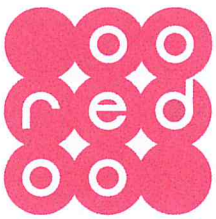
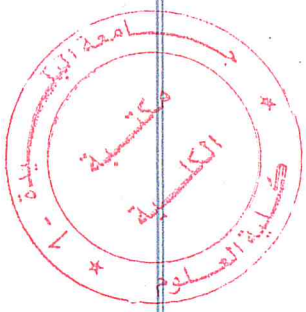
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de laRecherche Scientifique

Université de Saad Dahleb – Blida





Faculté des Sciences
Département d'informatique

Mémoire de MASTER
Domaine : Mathématique et Informatique
Spécialité : Informatique
Option :(Systèmes d'informatiques et réseaux-SIR)

THEME

Optimisation des réseaux IP/MPLS dans l'architecture QoS/diffserv

Proposé et dirigé par :

M. SAGHIR Moncef
M. KHETTAL Mohamed

Promoteur:

M. OULD KHAOUA Mohamed

Soutenu le /06/2018

Devant le jury composé de :

Président de jury :Mme.BOUTOUMI
Examinateur :M.DOUGA

Présenté par :

Melle. MANSOUR Warda
Melle. FELLAH Karima

Promotion : 2017-2018

Dédicace

Je dédie ce travail :

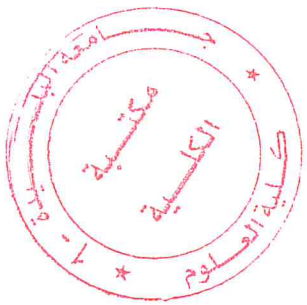
A mes très chers parents qui m'ont soutenues durant toute ma vie, et qui m'ont donné la volonté d'arrivé jusque-là.

A mon frère « Redha » et mes sœurs « Fatima et Meriem » qui m'ont encouragé et que je les Considères comme un exemple.

A mon Binôme « Karima »

A mes chères cousines et mes amis qui m'ont toujours soutenu…

Warda



Dédicace

A mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

A mes frères : yacine, Abdghani, mourad et mes sœurs : Lilia Amira ,Saida qui m'ont soutenus et encouragés tout au long de mes années d'études..

A tous mes amis qui m'ont toujours soutenu

karima

Remerciement

Qu'il nous soit permis d'exprimer nos profonde gratitude à :

ALLAH Tout Puissant, pour nous avoir donné la force, la santé et la patience nécessaire pour accomplir ce travail ; et nous avoir permis de le mener à bien.

Nous tenons à remercier chaleureusement notre promoteur « Mr.Ould khaoua Mohamed » et L'encadreur «Mr.Moncef Saghir» ainsi que le Co-Encadreur «Mr.Khettal Mohamed» Pour leurs présence leurs paissance et leurs bonté..

Nous remercions également les membres de jury, devant qui nous avons l'honneur de présenter notre travail, et qui ont pris la peine de lire avec soin, ce mémoire pour juger son contenu.

Nos remerciements vont aussi à tous ceux qui nous ont aidé, et contribué, de près ou de loin à la réalisation de ce modeste travail.

Résumé

Nous avons effectué notre projet de fin d'étude au sein de l'opérateur **SPA WATANIYA TELECOM ALGERIE « Ooredoo »** au niveau du département **« Network Management Support »** et plus précisément au sein du service **« IP/MPLS »,** La finalité de notre recherche été de trouver une solution permettant d'optimiser et améliorer la qualité de service **« QoS »** dans un réseau MPLS. Toute on se basant sur divers protocoles de routage permettant ainsi à l'opérateur l'exploiter entièrement son infrastructure de réseau etd'assurer la bonne qualité de service **« QoS »,** souhaitée par le client.

Abstract

We have conducted our Master research project at the private telecommunication operator **SPA WATANIYA TELECOM ALGERIA « Ooredoo »** at the department level of **« Networ Management Support»** within the **«IP /MPLS»** section , The purpose of our research was to find a solution allowing the optimization and the improving of quality of service **« QoS »** in an MPLS network, based on various routing protocols so that an operator can fully exploit its network infrastructure and ensure the quality of service **« QoS »** requierd by the customer.

ملخص

قمنا بتحقيق بحوثاتنا التطبيقية الخاصة بمشروع مذكرة نهاية الدراسة ماستر2 على مستوى **الشركة الوطنية للاتصالات الجزائرية** المعروفة بالاسم التجاري**« Ooredoo»** بمستوى دائرة **« دعم إدارة الشبكة»** قسم **«IP / MPLS »** لإيجاد حل لتحسين نوعية الخدمات في شبكة **« MPLS»** الذي يستند على بروتوكولات التوجيه المختلفة بحيث يمكن للمشغل استغلال البنية التحتية للشبكة بشكل كامل وضمان جودة الخدمة التي يرغبها الزبون.

Table des matières

Listes des figures

Chapitre I : Le réseau IP/MPLS

Chapitre II : Les Services MPLS

Chapitre III : L'étude de trafic « Ooredoo »

Chapitre IV : Implémentation et déploiement.

Liste des tableaux

Chapitre III : L'étude de trafic de réseau « Ooredoo ».

Chapitre IV: Implémentation et déploiement.

	A
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
	B
BE	Best Effort
BGP	Border Gateway Protocol
	C
CBQ	Class Based Queuing
CS	Class Selector
	D
DRR	Deficit Round Robin
DSCP	Differentiated Services Code Point
	E
EF	Expedited Forwarding
	F
FEC	Forwarding Equivalency Classes
FIFO	First In First Out
FRR	Fast ReRoute
	I
IETF	Internet Engineering Task Force
IOS	Internetwork Operating System
IP	Internet Protocol
	L
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switched Router
	M
MPLS	MultiProtcol Label-Switching
	O
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
	P

PHB	Per Hop Behavior
PQ	Priority Queuing
	Q
QoS	Quality of Service
	R
RED	Random Early Discard
RIO	RED with In and Out
RSVP	ReSource Reservation Protocol
RSVP-TE	Ressource ReSerVation Protocol – Traffic Engineering
	S
SLA	Service Level Agreement
	T
ToS	Type of Service
	V
VPN	Virtual Private Network
VRF	VPN Routing and Forwarding
	W
WRED	Weighted Random Early Discard

Introduction Générale

Introduction :

Avec l'évolution des tailles des entreprises, la croissance des systèmes d'information et la diversification des besoins des applications dans le domaine de transmission de données, la gestion des multiservices s'avère primordiale pour instaurer la notion QoS dans les réseaux. Certes le développement internet et la simplicité du protocole IP font de celui-ci un protocole presque universel, mais son aspect non connecté implique une difficulté d'intégration de service temps réel qui exigent un certain degré de QoS.

Les technologies réseaux qui ont suivi IP et qui visent à améliorer l'acheminement des données ont essayé de remédier au problème de QoS, principalement les réseaux ATM pour le lancement de la transmission des données en temps réel en gérant les classes de trafic.

Mais vu le coût élevé des commutateurs ATM et la difficulté de faire cohabiter ATM avec d'autres technologies réseau, il a fallu concevoir une technologie de commutation qui permettra une meilleure QoS avec la souplesse et la possibilité d'intégration sur différents types de réseaux (Ethernet, FR, ATM…), d'où l'apparition de la technologie de commutation par étiquettes ou MPLS (Multi Protocol Label Switching).

Le MPLS est un standard qui a vu le jour au milieu des années 90 et qui est venu suite aux demandes de plusieurs compagnies qui recherchaient un moyen de simplifier le mécanisme de routage dans les réseaux IP de façon simple et sans aller consulter les grandes tables de routage lors de l'acheminement des paquets.

Le déploiement des applications multimédias exigent une Qualité de Service adaptée. L'amélioration de la capacité de l'Internet devient nécessaire afin de supporter ces nouveaux types de services par le développement de nouvelles technologies.

La QoS comporte une série de paramètres permettant de caractériser les garanties qui peuvent être fournies pour chaque type de flux, tel que la bande passante (kbps), le délai (ms), la gigue (ms) et le taux de perte des données (%). L'intégration de la QoS dans les réseaux est également un sujet de recherche majeur qui suscite l'intérêt de plusieurs équipes de recherches académiques et industrielles .On peut citer entre autres les solutions de la QoS :

- **IntServ (Integrated Service)** : elle repose sur des messages du protocole RSVP (Ressource Reservation Protocol). Ce modèle convient plutôt aux réseaux de petite taille, mais qui ne sont pas vraiment adaptés à l'Internet .

- **DiffServ (Differentiated Services)** : qui permet de traiter les flux séparément selon leurs Caractéristiques (type de service, type d'application), par un routage plus précis et par le traitement différencié des paquets .DiffServ traite les différentes classes de Services dans leur ensemble et non plus flux par flux. Il place simplement les différents types de trafic dans des files d'attentes différentes selon leur priorité dans les routeurs.

Bien que ces techniques apportent des éléments de solutions, il faut noter qu'elles ne garantissent pas complètement la QoS de bout en bout et doivent être complétées par des techniques de routage adaptatif .Des mécanismes d'ingénierie de trafic (TE) s'avèrent, par conséquent, nécessaires pour répondre à tous ces besoins. [18]

L'ingénierie de trafic représente l'ensemble des fonctions permettant de contrôler l'acheminement du trafic dans le réseau afin d'optimiser l'utilisation des ressources et de réduire les risques de congestion tout en garantissant la QoS.

L'évaluation de performances d'un réseau peut être étudiée en utilisant des simulations adéquates. En effet, ces outils possèdent l'avantage de pouvoir étudier facilement des réseaux de tailles et de complexités différentes. [18]

Objectifs de recherche :

La technologie MPLS a été de plus en plus déployée dans le réseau IP principal, Un certain nombre de tentatives ont été faites pour augmenter le MPLS avec des mécanismes de

QoS, tel que le DiffServ et le IntServ. Aussi TE a également été suggéré pour MPLS pour aider à contrôler l'utilisation des liens et améliorer l'utilisation de la bande passante.

Dans notre travail déployé au sein de l'opérateur de télécommunication « Ooredoo » pour but de proposer une meilleure optimisation de l'infrastructure d'un réseau de télécommunication IP/MPLS dans l'architcture DiffServ. Nous exposons une étude comparative entre les protocoles LDP et RSVP-TE en appliquant les nouveaux mécanismes de la qualité de service. Nous proposons, aussi l'amélioration de la QoS et du routage à travers le protocole MPLS en vue de l'optimisation de l'utilisation des ressources, la minimisation des risques de congestion et la réactivité en cas de panne ou d'instabilité du réseau.

Organisation de document:

1. L'étude des différents protocoles utilisés au niveau d'un réseau IP MPLS.
2. L'étude et la classification du trafic acheminé via le réseau Ooredoo.
3. Concevoir une solution pour la gestion de la QoS afin gérer au mieux le flux de Trafic.
4. Valider la solution en l'implémentant sur une maquette.
5. Superviser le réseau sous l'outil de supervision « PRTG ».

Le réseau IP/MPLS

Introduction :

Un réseau en général est le résultat de la connexion de plusieurs machines entre elles, afin que les utilisateurs et les applications qui fonctionnent sur ces dernières puissent échanger des informations. Le terme réseau en fonction de son contexte peut avoir plusieurs significations. Il peut désigner l'ensemble des machines ou l'infrastructure informatique d'une organisation avec les protocoles qui sont utilisés, ce qui est le cas lorsque l'on parle d'Internet.

Le terme réseau peut également être utilisé pour décrire la façon dont les machines d'un site sont interconnectées. C'est le cas lorsqu'on dit que les machines d'un site (sur un réseau local) sont sur un réseau Ethernet, Token Ring, réseau en étoile, réseau en bus,... Il peut aussi spécifier le protocole qui est utilisé pour que les machines puissent communiquer entre elles.

Aujourd'hui, les réseaux se retrouvent à l'échelle planétaire. Le besoin d'échange de l'information est en pleine évolution. Pour se rendre compte de ce problème il suffit de regarder comment fonctionnent des grandes sociétés en se basant sur des réseaux de transport.

Dans les années 90, avec l'évolution de l'informatique, les réseaux de télécommunication ont marqué l'intégration de nouveaux services tels que la téléphonie IP (*Internet Protocol*), la vidéoconférence, la diffusion audio et vidéo etc. Qui sont des applications très exigeantes en termes de délai et de débit, ce qui a conduit à une explosion de trafic dans les réseaux à cette époque. En effet, Les architectures et les technologies réseaux qui existaient auparavant, elles avaient été conçues pour la transmission de données de faibles tailles d'une adresse source vers une autre adresse de destination.

Cette intégration des nouveaux services dans des réseaux de performance, fiabilité et capacité limitées a pénalisé la qualité de transmission: retard de transfert énorme, taux de perte et d'erreurs considérables, coûts de traitement de données élevés etc. Ce type de problème sérieux, a mené les opérateurs de *télécommunication* à augmenter la qualité, capacité et les fonctionnalités de leurs réseaux en immigrant vers les réseaux de prochaine génération NGN (*Next Generation Network*) qui répondent aux exigences de la qualité de transmission et de service en générale et notamment ils permettent le transport des flux en temps réel d'où la naissance de la technologie IP/MPLS (*Internet Protocol / Multi-Protocol Label Switching)*.

.

Dans ce chapitre, nous allons exposer le réseau IP/MPLS, des notions de base seront

présentées dans un premier temps, en suite une description de différents éléments composant cette infrastructure, après on expliquera le fonctionnement de ce type de réseau ainsi que quelques 'une de ses particularités.

I. Les réseaux IP/MPLS

I.1. Eléments d'un réseau MPLS :

La technologie MPLS est fondée pour transporter des paquets IP sur des sous-réseaux travaillant en mode commuté. Les nœuds contenant dans MPLS sont des routeurs-commutateurs capables de remonter soit au niveau IP pour effectuer un routage, soit au niveau trame pour effectuer une commutation pour cette raison le IP/MPLS est considéré de la couche 2.5 du modèle OSI (*Open System Interconnection)*. [1]

Un domaine MPLS est composé de deux sortes de routeurs : les LER (*Label Edge Router*) et les LSR (*Label Switch Router*). Sachant que les LSR sont les routeurs de cœur de MPLS et les LER sont des routeurs de bordures permettant de faire la transition entre le domaine MPLS et les autres réseaux de clients IP.

I .1.1. Le LSR :

C'est un routeur de cœur du réseau MPLS qui effectue la commutation sur les labels et qui participe à la mise en place du chemin par lequel les paquets sont acheminés. Lorsque le routeur LSR reçoit un paquet labélisé, il marque la permutation en remplaçant une étiquette (Label) entrante avec une nouvelle étiquette sortante selon le chemin LSP, cette opération de permutation est appelée « *SWAP* ». [1]

I .1.2. Le LER :

Il s'agit d'un routeur d'accès qui gère le trafic entrant dans le réseau MPLS et possédant à la fois des interfaces IP traditionnelles et des interfaces connectées au réseau MPLS. On en distingue deux types:

- ***Ingress* LER :** Ce type de nœud gère les trafics entrant dans un réseau MPLS, notamment la distribution de labels, c'est ce qu'on appelle l'opération « *PUSH* ».
- ***Egress* LER :** Il gère les trafics sortant du réseau MPLS, notamment la suppression des traces MPLS (Labels) sur les paquets IP, c'est ce qu'on appelle l'opération « *POP* ». [1]

I.1.3. L'entête MPLS :

L'entête MPLS a une taille de 4 octets et il est composé des champs suivants:

- **Label** (20bits): la valeur du label.
- **Exp** (3bits): indique la classe de service au quelle le paquet appartient.
- **S** (1bit): représente un empilement de labels. Le bit "S" est à 1 quand le dernier label de la pile est atteint.
- **TTL** (8 bits): « Time To Live » représente la durée de vie du paquet, il sert à éviter de faire circuler les paquets en boucles infinies.

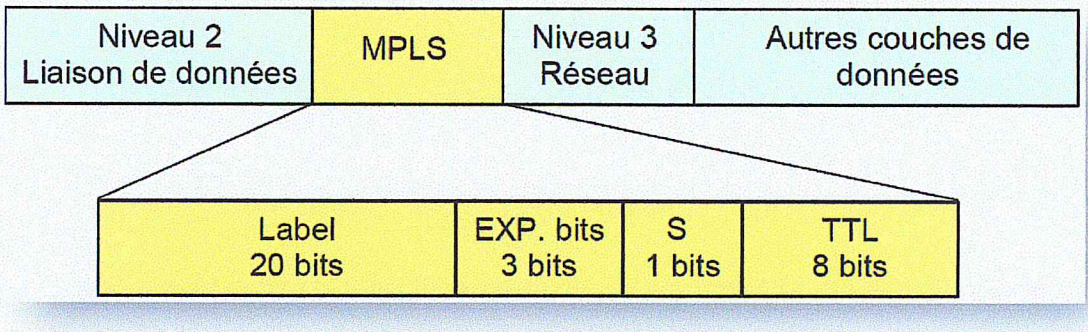


Figure I.1 : Format de base de l'entête MPLS.

I .1.4. Les étiquettes (Label) :

Les labels sont des simples nombres entiers codés sur une courte valeur de longueur fixe de 20 bits affectée à un paquet circulant à l'intérieur du réseau MPLS. Un label a une signification locale entre deux routeurs LSR adjacents et il mappe le flux de trafic entre eux.

A chaque bond le long du chemin traversé par le paquet, un label est utilisé pour chercher des informations de routage (*Next Hop*, interface de sortie).

Les actions à réaliser sur le label sont les suivantes: insérer (*PUSH*), permuter (*SWAP*) et retirer (*POP*). [11]

I .1.5. Le FEC (Forwarding Equivalence Class) :

Il représente un groupe de paquets ayant les mêmes propriétés (adresse IP source, adresse IP de destination, paramètres de QoS…). Tous les paquets d'un tel groupe reçoivent le même traitement au cours de leur acheminement. Ainsi, les paquets d'un même FEC possèdent la même valeur de Label. [11]

I .1.6. Le LSP (Label Switched Path) :

C'est un chemin pour un paquet de données (FEC) dans un réseau basé sur MPLS où une séquence de labels à chaque nœud du chemin allant de la source à la destination.

Les LSP sont établis avant la transmission des données ou lors la détection d'un flot qui souhaite traverser le réseau. Le LSP est unidirectionnel donc le trafic de retour doit prendre un autre LSP.

I .2. Les protocoles utilisés par IP/MPLS :

Le réseau IP/MPLS se base sur plusieurs protocoles de routage et de signalisation afin de transporter l'information de bout en bout, dont les protocoles suivants :

I .2.1 Le protocole OSPF (Open Shortest Path First) :

C'est un protocole d'accès réseau interne, utilisé pour faciliter le routage du trafic dans les réseaux de longue distance, il fonctionne sur la base d'un algorithme qui calcule le plus court chemin d'un point de réseau à un autre en établissant un arbre logique via l'utilisation d'une base de données distribuées qui permet de garder en mémoire l'état des liaisons.

L'OSPF est un protocole de routage intra-domaine, c'est-à-dire qu'il ne diffuse les informations de routage qu'entre les routeurs appartenant à un même système autonome (AS), sachant que ce dernier est divisé en plusieurs zones de routages qui contiennent des routeurs intermédiaires, cette division introduit le routage hiérarchique. Chaque zone (*Area*) possède sa propre topologie et ne connaît pas les topologies des autres zones du système autonome.

La zone *backbone* est une zone particulière constituée d'un ou plusieurs routeurs interconnectés et devant être le centre de toutes les zones. Autrement dit, toutes les zones doivent être connectées physiquement au *Backbone*, afin de limiter le trafic de routage, de réduire la fréquence des calculs du plus court chemin par l'algorithme de l'OSPF et d'avoir une table de routage plus petite, ce qui accélère le fonctionnement de ce protocole. [12]

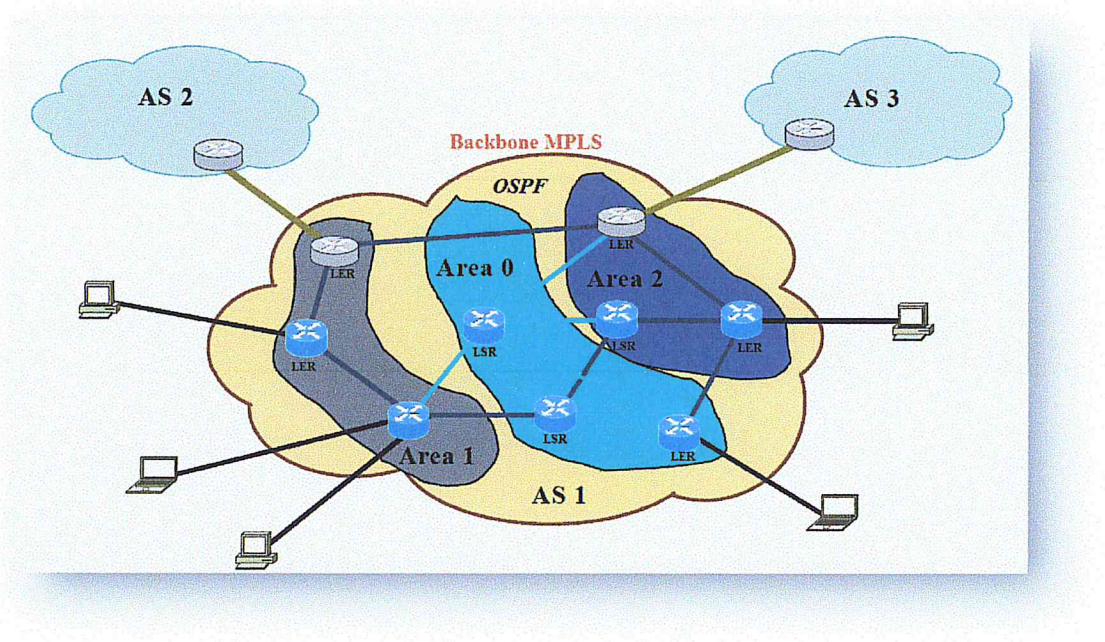


Figure I.2 : le protocole OSPF sous l'architecture IP/MPLS

*I .2.2. Le protocole BGP (*Border Gateway Protocol*) :*

Le protocole BGP est un protocole de routage entre AS, les informations de routage se présentent sous la forme de la suite des numéros d'AS à traverser pour atteindre la destination. C'est le BGP qui assure la propagation des routes à l'échelle mondiale, c'est un protocole très robuste et très « *Scalable* » : les tables de routage BGP peuvent comprennent plus de 90000 routes.

Quand le BGP est utilisé entre AS, le protocole est connu sous le nom de BGP Externe (E-BGP). Si BGP est utilisé à l'intérieur d'un AS pour échanger des routes, alors le protocole est connu sous le nom de BGP interne (I-BGP). Les routeurs de frontière (LER) communiquant par i-BGP doivent être complètement maillés (mais pas directement connectés).

Les voisins BGP échangent toutes leurs informations de routage au début, lors de l'établissement d'une connexion TCP (*Transmission Control Protocol*) entre eux. Par la suite, les routeurs BGP n'envoient à leurs voisins que des mises à jour des tables de routage lorsqu'un changement de route est détecté ou qu'une nouvelle route est apprise.

BGP est un protocole conçu pour annoncer des préfixes. A la base, il est pensé pour IPv4 mais s'est étendu à d'autres formats d'adresses. Ce sont ces extensions qui font qu'on parle de MP-BGP. Une extension très connue est l'évolution de BGP pour transporter des préfixes VPN IPv4. A proprement parler il n'y a qu'un protocole BGP et des évolutions sous forme de MP-BGP. [2]

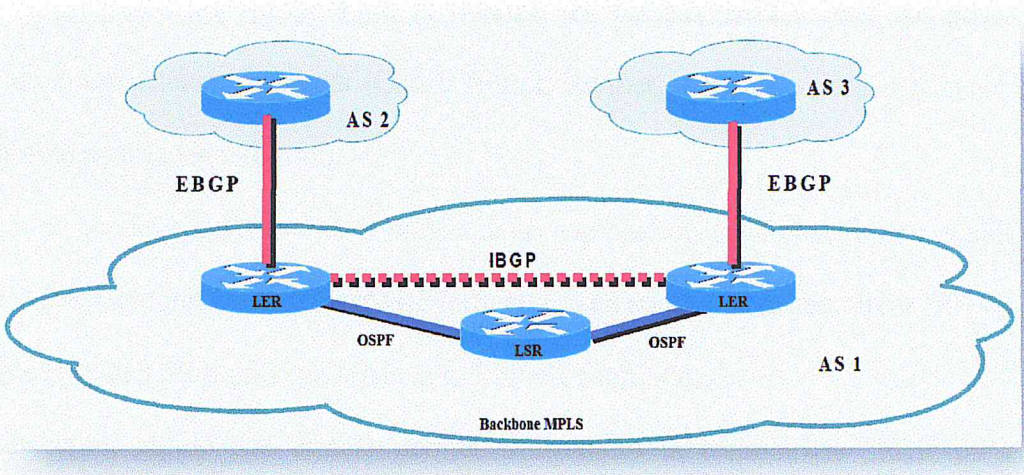


Figure I.3 : le protocole BGP sous l'architecture IP/MPLS.

I .2.3. Le protocole LDP (Label Distribution Protocol) :

C'est un protocole permettant d'apporter aux LSR les informations sur **l'association des labels** dans un réseau MPLS. Il s'utilise pour **associer les labels aux FEC**, et **créer des LSPs**. construit la **table de commutation des labels** sur chaque routeur et il se base sur le protocole OSPF pour la commutation. [7]

I.2.4.Le protocole RSVP-TE (Ressource ReSerVation Protocol – Traffic Engineering) :

Le protocole RSVP-TE est une modification du protocole RSVP (*ReSource Reservation Protocol*) développé par « IETF MPLS working group ». C'est un protocole de réservation des ressources, il permet à MPLS-TE (*MPLS – Traffic Engineering*) de réserver des tunnels (LSP), pour cela le RSVP-TE signale aux routeurs qu'ils vont faire partie d'un tunnel qui sera créé, puis il établira une session dans chaque routeur pour maintenir le tunnel opérationnel.

Ce protocole est très utilisé dans les réseaux MPLS pour deux raisons principales :

Cette méthode est utilisée par les protocoles LDP et RSVP-TE. Le LSP n'est plus déterminé à chaque bond contrairement au routage implicite. Ce qui permet au MPLS de faire de l'ingénierie de trafic afin d'utiliser efficacement les ressources du réseau et d'éviter les points de forte congestion en répartissant le trafic sur l'ensemble du réseau.

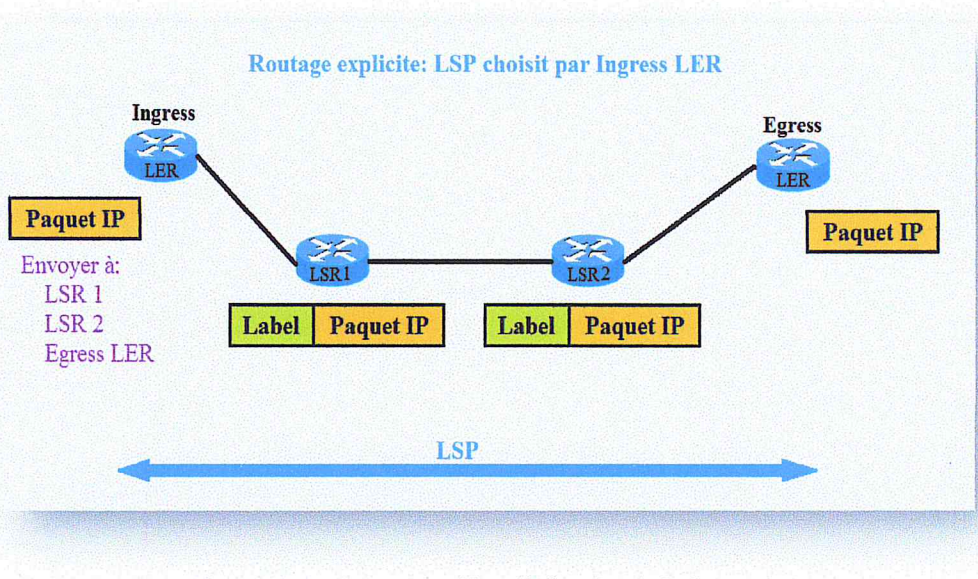


Figure I.7 : Routage Explicite des labels.

Conclusion :

Nous avons commencé ce chapitre par la mise en évidence de la technologie IP/MPLS qui avait été créé initialement pour améliorer les performances des réseaux hauts débit notamment en termes de routage, alors on a exposé les différents éléments de cette technologie d'où on a montré que cette dernière est basée sur la commutation de label qui est plus efficace que le routage classique de paquets IP ce qui simplifie l'acheminement du trafic IP dans les réseaux complexes et qui permet de transporter des types de trafic sensibles au délai telle que la voix sur IP (le transport des paquets sera plus rapide).

Nous avons pu constater dans ce chapitre que le MPLS est fondé sur plusieurs protocoles de routage et de signalisation afin d'assurer son majeur objectif (le transport des paquets IP) en exploitant une de ces pertinentes particularités : les VPN qui rendent les services des clients plus extensibles et flexibles pour faciliter un déploiement à grande échelle, donc les VPN ont formé un nouveau paradigme et ils se sont étendus rapidement à l'ensemble des

entreprises , dans le même temps, ils deviennent de plus en plus sophistiqués et permettent de prendre en charge le contrôle et la gestion des réseaux d'entreprise.

On a vu aussi que le MPLS offre aussi la possibilité d'automatiser et optimiser les liens constituant sa propre infrastructure dans le but d'équilibrer la charge du trafic sur divers liens ou routeurs, afin qu'aucun de ces composants ne soit surcharge ou sous-utilisés. Cela permet donc à un opérateur d'exploiter entièrement son infrastructure de réseau et garantir la bonne qualité de service souhaitée par le client.

Pour toutes ces raisons l'architecture IP/MPLS est devenue la plus utilisée par les opérateurs de télécommunication, et elle demeure toujours en cours de normalisation et de développement.

Malgré toutes ces évolutions, la pratique a montré que l'exploitation de réseau IP/MPLS seul reste toujours incapable de s'adapter aux fortes exigences de certains types de trafics, ce qui rend le service offert par le fournisseur insuffisant pour atteindre la satisfaction des clients à cause des problèmes de transmission au sein de ce type de réseau, ceci a encouragé les chercheurs d'implémenter les principes de trafics engineering « TE » et la qualité de service au niveau de IP/MPLS afin d'avoir une meilleure gestion de réseau où l'apparition de l'architecture QoS/MPLS.

Les services MPLS

Introduction :

A l'apparition des réseaux de télécommunication, l'objectif principal était l'acheminement des paquets d'une adresse source vers une autre de destination, indépendamment de leurs temps de transit ou autres contraintes supplémentaires. Les trafics qui circulaient sur les réseaux étaient de faible taille et non exigeants, et donc les réseaux IP/MPLS fournissant des services qui ont fonctionnés avec succès.

Parmi les services implémentés dans les réseaux IP/MPLS on trouve les VPN qui permettent aux plusieurs sites d'être interconnectés d'une manière transparente à travers un réseau de fournisseurs de service.

Un peu plus tard, l'intégration et le développement de nouveaux services tels que la vidéoconférence, les applications multimédia, etc. Ces derniers ont influencé sur la circulation des paquets dans les réseaux. D'où la qualité de service (QoS) a prouvé son apparence. Pour cela, deux modèles de services ont été proposés : **Intserv** et **DiffServ.** Le premier est utilisé seulement pour les réseaux locaux, car il ne résiste pas au facteur d'échelle, par contre le second résiste au facteur d'échelle, puisqu'il classifie des flux au début ensuite il applique des règles de la QoS en fonction des besoins et contraintes de ces catégories de flux.

Malgré les avantages offerts par la qualité de service (QoS) et les VPN cela reste insuffisant de répondre aux différentes exigences des clients. De nouvelles contraintes en termes de qualité de service (QoS) et de disponibilité du réseau sont apparues. Des mécanismes supplémentaires d'ingénierie de trafic, de QoS et de sécurisation deviennent alors nécessaires.

L'ingénierie de trafic regroupe l'ensemble des méthodes et mécanismes de contrôle du routage permettant d'optimiser l'utilisation des ressources, tout en garantissant la QoS (bande passante, délais...). L'objectif de ces mécanismes est de maximiser la quantité de trafic pouvant transiter dans un réseau afin de retarder l'investissement dans de nouvelles infrastructures.

Dans ce chapitre, on va exposer les différents services Mpls Telle que les VPN .On va aussi citer les paramètres de qualité de service, ensuite on verra les modèles d'architecture de service les plus utilisés, pour présenter au final les mécanismes de la qualité de service toute en intégrant l'ingénierie of trafic « TE » .

II. Les services MPLS:

II.1. Les VPN/MPLS (Virtual Private Network):

Les réseaux privés virtuels VPN sont nés dans un premier temps du besoin des entreprises de pouvoir rendre accessible leur réseau local privé depuis l'extérieur. Par la suite, les VPN ont était développés pour interconnecter les réseaux locaux distants quelque fois de plusieurs milliers de kilomètres.

Les VPN basés sur l'architecture IP/MPLS sont communément appelés BGP/MPLS-VPN, vu que le protocole BGP est utilisé pour la distribution des informations de routage à travers le *Backbone*, et que MPLS est utilisé pour acheminer le trafic d'un site du VPN à un autre. [3]

Pour établir un tel lien ou un chemin ce qu'on appelle un « Tunnel » entre deux sites, les données passent la plupart du temps au travers d'un réseau public comme internet. C'est pourquoi il est nécessaire de sécuriser les connexions. Plusieurs critères entre donc en jeu:

- **Authentification**: identification de l'entité qui tente d'accéder au VPN.
- **Confidentialité**: garantie que seules les personnes autorisées sont en mesures d'accéder à l'information.
- **Intégrité**: protection du flux de données contre d'éventuelles tentatives d'altération ou de dommage par un intrus.

II.1.1. Service L2VPN /MPLS :

Un service de VPN au niveau de la couche 2. Son but est de simuler un réseau LAN à travers l'utilisation d'un réseau MPLS classique. Là encore, la plus grande partie de traitements va s'effectuer sur les PE tout comme les VPNs de niveau 3. Chaque PE maintient une table d'adresses MAC appelée table VFI.

A ce niveau-là, le mapping des FEC s'effectue directement par rapport aux adresses MAC et non les adresses IP. Le principe est similaire à la commutation classique de niveau 2 : Une trame arrive sur un PE qui consulte sa table VFI pour vérifier l'existence de l'adresse et la commute si trouvée. Le cas échéant, le PE, qui émule ce commutateur, va envoyer la trame sur tous les ports logiques relatifs à l'instance VPLS concernée.

II.1.2. Service L3VPN /MPLS :

MPLS où l'on définit des VPN discriminés par un label supplémentaire (en plus du label de commutation). Chaque VPN possède sa propre table de routage IP dans le concept de

Routage et Transfert Virtuel « Virtual Routing and Forwarding » (VRF) impliquant une notion de « Route Distinguisher » et « Route target » (RD et RT).

Un VPN est défini par une collection de politiques qui contrôlent la connectivité d'un ensemble de sites. Ainsi, un site client est connecté au réseau du SP (*service provider*) par un ou plusieurs ports, et le SP associe à chaque port une table de routage VPN appelée « VRF » (*VPN Routing and Forwarding*).

Chaque VPN est associé avec un ou plusieurs VPN de routage ou instances de transmission (VRF). Une VRF permet de créer plusieurs routeurs logiques dans un même routeur physique.

La VRF empêche l'information d'être envoyée en dehors du VPN et permet au même sous-réseau d'être utilisé dans plusieurs VPN sans causer de problèmes d'adresse IP en double.

Le protocole de routage utilisé est BGP. Celui-ci échange les routes annoncées dans le VPN via son extention Multi-Protocol BGP (MPBGP). Comme pour le fonctionnement traditionnel de BGP, il est possible d'utiliser ou non un route reflector (RR).

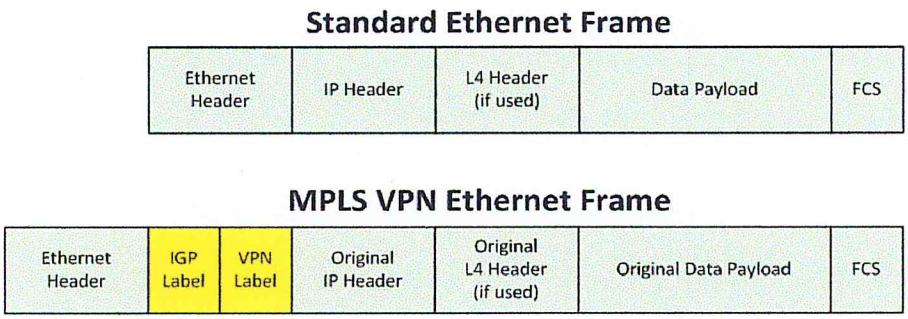


Figure II.1 : Trame Ethernet L3VPN/MPLSMPLS.

Une terminologie particulière est employée pour désigner les routeurs (en fonction de leur rôle) dans l'architecture MPLS / VPN :

- **P** (*Provider*) : ces routeurs, composant le cœur du *Backbone* MPLS, qui n'ont aucune connaissance de la notion de VPN. Ils se contentent d'acheminer les données grâce à la commutation de labels.
- **PE** (*Provider Edge*) : ces routeurs sont situés à la frontière du *Backbone* MPLS et ils ont par définition une ou plusieurs interfaces reliées à des routeurs clients.

➢ **CE** (*Customer Edge*) : ces routeurs appartiennent au client et n'ont aucune connaissance des VPN ou même de la notion de label. Tout routeur « traditionnel » peut être un routeur CE, quel que soit son type ou la version d'IOS utilisée.

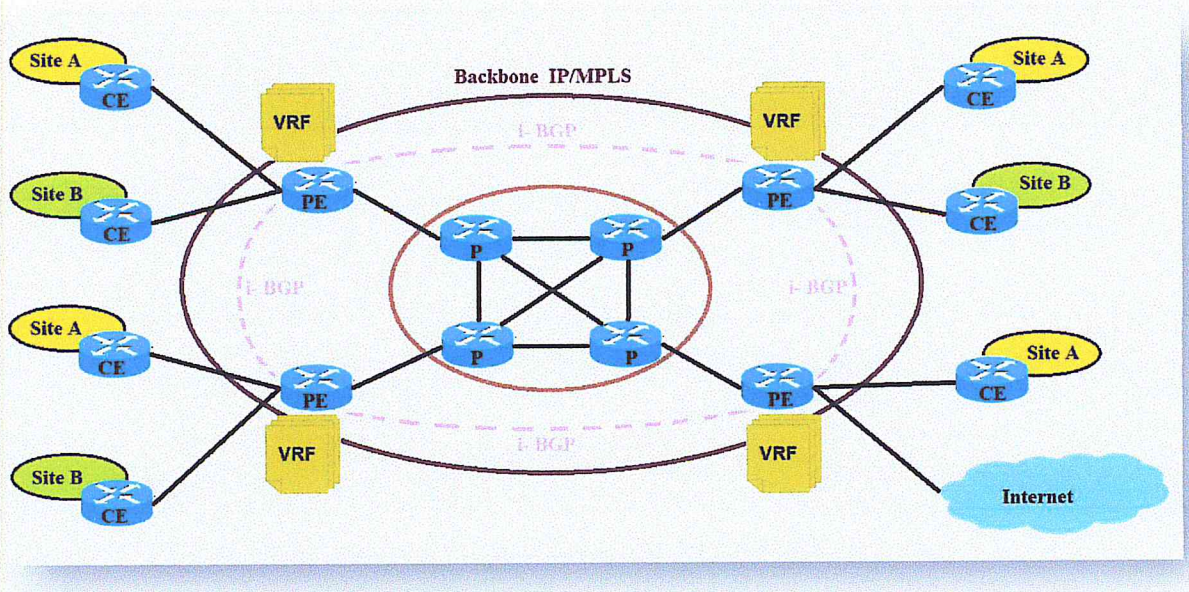


Figure II.2 : L'emplacement des routeurs dans une architecture MPLS/VPN.

II.2. Service de qualité de service (QoS):

II.2.1. Définition de la qualité de service :

La QoS se définit comme étant la capacité à véhiculer dans de bonnes conditions un type de trafic dans un réseau de télécommunication en tenant compte de ses exigences. Ces exigences varient selon le type de trafic. En général, il s'agit d'optimiser les paramètres suivants: délai, débit, gigue, taux de perte, taux d'erreur, coût, priorité etc. [5]

II.2.2 Quelques types de trafics :

Parmi les types de trafic les plus utilisés dans une architecture réseau sont :

➢ **La voix téléphonique :** C'est un trafic de la classe conversationnelle, définie parmi les applications en temps réel différenciées par ses traitements et ses exigences en ressources (applications critiques de haut niveau de priorités).

➢ **La multimédia :** Il existe plusieurs types de données multimédia (la vidéo, les jeux interactives, la visioconférence, etc.)

➢ **Les données (data) :** Les e-mails électroniques, les SMS, etc.

II.2.3 Paramètres de la QoS :

La QoS est un enjeu très essentiel pour un opérateur de télécommunication, c'est la raison pour laquelle elle représente un pourcentage conséquent des principales occupations des opérateurs. En effet, la QoS fournit dans un réseau de télécommunication est toujours visualisée et mesurée afin de la rendre de plus en plus proche de l'optimale. Pour pouvoir assurer et évaluer la QoS, il suffit d'évaluer ses paramètres. Plusieurs paramètres sont présentés : [6]

- **La disponibilité :** La disponibilité d'un réseau se définit comme étant le rapport entre le temps de bon fonctionnement du service et le temps total d'ouverture de service. C'est la forme la plus évidente de la QoS puisqu'elle représente la possibilité d'utiliser le réseau.
- **Le délai :** C'est la durée qu'un flot d'informations nécessite pour arriver à l'adresse de destination à partir d'une adresse source. Il est aussi appelé le délai de bout en bout, le délai d'aller simple ou la latence.

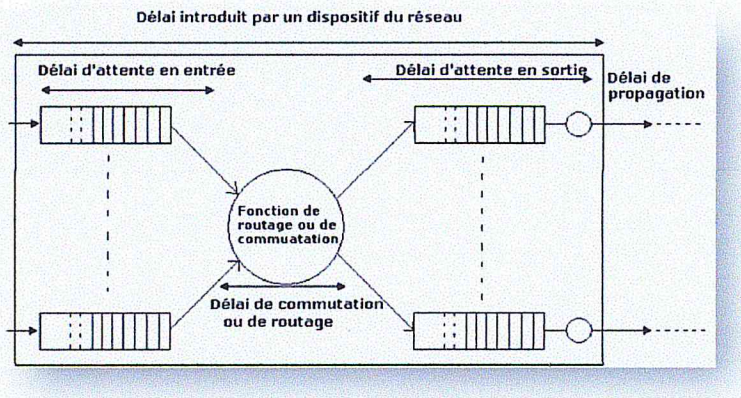


Figure II.3 : Paramètre de délai.

- **Le débit :** Ce paramètre est considéré comme étant le taux de transfert maximum pouvant être maintenu entre deux points terminaux. Il représente le volume d'informations transité dans le réseau par unité du temps.

- **La gigue** La gigue se définit comme la variation des délais d'acheminement des paquets sur le réseau. Ce paramètre est particulièrement sensible pour les applications multimédia qui requièrent un délai inter-paquet (paquets consécutifs) relativement stable. Il dépend principalement du type et volume du

trafic sur le réseau et du type et nombre d'équipements réseau.

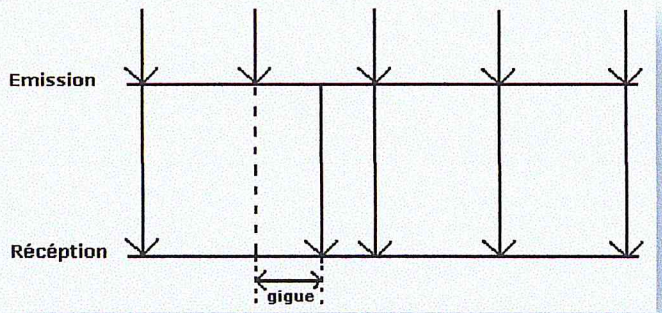


Figure II.4 : Paramètre de gigue.

- **Le taux de perte :** Ce paramètre représente le pourcentage des unités de données (paquets) qui ne peuvent pas atteindre leurs destinations dans un intervalle de temps spécifique, il peut s'exprimer aussi en nombre de paquets perdus par unité du temps.Le taux de perte peut être le résultat d'un rejet de paquets lorsque les ressources sont saturées. Les routeurs peuvent rejeter des paquets pour d'autres raisons tels que :
  - CPU est congestionné et ne peut pas traiter les paquets (la file d'attente d'entrée est saturée).
  - Le routeur détecte une erreur dans le paquet (paquet corrompu).
  - Le routeur rejette les paquets les moins prioritaires en cas de congestion dans le réseau.
  - Une erreur de routage.
  - La fiabilité du dispositif de transmission.

II.2.4. Les types de signalisation :

La signalisation constitue un autre élément essentiel de la qualité de service. Elle permet de réserver et libérer des ressources du réseau, et de diffuser des informations de contrôle à travers un réseau. Il existe deux types fondamentaux de signalisations :

- **Signalisation intra-bande (*In-Band*):** la signalisation portée par l'information elle-même (dans une trame Ethernet on ajoute un champ qui exprime un niveau de priorité) cela veut dire que la priorité du paquet est intégrée dans l'en-tête du paquet IP lui-même.

➢ **Signalisation hors-bande (*out-of-band*):** la signalisation est transportée vers les équipements avant la transmission de données via le protocole RSVP.

II.2.5. Les modèles de la QoS :

Pour pouvoir garantir la QoS des flux transportés selon leurs exigences en terme de délai, bande passante, débit, etc. La mise en œuvre de la QoS a connue au début un service traditionnel « best effort ». Après il a fallu que le réseau soit capable d'isoler les flots pour leur fournir la QoS requise en leur offrant un traitement spécifique. Pour cela l'IETF a défini deux approches de la QoS : Services Intégrés (*IntServ*) et Services Différenciés (*DiffServ*)[13]

II.2.5.1. L'architecture IntServ :

Le service intègre IntServ est un service orienté flot, c'est-à-dire que chaque flot peut faire sa demande spécifique de la qualité de service pour obtenir une garantie précise, C'est un modèle basé sur le protocole de signalisation RSVP.

Dans ce modèle, les routeurs reçoivent une demande via le protocole RSVP, ils peuvent l'accepter ou la refuser. Si la demande est acceptée, l'émetteur peut donc envoyer ces paquets aux routeurs qui vont les placer dans une file d'attente selon la classe de service demandée.

II.2.5.2. L'architecture DiffServ :

Le modèle DiffServ apporte une QoS différenciée pour chaque classe de service (constituée d'une agrégation de micro-flux) selon un contrat prédéfini SLA (*Service Level Agreement*) avec l'émetteur des flux de données. Ce contrat définit un ensemble de paramètres (bande passante garantie, pic de données acceptés, comportement en cas de non-respect du contrat, etc.), ainsi que les micro-flux associés à chaque classe de service. [9]

Un domaine DiffServ définit un ensemble de nœuds réseau appliquant une politique de QoS commune. Il s'articule autour de deux types de traitements:

➢ Les traitements complexes effectués dans les **nœuds d'extrémité**.

➢ Les traitements simples des **nœuds intermédiaires**.

Les routeurs d'entrée d'un domaine DiffServ distinguent les classes des paquets véhiculés en consultant le champ DSCP (*DiffServ Code Point*) au niveau de l'entête IP ensuite à l'aide d'un ensemble de modules appliquant des mécanismes complexes de QoS (*classifier, marking, metering, shaping ou dropping*), par contre les nœuds intermédiaires ne font qu'acheminer les paquets (le routage) selon leur priorité attribuée a l'entré de domaine, ils

font aussi l'ordonnancement et la gestion des filles d'attente alors ils sont des routeurs simples, moins intelligents et moins couteux que les routeurs de bordures.

L'architecture de la QoS/DiffServ est basée sur la notion d'agrégat et la valeur de FEC. En effet les routeurs d'extrémité effectuent un tri pour le trafic entrant dans le réseau selon leurs caractéristiques commun (champs TOS, adresse IP, etc.). Ils affectent une valeur de FEC qui sera fixe pendant le parcours de ce dernier (agrégat) dans le domaine DiffServ, de cette manière les routeurs ne sont pas chargés de mémoriser les caractéristiques de chaque paquet (priorités, classe de service, adresses IP, etc.) mais seulement pour les agrégats (ensemble de flux ont les même caractéristiques). Dans l'architecture DiffServ il existe trois classes de services :

- **Le BE (*Best Effort*) :** Le principe du *Best Effort* se traduit par une simplification à l'extrême des équipements d'interconnexion. Quand la mémoire d'un routeur est saturée, les paquets sont rejetés.

- **Le EF (*Expedited Forwarding*) :** Ce service est destiné aux applications les plus exigeantes en termes de QoS (faible perte, faible délai, faible gigue, et bande passante garantie). Le principe de base et de garantir une taille des files d'attente dans les routeurs de cœur la plus petite possible.

- **Le AF (*Assured Forwarding*) :** Le service AF englobe quatre classes de traitement garantissant chacune une bande passante minimale, une taille de buffer minimale et un délai minimum, sachant que chaque classe comprenant 3 niveaux de priorité entre les paquets d'une même classe AF (*Drop Precedence*). Cette notion de précédence est utilisée pour déterminer l'ordre de rejet des paquets en cas de congestion.

II.2.6. Fonctionnement de la QoS :

Les routeurs qui disposent de la qualité de service sont des routeurs intelligents qui possèdent des fonctionnalités et les mécanismes suivants: la classification et le filtrage de trafic, le contrôle et la gestion de la file d'attente et l'ordonnancement de cette dernière. [6]

II.2.6.1. La classification de trafic (Classifier) :

Avant cette opération le routeur effectuera une phase essentielle qui consiste à identifier le type de trafic (donnée quelconque, voix, trafic multimédia, etc.) pour l'attribuer une classe de service (EF, AF et BE). Il sélectionne les paquets dans un flot selon leurs champs (DSCP, ToS, protocole, etc.) et il détermine également si ces paquets peuvent accéder à leurs classes en se basant sur le contrat SLA passé entre l'utilisateur et l'opérateur.

II.2.6.2. Le contrôle de trafic :

Le contrôle de trafic est assuré par: le métreur (*Meter*), le marqueur (*Marker*), le lisseur (*Shapper*) et le suppresseur (*Dropper / policing*). [15]

➢ **Meter :** Ce mécanisme évalue les caractéristiques des trafics de cette classe (débits moyen, débit maximal, etc.). Si ces caractéristiques dépassent le contrat passé avec l'opérateur (SLA), le Meter indique dans l'entête que le paquet est non valide donc peut par la suite être traité en best-effort ou même supprimé en cas de congestion. Sinon le paquet sera marqué valide.

➢ **Marker :** C'est à ce niveau qui se réalise l'agrégation des flots en classes. Le Marker détermine le PHB (*Per Hop Behavior*) du paquet, et en accord avec les informations transmises par le Meter, positionne le champ DSCP (marquage de la classe). Il est important de noter que cela n'est pas fait par le Classifier, car un même flot suivant les conditions de trafic peut être marqué différemment.

➢ **Shaper :** permet le lissage de trafic en retardant certains paquets de telle sorte qu'il respecte le débit contractuel, il régule les flots suivant les caractéristiques de leur classe.

➢ **Policer et Dropper :** si les paquets ne sont pas conformes, ils peuvent être soit supprimées, soit traitées comme *best effort*, en étant marqué non conforme, avec la possibilité de suppression en cas de surcharge du réseau.

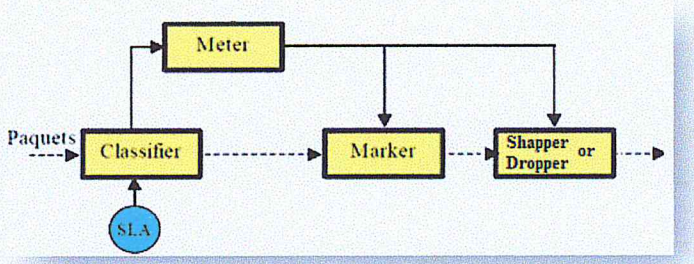


Figure II.5 : Etapes de contrôle de trafic

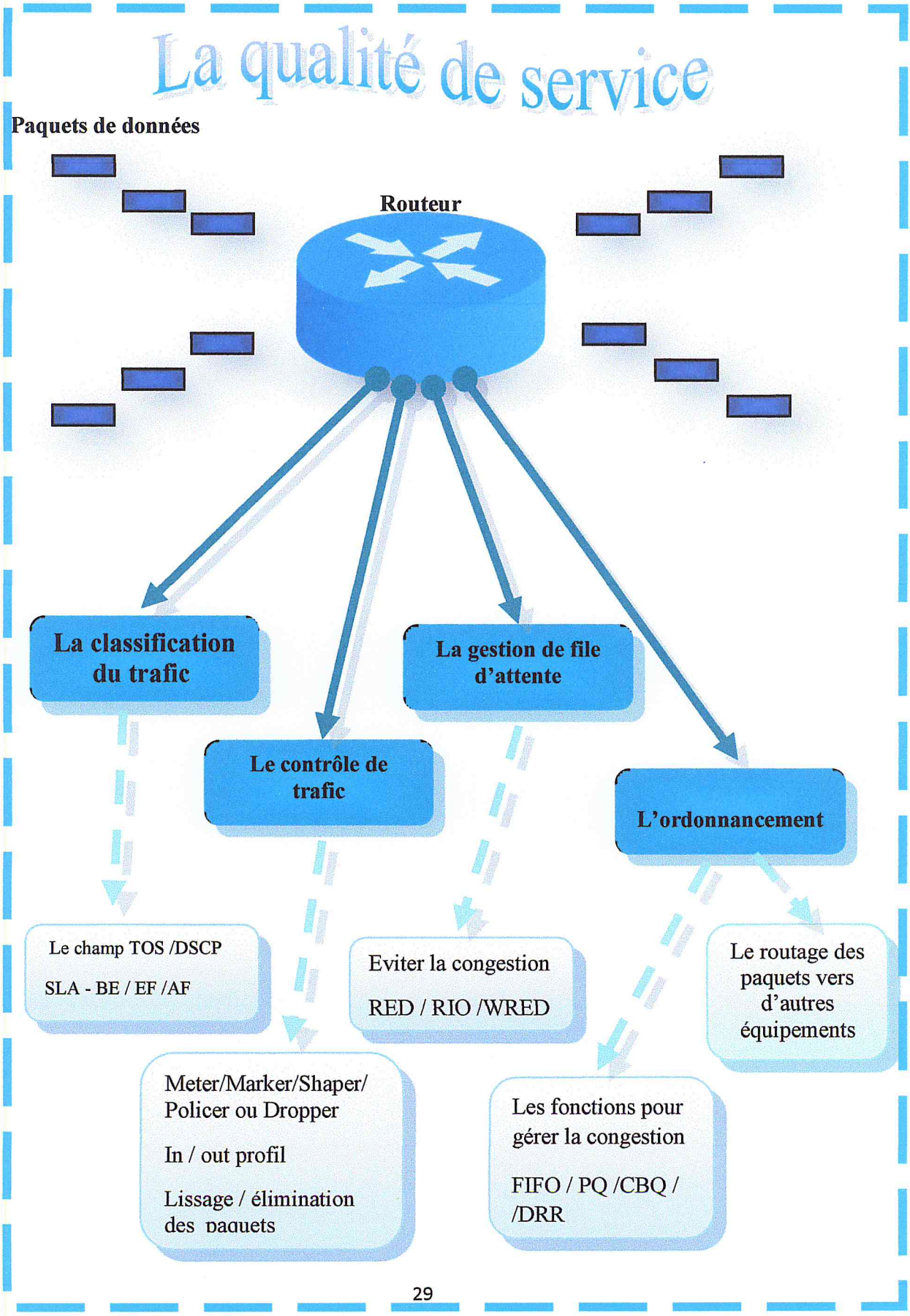
II.2.6.3. La gestion des files d'attente (Buffer) :

La gestion des files d'attente s'occupent de l'opération d'élimination des paquets appartenant à une même file en cas de congestion dans une file d'attente. Pour réduire la taille des files d'attente, il est nécessaire de faire appel à des mécanismes de gestion actifs sur les trafics comme RED, RIO, WRED, etc. [16]

II.2.6.4. L'ordonnancement :

Il consiste à vider les files d'attente vers l'interface de sortie du routeur, plusieurs politiques d'ordonnancement sont implémentées au niveau des routeurs: FIFO - PQ - CBQ - DRR - etc.

II.2.7. Résumé générale des mécanismes QoS :



II.2.8.L'interprétation de la QoS entre IP/MPLS et l'architecture DiffServ :

L'architecture de DiffServ utilise les 8 bits du champ TOS (*Type of Service*) de l'entête de paquet IP et les divise en deux parties, les premiers six bits sont réservés pour classifier le trafic dans des classes DSCP (PHB) selon le contrat SLA au niveau de nœud d'entré de domaine DiffServ dont les trois premiers bits pour la classification et les autres trois bits pour la priorité, tandis que les deux derniers bits ne sont pas pour le moment utilisés (CU).

Comme vous avez sûrement pu le remarquer lorsque nous avons abordé le format du label dans le chapitre précédent que l'entête MPLS contient aussi son propre champ EXP réservé à la qualité de service qui est codé sur 3 bits, alors que les DSCP sont codés sur 6 bits dont on a 3 bits de priorités (8 FEC), cela ne pose pas de problèmes car les 3 bits d'EXP sont suffisants pour stocker les valeurs.

Donc la valeur de 3 bits de classification de champ DSCP sera interprétée directement dans le champ EXP situé dans l'en-tête IP selon la stratégie de fournisseur MPLS.

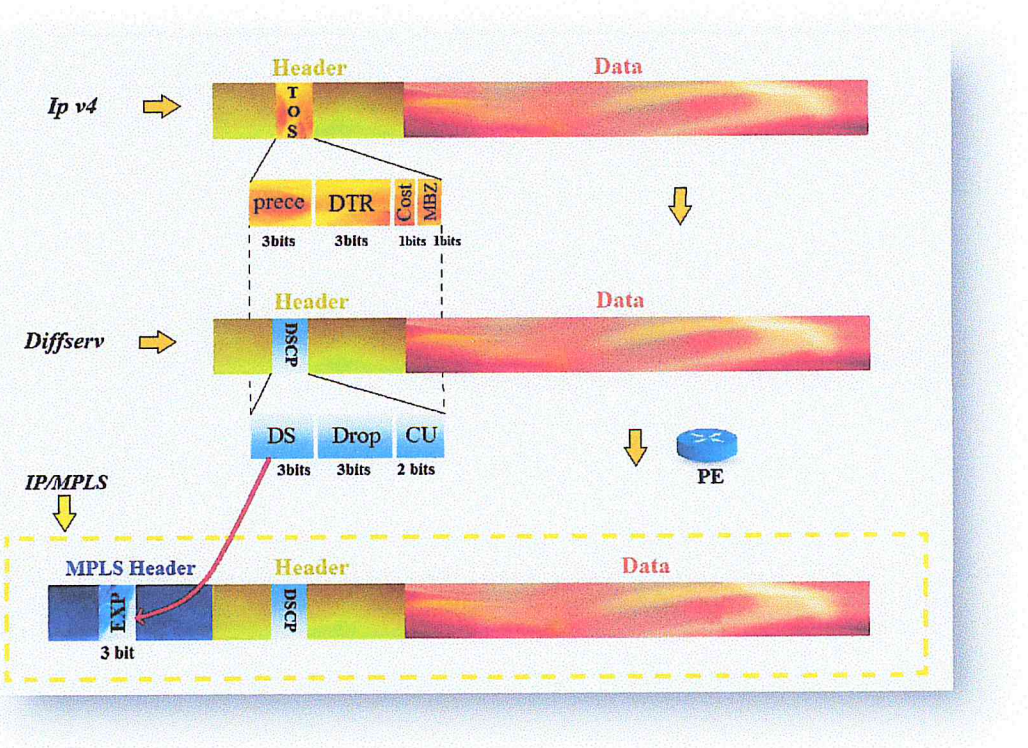


Figure II.6: Interprétation de la QoS au niveau de l'architecture Diffserv/MPLS.

II.3. Service TE (Engénierie of traffic) :

II.3.1. Limitations du routage IP en termes d'ingénierie de trafic :

Avec des réseaux IP, on dispose de peu d'outils pour effectuer à la fois du partage de charge en plusieurs chemins, router explicitement du trafic en fonction de ses qualités et éventuellement réserver des ressources.

Pour assurer ces fonctions, il est nécessaire de combiner des mécanismes de niveau 3, comme les classes de services, le partage de charge, la manipulation de métrique, et des mécanismes de niveau 2, comme la configuration des circuits virtuels qui permet de créer une topologie logique correspondant aux besoins. Ces combinaisons apportent de la complexité, influent généralement sur tout le trafic, et ont donc leurs limites. [17]

II.3.2. MPLS et Ingénierie de trafic :

Le routage explicite proposé par IPv4, IP source routing, est inefficient parce qu'il suppose que chaque paquet contienne la description du chemin emprunté dans le réseau. Cela présente plusieurs défauts majeurs : des risques liés à la sécurité, une surcharge importante des paquets, un traitement complexe dans les routeurs internes du réseau pour chaque paquet. Ce mode n'a jamais vraiment été implémenté et utilisé, il est toutefois repris dans IPv6. [17]

II.3.2.1. Présentation

L'ingénierie de trafic appliquée aux réseaux MPLS est normalisée sous le nom MPLS-TE (Multi Protocol Label Switching - Traffic Engineering)

MPLS-TE permet l'établissement de LSP-TE (Label Switched Path – Traffic Engineering), routés explicitement ou dynamiquement, en fonction de contraintes relative à une topologie TE. Ces LSP-TE peuvent être assimilés à des connexions point-à-point, un mode « circuit » est alors créé dans les réseaux IP/MPLS, s'appuyant sur le routage interne, mais fonctionnant en parallèle.

La technologie MPLS-TE permet également de répondre à des exigences de haute disponibilité et de sécurisation des services notamment temps réels via le mécanisme MPLS-TE Fast-Reroute.

II.3.2.2. Type de réservation:

RSVP-TE propose plusieurs types de réservations :

- **Fixed Filter (FF) :** Une réservation de label est effectuée par nœud émetteur. Ces ressources ne sont pas partagées.
- **Wildcard Filter (WF)** : Une réservation de label est effectuée quel que soit le nombre de nœuds émetteurs. Cette technique présente des avantages pour les connexions multipoints à point. (ex : conférences téléphoniques, etc.)
- **Shared Explicit (SE) :** Il permet au récepteur d'inclure explicitement chaque émetteur dans la réservation. Chaque émetteur a la possibilité de spécifier sa route. Il peut donc exister de multiples LSP.

La réservation des ressources et le choix de LSP se basent sur les paramètres suivants (Bande Passante, Priorité et Affinité).

Affinité : La notion d'affinité est simplement une valeur sur 32 bits spécifiée sur les interfaces des routeurs MPLS. La sélection du chemin s'effectue alors en indiquant une affinité et un masque (sur le routeur initiant le tunnel) et attribué des drapeaux aux interfaces au long de chemin.

Bande passante : C'est la capacité maximale que le tunnel peut utiliser il est attribué pour chaque LSP lors de la création des tunnels.

Priorité : La priorité de mise en place comprise entre 0 et 7 où 0 est la valeur ayant la priorité la plus élevée, elle est utilisée pour déterminer si cette session peut en préempter une autre.

II.3.2.3. Type de tunnels :

Les tunnels MPLS peuvent être créés en indiquant la liste des routeurs à emprunter (méthode explicite) ou bien en utilisant la notion d'affinité (méthode dynamique).

- **Tunnel dynamique**

Chaque nœud prend une décision indépendante de lier une étiquette à une FEC. Il distribue ensuite l'étiquette sur ses nœuds voisins. Ceci est similaire au routage IP classique, chaque nœud prend une décision indépendante de comment transmettre un paquet .

- **Tunnel Explicit**

Les PE spécifient une liste des nœuds par lesquels le flux de données traverse. Le chemin

spécifié peut ne pas être optimal, mais des ressources peuvent être réservées afin d'assurer une certaine qualité de service au trafic de données tout au long du chemin.

Conclusion :

Dans cette section on a vu les principaux services MPLS, telle que les VPN « L2vpn ; L3vpn » .Qui ont été conçus afin de garantir la qualité de service avec des solutions VPN/MPLS .On a vu aussi le service de qualité de service (QoS) qui a fait son apparition avec l'évolution du trafic et afin de répondre aux différents exigences de clients en garantissant différentes contraintes, On a démontrés par la suite les principales architectures de la qualité de service IntServ et DiffServ où la première est utilisée dans les petits réseaux tels que les réseaux d'accès, par contre la deuxième a été implémentée dans les réseaux d'une capacité énorme tels que les réseau mondiaux « internet »..

Avec l'explosion de la surcharge de trafic, les fournisseurs de télécommunication font inventer et implémenter quelques particularités de IP/MPLS dont l'optimisation de l'infrastructure à l'aide de l'ingénierie de trafic « TE » afin de rendre la qualité de service plus proche de l'optimale toute en optimisant les ressources disponible, tels que les liens (Fibre optique, faisceau hertzien…) routeurs, commutateurs... et donc optimiser le coût.

Dans le chapitre suivant, on va essayer d'appliquer le mécanisme de qualité de service (QoS) et l'ingénierie de trafic « TE » sur le réseau backbone MPLS actuel toute en optimisant la capacité de ses liaisons inter-sites.

L'étude de trafic du réseau « Ooredoo »

Introduction :

L'implémentation d'un réseau de télécommunication passe principalement par trois cycles en commençant par la planification puis l'investissement et finalement l'exploitation, et plus ce qu'on a indiqué antérieurement concernant la recherche de l'opérateur à atteindre le meilleurs compromis entre le coût et la qualité de service, l'optimisation et l'ingénierie de trafic ont aussi besoin d'une intervention qui suit les trois cycles cités au début , cette phase est très essentielle qui consiste à faire la surveillance de bon fonctionnement de réseau, elle permet d'intervenir immédiatement en cas d'un problème, d'une panne ou un manque de capacité à véhiculer le trafic des abonnés. Cette étape s'appelle la supervision du réseau qui permet d'offrir des solutions modulaires et extensibles, elle sert à surveiller en temps réel les ressources du réseau grâce à un outil de supervision à travers un déclenchement des alarmes instantanément lorsqu'un incident se produit sur le réseau, donc cet outil va inspecter l'état de tous types d'équipements réseaux (support de transmission, commutateurs, routeurs, état des liens, etc.).

Et afin de faire une étude globale sur le fonctionnement du trafic au niveau de L'opérateur « Ooredoo » nous avons fait une visite avec le responsable sur les différents services Telle que service de transmission ,service managements et plus particulièrement le service Mpls ».Afin de connaitre mieux le fonctionnement de ce service et générer les différents trafic qui consiste à superviser le réseau dans le but d'optimiser le fonctionnement et les performances des équipements pour rendre le réseau de télécommunication « Ooredoo » plus rentable et plus robuste.

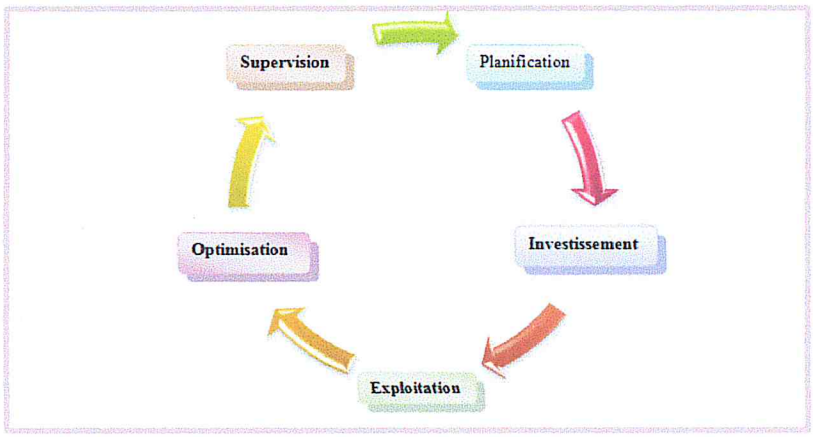
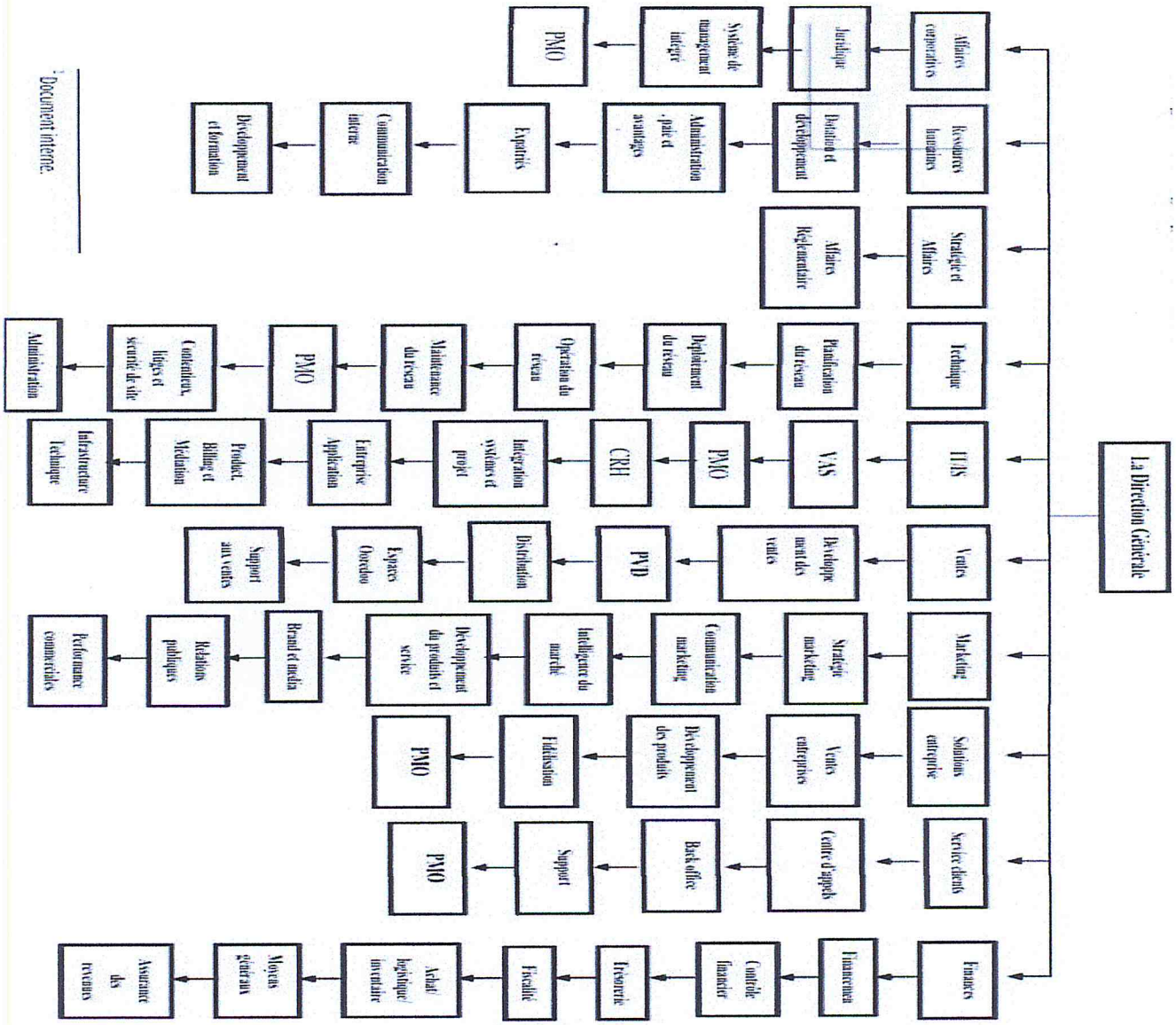


Figure III.1 : Les cycles utilisés par « Ooredoo ».

III.1.Présentation de « Ooredoo »:

Premier opérateur multimédia de téléphonie mobile en Algérie, Nedjma, devenue Ooredoo le 21 novembre 2013, est la filiale algérienne du Groupe Ooredoo. Présent en Algérie depuis le 23 décembre 2003, date d'obtention de la licence de fourniture des services de téléphonie mobile en Algérie, la marque Nedjma a été commercialement lancée le 24 août 2004, en offrant aux Algériens, qu'ils soient clients particuliers ou entreprises, une gamme d'offres et de services novateurs, en respect avec les standards internationaux.



III.2 Figure : Organigramme de « Ooredoo ».

III.2. Fonctionnement des réseaux Mpls :

De nos jours la productivité et l'évolutivité des entreprises dépendent du bon fonctionnement de leur réseau et des performances de la technologie, pour cela « Ooredoo » a mis en place un réseau de nouvelle génération qui s'appuie sur la technologie MPLS, en mettant à la disposition des clients un réseau performant capable de garantir une excellente qualité de service et une souplesse dans la gestion du réseau.

III.2.1. Le core Mpls :

Le réseau « El wataniya Telecom Algérie » est un réseau de nouvelle génération, qui dispose d'une architecture de fonctionnement en couche, Une couche Coeur (Core) et une couche Périphérie (Edge).

Il est composé de routeurs d'une très grande capacité qui sont installés aux niveaux des centres RMS (Réseau Multi-services) :

Routeur Core (P) : Sont des routeurs installés au coeur du Backbone et Ils se contentent d'acheminer les données grâce à la commutation de labels.

Routeur Périphérie (PE) : Sont des routeurs installés à la frontière du Backbone, Le rôle du PE consiste à relier les routeurs clients.

- Les tableaux ci-dessous représente la gamme utilisée par les nœuds de « El Wanatniya Telecom » , ainsi que les différents support d'interconnexion toute en spécifiant leurs type et leurs débits .

Type	Nombre	Plateforme
P	10	Ericsson SSR 8020
PE	30	Ericsson SSR 8004/6672

Tableau III.1 :Type et nombre de routers utilisé par « Ooredoo ».

	Type	Débit
Fibre Optique (FO)	DWDM	1 Gbps
		10 Gbps
		100 Gbps
Micro Wave (MW)	PDH	2 Mbps - 600 Mbps
	SDH	150 Mbps - 2.4 Gbps

Tableau III.2 : Les supports d'interconnexion utilisé par « Ooredoo ».

III.2.2. Architecture du Backonne Mpls « Ooredoo » :

Le réseau de Backbone IP/MPLS de « Ooredoo » est déployé par plusieurs CE « Costumer-Edje » Telle que :

- RNC
- MSS
- SGSN

Qui eux sont attachés par la suite à des routeurs de bordures « PE » qui garanstissent l'acheminement des paquets au sein du backbonne Mpls .

La figure ci-dessous représente le backonne réel de « Ooredoo» :

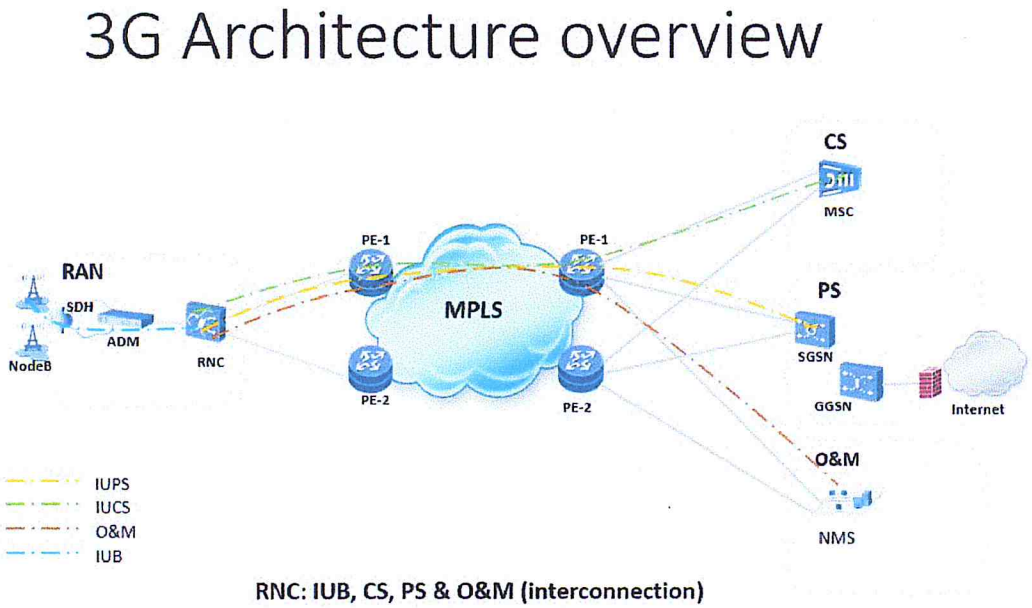


Figure III.3 : L'architecture du Backbonne Mpls de «Ooredoo ».

III.2.3 la structure de Backbonne MPLS :

- Le shéma ci-dessous représente l'architecture réel du backbonne « MPLS » qui est géré dans trois régions principales (Oran/Alger/Constantine).
- Telle que chaque région est liée à plusieurs PE « Provider-Edge » qui sont eux par la suite lié à des CE «Customer-Edge ».
- Chaque CE « Customer-Edge » telle que (Node-b/ Boutique/RNC/SGSN) gère une catégorie de trafic bien précise .

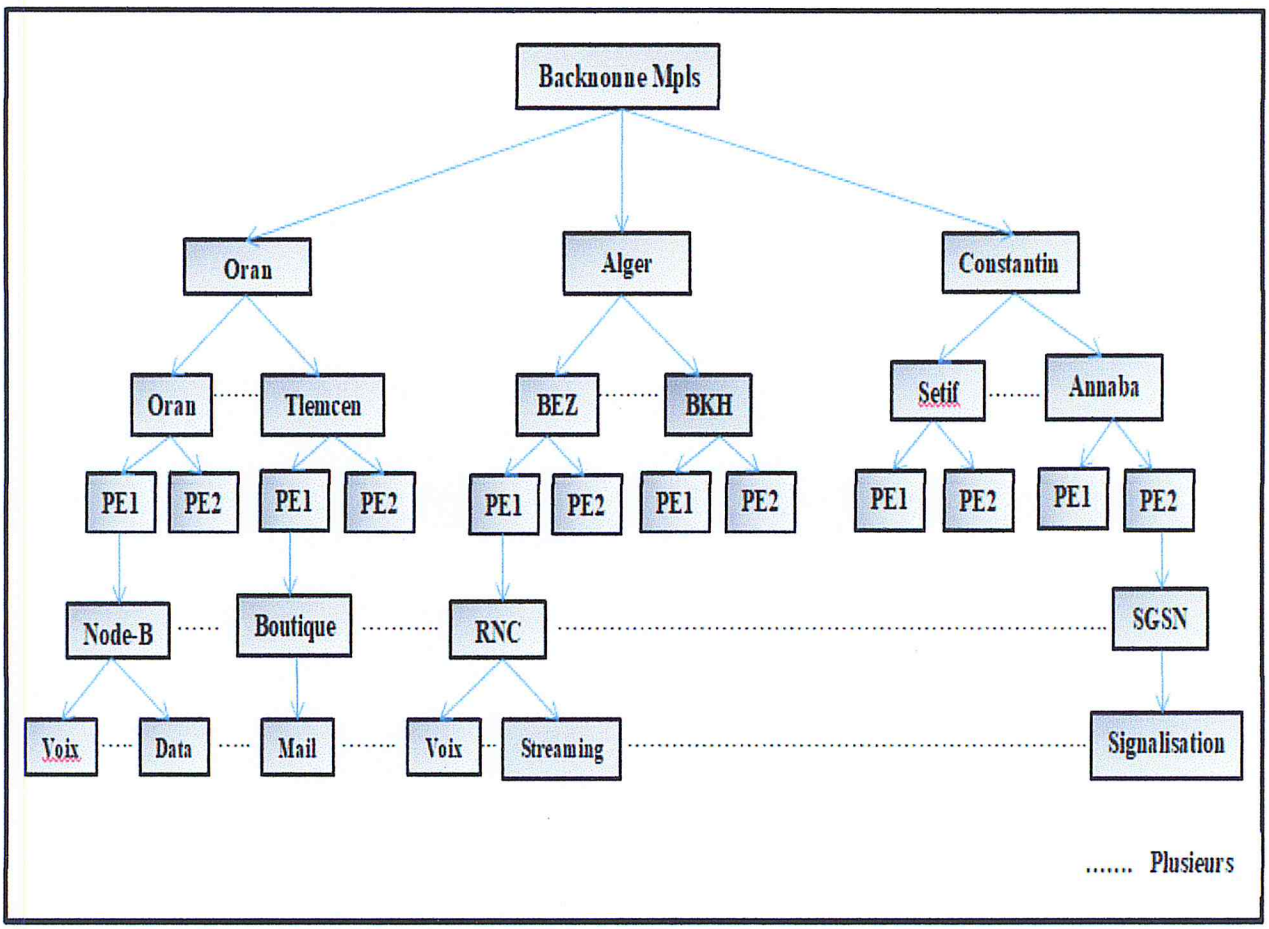


Figure III.4 : La structure de backbonne MPLS.

III.3. Services de « Ooredoo » :

Lors de notre visite à « Ooredoo » , nous avons eu la chance de connaitre les différents services telle que (BSS,NSS,GPRS ,IT) , qui travaillent en collaboration afin de traiter toutes les parties nécessaires « access,core ,data ,ip »afin de satisfaire les besoins des CE « Customer-Edje » .

Equipements	Service approprié	Description du service
RNC Node b Bts Enode b Bsc	Service BSS	C'est le service qui s'occupe de la partie access (RAN) du réseau.
MSC MGW	Service NSS	C'est le service qui s'occupe de la partie core du réseau.
GGSN SGSN	Service GPRS	C'est le service qui s'occupe de la partie data de réseau PS (packet switching).
IGW	Service IT	C'est le service qui s'occupe de la partie IP du réseau.

Tableau III.3 : Les services de « Oredoo »

Le tableau ci-dessous contient les clients de « Ooredoo » , et décrit le rôle de chacun de ces derniers

Type	Exemple	Description
CE	RNC	Un élément de la partie accès radio (UTRAN) d'un réseau de téléphonie mobile **3G UMTS ,** qui contrôle les transmissions radio des stations de base Node-B.
	SGSN	C'est une passerelle parementant l'établissement des données dans les réseaux mobiles GSM ;GPRS ;EDGE et UMTS.
	Node-B	Une station de base dans un réseau mobile UMTS. C'est l'équivalent de la BTS dans les réseaux GSM.
	BTS	Station de transmission de base ou station émettrice- réceptrice de base. Est un des éléments de base du système cellulaire de téléphonie mobile GSM.
	e-Node-B	Station de base des réseaux mobile basés sur les technologies LTE ou LTE advanced. C'est l'équivalent de Node-B dans les réseaux UMTS.
	BSC	Le controleur de station de base est l'un des éléments du réseaux GSM. Son rôle est de commander un certains nombre de BTS.
	MSC	L'équipement de téléphonie mobile (GSM/2G),chargé du routage dans le réseau, de l'interconnexion avec les autres réseaux et de la coordination des appels.
	MGW	Média gateway équipement qui permet la connexion entre des réseaux de la nouvelle génération, et des réseaux fixe ou mobile. Au travers plusieurs protocoles de signalisation.
	GGSN	Une passerelle d'interconnexion entre le réseau packet mobile et les réseaux de IP externes .
	IGW	Internet getway : sortie internet vers l'international.
	Boutiques	Espaces Ooredoo
	Clients ISP	Internet service provider «ISP » offrent une connexion à internet.

Tableau III.4 : Description des clients de « Ooredoo ».

En passant par tous les services, nous avons constaté que le service BSS rencontre plus de problèmes que les autres services à cause du changement des équipements, citons :

« drop ».

Pour mieux comprendre, nous prenons un exemple de un des équipements du service BSS qui est le RNC , les figures suivante présente la laison d'une RNC au backbonne MPLS.

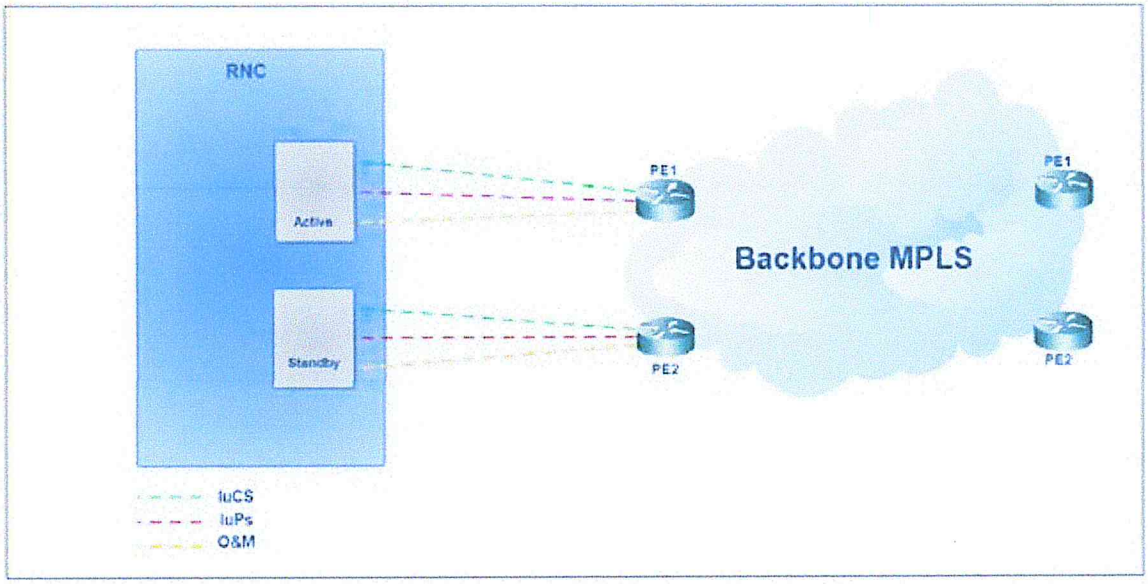


Figure III.5 : Les liasons d'une RNC avec le Backbonne MPLS.

La rénovation des équipements ainsi que la 4G intégré résulte une variation du trafic et une augmentation a été repéré ce à l'aide des graphes ce qui a influé la stabilité et la perfermancs du réseau , nous citons ci-dessous les types de trafic dans le service BSS :

- Signialisation : divisée en deux types :
  - Signialisation voix.
  - Signialisation data.
- Voix.
- Data
- Managment

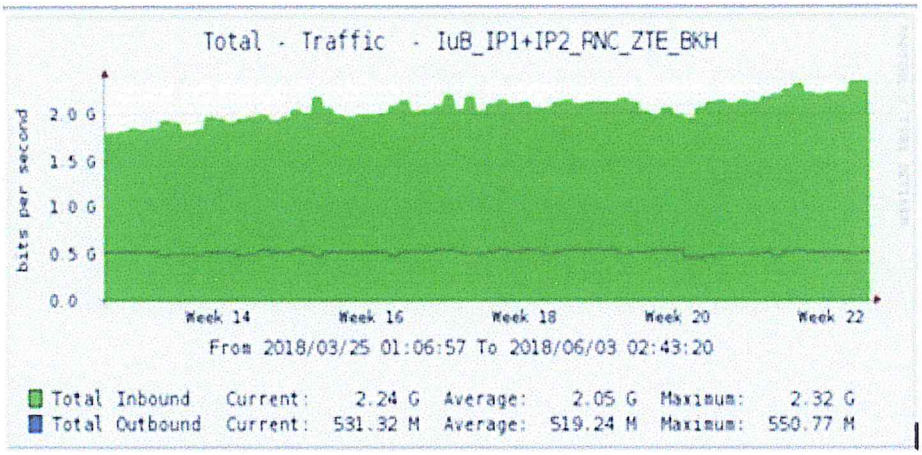


Figure III.6 : Evolution de trafic au niveau de BKH.

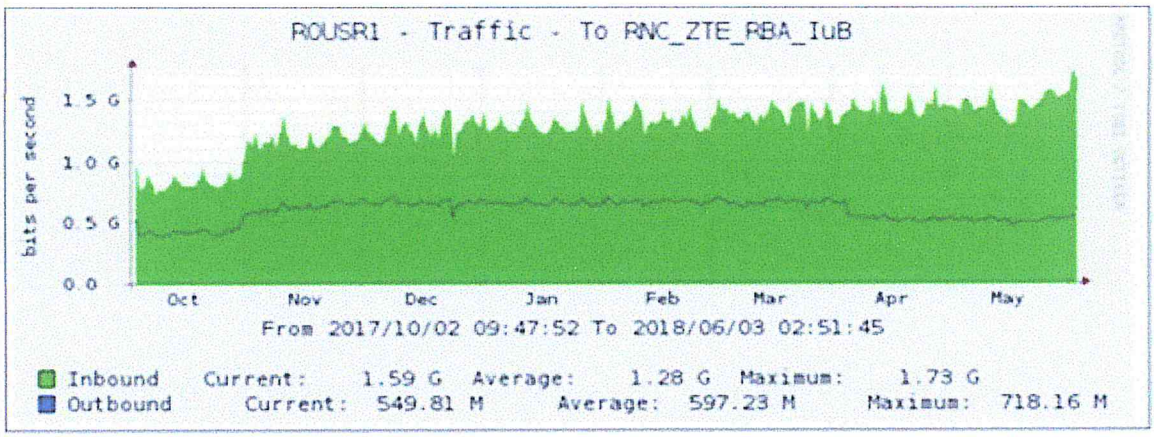


Figure III.7 : Evolution de trafic au niveau de RNC Rouiba.

D'après les graphes nous remarquons que :

- Au niveau du RNC BKH le trafic passe de **1.83G** vers **2.32G** le 25/03 et 03/06.
- Au niveau du RNC Rouiba le trafic passe de **0.8G** vers **1.73G** entre le mois octobre et juin.

III.3. 1 :Le choix du service IP MPLS :

L'opérateur « Ooredoo » se compose de plusieurs services comme nous avons vu précédemment , notre choix a opter pour le service « **IP OPS Networking IP/MPLS et IT** » pour les raisons suivantes :

- Le service qui correspond le mieux avec nos études et notre spécialité.

- Nous permet de se familiariser et de connaitre mieux la gestion de Backbone « **IP/MPLS** » et de transport des donnés de l'opérateur.

III.4. Classification de trafic dans l'architecture « Diff Serv/Mpls » :

On a vu préalablement que le trafic réseau entrant dans un domaine DiffServ est soumis à la classification où les opérateurs de réseaux veulent un contrôle serré sur les volumes et les types de trafic dans une classe donnée.

La circulation de trafic dans chaque classe est soumise d'un comportement PHB qui est déterminé par le champ DSCP de l'en-tête IP.

En pratique, le réseau d'opérateur « Ooredoo» utilise quatre comportements qui sont :

➢ ***Par défaut* PHB** : il contient tout le trafic qui ne répond pas aux exigences de l'une des autres classes définies, typiquement il possède des caractéristiques de transfert BE. Le DSCP recommandé pour ce type est 000000.

➢ ***Expedited Forwarding* (EF) :** il est dédié aux services en temps réel qui nécessite un faible retard, faible perte et faible gigue, Le DSCP recommandé pour le transfert accéléré est 101110.

➢ ***Assured Forwarding* (AF)** : ce comportement garantit un acheminement qui permet à l'opérateur de fournir une assurance de livraison, tans que le trafic ne dépasse pas certains conditions, le AF définit quatre classes suivant d'une priorité de chute (élevée, moyenne ou faible) où il y'aura douze codage DSCP de AF11 jusqu'à AF43 (voir le tableau IV.1).

➢ ***Celector classe* (CS)** ; généralement cette classe est utilisée pour les protocoles de signalisation, son DSCP est sous la forme « xxx000 » où les trois premiers bits sont des bits de priorité IP.

		PHB	DSCP (Bits)	DSCP (Décimal)
		Par défaut (CS0)	000000	0
		CS 1	001000	8
Classe 1 (la plus haute priorité)	priorité de rejet élevée	AF 11	001010	10
	priorité de rejet moyenne	AF 12	001100	12
	priorité de rejet faible	AF 13	001110	14
		CS 2	010000	16
Classe 2	priorité de rejet élevée	AF 21	010010	18
	priorité de rejet moyenne	AF 22	010100	20
	priorité de rejet faible	AF 23	010110	22
		CS 3	011000	24
Classe 3	priorité de rejet élevée	AF 31	011010	26
	priorité de rejet moyenne	AF 32	011100	28
	priorité de rejet faible	AF 33	011110	30
		CS4	100000	32
Classe 4 (la plus moins priorité)	priorité de rejet élevée	AF 41	100010	34
	priorité de rejet moyenne	AF 42	100100	36
	priorité de rejet faible	AF 43	100110	38
		CS 5	101000	40
		EF	101110	46
		CS 6	110000	48
		CS 7	111000	56

Tableau III.5 : Valeurs DSCP attribués par « Ooredoo » aux différentes classes de trafics.

Type de trafic	Priorité	Class QoS	Valeur numérique	
			DSCP	EXP
Signalisation	P0	EF	46	5
Voix	P1	EF	46	5
Data	P3	AF/BE	18	2
Management	P2	AF	38	4

Tableau III.6 : Interprétation des valeurs DSCP en EXP par « Ooredoo ».

Conclusion :

Durant tout ce qu'on a vu dans les premiers chapitres on a décidé de compléter avec cette partie d'analyse théorique un extrait d'un environnement intégralement professionnel. Alors on a essayé de faire un aperçu statistique sur le trafic circulant dans le réseau de l'opérateur« Ooredoo » en quelques intervalles de temps et dans les différents instants de jours ou de semaines.

Cette étude consiste dans un premier temps à auditer l'existant de réseau MPLS de « Ooredoo » en matière des équipements, services offerts ainsi que les performances de réseau. Cette première phase a été suivie par l'analyse et la collecte des informations afin de définir un plan d'actions de la mise en marche de la nouvelle solution qui est bâti sur l'ingénierie de trafic et la qualité de services (INTserv et Difserv).

L'implémentation de MPLS/QoS sous GNS 3

Introduction :

Nous avons réalisé dans ce dernier chapitre une application qui illustre un *Backbone* IP/MPLS intégré dans un domaine DiffServ en appliquant l'optimisation de la qualité de service (*Load Balancing*) à travers la configuration de toutes les principales étapes des chapitres précédents tels que la configuration de base des routeurs CISCO (C 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T7) .La configuration du cœur IP/MPLS et les VPN (VRF,BGP,etc.), aussi le routage explicite (FR- Tunnels, etc.), avec quelques protocoles utilisés au niveau de IP/MPLS (OSPF, LDP, RSVP-TE, etc.), et au final on a aussi pris en compte la configuration de la QoS (DSCP, bande passante…).

La réalisation pratique de ces configurations sur le réseau Ooredoo avec son propre *Backbone* (IP/MPLS), Telle que une petite erreur ou une mal configuration peut engendrer des bugs et des problèmes sur le réseau de l'opérateur, sachant que cette opération nécessite des équipements spéciaux et très puissants qui peuvent gérer des milliers d'appels et de communications simultanément.

Heureusement notre opérateur nous a proposé d'utiliser l'émulateur professionnel « GNS3 » qui est propre à « Cisco » avec l'analyseur « Wireshark » appelé souvent le snifer de paquet. La combinaison de ces deux outils nous a permis de réaliser notre partie pratique en implémentant des configurations réels et professionnelles utilisées par l'opérateur « Ooredoo » sans toucher ni les routeurs ni les liens de l'opérateur, de ce fait on va présenter et installer l'émulateur GNS3 et le snifer de paquet Wireshark , ensuite on présentera l'implémentation de notre configuration sur différents équipements (CE – PE –P) constituant un réseau MPLS/DiffServ afin de mettre en évidence notre étude théorique.

IV.1. Présentation de l'émulateur GNS 3 :

Le GNS3 est un émulateur graphique de réseaux qui nous permet de créer des topologies de réseaux complexes et d'en établir des simulations. Ce logiciel, en lien avec Dynamips (simulateur IOS), est un excellent outil pour l'administration des réseaux CISCO, les laboratoires réseaux ou les personnes désireuses de s'entraîner avant de passer les

certifications CCNA, CCNP, CCIP ou CCIE. De plus, il est possible de s'en servir pour tester les fonctionnalités des IOS Cisco ou de tester les configurations devant être déployées dans le futur sur des routeurs réels. Ce projet est évidemment Open Source et multi-plates-formes. Il est possible de le trouver pour Mac OS X, Windows et évidemment pour une distribution Linux. Sachant que l'utilisateur doit fournir ses propres images IOS pour utiliser GNS3.

IV.1.1. Présentation d'une image IOS :

Avant d'entamer l'architecture souhaitée sous logiciel GNS3, il faut incorporer d'abord une image IOS correspondante à l'équipement désiré qui est unique pour chaque routeur et elle se comporte comme un système d'exploitation (Mac OS, Windows…) dans le cas d'une machine.

IV.1. 2. Présentation de Dynamips :

Dynamips est un simulateur de routeurs Cisco capable de faire fonctionner des images Cisco IOS non modifiées comme si elles s'exécutaient sur de véritables équipements. Le rôle de Dynamips n'est pas de remplacer de véritables routeurs, mais de permettre la réalisation de maquettes complexes avec de vraies versions d'IOS. Contrairement à certains autres produits, il ne s'agit pas d'une simulation de la ligne de commande IOS et de son fonctionnement, mais d'une simulation complète du « hardware ». Dynamips peut être utilisé à des fins de formation, d'expérimentation, aide au diagnostic, validation de configurations, etc.

Dynamips est écrit en langage C, sous licence GPL. Les plateformes hôtes supportées sont de type PC sous Linux, Mac Os X et Windows. Un portage sur d'autres plateformes Unix est également possible.

Les gammes de routeurs émulés dans notre cas sont: Cisco 3725.

IV.1.3. Configuration de paramètre « IDLE PC » :

Parmi les empêchements qu'on a coïncidés après l'obtention des images Cisco IOS est la compatibilité de ces derniers avec les performances de notre machine, pour cette raison on a dû configurer la valeur de « IDLE PC » qui sert à trouver le meilleur couple entre l'image IOS du routeur et les capacités de processeur de la machine utilisée afin de pouvoir émuler le maximums de routeurs sans avoir des problèmes de bugs, il faut tenir en compte que cette valeur de « IDLE-PC » ne fait que l'optimisation de l'utilisation de processeur de la machine

car l'emploi des routeurs CISCO est toujours limité puisque le GNS3 émule des routeurs CISCO réels qui demandent des machines hyper-performantes.

La figure ci-dessous représente l'état du processeur (i5) qui est atteint à 100% après avoir démarrer tous les routeurs de notre topologie.

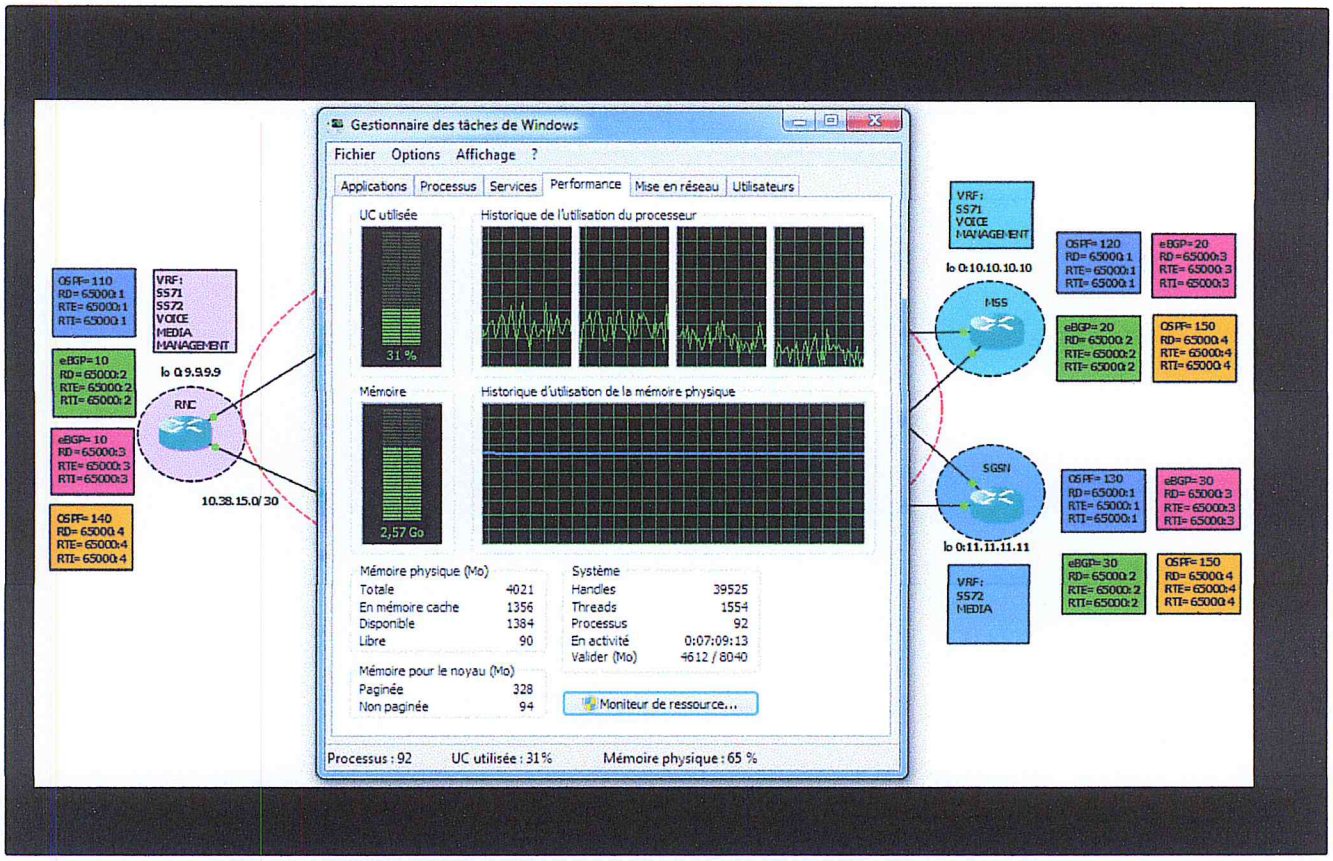


Figure IV.1 : L'état de notre processeur lors l'exploitation de GNS3.

IV.2. Présentation de la maquette d'émulation :

Dans le cadre de notre déploiement, nous avons réalisé la topologie physique ci-dessous constituée de quatre PE, quatre P et trois sites CE interne (RNC, MSS et SGSN).

Pour notre simulation, nous avons choisi la gamme de routeurs C3725 car ils possèdent les avantages suivants :

➢ Les Cisco 3725 sont des routeurs compacts haute performance, conçus pour un déploiement à la périphérie du réseau et dans le centre de données, où les performances et les services sont essentiels pour faire face aux besoins des entreprises, des administrations et des fournisseurs de services.

➢ Les Cisco 3725 de point vu pratique, ils sont capables de supporter plusieurs services compliqués et ils disposent tous des commandes nécessaires pour notre configuration (MPLS, Tunnel,QoS, Fast Reroute, etc.).

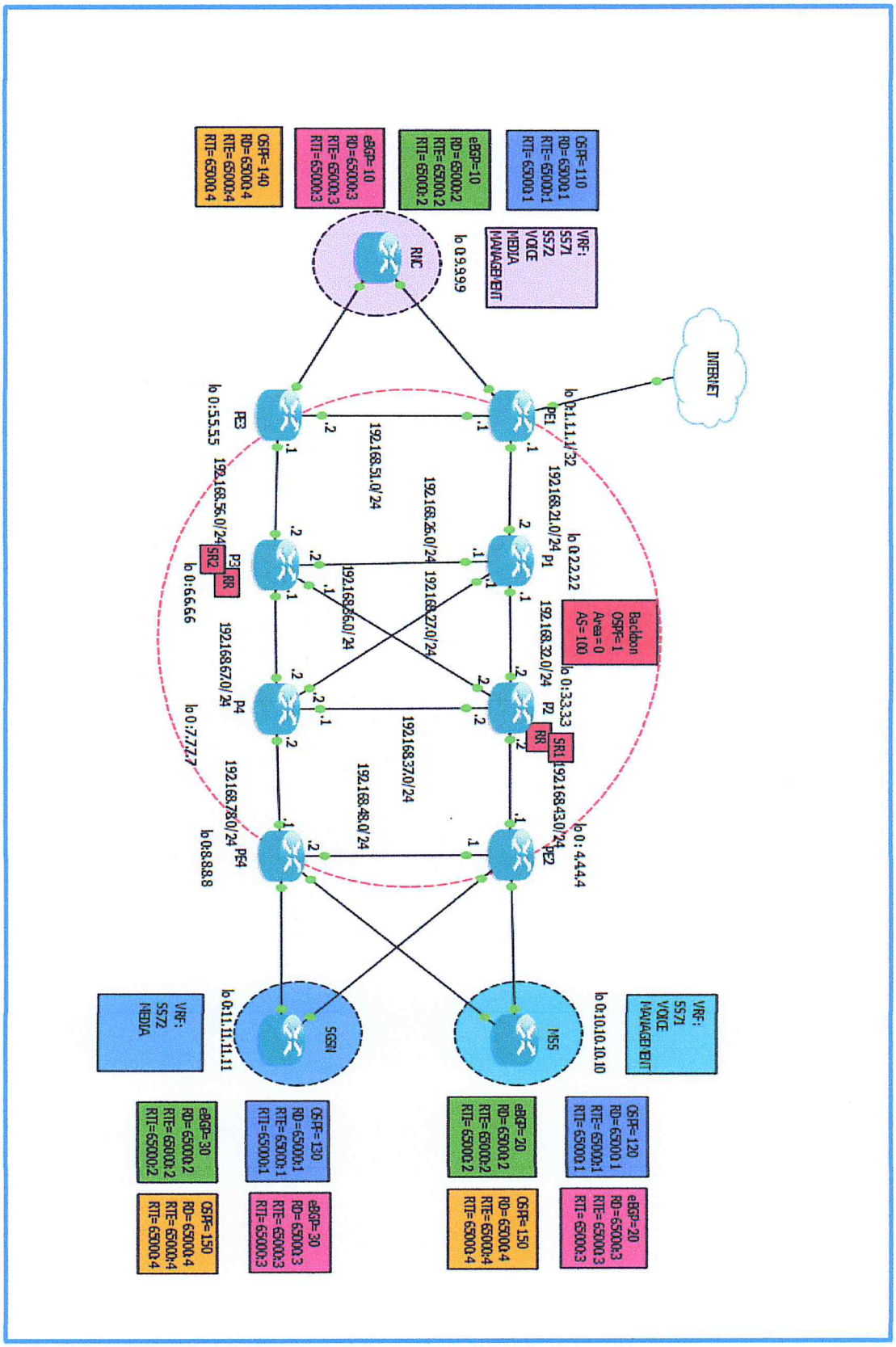


Figure IV.2 : La maquette utilisée pour notre simulation

IV.2.1. Le choix de l'adressage pour notre maquette d'émulation :

Nous avons aléatoirement choisi un plan d'adressage pour notre maquette que nous avons consigné dans le tableau suivant :

Nœuds	Interfaces	Adresses
PE-1	Loopback	1.1.1.1
	F0/0	192.168.51.1
	F1/0	192.168.21.1
	F2/0	192.168.51.1
PE-2	Loopback	4.4.4.4
	F0/0	192.168.10.1
	F0/1	192.168.10.2
	F1/0	192.168.43.1
	F2/0	192.168.48.1
PE-3	Loopback	5.5.5.5
	F0/1	192.168.91.2
	F1/0	192.168.56.1
	F2/0	192.168.51.2
PE-4	Loopback	8.8.8.8
	F0/0	192.168.11.1
	F0/1	192.168.11.2
	F1/0	192.168.78.1
	F2/0	192.168.48.2
P-1	Loopback	8.8.8.8
	F0/0	192.168.11.1
	F0/1	192.168.11.2
	F1/0	192.168.78.1
	F2/0	192.168.48.2
P-2	Loopback	8.8.8.8
	F0/0	192.168.11.1
	F0/1	192.168.11.2
	F1/0	192.168.78.1
	F2/0	192.168.48.2

P-3	Loopback	8.8.8.8
	F0/0	192.168.11.1
	F0/1	192.168.11.2
	F1/0	192.168.78.1
	F2/0	192.168.48.2
P-4	Loopback	7.7.7.7
	F0/0	192.168.37.1
	F0/1	192.168.67.2
	F1/0	192.168.78.2
	F2/0	192.168.27.2
RNC	Loopback	9.9.9.9
	F0/0	192.168.90.2
	F0/1	192.168.91.1
MSS	Loopback	10.10.10.10
	F0/0	192.168.10.2
	F0/1	192.168.11.2
SGSN	Loopback	11.11.11.11
	F0/0	192.168.10.1
	F0/1	192.168.11.1

TableauIV.1 : Plan d'adressage pour notre maquette..

- Le tableau suivant présente les valeurs affectées aux VRF, RD et RT pour les clients RNC ; MSS ; SGSN

Nom-Vrf	Nom-Client	Route-Distinguisher	RT-Import	RT-Export	Protocol
SS71	RNC/MSS	65000 :1	RTI=65000 :1	RTE=65000 :1	Ospf110/120
SS72	RNC/SGNS	65000 :1	RTI=65000 :1	RTE=65000 :1	Ospf110/130
Voice	RNC/MSS	65000 :2	RTI=65000 :2	RTE=65000 :2	Ebgp10/20
Media	RNC/SGNS	65000 :3	RTI=65000 :3	RTE=65000 :3	Ebgp10/30
Mangment	RNC/MSS	65000 :4	RTI=65000 :4	RTE=65000 :4	Ospf140/150

Tableau IV.2 : Plan d'adressage pour les Vrfs

➢ Les tableaux suivants présente les plans d'adressage pour chaque Vrf :

User-plan	Interface-source	@source	vlan	Interface-destination	@destination	vlan	Client-destination
Plan1/contrôl plan-voix	F0/0.150	10.37.15.1/30	150	F0/0.150	192.168.64.1/30	150	MSS
Plan2/contrôl plan-voix	F0/1.160	10.38.15.1/30	160	F0/0.160	192.168.65.1/30	160	SGSN
Plan1/contrôl plan-data	F0/0.151	10.37.15.5/30	151	F0/1.151	192.168.70.1/30	151	MSS
Plan2/contrôl plan-data	F0/1.161	10.38.15.5/30	161	F0/1.161	192.168.71.1/30	161	SGSN

Tableau IV.3 : Plan d'adressage pour Vrf SS7

User-plan	Interface-source	@source	vlan	Interface-destination	@destination	vlan	Client-destination
IUC-UP1	F0/0.250	10.239.149.1/20	250	F0/0.250	192.168.80.1/30	250	MSS
IUC-UP2	F0/1.251	10.239.149.5/30	251	F0/1.251	192.168.81.1/30	251	MSS

Tableau IV.4 : Plan d'adressage pour Vrf voix.

User-plan	Interface-source	@source	vlan	Interface-destination	@destination	vlan	Client-destination
IUPs-UP1	F0/0.252	10.240.149.1/20	252	F0/0.252	10.118.34.1	252	SGSN
IUPs-UP2	F0/1.252	10.240.149.5/30	252	F0/1.252	10.118.34.5	252	SGSN

Tableau IV.5 : Plan d'adressage pour Vrf media.

User-plan	Interface-source	@source	vlan	Interface-destination	@destination	vlan	Client-destination
OM-RNC1	F0/0.105	10.46.6.1/30	105	F0/0.105	10.242.149.1	105	MMS
OM-RNC2	F0/1.106	10.46.6.5/30	106	F0/1.106	10.242.149.5	106	MMS

Tableau IV.6: Plan d'adressage pour Vrf managment.

IV.3. Le plan de configuration de notre maquette :

1) La configuration basique de la maquette.
2) La configuration de « IP/MPLS ».
3) Le déploiement des VPN.
4) L'implémentation de Traffic Engineering.
5) Le déploiement de la QoS.

IV.3.1. La configuration basique de la maquette :

Pour commencer, il faut tout d'abord configurer les adresses IP sur les interfaces des routeurs ainsi les types des liens reliant différents nœuds (Fast-Ethernet) ensuite la configuration des protocoles de routage. Pour cela on a utilisé les commandes suivantes :

Les commandes	Définition
Enable	Le passage en mode privilégie.
Hostname	Attribuer un nom à l'équipement.
configure terminal	Passage en mode configuration du terminal.
Interface **serial** *id*	déclarer l'interface serial.
loopback *id*	déclarer l'interface identifiant le routeur.
ip address *x.x.x.x y.y.y.y*	Affectation d'une adresse IP et un masque à l'interface.
router OSPF *process-id*	Activer le protocole OSPF sur le routeur avec le choix d'un numéro de processus.
Network *x.x.x.x y.y.y.y* **area** *Id*	Déclaration des réseaux participant au processus OSPF.
Exit	Sortir de la configuration

Tableau IV.7: Les commandes de la configuration basique de la maquette

IV.3.2. La configuration de IP/MPLS :

Pour implémenter la technologie MPLS dans un réseau, il faut tout d'abord activer le CEF (*Cisco Expresse Forwarding*). L'activation de MPLS diffère suivant la position du routeur dans le *Backbone*. Dans les deux routeurs « P », nous avons activé MPLS sur toutes les interfaces, tandis que dans les deux autres routeurs « PE », l'activation est seulement faite sur les interfaces reliant directement aux routeurs P. On a utilisé les commandes suivantes :

Les commandes	Définition
Cef	Permet la circulation des trames MPLS.
mpls ip	Activation de MPLS.
mpls label protocol ldp	Déclaration de protocole LDP.

Tableau IV.8 : Les commandes de la configuration IP/MPLS.

IV.3.3. Le déploiement des VPN :

Dans cette partie on doit intervenir au niveau des nœuds de bordures ainsi les nœuds de clients en utilisant les commandes suivantes : [10]

Les commandes	Définition
ip vrf *le-nom-de-vrf*	Création d'une table de routage VRF pour un client.
rd *valeur*	Le RD crée des tables de routage de transmission, Le RD est ajouté au début des en-têtes IPv4 du client pour les convertir en préfixes globalement uniques VPNv4.
route-target export *valeur* **route-target import** *valeur*	Configurer l'importation et l'exportation des stratégies pour les communautés BGP.
ip vrf forwarding *nom-de-vrf*	Associer VRF avec une interface.
redistribute bgp *id-process* **subnets**	Association des adresses réseau aux tables de routage VRF.
Router bgp *id-process*	Activation le protocole de routage BGP.

neighbor *x.x.x.x* **remote-as** *id* **neighbor** *x.x.x.x* **update-source loopback** *id*	Déclaration les adresses Loopback des PE voisins dans le même domaine AS.
address-family vpnv4 **neighbor** *x.x.x.x* **activate** **neighbor** *x.x.x.x* **send-community extended** **exit-address-family**	Configuration de « address-family » BGP VPNv4 par l'activation des familles d'adresse IPv4 et VPNv4.
address-family ipv4 vrf *le-nom-de-vrf* **redistribute ospf** *id-prosses* **vrf** *nom-de-vrf*	Sélectionner de protocole de routage associé à une vrf.
router bgp *id-process* **neighbor** *x.x.x.x* **remote-as** *id* **neighbor** *x.x.x.x* **update-source Loopback** *id*	Configuration les voisins de i-BGP en utilisant loopback comme adresse source.
router ospf *id-process* **vrf** *nom-de-vrf* **router-id** *x.x.x.x*	Associer un protocole de routage à une VRF.
redistribute bgp ***numéro-de-AS*** **subnets**	Redistribuer les routes BGP (incluant celles de sous réseau) en OSPF.

Tableau IV.9 : Les commandes de la configuration VPN.

IV.3.4. L'implémentation de *Traffic Engineering* :

Une fois les configurations de base sont bien effectuées, on est passé aux choses sérieuses, qui consistent à faire la configuration de l'ingénierie de trafic via les commandes suivantes : [17]

Les commandes	Définition
ip rsvp bandwith *id*	Configuration d'une valeur en Kbps pour la bande passante des RSVP qui seront utilisés pour la signalisation et l'allocation des ressources pour TE
interface tunnel *id*	Créer une interface pour le tunnel utilisé.
ip unnumbered loopback *id*	Associer l'adresse loopback à l'interface de tunnel.
tunnel destination x.x.x.x	Attribuer l'adresse de destination de tunnel.

tunnel mode mpls traffic-eng	Déclaration de l'opération de l'ingénierie de trafic.
tunnel mpls traffic-eng autoroute announce	Annoncer l'interface tunnel dans la table de routage OSPF.
tunnel mpls traffic-eng priority *id*	Annoncer la priorité de tunnel.
tunnel mpls traffic-eng bandwidth *id*	Annoncer la bande passante en Kbps utilisée par le tunnel.
tunnel mpls traffic-eng path-option *id* **explicit name** *nom*	Configuration des chemins explicites avec les adresses IP des routeurs *next-hop* présents sur un chemin LSP.
next-address *x.x.x.x*	Déclarer les interfaces d'entrée des routeurs où il doit passer le tunnel.
tunnel mpls traffic-eng record-route	Il permet d'enregistrer la route empruntée par le trafic.
tunnel mpls traffic-eng fast-reroute	Annoncer à l'interface tunnel qu'il y'aura un re-routage dans le cas échant.
mpls traffic-eng auto-tunnel backup nhop-only	Déclaration qu'il y'aura un re-routage de chemin au niveau d'un P.
mpls traffic-eng backup-path Tunnel *id*	Elle se déclare au niveau des « P » afin d'ouvrir des chemins secondaire.

Tableau IV.10 : Les commandes de la configuration TE.

IV.3.5. Le déploiement de la QoS/MPLS :

Pour implémenter la QoS, nous avons utilisé le modèle MQC (*Modular QoS CLI*) qui se subdivise en réalité en trois grandes étapes à savoir :

- La classification du trafic basée sur les critères définis par l'utilisateur.
- La configuration des stratégies de la QoS pour chacune des catégories définies.
- L'association des stratégies ou d'une stratégie de la QoS à une interface.

Pour cela nous avons utilisé quelques commandes citées au-dessous :

Les commandes	Définition
1- La classification de trafic	
class-map *nom-de-la-classe*	Créer une classe de trafic correspondant aux critères donnés par l'utilisateur et même pour spécifier le nom de la classe.
match-any	Elle est utilisée lorsqu'un seul critère d'une classe doit être rempli pour un paquet.
match-all	Elle est utilisée lorsque tous les critères d'une classe doivent être remplis pour un paquet, pour faire la correspondance avec la classe du trafic spécifié.
match cos *valeur-de-cos*	Pour configurer les critères pour une classe basée sur le marquage CoS de la couche 2.
match input-interface *nom-de-l'interface*	Elle permet de configurer les critères pour une classe basée sur une interface d'entrée spécifiée.
match ip dscp *valeur-de-dscp*	Elle permet de configurer les critères pour une classe basée sur la valeur du champ DSCP, on peut combiner dans une seule déclaration match jusqu'à huit valeurs DSCP, ces valeurs qui varient entre 0 et 63.
match ip precedence *valeur-de-ip precedence*	Elle permet de configurer les critères pour une classe basée sur la valeur IP *precedence*, on peut combiner dans une seule déclaration match jusqu'à quatre valeurs. La valeur d'IP *precedence* varie entre 0 et 7.
match protocol *protocole*	Elle permet de configurer les critères pour une classe basée sur un protocole spécifié.
match source-address mac *adresse*	Elle permet de configurer les critères pour une classe basée sur l'adresse MAC source.
2- La configuration des stratégies	
policy-map *nom de la stratégie*	Elle permet de configurer une stratégie avant de l'assigner à une classe de trafic particulière avec la commande **class.**
bandwidth {*bande passante-kbps* | **percent** *pourcentage*}	L'indication de la bande passante minimale à garantir pour une classe de trafic. Ce minimum de bande passante peut être spécifié en kbps ou en pourcentage.
fair-queue *nombre-de-files-d'attente*	Précision sur le nombre de files d'attente qui va être réservées à la classe.
police *bps burst-normal burst-*	Indication les limitations de la bande passante maximale

max **conform-action** *action* **exceed-action** *action* **violate-action** *action*	associée par une action «transmet» ou «drop» les paquets.
queue-limit *nombre-de-paquets*	Précision sur le nombre maximum de paquets dans la file d'attente.
random-detect	Activation de l'algorithme de suppression de paquets WRED pour une classe de trafic qui a une bande passante garantie.
set cos *valeur-de-cos*	La spécification d'une valeur ou des valeurs de CoS à associer avec le paquet, le nombre est entre 0 et 7.
set ip dscp *valeur-de-ip-dscp*	Spécifications de l'IP/DSCP des paquets dans la classe du trafic.
Set mpls experimental topmost *id*	Spécifications de « Exp » des paquets dans la classe du trafic.
3- Assignation des stratégies crées aux différentes interfaces	
service-policy input/output *nom-de-la stratégie.*	Attachement de la stratégie spécifiée à l'interface choisie.

Tableau IV.11 : Les commandes de la configuration QoS/Mpls.

IV.4. L'interprétation des résultats :

Cette partie consiste à interpréter les résultats obtenus par le test de notre réseau avec l'utilisation de GNS 3, WireShark.

➢ **Remarque**:

- On a exposé les résultats dans un seul routeur pour chaque type (CE, PE et P) car d'un côté, la configuration est pratiquement la même pour les autres routeurs et d'un autre coté on est censé respecter les normes de la thèse (limitation de nombre de pages).
- Notre maquette réalisée est homogène, tous les routeurs implémentés sont de modèle Cisco 3725 et les interfaces physique sont de type « Fast Ethernet».

IV.4.1. Le routeur PE1 :

Afin de vérifier l'activation des adresses IP sur les différentes interfaces physique « Fast Ethernet » et logique « looback» sur le routeur PE1 on a exécuté la commande « Show ip interface brief » :

```
PE1#sh ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  up                    up
FastEthernet0/0.105        10.46.6.2       YES NVRAM  up                    up
FastEthernet0/0.150        10.37.15.2      YES NVRAM  up                    up
FastEthernet0/0.151        10.37.15.6      YES NVRAM  up                    up
FastEthernet0/0.250        10.239.149.2    YES NVRAM  up                    up
FastEthernet0/0.252        10.240.149.2    YES NVRAM  up                    up
FastEthernet0/1            unassigned      YES NVRAM  up                    up
FastEthernet1/0            192.168.21.3    YES NVRAM  up                    up
FastEthernet2/0            192.168.51.1    YES NVRAM  up                    up
Loopback0                  1.1.1.1         YES NVRAM  up                    up
Tunnel1                    1.1.1.1         YES TFTP   up                    down
```

Ensuite on a dû vérifier le fonctionnement de protocole de routage OSPF avec ses routeurs voisins par la consultation des tables de routage via les commandes « show ip ospf neighbors » et « show ip ospf database » :

```
PE1#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
5.5.5.5           1   FULL/DR         00:00:35    192.168.51.2    FastEthernet2/0
2.2.2.2           1   FULL/DR         00:00:30    192.168.21.2    FastEthernet1/0
9.9.9.9           1   FULL/DR         00:00:34    10.37.15.5      FastEthernet0/0.151
9.9.9.9           1   FULL/DR         00:00:34    10.37.15.1      FastEthernet0/0.150
10.240.149.5      1   FULL/DR         00:00:34    10.46.6.1       FastEthernet0/0.105
```

➢ Pour le protocole OSPF 1 de Backbone :

```
PE1#sh ip ospf 1 database

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         230         0x80000006 0x0018AE 3
2.2.2.2         2.2.2.2         209         0x80000005 0x002F68 5
3.3.3.3         3.3.3.3         228         0x80000007 0x00A48C 5
4.4.4.4         4.4.4.4         189         0x80000003 0x0096E6 3
5.5.5.5         5.5.5.5         223         0x80000003 0x00F25E 3
6.6.6.6         6.6.6.6         215         0x80000005 0x0015B1 5
7.7.7.7         7.7.7.7         215         0x80000005 0x004D3C 5
8.8.8.8         8.8.8.8         189         0x80000003 0x0007FE 3
```

➢ Pour le protocole OSPF 110 de Client RNC :

```
PE1#sh ip ospf 110 database

            OSPF Router with ID (10.37.15.6) (Process ID 110)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
9.9.9.9         9.9.9.9         419         0x80000003 0x00EB61 5
10.37.15.6      10.37.15.6      460         0x80000007 0x006E08 2
10.38.15.6      10.38.15.6      418         0x80000002 0x00A4D0 2

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.37.15.2      10.37.15.6      464         0x80000001 0x0079D6
10.37.15.6      10.37.15.6      464         0x80000001 0x0051FA
10.38.15.2      10.38.15.6      419         0x80000001 0x006DDF
10.38.15.6      10.38.15.6      419         0x80000001 0x004504

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.10.10.10     10.37.15.6      397         0x80000001 0x00F90E 3489661028
11.11.11.11     10.37.15.6      397         0x80000001 0x00CB38 3489661028
192.168.64.0    10.37.15.6      397         0x80000001 0x00DCB6 3489661028
192.168.65.0    10.37.15.6      401         0x80000001 0x00D1C0 3489661028
192.168.70.0    10.37.15.6      402         0x80000001 0x009AF2 3489661028
192.168.71.0    10.37.15.6      403         0x80000001 0x008FFC 3489661028
```

➢ En suite on a passé à la vérification de la configuration IP/MPLS, tout d'abord il fallait vérifier le MPLS, s'il est opérationnel sur l'interface relié avec le backbone, ainsi si le protocole LDP est activé, en utilisant la commande « Show mpls interfaces » :

```
PE1#sh mpls interfaces
Interface              IP             Tunnel   Operational
FastEthernet1/0        Yes (ldp)      No       Yes
FastEthernet2/0        Yes (ldp)      No       Yes
```

➢ Puis nous sommes allés voir si les labels ont été distribués dans notre réseau, en ayant exécuté la commande « show *mpls forwording table»* sur notre Ingress LER .

```
PE1#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Aggregate   10.37.15.0/30[V] 0
17     Aggregate   10.37.15.4/30[V] 0
18     17          6.6.6.6/32       8356       Fa2/0       192.168.51.2
19     19          3.3.3.3/32       5152       Fa2/0       192.168.51.2
20     Pop tag     2.2.2.2/32       7655       Fa1/0       192.168.21.2
21     28          192.168.67.0/24  0          Fa2/0       192.168.51.2
       19          192.168.67.0/24  0          Fa1/0       192.168.21.2
22     Pop tag     192.168.56.0/24  0          Fa2/0       192.168.51.2
23     29          192.168.43.0/24  0          Fa2/0       192.168.51.2
24     23          192.168.37.0/24  0          Fa1/0       192.168.21.2
25     Pop tag     192.168.27.0/24  0          Fa1/0       192.168.21.2
26     Pop tag     192.168.26.0/24  0          Fa1/0       192.168.21.2
27     30          192.168.36.0/24  0          Fa2/0       192.168.51.2
28     Pop tag     192.168.32.0/24  0          Fa1/0       192.168.21.2
29     Untagged    9.9.9.9/32[V]    0          Fa0/0.151   10.37.15.5
       Untagged    9.9.9.9/32[V]    0          Fa0/0.150   10.37.15.1
30     Untagged    10.38.15.4/30[V] 0          Fa0/0.151   10.37.15.5
       Untagged    10.38.15.4/30[V] 0          Fa0/0.150   10.37.15.1
31     Untagged    10.38.15.0/30[V] 0          Fa0/0.151   10.37.15.5
       Untagged    10.38.15.0/30[V] 0          Fa0/0.150   10.37.15.1
32     25          7.7.7.7/32       5132       Fa1/0       192.168.21.2
33     29          192.168.78.0/24  0          Fa1/0       192.168.21.2
34     Pop tag     5.5.5.5/32       54         Fa2/0       192.168.51.2
35     30          8.8.8.8/32       4047       Fa1/0       192.168.21.2
36     18          4.4.4.4/32       0          Fa2/0       192.168.51.2
       27          4.4.4.4/32       0          Fa1/0       192.168.21.2
37     28          192.168.48.0/24  0          Fa1/0       192.168.21.2
```

- Puis on a passe au déploiement des VPN pendant lesquels nous avons créé au niveau des routeurs PE des VRF pour chaque client où on peut vérifier l'attachement des clients au routeur PE-A par la commande « Show ip vrf » :

```
PE1#sh ip vrf
  Name                             Default RD          Interfaces
  Management                       65000:4             Fa0/0.105
  Media                            65000:3             Fa0/0.252
  SS7                              65000:1             Fa0/0.150
                                                       Fa0/0.151
  Voice                            65000:2             Fa0/0.250
```

- Et via la commande « Show ip route vrf SS7 » on a pu voir la table de routage VRF par le client-SS7 par exemple :

```
PE1#sh ip route vrf SS7

Routing Table: SS7
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     192.168.64.0/30 is subnetted, 1 subnets
B       192.168.64.0 [200/0] via 4.4.4.4, 00:26:27
     9.0.0.0/32 is subnetted, 1 subnets
O       9.9.9.9 [110/11] via 10.37.15.5, 00:27:35, FastEthernet0/0.151
                [110/11] via 10.37.15.1, 00:27:25, FastEthernet0/0.150
```

- A la suite on a inspecté le protocole de routage I-BGP. Pour le vérifier on a utilisé la commande suivante : « show ip bgp vpnv4 all summary » :

```
PE1#show ip bgp vpnv4 all summary
BGP router identifier 1.1.1.1, local AS number 100
BGP table version is 34, main routing table version 34
11 network entries using 1540 bytes of memory
17 path entries using 1156 bytes of memory
10/9 BGP path/bestpath attribute entries using 1240 bytes of memory
4 BGP rrinfo entries using 96 bytes of memory
5 BGP extended community entries using 200 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 1) using 32 bytes of memory
BGP using 4264 total bytes of memory
BGP activity 11/0 prefixes, 17/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4   100      55      38       34    0    0 00:30:57        6
6.6.6.6         4   100      46      37       34    0    0 00:30:43        6
```

- Et a fin de vérifier l'existence de SS7 dans la base de données de routeur PE1, on a affecté un « Ping » vers la VRF de SS7 :

```
PE1#ping vrf SS7 9.9.9.9

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 9.9.9.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/32/56 ms
```

➢ Configurations des interfaces :

Après on a consulté les résultats concernant le déploiement de TE où les captures d'écran ci-dessous nous présente le résumé des tunnels MPLS-TE sur le routeur Ingress PE1 que nous avons Simulés.

Nous allons tout d'abord commencer par la configuration des interfaces. D'où chaque interface qui fait partie de tunnel « **121** » , « **122** » doit permettre le Trafic Engineering. Cela consiste à déclarer pour toutes ces interfaces les paramètres suivants :

- Activation de Trafic Engineering sur l'interface.
- Bande passante réservable par le TE.
- Affinité de l'interface dans la topologie TE (valeur en hexadécimal).

La figure suivante représente les configurations nécessaires à chaque routeur qui participe au trafic engineering avec les interfaces « **Fa1/0** » et « **Fa2/0** ».

```
PE1(config)#mpls traffic-eng tunnels
PE1(config)#
PE1(config)#router ospf 1
PE1(config-router)#mpls traffic-eng router-id loopback 0
PE1(config-router)#mpls traffic-eng area 0
PE1(config-router)#exit
PE1(config)#
PE1(config)#
PE1(config)#int f1/0
PE1(config-if)#mpls traffic-eng tunnels
PE1(config-if)#mpls traffic-eng attribute-flags 0x1
PE1(config-if)#ip rsvp bandwidth 1024
PE1(config-if)#exit
PE1(config)#
PE1(config)#int f2/0
PE1(config-if)#mpls traffic-eng tunnels
PE1(config-if)#mpls traffic-eng attribute-flags 0x1
PE1(config-if)#ip rsvp bandwidth 1024
PE1(config-if)#
PE1(config-if)#exit
```

➢ Création d'un tunnel dynamique :

La configuration détaillée ci-dessous est celle de tunnel « **121**» et de tunnel « **122** » crée sur « **PE_1** » avec les paramètres suivants :

- **Destination du tunnel:** C'est l'adresse Loopback du routeur cible de sortie du tunnel « **PE_2** »

- **Priorités :** L'établissement de priorité des tunnels est comme suite priorité **(5)** pour le tunnel « **121**» et priorité **(1)** pour le tunnel « **122**»).
- **Bande passante :** TE requise pour l'établissement du tunnel « **121**» est de **(500 Kbps)** et **(800 Kbps)** pour celle de tunnel « **122**» .
- **Affinités :** « **0x1** » avec un mask « **0xF** » permettent de définir les liens à inclure, à préférer et à exclure du chemin pour les deux tunnels.
- **Path-Option :** Elle sert à définir l'option dynamique pour le chemin qui est définis à « **1** » dans notre cas.

La figure ci-dessus présente la configuration de tunnel principale « **121**» entre « **PE_1** »et « **PE_2** ».

```
PE1(config)#interface Tunnel121
PE1(config-if)# ip unnumbered Loopback0
PE1(config-if)# mpls ip
PE1(config-if)# tunnel mode mpls traffic-eng
PE1(config-if)# tunnel destination 4.4.4.4
PE1(config-if)# tunnel mpls traffic-eng autoroute announce
PE1(config-if)# tunnel mpls traffic-eng priority 5 5
PE1(config-if)# tunnel mpls traffic-eng bandwidt
*Mar  1 00:20:57.715: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel121, changed state to downh  500
PE1(config-if)# tunnel mpls traffic-eng affinity 0x1 mask 0xF
PE1(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
PE1(config-if)#
*Mar  1 00:24:07.075: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel121, changed state to up
```

La figure ci-dessus présente la configuration de tunnel back-up « **122**» entre « **PE_1** »et « **PE_2** ».

```
PE1(config)#interface Tunnel122
PE1(config-if)#
*Mar  1 00:30:34.755: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel122, changed state to down
PE1(config-if)#ip unnumbered Loopback0
PE1(config-if)#mpls ip
PE1(config-if)#
PE1(config-if)#tunnel mode mpls traffic-eng
PE1(config-if)# tunnel destination 4.4.4.4
PE1(config-if)#tunnel mpls traffic-eng autoroute announce
PE1(config-if)# tunnel mpls traffic-eng priority 1 1
PE1(config-if)#tunnel mpls traffic-eng bandwidt 800
PE1(config-if)#tunnel mpls traffic-eng affinity 0x1 mask 0xF
PE1(config-if)#tunnel mpls traffic-eng path-option 1 dynamic
PE1(config-if)#
PE1(config-if)#exit
*Mar  1 00:31:53.139: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel122, changed state to up
```

➢ Statut et informations sur le tunnel :

Le résultat de la commande (show mpls traffic-eng tunnel) permet d'obtenir les informations essentielles suivantes :

- **Status** : Signal de l'état administratif et opérationnel du tunnel TE.
- **Path option** : Chemin sélectionné pour le tunnel et son poids.
- **Config Paramètres** : Paramètres généraux du tunnel.
- **OutLabel** : Label de commutation MPLS associé au Tunnel.
- **RSVP Signalling Info** : Informations sur la signalisation RSVP du chemin du tunnel.

La figures ci-dessous représente le chemin pour le tunnel **121»**

```
Name: PE1_t121                          (Tunnel121) Destination: 4.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type dynamic (Basis for Setup, path weight 12)

  Config Parameters:
    Bandwidth: 500      kbps (Global)  Priority: 5  5   Affinity: 0x1/0xF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 500      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : FastEthernet1/0, 31
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 121, Tun_Instance 299
    RSVP Path Info:
      My Address: 192.168.21.3
      Explicit Route: 192.168.21.2 192.168.32.1 192.168.32.2 192.168.43.2
                      192.168.43.3 4.4.4.4
```

La figures ci-dessous représente le chemin pour le tunnel « **122**» .

```
Name: PE1_t122                          (Tunnel122) Destination: 4.4.4.4
  Status:
    Admin: up         Oper: up     Path: valid       Signalling: connected

    path option 1, type dynamic (Basis for Setup, path weight 14)

  Config Parameters:
    Bandwidth: 800      kbps (Global)  Priority: 1  1   Affinity: 0x1/0xF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 800      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : FastEthernet2/0, 63
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 122, Tun_Instance 11
    RSVP Path Info:
      My Address: 192.168.51.1
      Explicit Route: 192.168.51.2 192.168.56.1 192.168.56.2 192.168.67.1
                      192.168.67.2 192.168.78.2 192.168.78.1 192.168.48.2
                      192.168.48.1 4.4.4.4
```

➢ Les protocoles utilisés par les interfaces :

Pour savoir quelles sont les interfaces utilisées par le protocole « **RSVP** ». nous allons exécuter la commande suivante « **show ip rsvp interface** ».

```
PE1#sh ip rsvp int
interface    allocated  i/f max  flow max sub max
Fa1/0        500K       1024K    1024K    0
Fa2/0        800K       1024K    1024K    0
```

Dans l'étape qui suit nous allons éteindre l'une des interfaces « **Fa1/0** » avec la commande « **no shutdown** » .

```
PE1(config)#int f1/0
PE1(config-if)#sh
PE1(config-if)#shutdown
```

Une fois l'interface « **Fa1/0** » est désactivée , il aura que l'interface « **Fa2/0** » qui sera liée au protocole « **RSVP** » comme la démontre la figure au-dessous .

```
PE1(config-if)#do sh ip rsvp int
interface    allocated  i/f max  flow max sub max
Fa1/0        0          1024K    1024K    0
Fa2/0        800K       1024K    1024K    0
```

➢ Traçage du chemin des tunnels :

Après avoir désactiver l'interface « **Fa1/0** » ,le chemin vers la destination PE-2 sera pris par l'interface « **Fa1/0** » qui correspond au tunnel « **122** » pour confirmer ça on exécute la commande suivante « **traceroute 4.4.4.4** ».

```
PE1#traceroute 4.4.4.4

Type escape sequence to abort.
Tracing the route to 4.4.4.4

  1 192.168.51.2 [MPLS: Label 40 Exp 0] 828 msec 844 msec 660 msec
  2 192.168.56.2 [MPLS: Label 31 Exp 0] 1016 msec 800 msec 1016 msec
  3 192.168.67.2 [MPLS: Label 31 Exp 0] 808 msec 732 msec 924 msec
  4 192.168.78.1 [MPLS: Label 38 Exp 0] 792 msec 728 msec 880 msec
  5 192.168.48.1 648 msec 952 msec 1424 msec
```

➢ Utilisation de TE avec des MPLS/VPN:

Pour affecter les tunnels aux clients nous avons procédé aux étapes suivantes :

- Créer une interface Loopback pour associer le tunnel au VRF approprié.
- Changer le comportement de routage : forcer le protocole BGP de prendre le chemin du tunnel et non pas le meilleur chemin du protocole IGP.
- Créer une route statique vers la destination à travers le tunnel.

➢ Aprés l'affectation des VRFs . On effectue un traceroute de Client « **RNC** » vers le Client «**MSS** » on exécutant la commande suivante «**traceroute 192.168.64.1 source 10.37.15.1** »

```
RNC#traceroute 192.168.64.1 source 10.37.15.1

Type escape sequence to abort.
Tracing the route to 192.168.64.1

  1 10.37.15.2 228 msec 264 msec 176 msec
  2 192.168.21.2 [MPLS: Labels 25/33 Exp 0] 768 msec 804 msec 1204 msec
  3 192.168.32.2 [MPLS: Labels 23/33 Exp 0] 868 msec 1040 msec 1080 msec
  4 192.168.64.2 616 msec 872 msec 800 msec
  5 192.168.64.1 952 msec 1172 msec 884 msec
```

➢ Dans l'étape suivante nous allons éteindre l'interface « **Fa1/0** » dans le router **PE-1** à travers la commande « **no shutdown** ».

```
PE1(config)#int f1/0
PE1(config-if)#sh
PE1(config-if)#shutdown
```

- Une fois l'interface « **Fa1/0** » est désactivée. Nous allons effectuer un test de ping pour voir le chemin pris après la coupure à travers la commande « **ping 192.168.64.1 source 10.37.15.1 repeat 300** ».

```
RNC#ping 192.168.64.1 source 10.37.15.1 repeat 300

Type escape sequence to abort.
Sending 300, 100-byte ICMP Echos to 192.168.64.1, timeout is 2 seconds:
Packet sent with a source address of 10.37.15.1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

- Afin de confirmer que le chemin pris après la coupure est le chemin qui correspond au tunnel qui correspond « **122** » , nous allons effectuer un autre traceroute vers le router de destination **PE-2.**

```
RNC#traceroute 192.168.64.1 source 10.37.15.1

Type escape sequence to abort.
Tracing the route to 192.168.64.1

  1 10.37.15.2 272 msec 356 msec 176 msec
  2 192.168.51.2 [MPLS: Labels 20/33 Exp 0] 1204 msec 1356 msec 800 msec
  3 192.168.56.2 [MPLS: Labels 25/33 Exp 0] 940 msec 1144 msec 956 msec
  4 192.168.67.2 [MPLS: Labels 25/33 Exp 0] 1028 msec 880 msec 1136 msec
  5 192.168.78.1 [MPLS: Labels 21/33 Exp 0] 916 msec 884 msec 1208 msec
  6 192.168.64.2 980 msec 972 msec 1036 msec
  7 192.168.64.1 1136 msec 1328 msec 1152 msec
RNC#
```

- **Configuration de la « QoS » :**

Pour suivre le déploiement de la QoS nous avons créé quelques classes de services on se basant sur l'identification de quelques types de flux selon leurs protocoles . Une fois ces classes créées avec tous leurs paramètres, nous les avons associés aux interfaces concernés dans le but d'optimiser le transport au niveau de notre cœur de réseau. Que nous avons vérifié à travers les deux commandes « show policy-map » et « show class-map » :

```
PE1#sh class-map
 Class Map match-all voix (id 1)
   Match access-group  30

 Class Map match-all data (id 2)
   Match access-group  40

 Class Map match-all management (id 3)
   Match access-group  50

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all signalisation (id 4)
   Match access-group  10
   Match access-group  20
```

```
PE1#sh policy-map
  Policy Map policy
    Class signalisation
      set mpls experimental topmost 5
    Class voix
      set mpls experimental topmost 5
    Class data
      set mpls experimental topmost 2
    Class management
      set mpls experimental topmost 4
```

- Dans cette étape nous allons effectuer un ping avec la valeur de « **DSCP =46** », qui prends la valeur **184** en « **TOS** » comme on peut le voir dans la figure au-dessous :

```
RNC#ping
Protocol [ip]:
Target IP address: 192.168.64.1
Repeat count [5]: 500
Datagram size [100]: 100
Timeout in seconds [2]: 2
Extended commands [n]: yes
Source address or interface: 10.37.15.1
Type of service [0]: 184
Set DF bit in IP header? [no]: yes
Validate reply data? [no]: yes
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 500, 100-byte ICMP Echos to 192.168.64.1, timeout is 2 seconds:
Packet sent with a source address of 10.37.15.1
Packet sent with the DF bit set
Reply data will be validated
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!
```

- Et enfin pour terminer ,nous allons voir que la valeur par défaut du champ EXP qui est égale à zéro, après la configuration elle prend la valeur identique au type de trafic (5 pour la classe ICMP) comme le montre la figure suivante :

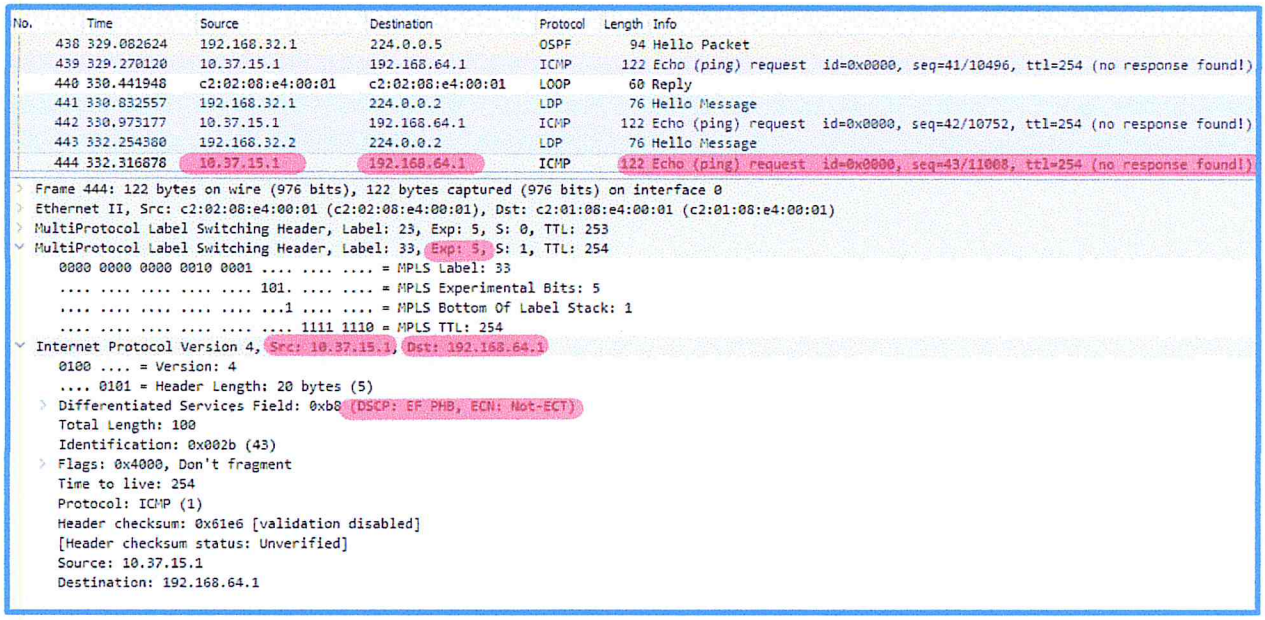


Figure IV.3 : La capture de l'apparition de champ « EXP » sous Wireshrark.

IV.4.2. Le router P2 (Route-Reflector 1) :

Afin de vérifier l'activation des adresses IP sur les différentes interfaces physique « Fast Ethernet » et logique « looback» sur le routeur P2 on a exécuté la commande « Show ip interface brief » :

```
P2#Show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.37.2    YES NVRAM  up                    up
FastEthernet0/1            192.168.32.2    YES NVRAM  up                    up
FastEthernet1/0            192.168.43.2    YES NVRAM  up                    up
FastEthernet2/0            192.168.36.2    YES NVRAM  up                    up
Loopback0                  3.3.3.3         YES NVRAM  up                    up
```

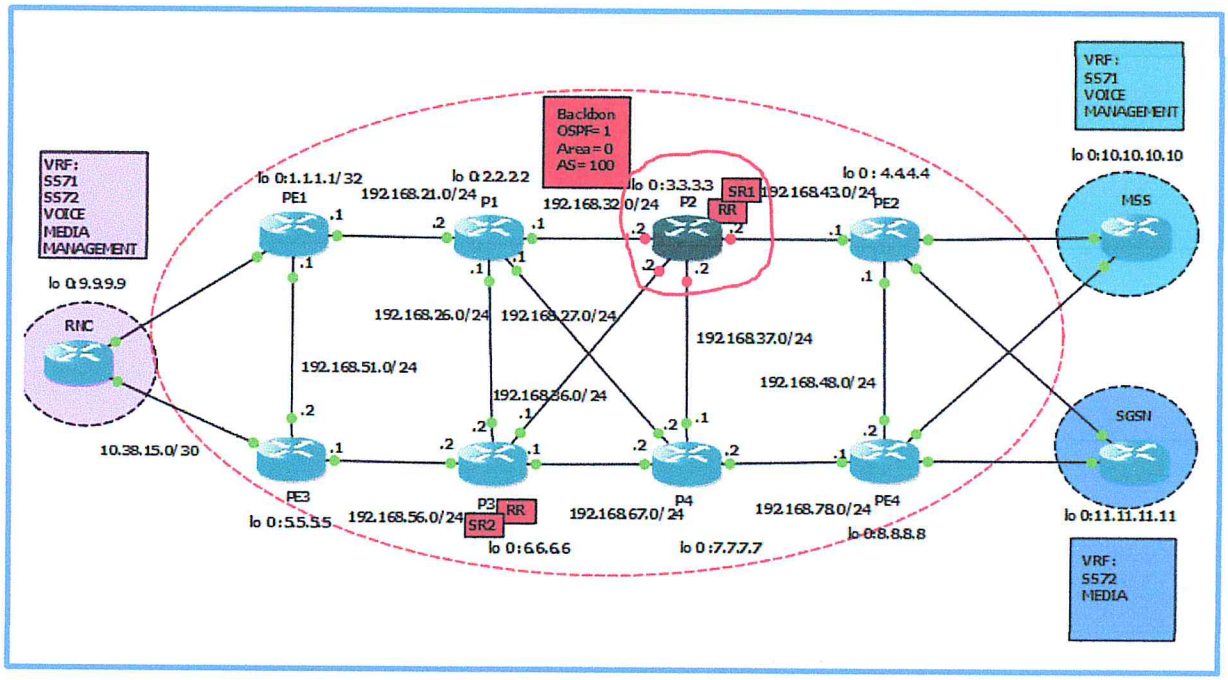
A la suite on a inspecté le protocole de routage I-BGP. Pour le vérifier on a utilisé la commande suivante : « show ip bgp vpnv4 all summary » :

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4   100      48      65       25    0    0 00:41:07        5
4.4.4.4         4   100      56      64       25    0    0 00:40:45        6
5.5.5.5         4   100      50      65       25    0    0 00:41:00        5
6.6.6.6         4   100      56      65       25    0    0 00:41:36       16
8.8.8.8         4   100      56      64       25    0    0 00:40:57        6
```

Par la suite on a vérifié l'acheminement des « **Route-Reflector /P2** » au niveau de LSR à travers la commande « Show ip bgp neighbors » :

```
BGP neighbor is 1.1.1.1,  remote AS 100, internal link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 00:43:25
  Last read 00:00:28, last write 00:00:30, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:               20          3
    Keepalives:            46         46
    Route Refresh:          0          0
    Total:                 67         50
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
  Output queue size: 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                               Sent       Rcvd
  Prefix activity:             ----       ----
    Prefixes Current:             0          0
    Prefixes Total:               0          0
    Implicit Withdraw:            0          0
    Explicit Withdraw:            0          0
    Used as bestpath:           n/a          0
    Used as multipath:          n/a          0

                                 Outbound    Inbound
  Local Policy Denied Prefixes:  --------    -------
    Total:                              0          0
  Number of NLRIs in the update sent: max 0, min 0

 For address family: VPNv4 Unicast
  BGP table version 25, neighbor version 25/0
  Output queue size: 0
  Index 1, Offset 0, Mask 0x2
  Route-Reflector Client
```

Et pour vérifier vraiment que « **Route-Reflector Backup /P3** »fonctionne, on fait éteindre le premier router P2.



Afin de vérifier le fonctionnement de Route-Reflector Backup , on a affecté un « Ping » de client RNC vers le client MSS:

```
RNC#ping 10.10.10.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 120/181/260 ms
```

IV.4.3. Le routeur client RNC:

Au début on a examiné les voisins de ce routeur par la commande « Show cdp neighbors » où on a trouver deux routeurs qui sont des LER (PE1/PE3).

```
RNC#Show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
PE1              Fas 0/0.150         174          R S I     3725      Fas 0/0.150
PE3              Fas 0/1.160         151          R S I     3725      Fas 0/1.160
```

On a pu aussi consulter la table de routage de routeur par la commande « Show ip route »:

```
Gateway of last resort is not set

     192.168.64.0/30 is subnetted, 1 subnets
B       192.168.64.0 [20/0] via 10.239.149.2, 00:46:24
     192.168.81.0/30 is subnetted, 1 subnets
B       192.168.81.0 [20/0] via 10.239.149.2, 00:46:24
     9.0.0.0/32 is subnetted, 1 subnets
C       9.9.9.9 is directly connected, Loopback0
     192.168.80.0/30 is subnetted, 1 subnets
B       192.168.80.0 [20/0] via 10.239.149.2, 00:46:24
     192.168.65.0/30 is subnetted, 1 subnets
B       192.168.65.0 [20/0] via 10.240.149.2, 00:46:24
     10.0.0.0/8 is variably subnetted, 16 subnets, 2 masks
B       10.10.10.10/32 [20/0] via 10.239.149.2, 00:46:26
C       10.37.15.0/30 is directly connected, FastEthernet0/0.150
C       10.46.6.0/30 is directly connected, FastEthernet0/0.105
C       10.38.15.0/30 is directly connected, FastEthernet0/1.160
C       10.37.15.4/30 is directly connected, FastEthernet0/0.151
C       10.46.6.4/30 is directly connected, FastEthernet0/1.106
C       10.38.15.4/30 is directly connected, FastEthernet0/1.161
B       10.239.159.65/32 [20/0] via 10.239.149.6, 00:46:38
B       10.118.34.4/30 [20/0] via 10.240.149.2, 00:46:29
B       10.118.34.0/30 [20/0] via 10.240.149.2, 00:46:29
B       10.242.149.4/30 [20/0] via 10.239.149.2, 00:46:30
C       10.240.149.4/30 is directly connected, FastEthernet0/1.252
B       10.242.149.0/30 [20/0] via 10.239.149.2, 00:46:30
C       10.240.149.0/30 is directly connected, FastEthernet0/0.252
C       10.239.149.0/30 is directly connected, FastEthernet0/0.250
C       10.239.149.4/30 is directly connected, FastEthernet0/1.251
     11.0.0.0/32 is subnetted, 1 subnets
B       11.11.11.11 [20/0] via 10.240.149.2, 00:46:31
     192.168.70.0/30 is subnetted, 1 subnets
B       192.168.70.0 [20/0] via 10.239.149.2, 00:46:32
     192.168.71.0/30 is subnetted, 1 subnets
B       192.168.71.0 [20/0] via 10.240.149.2, 00:46:32
```

Pour tester réellement la fonctionnalitée de VPN on a testé la liaison de site client 1 à partir de « RNC » vers son deuxième site SGSN à travers un Ping.

```
RNC#ping 11.11.11.11

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.11.11.11, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 144/204/312 ms
```

Et afin de parcourir le chemin traversé entre ces deux sites on a tapé la commande « traceroute » :

```
RNC#traceroute 11.11.11.11

Type escape sequence to abort.
Tracing the route to 11.11.11.11

  1 10.37.15.6 84 msec
    10.37.15.2 132 msec
    10.37.15.6 72 msec
  2 192.168.21.2 [MPLS: Labels 27/34 Exp 0] 144 msec 248 msec 280 msec
  3 192.168.27.2 [MPLS: Labels 30/34 Exp 0] 200 msec 228 msec 256 msec
  4 192.168.78.1 [MPLS: Labels 19/34 Exp 0] 192 msec 180 msec 168 msec
  5 192.168.65.2 [MPLS: Label 34 Exp 0] 168 msec 180 msec 184 msec
  6 192.168.65.1 248 msec 184 msec 140 msec
```

IV.4.4 Supervision des résultats :

Nous avons utiliser l'outil PRTG pour superviser notre réseau et l'établissement des graphes suivants :

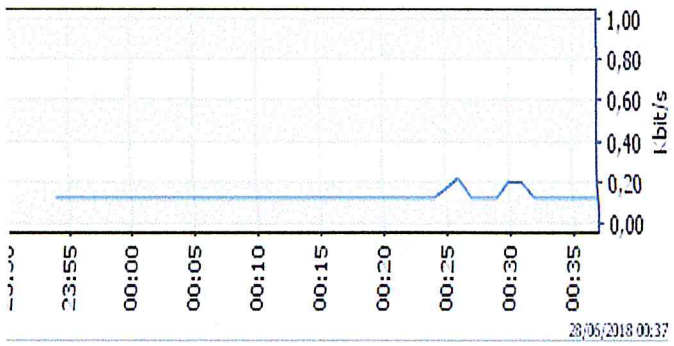


Figure IV.4 : Le débit avant l'implémentation de QoS « trafic entrant »

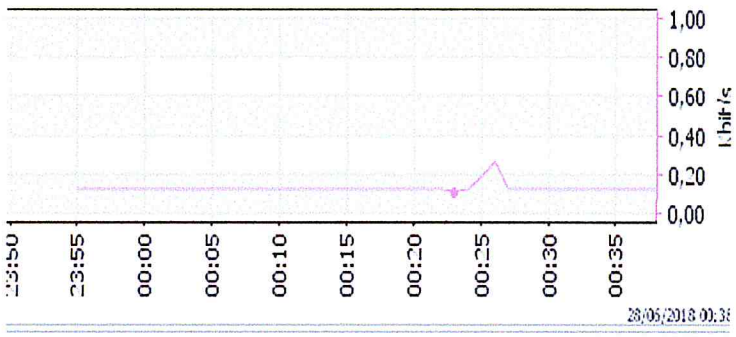


Figure IV.5 :Le débit avant l'implémentation de QoS « trafic sortatnt »

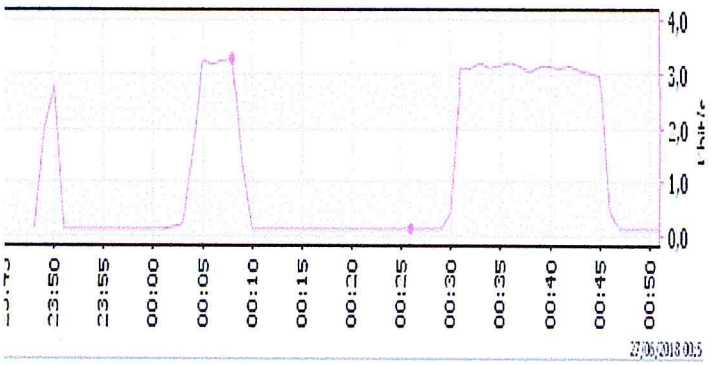


Figure IV.6 :Le débit après l'implémentation de QOS « trafic entrant »

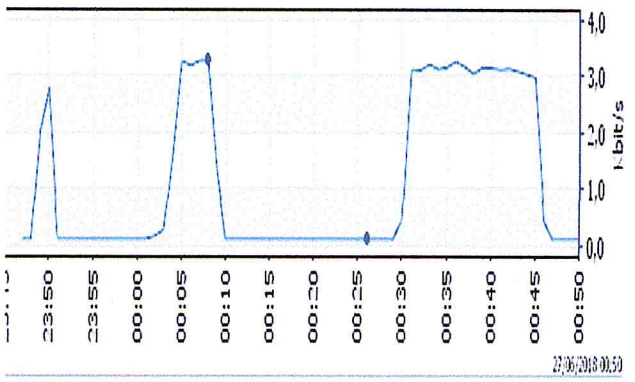


Figure IV.7 :Le débit après l'implémentation de QOS« trafic sortant »

Conclusion :

Cette dernière partie est assez riche d'informations, elle nous a permis de faire une démonstration pratique afin de montrer quelques résultats obtenus lors de notre investigation scientifico-technologique sur le thème **« L'optimisation de la qualité de service sous l'architecture Diffserv/Mpls ».**

Dans ce chapitre, nous avons tous d'abord présenté l'émulateur GNS3 de CISCO et ces paramètres. Il est à signaler que nous avons pu parvenir à utiliser un IOS récent (C3700) après des multiples difficultés que nous avons pu surmonter avec (C7200) en s'appuyant sur plusieurs profondes recherches ainsi qu'une documentation très difficile à trouver.

Ensuite, nous avons conçu une topologie réseau permettant de mettre en œuvre les principales fonctionnalités QoS/MPLS. Une topologie qui a consisté à interconnecter

plusieurs sites d'un réseau opérateur « Ooredoo » utilisant IP/MPLS comme infrastructure de transport. Tout en déployant différentes configuration de la QoS, les VPN et le traffic engeniering à l'aide de l'implémentation de divers protocoles de signalisation et de routage (RSVP-TE, LDP, OSPF et BGP) sous l'outil GNS3 en collaboration avec l'analyseur Wirshark et l'injecteur de trafic Net tester pendant lesquels on a présenté les différents résultats obtenus.

Tous ces phases complémentaires nous ont soutenu à bien assimiler tous ce qu'on a vu précédemment dans la partie théorique et au final, grâce à cette expérience on est devenu expérimenté et capable de réaliser et implémenter une configuration sous une infrastructure réelle avec la coopération de l'équipe IP/MPLS de « Ooredoo ».

Conclusion générale

Conclusion générale :

L'objectif de cette thèse était l'étude de l'optimisation de la qualité de service sur une architecture DiffServ/MPLS qui était faite avec la collaboration de la société «Wataniya Télécom Algérie » d'où on a étudié toutes les étapes citées nécessaire.

Dans une première partie nous avons constaté que la technologie IP/MPLS initialement avait été créée pour améliorer les performances des réseaux haut-débits, elle a pris une place prépondérante dans les réseaux longue distance opérateurs. Son premier but était l'optimisation de temps de traitement des paquets au sein de backbonne notamment en terme de routage ce qui a rendu le transport des flux plus rapide, d'un autre coté on a conclu que IP/MPLS est un système fiable qui permet la mise en place des réseaux privés VPN qui sont des moyens plus souples et plus économiques pour interconnecter un ensemble de sites où la confidentialité est assurée par la préservation des plans de routage (VRF) de chaque clients afin de faciliter un déploiement à grande échelle. Et nous avons également pu constater que l'une des principales applications offerte par IP/MPLS était la réalisation de « *Traffic engineering* » qui est un élément crucial pour un réseau d'opérateur où il permet d'optimiser l'utilisation des ressources d'une infrastructure réseau afin d'éviter la congestion, ainsi le re-routage rapide en cas d'une panne au niveau des liens ou les routeurs grâce au protocole RSVP-TE. A l'issue de ce chapitre, nous avons pu aviser que la technologie IP/MPLS s'était orientée beaucoup plus vers le développement de la qualité de service où on a entamé le second chapitre.

Dans la seconde partie de ce document, nous avons identifié les services qui définissent le réseau « IP/MPLS », et sur lesquels il faut agir pour garantir une qualité de service au niveau des nœuds d'un réseau toute en respectant les métriques qui nous permettent de définir la qualité de service dont le débit, le délai et par conséquent la gigue, ainsi que le taux de pertes des données. Ces éléments sont en effet essentiels pour juger la qualité de transmission des informations et par conséquent, sont primordiaux pour définir la qualité de service.

Par ailleurs, nous avons évoqué les mécanismes de qualité de service à un niveau global, en présentant les architectures proposées par les groupes de travail « IntServ » et «DiffServ », où la première approche définit trois classes de service : le service « best-effort » pour permettre d'acheminer les données sans aucune garantie, le service à charge contrôlée pour offrir une certaine garantie de débit et enfin le service garanti qui permet d'attribuer de meilleures garanties temporelles en débit que les précédents services. Après on a entamé l'approche la plus importante

« DiffServ » qui permet d'effectuer une différenciation de services en se basant sur les agrégats dans lesquels les paquets sont ainsi marqués au niveau de leur en-tête pour leur attribuer des degrés de priorités selon les applications auxquelles ils appartiennent et leurs besoins en débit, délai et/ou pertes. Dans cette architecture, le service « Best Effort » est destiné aux applications à faible priorité, le service « Assured Forwarding » dédié aux applications qui demandent une priorité moyenne, telle la navigation sur le Web et enfin le service à forte priorité « Expedited Forwarding » associé aux applications temps-réel.

.

Afin de finalisé notre étude théorique on a réalisé une étude générale sur le trafic passé par l'opérateur « Ooredoo » pour avoir une vue pratique sur la circulation de flux en temps réel au niveau d'une infrastructure IP/MPLS .

A la fin de ce projet nous avons mis en place une plateforme IP/MPLS où on a déployé toues la partie théorique sous l'outil GNS3 à l'aide de Wireshark en parallèle pendant lesquels nous avons exprimé les différentes configurations implémentées ainsi que l'interprétation de leurs résultats.

Finalement à travers cette expérience nous avons appris beaucoup d'informations concernant les configurations et les outils propre à Cisco, ainsi cette formation nous a ouvert aussi le passage d'accéder au domaine pratique d'où on a travaillé avec une équipe bien expérimentée en IP/MPLS ils nous ont accueilli avec une grande joie dans leur milieu professionnel et ils nous ont transmis suffisamment d'éclaircissement sur leur propre infrastructure réseau.

Annexes

1. Annexe A : Définition les éléments de base de la QoS.

	Définition
Le trafic (flux)	Le terme trafic fait référence à la circulation des flux d'informations sur un réseau informatique.
La congestion	La congestion d'un réseau informatique est le ralentissement global de la circulation du trafic quand il y a augmentation de ce dernier.
La bande passante	La largeur de la plage de fréquences utilisée pour la transmission du signal sur une liaison télécom. Elle est exprimée en Hertz (Hz, Khz, Mhz, Ghz).
Le niveau de service	C'est un indicateur de la qualité de service fonctionnel offert par le réseau
La classe de service	C'est un ensemble de trafics qui partagent des propriétés similaires, qui sont traités de la même façon dans le réseau et avec le même degré de priorité de service.
Ethernet	C'est un protocole de réseau local à commutation de paquets. Bien qu'il implémente la couche physique (PHY) et la sous-couche *Media Access Control* (MAC) du modèle IEEE 802.3, le protocole Ethernet est classé dans les couche de liaison de données (niveau 2) et physique (niveau 1),

2. Annexe B : Définition des termes utilisés dans la thèse.

	Définition
	I
IETF	L'Internet Engineering Task Force, abrégée IETF, littéralement traduit de l'anglais en « Détachement d'ingénierie d'Internet » est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration de standards Internet. L'IETF produit la plupart des nouveaux standards d'Internet.
IOS	IOS abréviation « Internetwork Operating System », système d'exploitation pour la connexion des réseaux, c'est un système d'exploitation propre à

	Cisco Systems et qui équipe la plupart de ses équipements.
IP	Internet Protocol (abrégé en IP) est une famille de protocoles de communication de réseau informatique conçus pour être utilisés par Internet. Les protocoles IP sont au niveau 3 dans le modèle OSI. Les protocoles IP s'intègrent dans la suite des protocoles Internet et permettent un service d'adressage unique pour l'ensemble des terminaux connectés.
	P
PHB	Le comportement par saut (PHB) est un terme utilisé dans des services différenciés (DiffServ) Il définit la politique et de priorité appliquées à un paquet lors de la traversée d'un bond (par exemple un routeur) à dans une architecture DiffServ.
	R
RSVP	Resource ReSerVation Protocol est un protocole de la couche transport du modèle OSI, permettant de réserver des ressources dans un réseau informatique
	S
SLA	Le service level agreement (SLA) est un document qui définit la qualité de service requise entre un prestataire et un client. Le service level agreement, que l'on pourrait traduire en français par « accord de niveau de service », « contrat de niveau de service », « garantie du niveau de service » ou plus simplement « convention de service » est donc un contrat (ou la partie du contrat de service) dans lequel on formalise la qualité du service en question. Dans la pratique, le terme SLA est quelquefois utilisé en référence aux modalités et/ou à la performance (du service) tel que défini dans le contrat.
	V
VoIP	La voix sur réseau IP, parfois appelée téléphonie IP ou téléphonie sur Internet, et souvent abrégée en "VoIP" (abrégé de l'anglais Voice over IP), est une technique qui permet de communiquer par voix à distance via le réseau Internet, ou tout autre réseau acceptant le protocole TCP/IP.

3. Annexe B : Les configurations réalisées sur la maquette.

RNC

```
!
!
!
!
!
!
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RNC
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
```

```
!
!
!
interface Loopback0
 ip address 9.9.9.9 255.255.255.255
 ip ospf 140 area 0
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.105
 encapsulation dot1Q 7
 ip address 10.46.6.1 255.255.255.252
 ip ospf 140 area 0
!
interface FastEthernet0/0.150
 encapsulation dot1Q 1 native
 ip address 10.37.15.1 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/0.151
 encapsulation dot1Q 2
 ip address 10.37.15.5 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/0.250
 encapsulation dot1Q 3
 ip address 10.239.149.1 255.255.255.252
!
interface FastEthernet0/0.252
 encapsulation dot1Q 5
 ip address 10.240.149.1 255.255.255.252
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.106
 encapsulation dot1Q 8
 ip address 10.46.6.5 255.255.255.252
 ip ospf 140 area 0
!
interface FastEthernet0/1.160
 encapsulation dot1Q 1 native
 ip address 10.38.15.1 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/1.161
 encapsulation dot1Q 2
 ip address 10.38.15.5 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/1.251
 encapsulation dot1Q 4
 ip address 10.239.149.5 255.255.255.252
!
interface FastEthernet0/1.252
 encapsulation dot1Q 6
 ip address 10.240.149.5 255.255.255.252
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
```

```
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 110
 log-adjacency-changes
!
router ospf 140
 log-adjacency-changes
!
router bgp 10
 bgp log-neighbor-changes
 neighbor 10.239.149.2 remote-as 100
 neighbor 10.239.149.2 ebgp-multihop 255
 neighbor 10.239.149.6 remote-as 100
 neighbor 10.239.149.6 ebgp-multihop 255
 neighbor 10.240.149.2 remote-as 100
 neighbor 10.240.149.2 ebgp-multihop 255
 neighbor 10.240.149.6 remote-as 100
 neighbor 10.240.149.6 ebgp-multihop 255
 !
 address-family ipv4
  redistribute connected
  neighbor 10.239.149.2 activate
  neighbor 10.239.149.2 send-community extended
  neighbor 10.239.149.2 next-hop-self
  neighbor 10.239.149.6 activate
  neighbor 10.239.149.6 send-community extended
  neighbor 10.239.149.6 next-hop-self
  neighbor 10.240.149.2 activate
  neighbor 10.240.149.2 send-community extended
  neighbor 10.240.149.2 next-hop-self
  neighbor 10.240.149.6 activate
  neighbor 10.240.149.6 send-community extended
  neighbor 10.240.149.6 next-hop-self
  no auto-summary
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
```

```
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

MSS

```
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname MSS
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
```

```
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 ip ospf 120 area 0
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.150
 encapsulation dot1Q 1 native
 ip address 192.168.64.1 255.255.255.252
 ip ospf 120 area 0
!
interface FastEthernet0/0.151
 encapsulation dot1Q 2
 ip address 192.168.70.1 255.255.255.252
 ip ospf 120 area 0
!
interface FastEthernet0/0.250
 encapsulation dot1Q 3
 ip address 192.168.80.1 255.255.255.252
!
interface FastEthernet0/0.251
 encapsulation dot1Q 4
 ip address 192.168.81.1 255.255.255.252
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 120
 log-adjacency-changes
!
router bgp 20
 bgp log-neighbor-changes
 neighbor 192.168.80.2 remote-as 100
 neighbor 192.168.80.2 ebgp-multihop 255
 neighbor 192.168.81.2 remote-as 100
 neighbor 192.168.81.2 ebgp-multihop 255
 !
 address-family ipv4
  redistribute connected
  neighbor 192.168.80.2 activate
  neighbor 192.168.80.2 send-community extended
  neighbor 192.168.80.2 next-hop-self
  neighbor 192.168.81.2 activate
  neighbor 192.168.81.2 send-community extended
  neighbor 192.168.81.2 next-hop-self
  no auto-summary
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
```

```
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
 ip ospf 130 area 0
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.160
 encapsulation dot1Q 2
 ip address 192.168.65.1 255.255.255.252
 ip ospf 130 area 0
!
interface FastEthernet0/0.161
 encapsulation dot1Q 1 native
 ip address 192.168.71.1 255.255.255.252
 ip ospf 130 area 0
!
interface FastEthernet0/0.252
 encapsulation dot1Q 5
 ip address 10.118.34.5 255.255.255.252
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 130
 log-adjacency-changes
!
```

```
router bgp 30
 bgp log-neighbor-changes
 neighbor 10.118.34.2 remote-as 100
 neighbor 10.118.34.2 ebgp-multihop 255
 neighbor 10.118.34.6 remote-as 100
 neighbor 10.118.34.6 ebgp-multihop 255
 !
 address-family ipv4
  redistribute connected
  neighbor 10.118.34.2 activate
  neighbor 10.118.34.2 send-community extended
  neighbor 10.118.34.2 next-hop-self
  neighbor 10.118.34.6 activate
  neighbor 10.118.34.6 send-community extended
  neighbor 10.118.34.6 next-hop-self
  no auto-summary
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

PE_1

```
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
```

```
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
ip vrf Mamagement
 bgp next-hop Loopback14
!
ip vrf Management
 rd 65000:4
 route-target export 65000:4
 route-target import 65000:4
!
ip vrf Media
 rd 65000:3
 route-target export 65000:3
 route-target import 65000:3
 bgp next-hop Loopback13
!
ip vrf SS7
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
 bgp next-hop Loopback11
!
ip vrf Voice
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
 bgp next-hop Loopback12
!
no ip domain lookup
!
mpls label range 1000 1499
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
```

```
!
ip tcp synwait-time 5
!
class-map match-all voix
 match access-group 30
class-map match-all data
 match access-group 40
class-map match-all management
 match access-group 50
class-map match-all signalisation
 match access-group 10
 match access-group 20
!
!
policy-map policy
 class signalisation
  set mpls experimental topmost 5
 class voix
  set mpls experimental topmost 5
 class data
  set mpls experimental topmost 2
 class management
  set mpls experimental topmost 4
!
!
!
!
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ip ospf 1 area 0
!
interface Loopback11
 ip address 11.11.11.1 255.255.255.255
!
interface Loopback12
 ip address 12.12.12.1 255.255.255.255
!
interface Loopback13
 ip address 13.13.13.1 255.255.255.255
!
interface Loopback14
 ip address 14.14.14.1 255.255.255.255
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 4.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 5 5
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng affinity 0x1 mask 0xF
 tunnel mpls traffic-eng path-option 1 dynamic
 tunnel mpls traffic-eng path-option 2 explicit name back
 no routing dynamic
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.105
 encapsulation dot1Q 7
 ip vrf forwarding Management
 ip address 10.46.6.2 255.255.255.252
 ip ospf 140 area 0
!
interface FastEthernet0/0.150
 encapsulation dot1Q 1 native
```

```
 ip vrf forwarding SS7
 ip address 10.37.15.2 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/0.151
 encapsulation dot1Q 2
 ip vrf forwarding SS7
 ip address 10.37.15.6 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/0.250
 encapsulation dot1Q 3
 ip vrf forwarding Voice
 ip address 10.239.149.2 255.255.255.252
!
interface FastEthernet0/0.252
 encapsulation dot1Q 5
 ip vrf forwarding Media
 ip address 10.240.149.2 255.255.255.252
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 192.168.21.3 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 service-policy output policy
 ip rsvp bandwidth 500
 ip rsvp resource-provider none
!
interface FastEthernet2/0
 ip address 192.168.51.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 140 vrf Management
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 110 vrf SS7
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.21.0 0.0.0.255 area 0
 network 192.168.43.0 0.0.0.255 area 0
 network 192.168.51.0 0.0.0.0 area 0
 network 192.168.51.0 0.0.0.255 area 0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 6.6.6.6 remote-as 100
 neighbor 6.6.6.6 update-source Loopback0
 !
 address-family ipv4
```

```
 redistribute connected
 neighbor 3.3.3.3 activate
 neighbor 6.6.6.6 activate
 no auto-summary
 no synchronization
exit-address-family
!
address-family vpnv4
 neighbor 3.3.3.3 activate
 neighbor 3.3.3.3 send-community extended
 neighbor 3.3.3.3 next-hop-self
 neighbor 6.6.6.6 activate
 neighbor 6.6.6.6 send-community extended
 neighbor 6.6.6.6 next-hop-self
exit-address-family
!
address-family ipv4 vrf Voice
 redistribute connected
 neighbor 10.239.149.1 remote-as 10
 neighbor 10.239.149.1 ebgp-multihop 255
 neighbor 10.239.149.1 activate
 neighbor 10.239.149.1 send-community extended
 neighbor 10.239.149.1 next-hop-self
 no synchronization
exit-address-family
!
address-family ipv4 vrf SS7
 redistribute ospf 110 vrf SS7
 no synchronization
exit-address-family
!
address-family ipv4 vrf Media
 redistribute connected
 neighbor 10.240.149.1 remote-as 10
 neighbor 10.240.149.1 ebgp-multihop 255
 neighbor 10.240.149.1 activate
 neighbor 10.240.149.1 send-community extended
 neighbor 10.240.149.1 next-hop-self
 no synchronization
exit-address-family
!
address-family ipv4 vrf Management
 redistribute ospf 140 vrf Management
 no synchronization
exit-address-family
!
ip forward-protocol nd
ip route 21.21.21.1 255.255.255.255 Tunnel1
ip route 22.22.22.1 255.255.255.255 Tunnel1
ip route 23.23.23.1 255.255.255.255 Tunnel1
ip route 24.24.24.1 255.255.255.255 Tunnel1
!
!
no ip http server
no ip http secure-server
!
ip explicit-path name backup enable
 next-address 5.5.5.5
 next-address 6.6.6.6
 next-address 7.7.7.7
 next-address 8.8.8.8
 next-address 4.4.4.4
!
ip explicit-path name back enable
 next-address 192.168.51.1
 next-address 192.168.51.2
 next-address 192.168.56.1
 next-address 192.168.56.2
 next-address 192.168.67.1
```

```
 next-address 192.168.67.2
 next-address 192.168.78.2
 next-address 192.168.78.1
 next-address 192.168.48.2
 next-address 192.168.48.1
!
access-list 10 permit 10.37.15.0 0.0.0.252
access-list 20 permit 10.37.15.0 0.0.0.252
access-list 30 permit 10.239.149.0 0.0.0.252
access-list 40 permit 10.240.149.0 0.0.0.252
access-list 50 permit 10.46.6.0 0.0.0.252
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

PE-2

```
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
```

```
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 192.168.37.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
interface FastEthernet0/1
 ip address 192.168.32.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 1024
!
interface FastEthernet1/0
 ip address 192.168.43.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 1024
!
interface FastEthernet2/0
 ip address 192.168.36.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
!
router ospf 1
 mpls traffic-eng router-id Loopback0
```

```
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.32.0 0.0.0.255 area 0
 network 192.168.36.0 0.0.0.255 area 0
 network 192.168.37.0 0.0.0.255 area 0
 network 192.168.43.0 0.0.0.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 5.5.5.5 remote-as 100
 neighbor 5.5.5.5 update-source Loopback0
 neighbor 6.6.6.6 remote-as 100
 neighbor 6.6.6.6 update-source Loopback0
 neighbor 8.8.8.8 remote-as 100
 neighbor 8.8.8.8 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 1.1.1.1 route-reflector-client
  neighbor 1.1.1.1 next-hop-self
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
  neighbor 4.4.4.4 route-reflector-client
  neighbor 4.4.4.4 next-hop-self
  neighbor 5.5.5.5 activate
  neighbor 5.5.5.5 send-community extended
  neighbor 5.5.5.5 route-reflector-client
  neighbor 5.5.5.5 next-hop-self
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-community extended
  neighbor 6.6.6.6 route-reflector-client
  neighbor 6.6.6.6 next-hop-self
  neighbor 8.8.8.8 activate
  neighbor 8.8.8.8 send-community extended
  neighbor 8.8.8.8 route-reflector-client
  neighbor 8.8.8.8 next-hop-self
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
```

```
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

PE-3

```
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
ip vrf Management
 rd 65000:4
 route-target export 65000:4
 route-target import 65000:4
 bgp next-hop Loopback34
!
ip vrf Media
 rd 65000:3
 route-target export 65000:3
 route-target import 65000:3
 bgp next-hop Loopback33
!
ip vrf SS7
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
 bgp next-hop Loopback31
!
ip vrf Voice
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
 bgp next-hop Loopback32
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
```

```
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
class-map match-all voix
 match access-group 30
class-map match-all data
 match access-group 40
class-map match-all management
 match access-group 50
class-map match-all singnalisation
 match access-group 10
 match access-group 20
class-map match-all signalisation
 match access-group 10
!
!
policy-map policy
 class signalisation
  set mpls experimental topmost 5
 class voix
  set mpls experimental topmost 5
 class data
  set mpls experimental topmost 2
 class management
  set mpls experimental topmost 4
!
!
!
!
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
 ip ospf 1 area 0
!
interface Loopback31
 ip address 31.31.31.1 255.255.255.255
!
interface Loopback32
 ip address 32.32.32.1 255.255.255.255
!
interface Loopback33
 ip address 33.33.33.1 255.255.255.255
!
interface Loopback34
 ip address 34.34.34.1 255.255.255.255
!
interface Tunnel2
```

```
 ip unnumbered Loopback0
 tunnel destination 8.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 5 5
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng affinity 0x1 mask 0xD
 tunnel mpls traffic-eng path-option 1 dynamic
 no routing dynamic
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.160
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.106
 encapsulation dot1Q 8
 ip vrf forwarding Management
 ip address 10.46.6.6 255.255.255.252
 ip ospf 140 area 0
!
interface FastEthernet0/1.160
 encapsulation dot1Q 1 native
 ip vrf forwarding SS7
 ip address 10.38.15.2 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/1.161
 encapsulation dot1Q 2
 ip vrf forwarding SS7
 ip address 10.38.15.6 255.255.255.252
 ip ospf 110 area 0
!
interface FastEthernet0/1.251
 encapsulation dot1Q 4
 ip vrf forwarding Voice
 ip address 10.239.149.6 255.255.255.252
!
interface FastEthernet0/1.252
 encapsulation dot1Q 6
 ip vrf forwarding Media
 ip address 10.240.149.6 255.255.255.252
!
interface FastEthernet1/0
 ip address 192.168.56.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 service-policy output policy
 ip rsvp bandwidth 500
!
interface FastEthernet2/0
 ip address 192.168.51.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 140 vrf Management
```

```
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 110 vrf SS7
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.51.0 0.0.0.255 area 0
 network 192.168.56.0 0.0.0.255 area 0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 6.6.6.6 remote-as 100
 neighbor 6.6.6.6 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  redistribute static
  neighbor 3.3.3.3 activate
  neighbor 6.6.6.6 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 next-hop-self
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-community extended
  neighbor 6.6.6.6 next-hop-self
 exit-address-family
 !
 address-family ipv4 vrf Voice
  redistribute connected
  redistribute static
  neighbor 10.239.149.5 remote-as 10
  neighbor 10.239.149.5 ebgp-multihop 255
  neighbor 10.239.149.5 activate
  neighbor 10.239.149.5 send-community extended
  neighbor 10.239.149.5 next-hop-self
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf SS7
  redistribute ospf 110 vrf SS7
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf Media
  redistribute connected
  neighbor 10.240.149.5 remote-as 10
  neighbor 10.240.149.5 ebgp-multihop 255
  neighbor 10.240.149.5 activate
  neighbor 10.240.149.5 send-community extended
  neighbor 10.240.149.5 next-hop-self
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf Management
  redistribute ospf 140 vrf Management
  no synchronization
 exit-address-family
```

```
!
ip forward-protocol nd
ip route 41.41.41.1 255.255.255.255 Tunnel2
ip route 42.42.42.1 255.255.255.255 Tunnel2
ip route 43.43.43.1 255.255.255.255 Tunnel2
ip route 44.44.44.1 255.255.255.255 Tunnel2
!
!
no ip http server
no ip http secure-server
!
access-list 10 permit 10.38.15.0 0.0.0.252
access-list 20 permit 10.38.15.0 0.0.0.252
access-list 30 permit 10.239.149.0 0.0.0.252
access-list 40 permit 10.239.240.0 0.0.0.252
access-list 50 permit 10.46.6.0 0.0.0.252
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

PE_4

```
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
```

```
!
!
ip vrf Management
 rd 65000:4
 route-target export 65000:4
 route-target import 65000:4
 bgp next-hop Loopback44
!
ip vrf Media
 rd 65000:3
 route-target export 65000:3
 route-target import 65000:3
 bgp next-hop Loopback43
!
ip vrf SS7
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
 bgp next-hop Loopback41
!
ip vrf Voice
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
 bgp next-hop Loopback42
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
class-map match-all voix
 match access-group 30
class-map match-all data
 match access-group 40
class-map match-all management
 match access-group 50
class-map match-all signalisation
 match access-group 10
 match access-group 20
```

```
!
!
policy-map policy
 class signalisation
  set mpls experimental topmost 5
 class voix
  set mpls experimental topmost 5
 class data
  set mpls experimental topmost 2
 class management
  set mpls experimental topmost 4
!
!
!
!
!
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
 ip ospf 1 area 0
!
interface Loopback41
 ip address 41.41.41.1 255.255.255.255
!
interface Loopback42
 ip address 42.42.42.1 255.255.255.255
!
interface Loopback43
 ip address 43.43.43.1 255.255.255.255
!
interface Loopback44
 ip address 44.44.44.1 255.255.255.255
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 5.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 5 5
 tunnel mpls traffic-eng bandwidth  500
 tunnel mpls traffic-eng affinity 0x1 mask 0xD
 tunnel mpls traffic-eng path-option 1 dynamic
 no routing dynamic
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.106
 encapsulation dot1Q 7
 ip vrf forwarding Management
 ip address 10.242.149.6 255.255.255.252
 ip ospf 150 area 0
!
interface FastEthernet0/0.151
 encapsulation dot1Q 2
 ip vrf forwarding SS7
 ip address 192.168.70.2 255.255.255.252
 ip ospf 120 area 0
!
interface FastEthernet0/0.161
 encapsulation dot1Q 1 native
 ip vrf forwarding SS7
 ip address 192.168.71.2 255.255.255.252
 ip ospf 130 area 0
!
interface FastEthernet0/0.251
 encapsulation dot1Q 4
 ip vrf forwarding Voice
```

```
 ip address 192.168.81.2 255.255.255.252
!
interface FastEthernet0/0.252
 encapsulation dot1Q 5
 ip vrf forwarding Media
 ip address 10.118.34.6 255.255.255.252
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 192.168.78.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 service-policy output policy
 ip rsvp bandwidth 500
!
interface FastEthernet2/0
 ip address 192.168.48.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 150 vrf Management
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 120 vrf SS7
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.48.0 0.0.0.255 area 0
 network 192.168.78.0 0.0.0.255 area 0
!
router ospf 130 vrf SS7
 log-adjacency-changes
 redistribute bgp 100 subnets
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 6.6.6.6 remote-as 100
 neighbor 6.6.6.6 update-source Loopback0
 !
 address-family ipv4
  neighbor 3.3.3.3 activate
  neighbor 6.6.6.6 activate
  no auto-summary
  no synchronization
 exit-address-family
 !
 address-family vpnv4
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 next-hop-self
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-community extended
```

```
 neighbor 6.6.6.6 next-hop-self
exit-address-family
!
address-family ipv4 vrf Voice
 redistribute connected
 neighbor 192.168.81.1 remote-as 20
 neighbor 192.168.81.1 ebgp-multihop 255
 neighbor 192.168.81.1 activate
 neighbor 192.168.81.1 send-community extended
 neighbor 192.168.81.1 next-hop-self
 no synchronization
exit-address-family
!
address-family ipv4 vrf SS7
 redistribute ospf 120 vrf SS7
 redistribute ospf 130 vrf SS7
 no synchronization
exit-address-family
!
address-family ipv4 vrf Media
 redistribute connected
 neighbor 10.118.34.5 remote-as 30
 neighbor 10.118.34.5 ebgp-multihop 255
 neighbor 10.118.34.5 activate
 neighbor 10.118.34.5 send-community extended
 neighbor 10.118.34.5 next-hop-self
 no synchronization
exit-address-family
!
address-family ipv4 vrf Management
 redistribute ospf 150 vrf Management
 no synchronization
exit-address-family
!
ip forward-protocol nd
ip route 31.31.31.1 255.255.255.255 Tunnel2
ip route 32.32.32.1 255.255.255.255 Tunnel2
ip route 33.33.33.1 255.255.255.255 Tunnel2
ip route 34.34.34.1 255.255.255.255 Tunnel2
!
!
no ip http server
no ip http secure-server
!
access-list 10 permit 192.168.70.0 0.0.0.252
access-list 20 permit 192.168.71.0 0.0.0.252
access-list 30 permit 192.168.81.0 0.0.0.252
access-list 40 permit 10.118.34.0 0.0.0.252
access-list 50 permit 10.242.149.0 0.0.0.252
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
```

```
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

P_1

```
!
!
!
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
```

```
!
ip tcp synwait-time 5
!
!
!
!
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 192.168.26.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
interface FastEthernet0/1
 ip address 192.168.32.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 500
!
interface FastEthernet1/0
 ip address 192.168.21.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 500
!
interface FastEthernet2/0
 ip address 192.168.27.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.21.0 0.0.0.255 area 0
 network 192.168.26.0 0.0.0.255 area 0
 network 192.168.27.0 0.0.0.255 area 0
 network 192.168.32.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
```

```
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end
```

P_2

```
PE-A#show running-config
Building configuration...

Current configuration : 4827 bytes
!
upgrade fpd auto
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE-A
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
ip source-route
ip cef
!
!
!
!
ip vrf client-A
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
!
ip vrf client-B
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
!
ip vrf client-C
 rd 65000:3
 route-target export 65000:3
 route-target import 65000:3
!
no ipv6 cef
multilink bundle-name authenticated
mpls traffic-eng tunnels
mpls label protocol ldp
!
!
voice dsp waitstate 0
!
```

```
!
!
!
!
!
!
!
!
!
!
memory-size iomem 0
archive
 log config
  hidekeys
!
!
!
class-map match-all multimedia
 match any
 match protocol http
 match protocol ospf
class-map match-all voice
 match any
 match protocol pop3
 match protocol ftp
 match protocol ldap
class-map match-all signalisation
 match any
 match protocol icmp
 match protocol rsvp
 match protocol ospf
 match protocol bgp
!
!
policy-map QOS
 class class-default
   bandwidth 1544
   fair-queue 128
   queue-limit 32 packets
   police 1544000 1540000 1543000 conform-action transmit  exceed-action drop  violate-action drop
  set dscp cs7
  set mpls experimental topmost 5
    random-detect
!
!
!
!
!
interface Loopback0
 ip address 2.2.2.2 255.0.0.0
 ip ospf 6500 area 0
!
interface Tunnel100
 description path PE-A=>P-1=>P-2=>PE-B
 ip unnumbered Loopback0
 ip ospf interface-retry 0
 tunnel destination 4.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 512
 tunnel mpls traffic-eng path-option 1 explicit name A->B
 tunnel mpls traffic-eng record-route
 tunnel mpls traffic-eng fast-reroute
 no routing dynamic
!
interface Tunnel200
```

```
 description path PE-A=>P-1=>P-3=>PE-C
 ip unnumbered Loopback0
 ip ospf interface-retry 0
 tunnel destination 3.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 512
 tunnel mpls traffic-eng path-option 10 explicit name A=>c
 tunnel mpls traffic-eng record-route
 tunnel mpls traffic-eng fast-reroute
 no routing dynamic
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface Serial1/0
 ip vrf forwarding client-A
 ip address 192.168.6.3 255.255.255.0
 ip ospf 100 area 0
 serial restart-delay 0
 service-policy output QOS
!
interface Serial1/1
 ip address 192.168.10.3 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 serial restart-delay 0
 service-policy output QOS
 ip rsvp bandwidth
!
interface Serial1/2
 description PATH PE-A=>CE-C-ALGER
 ip vrf forwarding client-C
 ip address 192.168.12.3 255.255.255.0
 ip ospf 300 area 0
 serial restart-delay 0
!
interface Serial1/3
 description PATH PE-A=>CE-B-ALGER
 ip vrf forwarding client-B
 ip address 192.168.11.3 255.255.255.0
 ip ospf 200 area 0
 serial restart-delay 0
!
router ospf 100 vrf client-A
 router-id 192.168.6.3
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 300 vrf client-C
 router-id 192.168.12.3
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 200 vrf client-B
 router-id 192.168.11.3
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 6500
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.6.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
!
```

```
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 65000
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 65000
 neighbor 4.4.4.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf client-C
  redistribute ospf 300 vrf client-C
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf client-B
  redistribute ospf 200 vrf client-B
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf client-A
  redistribute ospf 100 vrf client-A
  no synchronization
 exit-address-family
!
ip forward-protocol nd
ip route 3.3.3.3 255.255.255.255 Tunnel200
no ip http server
no ip http secure-server
!
!
!
ip explicit-path name A->B enable
 next-address 192.168.10.2
 next-address 192.168.4.3
 next-address 192.168.8.2
!
ip explicit-path name A=>c enable
 next-address 5.5.5.5
 next-address 6.6.6.6
 next-address 3.3.3.3
!
logging alarm informational
!
!
!
!
control-plane
!
!
!
mgcp fax t38 ecm
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 stopbits 1
line aux 0
```

```
 stopbits 1
line vty 0 4
 login
!
End
```

P_3

```
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P3
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
```

```
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 192.168.26.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
interface FastEthernet0/1
 ip address 192.168.67.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 500
!
interface FastEthernet1/0
 ip address 192.168.56.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 500
!
interface FastEthernet2/0
 ip address 192.168.36.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.26.0 0.0.0.255 area 0
 network 192.168.36.0 0.0.0.255 area 0
 network 192.168.56.0 0.0.0.255 area 0
 network 192.168.67.0 0.0.0.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 100
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 3.3.3.3 remote-as 100
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 5.5.5.5 remote-as 100
 neighbor 5.5.5.5 update-source Loopback0
 neighbor 8.8.8.8 remote-as 100
 neighbor 8.8.8.8 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 1.1.1.1 activate
  neighbor 1.1.1.1 send-community extended
  neighbor 1.1.1.1 route-reflector-client
  neighbor 1.1.1.1 next-hop-self
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
  neighbor 3.3.3.3 route-reflector-client
  neighbor 3.3.3.3 next-hop-self
```

```
 neighbor 4.4.4.4 activate
 neighbor 4.4.4.4 send-community extended
 neighbor 4.4.4.4 route-reflector-client
 neighbor 4.4.4.4 next-hop-self
 neighbor 5.5.5.5 activate
 neighbor 5.5.5.5 send-community extended
 neighbor 5.5.5.5 route-reflector-client
 neighbor 5.5.5.5 next-hop-self
 neighbor 8.8.8.8 activate
 neighbor 8.8.8.8 send-community extended
 neighbor 8.8.8.8 route-reflector-client
 neighbor 8.8.8.8 next-hop-self
exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
End
```

P_4

```
!
!
!

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname P4
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
```

```
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
!
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
!
!
!
!
interface Loopback0
 ip address 7.7.7.7 255.255.255.255
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 192.168.37.1 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
interface FastEthernet0/1
 ip address 192.168.67.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x2
 ip rsvp bandwidth 500
!
interface FastEthernet1/0
 ip address 192.168.78.2 255.255.255.0
 duplex auto
 speed auto
```

```
 mpls ip
 mpls traffic-eng tunnels
 mpls traffic-eng attribute-flags 0x1
 ip rsvp bandwidth 500
!
interface FastEthernet2/0
 ip address 192.168.27.2 255.255.255.0
 duplex auto
 speed auto
 mpls ip
 mpls traffic-eng tunnels
 ip rsvp bandwidth 500
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 192.168.27.0 0.0.0.255 area 0
 network 192.168.37.0 0.0.0.255 area 0
 network 192.168.67.0 0.0.0.255 area 0
 network 192.168.78.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
End
```

La bibliographie

[1] : Claude Servin , *Réseaux & Télécoms,* 2ème édition, Éditions DUNOD,Paris,France,2003.

[2] : Guy Pujolle, *Les Réseaux*, 6ème édition, Éditions EYROLLES, Paris, France, 2008.

[3] : Ivan Pepelnjak, Jim Guichard, *Architectures MPLS et VPN*, Éditions Cisco Systems.

[4] : Khodor ABBOUD : *Conception et évaluation d?un modèle adaptatif pour la qualité de service dans les réseaux MPLS,* Doctorat, École centrale de Lille, 2010.

[5] : Traffic Engineering with MPLS, ISBN : 1-58705-031-5 : CISCO : Livre très complet sur la QoS dans MPLS et sur l'ingénierie de trafic.

[6] : J.A. García-Macías, F. Rousseau, G. Berger-Sabbatel, L. Toumi, et Andrzej Duda, "Différenciation des services sur les réseaux sans fil 802.11", Colloque Francophone sur l'Ingénierie des Protocoles, 27-30 Mai 2002, Montréal, Canada

[7] : Rami LANGAR ; Mécanismes de Gestion de la Mobilité et Evaluation de Performance dans les Réseaux Cellulaires tout-IP ; Thèse Doctorat ; 2006.

[8] : Oussama FOUDHAILI ; Analyse des performances de MPLS en terme de Traffic Engineering dans un réseau multiservice ; Projet de Fin d'Etudes en Ingénierie des Réseaux ; 2004

[9] : T. Nandagopal, N. Venkitaraman, R. Sivakumar and V. Barghavan, "Delay Differentiation and Adaptation in Core Stateless Networks", In Proceedings of IEEE INFOCOM 2000, pp. 421-430, Tel-Aviv, Israel, April 2000.

[18]: Abdeljalil SAIKA; Contribution à la gestion de la qualité de service (QOS) et à l'optimisation du trafic dans les Réseaux IP-MPLS ;Thèse Doctorat ; 2012.

La webographie

[10] : http://www.cisco.com/en/US/docs/ios/mpls/command/reference/mp_book.html

[11] : http://fr.scribd.com/doc/80398983/mpls

[12] : http://www.memoireonline.com/03/11/4293/m_Mise-en-oeuvre-dun-coeur-de-reseau-IPMPLS10.html

[13] : http://www-igm.univ-mlv.fr/~dr/XPOSE2006/marot/index.html

[14] : http://www.bonnefoy.eu/25_OSPF

[15] : http://www.httr.ups-tlse.fr/pedagogie/cours/tcp-ip/diffserv/index.html

[16] : http://www-r2.u-strasbg.fr/~pansiot/enseignement/Master%20Recherche/QoSrouteur.pdf

[17] : https://www.renater.fr/IMG/pdf/Memoire_IRSM_Nicolas_Garnier-2.pdf