figure 2-15 Route Reflectors

Le système autonome 3945 possède plusieurs routeurs à travers le monde tous reliés au cœur du réseau situé à Alger. Tous les routeurs ont une liaison IBGP avec 'AlgerCore'. 'AlgerCore' connait toutes les routes externes grâce aux annonces IBGP de ses pairs internes.

On va faire de 'AlgerCore' un Route Reflector, son rôle sera de propager une route apprise via IBGP d'un pair interne vers tous les autres pairs internes à l'exception de celui dont il a appris la route toujours avec IBGP.

Avec la propagation des routes faites par 'AlgerCore' chaque routeur du système autonome connaitra toutes les routes vers les destinations externes.

Il est à noter que seul le Route Reflector est autorisé à propager avec IBGP les routes apprises via IBGP.

ii. Concept des Route Reflectors (RR)

On appelle réflexion de route, l'opération de propagation de route apprises par IBGP vers d'autres pairs IBGP à l'exception du pair à l'origine de la route.

Les pairs internes d'un RR sont divisés en deux catégories (T. Bates E. C., 2006):

- Pairs clients
- Pairs non clients

On appelle l'ensemble du RR et ses pairs clients un cluster.

La propagation de l'information de routage par le RR dépend du type de pair par lequel l'information a été reçue :

- Une route reçue d'un pair non client ne peut être propagée qu'aux pairs clients.
- Une route reçue d'un pair client peut être propagée au pairs clients et non clients.

Il est à noter que les pairs non clients ont toujours besoin d'avoir une topologie totalement maillée avec le RR et tous les autres pairs non clients.

Les pairs clients d'un RR n'ont aucune vision de la réflexion de route, de leur point de vue ils forment une relation IBGP conventionnel avec le RR.

Un système autonome peut avoir plusieurs clusters. Un RR d'un cluster traite un autre RR d'un autre cluster comme n'importe quel autre routeur BGP. Un RR peut avoir un autre RR comme pair client ou non client.

Un même cluster peut avoir plus d'un RR, les RR additionnels restent en stand-by si le RR principal tombe en panne un autre RR prend immédiatement le relai.

Les RR peuvent être utilisés dans les confédérations.

iii. Prévention des boucles de routage

La règle à laquelle le RR déroge a été créé pour empêcher des boucles de routage de se former. Une mauvaise configuration d'un RR peut introduire des boucles de routage dans un cluster. Pour éviter que cela n'arrive la RFC 4456 introduit deux nouveaux attributs (T. Bates E. C., 2006):

- ORIGINATOR_ID : c'est un attribut 'optional nontransitive'. C'est un entier de 4 octets. Il est créé par le RR lors d'une réflexion de route. Cet entier correspond à l'identifiant de l'auteur de la route. L'attribut ORIGINATOR_ID ne doit pas être remplacé s'il existe déjà. Si un routeur reçoit une annonce avec un attribut ORIGINATOR_ID qui correspond à son ID alors il doit ignorer cette annonce.
- CLUSTER_LIST : c'est un attribut 'optional nontransitive'. C'est un vecteur de CLUSTER_ID, il représente les clusters par lesquels la réflexion de l'information de routage est passée. Quand un RR réfléchi une route il doit ajouter son CLUSTER_ID au CLUSTER_LIST. En utilisant cette information un RR peut savoir si une information de routage a tournée en boucle pour revenir vers lui. Si un RR retrouve son CLUSTER_ID dans le CLUSTER_LIST alors il doit ignorer cette annonce.

9. Les politiques de routage

BGP est à la base un protocole d'échange d'informations d'accessibilité entre des entités sous administrations différentes. Ces entités peuvent être des entreprises concurrentes, des gouvernements rivaux…

Réguler, contrôler et filtrer les informations d'accessibilité rentrantes et sortantes est une nécessité absolue.

Une mauvaise manipulation soit volontaire ou involontaire des informations d'accessibilité peut causer de très grands dégâts, au réseau à l'origine de la mauvaise manipulation mais aussi aux autres réseaux avec lesquels il partage des informations d'accessibilité.

Les administrateurs d'un système autonome peuvent aussi avoir des préférences en ce qui concerne l'acheminement de leur trafic, ils peuvent acheminer leur trafic par un AS voisin au lieu d'un autre AS voisin. Ils peuvent aussi vouloir déterminer des règles en ce qui concerne le trafic à destination de leur AS ou transitant par leur AS.

Pour reprendre les termes de Susan HARES un des trois concepteurs de BGP et président actuel du département de routage inter-domaine (Inter-domain routing) responsable du développement de BGP au sein de l'IETF : ''BGP est un protocole de politique avant d'être un protocole de routage, il a été conçu pour échanger des informations de routage entre des administrations qui ne s'entendent pas entre elles'' (Susan Hares, 2017)

BGP est le seul protocole possédant des mécanismes qui permettent de répondre à ces besoins de régulation, de contrôle et filtrage de l'information d'accessibilité. Ces mécanismes sont représentés par un moteur de politique.

Une politique est une ligne de conduite raisonnée. (CNRTL, n.d.)

Une politique est une méthode d'action définie, choisie parmi des alternatives et à la lumière de conditions données pour guider et déterminer des décisions présentes et futures. (Merriam-Webster, n.d.)

En ce qui concerne le routage, une politique de routage est une suite de règles définies par une administration dans le but de contrôler les décisions de routage.

La politique de routage modifie le processus de décision BGP qui par default choisi le chemin optimal d'une façon à assurer le comportement du trafic voulu par la politique de routage.

Une implémentation des fonctions basiques de BGP est très simple alors l'implémentation du moteur de politique de routage est bien plus compliquée. Les solutions BGP open source ont des moteurs de politique de routage très limités, pour avoir un moteur de politique plus complet les administrateurs des réseaux s'orientent vers des solutions commerciales.

Dans le cas de notre étude l'implémentation du moteur de politique BGP sur Cisco IOS sera utilisée.

A. BGP sur Cisco IOS

Le modèle BGP proposé par la RFC 4271 est un modèle conceptuel, les implémentations ne sont pas obligées de suivre à la lettre la RFC tant que l'implémentation reproduit les fonctionnalités et comportements qui y sont décrits.

La RFC 4271 propose 3 bases de données pour stocker les informations de routage, l'Adj-RIBs-In, Loc-RIB et l'Adj-RIBs-Out.

- Les messages UPDATE reçus sont stockés dans l'Adj-RIBs-In. Une fois les routes dans l'Adj-RIBs-In soumis au processus de décision BGP les routes sélectionnées sont placées dans la Loc-RIB.
- Les informations dans la Loc-RIB sont utilisées de deux façon : a) Le processus de sélection de route compare les routes de la Loc-RIB aux routes de la RIB, il ajoute les routes manquantes dans la RIB depuis la Loc-RIB, si une route déjà existante dans la RIB est retrouvée dans la Loc-RIB le processus de sélection place la route avec la distance administrative la plus petite dans la RIB. b) La Loc-RIB envoie à la Adj-RIBs-Out les informations à annoncer.
- L'Adj-RIBs-Out est utilisée pour stocker les informations de routage prêtes à être envoyées dans des messages UPDATE

En théorie tout semble fonctionner parfaitement, mais en pratique il y a redondance de l'information. Les informations contenues dans les trois bases de données sont les mêmes. Sachant que les informations contenues dans les tables BGP d'un routeur de production peuvent facilement atteindre le million de routes. Ceci représente un gouffre en termes de mémoire.

Il n'est pas pratique d'avoir l'Adj-RIBs-In, la Loc-RIB et l'Adj-RIBs-Out comme trois bases de données séparées avec des informations redondantes, Cisco IOS utilise une seule base de données avec des marqueurs pour indiquer où se situe l'information (Adj-RIBs-In, Loc-RIB, Adj-RIBs-Out).

Pour optimiser d'avantage la consommation en mémoire, les informations appartenant à l'Adj-RIBs-In contenues dans la base de données sont stockées d'une manière temporaire est sont effacées après traitement. Pour les messages UPDATE sortants l'Adj-RIBs-Out n'est plus nécessaire, les informations à annoncer sont calculées depuis la Loc-RIB à chaque fois qu'on a besoin.

a. Les Queues BGP et les versions des tables.

Cisco IOS implémente seulement la Loc-RIB. Pour ce qui est de l'Adj-RIBS-In et l'Adj-RIBS-Out il utilise des queues appelées InQ et OutQ. Pour chaque pair BGP deux queues InQ et OutQ lui sont associées. Les queues stockent les routes reçues non-traitées pour la InQ, et les routes à annoncer pour la OutQ. Une route traitée ou annoncée sera retirée de la queue.

Pour que BGP sache quelle route traiter ensuite ou quelle route annoncer, il utilise un compteur appelé 'Table Version'. Ce compteur va aider à garder une trace de ce qui a été reçu ce qui a été traité, et ce qui reste à être annoncer.

On trouve une 'Table Version' dans :

- La RIB
- Chaque pair BGP a son propre compteur
- Le processus BGP
- Chaque préfixe

Au démarrage le processus BGP initialise sa 'Table Version' à 1, et assigne le nombre 1 à la RIB et à tous ses pairs avec lesquels il a une session établie.

À la réception d'un préfixe le compteur 'Table Version' du processus BGP est incrémenté de 1, et le nombre obtenu est aussi assigné au préfixe comme valeur de sa 'Table Version'.

Si un préfixe change, alors le compteur 'Table Version' du processus BGP sera incrémenté de 1, et le nombre obtenu sera assigné de nouveau au préfixe comme sa nouvelle valeur de 'Table Version'.

Si un préfixe est retiré de la table BGP, le compteur 'Table Version' du processus BGP sera incrémenté de 1.

Le processus BGP ajoute, retire et met à jour les préfixes de la RIB dans l'ordre ou ils sont arrivés, retirés ou modifiés. Et à chaque ajout, retrait ou mise à jour d'un préfixe le processus BGP change la valeur du compteur 'Table Version' de la RIB à celui du préfixe ajouté, retiré ou modifié.

Les préfixes sont annoncés aux pairs dans l'ordre croissant selon leur numéro de 'Table Version', à chaque fois que le processus BGP annonce à un pair un préfixe il change la valeur de Table Version de ce pair à celui du préfixe annoncé.

b. Les processus BGP

Les processus BGP sont responsables des informations contenues dans la Loc-RIB. Ils varient selon la version de l'implémentation. Il y a des processus constants dans les implémentations Cisco IOS qui sont (Leahy, n.d.):

- BGP Open : s'exécute lors de l'établissement de la session TCP.
- BGP I/O : gère la lecture, écriture et l'exécution des messages UPDATE et KEEPALIVE.
- BGP Router : Interagie avec la RIB, gère les liaisons avec les pairs et calcule et sélectionne le meilleur chemin.
- BGP Scanner : scanne périodiquement (toutes les 60 secondes par défaut) la RIB pour déterminer si des préfixes et attributs doivent être retirés. Il scanne aussi la table BGP pour vérifier si les adresses NEXT_HOP sont toujours accessibles.

Les processus BGP I/O, Router et Scanner sont responsable du traitement des messages UPDATE et du maintien des tables de routage.

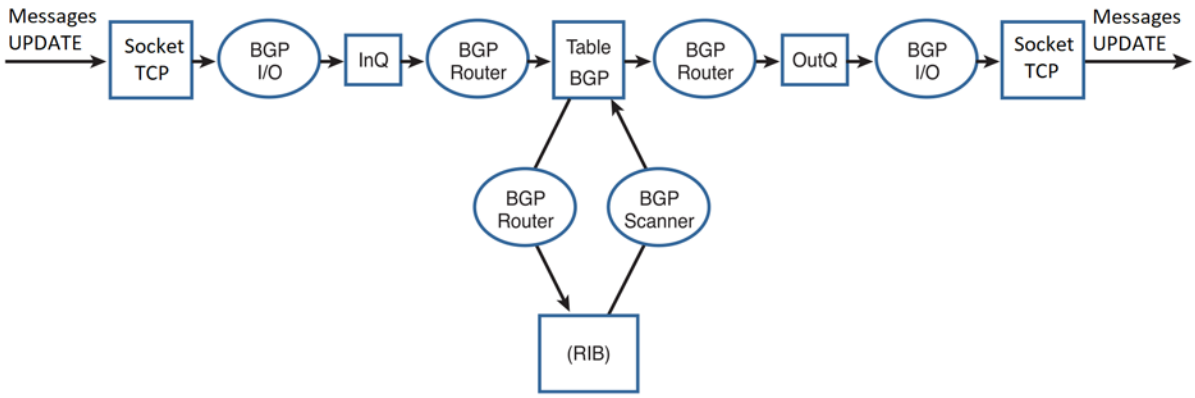

Figure 2-16 traitement d'un message UPDATE à travers l'implémentation Cisco IOS de BGP

La figure 2-16 schématise dans un ordre séquentiel les étapes par lesquelles le message UPDATE doit passer (Smith, 2003)

- Un message UPDATE est reçu sur le socket TCP correspondant à une liaison BGP avec un pair, puis il est stocké dans un buffer.
- Le message UPDATE dans le buffer est lu par le processus BGP I/O pour extraire l'information de routage. L'information de routage est ensuite placée dans l'InQ du pair approprié.

- Le processus BGP Router applique les politiques de routage au routes présentes dans l'InQ avant de recalculer le meilleur chemin vers chaque destination.
- Après le calcul des meilleurs chemins le processus BGP Router insert toutes les routes dans la Loc-RIB (table BGP) y compris les routes les moins optimales.
- Le processus BGP Router insert les routes sélectionnées comme meilleurs chemins de la Loc-RIB dans la RIB. Le processus BGP Router modifie ou retire une route de la RIB si elle a été retirée ou modifiée dans la Loc-RIB.
- Le processus BGP Scanner surveille tout changement qui peut affecter les informations que BGP stocke dans la Loc-RIB. Par exemple le changement de l'état d'accessibilité du NEXT_HOP qui peut rendre une route dans la Loc-RIB invalide.
- Le processus BGP Scanner se charge aussi de placer les routes injectées localement par BGP de la RIB dans la Loc-RIB.
- Le processus BGP Router applique les politiques de routage aux routes à annoncer présentes dans la Loc-RIB avant de les placer dans la OutQ de chaque pair à qui annoncer la route.
- Le processus BGP I/O écrit des messages UPDATE pour contenir chaque information de routage présente dans l'OutQ d'un pair puis place les messages dans les buffers TCP du socket d'un pair pour être envoyé à ce pair.

Cisco IOS applique les politiques de routage aux informations d'accessibilité présentes dans les queues InQ avant qu'elles soient utilisées dans le processus de sélection de chemin puis les applique de nouveau aux informations d'accessibilité présentes dans la Loc-RIB avant de les annoncer à ses pairs.

c. Les outils de la politique de routage

Cisco IOS implémente des outils pour exprimer et appliquer les politiques de routage :

i. Outils de filtrage

Les outils de filtrage permettent de filtrer les informations d'accessibilité qui peuvent passer ou doivent être bloquer. Il existe plusieurs techniques de filtrage qui différent dans les critères de filtrage.

(1). Filtrage par NLRI

On peut filtrer des informations d'accessibilité en se basant sur le préfixe des routes contenu dans les informations d'accessibilité. Cette méthode de filtrage offre beaucoup de précision, elle est la plus efficace quand il y a un petit nombre de route à filtrer. Cisco IOS fourni deux outils pour filtrer sur la base des NLRI.

- Distribute Lists :

C'est l'outil le plus simple à utiliser, un filtre est défini par une liste d'accès contenant une suite de préfixes et leur longueur qui seront autorisés ou bloqués, à la fin de la liste d'accès tout préfixe qui n'a pas été explicitement mentionné sera automatiquement bloqué.

La liste d'accès est ensuite appliquée à un pair dans une direction entrante ou sortante.

- Prefix Lists

Les Prefix Lists sont très similaires au Distribute Lists, ils offrent les mêmes fonctionnalités de filtrage, cependant les Prefix Lists on quelques avantages en termes de facilité d'écriture et de lecture pour l'opérateur humain par rapport aux Distribute Lists.

En plus de la facilité d'interprétation les Prefix Lists sont plus optimisés que les Distribute List, ils requièrent moins de cycle CPU et ont donc de meilleures performances.

Il est à noter que les Distribute List et les Prefix Lists sont mutuellement exclusives. On ne peut pas utiliser une Distribute List et une Prefix List en même temps pour un pair sur la même direction.

(2). Filtrage par AS_PATH

BGP possède une perspective globale des routes avec l'attribut AS_PATH. L'attribut AS_PATH nous permet de voir les systèmes autonomes qui constitue la route vers la destination.

En pratique il est plus sensé d'appliquer des politiques de filtrage en se basant sur les AS qu'une route doit traverser que d'appliquer les politiques de filtrage en se basant sur un préfixe individuel.

Cisco IOS offre un filtre AS-Path qui est un outil d'identification de route BGP par rapport à leur attribut AS_PATH. Les filtre AS-Path fonctionnent avec les expressions régulières.

Les expressions régulières permettent de faire correspondre un modèle défini avec l'expression régulière au contenu de l'AS_PATH.

Une fois la correspondance faite le filtre AS-Path peut autoriser ou bloquer l'information d'accessibilité selon la configuration.

ii. Outils de modification Route Map

Cisco IOS propose un outil très flexible et précis pour exprimer les politiques de routage les plus complexes. Cet outil nommé Route Map peut utiliser les outils de filtrage en combinaison les uns avec les autres, pour faire correspondre des routes par rapport à leur préfixes et leur AS_PATH au même temps.

Les Routes Map peuvent utiliser plusieurs attributs contenus dans les informations d'accessibilité pour vérifier la correspondance d'une route à un modèle défini.

Une fois la correspondance d'une route faite, une Route Map peut autoriser ou bloquer la route, mais peut aussi modifier ses attributs.

Modifier les attributs d'une route peut être très dangereux, particulièrement si on annonce ces routes modifiées à d'autres pairs. Cela peut engendrer des inconsistances dans le routage.

Les administrations qui s'échangent des informations de routage avec BGP ne se font souvent pas confiance, avec la possibilité d'altérer les informations de routage cela rend encore plus difficile les relations entre ces systèmes autonomes. Les administrateurs des systèmes autonomes doivent se mettre d'accord sur les informations de routage à échanger avant la formation d'une liaison BGP. Une fois la liaison établie, toute information envoyée qui ne correspond pas aux accords initiaux doit être bloquée. C'est le rôle des politiques de routage de définir quelles informations peuvent être autorisées et quelles informations doivent être bloquées.

d. Changement dans la politique de routage

BGP a été conçu pour gérer des bases de données de plusieurs centaines de milliers de routes. Pour être efficace BGP communique le moins possible. BGP ne communique que s'il y a un changement dans la topologie du réseau.

Cisco IOS utilise des queues temporaires pour les messages UPDATE entrants. Une fois un message UPDATE traité il est retiré définitivement de la queue. Un problème survient avec ce modèle de fonctionnement.

Les politiques de routage sont appliquées aux informations présentes dans la InQ avant d'être insérées dans la Loc-RIB, puis sont effacées. Si un changement de la politique de routage est effectué alors Cisco IOS n'a plus aucun moyen d'appliquer les nouvelles politiques de routage aux informations d'accessibilité déjà présentes dans la Loc-RIB car leur message UPDATE a été effacé. Les nouvelles politiques de routage n'affecteront que les futurs messages UPDATE et pas à ceux qui ont précédé la nouvelle politique de routage.

Pour faire appliquer les nouvelles politiques de routage aux précédents messages UPDATE il existe des solutions :

i. La réinitialisation des sessions BGP

Le processus BGP redémarre ses sessions BGP avec ses pairs :

Pour ce faire le processus ferme les sessions BGP avec ses pairs afin de retirer les routes apprises de ces pairs. Puis ouvre de nouvelles sessions avec ces mêmes pairs. Une fois une nouvelle session établie avec un pair, les deux pairs commencent à s'échanger des messages UPDATE, ainsi les nouvelles politiques de routage vont être appliquées à ces informations d'accessibilité.

La réinitialisation des sessions BGP peut avoir un effet boule de neige sur les réseaux très larges. Ça peut prendre plusieurs minutes pour propager les messages de retraits puis encore des messages d'annonce à travers tout le réseau.

L'usage de cette méthode est très peu recommandé et peut avoir des conséquences désastreuses, sur un réseau de production ou le temps de convergence doit être de l'ordre de la milliseconde une coupure de plusieurs dizaines de minute serait une catastrophe financière.

ii. La reconfiguration-SOFT

Pour éviter les problèmes engendrés par une réinitialisation des sessions BGP une meilleure technique a été développée. Cette technique consiste à garder les informations d'accessibilité reçues des pairs pour les réévaluer en cas de changement de la politique de routage.

Pour garder les informations de routage de manière permanente on réintroduit l'Adj-RIBs-In qui va stocker toutes les informations d'accessibilité comme elles ont été reçues des pairs.

Si un changement dans la politique de routage est effectué, le routeur peut réévaluer les informations contenues dans l'Adj-RIBs-In avec les nouvelles politiques de routages sans avoir besoin de perturber les sessions BGP déjà établies avec les pairs.

Il y a un prix à payer pour activer la reconfiguration-soft, les bases données Adj-RIBs-In et Adj-RIBs-Out ont été enlevées pour économiser de la mémoire. Réintroduire l'Adj-RIBs-In va augmenter significativement la mémoire nécessaire au fonctionnement de BGP. C'est pour cette raison que beaucoup de routeur avec une mémoire limitée ne sont pas capable d'utiliser la reconfiguration-soft.

iii. Route Refresh

La reconfiguration-soft est une solution assez couteuse en termes de mémoire, c'est pour cela qu'une autre solution à dû être mise en place.

Le but initial de la réinitialisation des sessions ou de la reconfiguration-soft est de faire appliquer des nouvelles politiques de routage à des informations d'accessibilité qui n'ont pas été modifiées ou traitées par de précédentes politiques de routage.

Au lieu de stocker toutes les informations reçues au détriment de la mémoire du routeur ou encore fermer toutes les sessions BGP au risque d'avoir de longues coupures de connexion dans le réseau il existe une solution très simple. Le standard initial proposé par la RFC 4271 n'avait pas prévu la suppression de l'Adj-RIBs-In et Adj-RIBs-Out des implémentations BGP, c'est pour cela qu'il a fallu développer un mécanisme complémentaire pour régler ce problème.

La RFC 2918 propose une solution alternative à la reconfiguration-soft qui supprimerait les couts additionnels en mémoire. La solution consiste à permettre à un routeur BGP de demander à un pair de lui réannoncer des informations d'accessibilité déjà annoncées.

La RFC 2918 ajoute une nouvelle capacité qui est le 'Route Refresh' un routeur capable de supporter cette capacité doit l'indiquer dans le message OPEN lors de l'établissement de la session BGP. Un routeur ne peut demander à un pair de lui réannoncer des routes que s'il supporte cette capacité.

La demande de rediffusion d'information de routage est un message BGP de type ROUTE REFRESH, le message ROUTE REFRESH ajoute à l'entête BGP les champs additionnels suivants (T. Bates C. S., 2007) :

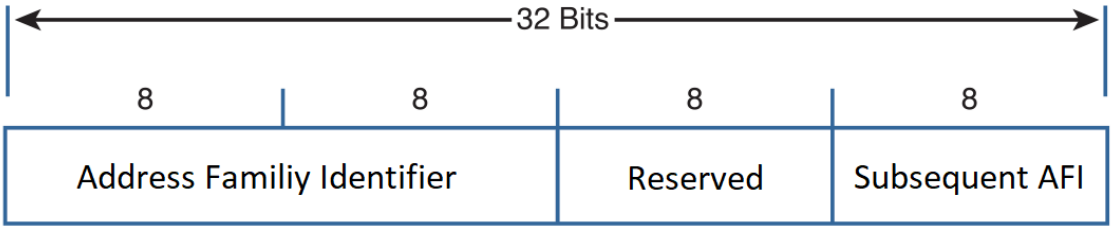

Figure 2-17 champs additionnels du message ROUTE REFRESH

- Address Family Identifier : C'est un champ de 2 octets. Ce champ combiné avec Subsequent AFI identifie la famille de protocole de couche réseau transportés, à laquelle l'adresse contenu dans l'attribut NEXT_HOP doit appartenir, l'encodage de l'adresse NEXT_HOP et la sémantique de l'information d'accessibilité.
- Reserved : champs d'un octet, tous ses bits doivent être à 1 lors de l'envoie. Ce champ doit être ignoré à la réception.
- Subsequent Address Family Identifier : champ d'un octet.

Un routeur peut envoyer un message ROUTE REFRESH à un pair qui a annoncé la capacité Route Refresh. Les <AFI> et <SAFI> que doit contenir ce message ROUTE REFRESH doivent être les mêmes que les <AFI> et <SAFI> reçus dans les capacités du message OPEN (les capacités sont incluses dans le champ des paramètres optionnel du message OPEN).

10. MultiProtocol BGP

BGP a été conçu pour transporter des informations d'accessibilité de couche réseau (NLRI), ce terme est assez vague. Dr. Yakov Rekhter le principal concepteur de BGP et MPLS avait l'habitude de dire ''Notre but est de faire marcher l'entreprise, si pour faire marcher l'entreprise il fallait mettre Shakespeare dans BGP je mettrais Shakespeare dans BGP'' (Sue Hares, 2012), bien que cela semble être une blague BGP peut bel et bien transporter Shakespeare dans ses NLRI.

Pour que BGP soit capable de transporter tout type d'information dans ses NLRI des extensions doivent être ajoutés à BGP, faisant ainsi de BGP le MultiProtocol BGP ou MP-BGP.

Les extensions BGP activent le support de nouvelles familles d'adresses, ces familles d'adresses peuvent être de couche liaison de donnée, réseau ou transport. Par exemple IPv4, IPv6, VPNv4, VPNv6, L2VPN, L3VPN… Ce sont des familles d'adresse dont le support par BGP peut être activé par les extensions BGP.

MP-BGP a été introduit par la RFC 4760 pour donner la capacité à BGP de supporter le routage pour plusieurs type d'NLRI. Les seules choses qui ont dû être ajoutées au BGP de la RFC 4271 pour pouvoir supporter plusieurs types d'NLRI sont (T. Bates C. S., 2007):

a) La capacité d'associer un protocole de couche réseau particulier avec l'information du prochain saut
b) La capacité d'associer un protocole de couche réseau particulier avec une information d'accessibilité de couche réseau.

Pour identifier individuellement un protocole de couche réseau associé à une information de prochain saut ainsi qu'une sémantique pour ses NLRI, on utilise la combinaison d'une famille d'adresse avec une sous famille d'adresse.

La RFC 4790 introduit deux attribut 'optional nontransitive' :

- MP_REACH_NLRI (Multiprotocol reachable NLRI)
- MP_UNREACH_NRLI (Multiprotocol unreachable NLRI)

Ces deux attributs sont contenus dans les messages UPDATE pour annoncer ou retirer des routes. Ces attributs identifient le protocole annoncé ainsi que le type d'adresses du protocole, l'information de prochain saut de la route et la NLRI

Les attributs sont contenus dans le champ Path Attribute du message UPDATE, l'attribut MP_REACH_NLRI est encodé comme suit :

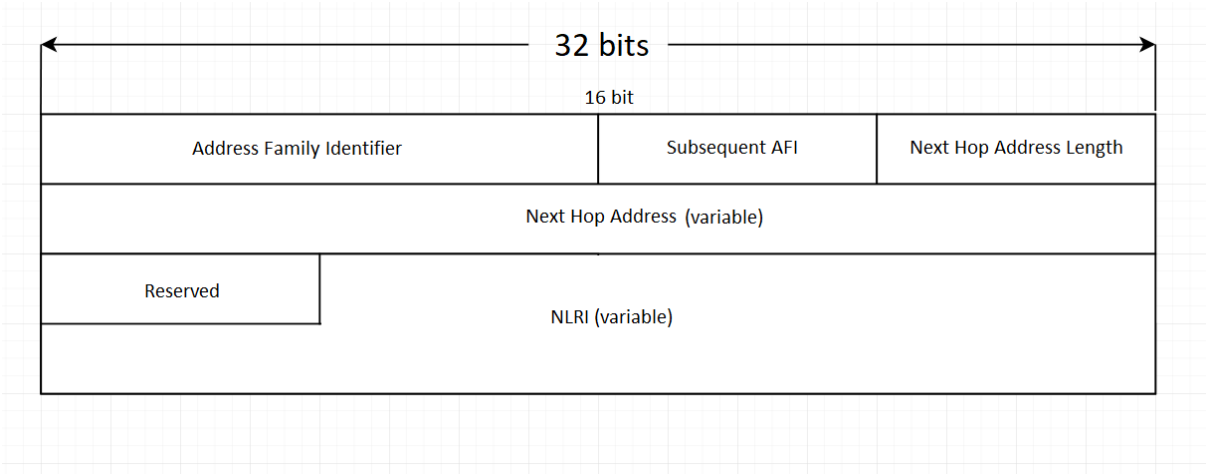

Figure 2-18 encodage de l'attribut MP_REACH_NLRI

- Address Family Identifier : ce champ de 2 octets, spécifie le protocole (IPX, AppleTalk, Decnet 4, Ethernet …) à qui la NLRI appartient.
- Subsequent Address Family Identifier : ce champ d'un octet, spécifie un type d'adresse fonctionnel appartenant au protocole déjà spécifié par l'AFI.
- Next Hop Address Length : champ d'un octet, spécifie la longueur en octet du champ Next Hop Address, la longueur maximale est de 256 octets.

- Next Hop Address : Champ à longueur variable, contient l'adresse du prochain saut sur le chemin vers la destination, le protocole de couche réseau de cette adresse est identifié par la combinaison AFI, SAFI.
- Reserved : champs d'un octet, tous ses bits doivent être à 1 lors de l'envoi. Ce champ doit être ignoré à la réception.
- Network Layer Reachability Information : champ à longueur variable, liste les NLRI des routes praticables annoncées dans l'attribut. La sémantique de la NLRI est identifiée par la combinaison de l'AFI et SAFI.

L'attribut MP_UNREACH_NLRI est encodé comme suit :

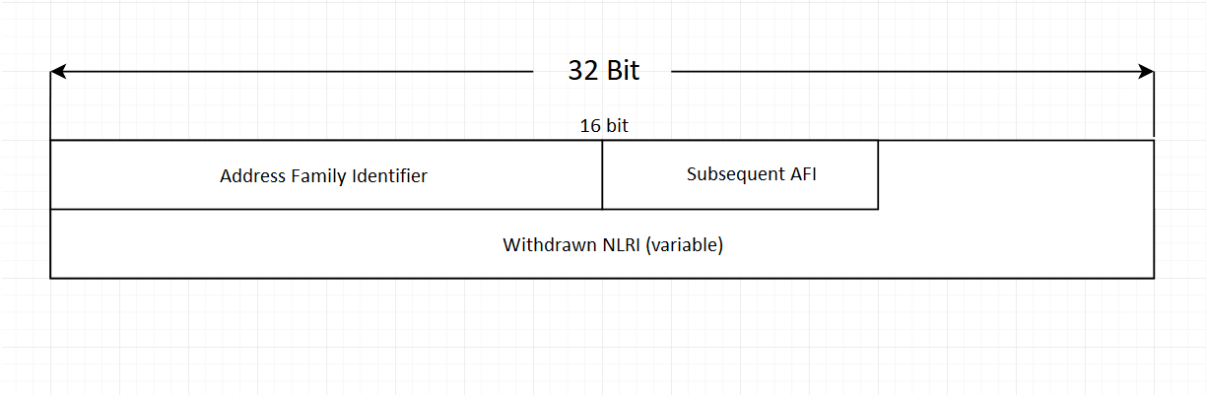

Figure 2-19 encodage de l'attribut MP_UNREACH_NLRI

Les deux champs Address Family Identifier et Subsequent Address Family Identifier sont les mêmes que pour l'attribut MP_REACH_NLRI.

- Withdrawn Routes : c'est un champ à longueur variable, liste les NLRI des routes à retirer. La sémantique de la NLRI est identifiée par la combinaison de l'AFI et SAFI.

Une NLRI est composé d'une 2-tuple ou plus de la forme (longueur, préfixe).

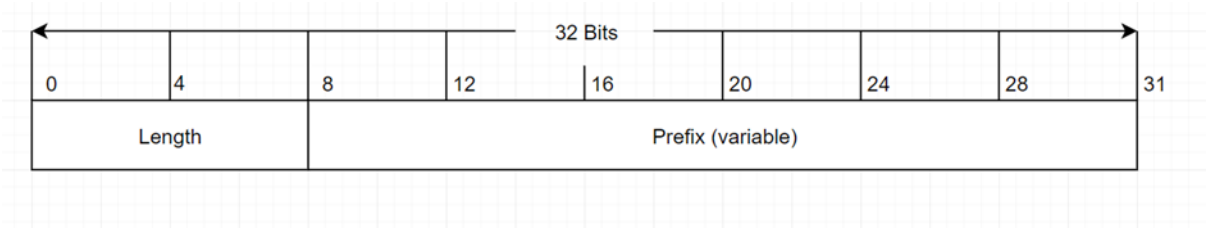

Figure 2-20 contenu du champ NLRI

- Length : champ d'un octet. Indique la longueur en bit du préfixe.
- Prefix : champ à longueur variable, contient un préfixe d'adresse suivi d'un nombre minimum de bits nécessaires pour remplir un octet.

A. La capacité de support des différents protocoles

Lors de l'établissement d'une session BGP entre deux pairs, ces pairs doivent se mettre d'accord sur les protocoles des NLRI qui vont être échangés. La RFC 5492 introduit l'annonce des capacité BGP avec un nouveau paramètre optionnel dans le message OPEN appelé ''Capabilities''.

Un routeur BGP détermine les capacités d'un pair en examinant la liste des capacités présentes dans le paramètre optionnel ''Capabilities''. La valeur du champ ''parameter type'' est de 2 pour les

''Capabilities'', dans le champ ''parameter value'' on peut trouver un ou plusieurs 3-tuplet de ce type (J. Scudder Juniper Networks, 2009) :

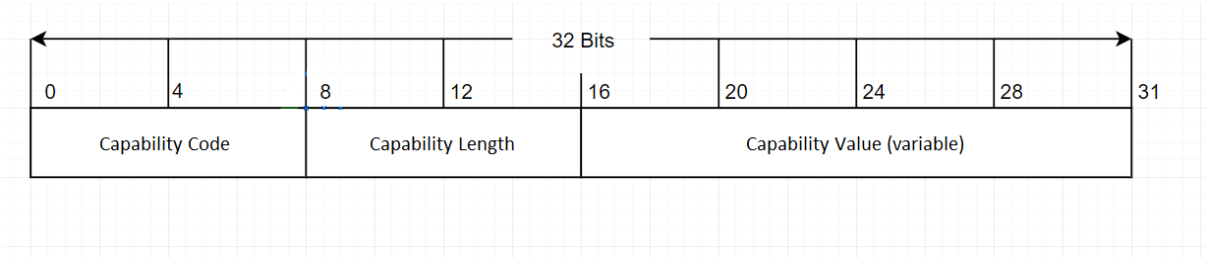

Figure 2-21 contenu du champ parameter value

- Capability Code : champ d'un octet. Contient un entier non signé qui identifie sans ambiguïté une capacité individuelle.
- Capability Length : champ d'un octet. Contient un entier non signé qui correspond à la longueur en octet du champ Capability Value
- Capability Value : champ à longueur variable. Ce champ est interprété selon la valeur du champ Capability Code.

Lors de l'établissement d'une session BGP avec un pair, le routeur BGP reçoit un message OPEN avec des capacités en paramètre. Il peut soit accepter les paramètres décrits et établir la session ou refuser les capacités et terminer la session en envoyant un message NOTIFICATION avec sous-code d'erreur 7.

Un message NOTIFICATION avec le sous-code 7 signifie que le pair ne supporte pas une ou plusieurs capacités, ce message de NOTIFICATION est envoyé seulement quand un pair connait une capacité mais ne la supporte pas. Un routeur BGP n'envoie pas de message NOTIFICATION et ne termine pas une session pour une capacité qu'il ne connait pas.

BGP utilise les capacités pour déterminer le support d'MP-BGP à l'établissement d'une session. Dans le champ ''Capability Value'' on utilise la valeur 1 pour indiquer le support d'MP-BGP. La valeur du champ Capability Length sera de 4 pour indiquer que la longueur du champ Capability Value est de 4 octets.

Le champ spécifie le protocole supporté en indiquant l'AFI et SAFI du protocole supporté :

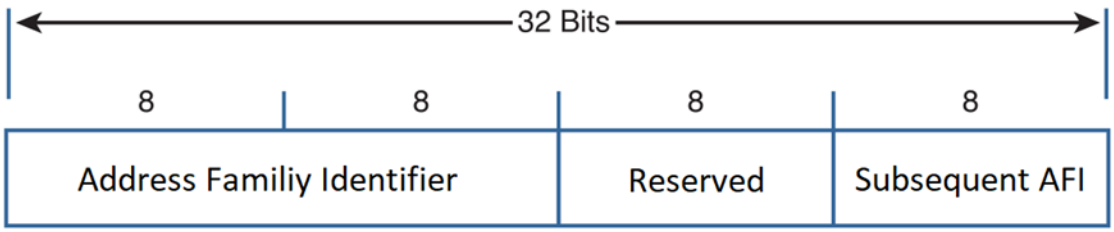

Figure 2-22 contenu du champ Capability Value

Un routeur BGP qui supporte plusieurs (AFI, SAFI) doit inclure chaque (AFI, SAFI) comme paramètre optionnel séparé.

11. BGP à très large échelle

BGP a été conçu dès le départ pour fonctionner sur des très larges réseaux. Les ingénieurs ont mis en place des outils qui ont pour but de réduire le nombre de taches routinières qu'un administrateur doit effectuer. Sans ces outils le nombre de ces taches grandirait exponentiellement au fur et à mesure que le réseau grandi.

A. Outils de configurations et d'optimisation

BGP peut s'adapter à la taille d'un réseau, il peut fonctionner aussi bien sur un réseau de trois routeurs qu'un réseau aussi grand qu'Internet.

Pour un opérateur humain sur un routeur BGP, il faut établir chaque liaison BGP manuellement, puis configurer ces liaisons individuellement et leurs appliquer leur politique de routage. Sur trois routeurs c'est assez simple, mais quand il s'agit d'un large réseau avec plusieurs milliers de liaisons il est impossible pour un opérateur humain de gérer toutes ces liaisons individuellement et ce sur chaque routeur sur lequel il opère.

Des outils ont été créés pour permettre aux opérateurs de gérer les liaisons d'une façon groupé plutôt qu'individuelle.

- Peer Groups : Permet de créer un groupe avec des configurations et politiques de routage, ensuite on peut ajouter des liaisons a ce groupe pour qu'elle hérite les configurations et politiques de routage du groupe.
- Peer Templates : Outil avancé et plus flexible pour créer des modèles de configuration (session template) et des modèles de politiques de routage (policy template). On applique un modèle de configuration et un modèle de politique de routage à une liaison pour qu'elle hérite d'eux.

Les Peer Groups et les Peer Templates sont mutuellement exclusifs. Une liaison avec un pair ne peut pas appartenir à un groupe et en même temps hériter d'un modèle de configuration ou de politique de routage.

L'une des améliorations majeures du Peer Template est l'héritage, un modèle de configuration ou de politique de routage peut hériter d'un autre modèle de configuration ou de politique de routage.

12. Les communautés BGP

Une communauté est une étiquette qui est collée à des routes qui partagent une quelconque propriété.

Il revient à l'administration d'un système autonome de déterminer à quelle communauté une route appartient. Les administrations de différents AS doivent se mettre d'accord sur les propriétés d'une communauté.

La communauté est communiquée avec l'attribut 'optional transitive' 'Community' inclus dans les messages UPDATE. Cet attribut contient un entier non signé de 32 bits qui identifie la communauté à qui appartient la route. Une route peut appartenir à plusieurs communautés.

L'entier qui identifie la communauté a une sémantique définie. Les deux premiers octets sont le numéro d'AS du système autonome qui a ajouté l'attribut communauté à la route, les deux derniers octets sont le numéro de la communauté.

L'utilité des communautés est de marquer des routes pour les système autonomes voisins, par exemple deux système autonomes voisins peuvent se mettre d'accord pour qu'une communauté contienne les routes congestionnées. Un AS peut ensuite marquer des routes avec cette communauté pour signifier que la route est congestionnée avant de les annoncer à son voisin. Une fois la route reçue, le voisin saura que la route est congestionnée et prendra alors les mesures nécessaires pour réduire la préférence de cette route et ne l'utilisera qu'en cas de dernier recours. Ceci est juste un exemple d'utilisation des communautés, le but des communautés est de communiquer quelque chose à un autre pair.

III. INTRODUCTION à MPLS

1. Définition de MPLS

Multi Protocol Label Switching, en français « commutation par étiquette de multiple protocoles », est une technique de transport de donnée pour les réseaux WAN de très haute performance. MPLS se base sur une étiquette pour transmettre un paquet d'un nœud à un autre plutôt que d'utiliser les informations contenues dans les entêtes de couche réseau et transport, Ainsi il évite la complexité de la recherche d'une route à travers une table de routage et facilite les décisions de qualité de service. MPLS peut encapsuler une très grande variété de protocole réseau d'où le nom « multi protocol ».

Les « labels » sont échangés entre les routeurs pour qu'ils puissent ainsi créer un mapping (une cartographie) label-à-label, ces labels sont attachés au paquet IP, ce qui permet aux routeurs de transmettre le paquet en se référant seulement au label sans jamais voir l'adresse de destination de couche réseau contenue dans le paquet. On appelle cela la commutation par label au lieu de routage de couche réseau.

2. Qu'est-ce qu'un fournisseur de service ?

Un fournisseur de service (Service Provider ou SP) est une entreprise qui met un service précis (inter connectivité, stockage, application, infrastructure) à la disposition de particuliers ou d'entreprises.

Pour assurer les services proposés un SP doit avoir l'infrastructure adéquate qui est un inter-réseau avec une bande passante très élevée. Cet inter-réseau généralement appelé backbone et a pour rôle de connecter plusieurs parties du réseau, fournissant un moyen de connexion entre les réseaux. Normalement un backbone doit avoir une capacité beaucoup plus grande que celle des réseaux qui sont connectés à lui.

3. Les prédécesseurs de MPLS

Avant MPLS les protocoles WAN les plus populaires étaient 'Asynchronous Transfer Mode ou ATM' et 'Frame Relay'. Ces deux protocoles étaient les plus rentables pour un réseau WAN pour transporter des protocoles variés. Néanmoins il est a noté que ces deux protocoles sont des protocoles de couche 2 ou couche liaison de donnée alors que MPLS ne l'est pas, et c'est ce qui donne la possibilité d'utiliser MPLS à travers ATM ou Frame Relay.

Les réseaux WAN basés sur Frame Relay sont assez rigides et dotés d'une très faible bande passante, ce qui ne répond pas du tout aux exigences d'aujourd'hui, Frame Relay n'est plus déployé mais cependant il y a toujours des régions du monde difficiles d'accès où Frame Relay est encore utilisé.

MPLS et ATM sont des protocoles assez différents, les deux protocoles offrent une communication orienté connexion (une session est établie entre chaque point terminal pour la durée de la communication). La différence la plus importante est dans l'encapsulation et le transport, MPLS peut transporter des paquets de longueur variable alors qu'ATM transporte uniquement des cellules de 53 octets, les paquets dans ATM doivent être segmentés en cellules puis réassemblés une fois arrivés à destination, ce qui ajoute une importante complexité et surcharge au flux de données. L'avantage le plus décisif qu'offre possible MPLS par rapport à ATM est que MPLS a été conçu pour être complémentaire au protocole IP alors que ATM est incompatible avec IP ce qui demande des adaptations complexes de la part des architectes réseaux, ce qui le rend inadapté au réseau d'aujourd'hui.

MPLS possède des fonctionnalités très appréciées par ses utilisateurs que les autres protocoles ne possèdent pas.

4. Les bénéfices de l'utilisation de MPLS

Les bénéfices les plus importants qu'apporte l'usage de MPLS

1. Une infrastructure réseau unifiée

MPLS étiquette chaque paquet entrant dans le réseau avec un label en se basant sur un critère préconfiguré et le fait commuter à travers le réseau. L'avantage de MPLS est qu'il peut commuter un très grand nombre de protocoles. En utilisant MPLS avec IP on augmente le nombre de protocoles possibles pouvant être transportés car ajouter un label au paquet IP permet à MPLS de transporter IPv4, IPv6, Ethernet, HDLC, PPP …

La fonctionnalité qui permet de transporter n'importe quelle couche 2 à travers un backbone MPLS est appelée « Any Transport over MPLS » ou AToM. Les routeurs sur le chemin qui commutent les paquets ne sont pas obligé de savoir ce qu'il y a dedans.

En d'autres termes MPLS permet l'acheminement de plusieurs protocoles dans un seul et même réseau, ce qui supprime le besoin de multiple réseau.

2. Le cœur du réseau n'a pas besoin d'utiliser BGP

Dans un réseau IP d'un SP, pour acheminer un paquet chaque routeur doit regarder dans sa table de routage pour trouver un chemin vers la destination du paquet. Si le paquet a une destination externe au SP, les routes externes doivent être présentes dans la table de routage de chaque routeur du réseau. BGP permet de diffuser ces routes externes entre les routeurs du réseau, ce qui fait que chaque routeur du SP doit utiliser BGP.

MPLS permet d'acheminer les paquets en regardant leur label au lieu de leur adresse IP, MPLS associe un label à un routeur de sortie, le label indique à chaque routeur interne a quel routeur de frontière le paquet doit être transmit afin d'être envoyé vers sa destination. Les routeurs internes n'ont plus besoin de connaitre les adresses IP de destination externes pour acheminer le paquet au bon routeur de frontière. Il n'est plus nécessaire d'utiliser BGP sur les routeurs internes. Ainsi on peut économiser beaucoup de ressources sur les routeurs internes, BGP est un fardeau car une table de routage avec les routes d'Internet représente facilement plus de 150 000 routes.

Mais cependant il est toujours nécessaire d'avoir BGP fonctionnel sur les routeurs de frontière, car ils ont toujours besoin de regarder l'adresse IP de destination et déterminer à quel autre routeur de frontière le paquet doit être transmit.

3. Traffic Engineering

L'idée derrière le Traffic Engineering (TE) en français ingénierie du trafic, est d'utiliser les ressources de l'infrastructure réseau d'une manière optimale.

Dans un réseau il y a toujours des chemins préférés, car ils ont une plus grande bande passante un meilleur délai, un minimum de sauts, IP choisi toujours le chemin le moins couteux. Mais il arrive que ce meilleur chemin soit saturé alors que d'autres chemins vers la même destination mais avec un plus grand cout ne soient pratiquement jamais empruntés.

Avec MPLS-TE implémenté un SP peut ;

- Diriger un certain trafic vers un certain préfixe de destination par un chemin désigné.
- Affecter un chemin à chaque type de service ainsi assurant une certaine qualité de service.
- Repartir le trafic de manière homogène sur le backbone.

4. La qualité de service

L'un des bénéfices premiers de MPLS est qu'il peut supporter la qualité de service, ce qui est très important pour les SP qui acheminent les paquet VoIP ou autres données sensibles au délai, la jigue et la perte de paquet. Ainsi MPLS peut prioriser les paquets selon leurs contenus.

5. MPLS VPN

Un VPN est une émulation d'un réseau privé à travers un réseau public. Le SP peut fournir deux modèles de VPN à ses clients

- Le modèle 'Overlay VPN'. Dans ce modèle le SP offre un lien point à point ou un circuit virtuel à travers son réseau reliant deux routeurs du client situés sur des zones géographiquement éloignées. Le routeur du client n'a aucune perception du réseau du SP et a l'impression qu'il est directement relié à son autre routeur sans aucun intermédiaire.
- Le modèle 'VPN peer-to-peer', dans ce modèle le SP transporte les données du client, mais crée aussi des relations de voisinage entre son routeur et celui du client puis partage des routes avec celui-ci.

Avant que MPLS n'existe il était possible de créer le modèle 'VPN peer-to-peer', mais le challenge était de garder les routes d'un client privée, et ne pas les distribuer à un autre client. Pour y parvenir les ingénieurs ont dû être inventifs, ils ont utilisé des listes d'accès et des filtres pour éviter que les routes d'un client se propagent vers un autre client. Mais cela était fastidieux. En effet, ajouter un nouveau client était chose très compliquée car toutes les listes d'accès et filtres devaient être changés.

'MPLS VPN' est une application qui a rendu le modèle 'peer-to-peer' viable et beaucoup plus facile à gérer. L'ajout ou la suppression d'un client demande beaucoup moins de temps et d'efforts.

La confidentialité dans MPLS VPN est garantie en utilisant le concept de routage et acheminement virtuel « Virtual routing/forwarding » ou VRF.

6. Un acheminement des données optimal

Avec 'Frame Relay' et 'ATM' qui sont purement des protocoles de couches 2, les routeurs s'interconnectent à travers les commutateurs 'Frame Replay' et 'ATM' par le moyen de circuits virtuels créés entre ces routeurs.

Pour qu'un routeur représentant un site puisse communiquer avec un autre routeur représentant un autre site il faut qu'ils soient reliés par un circuit virtuel. Pour avoir une connectivité totale il est possible pour l'entreprise d'acheter un circuit virtuel de chacun de ses sites vers tous ses autres sites, mais cela reviendrait beaucoup trop cher. La solution pour l'entreprise est d'acheter le nombre minimum de circuit virtuel pour réaliser une connectivité vers chacun de ses sites, puis pour faire communiquer deux sites qui n'ont pas de circuit virtuel direct il suffit de relayer d'un site à l'autre jusqu'à destination.

Le problème pour le SP avec cette solution est qu'en relayant les données d'un site à l'autre, les mêmes données traversent les même commutateur plusieurs fois, ce qui est un usage non optimal des ressources.

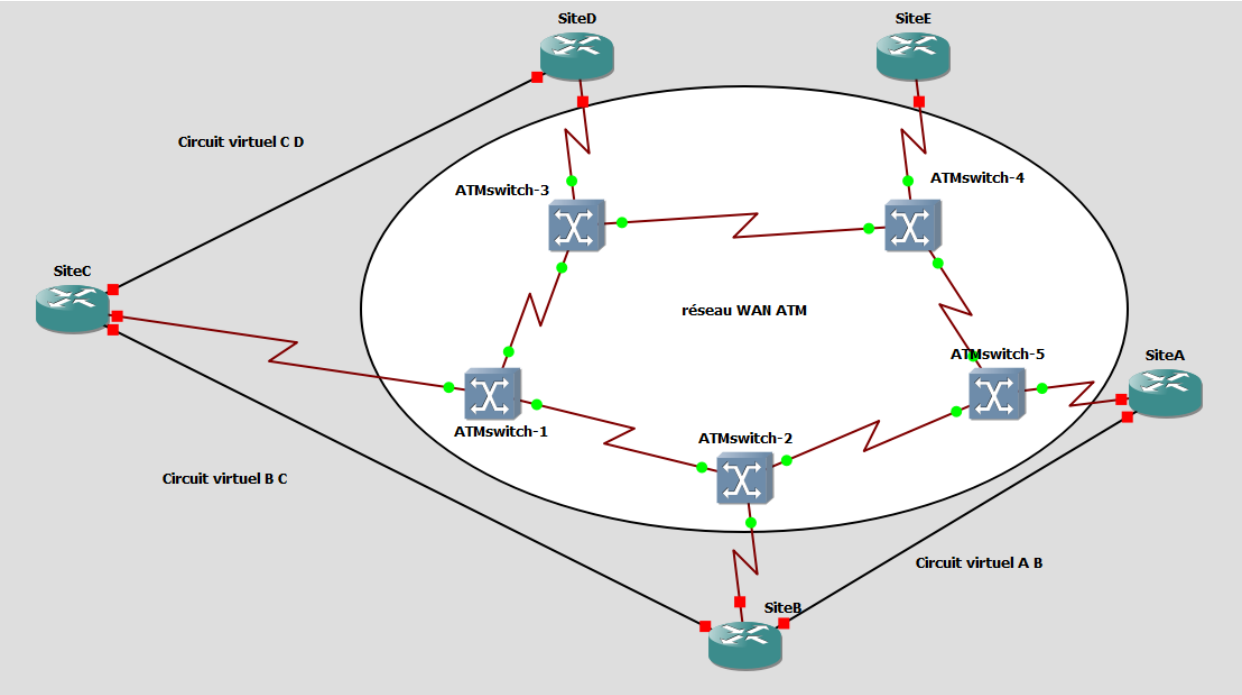

Figure 3-1 Commutation de données avec ATM

Dans la figure 3-1, pour que le site A communique avec le site D il faut qu'il fasse relayer ses données par les sites B et C.

Quand on utilise 'MPLS VPN' les données vont directement vers la destination sans prendre aucun détour d'où un acheminement optimal.

7. Une intégration d'IP sur ATM

ATM a eu beaucoup de succès par le passé dans les réseaux WAN, mais son usage s'est strictement limité au WAN au cœur d'un SP. Bien que l'intégration d'IP sur ATP ne soit pas chose facile, ces

mêmes SP qui ont utilisé ATM avait aussi déployé des 'backbone IP'. La communauté des ingénieurs a proposé plusieurs solutions pour y parvenir, au final le forum d'ATM a proposé une solution complète mais très complexe pour intégrer IP sur ATM.

Toutes les solutions proposées étaient insatisfaisantes, une meilleure solution pour intégrer IP sur ATM a été un facteur important dans l'invention de MPLS.

5. L'histoire de MPLS

En 1996 l'entreprise 'Ipsilon Networks' qui est spécialisé dans la commutation réseau particulièrement sur les WAN avec ATM ont proposé l'idée de la commutation basée sur le flux pour les réseaux ATM. L'idée été bonne mais n'a pas eu beaucoup de succès car elle reposait sur ATM.

Plus tard 'Cisco' proposa la même chose mais cette fois ci sans se limiter à ATM, cette solution fut appelée « Tag Switching » et Cisco en était le seul propriétaire.

Plus tard Cisco a remis « Tag Switching » à l'IETF qui est un groupe ouvert qui participe à l'élaboration des standards d'Internet afin que cela soit standardisé, « Tag Switching » est devenu la norme mondiale qu'est MPLS avec la toute première RFC 2547 intitulé « BGP/MPLS VPNs »

A. Vue globale de MPLS

Dans les réseaux conventionnels, pour l'acheminement de paquet à travers une série de routeurs une décision de routage doit se faire de façon indépendante sur chaque routeur que le paquet traverse. Le routeur doit analyser l'entête de couche réseau pour déterminer quel sera le prochain saut (routeur) à acheminer le paquet et si la qualité de service est configurée le routeur devra analyser l'entête de couche transport pour déterminer la classe du paquet (QoS). Les entêtes de couche réseau et transport contiennent beaucoup d'informations inutiles à la prise de décision de routage.

Dans les réseaux MPLS les routeurs n'ont pas besoin de recalculer à chaque saut le prochain saut ou de définir la classe du paquet, Les paquets sont étiquetés dès leurs entrées dans le réseau MPLS et les routeurs participants dans l'acheminement du paquet n'ont guère besoin d'analyser le paquet, quand un routeur reçoit un paquet, son étiquète seule permet de déterminer le prochain saut ainsi que la classe du paquet.

L'acheminement de trame étiquetée est bien plus simple que le routage de couche réseau car les étiquettes MPLS sont bien plus faciles à interpréter et à associer avec une interface de sortie qu'une adresse de couche réseau conventionnelle. MPLS combine la simplicité de la commutation de paquet trouvé dans les réseaux ATM et Frame Relay avec la simplicité de la conception, déploiement et maintenance des réseau IP.

Les possibilités offertes par MPLS on permit l'invention de plusieurs applications basées sur MPLS tel que MPLS VPN, Traffic Engineering, AToM et VPLS.

6. L'architecture de MPLS

Ce chapitre présente le fonctionnement de MPLS ainsi que ses détails de conception.

MPLS est indépendant du protocole qu'il transporte d'où le nom « multi protocol ». Il effectue une commutation par étiquette « label ». La prochaine section traitera de cette commutation par étiquette.

7. MPLS dans le modèle OSI

Les ingénieurs ont mis au point le modèle OSI qui est un standard dans la communication réseaux, le rôle de ce modèle et de permettre aux protocoles distribués sur 7 différentes couches de travailler en collaboration avec les autres protocoles des couches adjacentes.

Chaque couche du modèle a un rôle bien défini ainsi que des fonctionnalités qui lui sont bien propres.

MPLS ne respecte pas la règle, il a des fonctionnalités de couche 2 (commutation), couche 3 (routage) et possède même des fonctionnalités des couches supérieures. De plus, dans l'encapsulation d'un paquet il ne possède pas sa propre place.

Le label MPLS se glisse entre l'entête de la couche 3 et l'entête de la couche2, mais il n'est ni un protocole de couche 2 ni de couche 3 car l'encapsulation couche 2 et couche 3 restent toujours présentes.

Les protocoles de couche 2 et 3 peuvent être n'importe quel protocole, MPLS est indépendant des autres protocoles.

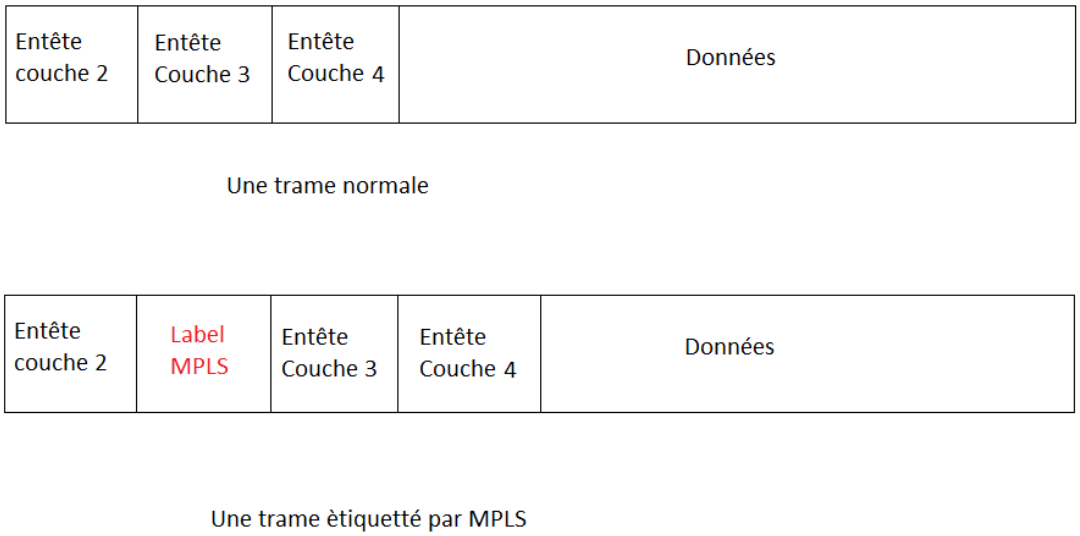

Figure 3-2 MPLS dans une trame.

Dans l'entête des protocoles qui transportent plusieurs types de protocoles des couches supérieures il est nécessaire d'indiquer dans un champ de cet entête quel est le prochain protocole encapsulé. L'entête est un label MPLS. Le tableau suivant liste, pour certains protocoles de couche liaison de donnée populaires, le champ désignant le protocole de l'entête suivant, et la valeur qu'il prend lorsqu'il s'agit d'une étiquette MPLS.

Protocole de couche 2	**Nom du champ identifiant**	**Valeur (hexadécimal) utilisé pour indiquer un label MPLS**
PPP	PPP protocol field	0281
Ethernet 802.3	Ethertype value	8847
Frame Relay	NLPID	80

8. Le label MPLS

C'est un champ de 32 bits avec une structure définie comme suit

Figure 3-2 Structure d'un Label MPLS

- Label (bits de 0 à 19) : Cette valeur peut aller de 0 jusqu'à $2^{20} - 1$ soit 1 048 575, ce sont les étiquettes utilisées par MPLS, les 16 premières valeurs sont réservées et ont une signification spéciale
- Traffic Class (bit de 20 à 22) : c'est des bits qui ont été réservés pour un usage expérimental à l'origine, ils ont fini par être utilisés pour la qualité de service par la suite.
- Bottom of Stack ou BoS (le 23eme bit) : il est possible de placer un label MPLS sur un paquet qui possède déjà un label ou plusieurs labels, pour savoir qui est le dernier label on utilise ce bit, 0 signifie que ce label est suivi par d'autres labels, 1 signifie que c'est le dernier label
- Time To Live (bits de 24 à 31) : ces 8 bits sont utilisés pour déterminer la durée de vie d'un paquet, il à la même fonctionnalité que le TTL trouvé dans l'entête IP, sa valeur est décrémentée de 1 à chaque saut et une fois arrivé à 0 le paquet est détruit.

(E. Rosen C. S., 2001)

9. Définitions de quelque termes liés à MPLS

Avant d'aller plus loin il est nécessaire de définir quelques termes liés à MPLS.

- Label (étiquette)

C'est un entier signé qui identifie de manière unique une FEC, souvent le Label à une signification locale au routeur.

- Paquet étiqueté

Un paquet étiqueté est un paquet auquel on a apposé une étiquète.

- Label Switching Router (LSR)

Le routeur à commutation par label est un routeur capable de supporter MPLS, il est capable de recevoir et émettre des paquets étiquetés avec un label MPLS. On peut distinguer trois types de LSR selon leur positionnement sur le chemin du paquet (Jean-Philippe Vasseur, 2005) :

    - Ingress LSR (entrée)

C'est le premier LSR sur le chemin du paquet qui n'est pas encore étiqueté, il est le premier à apposer 'insérer' un label dans la trame puis commute la trame dans le réseau MPLS.

    - Egress LSR (sortie)

C'est le dernier LSR sur le chemin du paquet, ce paquet enlève tous les labels présents dans le paquet.

- Intermediate LSR (intermédiaire)

Les LSR intermédiaire reçoivent des paquets déjà étiquetés, peuvent faire une opération dessus qui est l'ajout, le changement ou suppression d'un label puis commutent la trame.

Il est à noter que les routeurs Ingress et Egress sont des routeurs de bordure (Edge LSRs).

Il est à noter qu'on utilise les deux termes paquet et trame, La distinction tient du fait que le paquet possède une encapsulation de couche 3 mais pas d'encapsulation de couche 2 alors que la trame possède les deux. L'encapsulation de couche 2 est indépendante de MPLS.

Chez les fournisseurs de service SP on peut trouver une autre terminologie pour les routeurs, on appelle :

- 'Provider Edge (PE) router', les routeurs du fournisseur de service présents à la bordure de son réseau.
- 'Provider (P) router', les routeurs du fournisseur qui sont présents à l'intérieur du réseau.

- Label Switched Path (LSP)

Ou chemin de commutation par label. C'est la séquence d'LSR que le paquet doit traverser à travers le réseau MPLS.

Le premier LSR à recevoir un paquet dans un LSP n'est pas forcément l'Ingress LSR, et le paquet peut être un paquet déjà libellé. Cela est souvent le cas quand un LSP est imbriqué dans un autre LSP ce qui est très souvent utilisé avec MPLS Traffic Engineering (MPLS-TE)

- Forwarding Equivalence Class (FEC)

Les paquets d'un même groupe(classe) prédéfini suivent le même chemin dans le réseau et sont traités de la même façon en ce qui concerne la commutation. Ces groupes sont appelés 'Forwarding Equivalence Class (FEC)'. Les paquets du même FEC sont étiquetés avec le même label. (L'inverse n'est pas toujours vrai deux paquets avec le même label peuvent appartenir à différentes FEC).

C'est l'ingress LSR qui décide à quel FEC appartient un paquet.

Les critères qu'utilise cet LSR pour définir à quelle FEC appartient le paquet sont :

  - L'adresse de destination couche 3.
  - La valeur du champs QoS présente dans l'entête couche 3 tel que DiffServ Code Point (DSCP) pour les paquets encapsulés avec IP.
- Label swap/swapping

C'est le fait de changer le label d'entré par une autre information (label, encapsulation, port …) équivalente de sortie

10. La signification des labels

Les labels MPLS ont une signification locale à deux LSRs adjacents. Le label identifie une FEC à laquelle appartient le paquet étiqueté.

La commutation se fait à base de label dans un réseau MPLS, pour donner un sens aux labels. Les LSRs adjacents doivent se mettre d'accord sur les valeurs de chaque FEC, quand un LSR envoie un paquet étiqueté avec un label, la valeur du label doit identifier d'une manière unique une FEC. La

valeur pour identifier une FEC A est locale à chaque routeur, LSR1 peut avoir une valeur de 19 pour FEC A alors que LSR2 peut avoir une valeur de 25 pour la même FEC A, Quand LSR1 commute un paquet de la FEC A à LSR2 il utilise la valeur 25 alors que LSR2 doit utiliser la valeur 19 pour la FEC A quand il envoie à LSR1 un paquet de la FEC A. Pour un paquet entrant on regarde la valeur locale au routeur, pour un paquet sortant on utilise la valeur du routeur à qui on envoie. La direction soit entrante ou sortante d'un paquet détermine si on doit utiliser les valeurs locales ou les valeurs du LSR voisin connecté à cette interface de sortie.

Quand un 'Ingress LSR' reçoit un paquet non libellé, il doit tout d'abord voir dans sa table de routage s'il y a une route vers l'adresse destination de la couche 3 de ce paquet. S'il possède une route alors il « appose » au paquet un label qui va le mener vers sa destination, sinon le paquet est détruit.

Au cours du chemin les autres LSR intermédiaire ne regardent pas l'adresse destination de la couche réseau mais se basent uniquement sur le label. Ils peuvent aussi effectuer des permutations sur les labels en les changeants avec d'autre label. Le paquet est acheminé jusqu'à l'Egress LSR, qui enlève carrément les labels présents et regarde l'adresse de couche réseau du paquet.

Pour que les LSR se mettent d'accord sur la signification des labels, un mécanisme est nécessaire pour dire aux LSR quels labels utiliser pour envoyer un paquet vers telle ou telle destination.

Les labels ont un sens local, ils sont uniques pour chaque paire de routeurs adjacents, ils n'ont aucune signification globale. Pour que deux LSRs voisins puissent se mettre d'accord sur quel label utiliser pour acheminer un paquet vers un ''Egress LSR'' il leurs faut un moyen de communication.

Pour faire communiquer les LSRs sur le sens des labels il y a deux moyens de faire (Ghein, 2007)

1. Ajuster un protocole de routage IP déjà existant pour qu'il soit capable de transporter l'information du label

Cette méthode a l'avantage de ne pas nécessiter l'usage d'un protocole supplémentaire. C'est le même protocole qui distribue les routes ainsi que les labels, ce qui assure la synchronisation.

Son inconvénient est que les protocoles de routage déjà existants doivent être modifiés pour pouvoir supporter cette nouvelle fonctionnalité ce qui n'est pas toujours facile.

L'ajout de cette fonctionnalité sur un protocole à vecteur de distance (Routing Information Protocol, Enhanced Interior Gateway Routing Protocol) est simple, parce que les routeurs annoncent les routes qu'ils ont directement dans leur table de routage, et il est très facile de les lier avec un label.

Les protocoles de routage à état de lien (Open Shortest Path First, Intermediate system to Intermediate system) ne fonctionnent pas de la même manière que les protocoles à vecteur de distance, il est beaucoup plus complexe de faire cela avec les protocoles à état de lien parce que chaque routeur annonce ses liens et les autres routeurs retransmettent ces mêmes annonces inchangées. Pour que deux routeurs voisins se mettent d'accord sur le label à utiliser pour un préfixe il faut qu'ils envoient un message avec le préfixe lié à un label, mais le problème est que si le routeur n'est pas à l'auteur de ce préfixe il n'a pas le droit de l'annoncer directement. L'ajout de cette fonctionnalité reviendrait à donner la possibilité aux routeurs d'annoncer des préfixes dont ils ne sont pas les auteurs, ceci va à l'encontre des règles de conception des protocoles à état de lien.

BGP est un protocole de routage externe, mais cependant il peut être utilisé pour transporter des préfixes couplés à des labels, il est principalement utilisé pour la distribution de label dans 'MPLS VPN'.

Bien qu'il soit plus facile de communiquer le sens des labels avec les protocoles internes à vecteur de distance, ces derniers ne sont pas utilisés dans les grands réseaux, les SP préfèrent utiliser les protocoles à état de liens pour les avantages qu'ils offrent par rapport aux protocoles a vecteur de distance. Le protocole externe BGP est le meilleur choix pour communiquer le sens des labels.

2. Utiliser un nouveau protocole dédié à la distribution des labels

Cette méthode consiste à implémenter un protocole additionnel sur les LSRs qui permettrait d'échanger des informations à propos des labels. Un protocole de distribution de label est un ensemble de procédures avec lesquels un LSR informe un autre de ses associations FEC à label.

Il existe trois protocoles de distributions de label qui sont :

- Tag Distribution Protocol (TDP)

TDP est le premier protocole pour distribuer des labels, il fut développé par 'Cisco' du temps où ils étaient encore propriétaire de « Tag Switching » qui devînt par la suite MPLS.

Après la standardisation de MPLS TDP fut rapidement remplacé par LDP. Aujourd'hui TDP est obsolète.

- Label Distribution Protocol (LDP)

Successeur de TDP, C'est le standard de l'IETF pour la distribution des labels. C'est un mécanisme pour les LSR afin de créer des LSP à travers le réseau en reliant directement (mappage) les informations de routage de la couche 3 avec les chemins de commutation de la couche 2.

- Resource Reservation Protocol (RSVP)

C'est un protocole de la couche 4 (transport), il a été conçu pour réserver des ressources à travers un réseau pour assurer une qualité de service. RSVP est utilisé dans les architectures « Integrated Services ».

L'architecture « Integrated Services » a eu très peu de succès comparé à 'Differentiated Services'. De nos jours il est extrêmement rare de voir l'architecture 'Integrated Services' déployée sur un réseau.

MPLS a redonné un nouveau souffle de vie à RSVP avec Traffic Engineering (RSVP-TE). Nous verrons en détail RSVP-TE dans le chapitre dédié à TE.

A. Les labels réservés

Les Labels de 0 à 15 sont des labels réservés, un LSR ne peut pas les associer avec une FEC. Ces labels ont une fonction spécifique :

- 3 - Implicit NULL Label : assigné par un Egress LSR à une FEC quand il ne veut pas associer un label à cette FEC. L'intérêt de cette procédure est d'éviter à l'Egress LSR de faire deux recherches dans sa LFIB ou RIB, en temps normal un LSR doit chercher le dernier label du paquet dans sa LFIB, s'il est le dernier LSR pour cet LSP alors c'est l'Egress LSR, il doit donc retirer le dernier label puis chercher de nouveau dans sa LFIB le prochain label pour commuter le paquet ou sa RIB s'il n'y a plus de label. Pour ces deux recherches la première n'est pas nécessaire et peut être éviter, ce qui ferait un gain de performance. La solution est

que l'Egress LSR signale à l'avant dernier LSR (en anglais Penultimate LSR) de l'LSP de lui envoyer des paquets sans le dernier label mais un Implicit NULL Label à la place. L'Egress LSR ignore l'Implicite NULL Label et cherche directement le prochain label ou l'adresse IP s'il n'y plus de label.

- 0 et 2- Explicit NULL Label : L'usage de l'Implicit Null rend la commutation de paquet plus efficace, mais il existe un inconvénient. Quand l'avant dernier LSR effectue PHP sur un paquet il le commute avec un label en moins ou sans aucun label s'il avait qu'un seul label. Lors du retrait d'un label, les informations qu'il contient dans le champ EXP qui concernent la qualité de service sont perdues aussi. Dans certain cas on voudrait garder l'information de qualité de service et la transmettre à l'Egress LSR.
  Dans les cas où on a besoin de l'information de qualité de service l'Implicit Null ne doit pas être utilisé, on utilise à la place l'Explicit Null, L'Egress LSR doit signaler à l'avant dernier LSR un label de 0 (IPv4) ou 2 (IPv6) associé avec la FEC pour laquelle il a besoin des informations de qualité de service.
- 1 - Router Alert Label : Ce label peut être présent n'importe où dans la pile de label à l'exception du fond de la pile (le label apposé par l'Ingress LSR). Quand un LSR reçoit un paquet avec le Router Alert Label au sommet de la pile de label, il doit alors examiner le paquet attentivement, le paquet ne sera pas commuter par l'ASIC (Application-Specific Integrated Circuit) mais par le logiciel. Le Router Alert Label sera retirer, LSR utilisera le label suivant pour déterminer comment commuter le paquet, Le Router Alert Paquet est ajouté de nouveau au paquet avant d'être expédier au prochain LSR. Ce label est utilisé pour les opérations de maintenance.
- 4 à 6 - non assignés.
- 7 - Entropy Label : Ce label active la fonctionnalité de 'Load Balancing' pour les réseaux MPLS en éliminant le besoin d'une inspection profonde du paquet par chaque LSR, cette inspection est faite qu'une seule fois par l'Ingress LSR sur un LSP pour extraire les information clés pour le 'Load Balancing' d'un paquet. L'Ingress LSR ajoute l'Entropy Label à la pile de label, ce label sera ensuite utilisé par les autres LSR pour effectuer les opérations de 'Load Balancing'.
- 8 à 12 - non assignés.
- 13 – Generic Associated Channel Label (GAC) : Ce label décrit par la RFC 5586 défini des mécanismes qui fournissent une solution de maintenance étendu au besoin des applications émergentes pour MPLS. (M. Bocci, 2009)
- 14 – Operation And Maintenance Alert Label (OAM) : Ce label décrit par la RFC 3429 est utilisé pour la détection d'erreur et la surveillance des performances. Cisco IOS n'utilise pas ce label, mais utilise à la place une technique appelé MPLS OAM.

B. Penultimate Hop Popping ou PHP

L'usage de l'Implicite NULL Label à la fin d'un LSP est appelé PHP 'Penultimate Hop Popping'.

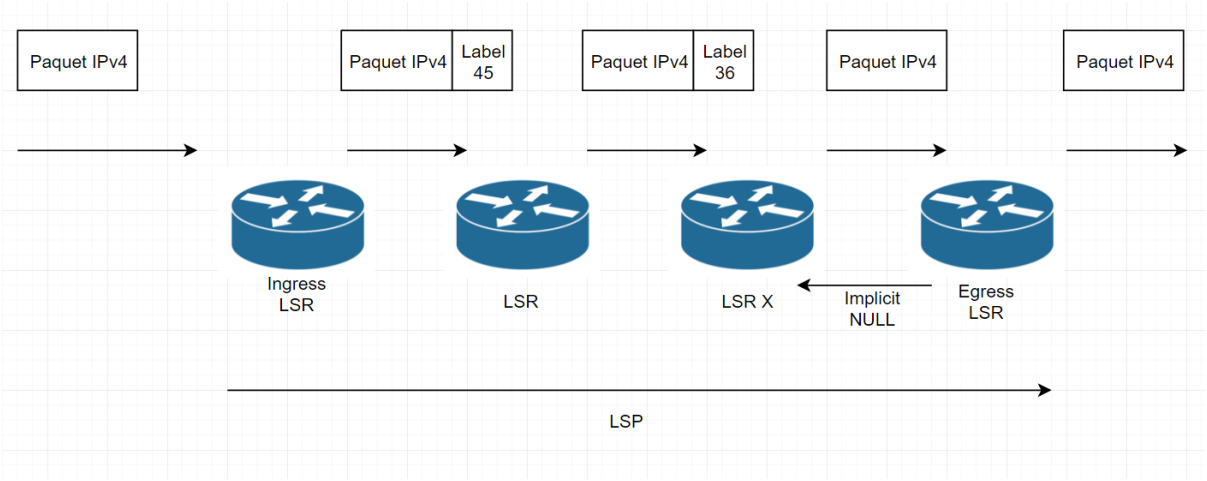

Figure 3-3 retrait du dernier label par l'avant dernier LSR de l'LSP

Dans la figure 3-3 LSR X est l'avant dernier LSR de l'LSP, il a reçu un Implicit-NULL label de la part de l'Egress LSR de cet LSP, donc il devra retirer le dernier label pour chaque paquet appartenant à cet LSP commuté vers l'Egress LSR.

SI on examinait la LFIB de LSR X, qui est une sorte de table de commutation pour les paquets étiqueté on trouverai les informations de la figure 3-4

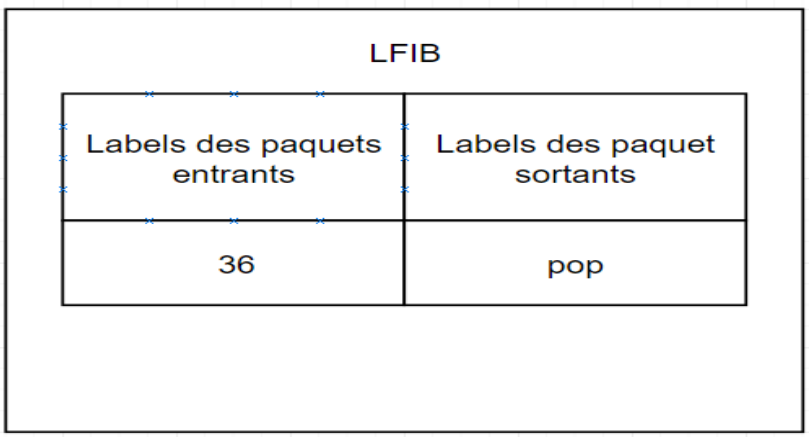

Figure 3-4 Table LFIB du LSR X

L'usage de l'Implicit NULL ne se limite pas seulement aux paquets avec un seul label, il peut aussi être utilisé pour les paquets étiquetés avec plusieurs labels. L'usage de l'Implicit NULL ne signifie pas que tous les labels seront retirés mais juste le dernier label de la pile.

PHP est le comportement par défaut sur les LSR Cisco IOS. L'Implicit NULL est annoncé automatiquement par le LSR pour les routes directement connectées au LSR ou qui ont été résumés par le LSR.

C. Label Information Base ou LIB

Label Information Base est une base de données qui stocké les associations FEC à label attribués localement ou reçues d'un pair MPLS.

D. Les piles de label

Un paquet étiqueté peut avoir plusieurs étiquettes sous forme de pile (dernier arrivé premier servi). Quand un LSR reçoit un paquet étiqueté il regarde seulement la dernière étiquette de la pile pour savoir comment commuter le paquet.

Les piles sont utilisées dans des architecture MPLS hiérarchique et aussi pour la QoS avec MPLS-TE et les tunnels VPN.

E. Implémentation MPLS sur Cisco IOS

Il est important de savoir comment est implémenté MPLS sur les solutions commerciales Cisco IOS pour la conception, diagnostique et maintenance des réseau MPLS en utilisant Cisco IOS.

a. Cisco Express Forwarding ou CEF

Nous avons divisé les fonctionnalités de routage en trois plans (plan de gestion, plan de contrôle et plan de donnée). De ces trois plans, les plans de contrôle et de donnée ont impact direct sur la vitesse de transite des paquets à travers le routeur. Cisco a développé trois approches pour aborder le transite des paquets à travers un routeur dés sont arrivé par une interface d'entrée jusqu'à son expédition par une interface de sortie. On appelle ces approches des modes de commutation de paquets :

- Process Switching : Ce mode est le plus ancien, quand un paquet arrive, il est décapsulé de son entête de couche liaison de donnée, le routeur examine son entête de couche réseau et détermine comment le commuter, un entête de couche liaison de donnée est réécrit, entête avec lequel le paquet est encapsulé avant d'être expédier par l'interface de sortie appropriée.

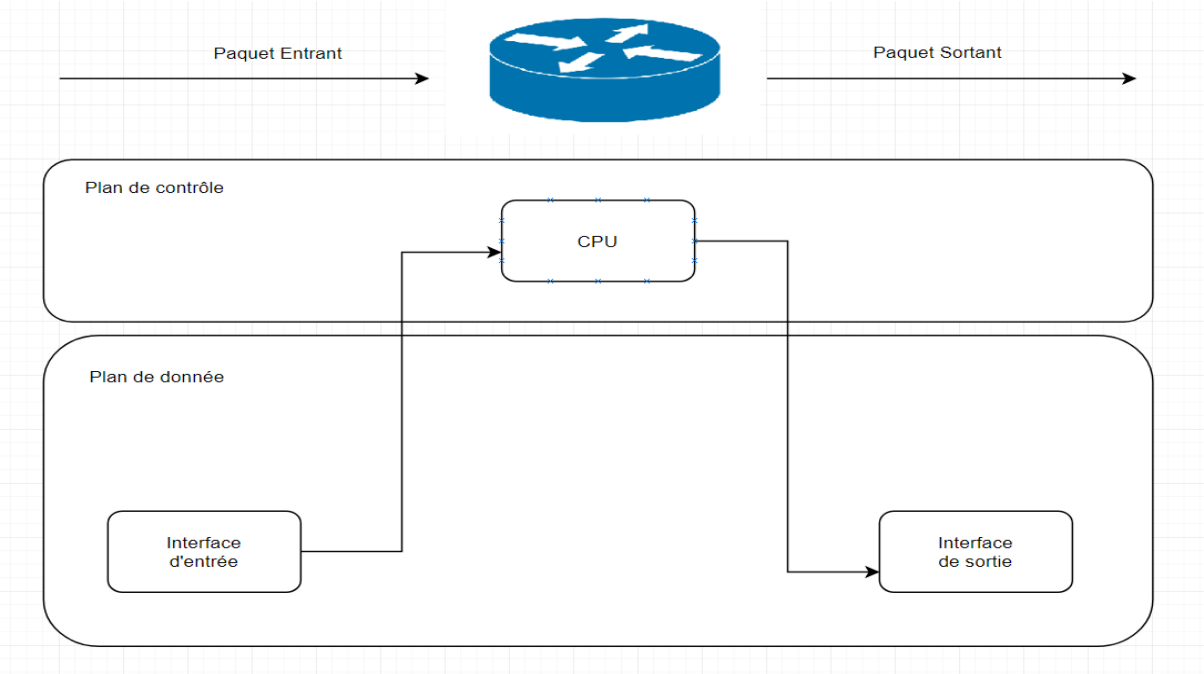

Figure 3-5 vue du mode Process Switching sur les plans de fonctionnalité réseau

Dans ce mode le CPU est directement impliqué dans le processus de commutation de paquet ce qui peut causer des pertes de performances significatives.

- Fast Switching : Ce mode est une amélioration de process switching, il ajoute un cache (table de commutation) qui contient les informations sur les commutations récentes. Pour une

commutation similaire à une commutation précédente il n'est pas nécessaire de solliciter le CPU de nouveau, on utilise l'information contenu dans le cache. Il est toujours nécessaire que le premier paquet soit traiter par le CPU.

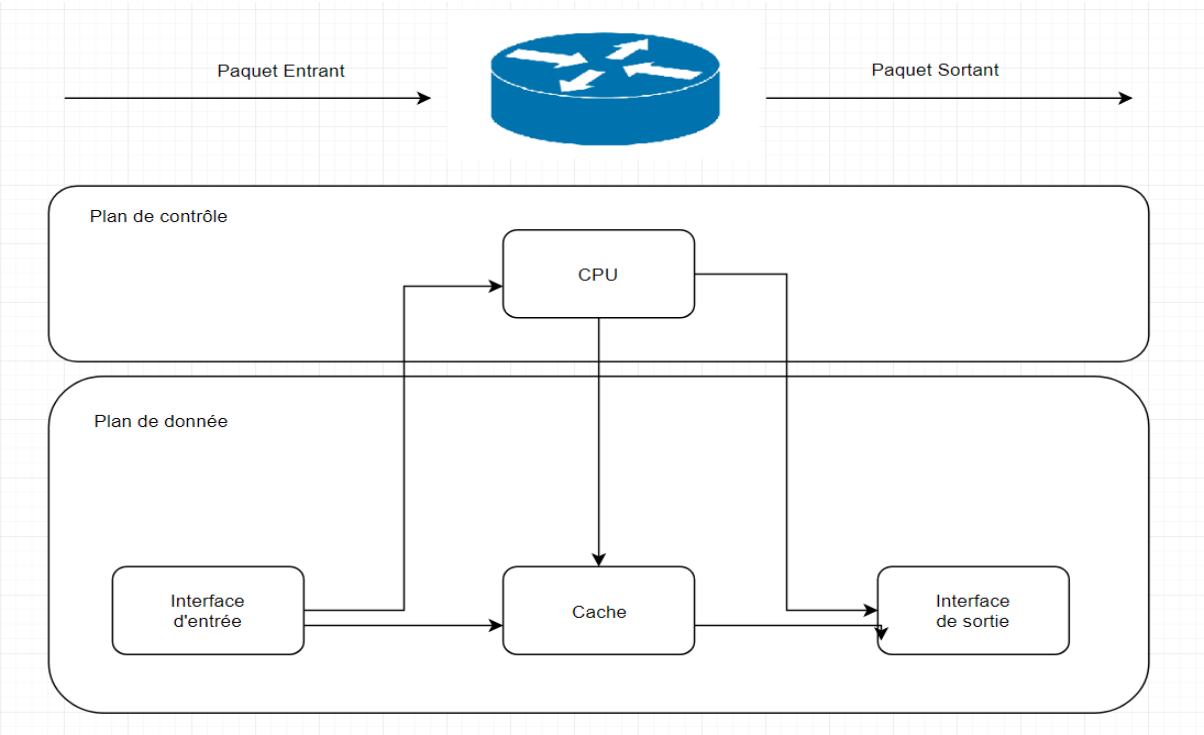

Figure 3-6 vue du mode Fast Switching sur les plans de fonctionnalité réseau

- Cisco Express Forwarding : C'est le mode le plus récent et aussi le plus rapide. CEF ajoute deux bases de données dans le plan de donnée. Les bases de données sont :
  - Forwarding Information Base ou FIB : contient des informations de routage de couche réseau
  - Table d'adjacence : contient des informations de couche liaison de donnée concernant les prochains sauts listés dans la FIB

CEF rempli la FIB à partir de la table de routage RIB et la table d'adjacence depuis le cache ARP. Contrairement à Fast Switching CEF n'a pas besoin de recourir au CPU pour la commutation du premier paquet, la table de commutation est créée à l'avance et non pas à la demande comme dans le cas de Fast Switching.

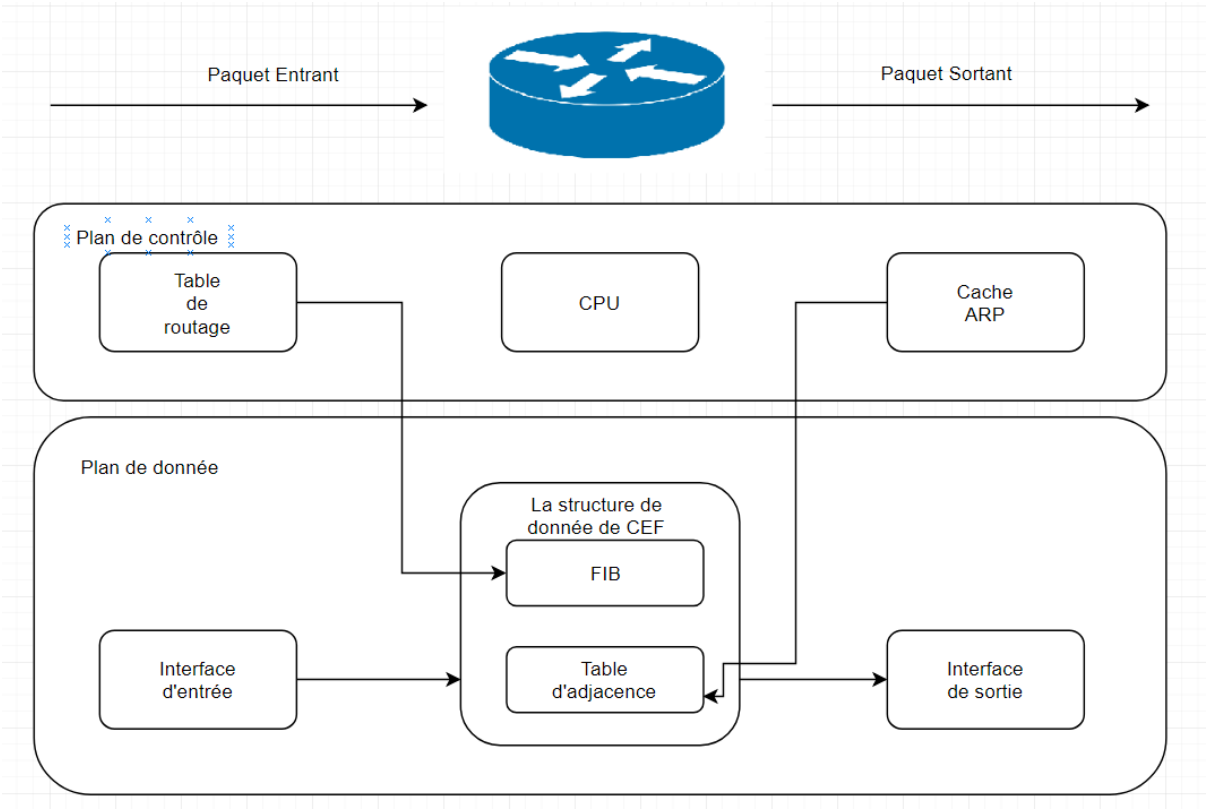

Figure 3-7 vue du mode CEF sur les plans de fonctionnalité réseau

CEF est essentiel pour le fonctionnement de MPLS, sans CEF, MPLS ne peut pas commuter de paquet. MPLS repose sur la structure et la logique de CEF pour le plan de donnée afin d'effectuer la commutation de paquet. MPLS ajoute à la structure de CEF une troisième base de données appelée MPLS Label Forwarding Instance Base ou LFIB. La LFIB est utilisé pour la commutation de paquets étiquetés. La LFIB est remplie à partir des informations de la RIB et de la LIB.

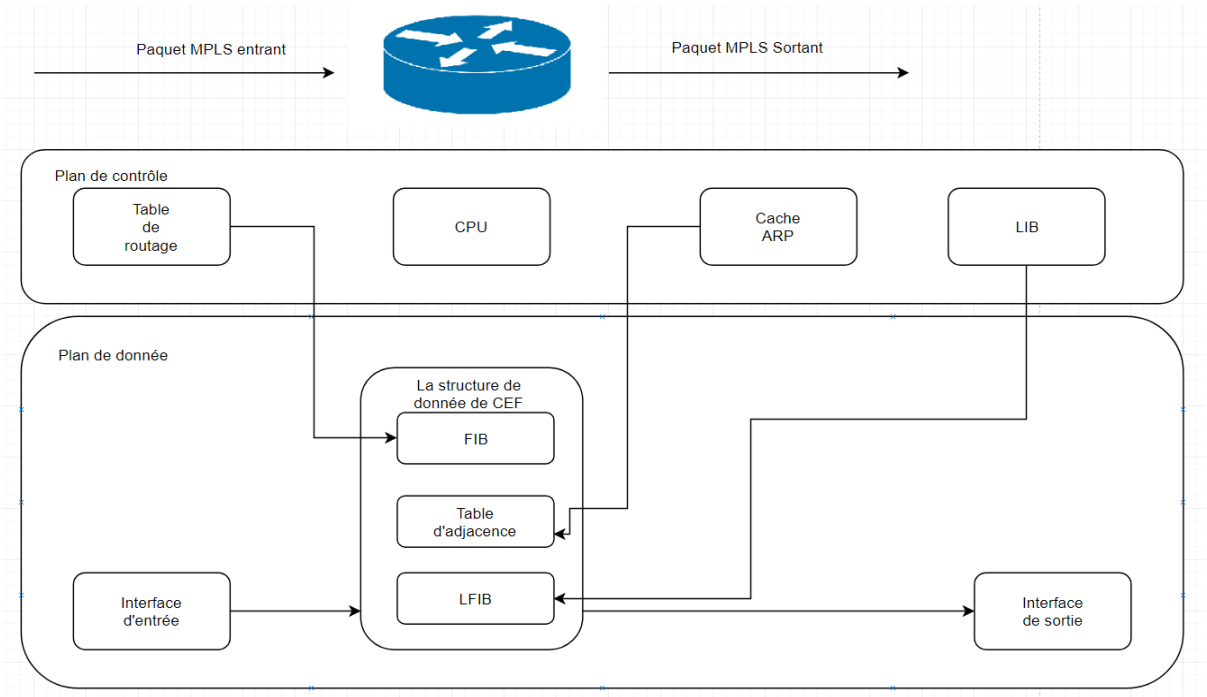

Figure 3-8 le modèle du plan de contrôle utilisé pour MPLS

F. La portée du sens des label dans un LSR

Les association FEC à label ont souvent un sens local au LSR qui les a créés. Mais on peut aussi avoir une séparation au sein du même LSR, les labels peuvent avoir un sens différent sur chaque interface. Un LSR peut associer le même label à deux FEC différentes sur deux interfaces différentes. On appelle cela en Anglais 'label scope per interface'. Ce mode d'espace de label est utilisé sur les interfaces ATM.

Pour les interfaces autre que les interfaces ATM on utilise des associations qui ont une portée sur toute la plateforme du LSR à l'exception des interface ATM. Ce mode d'espace est appelé 'label scope per plateform'

G. Notions de LSR Upstream et Downstream

Dans l'acheminement d'un paquet étiqueté on peut qualifier les LSR participant dans l'acheminement comme Upstream et Downstream selon la signification de l'étiquète apposée au paquet.

Pour identifier un LSR comme Upstream ou Downstream par rapport à une étiquète, l'étiquète doit représenter une FEC pour les paquets qui sont acheminé du LSR de départ Upstream vers le LSR de destination Downstream.

H. L'attribution des labels

L'attribution des labels est le fait d'associer un label à une FEC. Ces associations de FEC à label sont faites au niveau des LSR Downstream. Une fois les associations faites les LSRs Downstream annoncent ces labels aux LSRs Upstream.

I. Les modes de distribution des labels

Dans l'architecture MPLS on peut avoir deux modes de distribution des associations FEC à label, les implémentations peuvent ne fournir qu'un des modes de distribution selon les interfaces utilisées par le LSR ou les deux en même temps. Les LSR doivent se mettre d'accord sur le mode à utiliser.

- Unsolicited Downstream ou UD : Les LSRs annoncent leurs associations de label à leurs pairs LSR, automatiquement et périodiquement.
- Downstream-on-Demand ou DoD : MPLS autorise les LSRs Upstream à demander explicitement à un LSRs Downstream le label associé à une FEC particulière.

Les LSR utilisent en général le mode UD, le mode DoD est utilisé pour les interfaces ATM.

J. Les modes de conservation des labels

Les LSRs soumettent les labels annoncés par leurs pairs à certaines règles de conservation selon le prochain saut de ces FEC. Si un LSR reçoit une association FEC à label d'un LSR et le prochain saut n'est pas le LSR qui a annoncé cette association alors il existe deux modes de conservation pour ce cas :

- Liberal Label Retention ou LLR : Le LSR va conserver le label pour être utiliser dans le future si le routeur annonceur devient le prochain saut pour cette FEC.
- Conservative Label Retention ou CLR : Le label est immédiatement détruit.

L'usage de LLR permet une convergence rapide en cas de changement dans la topologie du réseau.

K. Les modes de contrôle des LSP

Les FEC sont souvent des préfixes d'adresse qui sont distribué dynamiquement par un protocole de routage. Il existe deux modes pour contrôler l'attribution des labels

- Idependent LSP Control : Les LSR créent une association FEC à label indépendamment les uns des autres. Chaque LSR crée une association FEC à label locale pour une FEC particulière et ce pour chaque FEC qu'il reconnait.
- Ordered LSP Control : Les LSR crée des associations FEC à label locales seulement pour les FEC pour lesquelles ils sont les Egress LSR ou pour lesquelles ils ont reçu une association du prochain saut pour cette FEC.

L'inconvénient avec Independent LSP control est que certains LSR termine l'attribution de leurs labels avant d'autres et commencent à commuter des paquets avant que le LSP ne soit totalement mis en place ce qui peut causer un mauvais traitement du paquet ou encore la perte de ce dernier.

11. La commutation des paquets étiquetés

La RFC 3031 (E. Rosen C. S., 2001) définit une structure appelée table NHLFE (Next Hop Label Forwarding Entry), les NHLFE sont utilisées pour commuter les paquets étiquetés, ils contiennent les informations suivantes :

- Le prochain saut pour le paquet.
- L'opération à effectuer (swap, push ou pop) sur le label stack.

La NHLFE peut contenir des informations optionnelles supplémentaires qui indiquent comment commuter les paquets, par exemple quel code de prochain entête utilisé dans l'entête de couche liaison de donnée.

Cisco IOS n'utilise pas de table NHLFE, il utilise à la place la Label Information Base (LIB) et La Label Forwarding Information Base (LFIB) pour la commutation de paquets. Ces tables contiennent des informations similaires à la table NHLFE.

A. Les opérations sur les labels

Lors de la réception d'un paquet un LSR Cisco IOS regarder sa table LFIB pour déterminer le prochain saut auquel commuter le paquet et quelles opérations effectuer, soit apposé, changer ou retirer un ou plusieurs labels au paquet.

a. Le comportent TTL lors des opérations sur les labels

Le label MPLS possède un champ TTL qui détermine la durée de vie du paquet, ou plutôt le nombre de saut que le paquet peut effectuer avant d'être détruit. Ceci assure que les paquets ne tournent pas indéfiniment dans le réseau MPLS en cas de boucle.

Le fonctionnement du TTL de MPLS est exactement le même avec celui d'IP. Quand un paquet IP rentre dans le réseau MPLS à travers un Ingress LSR, la valeur TTL présente dans le champ IP est décrémenté de 1 avant d'être copier dans le champ TTL du label MPLS qui va être apposé au paquet. Lors de la sortie du paquet étiqueté du réseau MPLS par un Egress LSR, la valeur du champ TTL du label MPLS est décrémenté de 1 avant d'être copier dans le champ TTL du paquet IP.

Pour les piles de label le comportement de label est le suivant selon l'opération effectué sur le label :

- Swap : lors de changement d'un label, on copie la valeur TTL du dernier label changé et on la décrémente de 1 pour être utilisé comme TTL du label remplaçant

- Push : Lors de l'ajout d'un label, on copie la valeur TTL du dernier label au sommet de la pile et on la décrémente de 1, pour être utilisé comme TTL du label à ajouter.
- Pop : Lors du retrait d'un label, on copie la valeur TTL du dernier label au sommet de la pile qui va être retiré et on la décrémente de 1, après le retrait du label cette nouvelle valeur TTL va être affectée au TTL du paquet au sommet de la pile.

Il est à noter que seulement la valeur TTL des labels au sommet de la pile est changé et non celle des labels sous-jacents.

b. La fragmentation des paquet MPLS

Les protocoles de couche liaison de donnée ne peuvent transporter qu'une quantité de donnée limitée en une seule trame. Certaines données excédent la quantité maximale (appelée Maximum Transmission Unit ou MTU), dans ces cas c'est le rôle de la couche réseau de fragmenter ces données afin de respecter la limite imposée par la couche liaison de donnée.

MPLS ajoute son entête à l'entête de couche réseau, ce qui engendre une légère augmentation à la quantité de donnée et par conséquent il y a possibilité de dépasser le MTU. Un Label MPLS est de 4 octets et un paquet peut avoir plusieurs labels. L'addition de la taille du paquet et tous ses labels ne doit pas dépasser l'MTU.

En général, les nœuds utilisateurs IP qui n'implémentent pas 'Path MTU Discovery' envoient des paquet IP qui ne dépassent pas 576 octets. Le MTU de la majorité des protocoles de couche liaison de donnée aujourd'hui est de 1500 octets ou plus. La probabilité qu'un paquet IP ait besoin d'être fragmenté même après l'ajout de la pile de label est très faible. Certain nœuds utilisateurs vont générer des paquets IP de 1500 octets quand l'adresse source et destination sous dans le même sous-réseau (partagent la même liaison de donnée), ces paquets ne passeront pas à travers un routeur donc ils n'auront pas besoin d'être fragmenté. Certains nœuds utilisateurs vont générer des paquet IP de 1500 octets quand l'adresse source et destination sous de la même classe réseau. Ceci est le seul cas où il y a risque de fragmentation quand ces paquets sont étiquetés avec un ou plusieurs labels. (E. Rosen D. T., 2001)

L'opération de fragmentation de paquet MPLS est assez similaire à la fragmentation de paquet IP qui ont le bit DF (don't fragment) à 0, quand un paquet est trop large pour être envoyé par la liaison de donnée il doit être fragmenté. Quand un LSR reçoit un paquet étiqueté avec une taille supérieure à la MTU de l'interface de sortie pour ce paquet il retire la pile de labels du paquet pour ensuite le fragmenté, une fois fragmenter la pile de labels doit être ajouté à chaque fragment avant d'être commuté. Si le bit DF est à 1 alors le LSR ne peut pas fragmenter le paquet par conséquent il détruit le paquet et envoie un message d'erreur ICMP à l'auteur du paquet.

La fragmentation affecte négativement les performances du réseau, elle doit être évitée si possible.

i. Maximum Initially Labeled IP Datagram Size

La RFC 3032 défini une taille maximale d'un paquet étiqueté appelée 'Maximum Initially Labeled IP Datagram Size'. C'est un entier qui peut avoir une valeur nulle ou positive.

Chaque LSR capable de :

- Recevoir un paquet IP étiqueté.
- Ajouter une pile de labels au paquet.
- Commuter un paquet étiqueté

Doit supporter la configuration du paramètre 'Maximum Initially Labeled IP Datagram Size'. Si la valeur est nulle, alors elle n'aura aucun effet.

Si la valeur de ce paramètre est positive, alors elle sera utilisée de la façon suivante si un paquet répond à ces conditions :

a) Le paquet n'est pas étiqueté
b) Le paquet IP ne possède pas le bit DF à 1 dans son entête, le bit DF empêche un paquet d'être fragmenté.
c) Le paquet a besoin d'être étiqueté avant d'être commuté.
d) La taille du paquet avant d'être étiqueté dépasse la valeur du paramètre.

Si le paquet répond à toute ces conditions alors

a) Le paquet IP doit être fragmenté, chacun de ses fragments ne doit pas dépasser la valeur du paramètre.
b) Chaque fragment doit être étiqueté et commuté.

(E. Rosen D. T., 2001)

ii. Cisco IOS Maximum Receive Unit

Cisco IOS utilise un paramètre appelé Maximum Receive Unit ou MRU, Un LSR informe un autre LSR de la taille maximale d'un paquet étiqueté qui peut être commuter sans avoir besoin d'être fragmenté pour une certaine FEC.

Les besoins d'une FEC à une autre peuvent différer. Et c'est souvent le cas avec MPLS VPN et MPLS-TE, une FEC peut avoir besoin de plusieurs labels alors qu'une autre a besoin d'un seul. Le MRU sert à identifier les besoins spécifiques d'une FEC et à les communiquer aux LSR Upstream. Ceci évite d'avoir recours à la fragmentation plus tard dans le LSP afin d'ajouter des labels à la pile.

iii. Path MTU Discovery

Lors de l'envoi de large quantité de donnée d'un nœud utilisateur à un autre nœud utilisateur, ces données sont envoyées en une série de paquet IP. Il est assez préférable que ces paquets n'aient pas à être fragmenté à travers le chemin vers leur destination. On appelle 'Path MTU ou PMTU' la taille maximale du paquet qui peut être acheminer à travers un chemin sans besoin de fragmentation. La valeur du Path MTU est celle du MTU le plus petit saut entre deux nœuds à travers le chemin.

Le protocole IP ne possède pas de mécanisme pour permettre à un nœud utilisateur de découvrir le PMTU d'un chemin quelconque. La pratique actuelle est d'utiliser comme PMTU le plus petit de 576 octets ou l'MTU du premier saut, pour chaque destination qui ne partage pas la même liaison de donnée.

c. Déterminer l'encapsulation de liaison de donnée à utiliser lors de la sortie de paquets du réseau MPLS

Lors de la sortie d'un paquet du réseau MPLS tous ses labels sont retirés, nous savons que les protocoles de couche inférieurs indiquent toujours à la couche supérieure le type de donnée ou le protocole de couche supérieure qu'elles encapsulent. Dans un réseau MPLS le champ de la couche liaison de donnée qui indique le prochain protocole indique une encapsulation MPLS. À la sortie du paquet du réseau il faut changer le champ indiquant le protocole suivant pour identifier le protocole de couche réseau utilisé, cependant les labels MPLS ne possèdent aucun champ pour identifier explicitement le protocole de couche réseau suivant.

Le premier label apposé par l'Ingress LSR sert à identifier le protocole utilisé en couche réseau. Quand un label est apposé à un paquet de couche réseau, le label doit être utilisé uniquement pour un protocole de couche réseau particulier, ou un ensemble spécifique de protocole de couche réseau à condition qu'ils soit possible de les distinguer par une inspection de l'entête de couche réseau. Durant le transite chaque fois que la valeur du label est remplacée, la nouvelle valeur doit aussi répondre aux mêmes critères. Si les conditions ne sont pas satisfaites, l'Egress LSR qui retire tous label MPLS ne pourra pas identifier le protocole de couche réseau. (E. Rosen D. T., 2001)

IV. Distribution des associations FEC à label

Le fonctionnement fondamental de MPLS repose sur l'étiquetage des paquets pour leur commutation. Deux LSRs doivent se mettre d'accord sur la signification des labels utilisé pour commuter des paquets entre eux. Pour parvenir à cet accord un LSR associe un label à une FEC puis doit informer son pair de cette association, ce processus est appelé 'distribution de label'.

Les méthodes de distribution de label ont été citées précédemment, cette section va se focalise sur la distribution de label avec LDP et BGP

1. Label Distribution Protocol ou LDP

LDP a été crée pour distribuer les labels MPLS. C'est un ensemble de procédures et de messages avec lesquels un LSR établi des LSP en associant des informations de routage de couche réseau à des information de commutation de couche liaison de donnée, puis en associant une FEC avec chaque LSP crée.

Les LSR qui utilise LDP et s'échangent des associations FEC à label sont appelés 'pairs LDP', on appelle leur relation 'session LDP'.

Pour notre étude on s'intéresse à LDP sur les liaisons de donnée point à point tel que (point-to-point protocol et Ethernet). Et pour la portée du sens des label (espace label) c'est par plateforme.

A. Les messages LDP

LDP utilise quatre types de messages :

- Messages de découverte : utilisés pour annoncer la présence d'un LSR et comme mécanisme keep-alive
- Messages de session : utilisés pour établir, maintenir ou terminer une session LDP avec un pair LDP
- Message d'annonce : utilisés pour créer, changer ou retiré des association FEC à label.
- Message de notification : utilisés pour signaler des erreurs ou pour fournir des transmettre des informations consultatives.

La structure du message LDP est la meme pour tous les types de messages (L. Andersson, 2007) :

Les messages LDP commencent par l'entête LDP suivi par le corps du message, voici le format paquet de l'entête LDP :

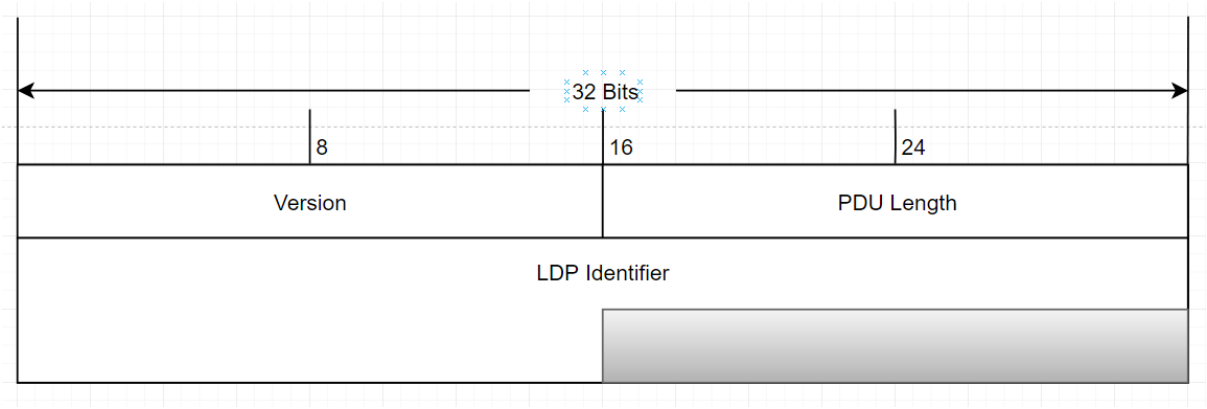

Figure 2-1 format paquet de l'entête LDP

- Version : Champ de deux octets, c'est un entier non signé qui indique la version du protocol.
- PDU Length : Champ de deux octets, c'est un entier qui spécifie la longueur totale en octets du message, les champs 'Version' et 'PDU Length' sont exclus.
- LDP Identifier : Champ de six octets, ce champ sert à identifier l'espace label du LSR qui a envoyé le message. Les quatre premiers octets identifient le LSR, cette valeur doit être unique dans le réseau. Les deux derniers octets, identifie l'espace label du LSR émetteur. Pour un espace label par plateforme les 16 bits doivent être à 0.

Tous les messages LDP utilisent ce format paquet pour le corps du message :

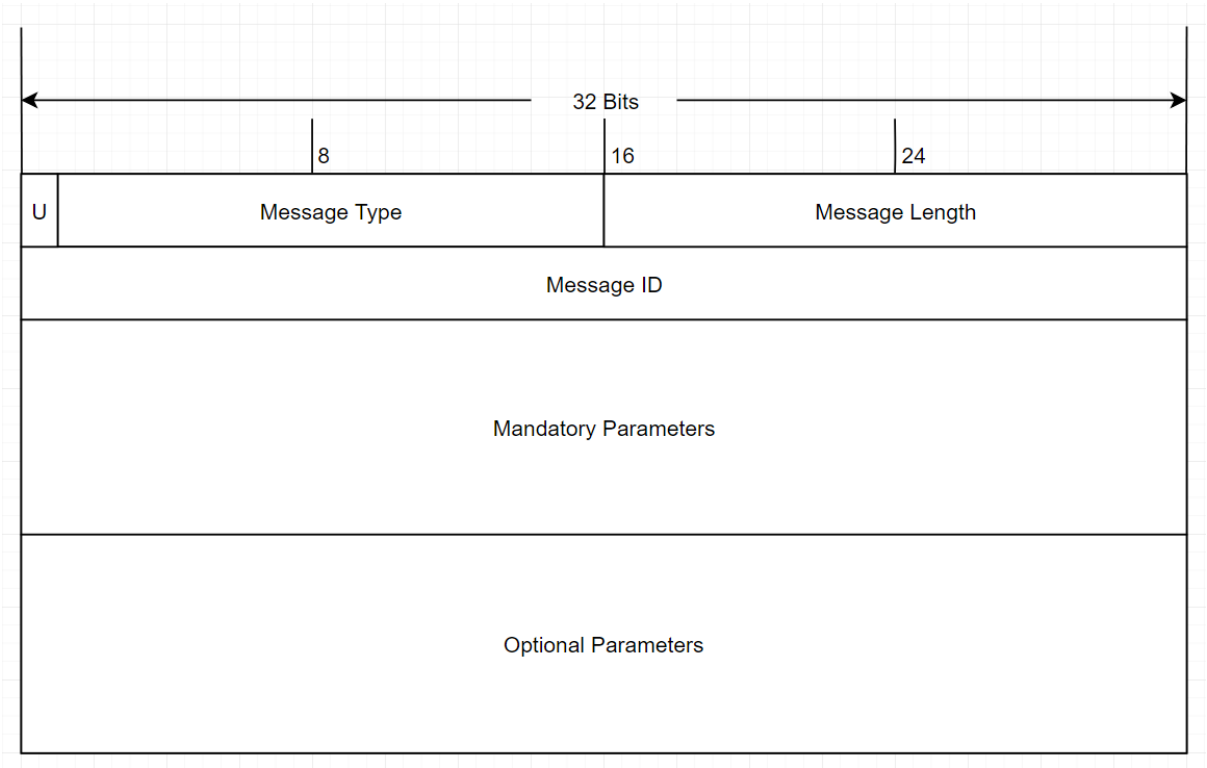

Figure 4-2 Format paquet des messages LDP

- Bit U : C'est le bit unknown (inconnu), lors de la réception d'un message inconnu, si le bit U est à 0 alors un message de notification est envoyé à l'auteur du message ; sinon si le bit est à 1 alors le message est ignoré sans rien envoyé à l'auteur.

- Message Type : champ de 15 bits, identifie le type du message. Le tableau suivant montre les codes type pour chaque message

Type de messages LDP (L. Andersson, 2007) :

Code type	Nom du message
0x0001	Notification
0x0100	Hello
0x0200	Initialization
0x0201	Keep-Alive
0x0300	Address
0x0301	Address Withdraw
0x0400	Label Mapping
0x0401	Label Request
0x0402	Label Withdraw
0x0403	Label Release
0x0404	Label Abort Request
[0x3E00-0x3EFF]	Vendor-Private
[0x3F00-0x3FFF]	Experimental

- Message Length : Champ de deux octets, indique la longueur en octet du corps message avec les champs U, Message Type, Message Length exclus.
- Message ID : Champ de quatre octets, utilisé pour identifier le message. Quand un LSR envoie un message de notification par rapport à un message LDP, il ajoute le Message ID du message LDP en question dans le champ Message ID du message de notification.
- Mandatory Parameters : Champ à longueur variable, contient des paramètre requis pour le message. Certains messages peuvent ne pas avoir de paramètres requis.
- Optional Parameters : Champ à longueur variable, contient un ensemble de paramètre optionnels pour le message.

a. Message Hello

Le mécanisme 'LDP Discovery' permet à un LSR de découvrir des pairs potentiels. Il existe deux variantes du mécanisme 'LDP Discovery' :

- Une variante standard. Permet de découvrir des LSR qui sont sur la même liaison de donnée.
- Une variante étendue. Permet la localisation d'un LSR qui n'est pas sur la même liaison de donnée.

Le mécanisme envoie périodiquement des messages Hello sur les interfaces du LSR. Les messages Hello sont envoyés comme des paquets UDP adressées aux port 'LDP Discovery' 646 à l'adresse multidiffusion de tous les routeurs sur la liaison de donnée (224.0.0.2).

Le message LDP Hello change le champs (Mandatory Parameters) du message de base par le champ suivant :

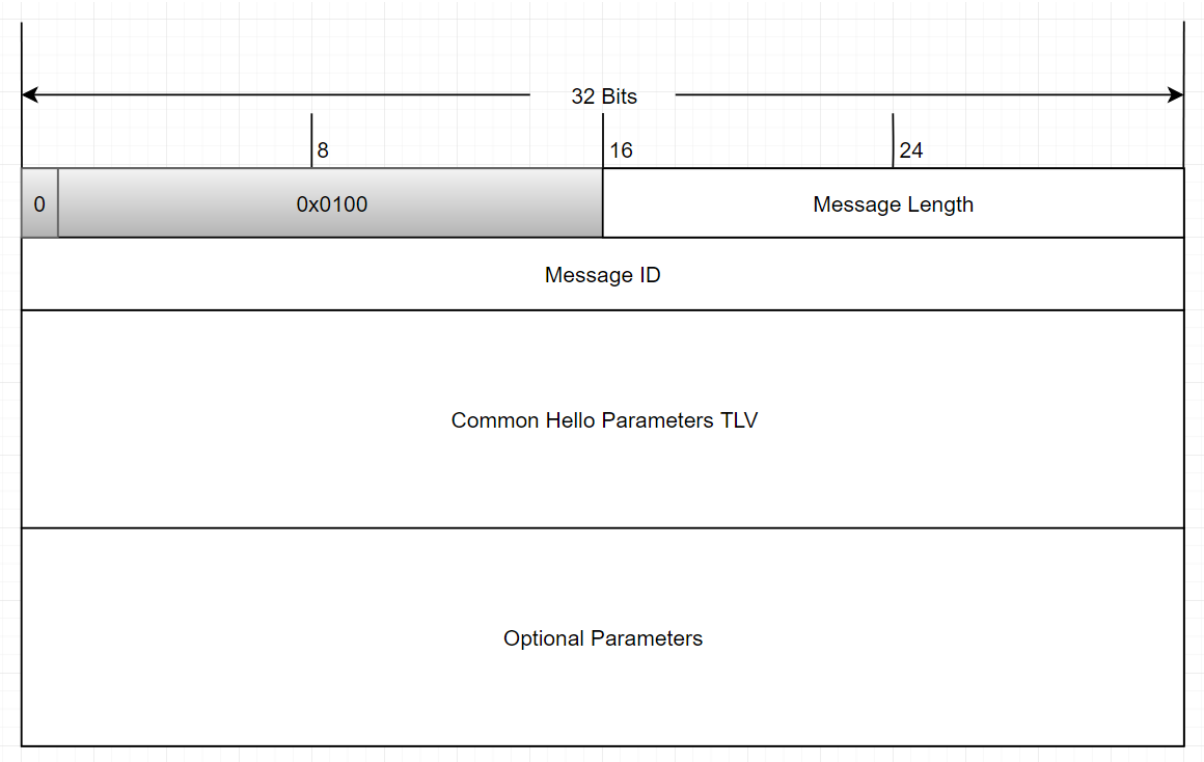

Figure 4-3 champs du message LDP Hello

- Common Hello Parameters TLV : Champ de huit octets, spécifie les paramètres communs à tous les messages Hello. Ce champ se divise en plusieurs champs comme suit :

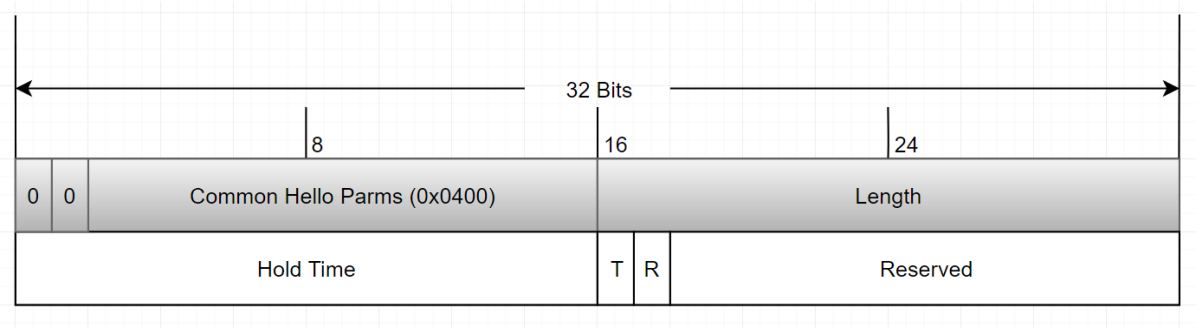

Figure 4-4 Contenu du champ Common Hello Parameters TLV

  - Hold Time : Champ de deux octets, c'est un entier non signé qui indique un délai en secondes qu'un LSR devrait attendre sans recevoir de message Hello d'un autre LSR avant de le déclarer mort.
  - T : Bit Targeted Hello. Si le bit est à 1 cela indique que c'est un Hello ciblé.
  - R : Bit Request Send Targeted Hello. Si le bit est à 1 alors ce message est une requête au destinataire pour envoyer des messages Hello ciblés périodiquement à la source du message.
  - Reserved : 14 bits, réservés.
- Optional Parameters : Champ à longueur variable. Spécifie des paramètres optionnels

Les messages LDP Hello servent à créer et maintenir des contiguïtés Hello 'Hello Adjacency' avec d'autres LSR. Les contiguïtés Hello ont un temps de vie 'Hold Time', Le temps de vie d'une contiguïté

est réinitialisé à chaque message Hello. Si un message Hello valide est reçu et qu'il n'y a pas de contiguïté qui lui est associée alors une nouvelle contiguïté est créée.

Un LSR peut être configuré pour former une session LDP avec un LSR qui ne partage pas une même liaison de donnée. Pour cela l'LSR devra utiliser des messages Hello ciblés pour former une contiguïté avec cet LSR. Un LSR peut accepter des Hello ciblés si une des deux conditions suivantes est réalisée :

- L'LSR à été spécifiquement configuré pour accepter les messages Hello ciblés
- L'LSR à été configuré pour envoyer des LSR ciblés à la source du message Hello ciblé qu'il a reçu.

b. Message d'initialisation

Pour établir une session LDP utilise des messages d'initialisation. Une fois la contiguïté établie entre deux LSR par 'LDP discovery', ces LSR vont tenter d'établir une session LDP entre eux. L'établissement d'une session LDP passe par deux étapes

- L'établissement de la connexion couche transport :

Les LSR vont déterminer quelle adresse utiliser pour la connexion TCP, pour cela LDP regarde l'adresse indiquée dans les paramètres optionnels pour la connexion TCP, si aucune adresse n'est spécifiée alors LDP utilisera l'adresse source du message Hello pour créer la connexion TCP.

LDP doit déterminer quel LSR aura le rôle maitre ou esclave. Pour cela une comparaison des adresses de connexion TCP doit être faites, l'LSR avec la plus grande adresse sera le maitre. L'LSR maitre devra établir la connexion TCP au port LDP (646) du LSR esclave.

- Initialisation de la session :

Une fois la connexion de couche transport établi les LSR doivent négocier les paramètres de la session, les paramètres négociés sont : la version du protocole LDP, La méthode de distribution des labels, paramètres spécifiques à la liaison de donnée. (Pour notre étude, on s'intéresse seulement aux sessions sur des liaison de donnée point à point et les espace de label par plateforme).

Le message LDP d'initialisation change le champs (Mandatory Parameters) du message de base par le champ suivant :

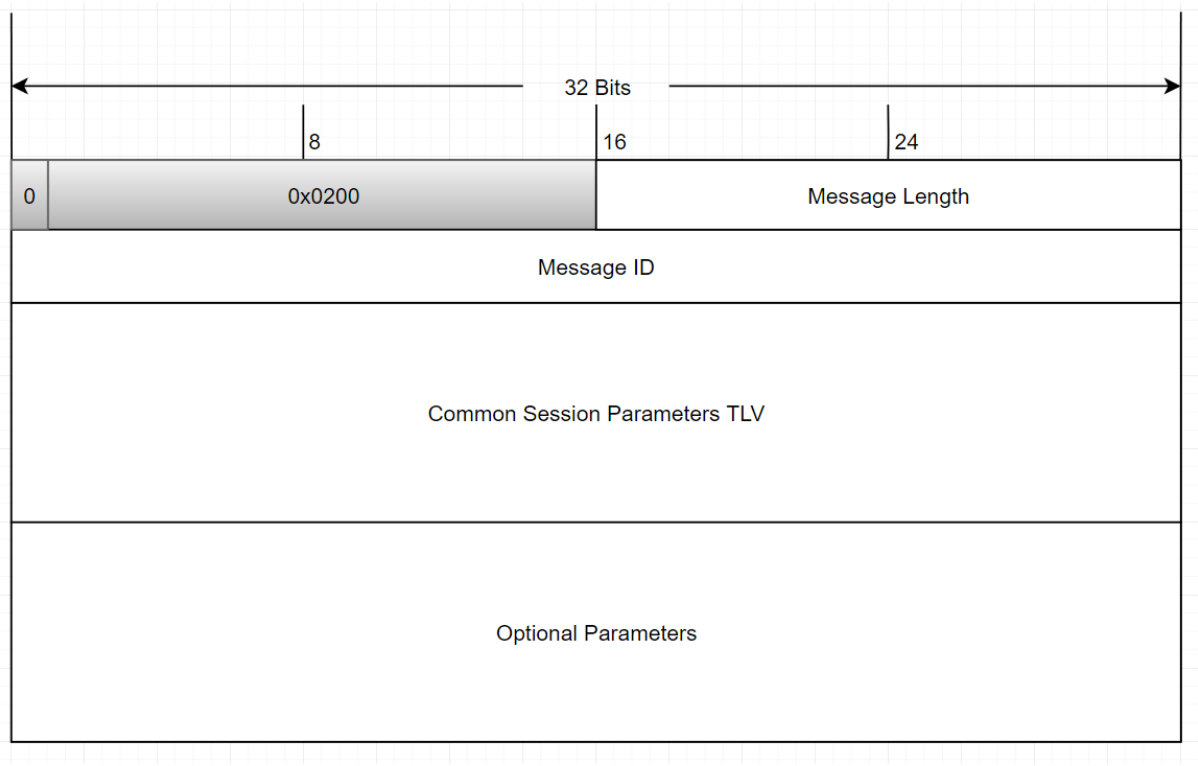

Figure 4-5 Champ du message LDP d'initialisation

- Common Session Parameters TLV : Champs de 18 octets, spécifie les paramètres proposés par LSR à l'origine du message. Ce champ contient les champs suivants :

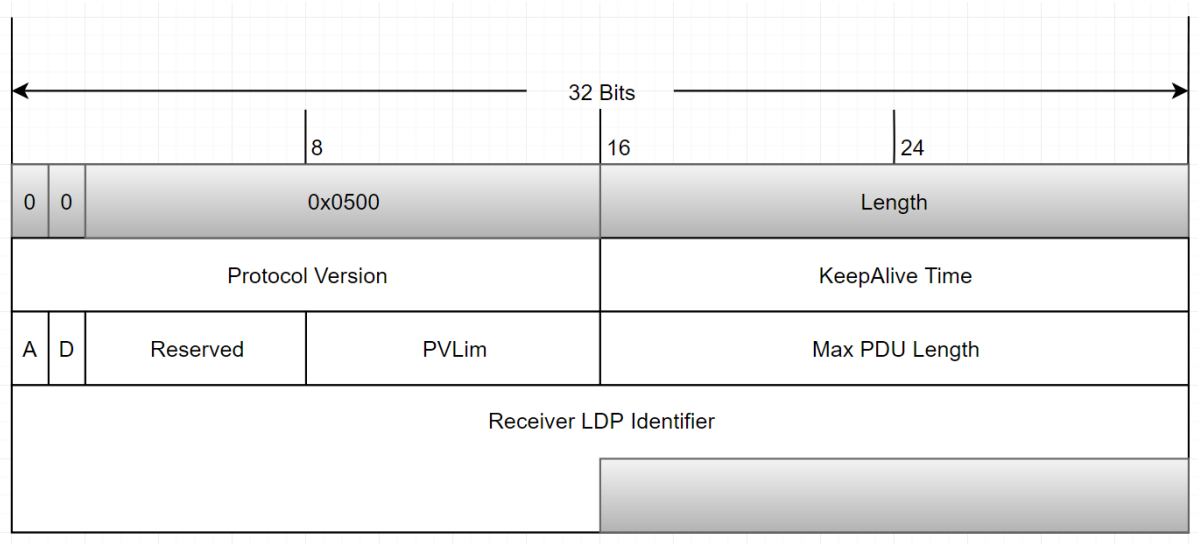

Figure 4-6 contenu du champ Common Session Parameters TLV

- Protocol Version : Champ de deux octets, indique la version du protocole LDP.
- KeepAlive Time : Champ de deux octets, c'est un entier non signé qui indique le temps en secondes du keep-alive proposé par l'auteur du message
- A : Bit Label Advertisement Discipline, indique le mode de distribution des labels
  - 1 : Downstream On Demand (DoD)
  - 0 : Downstream Unsolicited (DU)

- D : Bit Loop Detection, indique si la détection de boucle avec 'Path Vector' est activée, 0 pour désactivée, 1 pour activée
- PVLim : Path Vector Limit. Champ d'un octet, indique la longueur maximale du 'Path Vector'
- Max PDU Length : Champ de deux octets, c'est un entier non signé qui indique la taille maximale autorisée en octet pour un PDU LDP de cette session.
- Receiver LDP Identifier : Identifie l'espace label du destinataire du message, Cet espace label est constitué de quatre octets qui identifient l'LSR et deux octets identifient l'espace label de cet LSR. L'SR compare la combinaison du champ 'LDP Identifier' présent dans l'entête du message et ce champ avec ses contiguïtés. Pour initialiser la session une correspondance doit être trouvée, sinon la session sera rejetée.

- Optional Parameters : Champ à longueur variable. Spécifie des paramètres optionnels.

La RFC 5036 décrit le comportement LDP lors de la négociation d'une session avec un automate d'état. La figure 4-7 présente cet automate

Les états de l'automate sont :

- Non Existent : Dans cet état la session TCP n'a pas été établie.
- Initialized : La session TCP a été établie avec succès, un rôle (maitre ou esclave) est attribué à l'LSR.
- OpenSent : Cet état concerne seulement l'LSR avec le rôle de maitre. Un message d'initialisation a été envoyé et une réponse est attendue.
- OpenRec : Dans cet état les deux LSR négocie les paramètres d'initialisation.
- Operational : Dans cet état LDP est totalement fonctionnel.

Les événements qui causent le changement d'état d'initialisation, représentés comme des arcs dans la figure 4-7 sont expliqués dans le tableau suivant :

Événement	Description
E1	Session TCP a été établie avec succès.
E2	Session TCP expirée ou réception de tout message LDP autre que le message LDP d'initialisation.
E3	Pour le rôle esclave seulement : Réception d'un message LDP d'initialisation acceptable, envoi d'un message d'initialisation LDP suivi d'un KeepAlive LDP.
E4	Pour le rôle maitre seulement : Envoi d'un message LDP d'initialisation.
E5	Pour le rôle maitre seulement : Réception d'un message LDP d'initialisation, envoi d'un Keepalive.
E6	Session TCP expirée ou réception de tout message LDP autre qu'un KeepAlive
E7	Session TCP expirée ou réception de tout message LDP autre que le message LDP d'initialisation.
E8	Réception d'un Keepalive.
E9	Réception de tout message LDP
E10	Réception d'un message LDP Shutdown.

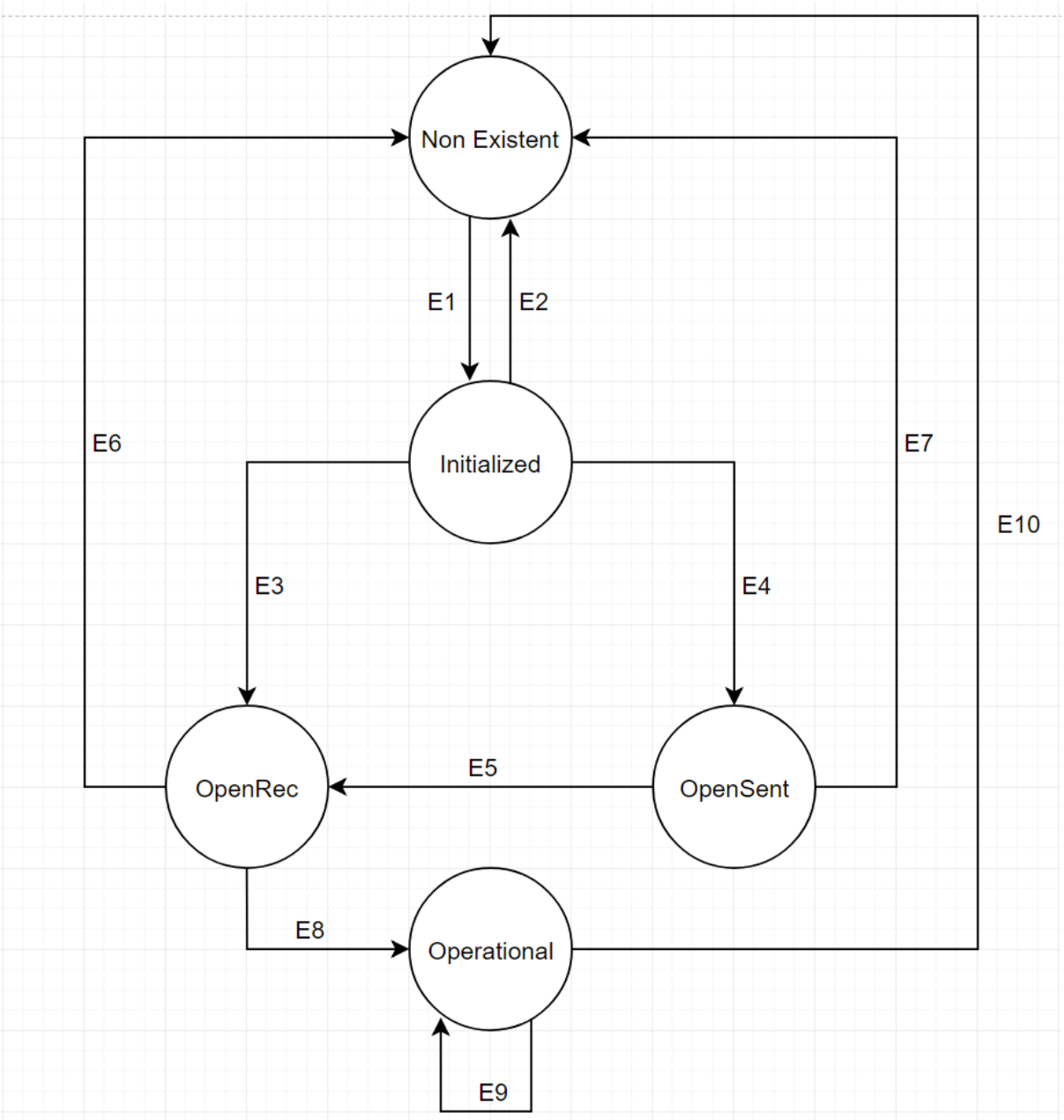

Figure 4-7 automate d'état pour l'initialisation d'une session LDP

c. Message KeepAlive

Les messages KeepAlive sont un mécanisme de surveillance de l'état de connexion d'une session LDP. Pour maintenir session LDP, il faut réinitialiser les compteur Hold Time des LSR. Tous message LDP envoyé à un pair réinitialise le compteur. Les messages KeepAlive sont envoyé périodiquement ce qui réinitialise le compteur à chaque fois. Les messages KeepAlive sont formatés comme suit :

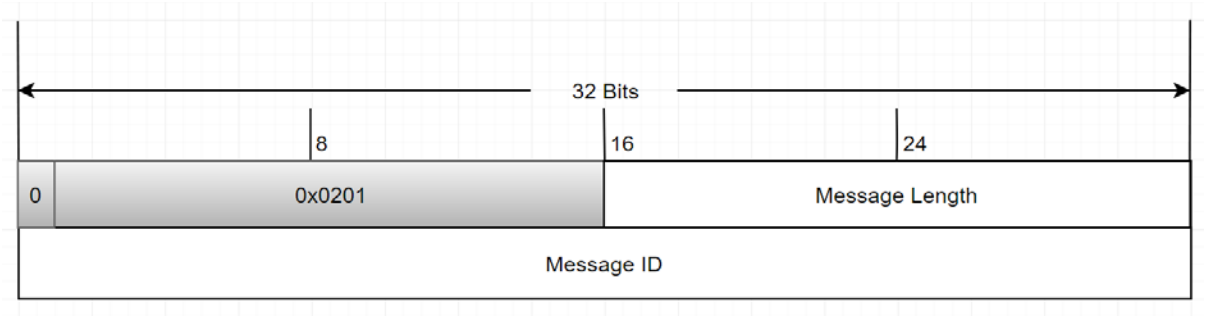

Figure 4-8 Format paquet d'un message KeepAlive

Le message LDP KeepAlive est constitué juste de l'entête LDP.

d. Messages LDP Address

Les message LDP Address sont utilisés par les LSR pour annoncer l'adresse de leur interface. Une fois la session LDP initialisée, avant de commencer à annoncer ou demander des associations de labels, les LSRs doivent d'abord annoncer les adresses de leurs interfaces. Pour chaque nouvelle interface activée un LSR doit annoncer l'adresse de cette nouvelle interface.

LDP sauvegarde les associations de label apprises dans une table appelée 'Label Information Base' ou LIB. Chaque entrée de la LIB pour un préfixe d'adresses est associée à un couple (LDP Identifier, Label). Il peut y avoir plusieurs labels dans LIB pour un même préfixe, pour déterminer lequel utiliser lors de la commutation de paquets MPLS le LSR a besoin de connaitre les adresses des interfaces des LSR qui lui sont voisin, avec ces adresses il est possible de savoir quel LSR est le prochain saut pour le préfixe.

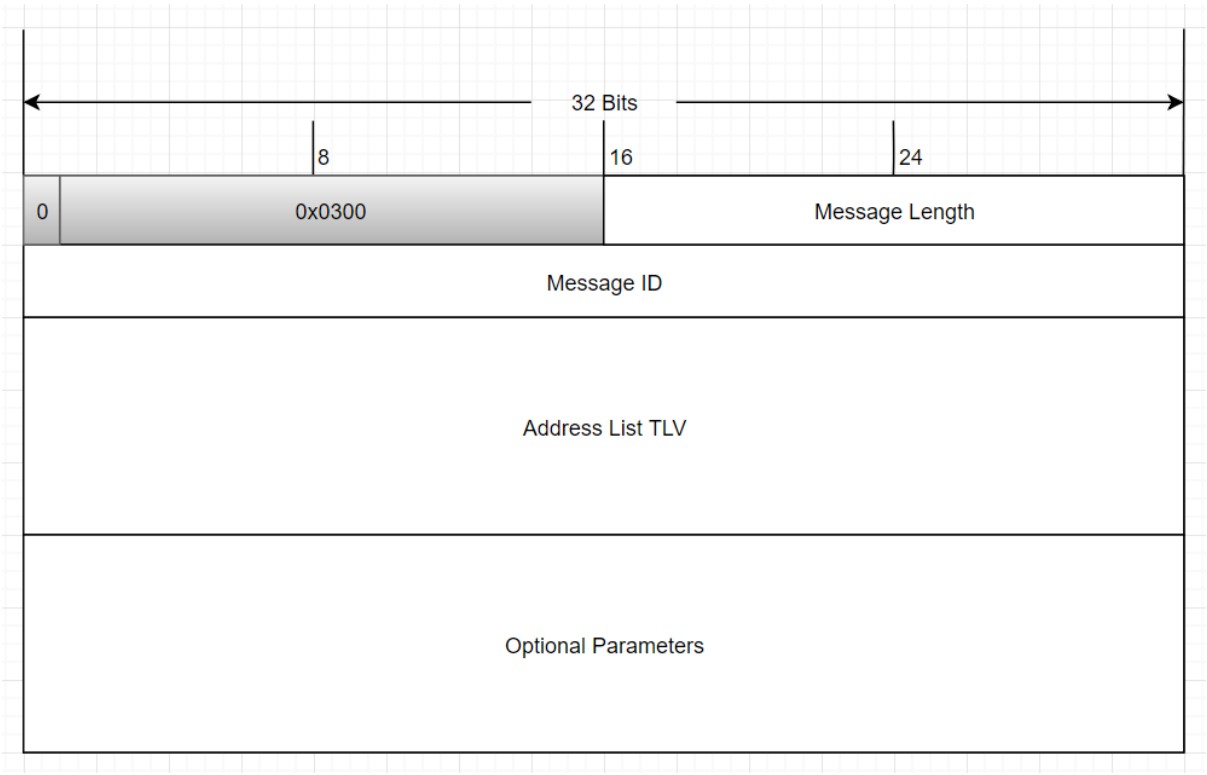

Figure 4-9 Format paquet d'un message LDP Address

V. Conception du réseau

La conception du réseau d'un fournisseur de service qui répond aux exigences d'aujourd'hui est une tâche particulièrement complexe. L'élaboration de la topologie du réseau ainsi que ainsi que le choix des technologies à utiliser pour répondre à des besoins spécifiques est du ressort des architectes réseau. La complexité de la conception du réseau ne réside pas dans le grand nombre de nœuds utilisés mais dans la conception du plan de contrôle (les protocoles de couche réseau à utiliser, la division des domaines de routage, définition des relations et interactions entre protocoles de routage).

L'architecte réseau se doit d'avoir une compréhension parfaite des technologies qu'il utilise lors de la conception mais aussi une expérience approfondie du comportement de ces technologies sur le terrain (les résultats pratiques peuvent différés des expectations théoriques).

La conception du réseau d'un fournisseur de service est au-delà de notre étude. Nous allons utiliser une architecture proposée par notre encadreur de stage. L'architecture est très simplifiée par rapport à un environnement de production, cette simplification a pour but d'enlever toute complexité non nécessaire, l'architecture offre le nécessaire pour réaliser notre étude.

1. L'architecture physique

L'architecture physique montre les différents nœuds du réseau ainsi que les liens physiques qui les relient.

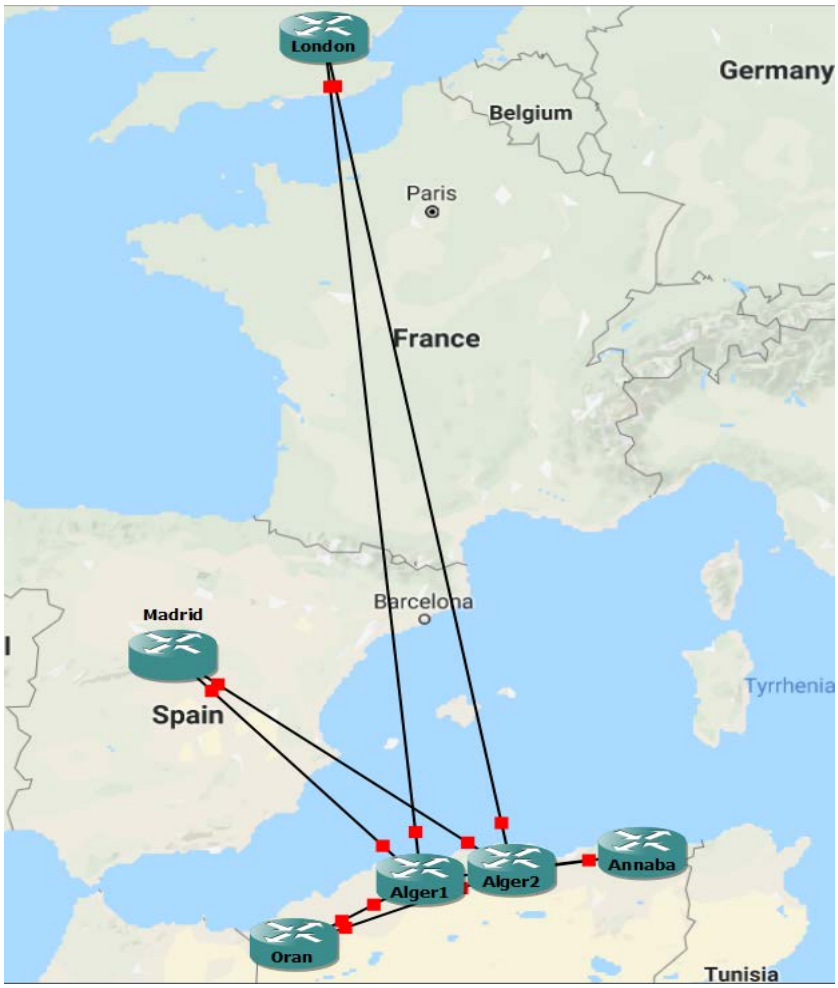

Figure 5-1 représentation géographique du backbone du réseau

Le backbone du réseau est constitué de six routeur principaux situés dans des zones géographiques éloignées. Les deux routeurs situés à Alger sont le cœur du réseau, chaque routeur possède une liaison vers chacun des deux routeurs situés à Alger.

2. Plan de contrôle

L'architecte réseau doit choisir un type de plan de contrôle à utiliser, il existe deux types de plan de contrôle :

- Plan de contrôle distribué : c'est le type standard, chaque routeur possède son propre plan de contrôle qui lui permet de prendre des décisions de routage pour le trafic qui le traverse.
- Plan de contrôle centralisé : le plan de contrôle qui permet aux routeurs de prendre des décisions de routage est situé dans un équipement dédié, les informations du plan de contrôle sont ensuite communiquées aux autres routeurs automatiquement. L'avantage avec ce type de plan de contrôle est que l'ingénieur n'aura pas besoin de configurer chacun des routeurs individuellement.

Pour notre étude nous utiliserons un plan de contrôle distribué.

3. Identification des routeurs

Les routeurs sont tous identifiés par un identifiant de quatre octets uniques, qui va correspondre à une adresse IP utilisée sur une interface logique pour chaque routeur.

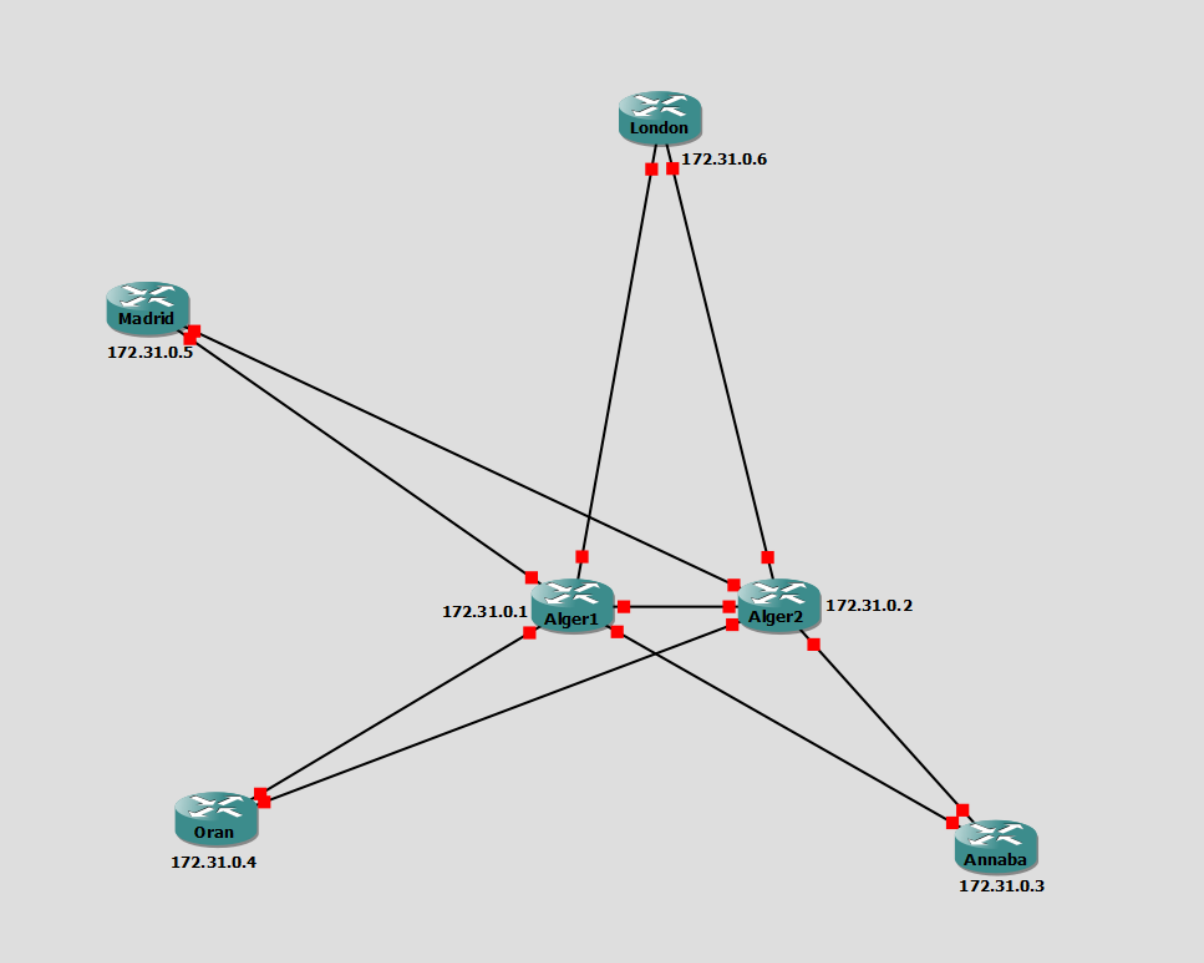

Figure 5-2 Identification des routeurs du backbone

Routeur	Identifiant
Alger1	172.31.0.1
Alger2	172.31.0.2
Annaba	172.31.0.3
Oran	172.31.0.4
Madrid	172.31.0.5
London	172.31.0.6

Les routeurs utiliseront leur identifiant pour s'identifier auprès des autres routeurs du backbone lors des communication internes.

4. Routage Interne

Les routeurs du backbone utiliseront OSPF pour propager les informations de routage internes.

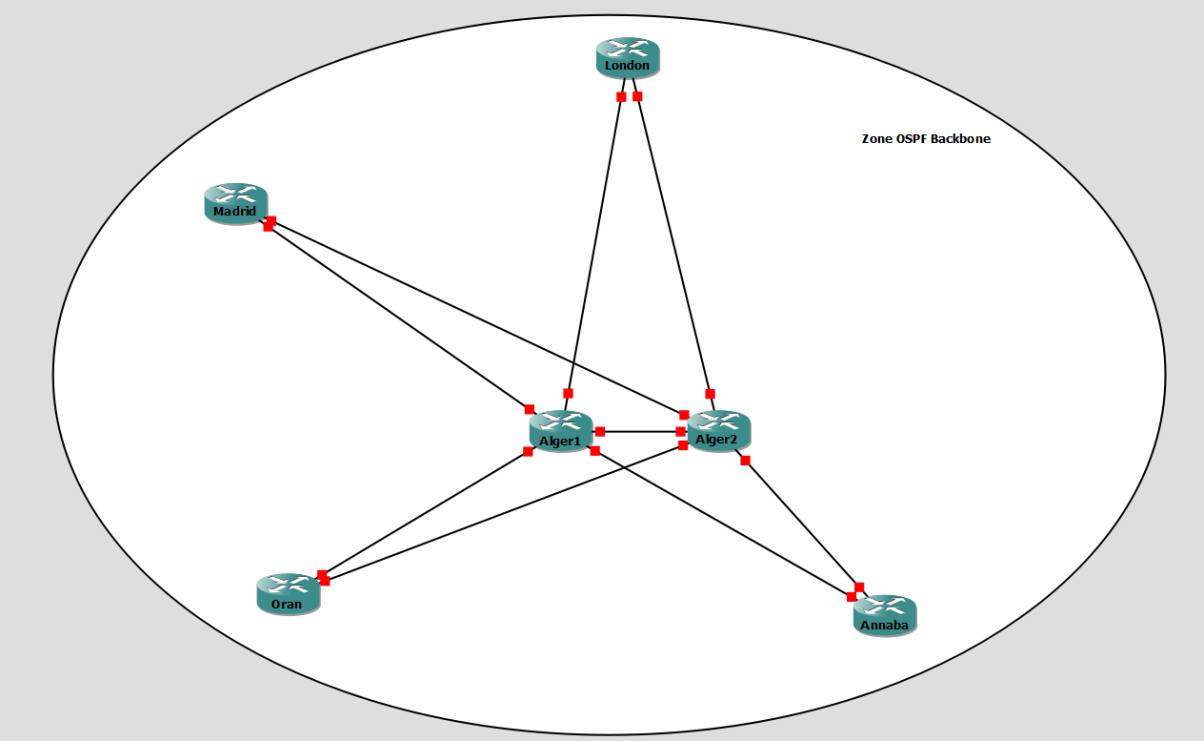

Figure 5-3 La zone backbone OSPF

Tous les routeurs font parti de la zone backbone d'OSPF 'Area 0'.

5. Relations IBGP et Route Reflector

Pour le partage d'informations de routage externes le backbone utilise IBGP, tous les routeurs ont une relation IBGP avec Alger1 et Alger2. Alger1 sera Route Reflector pour se cluster, Alger2 sera un Backup pour Alger1, en cas de panne dans le routeur Alger1, Alger2 prendra le relai et deviendra Route Reflector pour le cluster.

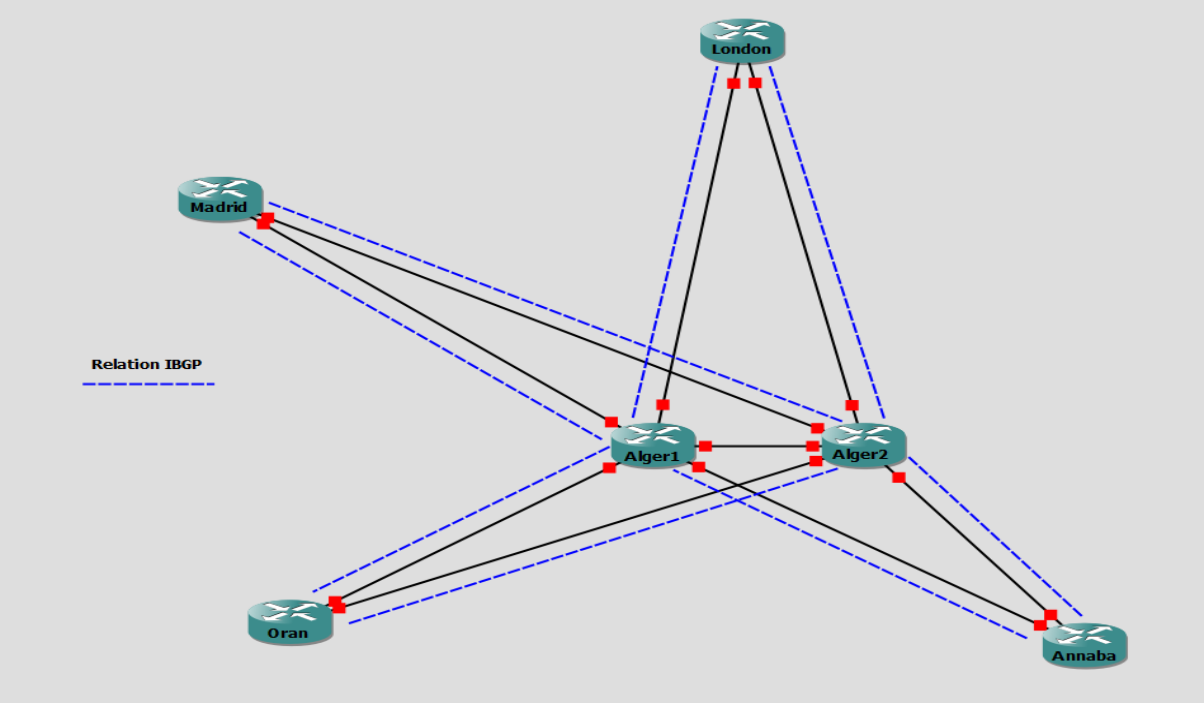

Figure 5-4 Relation IBGP entre les routeurs du Backbone

6. Les Routeurs de bordure

Les routeurs Madrid et London seront des routeurs de bordure (Provider Edge ou PE), ces deux routeurs seront reliés à d'autres fournisseurs de service. Les routeurs Annaba et Oran sont aussi des PE, ils seront reliés aux routeurs des clients d'ICOSNET (Customer Edge ou CE).

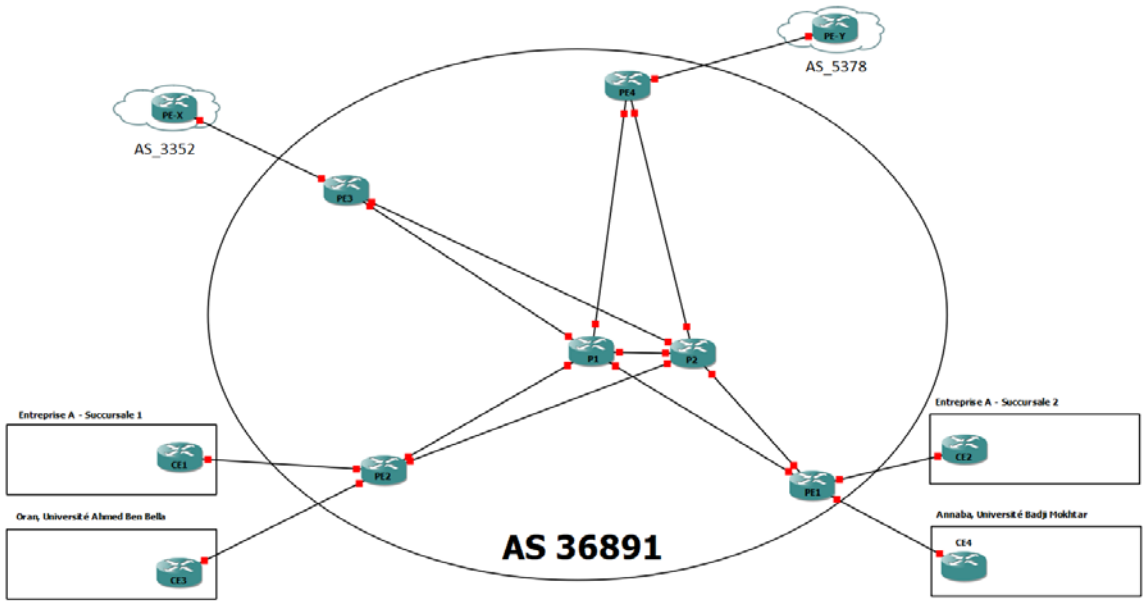

Figure 5-5 Bordure du réseau

L'AFRINIC a attribué à l'entreprise ICOSNET le numéro de système autonome 36891. Les préfixes d'adresse IP publique allouées à ICOSNET sont : 196.41.224.0/19, 196.41.250.0/24, 196.41.252.0/24, 197.140.0.0/14

ICOSNET collabore avec d'autres fournisseurs de service, La liste de ces fournisseurs peut être trouvée sur le site web de l'entreprise :

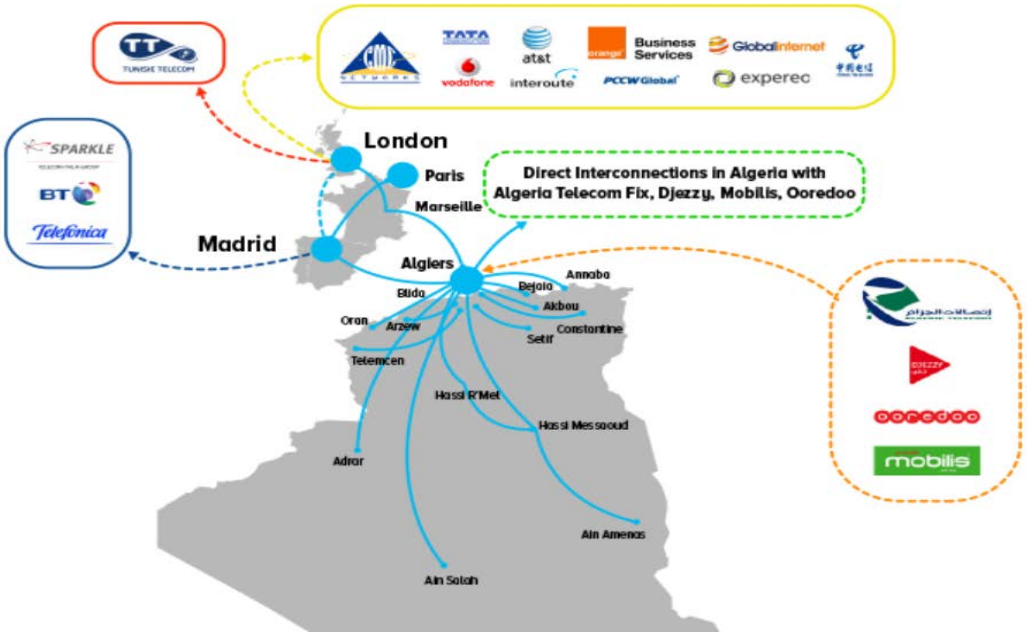

Figure 5-6 Partenaires d'ICOSNET

Pour retirer toute complexité non nécessaire, lors de notre étude nous allons réduire le nombre de partenaire à deux. Les AS 5378 et 36891.

ICOSNET est assez étendu sur le territoire national, pour notre étude nous allons nous limiter à deux villes, et trois clients fictifs.

7. Adressage utilisé dans le réseau

Les interfaces de chaque routeur doivent avoir une adresse IP, et chaque liaison de donnée doit avoir une adresse de réseau. Voici l'adressage prévu pour le réseau :

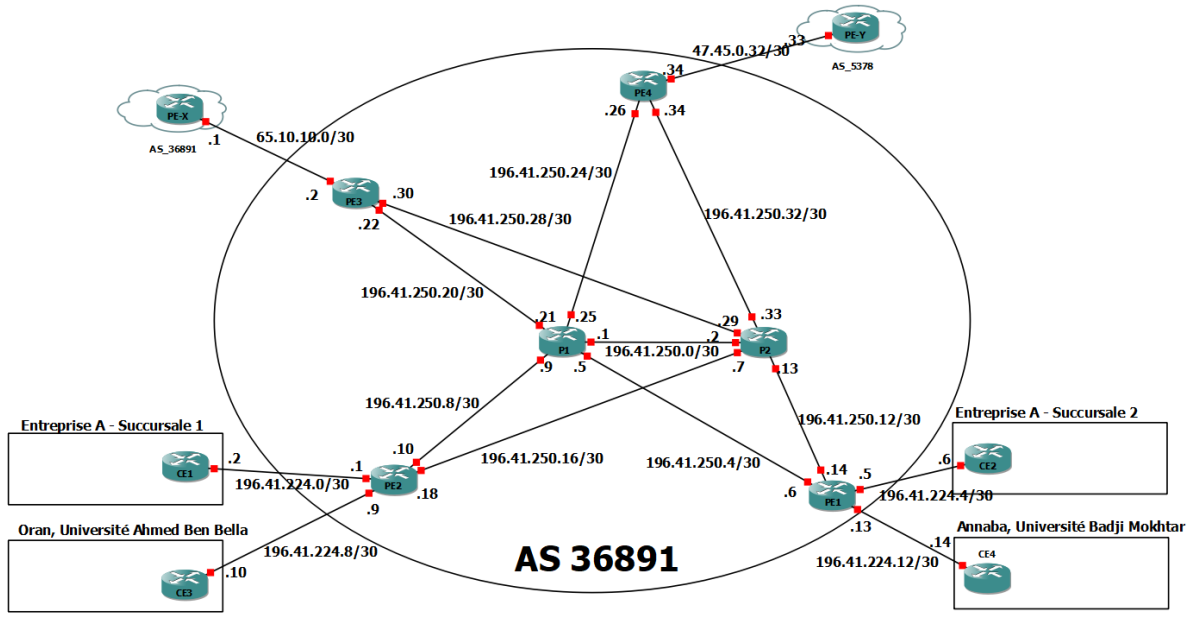

Figure 5-7 Adressage des liaisons de donnée et interfaces

8. Politiques de routage BGP

Le plan de contrôle dirige par défaut le trafic par le chemin le plus optimal. Pour une entreprise il est nécessaire d'appliquer des politiques de routage pour changer le comportement du plan de contrôle pour qu'il prenne des décisions de routage selon les préférences de l'entreprise et non selon le meilleur chemin.

ICOSNET est un Stub AS, ce qui veut dire que tout trafic externe entrant dans l'AS est destiné pour l'AS, et tout trafic sortant de l'AS provient de l'AS. Pour ne pas devenir un AS de transite ICOSNET doit appliquer une politique de filtrage des routes annoncées aux autres AS. Les seuls préfixes d'adresse que les routeurs Madrid (PE3) et Londres (PE4) sont autorisés à annoncer à leurs pairs BGP externes sont les préfixes qui appartiennent à ICOSNET.

Pour le trafic généré dans l'AS qui à une destination externe à l'AS, le trafic doit être envoyer à un partenaire, une fois un paquet sorti de son AS ICOSNET n'a plus aucun contrôle sur l'acheminement de ce paquet. ICOSNET possède plusieurs partenaires, elle doit négocier avec ses partenaires de la façon avec laquelle ils vont traiter les paquets qu'elle leur envoie. Il peut y avoir des préférences au niveau de l'AS par rapport au partenaire à utiliser pour l'acheminement d'un certain type de paquets spécifique.

Dans notre version simplifiée du réseau, nous allons utiliser l'AS 5378 pour tout trafic vers une destination qui est l'AS 5378 ou un AS directement connectés à l'AS 5378. Tous autres trafics vers une d'autres destination externe seront acheminés par l'AS 3352.

9. MPLS

MPLS va permettre au backbone de commuter les paquets en se basant sur le label MPLS aux lieux d'une adresse réseau.

MPLS est simple est flexible, il va permettre au backbone non seulement une commutation plus rapide, mais aussi le support d'un grand nombre de technologies et protocoles à travers une infrastructure commune. En installant MPLS un fournisseur de service va pouvoir offrir à ses clients de nouveaux services à couts très réduits car le besoin de multiples réseaux n'est plus nécessaire, pour offrir ces services, un seul réseau de base suffit.

On implémentera MPLS en 'mode trame', le label MPLS sera apposé entre l'entête de couche réseau et l'entête de couche liaison de donnée.

Pour simplifier notre étude toute complexité non nécessaire sera retiré, le réseau MPLS aura une hiérarchie plate. Le sens des labels aura une portée par plateforme. Le mode de distribution à utiliser est 'Downstream Unsolicited ou DU'. Le mode de rétention des labels sera 'Liberal Label Retention ou LLR'.

10. MultiProtocol-BGP, VRF et MPLS VPN

MultiProtocol BGP ou MP-BGP fourni un à BGP les moyens pour transporter des informations de routage (information d'accessibilité en général) de différentes familles de protocoles de couche réseau. Dans les backbone des fournisseurs de service modernes MP-BGP est principalement utilisé avec MPLS pour faciliter le contrôle et la commutation des VPNs MPLS.

MP-BGP et MPLS permettront de fournir aux clients, un service de couche liaisons de donnée et réseau privé à travers le backbone qui est publique. Ceci permet de réduire les couts de l'infrastructure du fournisseur et ainsi être plus compétitif. La simplicité de MPLS VPN enlève la complexité de conception et maintenance connues sur les VPN traditionnels.

On implémentera des VRF sur les routeurs qui sont reliés au clients, PE1 et PE2. Les VRF serviront à récupérer les tables de routage des clients et les isolés. BGP récupéra ensuite le contenu de ces tables sous forme d'adresses VPNv4.

Pour faire fonctionner MPLS VPN ils est nécessaire de propager les préfixes contenus dans les VRF vers tous les PE connectés à des clients. On utilisera MP-BGP pour propager les préfixes VPNv4, les messages UPDATE auront un attribut de communauté étendue qui permettra aux routeurs ayant reçu le message UPDATE de savoir dans quelle VRF installer le préfixe.

VI. Implémentation

Une fois la conception du réseau terminée. L'implémentation du réseau est le travail des ingénieurs réseau. Pour l'implémentation du réseau vu dans le chapitre précèdent, l'entreprise ICOSNET nous a fourni un environnement d'étude et d'expérimentation conçu pour tester des architecteurs réseau avant de les implémenter dans un environnement de production.

1. L'environnement d'étude

L'environnement d'étude est un serveur 'bare-metal Hypervisor', c'est une solution commerciale vendu par VMWare appelé ESXI. Cet Hyperviseur permet la virtualisation d'un très grand nombre de machine virtuelles au même temps.

Nous utilisons des images de Cisco IOS sur Linux 'IOS Over Linux' pour virtualiser les routeurs. On obtient le même résultat qu'en utilisant Cisco IOS sur du matériel physique.

Pour virtualiser les liens on utilise des switches virtuels entre les machines virtuelles pour former une liaison point à point.

2. Affectation des adresses IP

La première étape est d'affecté les adresse de la figure 5-7 aux interfaces des routeurs, cela permet la connectivité de couche réseau.

```
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                196.41.250.14   YES manual up                    up
Ethernet0/1                196.41.250.6    YES manual up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down down
Ethernet0/3                unassigned      YES NVRAM  administratively down down
Ethernet1/0                196.41.224.5    YES manual up                    up
Ethernet1/1                196.41.224.13   YES manual up                    up
Ethernet1/2                unassigned      YES NVRAM  administratively down down
Ethernet1/3                unassigned      YES NVRAM  administratively down down
Serial2/0                  unassigned      YES NVRAM  administratively down down
Serial2/1                  unassigned      YES NVRAM  administratively down down
Serial2/2                  unassigned      YES NVRAM  administratively down down
Serial2/3                  unassigned      YES NVRAM  administratively down down
Serial3/0                  unassigned      YES NVRAM  administratively down down
Serial3/1                  unassigned      YES NVRAM  administratively down down
Serial3/2                  unassigned      YES NVRAM  administratively down down
Serial3/3                  unassigned      YES NVRAM  administratively down down
Loopback0                  172.31.0.4      YES manual up                    up
```

Ceci est la liste des interfaces affiché par PE1, on voit que les adresses IP sont bien associées aux interfaces, et que les interfaces sont fonctionnelles. L'interface logique 'Loopback0'est aussi présente et possède l'adresse IP '172.31.0.4'. PE1 sera identifié dorénavant par cette adresse.

Tous les autres routeurs du backbone afficheront une liste similaire avec les adresses IP qui leur en sont assignés.

3. Implémentation d'OSPF

Nous activerons OSPF sur toutes les interfaces des routeurs reliés au backbone. Nous avons commencé par P1 avant tous les autres routeurs, voici une capture du trafic réseau entre P1 et P2 avant qu'OSPF ne soit activé sur P2 :

```
> Frame 79: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:01:00 (aa:bb:cc:00:01:00), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
> Internet Protocol Version 4, Src: 196.41.250.1, Dst: 224.0.0.5
v Open Shortest Path First
  v OSPF Header
      Version: 2
      Message Type: Hello Packet (1)
      Packet Length: 44
      Source OSPF Router: 172.31.0.1
      Area ID: 0.0.0.0 (Backbone)
      Checksum: 0x3f82 [correct]
      Auth Type: Null (0)
      Auth Data (none): 0000000000000000
  v OSPF Hello Packet
      Network Mask: 255.255.255.252
      Hello Interval [sec]: 10
    > Options: 0x12 ((L) LLS Data block, (E) External Routing)
      Router Priority: 1
      Router Dead Interval [sec]: 40
      Designated Router: 0.0.0.0
      Backup Designated Router: 0.0.0.0
  > OSPF LLS Data Block
```

Figure 6-1 message OSPF Hello de P1

Ceci est un message OSPF Hello intercepté avec l'outil de capture Wireshark, Dans l'entête de couche réseau on voit que la source du message est l'adresse 196.41.250.1, cette adresse correspond à l'interface de P1 qui est lié à la même liaison de donnée que P2. La destination de ce message est 224.0.0.5 qui est une adresse multicast pour tous routeurs utilisant OSPF.

L'entête suivante est l'entête OSPF, on retrouve les champs vus dans le chapitre OSPF, on note que l'identifiant utilisé par le routeur est 172.31.0.1, ce qui correspond bien à l'identifiant de P1. Et la zone OSPF correspond à la zone backbone 'Area 0'.

```
> Frame 416: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:02:00 (aa:bb:cc:00:02:00), Dst: aa:bb:cc:00:01:00 (aa:bb:cc:00:01:00)
> Internet Protocol Version 4, Src: 196.41.250.2, Dst: 196.41.250.1
v Open Shortest Path First
  v OSPF Header
      Version: 2
      Message Type: Hello Packet (1)
      Packet Length: 48
      Source OSPF Router: 172.31.0.2
      Area ID: 0.0.0.0 (Backbone)
      Checksum: 0x1704 [correct]
      Auth Type: Null (0)
      Auth Data (none): 0000000000000000
  v OSPF Hello Packet
      Network Mask: 255.255.255.252
      Hello Interval [sec]: 10
    > Options: 0x12 ((L) LLS Data block, (E) External Routing)
      Router Priority: 1
      Router Dead Interval [sec]: 40
      Designated Router: 196.41.250.1
      Backup Designated Router: 196.41.250.2
      Active Neighbor: 172.31.0.1
  > OSPF LLS Data Block
```

Figure 3-2 Message OSPF Hello de P2

Après réception du message OSPF Hello de P1, P2 renvoi un message OSPF Hello avec l'identifiant de P1 inclus.

Lors de la formation de la relation de voisinage OSPF le routeur P2 imprime les messages suivants :

```
*Sep 25 21:05:49.990: OSPF EVENT Et0/0: Route adjust
*Sep 25 21:05:49.990: OSPF-1 ADJ   Et0/0: Route adjust notification: UP/UP
*Sep 25 21:05:49.990: OSPF-1 ADJ   Et0/0: Interface going Up
```

Ceci indique que l'interface OSPF Et0/0 est fonctionnelle, et peut désormais commencer à envoyer des messages Hello.

```
*Sep 25 21:05:49.990: OSPF-1 PAK  : Et0/0: OUT: 196.41.250.2->224.0.0.5: ver:2 type:1 len:44
rid:172.31.0.2 area:0.0.0.0 chksum:3F81 auth:0
```

Message OSPF Hello envoyé sur l'interface Et0/0, les paramètres du message sont aussi indiqués.

```
*Sep 25 21:05:49.990: OSPF-1 ADJ   Et0/0: Interface state change to UP, new ospf state WAIT
```

L'état de l'interface OSPF est maintenant WAIT.

```
*Sep 25 21:05:49.991: OSPF-1 PAK  : Et0/0:  IN: 196.41.250.1->196.41.250.2: ver:2 type:1
len:48 rid:172.31.0.1 area:0.0.0.0 chksum:D530 auth:0
```

Un message OSPF Hello de P1 a été reçu.

```
*Sep 25 21:05:49.991: OSPF-1 ADJ   Et0/0: 2 Way Communication to 172.31.0.1, state 2WAY
```

L'état de l'interface OSPF est maintenant 2-Way. La connexion bidirectionnelle a bien été établie.

```
*Sep 25 21:05:49.991: OSPF-1 ADJ   Et0/0: Nbr 172.31.0.1: Prepare dbase exchange
```

Les deux nœuds OSPF vont maintenant se préparer à l'échanger des messages descriptifs de la base de données d'état de lien

```
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: Rcv DBD from 172.31.0.1 seq 0x1E3F opt 0x52 flag 0x7
len 32  mtu 1500 state EXSTART
```

L'état de l'interface OSPF est maintenant ExStart. La séquence de départ est 0x1E3F.

```
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: First DBD and we are not SLAVE
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: NBR Negotiation Done. We are the MASTER
```

Les nœuds ont déterminé que P2 est le maitre et P1 est l'esclave.

```
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: Rcv DBD from 172.31.0.1 seq 0x13BF opt 0x52 flag 0x0
len 32  mtu 1500 state EXCHANGE
```

L'état de l'interface OSPF est maintenant EXCHANGE, les deux nœuds vont s'échanger des messages descriptifs de la base de données d'état de lien.

```
*Sep 25 21:05:49.991: OSPF-1 PAK  : Et0/0: OUT: 196.41.250.2->196.41.250.1: ver:2 type:2
len:32 rid:172.31.0.2 area:0.0.0.0 chksum:E61A auth:0
```

P2 envoie un message OSPF de type 2, c'est un message descriptif de la base de données d'état de lien

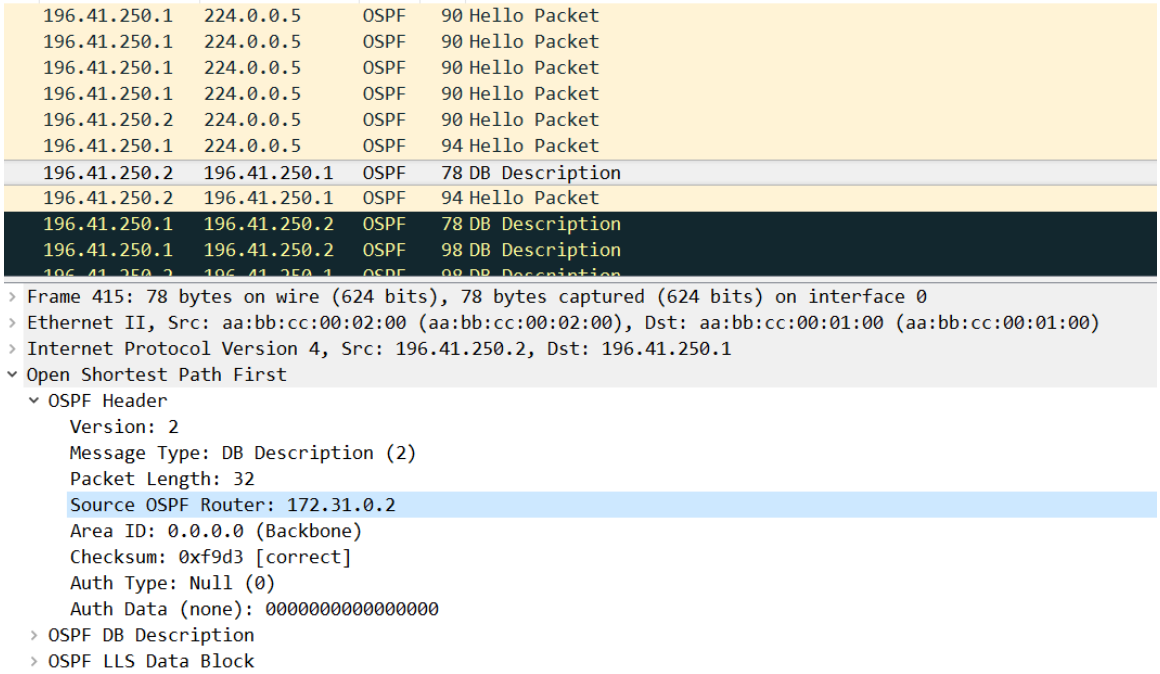

Figure 4-3 capture d'un message OSPF descriptif de la base de données d'état de lien

Au même instant le message descriptif de la base de données d'état de lien a été intercepté par le logiciel de capture de paquets.

```
*Sep 25 21:05:54.660: OSPF-1 PAK  : Et0/0:  IN: 196.41.250.1->196.41.250.2: ver:2 type:2
len:32 rid:172.31.0.1 area:0.0.0.0 chksum:E621 auth:0
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: Exchange Done with 172.31.0.1
```

L'échange des messages descriptifs de la base de données est désormais terminé, l'interface OSPF passe à l'état Loading, Dans cet état les deux nœuds vont se demander puis s'envoyer les LSA qui leurs manques.

```
*Sep 25 21:05:54.660: OSPF-1 PAK  : Et0/0:  IN: 196.41.250.1->196.41.250.2: ver:2 type:3
len:48 rid:172.31.0.1 area:0.0.0.0 chksum:8F19 auth:0
```

P1 demande à P2 une LSA qu'il ne possède pas avec un message de requête d'état de lien.

```
*Sep 25 21:05:54.660: OSPF-1 PAK  : Et0/0: OUT: 196.41.250.2->196.41.250.1: ver:2 type:4
len:156 rid:172.31.0.2 area:0.0.0.0 chksum:3BD1 auth:0
```

P2 répond à la requête de P1 avec un message de mise à jour d'état de lien.

```
*Sep 25 21:05:54.660: OSPF-1 ADJ   Et0/0: Send LS REQ to 172.31.0.1 length 36
*Sep 25 21:05:54.660: OSPF-1 PAK  : Et0/0: OUT: 196.41.250.2->196.41.250.1: ver:2 type:3
len:36 rid:172.31.0.2 area:0.0.0.0 chksum:F974 auth:0
*Sep 25 21:05:54.661: OSPF-1 PAK  : Et0/0:  IN: 196.41.250.1->196.41.250.2: ver:2 type:4
len:112 rid:172.31.0.1 area:0.0.0.0 chksum:E9AA auth:0

*Sep 25 21:05:54.661: OSPF-1 ADJ   Et0/0: Synchronized with 172.31.0.1, state FULL
*Sep 25 21:05:54.661: %OSPF-5-ADJCHG: Process 1, Nbr 172.31.0.1 on Ethernet0/0 from LOADING to
FULL, Loading Done
```

Une fois les bases de données synchronisées l'état de l'interface passe à full.

Une fois OSPF activé sur tout les routeurs et les relations de voisinage fonctionnelles, on a affiché la base de données d'état de lien sur le routeur PE4 (identifié par 172.31.0.6) :

```
            OSPF Router with ID (172.31.0.6) (Process ID 1)

                Router Link States (Area 0)
```

```
Link ID         ADV Router      Age         Seq#       Checksum Link count
172.31.0.1      172.31.0.1      20          0x80000017 0x00CDD5 6
172.31.0.2      172.31.0.2      1129        0x8000001A 0x0080DC 6
172.31.0.3      172.31.0.3      1225        0x80000004 0x004E94 5
172.31.0.4      172.31.0.4      21          0x80000005 0x00B233 5
172.31.0.5      172.31.0.5      1164        0x80000003 0x003B46 3
172.31.0.6      172.31.0.6      1117        0x80000004 0x001558 3

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
196.41.250.1    172.31.0.1      1294        0x80000010 0x005884
196.41.250.6    172.31.0.4      21          0x80000001 0x00429F
196.41.250.9    172.31.0.1      1273        0x80000001 0x0034AE
196.41.250.13   172.31.0.2      1364        0x80000001 0x001EBD
196.41.250.17   172.31.0.2      1273        0x80000001 0x00E7F0
196.41.250.21   172.31.0.1      1185        0x80000001 0x00D7FC
196.41.250.25   172.31.0.1      1126        0x80000001 0x00BD12
196.41.250.29   172.31.0.2      1183        0x80000001 0x008B3F
196.41.250.33   172.31.0.2      1129        0x80000001 0x007154
```

On constate une liste des nœuds OSPF du backbone intitulée ‘Router Link States (Area 0)’, pour chaque nœud il est précisé le nombre de lien qu’il possède ‘Link’. Voici les détails des liens pour le PE2 identifié par (172.31.0.4) :

```
            OSPF Router with ID (172.31.0.6) (Process ID 1)

                Router Link States (Area 0)

  LS age: 792
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 172.31.0.4
  Advertising Router: 172.31.0.4
  LS Seq Number: 80000005
  Checksum: 0xB233
  Length: 84
  Number of Links: 5

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 172.31.0.4
     (Link Data) Network Mask: 255.255.255.255
      Number of MTID metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 196.41.224.12
     (Link Data) Network Mask: 255.255.255.252
      Number of MTID metrics: 0
       TOS 0 Metrics: 10

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 196.41.224.4
     (Link Data) Network Mask: 255.255.255.252
      Number of MTID metrics: 0
       TOS 0 Metrics: 10

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 196.41.250.6
     (Link Data) Router Interface address: 196.41.250.6
      Number of MTID metrics: 0
       TOS 0 Metrics: 10
```

Pour chaque lien connecté au nœud il est précisé si un réseau de transite ‘Transit Network’ ou réseau de bout ‘Stub Network’. Pour les Stub une adresse réseau et un masque sont aussi indiqué, dans cet extrait on retrouve les adresses des sous-réseaux de CE2 et CE4.

4. Implémentation de BGP

Pour échanger des informations de routage avec les systèmes autonomes 5378 et 3352 il faut utiliser BGP entre les routeurs qui relient ICOSNET à ces AS. On commence par établir une session BGP entre PE4 et PE-Y

La capture suivante est une suite de messages affichés par TE4 lors de l'établissement de la session BGP :

```
*Sep 26 01:29:19.280: BGP: 47.45.0.33 active went from Idle to Active
```

L'état de BGP est passé de l'état IDLE à Active

```
*Sep 26 01:29:19.280: BGP: 47.45.0.33 active went from Active to OpenSent
```

Le routeur passe de l'état Active à l'état OpenSent, ceci indique que la session TCP a été établie avec succès.

```
*Sep 26 01:29:19.280: BGP: 47.45.0.33 active sending OPEN, version 4, my as: 36891, holdtime
180 seconds, ID AC1F0006
```

Envoi d'un message OPEN de la part de PE4.

```
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active rcv message type 1, length (excl. header) 38
*Sep 26 01:29:19.286: BGP: ses global 47.45.0.33 (0xF2616468:0) act Receive OPEN
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active rcv OPEN, version 4, holdtime 180 seconds
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active OPEN has 4-byte ASN CAP for: 5378
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active rcvd OPEN w/ remote AS 5378, 4-byte remote AS
5378
```

TE4 vient de recevoir un message OPEN de la part de PE-Y situé dans le système autonomes 5378.

```
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active went from OpenSent to OpenConfirm
```

L'état de BGP passe à OpenConfirm, le routeur va attendre de recevoir un message KEEPALIVE pour passer à l'état Established.

```
*Sep 26 01:29:19.286: BGP: 47.45.0.33 active went from OpenConfirm to Established
```

L'état de BGP est Established, la session BGP est fonctionnelle.

On inspectons la Loc-RIB de PE4 on trouve le résultat suivant :

```
BGP table version is 4, local router ID is 172.31.0.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  47.47.0.0/16     47.45.0.33               0             0 5378 i
 *>  196.41.224.0/19  0.0.0.0                  0         32768 i
 *>  196.41.250.0     0.0.0.0                  0         32768 i
```

Le réseau 47.47.0.0/16 n'appartient pas à ICOSNET, d'ailleurs la colonne 'Path' indique que pour s'y rendre il faut passer le système autonomes 5378. Et le prochain saut pour cette route c'est l'adresse IP de l'interface de PE-Y. On peut donc conclure que les deux pairs ont commencé à s'échanger des préfixes d'adresse.

Un logiciel de capture a été mis en place entre PE4 et PE-Y, le message BGP UPDATE dans lequel le préfixe 47.47.0.0/16 a été annoncé a été capturé

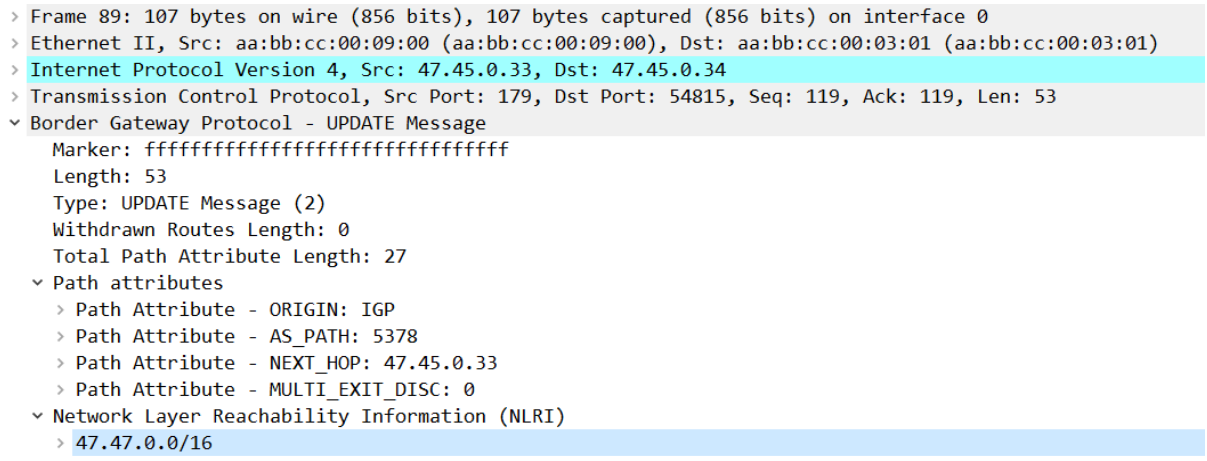

Figure 6-4 Capture d'un message BGP UPDATE

On retrouve le préfixe 47.47.0.0/16 dans les NLRI du message UPDATE, les attributs chemins y sont aussi dans le champ 'Path Attributes'.

Une session BGP a aussi été établie entre PE3 et PE-X.

Pour propager les routes externes dans le système autonomes il faut établir des sessions IBGP, tous les routeurs auront une session IBGP avec P1 et P2 qui seront des Route Reflector.

```
 BGP router identifier 172.31.0.1, local AS number 36891
 BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.31.0.2      4        36891       0       0        1    0    0 never     Idle
172.31.0.3      4        36891       0       0        1    0    0 never     Active
172.31.0.4      4        36891       0       0        1    0    0 never     Idle
172.31.0.5      4        36891       2       2        1    0    0 00:00:05        0
172.31.0.6      4        36891       2       2        1    0    0 00:00:01        0
```

On affiche la liste des pairs BGP sur P1, on peut se rendre compte que la session a été établie avec PE3 et PE4, pour P2, PE1 et P2 la session n'est pas encore établie, l'état de BGP est soit Idle ou Active (les sessions sont en cours d'établissement).

Une fois les sessions BGP établies, les routes externes vont être propagées à travers le système autonome, en inspectant la liste des pairs BGP de PE1, on trouve seulement P1 et P2

```
BGP router identifier 172.31.0.4, local AS number 36891
BGP table version is 9, main routing table version 9

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.31.0.1      4        36891      13      10        9    0    0 00:06:50        4
172.31.0.2      4        36891      13      10        9    0    0 00:06:49        4
```

Mais si on regarde la Loc-RIB de PE1 on retrouve les routes annoncées par PE-X et PE-Y

```
BGP table version is 9, local router ID is 172.31.0.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 47.47.0.0/16     172.31.0.6               0    100      0 5378 i
 * i                  172.31.0.6               0    100      0 5378 i
 *>i 68.0.0.0         172.31.0.5               0    100      0 3352 i
 * i                  172.31.0.5               0    100      0 3352 i
```

Ceci prouve que les Routes Reflector ont propagé les routes apprises depuis PE3 et PE4 via IBGP à PE1 et PE2 via IBGP.

5. Politiques de routage

Une fois la session EBGP entre PE3 et PE-X établie, une capture de paquet a intercepté le message UPDATE suivant :

```
> Frame 65: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:06:01 (aa:bb:cc:00:06:01), Dst: aa:bb:cc:00:0a:00 (aa:bb:cc:00:0a:00)
> Internet Protocol Version 4, Src: 65.10.10.2, Dst: 65.10.10.1
> Transmission Control Protocol, Src Port: 41549, Dst Port: 179, Seq: 96, Ack: 171, Len: 131
v Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 50
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 24
  > Path attributes
  v Network Layer Reachability Information (NLRI)
    > 47.47.0.0/16
> Border Gateway Protocol - UPDATE Message
> Border Gateway Protocol - UPDATE Message
```

Figure 6-5 Message BGP update entre PE3 et PE-X

Un message UPDATE ayant comme source PE3 et comme destination PE-X contient dans ses NLRI un préfixe du système autonome 5378.

Si on affiche sur PE3 les routes qu'il annoncées à PE-X (65.10.10.1), on obtient le résultat suivant :

```
PE3#show ip bgp neighbor 65.10.10.1 advertised-routes
BGP table version is 9, local router ID is 172.31.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 47.47.0.0/16     172.31.0.6               0    100      0 5378 i
 *>  196.41.224.0/19  0.0.0.0                  0         32768 i
 *>  196.41.250.0     0.0.0.0                  0         32768 i
```

On voit que le préfixe 47.47.0.0/16 qui appartient à 5378 a été annoncé à PE-X.

En annonçant le préfixe de l'AS 5378 à l'AS 3352 notre AS risque de devenir un transite entre les deux AS. ICOSNET n'est pas un AS de transite, pour éviter de le devenir il faudrait appliquer une politique de routage sur les routeurs utilisant EBGP. Cette politique aura pour but de bloquer les préfixes n'appartenant pas à ICOSNET d'être annoncés à un pair BGP externe.

On utilisera une 'Prefix List' pour filtrer les routes à annoncer via EBGP. La 'Prefix List' va contenir les préfixes d'adresse IP alloués à ICOSNET par l'AFRINIC.

```
ip prefix-list ICOSNETOUT seq 5 permit 196.41.224.0/19
ip prefix-list ICOSNETOUT seq 10 permit 196.41.250.0/24
ip prefix-list ICOSNETOUT seq 15 permit 196.41.252.0/24
ip prefix-list ICOSNETOUT seq 20 permit 197.140.0.0/14
```

On appliquera cette ‘Prefix List’ sur PE3 et PE4 pour les pairs PE-X et PE-Y sur la direction sortante.

```
PE3#show run | S bgp
router bgp 36891
 neighbor 65.10.10.1 remote-as 3352
 neighbor 65.10.10.1 prefix-list ICOSNETOUT out

PE4#show run | S bgp
router bgp 36891
 neighbor 47.45.0.33 remote-as 5378
 neighbor 47.45.0.33 prefix-list ICOSNETOUT out
```

Une fois la ‘Prefix List’ appliquée aux deux routeurs pour la direction sortante vers leurs pairs externes, on inspectera les routes annoncées par ces routeurs à leur pair externe

```
PE4#show ip bgp neighbor 47.45.0.33 advertised-routes
BGP table version is 11, local router ID is 172.31.0.6

     Network          Next Hop            Metric LocPrf Weight Path
 *>  196.41.224.0/19  0.0.0.0                  0         32768 i
 *>  196.41.250.0     0.0.0.0                  0         32768 i

Total number of prefixes 2

PE3#show ip bgp neighbor 65.10.10.1 advertised-routes
BGP table version is 9, local router ID is 172.31.0.5

     Network          Next Hop            Metric LocPrf Weight Path
 *>  196.41.224.0/19  0.0.0.0                  0         32768 i
 *>  196.41.250.0     0.0.0.0                  0         32768 i

Total number of prefixes 2
```

Les préfixes externes ne sont plus annoncés, la ‘Prefix List’ filtre les préfixes, et n’autorise que les préfixe appartenant à ICOSNET.

Il nous est demandé d’utiliser le système autonomes 5378 pour le trafic qui a une destination à un saut d’AS ou moins de l’AS 5378. C’est-à-dire le trafic sortant par PE4 n’a le droit le transiter que par l’AS 5378. Tout autre trafic vers l’extérieur doit sortir par PE3.

Nous avons ajouté des systèmes autonomes supplémentaires, cela a pour but d’agrandir la table de routage BGP. Ces systèmes autonomes ne seront pas reliés à ICOSNET.

Après l’ajout des systèmes autonomes voici le contenu de la Loc-RIB de PE3 et PE4

```
PE3#show ip bgp
BGP table version is 27, local router ID is 172.31.0.5

     Network          Next Hop            Metric LocPrf Weight Path
 *   1.0.0.0          65.10.10.1                             0 3352 791 790 789 i
 * i                  172.31.0.6               0    100      0 5378 789 i
 *>i                  172.31.0.6               0    100      0 5378 789 i
 *   2.0.0.0          65.10.10.1                             0 3352 791 790 789 i
 * i                  172.31.0.6               0    100      0 5378 789 i
 *>i                  172.31.0.6               0    100      0 5378 789 i
 *   3.0.0.0          65.10.10.1                             0 3352 791 790 789 i
 * i                  172.31.0.6               0    100      0 5378 789 i
 *>i                  172.31.0.6               0    100      0 5378 789 i
 *>  30.10.0.0/24     65.10.10.1                             0 3352 791 i
 *>  30.20.0.0/24     65.10.10.1                             0 3352 791 i
     Network          Next Hop            Metric LocPrf Weight Path
 *>  30.30.0.0/24     65.10.10.1                             0 3352 791 i
 *   47.47.0.0/16     65.10.10.1                             0 3352 791 790 789 5378 i
 *>i                  172.31.0.6               0    100      0 5378 i
 * i                  172.31.0.6               0    100      0 5378 i
 *>  66.6.6.0/24      65.10.10.1                             0 3352 65666 i
 *>  66.6.7.0/24      65.10.10.1                             0 3352 65666 i
 *>  66.6.8.0/24      65.10.10.1                             0 3352 65666 i
 *>  66.6.9.0/24      65.10.10.1                             0 3352 65666 i
 *>  68.0.0.0         65.10.10.1               0             0 3352 i
 *   89.0.0.0/16      65.10.10.1                             0 3352 791 790 789 5378 65655 i
 * i                  172.31.0.6               0    100      0 5378 65655 i
```

```
 *>i                  172.31.0.6               0    100      0 5378 65655 i
 *    89.1.0.0/16     65.10.10.1                             0 3352 791 790 789 5378 65655 i
 * i                  172.31.0.6               0    100      0 5378 65655 i
 *>i                  172.31.0.6               0    100      0 5378 65655 i
 *    89.2.0.0/16     65.10.10.1                             0 3352 791 790 789 5378 65655 i
 * i                  172.31.0.6               0    100      0 5378 65655 i
 *>i                  172.31.0.6               0    100      0 5378 65655 i
     Network          Next Hop            Metric LocPrf Weight Path
 *>  196.41.224.0/19  0.0.0.0                   0         32768 i
 *>  196.41.250.0     0.0.0.0

PE4#show ip bgp
BGP table version is 31, local router ID is 172.31.0.6

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          47.45.0.33                             0 5378 789 i
 *>  2.0.0.0          47.45.0.33                             0 5378 789 i
 *>  3.0.0.0          47.45.0.33                             0 5378 789 i
 * i 30.10.0.0/24     172.31.0.5               0    100      0 3352 791 i
 *>i                  172.31.0.5               0    100      0 3352 791 i
 *                    47.45.0.33                             0 5378 789 790 791 i
 * i 30.20.0.0/24     172.31.0.5               0    100      0 3352 791 i
 *>i                  172.31.0.5               0    100      0 3352 791 i
 *                    47.45.0.33                             0 5378 789 790 791 i
 * i 30.30.0.0/24     172.31.0.5               0    100      0 3352 791 i
 *>i                  172.31.0.5               0    100      0 3352 791 i
 *                    47.45.0.33                             0 5378 789 790 791 i
     Network          Next Hop            Metric LocPrf Weight Path
 *>  47.47.0.0/16     47.45.0.33               0             0 5378 i
 *    66.6.6.0/24     47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *>i                  172.31.0.5               0    100      0 3352 65666 i
 * i                  172.31.0.5               0    100      0 3352 65666 i
 *    66.6.7.0/24     47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *>i                  172.31.0.5               0    100      0 3352 65666 i
 * i                  172.31.0.5               0    100      0 3352 65666 i
 *    66.6.8.0/24     47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *>i                  172.31.0.5               0    100      0 3352 65666 i
 * i                  172.31.0.5               0    100      0 3352 65666 i
 *    66.6.9.0/24     47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *>i                  172.31.0.5               0    100      0 3352 65666 i
 * i                  172.31.0.5               0    100      0 3352 65666 i
 *    68.0.0.0        47.45.0.33                             0 5378 789 790 791 3352 i
 *>i                  172.31.0.5               0    100      0 3352 i
 * i                  172.31.0.5               0    100      0 3352 i
 *>  89.0.0.0/16     47.45.0.33                             0 5378 65655 i
     Network          Next Hop            Metric LocPrf Weight Path
 *>  89.1.0.0/16     47.45.0.33                             0 5378 65655 i
 *>  89.2.0.0/16     47.45.0.33                             0 5378 65655 i
 * i 196.41.224.0/19  172.31.0.5               0    100      0 i
 * i                  172.31.0.5               0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 * i 196.41.250.0     172.31.0.5               0    100      0 i
 * i                  172.31.0.5               0    100      0 i
 *>                   0.0.0.0                  0         32768 i
```

Pour pouvoir limiter les transactions entre PE4 et PE-Y au trafic entre ICOSNET et une destination située dans l'AS 5378 ou un AS directement connecté à l'AS 5378, on doit filtrer les préfixes reçus par PE4 de la part de PE-Y.

Voici la liste des préfixes que PE4 a reçu de PE-Y

```
PE4#show ip bgp neighbor 47.45.0.33 routes
BGP table version is 31, local router ID is 172.31.0.6

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          47.45.0.33                             0 5378 789 i
 *>  2.0.0.0          47.45.0.33                             0 5378 789 i
 *>  3.0.0.0          47.45.0.33                             0 5378 789 i
 *   30.10.0.0/24     47.45.0.33                             0 5378 789 790 791 i
 *   30.20.0.0/24     47.45.0.33                             0 5378 789 790 791 i
 *   30.30.0.0/24     47.45.0.33                             0 5378 789 790 791 i
 *>  47.47.0.0/16     47.45.0.33               0             0 5378 i
 *   66.6.6.0/24      47.45.0.33                             0 5378 789 790 791 3352 65666 i
```

```
 *   66.6.7.0/24      47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *   66.6.8.0/24      47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *   66.6.9.0/24      47.45.0.33                             0 5378 789 790 791 3352 65666 i
 *   68.0.0.0         47.45.0.33                             0 5378 789 790 791 3352 i
 *>  89.0.0.0/16      47.45.0.33                             0 5378 65655 i
 *>  89.1.0.0/16      47.45.0.33                             0 5378 65655 i
 *>  89.2.0.0/16      47.45.0.33                             0 5378 65655 i
 Total number of prefixes 15
```

On utilisera l'attribut AS_PATH pour filtrer les préfixes, on autorisera que les préfixe ayant un AS_PATH commençant par 5378 suivis d'un seul AS ou aucun AS. La formulation de l'expression régulière qui permettra ce filtrage est : ^5378_[0-9]*$. On appliquera cette expression régulière pour filtrer les NLRI envoyées par PE-Y

```
PE4#show run | S as-path
ip as-path access-list 1 permit ^5378_[0-9]*$
```

```
PE4#show run | S bgp
router bgp 36891
 network 196.41.224.0 mask 255.255.224.0
 network 196.41.250.0
 neighbor 47.45.0.33 remote-as 5378
 neighbor 47.45.0.33 prefix-list ICOSNETOUT out
 neighbor 47.45.0.33 filter-list 1 in
```

Un filtre AS_PATH a été crée avec l'expression régulière puis appliqué à PE-Y dans la direction entrante.

Après l'application du filtre et réinitialisation de la session BGP voici les préfixes reçus par PE4 et qui ont été acceptés

```
PE4#show ip bgp neighbor 47.45.0.33 routes
BGP table version is 104, local router ID is 172.31.0.6

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          47.45.0.33                           0 5378 789 i
 *>  2.0.0.0          47.45.0.33                           0 5378 789 i
 *>  3.0.0.0          47.45.0.33                           0 5378 789 i
 *>  47.47.0.0/16     47.45.0.33               0           0 5378 i
 *>  89.0.0.0/16      47.45.0.33                           0 5378 65655 i
 *>  89.1.0.0/16      47.45.0.33                           0 5378 65655 i
 *>  89.2.0.0/16      47.45.0.33                           0 5378 65655 i

Total number of prefixes 7
```

Les seuls préfixes présents sont les préfixes appartenant à l'AS 5378 ou un AS qui lui est directement connecté. Le nombre de préfixes a été réduit de 15 à 7.

6. Implémentation de MPLS

L'activation d'MPLS sur le Cisco IOS est très simple. Il suffit d'activer LDP sur les interfaces du routeur pour qu'il commence à échanger les associations de labels

Après activation d'LDP sur P2, on constate immédiatement sur le logiciel de capture de paquet LDP hello, le message à comme source l'adresse IP de l'interface reliant P2 à PE4, et la destination est 224.0.0.2 qui l'adresse de multidiffusion pour tous les routeurs sur la liaison de donnée. On retrouve aussi l'identifiant de P2 (172.31.0.2) dans le champ LSR ID du message. Le port UDP utilisé est 646.

```
> Frame 40: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:02:10 (aa:bb:cc:00:02:10), Dst: IPv4mcast_02 (01:00:5e:00:00:02)
> Internet Protocol Version 4, Src: 196.41.250.33, Dst: 224.0.0.2
> User Datagram Protocol, Src Port: 646, Dst Port: 646
v Label Distribution Protocol
    Version: 1
    PDU Length: 30
    LSR ID: 172.31.0.2
    Label Space ID: 0
  v Hello Message
      0... .... = U bit: Unknown bit not set
      Message Type: Hello Message (0x100)
      Message Length: 20
      Message ID: 0x00000000
    > Common Hello Parameters TLV
    > IPv4 Transport Address TLV
```

Figure 6-6 Message LDP Hello

Après un échange de message Hello entre P2 et PE4, les deux routeurs commencent à initialiser une session LDP avec des messages d'initialisation.

```
> Frame 44: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:03:10 (aa:bb:cc:00:03:10), Dst: aa:bb:cc:00:02:10 (aa:bb:cc:00:02:10)
> Internet Protocol Version 4, Src: 172.31.0.6, Dst: 172.31.0.2
> Transmission Control Protocol, Src Port: 34926, Dst Port: 646, Seq: 1, Ack: 1, Len: 56
v Label Distribution Protocol
    Version: 1
    PDU Length: 52
    LSR ID: 172.31.0.6
    Label Space ID: 0
  v Initialization Message
      0... .... = U bit: Unknown bit not set
      Message Type: Initialization Message (0x200)
      Message Length: 42
      Message ID: 0x00000002
    v Common Session Parameters TLV
        00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
        TLV Type: Common Session Parameters TLV (0x500)
        TLV Length: 14
      v Parameters
          Session Protocol Version: 1
          Session KeepAlive Time: 180
          0... .... = Session Label Advertisement Discipline: Downstream Unsolicited proposed
          .0.. .... = Session Loop Detection: Loop Detection Disabled
          Session Path Vector Limit: 0
          Session Max PDU Length: 0
          Session Receiver LSR Identifier: 172.31.0.2
          Session Receiver Label Space Identifier: 0
    > Unknown TLV type (0x0506)
```

Figure 6-7 message LDP d'initialisation

Le message d'initialisation est envoyé par PE4 à P2. Le Port TCP source utilisé est 34926 et le port de destination est 646, cela indique que PE4 est le maitre et P2 l'esclave.

Le mode de distribution de label proposé par PE4 est le mode Downstream Unsolicited, le 'label space' est 0, ce qui indique un sensé de label qui a comme portée la plateforme entière.

```
> Internet Protocol Version 4, Src: 172.31.0.6, Dst: 172.31.0.2
> Transmission Control Protocol, Src Port: 34926, Dst Port: 646, Seq: 593, Ack: 601, Len: 136
> [2 Reassembled TCP Segments (654 bytes): #47(518), #52(136)]
v Label Distribution Protocol
    Version: 1
    PDU Length: 650
    LSR ID: 172.31.0.6
    Label Space ID: 0
  v Address Message
      0... .... = U bit: Unknown bit not set
      Message Type: Address Message (0x300)
      Message Length: 26
      Message ID: 0x00000004
    v Address List TLV
        00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
        TLV Type: Address List TLV (0x101)
        TLV Length: 18
        Address Family: IPv4 (1)
      v Addresses
          Address 1: 196.41.250.26
          Address 2: 196.41.250.34
          Address 3: 47.45.0.34
          Address 4: 172.31.0.6
```

Figure 6-8 LDP Address Message

Toujours dans le processus d'initialisation, on retrouve un LDP Address Message, il a été envoyé par PE4 à P2, ce message contient la liste des adresse réseau des interfaces de PE4. Ces adresses vont permettre à P2 de déterminer si PE4 est le prochain saut pour un label.

```
v Label Mapping Message
    0... .... = U bit: Unknown bit not set
    Message Type: Label Mapping Message (0x400)
    Message Length: 24
    Message ID: 0x00000005
  v Forwarding Equivalence Classes TLV
      00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
      TLV Type: Forwarding Equivalence Classes TLV (0x100)
      TLV Length: 8
    v FEC Elements
      v FEC Element 1
          FEC Element Type: Prefix FEC (2)
          FEC Element Address Type: IPv4 (1)
          FEC Element Length: 30
          Prefix: 47.45.0.32
  v Generic Label TLV
      00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
      TLV Type: Generic Label TLV (0x200)
      TLV Length: 4
      .... .... .... 0000 0000 0000 0000 0011 = Generic Label: 0x00003
v Label Mapping Message
    0... .... = U bit: Unknown bit not set
    Message Type: Label Mapping Message (0x400)
    Message Length: 24
    Message ID: 0x00000006
  v Forwarding Equivalence Classes TLV
      00.. .... = TLV Unknown bits: Known TLV, do not Forward (0x0)
      TLV Type: Forwarding Equivalence Classes TLV (0x100)
      TLV Length: 8
```

Figure 6-9 Message LDP Label Mapping

Dans ce message LDP Label Mapping, PE4 envoie à P2 ces association de label, on retrouve la FEC pour le préfixe 47.45.0.32 associé au label 3, c'est l'Implicit NULL qui permet le Penultimate Hop Popping.

```
P2#show mpls ip binding
  47.45.0.32/30
        out label:    imp-null  lsr: 172.31.0.6:0
  172.31.0.1/32
        in label:     16
        out label:    16        lsr: 172.31.0.6:0
```

En inspectant la LIB de P2 on retrouve le préfixe 47.45.0.32 associé à un label de sortie d'Implicit NULL, le prochain saut pour se préfixe est PE4 avec le 'label space' 0. C'est cette association a été apprise avec le message LDP Label mapping vu dans la figure 6-9.

Après activation d'LDP sur tous les routeurs, on inspecte la table de voisinage LDP sur P2

```
P2#show mpls ldp neighbor
    Peer LDP Ident: 172.31.0.6:0; Local LDP Ident 172.31.0.2:0
        TCP connection: 172.31.0.6.34926 - 172.31.0.2.646
        State: Oper; Msgs sent/rcvd: 80/83; Downstream
        Up time: 00:51:11
        LDP discovery sources:
          Ethernet0/1, Src IP addr: 196.41.250.34
        Addresses bound to peer LDP Ident:
          196.41.250.26   196.41.250.34   47.45.0.34      172.31.0.6
    Peer LDP Ident: 172.31.0.1:0; Local LDP Ident 172.31.0.2:0
        TCP connection: 172.31.0.1.646 - 172.31.0.2.58351
        State: Oper; Msgs sent/rcvd: 26/26; Downstream
        Up time: 00:04:10
        LDP discovery sources:
          Ethernet0/0, Src IP addr: 196.41.250.1
        Addresses bound to peer LDP Ident:
          196.41.250.1    196.41.250.5    196.41.250.9    196.41.250.21
          196.41.250.25   172.31.0.1
    Peer LDP Ident: 172.31.0.5:0; Local LDP Ident 172.31.0.2:0
        TCP connection: 172.31.0.5.26351 - 172.31.0.2.646
        State: Oper; Msgs sent/rcvd: 24/28; Downstream
        Up time: 00:02:34
        LDP discovery sources:
          Ethernet1/2, Src IP addr: 196.41.250.30
        Addresses bound to peer LDP Ident:
          196.41.250.22   196.41.250.30   65.10.10.2      172.31.0.5
    Peer LDP Ident: 172.31.0.4:0; Local LDP Ident 172.31.0.2:0
        TCP connection: 172.31.0.4.36242 - 172.31.0.2.646
        State: Oper; Msgs sent/rcvd: 24/24; Downstream
        Up time: 00:02:08
        LDP discovery sources:
          Ethernet0/3, Src IP addr: 196.41.250.14
        Addresses bound to peer LDP Ident:
          196.41.250.14   196.41.250.6    196.41.224.5    196.41.224.13
          172.31.0.4
    Peer LDP Ident: 172.31.0.3:0; Local LDP Ident 172.31.0.2:0
        TCP connection: 172.31.0.3.44428 - 172.31.0.2.646
        State: Oper; Msgs sent/rcvd: 24/23; Downstream
        Up time: 00:01:42
        LDP discovery sources:
          Ethernet1/1, Src IP addr: 196.41.250.18
        Addresses bound to peer LDP Ident:
          196.41.250.10   196.41.250.18   196.41.224.1    196.41.224.9
          172.31.0.3
```

On retrouve les cinq autres routeurs du backbone, P2 a formé une session LDP avec tous les autres routeurs et a fini d'échanger les associations FEC à label.

Les tables LIB contiennent une entrée pour chaque préfixe, au préfixe est associé un label d'entrée, aucun ou plusieurs labels de sortie, pour chaque label de sortie on à un prochain saut pour ce label, voici une fraction de la table LIB du routeur PE1

```
PE1#show mpls ip binding
  172.31.0.1/32
        in label:     16
        out label:    16        lsr: 172.31.0.2:0
        out label:    imp-null  lsr: 172.31.0.1:0     inuse
  172.31.0.2/32
        in label:     17
        out label:    imp-null  lsr: 172.31.0.2:0     inuse
        out label:    16        lsr: 172.31.0.1:0
```

```
  172.31.0.3/32
        in label:     18
        out label:    17        lsr: 172.31.0.2:0     inuse
        out label:    17        lsr: 172.31.0.1:0     inuse
  172.31.0.4/32
        in label:     imp-null
        out label:    18        lsr: 172.31.0.2:0
        out label:    18        lsr: 172.31.0.1:0
```

Le mot clé 'inuse' indique que ce label de sortie est utilisé pour commuter le paquet, le label d'entrée est remplacé par le label de sortie avant d'être envoyé au prochain saut. Quand le label d'entrée est l'Implicit Null ça veut dire que le paquet MPLS est destiné au routeur lui-même.

La commutation à travers le backbone se fait désormais avec MPLS. Les paquets interceptés sur le réseau sont désormais étiqueté avec MPLS.

```
> Frame 33: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
> Ethernet II, Src: aa:bb:cc:00:04:00 (aa:bb:cc:00:04:00), Dst: aa:bb:cc:00:02:30 (aa:bb:cc:00:02:30)
v MultiProtocol Label Switching Header, Label: 23, Exp: 0, S: 1, TTL: 255
    0000 0000 0000 0001 0111 .... .... .... = MPLS Label: 23
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
> Internet Protocol Version 4, Src: 196.41.250.14, Dst: 196.41.224.9
> Internet Control Message Protocol
```

Figure 5-10 Paquet étiqueté avec MPLS

7. Implémentation des VRF

Certains clients dispersés géographiquement, peuvent demander au fournisseur de service une liaison point à multipoint privée entre leurs différents sites. Le fournisseur de service doit alors participer à un échange d'information de routage active avec son client. Pour garder les informations d'accessibilité du client privée, le fournisseur de service doit utiliser des VRF.

Un VRF (Virtual Routing/Forwarding) est une instance d'une table de routage couplée à un VPN. Les PE liés aux clients vont avoir une instance VRF pour chaque VPN client qui leur est attaché. L'interface reliant un PE à un CE peut appartenir qu'à une seule VRF. De cette façon chaque paquet reçu sur l'interface peut être identifié sans aucune ambiguïté comme appartenant à cette VRF.

L'entreprise A, est un client fictif d'ICOSNET. L'entreprise A possède plusieurs sites sur le territoire national, elle veut les relier avec une liaison point à multipoint entre tous ses sites. Pour enlever toute complexité non nécessaire nous limiterons le nombre de sites de l'entreprise à deux.

Pour créer une VRF sur Cisco IOS, il suffit juste de lui donner un nom, on crée une VRF sur PE1 :

```
PE1#show run | S vrf
ip vrf EntrepriseA
 ip vrf forwarding EntrepriseA
```

Une fois la VRF créée, on associe cette VRF à l'interface reliant PE1 à CE2.

```
PE1#show run | S interface Ethernet1/0
interface Ethernet1/0
 description Link to CE2
 ip vrf forwarding EntrepriseA
 ip address 196.41.224.5 255.255.255.252
```

La VRF de l'entreprise A est désormais fonctionnelle sur PE1

```
PE1#show ip route vrf EntrepriseA

Routing Table: EntrepriseA
Gateway of last resort is not set
```

```
      196.41.224.0/24 is variably subnetted, 2 subnets, 2 masks
C        196.41.224.4/30 is directly connected, Ethernet1/0
L        196.41.224.5/32 is directly connected, Ethernet1/0
```

La table VRF contient que les routes des liens directement connectés. Aucun protocole de routage n'a encore été activé entre PE1 et CE2.

Pour propager ses routes au fournisseur de service l'entreprise A à choisi OSPF comme protocole de routage. On va établir une relation de voisinage sur PE1 et PE2 avec les routeurs du client

```
PE1#show ip ospf 6 neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.10.218.1       1   FULL/DR         00:00:37    196.41.224.6    Ethernet1/0

PE2#show ip ospf 2 neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.50.14.2        1   FULL/BDR        00:00:38    196.41.224.2    Ethernet1/0
```

Une fois les relations de voisinages de voisinage fonctionnelles entre PE1 à CE2 et PE2 à CE1, le client peut commencer à envoyer ses routes

```
PE1#show ip route vrf EntrepriseA
Routing Table: EntrepriseA
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 4 subnets
O        10.10.215.1 [110/11] via 196.41.224.6, 00:00:20, Ethernet1/0
O        10.10.216.1 [110/11] via 196.41.224.6, 00:00:20, Ethernet1/0
O        10.10.217.1 [110/11] via 196.41.224.6, 00:00:20, Ethernet1/0
O        10.10.218.1 [110/11] via 196.41.224.6, 00:00:20, Ethernet1/0
      196.41.224.0/24 is variably subnetted, 2 subnets, 2 masks
C        196.41.224.4/30 is directly connected, Ethernet1/0
L        196.41.224.5/32 is directly connected, Ethernet1/0

PE2#show ip ospf 2 neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.50.14.2        1   FULL/BDR        00:00:38    196.41.224.2    Ethernet1/0


PE2#show ip route vrf EntrepriseA
Routing Table: EntrepriseA
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 5 subnets
O        10.50.10.1 [110/11] via 196.41.224.2, 00:27:19, Ethernet1/0
O        10.50.11.1 [110/11] via 196.41.224.2, 00:27:19, Ethernet1/0
O        10.50.12.2 [110/11] via 196.41.224.2, 00:27:19, Ethernet1/0
O        10.50.13.2 [110/11] via 196.41.224.2, 00:27:19, Ethernet1/0
O        10.50.14.2 [110/11] via 196.41.224.2, 00:27:19, Ethernet1/0
      196.41.224.0/24 is variably subnetted, 2 subnets, 2 masks
C        196.41.224.0/30 is directly connected, Ethernet1/0
L        196.41.224.1/32 is directly connected, Ethernet1/0
```

Les routes du client ont été reçues et ajoutées à la VRF nommée EntrepriseA. Maintenant pour propager ces routes aux autres sites du client, il faut les distribuer dans BGP.

Avant de distribuer les routes du client dans BGP il faudrait un moyen de distinguer les routes une fois injectées dans BGP, pour cela on utilisera ce qu'on appelle le Route-Distinguisher ou RD, un RD est un champ de 64 bits utilisé pour rendre un préfixe IPv4 unique dans MP-BGP. On appelle la combinaison RD et préfixe IPv4, VPNv4. Une adresse VPNv4 est une adresse de 96 bits. Les 64 bits du RD sont utilisé avec le format suivant ASN:nn, ASN c'est le numéro du système autonome et nn est un nombre unique que le fournisseur de service attribue à une VRF.

```
PE1#show vrf detail
VRF EntrepriseA (VRF Id = 1); default RD 36891:1; default VPNID <not set>
PE2#show ip vrf detail
VRF EntrepriseA (VRF Id = 1); default RD 36891:1; default VPNID <not set>
```

Nous avons associé à la VRF EntrepriseA le RD 36891:1 sur les deux routeur PE1 et PE2.

Les adresse VPNv4 permettent à BGP de distinguer entre les routes IPv4 régulières et les routes VPN. Mais ne permettent pas à un routeur de savoir à quelle VRF appartient un préfixe VPNv4 reçue dans un message BGP UPDATE. Pour déterminer à quelle VRF appartiennent les préfixes des messages UPDATE on utilise des Route-Target ou RT. Un RT est une communauté étendue, elle permet de déduire quelle route un PE doit extraire d'MP-BGP, et dans quelle VRF les insérés.

On configure le RT pour chaque VRF individuellement, voici la configuration des VRF de PE1 et PE2 pour la VRF EntrepriseA

```
PE1#show run | S ip vrf
ip vrf EntrepriseA
 rd 36891:1
 route-target export 36891:1
 route-target import 36891:1
 ip vrf forwarding EntrepriseA

PE2#show run | S ip vrf
ip vrf EntrepriseA
 rd 36891:1
 route-target export 36891:1
 route-target import 36891:1
 ip vrf forwarding EntrepriseA
```

C'est ainsi que les préfixes des clients restent privés même s'ils sont annoncés à travers le réseau public.

A présent nous devons activer le support de la famille d'adresse VPNv4 sur les routeurs du backbone.

```
PE2#show run | S bgp
router bgp 36891
 address-family vpnv4
  neighbor 172.31.0.1 activate
  neighbor 172.31.0.1 send-community both
  neighbor 172.31.0.1 next-hop-self
  neighbor 172.31.0.2 activate
  neighbor 172.31.0.2 send-community both
  neighbor 172.31.0.2 next-hop-self
 exit-address-family
```

Le support de la famille d'adresse VPNv4 vers P1 et P2 a été activé sur PE2.

Les VRF des clients doivent être redistribués dans BGP comme préfixe VPNv4, voici la commande pour redistribuer les routes OSPF de CE1 dans BGP

```
PE2#show run | S bgp
router bgp 36891
 address-family ipv4 vrf EntrepriseA
  redistribute ospf 2
 exit-address-family
```

Après la redistribution, on peut retrouver les préfixes de CE1 dans la table BGP pour la famille d'adresse VPN4 de PE2

```
PE2#show ip bgp vpnv4 vrf EntrepriseA
BGP table version is 19, local router ID is 172.31.0.3
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 36891:1 (default for vrf EntrepriseA)
```

```
 *>  10.50.10.1/32    196.41.224.2            11         32768 ?
 *>  10.50.11.1/32    196.41.224.2            11         32768 ?
 *>  10.50.12.2/32    196.41.224.2            11         32768 ?
 *>  10.50.13.2/32    196.41.224.2            11         32768 ?
 *>  10.50.14.2/32    196.41.224.2            11         32768 ?
 *>  196.41.224.0/30  0.0.0.0                  0         32768 ?
```

BGP va propager les adresses de CE1, d'ailleurs on les retrouve dans la VRF de PE1

```
PE1#show ip route vrf EntrepriseA
Routing Table: EntrepriseA
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 9 subnets
O        10.10.215.1 [110/11] via 196.41.224.6, 05:20:00, Ethernet1/0
O        10.10.216.1 [110/11] via 196.41.224.6, 05:20:00, Ethernet1/0
O        10.10.217.1 [110/11] via 196.41.224.6, 05:20:00, Ethernet1/0
O        10.10.218.1 [110/11] via 196.41.224.6, 05:20:00, Ethernet1/0
B        10.50.10.1 [200/11] via 172.31.0.3, 00:15:38
B        10.50.11.1 [200/11] via 172.31.0.3, 00:15:38
B        10.50.12.2 [200/11] via 172.31.0.3, 00:15:38
B        10.50.13.2 [200/11] via 172.31.0.3, 00:15:38
B        10.50.14.2 [200/11] via 172.31.0.3, 00:15:38
      196.41.224.0/24 is variably subnetted, 3 subnets, 2 masks
B        196.41.224.0/30 [200/0] via 172.31.0.3, 00:15:38
C        196.41.224.4/30 is directly connected, Ethernet1/0
L        196.41.224.5/32 is directly connected, Ethernet1/0
```

Les préfixes de CE1 marqués avec B ont été insérés dans la VRF par BGP.

Il faudrait maintenant annoncer les routes de CE1 à CE2, pour cela il faut redistribués les préfixes BGP de la VRF dans OSPF.

```
PE2#show run | S ospf
router ospf 2 vrf EntrepriseA
 redistribute bgp 36891 subnets
```

Une fois les routes de CE1 connu par CE2 et vice versa, la communication entre les deux succursales de l'entreprise A peut commencer de façon privée à travers un réseau public.

MPLS VPN s'active automatiquement une fois les VRF complétée. On a capturé un message en provenance de la succursale A vers la succursale B de l'entreprise A.

```
› Frame 53: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
› Ethernet II, Src: aa:bb:cc:00:08:10 (aa:bb:cc:00:08:10), Dst: aa:bb:cc:00:02:11 (aa:bb:cc:00:02:11)
⌄ MultiProtocol Label Switching Header, Label: 18, Exp: 0, S: 0, TTL: 254
    0000 0000 0000 0001 0010 .... .... .... = MPLS Label: 18
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom Of Label Stack: 0
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
⌄ MultiProtocol Label Switching Header, Label: 33, Exp: 0, S: 1, TTL: 254
    0000 0000 0000 0010 0001 .... .... .... = MPLS Label: 33
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
    .... .... .... .... .... .... 1111 1110 = MPLS TTL: 254
› Internet Protocol Version 4, Src: 10.50.10.1, Dst: 196.41.224.6
› Internet Control Message Protocol
```

Figure 7-11 Paquet étiqueté avec deux label MPLS

La présence deux label indique que c'est un paquet MPLS VPN, le label au sommet de la pile (18) a été inséré par LDP pour commuter le paquet à l'Egress LSR, le label 33 qui est le dernier 'Bottom' a été apposé par BGP, ce label sert à orienter le paquet vers la bonne VRF une fois arrivé à l'Egress LSR.

Si on regarde de plus près la VRF VPNv4 de PE1 on retrouve on va trouver que pour envoyer un paquet à la destination 196.41.224.6 situé dans succursale A

```
PE1#show ip bgp vpnv4 all 196.41.224.6
BGP routing table entry for 36891:1:196.41.224.4/30, version 6
Paths: (1 available, best #1, table EntrepriseA)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
    0.0.0.0 (via vrf EntrepriseA) from 0.0.0.0 (172.31.0.4)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:36891:1 OSPF DOMAIN ID:0x0005:0x000000060200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:196.41.224.5:0
      mpls labels in/out 33/nolabel(EntrepriseA)
      rx pathid: 0, tx pathid: 0x0
```

La VRF indique qu'il faut utiliser le label 33 pour la destination 196.41.224.4/30. Ce qui correspond à la capture de paquet de la figure 7-11.

MPLS VPN est désormais opérationnel entre les deux succursales de l'entreprise A.

VII. Références

Beijnum, I. v. (2002). *BGP Building Reliable Networks with the Border Gateway Protocol.* O'Reilly & Associates.

CNRTL. (s.d.). *Politique*. Récupéré sur http://www.cnrtl.fr: http://www.cnrtl.fr/definition/politique

E. Rosen, C. S. (2001, Junuary). *Multiprotocol Label Switching Architecture.* Récupéré sur IETF: https://tools.ietf.org/html/rfc3031

E. Rosen, D. T. (2001, Janvier). *MPLS Label Stack Encoding.* Récupéré sur https://tools.ietf.org: https://tools.ietf.org/html/rfc3032

Ghein, L. D. (2007). *MPLS Fundamentals.* Cisco Press.

Goodin, D. (2010, Nov 17). Chinese ISP hijacked US military, gov web traffic. *theregister*. Récupéré sur https://www.theregister.co.uk/2010/11/17/bgp_hijacking_report/

J. Moy, A. C. (1998, April). *OSPF Version 2.* Récupéré sur https://tools.ietf.org: https://tools.ietf.org/html/rfc2328

J. Scudder Juniper Networks, R. C. (2009, February). *Capabilities Advertisement with BGP-4.* Récupéré sur https://tools.ietf.org: https://tools.ietf.org/html/rfc5492

Jean-Philippe Vasseur, F. L. (2005). *Definitive MPLS Network Designs.* Cisco Press.

Jeff Doyle, e. D. (2001). *Routing TCP/IP, Volume II (CCIE Professional Development).* Cisco Press.

Leahy, E. (s.d.). *bgp overview*. Récupéré sur http://ericleahy.com: http://ericleahy.com/index.php/bgp-overview/

M. Bocci, E. M.-L. (2009, juin). *MPLS Generic Associated Channel.* Récupéré sur tools.ietf.org: https://tools.ietf.org/html/rfc5586

Merriam-Webster. (s.d.). *Policy*. Récupéré sur merriam-webster: https://www.merriam-webster.com/dictionary/policy

Narbik Kocharians, T. V. (2015). *CCIE Routing and Switching v5.0 Official Cert Guide, Volume 2 Fifth Edition.* Cisco Press.

P. Traina, B. R. (2007, aout). *Autonomous System Confederations for BGP.* Récupéré sur tools.ietf.org: https://tools.ietf.org/html/rfc5065

Rosen, E. C., & Inc., B. B. (1982, October). *EXTERIOR GATEWAY PROTOCOL (EGP).* Récupéré sur https://tools.ietf.org: https://www.rfc-editor.org/rfc/rfc827.txt

Smith, P. (2003). *Day in the Life of a BGP Update in Cisco IOS.*

Sue Hares, R. W. (2012, septembre 22). Show 117 – A Rope, A Chair and Helping Hands – Sue Hares and the IETF. (G. Ferro., Intervieweur)

Susan Hares, R. W. (2017, septembre). Show 355: What's Wrong With BGP? – IETF 99. (D. CONRY-MURRAY, Intervieweur)

T. Bates, C. S. (2007, Janvier). *Multiprotocol Extensions for BGP-4.* Récupéré sur tools.ietf.org: https://tools.ietf.org/html/rfc4760

T. Bates, E. C. (2006, Avril). *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP).* Récupéré sur tools.ietf.org: https://tools.ietf.org/html/rfc4456

Y. Rekhter, T. Li, S. Hares. (2006, Junuary). Récupéré sur IETF: https://tools.ietf.org/html/rfc4271